PIANO: PATH-INTEGRAL ADAPTIVE NEURAL ODES ON GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many natural and engineered spatiotemporal processes evolve through continuous interactions over a network, while the spatial range of relevant interactions can vary with the system state. Existing graph neural ODEs typically rely on a predetermined hop structure within each vector-field evaluation, preventing the propagation mass assigned to different spatial orders from adapting with the evolving state. We introduce PIANO (*Path-Integral Adaptive Neural ODE*), whose instantaneous spatial operator is a *state-adaptive path-integral propagator*: an expected finite-step graph-walk propagator under a state-conditioned distribution over walk lengths. A lightweight topology-aware gate further enables node-wise modulation when structural roles are informative. The adaptive operator is implemented with sparse graph operations whose cost scales linearly with maximum propagation order, edge count, and feature dimension. We show that each realised spatial operator is non-expansive, establish well-posedness of the nonlinear flow, and, for a frozen linear surrogate, characterise how the maximum propagation order and integration horizon jointly determine a sufficient finite-hop approximation radius. Controlled experiments validate the proposed adaptive propagation and topology-aware mechanisms. On ocean and severe-weather forecasting, PIANO achieves strong performance with a compact model and further extends effectively to large-scale sparse graph learning.

1 INTRODUCTION

Many natural and engineered spatiotemporal processes—including ocean circulation, convective storms, transportation networks, and coupled oscillators—evolve continuously through interactions over a spatial or relational network (Vallis, 2017; Veillette et al., 2020; Yu et al., 2018; Acebrón et al., 2005). Crucially, the relevant interaction range need not remain fixed: a node may be governed primarily by nearby interactions in one regime but depend on broader spatial context in another. A useful continuous-time graph model should therefore represent not only *where* interactions occur and *how* states evolve, but also *how far* information should propagate as the state changes.

Existing architectures adapt spatial range in different ways, but such adaptation is typically tied to discrete propagation depth or local graph structure rather than continuously adjusted along the latent trajectory. Discrete adaptive-hop methods such as DAGNN (Liu et al., 2020) and GAMLP (Zhang et al., 2022) reweight representations obtained at different propagation depths, while AMP (Errica et al., 2025) learns propagation depth and message filters. These mechanisms operate in discrete architectures rather than continuously adapting the order-wise propagation mixture along an ODE trajectory. Graph neural ODEs (Chen et al., 2018; Xhonneux et al., 2020; Fang et al., 2021; Chamberlain et al., 2021) replace discrete depth with integration, but most surveyed methods rely on a predetermined hop structure within each vector-field evaluation. Learning the graph or adapting one-hop edge weights changes *which neighbours* interact, but **not how propagation mass is distributed across multiple spatial orders** as the latent state evolves. Higher temporal order does not resolve this limitation; HOPE (Luo et al., 2023), for example, uses second-order temporal dynamics while retaining local instantaneous interactions. Temporal and spatial order are distinct: temporal order determines how dynamics evolve through time, whereas spatial order determines how far information is propagated within each vector-field evaluation. This leaves a central question for continuous-time graph learning:

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.

*How can a continuous-time graph model **dynamically redistribute propagation mass across spatial orders** as the latent state evolves, while retaining continuous temporal evolution and affordable explicit multi-order propagation?*

We answer this question with PIANO (*Path-Integral Adaptive Neural ODE*). PIANO treats the latent state as an evolving graph signal and predicts a walk-length distribution $\boldsymbol{\alpha}(\mathbf{H}, t)$. Each k-step component propagates information along length-k graph walks, while their adaptive mixture shifts between local, broader-range, and multiscale interaction as the state evolves. A selective topology-aware gate $\boldsymbol{\gamma}$ modulates destination-node updates when structural roles are informative. Thus $\boldsymbol{\alpha}$ controls *how far*, $\boldsymbol{\gamma}$ controls *where*, and the integration horizon T controls *how long* propagation is composed (Figure 1). Nested sparse multiplications cost $\mathcal{O}(L|E|d)$, where L is the maximum walk order, $|E|$ the edge count, and d the feature width; each k-step contribution is computed by repeated sparse one-step propagation, without constructing graph-power matrices. Our analysis proves each realised spatial operator non-expansive (Lemma 1), establishes a well-posed nonlinear flow (Theorem 1), and derives a sufficient finite-hop radius from L and T for a frozen linear surrogate (Theorem 2).

