# GRAPH LOGIC FLOWS: GLOBAL CONSTRAINED TRANSPORT FOR DYNAMIC GRAPH PREDICTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Temporal graph models typically hide each state transition inside memory, attention, or neighbourhood operations, making the transition difficult to inspect as a single computational object. **Graph Logic Flows (GLF)** takes a different view: each snapshot transition is defined by one global constrained transport problem on a learned graph geometry. GLF transports a node belief distribution over learned positive edge weights; temporal total variation acts directly on the state update, while Bellman consistency shapes the learned graph potential during training. Solving the transport problem produces the next belief state and edge transport variable together with numerical and structural diagnostics of the same transition. We derive a finite-step bound for the state-coupled residual penalty and specialise standard Wasserstein stability results to the learned geometry. On the JODIE Wikipedia, Reddit, and LastFM benchmarks under a 100-negative full-node protocol, GLF has mean MRR close to TGN on Wikipedia and LastFM and higher than TGN on Reddit. Residual ablations show that the structural terms materially affect prediction, while fixed-probe rankings recover the temporal smoothness induced by TV. On Wikipedia, GLF trains substantially faster than TGAT and DyGFormer, and the matched single-file implementation processes the tested snapshots faster than all three baseline reproductions. Together, these results suggest a different design principle for temporal graph learning: make the state transition an explicit, solved object so that prediction and evidence about how the state changed arise from the same update.

## 1 INTRODUCTION

Every temporal graph model must decide how a new snapshot changes its internal state. TGN carries that change through event-driven memory, TGAT through temporal attention, and DyGFormer through attention over interaction histories (Rossi et al., 2020; Xu et al., 2020; Yu et al., 2023). In these architectures, the transition is distributed across recurrent, sampled, or local operations. The model produces a new state, but the transition itself is not represented as one solved object whose behaviour can be examined directly.

**Graph Logic Flows (GLF)** takes the snapshot transition itself as that object. GLF maintains a belief distribution $\mu_t$ over nodes and computes the next state by transporting belief mass over the current graph. Learned positive edge weights define a snapshot-specific transport geometry, while the edge transport variable and next belief state are solved jointly in one global constrained update rather than assembled through a stack of local propagation steps. The same optimisation that produces the state also exposes quantities describing transport effort, numerical optimality, energy change, and structural residual behaviour.

Structural residuals play distinct roles in this update. We use a Bellman residual for local potential consistency against observed edge weights and a temporal total-variation residual for change in the belief state. Temporal TV is coupled directly to the JKO state variables and therefore acts on the solved transition. Bellman consistency instead shapes the learned graph potential during training and remains a monitored structural quantity once the parameters are fixed. Link prediction is computed from the resulting belief state together with the learned scorer.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

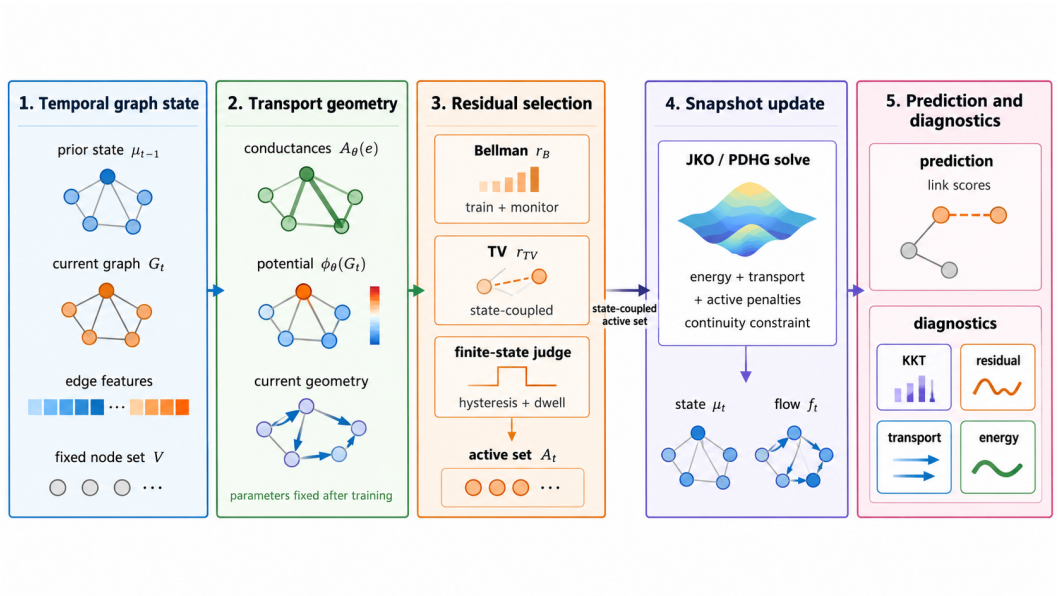


Figure 1: **Overview of Graph Logic Flows at one snapshot.** GLF combines the previous belief state and current graph to construct a snapshot-specific transport geometry. Bellman consistency shapes and monitors the learned graph potential, while temporal TV is coupled to the state update. The JKO/PDHG solve produces the updated belief state and edge transport variable, from which link predictions and solver diagnostics are obtained.

Figure 1 summarises the complete snapshot update: the current graph defines the transport geometry, the two residuals enter through their implemented roles, and one global solve produces the state used for prediction together with solver diagnostics.

GLF uses established optimal transport tools to define this temporal update. The implicit step follows the JKO view of Wasserstein gradient flow (Jordan et al., 1998; Ambrosio et al., 2005), and its constrained-flow interpretation is related to reflected Wasserstein dynamics (Maury et al., 2011). GRAND and CTAN provide neighbouring continuous-dynamics views of graph propagation (Chamberlain et al., 2021; Gravina et al., 2024). Here, learned geometry, structural residuals, link scoring, and solver diagnostics all refer to the same snapshot transition. The theory follows the same division: standard Wasserstein and graph-structured arguments describe contraction, tracking, and sensitivity in the learned geometry, while our finite-step result characterises the state-coupled penalty used by the implemented update.

The experiments examine the same transition from prediction, mechanism, and computational perspectives. Under the reported 100-negative full-node protocol, GLF has mean MRR close to TGN on Wikipedia and LastFM and higher than TGN on Reddit. Fixed-probe analysis recovers the temporal smoothness encouraged by TV, while residual ablations show that the structural terms materially affect prediction. On Wikipedia, GLF also trains substantially faster than TGAT and DyGFormer, and the matched single-file inference implementations show a fast complete snapshot update. Solver diagnostics and the fixed-window history experiment further characterise how the transport update behaves across runs and training histories.

**Contributions.** Our contributions are as follows.

- We formulate the transition from one temporal graph snapshot to the next as a single global constrained transport problem. Learned edge weights define a snapshot-specific geometry, and a JKO/PDHG solve produces the updated belief state and edge transport variable.
- We connect prediction with structural and numerical inspection of the same transition. Temporal TV acts directly on the state update, Bellman consistency shapes the learned

potential during training, and the solve exposes diagnostics including residual behaviour, transport action, energy change, and numerical optimality.

- We study the resulting update empirically and theoretically. On the reported JODIE benchmarks, GLF supports competitive dynamic link prediction and fast snapshot updates on Wikipedia; the accompanying analysis gives fixed-phase stability, finite-penalty residual control, and bounded tracking under controlled target drift.