Multi-order propagation admits a natural graph-walk interpretation: the weight assigned to each finite walk length can be viewed as propagation mass allocated to the corresponding spatial order. Classical heat diffusion (Chung, 2007; van Dijk et al., 2018) exhibits this finite-walk structure, with continuous diffusion represented as an expectation over finite-step random-walk propagators under a Poisson walk-length law. However, once the diffusion time τ is specified, this law fixes how propagation mass is distributed across walk lengths and cannot adapt to changes in the evolving graph state. PIANO retains the finite-walk construction but replaces the fixed Poisson law with a state-conditioned categorical distribution:

$$\begin{aligned} \text{Classical diffusion:} \quad & e^{\tau(P-I)} = \mathbb{E}[P^{K_\tau}], && K_\tau \sim \text{Poisson}(\tau), \\ \text{PIANO:} \quad & M_\theta(\mathbf{H}, t) = \mathbb{E}\Big[\hat{A}^{K_{\mathbf{H},t}} \mid \mathbf{H}, t\Big], && K_{\mathbf{H},t} \sim \text{Cat}\big(\boldsymbol{\alpha}(\mathbf{H}, t)\big). \end{aligned} \tag{1}$$

Thus $\boldsymbol{\alpha}(\mathbf{H}, t)$ dynamically redistributes propagation mass across spatial orders $0, \ldots, L$ as the latent state evolves. The expectation $\mathbb{E}[K_{\mathbf{H},t} \mid \mathbf{H}, t]$ provides an interpretable effective walk length, while the full distribution captures multiscale mixing across spatial orders.

Controlled interventions validate adaptive propagation and clarify gate utility (Section 5.1). Transfer studies show compact, competitive performance not only on natural spatiotemporal scenarios—ocean forecasting (Ocean ΔSLA) and severe-weather forecasting (SEVIR VIL)—but also on static graphs for transductive node classification (Cora, Citeseer, PubMed) and large-scale sparse full-graph learning (ogbn-arxiv) (Section 5.2).

Contributions.

- **State-adaptive spatial-order weighting.** We formulate graph propagation as a path-integral propagator with a state-conditioned walk-length distribution, dynamically redistributing propagation mass across spatial orders. A lightweight topology-aware gate provides selective node-wise modulation when useful.
- **Stable and efficient continuous flow.** We evaluate the adaptive propagator using L sparse matrix multiplications, giving $\mathcal{O}(L|E|d)$ propagation cost without dense graph powers. We establish pointwise non-expansiveness of each realised spatial operator and well-posedness of the nonlinear flow, and show for a frozen linear surrogate how L and T jointly control a sufficient finite-hop approximation radius.
- **Mechanism validation and cross-domain transfer.** Controlled Graph-wave-Scale interventions show that the learned walk-length distribution adapts to spatial interaction range and is functionally used; Springs identifies when topology-aware modulation contributes. Experiments demonstrate transfer across ocean and severe-weather forecasting, transductive node classification, and large-scale sparse full-graph learning.

Table 1 summarises the instantaneous spatial operators of representative continuous-time graph models, while Appendix H provides the extended comparison.

Table 1: **Operator-level comparison.** The second column abstracts the spatial interaction in one evaluation of $f(\cdot)$; non-spatial terms are omitted. "Explicit order" is the largest spatial hop directly accessed within one vector-field evaluation: for polynomial operators this is the highest adjacency power, while local message-passing fields have order 1. It does not denote the receptive field accumulated through integration. "Order weights adapt?" indicates whether the relative weighting of explicit propagation orders is recomputed from the evolving state; a dash denotes that no explicit order decomposition is exposed. Nuanced adaptive baselines are discussed in Appendix H.2.

Method	Spatial propagation in $f(\cdot)$	Explicit order	Spatial parametrization	Order weights adapt?
CGNN (ICML 2020)	$(\hat{A} - I)\mathbf{H}$	1	Fixed diffusion	×
STGODE (KDD 2021)	$\hat{A}\,\mathbf{H}$	1	Fixed graph	×
GRAND (ICML 2021)	$(\mathbf{A}(\mathbf{H}) - I)\mathbf{H}$	1	State-dependent edges	×
STG-NCDE (AAAI 2022)	$\hat{A}(\mathbf{E})\mathbf{H}$	1	Learned graph	×
MTGODE (TKDE 2022)	$(\hat{A}(\mathbf{E}) - I)\mathbf{H}$	1	Learned graph	×
GREAD (ICML 2023)	$(\hat{A} - I)\mathbf{H}$	1	Fixed diffusion	×
GRAM-ODE (TMLR 2023)	coupled $(\hat{A} - I)\mathbf{H}$	1	Fixed graph	×
HOPE (ICML 2023)	$(\hat{D}^t)^{-1}\hat{A}^t\mathbf{H}$	1	Edge-state-dependent graph	×
GG-ODE (KDD 2023)	$\sum_{j\in\mathcal{N}_i(\mathbf{H})} f_{\mathrm{GNN}}(\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_j)$	1	State/env.-dependent neighbors	×
STDDE (WWW 2024)	$\hat{A}\,\mathbf{H}(t-\tau)$	1	Learned delay	×
CSG-ODE (ICML 2025)	$\dot{\mathbf{C}} = \mathcal{G}_\theta(\mathbf{Z}),\ \dot{\mathbf{Z}} = \cdots + g(\mathbf{C})$	–	State-dependent control	–
PIANO (ours)	$\big(\sum_{\ell=0}^{L} \alpha_\ell(\mathbf{H}, t)\hat{A}^\ell\big)\mathbf{H}$	L	State-conditioned order weights	✓

2 Preliminaries: Continuous-Time Graph Dynamics

Let $G = (V, E)$ be a weighted graph with $N = |V|$, adjacency matrix A, and degree matrix D. Throughout, the graph satisfies the regularity conditions in Assumption **(A1)**, which ensure that D is invertible.[1] We use the symmetrically normalised adjacency $\hat{A} = D^{-1/2}AD^{-1/2}$ and the corresponding random-walk transition matrix $P = D^{-1}A$. Since $\hat{A} = D^{1/2}PD^{-1/2}$, their finite-step propagators are related by the same similarity transform (Chung, 1997; Barlow, 2017). We write $d_G(i, j)$ for shortest-path distance in the off-diagonal support graph of A.

An encoder maps the input history to $\mathbf{H}(0) \in \mathbb{R}^{N\times d}$, which evolves according to

$$\frac{d\mathbf{H}(t)}{dt} = f_\theta(\mathbf{H}(t), t), \qquad t \in [0, T], \tag{2}$$

where f_θ is the learned vector field and T is the integration horizon. A decoder produces the forecast from $\mathbf{H}(T)$, or from sampled states $\{\mathbf{H}(t_j)\}$ for multi-step prediction. Many continuous-time graph models can be viewed as combining an instantaneous spatial operator $M(\mathbf{H}, t)$ with a channel map:

$$f_\theta(\mathbf{H}, t) = \mathcal{F}_\theta(M(\mathbf{H}, t)\mathbf{H}, t)\,. \tag{3}$$

Their instantaneous spatial mixing is therefore characterised largely by the structure of $M(\mathbf{H}, t)$, as summarised in Table 1. PIANO instead explicitly decomposes its spatial operator across propagation orders and makes their relative weights depend on $(\mathbf{H}, t)$.

3 Method

PIANO follows an encoder–ODE–decoder architecture. The encoder maps the observed history to $\mathbf{H}(0)$; the latent state is then evolved to time T by a vector field whose primary spatial component is a state-adaptive path-integral propagator, augmented by selective topology-aware node modulation. The resulting state is decoded for the target task. Figure 1 illustrates these two spatial controls.