## 2 Related Work

**Temporal graph learning.** Temporal graph models operating in continuous time differ mainly in how they retain and retrieve history. TGN uses event driven node memories (Rossi et al., 2020), TGAT uses temporal attention (Xu et al., 2020), DyGFormer uses patched self attention over interaction histories (Yu et al., 2023), and DyRep, CAW, TCL, and GraphMixer provide point process, walk based, contrastive, and MLP based alternatives (Trivedi et al., 2019; Xu et al., 2020; Wang et al., 2021; Cong et al., 2023). Evaluation work has also shown that temporal link prediction is sensitive to recurrence and negative sampling: EdgeBank exposes the strength of memorisation baselines, while TGB provides broader datasets and more realistic protocols (Poursafaei et al., 2022; Huang et al., 2023). GLF leaves the prediction task unchanged but changes how the temporal state is updated. History is carried by the evolving belief state, and each new state is obtained from a global transport solve.

**Continuous graph dynamics and optimal transport.** GRAND views graph learning as continuous diffusion and studies stable discretisations (Chamberlain et al., 2021). CTAN targets long range propagation in continuous time dynamic graphs through antisymmetric dynamics (Gravina et al., 2024). GLF instead uses an implicit optimal transport update on a learned edge geometry. Its JKO discretisation and contraction arguments draw on standard Wasserstein gradient flow theory (Jordan et al., 1998; Ambrosio et al., 2005), while the constrained interpretation is related to Wasserstein crowd motion models (Maury et al., 2011). In GLF, however, the JKO step is the temporal prediction update itself, and the quantities produced by the solve remain available for inspection.

**Long range and constraint aware graph learning.** Oversquashing formalises how distant information can be compressed by repeated local aggregation (Alon & Yahav, 2020; Banerjee et al., 2022). Rewiring and virtual node methods address the problem by changing the communication graph (Attali et al., 2024; Karhadkar et al., 2022; Southern et al., 2024). GLF takes a different route: it preserves the observed edge set $E_t$ and recomputes transport conductances on the current edges as the graph changes. Constraint aware and algorithmic graph learning inject known structure into the computation (Xu et al., 2019). GLF follows that motivation by placing Bellman and temporal TV residuals inside the snapshot update and selecting their activation at runtime.

## 3 Problem Setup

GLF describes a snapshot transition using three ingredients: a belief state that evolves over time, a transport geometry that determines how that state can move over the graph, and structural residuals that measure whether the update respects the conditions we wish to monitor. We introduce these objects in that order.

At snapshot $t$, let $G_t = (V, E_t, X_t)$ be a directed graph with $n = |V|$ nodes and $m_t = |E_t|$ edges. Following the JODIE benchmark setting (Kumar et al., 2019), the node set $V$ is fixed across snapshots while the edge set $E_t$ evolves. GLF represents its temporal state by a probability distribution $\mu_t \in \Delta^n$, where

$$\Delta^n = \{\mu \in \mathbb{R}^n_{\geq 0} : \mathbf{1}^\top \mu = 1\}.$$

Thus $\mu_t(i)$ records the belief mass assigned to node $i$ at snapshot $t$. Because consecutive states live on the same simplex, a snapshot update can be viewed as redistributing mass from $\mu_{t-1}$ to $\mu_t$ over the current graph, and temporal change can be measured directly between the two states.

**Transport geometry.** Let $M_t \in \mathbb{R}^{n \times m_t}$ be the signed incidence matrix, with $(M_t)_{i,e} = +1$ and $(M_t)_{j,e} = -1$ for an edge $e = (i \to j)$. For positive edge weights $A : E_t \to \mathbb{R}_{>0}$, let

$$D_A = \mathrm{diag}(A(e)), \qquad q = D_A f, \qquad \mathrm{div}_A(f) = M_t q = M_t D_A f,$$

and define

$$\|f\|_A^2 = \sum_{e \in E_t} \frac{f_e^2}{A(e)}.$$

Here $f \in \mathbb{R}^{m_t}$ is the edge coordinate used by the solver, while $q$ is the corresponding mass flux entering the continuity equation. In flux coordinates,

$$\|f\|_A^2 = \sum_{e \in E_t} \frac{q_e^2}{A(e)^3},$$

so increasing $A(e)$ lowers the action required for a fixed physical flux. Thus $A$ is the learned positive parameter of the transport geometry, with effective flux conductance $A(e)^3$.

For $\mu_0, \mu_1 \in \Delta^n$, we use the corresponding Benamou–Brenier-style dynamic transport objective (Benamou & Brenier, 2000):

$$W_{G_t,A}^2(\mu_0, \mu_1) = \inf_{(\mu_s, f_s)} \int_0^1 \|f_s\|_A^2 \, ds, \qquad \mu_s|_{s=0} = \mu_0, \quad \mu_s|_{s=1} = \mu_1, \quad \frac{d}{ds}\mu_s + \mathrm{div}_A(f_s) = 0. \tag{1}$$

Equivalently, the continuity equation is $\frac{d}{ds}\mu_s + M_t q_s = 0$ with $q_s = D_A f_s$. This makes explicit how the learned edge weights determine the cost of transporting belief mass over the current graph.

**Snapshot energy.** Transport alone determines how expensive a state change is, but it does not specify which state is preferred on the current graph. GLF therefore associates each snapshot with the energy

$$E_\theta(\mu; G) = \mathrm{Ent}(\mu) + \langle \phi_\theta(G), \mu \rangle + \tfrac{\lambda}{2}\|\mu\|_2^2, \qquad \lambda > 0, \tag{2}$$

where $\phi_\theta(G) \in \mathbb{R}^n$ is a learned graph-dependent potential implemented by a small MLP on graph features. The potential makes the preferred belief state depend on the current graph, while the entropy and quadratic terms regularise the state. These terms also provide the convexity structure used later in the JKO update and in the stability analysis.

**Structural residuals.** The final ingredient is a small set of quantities that describe structural behaviour of the snapshot update. Let $[x]_+ = \max(x, 0)$. We use

$$\begin{aligned} B_{\mathrm{Bell}}(\phi, G) &= \max_{(i,j) \in E_t} [\phi_i - \phi_j - w_{ij}]_+, \\ B_{\mathrm{TV}}(\mu_t, \mu_{t-1}) &= \|\mu_t - \mu_{t-1}\|_1. \end{aligned} \tag{3}$$

The Bellman quantity measures local potential inconsistency relative to the observed base edge weights $w_{ij} = w_{\mathrm{base}}(i, j)$. Importantly, these weights are distinct from the learned conductances $A_\theta$: $w_{\mathrm{base}}$ defines the local consistency condition, whereas $A_\theta$ defines the transport geometry.

Computationally, the Bellman signal is formed from $\mathrm{ReLU}(M_t\phi - w_{\mathrm{base}})$ over the top $k$ highest-weight edges, with $k = 2048$ by default, and weighted by $w_{\mathrm{base}} \cdot A_\theta$. The temporal-TV residual instead measures how much the node belief distribution changes between two consecutive snapshots. Both quantities are convex in their relevant arguments, but they enter GLF differently. Temporal TV is coupled directly to the JKO state variables and can therefore affect the solved state, while Bellman consistency shapes the learned graph potential during training and remains a monitored structural quantity at test time.