[1] In this section, A denotes the self-loop-augmented adjacency. Dataset appendices may use A for the raw loop-free adjacency before explicitly forming $A + I$.

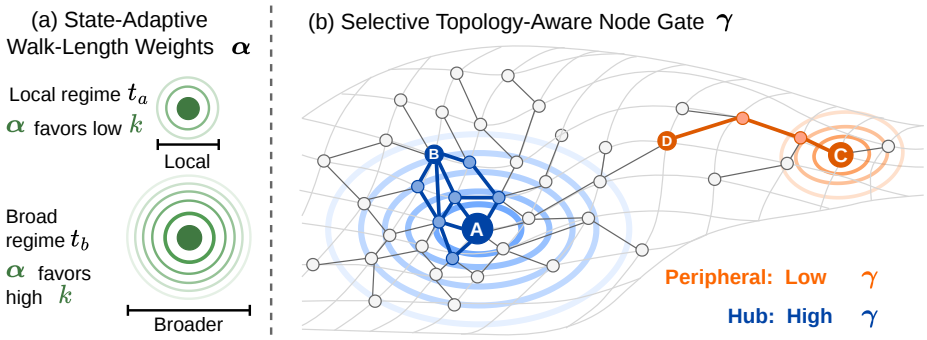


Figure 1: View spatiotemporal dynamics as a changing, wave-like signal over a network. **(a)** PIANO reads the current signal and shifts $\boldsymbol{\alpha}$ from short walks in a local regime t_a to longer walks in a broad regime t_b (§3.1). **(b)** The selective topology-aware gate $\boldsymbol{\gamma}$ modulates where that information produces stronger updates. Figure 3 further shows how a structural hub in loop-rich motifs has a stronger closed-walk signature than a peripheral node (§3.2). Together, the two mechanisms adapt how far and where information propagates as the latent state evolves.

3.1 State-Adaptive Path-Integral Propagator

Following the walk-length interpretation in Equation (1), each $\hat{A}^k$ aggregates contributions carried by length-k graph walks. We define the state-adaptive path-integral propagator as

$$M_\theta(\mathbf{H}, t) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\hat{A}^k, \tag{4}$$

where $\boldsymbol{\alpha}(\mathbf{H}, t)$ is a state-conditioned distribution over walk lengths $k = 0, \ldots, L$, and L is the maximum explicit order. Equivalently, if $K_{\mathbf{H},t} \sim \mathrm{Cat}(\boldsymbol{\alpha}(\mathbf{H}, t))$, then $M_\theta(\mathbf{H}, t) = \mathbb{E}[\hat{A}^{K_{\mathbf{H},t}} \mid \mathbf{H}, t]$. This mirrors the finite-walk viewpoint of classical diffusion in Equation (1): diffusion uses a fixed Poisson law, whereas PIANO learns a finite walk-length law conditioned on the evolving state.

The walk-length weights are predicted from the current graph state and integration time:

$$\boldsymbol{\alpha}(\mathbf{H}, t) = \mathrm{softmax}(\mathrm{MLP}_\omega[\mathrm{pool}(\mathbf{H}) \| \tau(t)]) \in \Delta^L, \tag{5}$$

where pool denotes node-wise mean pooling and $\|$ denotes feature concatenation. The time encoding is the scalar solver time, $\tau(t) = t$, broadcast across samples. $\Delta^L = \{\boldsymbol{\alpha} \in \mathbb{R}^{L+1} : \alpha_k \geq 0, \sum_{k=0}^{L} \alpha_k = 1\}$ is the probability simplex over walk lengths $0, \ldots, L$.

The corresponding expected walk length is

$$\bar{k}(\mathbf{H}, t) = \sum_{k=0}^{L} k\, \alpha_k(\mathbf{H}, t), \tag{6}$$

which we use as an interpretable diagnostic of effective propagation scale.