These definitions separate the three roles used in the next section: $A_\theta$ determines *where* belief can move cheaply, $E_\theta$ determines *which* states are preferred on the current graph, and the residuals describe *which structural conditions* are active or violated during the transition.

# 4 METHOD

At snapshot $t$, GLF takes the previous belief state $\mu_{t-1}$ and the current graph $G_t$, constructs the transport geometry, selects the active structural conditions, solves one global state update, and scores links from the resulting state. The same update also exposes the optimisation quantities used later as diagnostics.

**What is learned and what is solved.** Training learns the graph-dependent networks and the link scorer. Their parameters are fixed after training. At every test snapshot, GLF rebuilds the incidence structure from $E_t$, reevaluates the conductances and graph potential on the current graph features, updates the controller state, and solves for $(\mu_t, f_t)$. Thus, $\theta$ determines how the current graph is mapped to a transport geometry and scoring function, whereas $(\mu_t, f_t)$ are the state and flow recomputed for that snapshot.

## 4.1 LEARN A SNAPSHOT-SPECIFIC TRANSPORT GEOMETRY

A small edge network produces a positive transport weight $A_\theta(e)$ for every $e \in E_t$. Under the flux coordinate $q = D_{A_\theta} f$, this induces the action $q_e^2/A_\theta(e)^3$, so larger $A_\theta(e)$ makes transport along that edge cheaper. The network is trained end to end through the unrolled solver. At each snapshot, the incidence matrix is rebuilt from the current edge set and the network is evaluated on current, degree-based edge features. A newly observed edge therefore receives its transport weight from its current features rather than from a stored edge identity.

The learned parameters remain fixed after training, but the geometry itself is snapshot dependent. Changes in $E_t$ and the current edge features change the conductances, while the graph potential and belief state are reevaluated on the current graph. GLF therefore adapts the solved transition to the observed snapshot without updating $\theta$ at test time.

## 4.2 SELECT THE ACTIVE RESIDUALS

The residual controller tracks the Bellman and temporal-TV signals together with graph drift. Thresholding, top-$K_{\max}$ selection, hysteresis, and dwell time produce an active set $\mathcal{A}_t$. The state-coupled subset $\mathcal{A}_t^{\text{state}}$ enters the JKO penalty. In the present implementation this subset contains temporal TV, while Bellman consistency shapes $\theta$ during training and remains part of the monitored controller state at test time.

**Algorithm 1** Runtime residual controller

1: **Input:** residual margins, graph drift, thresholds, $K_{\max}$, previous controller state
2: Mark residuals eligible when their margin or drift guard crosses threshold
3: **if** more than $K_{\max}$ residuals are eligible **then**
4: Keep the largest normalised margins
5: **end if**
6: Apply hysteresis and the minimum dwell time
7: **Return:** active residual set $\mathcal{A}_t$

## 4.3 SOLVE ONE CONSTRAINED SNAPSHOT UPDATE

For a fixed active set, GLF updates the snapshot by solving

$$\min_{\mu \in \Delta^n,\, f \in \mathbb{R}^{m_t}} \quad E_\theta(\mu; G_t) + \frac{1}{2\tau} \sum_{e \in E_t} \frac{f_e^2}{A_\theta(e)} + \sum_{r \in \mathcal{A}_t^{\text{state}}} \rho_{r,t} \Psi(R_{r,t}) \tag{4}$$

$$\text{s.t.} \quad \mu - \mu_{t-1} + \tau \operatorname{div}_{A_\theta}(f) = 0,$$

where $\tau > 0$ is the snapshot step size, $\Psi(s) = (s_+)^2$, and $\mathcal{A}_t^{\text{state}}$ contains the active residuals coupled to $(\mu, f)$. The simplex constraint is handled by projection, while the continuity equation and active penalties are handled with primal-dual hybrid gradient (PDHG) iterations (Chambolle & Pock, 2011). Equation equation 4 is the computation implemented by GLF; its reflected-Wasserstein interpretation refers to the corresponding continuous-time view rather than to a separate module.

**Algorithm 2** GLF update at snapshot $t$

1: **Input:** $G_t$, previous belief $\mu_{t-1}$, trained parameters $\theta$
2: Compute conductances $A_\theta(E_t)$ and potential $\phi_\theta(G_t)$
3: Evaluate residual margins and graph drift; apply thresholds, top $K_{\max}$ selection, hysteresis, and dwell time to obtain $\mathcal{A}_t$
4: Solve Eq. equation 4 with the fixed PDHG budget to obtain $(\mu_t, f_t)$
5: Score candidate links using Eq. equation 5
6: Return scores, $\mu_t$, and diagnostics $(\Delta E, \mathrm{KKT}, R^+_{r,t}, \|f_t\|_{A_\theta}/\tau)$

The objective balances three effects. The energy favours states compatible with the current graph, the transport term penalises movement from $\mu_{t-1}$ according to the learned transport weights, and the active penalties discourage state-coupled residual violations. The continuity constraint links changes in node belief to the physical flux $q = D_{A_\theta} f$. GLF solves $\mu_t$ and $f_t$ jointly over the whole snapshot, so the resulting state is shared by all link queries at that snapshot rather than recomputed for each candidate edge.

Each PDHG iteration is $O(|E_t|)$ for the transport operations. End-to-end runtime additionally includes the fixed iteration budget, controller computation, Bellman top-$k$ aggregation, and link scoring.

Algorithm 2 summarises the complete snapshot update. The returned diagnostics describe the same transition that produces $\mu_t$, rather than a separate post-processing step.

### 4.4 Score links and report diagnostics

The transport state $\mu_t$ is global, whereas link prediction is source conditioned. For a query $(u, v, t)$ we use

$$\mathrm{score}(u, v) = \alpha \log \mu_t(v) + \beta(z_u^\top z_v + b_v), \tag{5}$$

where $z_u, z_v \in \mathbb{R}^{64}$ are learned embeddings, $b_v$ is a destination bias, $\alpha = 5$ is fixed, and $\beta$ is learned. The belief term supplies a global preference over destinations, while the embedding term provides a lightweight source-specific correction.

The snapshot update records the implementation's primal-dual KKT residual, positive active residual $R^+_{r,t} = [R_{r,t}]_+$, energy change across the update, and transport action $\|f_t\|_{A_\theta}/\tau$. These quantities characterise the numerical and structural behaviour of the solved transition.

## 5 Theory

The theory is organised around the implemented update in Eq. equation 4. We begin with the standard contraction property of a fixed constrained phase, then analyse the finite penalty used by GLF, and finally track a changing target. This organisation separates established Wasserstein consequences from the finite penalty statement specific to the GLF update. Appendix B follows the same order, with a further weighted resolvent consequence for spatial sensitivity in Appendix B.4.

**Assumptions.** We assume: **(A1)** for each fixed snapshot and controller phase, $E_\theta(\cdot; G_t)$ is $\kappa$ geodesically convex for some $\kappa > 0$ in the transport geometry used by the update; **(A2)** the local KKT system of the exact constrained proximal problem is strongly regular on the solution neighbourhood, with uniformly bounded active multipliers over the finite horizon considered; **(A3)** the active state coupled residuals define convex feasible sets and the squared hinge penalties in Eq. equation 4 are evaluated on that neighbourhood; **(A4)** hysteresis and dwell time imply finitely many active set switches on a finite horizon; and **(A5)** within a fixed controller phase, the instantaneous constrained minimiser moves by at most $L\tau$ between successive snapshots, with successive transport metrics uniformly equivalent on the same neighbourhood.

**Proposition 1** (Fixed phase contraction)**.** *Under (A1) to (A3), for a fixed active convex constraint set $\mathcal{C}$, the reflected Wasserstein flow satisfies the standard Evolution Variational Inequality*

$$\frac{d}{dt}\frac{1}{2}W^2_{G_t,A_\theta}(\mu_t, \nu) + \kappa W^2_{G_t,A_\theta}(\mu_t, \nu) \le E_\theta(\nu; G_t) - E_\theta(\mu_t; G_t), \qquad \nu \in \mathcal{C}. \tag{6}$$

*Thus a fixed controller phase contracts in the learned transport geometry, and the JKO step is its implicit discretisation. Appendix B.1 gives the derivation and states where the normal cone reflection enters.*

**Theorem 1** (Finite step control of state coupled residuals)**.** *Assume (A1) to (A4),* $\Psi(s) = (s_+)^2$*, and* $\rho_{r,t} \geq \underline{\rho}$ *for every active state coupled residual. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{r\in\mathcal{A}_t^{\text{state}}} R_{r,t}^{+}(\mu_t) \leq c_1\tau + \frac{c_2}{\underline{\rho}}, \tag{7}$$

*where* $c_1, c_2$ *depend on the local strong regularity constants, bounded active multipliers, and residual regularity on the finite horizon. In the present implementation the bound applies to temporal TV; Bellman consistency is a parameter-level training signal and test-time diagnostic. Appendix B.2 gives the derivation, while Section 6.4 reports KKT residuals for the fixed-budget PDHG implementation.*

**Tracking a changing target.** Let $\mu_t^\star$ be the constrained minimiser for the current snapshot and active set. Under (A5), combining the fixed phase contraction with the $O(\tau)$ JKO discretisation error gives, after the contraction transient,

$$W_{G_t,A_\theta}(\mu_t, \mu_t^\star) \leq C_1\tau + C_2 L/\kappa. \tag{8}$$

Assumption (A5) controls the motion of the target while $\theta$ remains fixed. Appendix B.3 gives the contraction recursion.

**Geometry and spatial sensitivity.** The linearised GLF optimality system is graph structured: local edge flows and residuals couple neighbouring variables in the current transport graph. Under the strong regularity conditions used in graph structured sensitivity analysis, perturbations therefore have exponentially decaying influence with distance in the corresponding KKT sparsity graph (Shin et al., 2022). Appendix B.4 states this consequence for the GLF linearisation; Section 6 evaluates the complementary predictive, residual, and solver behaviour of the update.

## 6 Experiments and Results

The experiments examine GLF from four complementary angles: dynamic link prediction, temporal rank stability, structural-residual ablations, and computational/solver behaviour. We use the JODIE Wikipedia (157K edges), Reddit (672K), and LastFM (1.3M) benchmarks (Kumar et al., 2019) with time-ordered train/validation/test partitions, comparing against TGN (Rossi et al., 2020), TGAT (Xu et al., 2020), and DyGFormer (Yu et al., 2023). Each positive test edge is ranked against 100 negatives sampled from the full node-ID universe; Appendix A.2 gives the exact sampler and tie rule. Unless stated otherwise, results use five random seeds; GLF uses 20 seeds for the main Reddit aggregate and seven on LastFM. Hyperparameters, seed sets, run provenance, and the Wikipedia history-window stress test are reported in Appendices A and C.3. MRR evaluates edge ranking, while KKT, residual, energy, and transport quantities characterise the solved transition.

### 6.1 Main results: dynamic link prediction

On Wikipedia, GLF obtains $0.319 \pm 0.055$ MRR, close to TGN at $0.326 \pm 0.031$ and below DyGFormer at $0.371 \pm 0.061$ and TGAT at $0.572 \pm 0.016$. On Reddit, the 20-seed GLF mean is $0.693 \pm 0.129$; a separate five-seed GLF run gives $0.654 \pm 0.149$, compared with $0.548 \pm 0.077$ for the five-seed TGN reproduction. A persistence heuristic reaches approximately $0.755$, reflecting the strong recurrence of this dataset. On LastFM, GLF obtains $0.116 \pm 0.062$ and TGN $0.125 \pm 0.027$; two GLF seeds yield MRR $0.052$ with zero Mean TV and are retained in the aggregate. Seed-level results and run provenance are reported in Appendices C.4, C.5, and A.3.

### 6.2 Temporal rank stability

Temporal TV encourages smooth state evolution. To test whether that behaviour reaches the scores, we fix a pool of 50 candidate destinations for each query node using seed 42 and compute Spearman $\rho$ between consecutive score vectors over the same identities. Wikipedia uses the same three seeds

Table 1: Dynamic link prediction test MRR with 100 negatives sampled from the full node ID universe (Appendix A.2). Values are mean ± std over random seeds.

| Method | Wikipedia | Reddit | LastFM |
|---|---|---|---|
| TGN (Rossi et al., 2020) | $0.326 \pm 0.031$ | $0.548 \pm 0.077$ | $\mathbf{0.125 \pm 0.027}$ |
| TGAT (Xu et al., 2020) | $\mathbf{0.572 \pm 0.016}$ | –[§] | –[†] |
| DyGFormer (Yu et al., 2023) | $0.371 \pm 0.061$ | –[§] | –[†] |
| GLF (ours) | $0.319 \pm 0.055$ | $\mathbf{0.693 \pm 0.129}$ | $0.116 \pm 0.062$ |

Unless otherwise stated, results use five random seeds. GLF Reddit uses 20 seeds and GLF LastFM seven. [†] TGAT and DyGFormer did not complete LastFM training within the single GPU compute budget. [§] Reddit TGAT and DyGFormer were evaluated for the fixed probe rank stability diagnostic, but not for the 100 negative MRR experiment.

Table 2: Fixed probe temporal rank stability measured by Spearman $\rho$ between consecutive snapshots.

| Method | Wikipedia $\rho$ | Reddit $\rho$ |
|---|---|---|
| TGAT (Xu et al., 2020) | $0.555 \pm 0.018$ | $0.778 \pm 0.015$ |
| DyGFormer (Yu et al., 2023) | $0.858 \pm 0.095$ | $0.915 \pm 0.017$ |
| TGN (Rossi et al., 2020) | $0.995 \pm 0.0007$ | $0.952 \pm 0.006$ |
| **GLF (ours)** | $\mathbf{0.999 \pm 0.0005}$ | $\mathbf{0.993 \pm 0.009}$ |

for all four methods. Reddit uses five GLF seeds and three baseline seeds; Appendix C.6 also gives GLF on the matched three seed subset. GLF has the highest reported rank stability in Table 2; on the matched three-seed Reddit subset it gives $0.996 \pm 0.003$. Because temporal TV explicitly promotes smooth state evolution, we interpret this result as a mechanism check: the smoothness imposed on the state is also visible in rankings over fixed candidate identities. Predictive accuracy is evaluated separately by MRR in Table 1.