This formulation lets the effective spatial range vary with the evolving state: mass on shorter walk lengths favours local interaction, mass on longer walk lengths favours broader-range interaction, and mass spread across several lengths yields multiscale mixing, as illustrated in Figure 1(a). The simplex constraint makes $M_\theta(\mathbf{H}, t)$ an interpretable convex combination of graph powers. Since $\|\hat{A}^k\|_2 \leq 1$, every realised operator satisfies $\|M_\theta(\mathbf{H}, t)\|_2 \leq 1$ (Lemma 1). Its polynomial form admits efficient evaluation without materialising $\hat{A}^2, \ldots, \hat{A}^L$. Define

$$\mathbf{G}_L = \alpha_L(\mathbf{H}, t)\mathbf{H}, \qquad \mathbf{G}_k = \alpha_k(\mathbf{H}, t)\mathbf{H} + \hat{A}\mathbf{G}_{k+1}, \quad k = L-1, \ldots, 0, \qquad M_\theta(\mathbf{H}, t)\mathbf{H} = \mathbf{G}_0. \tag{7}$$

The recursion requires L sparse–dense matrix multiplications and no dense graph powers, giving $\mathcal{O}(L|E|d)$ propagation cost. Complete vector-field and complexity details are given in Appendix B.2.

3.2 SELECTIVE TOPOLOGY-AWARE NODE MODULATION

The walk-length weights $\boldsymbol{\alpha}(\mathbf{H}, t)$ are shared across nodes: they adapt the propagation scale of the graph as a whole, while $M_\theta(\mathbf{H}, t)\mathbf{H}$ still produces node-specific aggregates. To provide lightweight node-wise modulation when structural roles are informative, we introduce a gate $\boldsymbol{\gamma} \in (0, 1)^N$ that controls each destination node's response:

$$f_\theta(\mathbf{H}, t) = \boldsymbol{\gamma} \odot \sigma(\mathcal{W}(M_\theta(\mathbf{H}, t)\mathbf{H})) - \delta\mathbf{H}, \tag{8}$$

where $\mathcal{W}$ is a shared affine channel map, σ is GELU, and $\delta \geq 0$ is an optional dissipative term. The product $\odot$ broadcasts γ_i across the d channels of node i.

For the i-th node,

$$\begin{aligned} \gamma_i &= \mathrm{sigmoid}(a\, c_i(\mathbf{H}, t) + b\|\mathbf{h}_i\|_2 + b_0)\,, \\ c_i(\mathbf{H}, t) &= \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)(\hat{A}^k)_{ii}, \end{aligned} \tag{9}$$

where a, b, b_0 are learned scalars and $c_i(\mathbf{H}, t) = [M_\theta(\mathbf{H}, t)]_{ii}$ is an $\boldsymbol{\alpha}$-weighted closed-walk score: $(\hat{A}^k)_{ii}$ measures the normalised weighted mass of length-k closed walks returning to node i (Estrada & Rodríguez-Velázquez, 2005; Estrada & Hatano, 2008). The node norm $\|\mathbf{h}_i\|_2$ captures current feature activity, while the signs of a and b determine whether increasing the closed-walk score or feature activity raises or lowers the gate. Given precomputed diagonals $\{\mathrm{diag}(\hat{A}^k)\}_{k=0}^{L}$, the gate adds only three scalar parameters. Figure 1(b) illustrates the positive-modulation case; complexity details and the hub–periphery example are provided in Appendices B.2 and C.9.

The encoder E_ϕ, vector field f_θ, and decoder D_ψ are trained jointly by minimising the task loss on $D_\psi(\mathrm{ODESolve}(f_\theta, E_\phi(\mathbf{X}), [0, T]))$, where $\mathbf{X}$ denotes the observed input history. Architectures, optimisation settings, batching details, and pseudocode are provided in Appendix B.1.