### 6.3 Ablation: structural residuals

Table 3: Structural-residual ablation on Wikipedia under the shorter three-epoch protocol.

| Configuration | Test MRR | $\Delta$ vs "barriers off" |
|---|---|---|
| GLF (barriers off) | $0.294 \pm 0.105$ | – |
| GLF (Bellman only) | $0.379 \pm 0.034$ | $+29\%$ |
| GLF (TV only) | $0.364 \pm 0.065$ | $+24\%$ |
| GLF (Bellman + TV) | $0.334 \pm 0.066$ | $+14\%$ |

Removing both residuals gives the lowest mean MRR, while Bellman-only and TV-only give the largest gains. The joint configuration remains above barriers off, showing a material but non-additive effect of the two structural terms under this schedule.

### 6.4 Computation and snapshot diagnostics

Table 4 reports complete training time rather than solver time in isolation. GLF requires $1{,}033$ s per seed, compared with $472$ s for TGN, $11{,}906$ s for TGAT, and $18{,}388$ s for DyGFormer. This measurement includes the fixed PDHG budget, controller computation, Bellman top-$k$ aggregation, and link scoring. The matched inference benchmark evaluates the complete processing path on the same four Wikipedia test bins. GLF processes a bin in a median of 1.87 s, compared with 6.24 s for TGN and 391.5–412.9 s for DyGFormer and TGAT. These measurements characterise the matched single-file implementations used in this study; the timing protocol and amortised per-edge results are reported in Appendix C.2.

Table 4: Training time per seed on Wikipedia using a single A100 GPU over five seeds.

| Method | Training time (s) | Speedup vs DyGFormer |
|---|---|---|
| TGN (Rossi et al., 2020) | 472 | 39× |
| GLF (ours) | 1,033 | 18× |
| TGAT (Xu et al., 2020) | 11,906 | 1.5× |
| DyGFormer (Yu et al., 2023) | 18,388 | 1× |

Table 5: Wikipedia inference latency on four common test bins using matched single file reproductions on one A100. Snapshot time covers each complete bin; per edge time is amortised over its edges.

| Method | Snap. med. (s) | p95 (s) | Edge med. (ms) | p95 (ms) | Edges/s |
|---|---|---|---|---|---|
| GLF | 1.87 | 1.87 | 0.47 | 0.47 | 2109 |
| TGN | 6.24 | 6.27 | 1.58 | 1.64 | 631 |
| DyGFormer | 391.5 | 396.2 | 101.1 | 103.4 | 10.1 |
| TGAT | 412.9 | 420.8 | 106.6 | 109.6 | 9.5 |

The solver diagnostics provide a complementary view of the update. Across 70,863 Wikipedia evaluation predictions, the KKT residual has median $7.8 \times 10^{-3}$ and a 10th to 90th percentile range of $[2.9 \times 10^{-3}, 4.4 \times 10^{-2}]$. The fixed-window history experiment varies the amount of earlier training history while holding the validation and test periods fixed. The runs separate into active-transport, zero-transport, and low-MRR regimes. Within the 12 active-transport runs, mean MRR spans $0.374$–$0.418$ across the four history settings, while KKT residuals and temporal-TV behaviour remain similar across cutoffs. Full seed-level results and the diagnostic signatures of these regimes are reported in Appendix C.3.

## 7 Conclusion

Graph Logic Flows treats the transition between graph snapshots as a single object to be solved. Learned conductances define how belief can move over the current graph, structural residuals determine the conditions acting on that movement, and the constrained transport solve produces both the next state and quantities that describe how the transition was obtained. This turns the snapshot update from an implicit sequence of local operations into an explicit optimisation object.

The experiments show that this formulation remains useful for dynamic link prediction while exposing behaviour that is otherwise hidden inside the update. Temporal-TV effects appear in the ranking dynamics, Bellman and TV residuals materially change prediction, and solver quantities distinguish qualitatively different update regimes. The computational results further show that solving the transition globally can remain practical in the tested setting. More broadly, GLF suggests a different design principle for temporal graph learning. Instead of asking only how information should be propagated across time, one can ask what geometry should govern the transition, what structural conditions should act on it, and what evidence the update should expose once it has been solved. In this view, prediction and diagnostics are read from the same solved transition. This makes global constrained transport a natural candidate for temporal models in which the behaviour of the update matters alongside the prediction it produces.

### AI use statement

Generative AI tools were used primarily for manuscript writing and language polishing, with additional checks of references and consistency between the text and reported results. The authors reviewed all AI-assisted revisions and take responsibility for the final manuscript.

### Reproducibility statement

The study uses existing temporal graph benchmarks and involves no human participants or newly collected personal data. The anonymised supplementary material provides supporting code and experimental records; further details are given in the appendices.

## REFERENCES


Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2005.

Hugo Attali, Davide Buscaldi, and Nathalie Pernelle. Rewiring techniques to mitigate oversquashing and oversmoothing in gnns: A survey. *arXiv preprint arXiv:2411.17429*, 2024.

Pradeep Kr Banerjee, Kedar Karhadkar, Yu Guang Wang, Uri Alon, and Guido Montúfar. Oversquashing in gnns through the lens of information contraction and graph expansion. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8. IEEE, 2022.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pp. 1407–1418. PMLR, 2021.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *International Conference on Learning Representations*, 2023.

Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long range propagation on continuous-time dynamic graphs. *arXiv preprint arXiv:2406.02740*, 2024.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 2056–2073, 2023. doi: 10.52202/075280-0099.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Kedar Karhadkar, Pradeep Kr Banerjee, and Guido Montúfar. Fosr: First-order spectral rewiring for addressing oversquashing in gnns. *arXiv preprint arXiv:2210.11790*, 2022.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Bertrand Maury, Aude Roudneff-Chupin, Filippo Santambrogio, and Juliette Venel. Handling congestion in crowd motion modeling. *arXiv preprint arXiv:1101.4102*, 2011.

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, pp. 32928–32941, 2022. doi: 10.52202/068431-2386.

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.

Sungho Shin, Mihai Anitescu, and Victor M. Zavala. Exponential decay of sensitivity in graph-structured nonlinear programs. *SIAM Journal on Optimization*, 32(2):1156–1183, 2022. doi: 10.1137/21M1391079.

Joshua Southern, Francesco Di Giovanni, Michael Bronstein, and Johannes F Lutzeyer. Understanding virtual nodes: Oversquashing and node heterogeneity. *arXiv preprint arXiv:2405.13526*, 2024.

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

Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *arXiv preprint arXiv:2105.07944*, 2021.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? *arXiv preprint arXiv:1905.13211*, 2019.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

# A Experimental details

## A.1 Hyperparameters

Table 6 reports the GLF hyperparameters used in the matched protocol experiments. Unless otherwise stated, the same configuration is used across datasets. Evaluation barrier weights are half the corresponding training values. In the recorded hyperparameter sweep this setting produced more stable $\mu_t$ trajectories without changing the rank order.

Table 6: GLF hyperparameters used in the matched protocol experiments.

| Component | Setting |
|---|---|
| JKO step size | $\tau = 0.12$ |
| Augmentation barrier weight | $\rho_{\text{aug}} = 3.0$ |
| Inner solver | $\text{iters}_{\mu} = 20$, $\text{iters}_{\text{admm}} = 4$, $\text{lr}_{\mu} = 0.12$ |
| Training barrier weights | Bellman $0.5$, temporal TV $0.02$ |
| Evaluation barrier weights | Bellman $0.25$, temporal TV $0.01$ |
| Optimiser | Adam |
| Edge conductance network | lr $8 \times 10^{-4}$, weight decay $10^{-5}$ |
| Scorer | lr $10^{-3}$, weight decay $10^{-5}$ |
| Scheduler | StepLR with decay factor $\gamma = 0.5$ |
| Training schedule | 6 epochs with patience 2 early stopping on validation MRR |
| Energy regularisation | $\lambda_{\text{energy}} = 5 \times 10^{-3}$, entropy weight $0.4$ |

## A.2 Negative sampling and ranking protocol

For each test edge $(u, v)$, GLF samples negatives uniformly from the full node ID range. The implementation draws $k + 16$ candidates, removes the true destination $v$, applies `torch.unique`, and keeps the first $k = 100$. The baseline reproductions draw 100 IDs directly from the same full node universe using `randint(0,n)` and use the same scoring and ranking rule. GLF additionally removes the target and de duplicates its candidate set; on Wikipedia, a 100 draw baseline set contains a resampled positive in roughly one percent of queries.

Wikipedia and LastFM are bipartite interaction graphs. Wikipedia has roughly 8.2k user nodes and 1k page nodes, so uniform full node sampling produces many source side user IDs that cannot be true destinations. The resulting MRR values describe this full node candidate universe; destination only evaluations in the original temporal graph literature use a different candidate set (Rossi et al., 2020; Xu et al., 2020). For every method the rank is

$$\text{rank} = 1 + \sum_{v' \in \mathcal{N}} \mathbf{1}\{s(u, v') > s(u, v)\},$$

so score ties favour the positive. Table 1 compares methods within this common candidate universe.

### A.3 SEEDS AND RUN PROVENANCE

The primary seed set is $\{1337, 2027, 3109, 4441, 5557\}$. LastFM adds seeds 6666 and 7777, while the Reddit main result uses 20 seeds. Table 7 lists the sets.

Table 7: Random seed sets used for GLF experiments.

| Experiment | Seeds |
|---|---|
| Primary GLF runs | 1337, 2027, 3109, 4441, 5557 |
| LastFM seven seed aggregate | Primary seeds plus 6666, 7777 |
| Reddit 20 seed MRR aggregate | Primary seeds plus 22222, 33333, 44444, 55555, 66666; 12345, 23456, 34567, 45678, 56789; 67890, 78901, 89012, 90123, 10101 |

Three Wikipedia aggregates appear in the paper and come from separate executions of the same protocol. The canonical Table 1 run is $0.319 \pm 0.055$ with seed values $\{0.340, 0.315, 0.400, 0.229, 0.314\}$. The matched TV diagnostic run is $0.378 \pm 0.033$ and is used in the Mean TV analysis where MRR and Mean TV were recorded from the same trained models. The history window experiment has its own six seed 100% history control, $0.291 \pm 0.109$. Repeated GPU executions can enter different optimisation regimes; each table therefore keeps results from one registered execution rather than mixing values across runs. The four rows in Table 3 belong to the separate shorter three-epoch ablation, whose joint Bellman+TV result is $0.334 \pm 0.066$; they are not mixed with the canonical $0.319 \pm 0.055$ Wikipedia run in Table 1.

## B THEORY DETAILS

The appendix follows Section 5 in the same order: fixed phase contraction, finite step residual control, tracking a changing target, and the additional geometry controlled sensitivity result.

### B.1 FIXED PHASE CONTRACTION

**Proposition 2** (EVI contraction)**.** *Assume (A1) to (A3). Fix an active state coupled set $\mathcal{A}^{\mathrm{state}}$ and let*

$$\mathcal{C} = \Delta^n \cap \bigcap_{r \in \mathcal{A}^{\mathrm{state}}} \{B_r \le 0\}$$

*be the corresponding feasible set in the JKO state variables. For the reflected Wasserstein flow constrained to $\mathcal{C}$, the following Evolution Variational Inequality holds for every $\nu \in \mathcal{C}$:*

$$\frac{d}{dt}\frac{1}{2}W^2_{G_t,A_\theta}(\mu_t,\nu) + \kappa W^2_{G_t,A_\theta}(\mu_t,\nu) \le E_\theta(\nu;G_t) - E_\theta(\mu_t;G_t). \tag{9}$$

*Thus, during a fixed controller phase, the flow contracts in the learned Wasserstein geometry and dissipates the energy $E_\theta$.*

*Derivation under (A1) to (A3).* Within a fixed phase, write $W = W_{G_t,A_\theta}$. Assumption (A1) gives the standard slope inequality for a $\kappa$ geodesically convex energy:

$$\langle \nabla_W E_\theta(\mu_t), \nu - \mu_t \rangle_W \ge E_\theta(\nu) - E_\theta(\mu_t) + \frac{\kappa}{2} W^2(\mu_t,\nu), \qquad \nu \in \mathcal{C}. \tag{10}$$

The reflected flow decomposes into the Wasserstein descent direction and a reflection term:

$$\dot{\mu}_t = -\nabla_W E_\theta(\mu_t) + u_t. \tag{11}$$

Here $u_t$ is the reflection field induced by the convex feasible set. Since this reflection acts through the normal cone of $\mathcal{C}$, it cannot move the trajectory away from any feasible comparison point:

$$\langle u_t, \nu - \mu_t \rangle_W \le 0, \qquad \nu \in \mathcal{C}. \tag{12}$$

Differentiating the half squared Wasserstein distance along equation 11 gives

$$\frac{d}{dt}\frac{1}{2}W^2(\mu_t,\nu) = \langle \dot{\mu}_t, \mu_t - \nu \rangle_W = -\langle \nabla_W E_\theta(\mu_t) - u_t, \mu_t - \nu \rangle_W. \tag{13}$$

Combining equation 10, equation 12, and equation 13 yields equation 9, following the standard EVI argument for $\kappa$ convex Wasserstein gradient flows with reflection on convex sets (Ambrosio et al., 2005; Maury et al., 2011). □

**Discrete update.** The JKO update in equation 4 is the implicit Euler discretisation of this reflected flow. Under the same regularity assumptions, standard JKO theory (Jordan et al., 1998; Ambrosio et al., 2005) gives stability and energy dissipation up to $O(\tau)$ discretisation error.

### B.2 Finite step residual control

The implemented JKO update uses finite squared hinge penalties in place of exact projection onto the active constraint set. The next result controls these residuals through the JKO step size and active penalty strength, restricted to residuals coupled to the JKO optimisation variables.