4 THEORETICAL ANALYSIS: STABILITY AND NON-LOCAL REACH

Equations (4) and (8) expose three distinct controls in PIANO: L sets the maximum explicit order of each realised propagator, $\boldsymbol{\alpha}(\mathbf{H}, t)$ distributes state-dependent mass across walk lengths $0, \ldots, L$, and integration over $[0, T]$ composes the resulting updates through continuous time.

We establish four complementary properties of this construction. **Pointwise non-expansive propagation.** Every state-adaptive path-integral propagator satisfies $\|M_\theta(\mathbf{H}, t)\|_2 \leq 1$ (Lemma 1). **Well-posed nonlinear flow.** The full state-dependent ODE admits a unique flow that is a homeomorphism and is bi-Lipschitz onto its image on bounded initial-state sets (Theorem 1). **Controlled non-local reach.** For a frozen linear surrogate, L and T jointly determine a sufficient finite-hop approximation radius (Theorem 2). **Topology-sensitive modulation.** The selective node gate's closed-walk score can distinguish structural configurations even when the transported activation is unchanged (Proposition 1). The finite-hop result applies only to the frozen linear surrogate; the other statements concern the realised path-integral propagator, nonlinear flow, and topology-aware gate of the implemented model, respectively.

We use $\|\cdot\|_2$ and $\|\cdot\|_F$ for spectral and Frobenius norms and use the following analytical conditions and architectural properties:

- **(A1)** A is real, symmetric, entrywise nonnegative, has self-loops, and has positive degrees. Consequently, $\hat{A} = D^{-1/2}AD^{-1/2}$ is symmetric and $\|\hat{A}\|_2 \leq 1$.
- **(A2)** The controller in Equation (5) uses a softmax, hence $\boldsymbol{\alpha}(\mathbf{H}, t) \in \Delta^L$ for all $(\mathbf{H}, t)$.
- **(A3)** The entrywise activation σ is globally L_σ-Lipschitz and satisfies $|\sigma(0)| = s_0 < \infty$.
- **(A4)** Equation (5) uses node-wise mean pooling, MLP_ω has finite weights and globally Lipschitz activations, and $\tau(t)$ is continuous on $[0, T]$. The gate has the form in Equation (9) with finite parameters.
- **(A5)** For finite $B_W, B_0 \geq 0$, the shared affine map satisfies
$$\|\mathcal{W}(X) - \mathcal{W}(Y)\|_F \leq B_W\|X - Y\|_F, \qquad \|\mathcal{W}(X)\|_F \leq B_W\|X\|_F + B_0,$$
and the leak satisfies $\delta \geq 0$.

Appendix C.1 verifies the basic analytic properties of the implemented normalisation, GELU, softmax, mean pooling, time encoding, and affine channel map. Appendix C.3 derives the regularity of the walk-length weights, path-integral propagator, and selective topology-aware node gate.

4.1 Non-expansive spatial filtering and a well-posed flow

Lemma 1 (Spectral form and pointwise non-expansiveness)**.** *Under* (A1)–(A2)*, let* $\hat{A} = U\Lambda U^\top$*, where* $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$ *and* $\lambda_i \in [-1, 1]$*. For every realised* $(\mathbf{H}, t)$*, the operator in Equation* (4) *satisfies*

$$M_\theta(\mathbf{H}, t) = U p_{\boldsymbol{\alpha}}(\Lambda) U^\top, \qquad p_{\boldsymbol{\alpha}}(\lambda) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t) \lambda^k, \tag{10}$$

and $\|M_\theta(\mathbf{H}, t)\|_2 = \max_i |p_{\boldsymbol{\alpha}}(\lambda_i)| \le 1$*. Consequently, for every* $X \in \mathbb{R}^{N\times d}$*,* $\|M_\theta(\mathbf{H}, t)X\|_F \le \|X\|_F$*. Thus, for each realised state and time, the simplex-constrained spatial operator is non-expansive in its input. Proof: Appendix C.2.*

The lemma controls each realised spatial operator. To handle its state dependence, Appendix C.3 establishes uniform Lipschitz bounds for the adaptive weights, spatial operator, and topology-aware gate, while Appendix C.4 gives an explicit finite-horizon trajectory bound under *(A1)–(A5)*. Hence the full vector field is locally Lipschitz on bounded sets, and its trajectories cannot escape to infinity in finite time.