**Theorem 2** (Finite step residual control)**.** *Assume (A1) to (A4). Let $\Psi(s) = (s_+)^2$, and let $\mathcal{A}_t^{\mathrm{state}} \subseteq \mathcal{A}_t$ denote the active residuals that depend on the JKO optimisation variables $(\mu, f)$. In this paper, $\mathcal{A}_t^{\mathrm{state}} = \mathcal{A}_t \cap \{B_{\mathrm{TV}}\}$. Let $R_{r,t}^+(\mu_t)$ denote the positive part of such an active residual. Suppose that all active state coupled penalty weights satisfy $\rho_{r,t} \geq \underline{\rho}$ for $r \in \mathcal{A}_t^{\mathrm{state}}$. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{r\in\mathcal{A}_t^{\mathrm{state}}} R_{r,t}^+(\mu_t) \leq c_1\tau + \frac{c_2}{\underline{\rho}}, \tag{14}$$

*where $c_1, c_2$ depend on local conditioning and residual regularity. Hence, as $\tau \to 0$ and $\underline{\rho} \to \infty$, the active positive state coupled residuals vanish. The Bellman residual is independent of the JKO state variables. Training controls it through gradients with respect to $\theta$, and the test time solver records it as a structural diagnostic.*

*Derivation under (A1) to (A4).* We restrict to $r \in \mathcal{A}_t^{\mathrm{state}}$, for which $\nabla_{\mu_t} R_{r,t}(\mu_t) \not\equiv 0$ and the penalty term contributes a nontrivial first order condition. Fix the active set during one controller phase. The JKO objective in equation 4 consists of three parts: the energy, the Wasserstein proximal term, and squared hinge penalties on the active residuals. By (A1) to (A2), the exact constrained proximal problem has a strongly regular local KKT system with bounded active multipliers. The finite squared hinge penalty is therefore a local perturbation of that system, while the JKO discretisation contributes the $O(\tau)$ term.

The contribution of the active state coupled penalties to the first order condition is

$$\sum_{r\in\mathcal{A}_t^{\mathrm{state}}} \rho_{r,t}\Psi'(R_{r,t}(\mu_t))\,\nabla R_{r,t}(\mu_t), \qquad \Psi'(s) = 2s_+. \tag{15}$$

The first order system for the penalised problem differs from the exact constrained KKT system through the finite penalty response. Under the strong regularity and bounded multiplier condition in (A2), the local solution map is Lipschitz in this perturbation, which yields

$$R_{r,t}^+(\mu_t) \leq C_1\tau + \frac{C_2}{\underline{\rho}}. \tag{16}$$

The first term comes from the finite JKO step size, while the second comes from using a finite penalty rather than an exact projection. Summing equation 16 over the state coupled active residuals and over time gives equation 14. The dwell time condition in (A4) prevents arbitrarily fast switching of $\mathcal{A}_t$, so the constants remain bounded over a finite horizon. Taking $\tau \to 0$ and $\underline{\rho} \to \infty$ gives the zero residual reflected flow limit for the state coupled residuals. □

### B.3 Tracking a changing target

Dynamic graphs change between snapshots, so the instantaneous constrained minimiser can move even when the learned parameters are fixed. Assumption (A5) states this movement directly at the time scale of one snapshot: the target moves by at most $L\tau$, and successive metrics remain uniformly equivalent on the local neighbourhood. The next result combines that target drift with the fixed phase contraction.

**Theorem 3** (Drift tracking)**.** *Assume (A1) to (A5). Let $\mu_t^\star$ be a minimiser of $E_\theta(\cdot; G_t)$ over the active feasible set at snapshot $t$. For sufficiently small $\tau$, and after the exponentially decaying contraction transient,*

$$W_{G_t,A_\theta}(\mu_t, \mu_t^\star) \leq C_1\tau + C_2\frac{L}{\kappa}. \tag{17}$$

*Derivation sketch.* Within a fixed controller phase, Proposition 2 gives contraction towards the current constrained minimiser. The JKO step inherits this contraction up to an $O(\tau)$ discretisation error. Let $e_t = W_{G_t,A_\theta}(\mu_t, \mu_t^\star)$. Uniform equivalence of successive metrics and the target drift bound in (A5) give a recursion of the form $e_{t+1} \le (1+\kappa\tau)^{-1} e_t + C\tau^2 + CL\tau$. Iterating the recursion and discarding the exponentially decaying transient yields $e_t \le C_1\tau + C_2 L/\kappa$, which is equation 17. The dwell time condition in (A4) allows the constants to be taken uniformly within each finite controller phase. □

### B.4 Geometry controlled sensitivity

The linearised KKT system of the GLF update inherits a sparse graph structure from the current edge set and the local residual couplings. Under the strong second order sufficient condition, linear independence constraint qualification, and uniform regularity assumptions for this graph structured system, exponential decay of sensitivity follows from standard graph structured nonlinear programming theory (Shin et al., 2022).

**Proposition 3** (Graph structured sensitivity)**.** *Let $d_{\mathcal{K}}(i,j)$ denote distance between node associated blocks $i$ and $j$ in the sparsity graph of the linearised KKT system. Under the regularity conditions above, there exist constants $C > 0$ and $q \in (0,1)$, uniform on the local solution neighbourhood, such that a perturbation $\xi_j$ entering block $j$ satisfies*

$$\left\| \frac{\partial \mu_t(i)}{\partial \xi_j} \right\| \le C q^{d_{\mathcal{K}}(i,j)}. \tag{18}$$

*Specialisation to the GLF linearisation.* Fix a snapshot and active set. Linearising the first order optimality conditions of Eq. equation 4 produces a graph structured KKT system whose blocks contain node states, edge flows, and local constraint variables. Edge flow terms couple incident nodes, while active local residuals add only their stated supports. The exponential decay of primal dual solution sensitivity for such graph structured nonlinear programs follows under strong second order sufficiency, linear independence of active constraints, and uniform conditioning (Shin et al., 2022). Projecting the resulting primal dual sensitivity bound onto the node state blocks gives Eq. equation 18. The constants depend on the local KKT conditioning and coupling strengths induced by the current conductances and active residuals. □

## C Additional experimental diagnostics

This section gives the solver and seed level records behind the aggregate results in the main paper.

### C.1 Solver diagnostics

Across 70,863 Wikipedia evaluation predictions from seeds 1337, 2027, and 3109, KKT residuals range from $2.8 \times 10^{-3}$ to $8.8 \times 10^{-2}$, with median $7.8 \times 10^{-3}$ and a 10th to 90th percentile interval of $[2.9 \times 10^{-3}, 4.4 \times 10^{-2}]$. The fractions of predictions with zero temporal TV are 33.0%, 54.2%, and 32.2% for the three seeds. Each run therefore has 46–68% of predictions with positive TV and belongs to the active transport regime. Snapshot level zero TV events in this diagnostic set are distinct from the zero transport regime in Appendix C.3, where TV is zero throughout the run.

### C.2 Inference latency protocol

The latency experiment uses the first four Wikipedia test bins, containing 3936, 3937, 3938, and 3937 edges. All four methods process exactly these edges on the same A100. Each model is trained for one epoch with seed 1337, then evaluated with `eval()` and `no_grad()`. CUDA is synchronised immediately before and after each timed segment. Data loading and training are outside the timer. The first GLF bin and the first 20 baseline edges serve as warm up and are excluded.

For GLF, a timed bin includes geometry construction, the default `GLFConfig()` PDHG update, and scoring of 101 candidates for every edge in the bin. For TGN, TGAT, and DyGFormer, snapshot time is the sum of edge level prediction and each model's memory or history update over the same bin. The comparison uses the same single file reproduction code family as the other baseline experiments.

TGAT and DyGFormer perform neighbour aggregation through per node Python loops in these reproductions, while GLF scoring is likewise edge by edge. The four bins vary by less than 4% within each method.

### C.3 History window stress test

Wikipedia contains 28 training bins. Validation bins 28–33 and test bins 34–39 are identical in all 24 runs. Cutoffs 0.4, 0.6, 0.8, and 1.0 train on the earliest 11, 17, 22, or 28 training bins, respectively. Each cutoff uses seeds $\{1337, 2027, 3109, 4441, 5557, 6666\}$ with the six epoch, patience 2 protocol and fixed barrier weights.

Table 8: Wikipedia history window test MRR. Superscripts indicate collapse (A), zero transport (B), and active transport (C).

| Seed | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|
| 1337 | $0.068^{A}$ | $0.074^{A}$ | $0.367^{C}$ | $0.057^{A}$ |
| 2027 | $0.414^{C}$ | $0.414^{C}$ | $0.330^{C}$ | $0.315^{B}$ |
| 3109 | $0.422^{C}$ | $0.384^{C}$ | $0.412^{C}$ | $0.397^{C}$ |
| 4441 | $0.075^{A}$ | $0.312^{B}$ | $0.049^{A}$ | $0.314^{B}$ |
| 5557 | $0.419^{C}$ | $0.425^{C}$ | $0.408^{C}$ | $0.351^{C}$ |
| 6666 | $0.310^{B}$ | $0.314^{B}$ | $0.310^{B}$ | $0.313^{B}$ |
| Mean $\pm$ std | $0.285 \pm 0.155$ | $0.320 \pm 0.119$ | $0.313 \pm 0.124$ | $0.291 \pm 0.109$ |

Regime A is validation and test collapse, with test MRR below 0.10; it occurs in 5/24 runs and has best validation MRR 0.037–0.077. Regime B is a zero transport solution: KKT residual is approximately $10^{-9}$, transport action and energy change are zero, temporal TV is approximately $10^{-14}$, top 1 TV is zero, and the logged Bellman term is NaN. It occurs in 7/24 runs with test MRR 0.310–0.315, close to the Wikipedia persistence value 0.305. The Bellman diagnostic is NaN in these zero transport runs, while the remaining solver quantities make the regime directly identifiable. Regime C has active transport, KKT median 0.005–0.020, and positive TV; it occurs in 12/24 runs with test MRR 0.330–0.425.

Table 9: History window results for active transport runs (Regime C).

| Cutoff | $n$ | Test MRR | KKT median | Top 1 TV |
|---|---|---|---|---|
| 0.4 | 3 | $0.418 \pm 0.003$ | 0.0131 | 0.00341 |
| 0.6 | 3 | $0.408 \pm 0.017$ | 0.0148 | 0.00341 |
| 0.8 | 4 | $0.379 \pm 0.033$ | 0.0111 | 0.00299 |
| 1.0 | 2 | $0.374 \pm 0.023$ | 0.0142 | 0.00332 |

Across active transport runs, the fixed late test interval shows similar performance across the four history cutoffs. The temporal barrier follows the same snapshot profile at all cutoffs: approximately 1.0–1.45 at the first test snapshot and 0.01–0.04 later in the horizon. Regime frequencies also have no monotone relation with cutoff: A occurs $2/1/1/1$ times, B $1/2/1/3$, and C $3/3/4/2$. Validation MRR separates Regime A from B/C, while transport action and KKT expose Regime B directly.

### C.4 Reddit seed level results

Table 10 gives the 20 GLF Reddit runs underlying Table 1. Their mean is $0.693 \pm 0.129$ and the median is $0.742$. Seventeen runs reach at least $0.70$, one is $0.678$, and two are below $0.45$.

All 20 registered runs are retained in the aggregate, including the two low MRR runs.

### C.5 Mean TV and seed level results

Mean TV is the per test edge change in belief mass on the top 1 candidate,

$$|\mu_t(v_{\text{top1}}) - \mu_{t-1}(v_{\text{top1}})|,$$

Table 10: Sorted GLF Reddit test MRR values over 20 seeds.

| Rank | MRR | Rank | MRR | Rank | MRR | Rank | MRR |
|---|---|---|---|---|---|---|---|
| 1 | 0.2545 | 6 | 0.7142 | 11 | 0.7480 | 16 | 0.7563 |
| 2 | 0.4129 | 7 | 0.7151 | 12 | 0.7550 | 17 | 0.7565 |
| 3 | 0.6569 | 8 | 0.7171 | 13 | 0.7552 | 18 | 0.7566 |
| 4 | 0.6777 | 9 | 0.7183 | 14 | 0.7553 | 19 | 0.7567 |
| 5 | 0.7010 | 10 | 0.7353 | 15 | 0.7561 | 20 | 0.7574 |
| Mean $\pm$ std | | $0.693 \pm 0.129$ | | Median | | 0.742 | |
| MRR $\geq 0.70$ | | 17 / 20 | | MRR 0.50 to 0.70 | | 1 / 20 | |
| MRR $< 0.45$ | | | | 2 / 20 | | | |

averaged across the test horizon.

Table 11: GLF Mean TV from runs in which MRR and Mean TV were recorded from the same trained models.

| Dataset | Seeds | Mean TV | MRR for same run |
|---|---|---|---|
| Wikipedia matched TV diagnostic | 5 | $0.00289 \pm 0.00161$ | $0.378 \pm 0.033$ |
| Reddit | 5 | $0.00362 \pm 0.00212$ | $0.654 \pm 0.149$ |
| LastFM | 7 | $0.00525 \pm 0.00430$ | $0.116 \pm 0.062$ |

The Wikipedia matched TV run is separate from the canonical $0.319 \pm 0.055$ aggregate in Table 1; it appears here because MRR and Mean TV were recorded from the same models. The Reddit row similarly uses the five seed diagnostic run, while Table 1 uses the 20 seed MRR aggregate.

Table 12: Seed level GLF test MRR values for the main dynamic link prediction experiments.

| Dataset | GLF seed level test MRR |
|---|---|
| Wikipedia | $\{0.340, 0.315, 0.400, 0.229, 0.314\}$ (5 seeds) |
| Reddit | full sorted list in Table 10 |
| LastFM | $\{0.052, 0.052, 0.075, 0.102, 0.127, 0.172, 0.233\}$ (7 seeds) |

For LastFM, seeds 6666 and 7777 produce nearly constant predictions over the test horizon, with Mean TV equal to zero. Table 13 reports the corresponding values.

### C.6 Rank stability details

Table 2 uses the same three Wikipedia seeds for all four methods. The values are $0.999 \pm 0.0005$ for GLF, $0.995 \pm 0.0007$ for TGN, $0.858 \pm 0.095$ for DyGFormer, and $0.555 \pm 0.018$ for TGAT.

On Reddit, the five seed GLF value is $0.993 \pm 0.009$. The baseline diagnostic seeds are $\{1337, 2027, 3109\}$; GLF on exactly those seeds gives $0.996 \pm 0.003$. Reddit TGAT and DyGFormer entries in Table 2 refer to this fixed probe diagnostic rather than the 100 negative MRR experiment. LastFM GLF rank stability is $0.9996 \pm 0.0003$ over three seeds, with per seed values $\{0.9993, 0.9999, 0.9997\}$.

Table 13: LastFM per seed MRR and Mean TV diagnostic (7 seeds).

| Seed | 1337 | 2027 | 3109 | 4441 | 5557 | 6666 | 7777 |
|---|---|---|---|---|---|---|---|
| MRR | 0.102 | 0.075 | 0.172 | 0.127 | 0.233 | 0.052 | 0.052 |
| Mean TV | 0.00524 | 0.00325 | 0.01137 | 0.00597 | 0.01092 | 0.00000 | 0.00000 |