Theorem 1 (Well-posed continuous flow)**.** *Under* (A1)–(A5)*, Equation* (2) *admits a unique solution on every finite interval* $[0, T]$ *from every initial state* $\mathbf{H}_0$*. Its time-*T *flow map*

$$\Phi_T : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}, \qquad \mathbf{H}_0 \mapsto \mathbf{H}(T; \mathbf{H}_0),$$

is a homeomorphism. Moreover, its restriction to every bounded initial-state set is bi-Lipschitz onto its image. Specifically, for

$$\mathcal{B}_{R_0} = \{\mathbf{H}_0 : \|\mathbf{H}_0\|_F \le R_0\},$$

there exists a finite $C_{T,R_0} \ge 1$ *such that, for all* $\mathbf{H}_{0,1}, \mathbf{H}_{0,2} \in \mathcal{B}_{R_0}$*,*

$$C_{T,R_0}^{-1} \|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F \le \|\Phi_T(\mathbf{H}_{0,1}) - \Phi_T(\mathbf{H}_{0,2})\|_F \le C_{T,R_0} \|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F.$$

The upper bound controls sensitivity to initial conditions, while the lower bound prevents distinct initial states from collapsing under the exact flow. Proof: Appendix C.5.

4.2 Spatial order and integration time jointly control reach

To isolate the roles of spatial order L and integration horizon T, consider the frozen linear surrogate $\dot{\mathbf{h}} = M_\theta \mathbf{h}$, where $M_\theta = \sum_{k=0}^{L} \alpha_k \hat{A}^k$ and $\boldsymbol{\alpha} \in \Delta^L$ is fixed. Its exact propagator is $\Pi_T = e^{TM_\theta} = \sum_{m=0}^{\infty} \frac{T^m}{m!} M_\theta^m$. A single application of M_θ has graph-polynomial degree at most L, whereas M_θ^m has degree at most mL. Thus L controls the spatial orders available within each application, while continuous integration composes these interactions over time.

Theorem 2 (Receptive-field bound for frozen linear propagation)**.** *Under* (A1)–(A2)*, let* $\Pi_{T,\kappa} := \sum_{m=0}^{\kappa} \frac{(TM_\theta)^m}{m!}$*. Then* $\Pi_{T,\kappa}$ *is a polynomial in* $\hat{A}$ *of degree at most* κL*, so* $[\Pi_{T,\kappa}]_{ij} = 0$ *whenever* $d_G(i, j) > \kappa L$*, and*

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le e^T \frac{T^{\kappa+1}}{(\kappa+1)!}.$$

Hence, for every $\varepsilon \in (0, 1)$*, there exists* $\kappa(\varepsilon) = \mathcal{O}(T + \log(1/\varepsilon))$ *such that* $\|\Pi_T - \Pi_{T,\kappa(\varepsilon)}\|_2 \le \varepsilon$*. A sufficient radius bound is therefore* $r(\varepsilon) := L\kappa(\varepsilon) = \mathcal{O}(L(T + \log(1/\varepsilon)))$*. This is a sufficient upper bound for the frozen linear surrogate, not a necessary radius or an exact equivalence between* L *and* T*. Proof: Appendix C.6.*

Interpretation. The theorem separates two complementary controls: L determines the maximum spatial order available within each operator application, whereas T controls how these interactions are composed through continuous depth. The result applies only to the frozen linear surrogate; Appendix D.6 examines the L–T interaction empirically in the trained nonlinear model.

4.3 TOPOLOGY CAN PROVIDE AN AUXILIARY CONTROL SIGNAL

For node i, define the order-L closed-walk signature $\mathbf{d}_i^{(L)} := \big((\hat{A}^0)_{ii}, \ldots, (\hat{A}^L)_{ii}\big)^\top$, so that the topology score in Equation (9) is $c_i(\mathbf{H}, t) = \boldsymbol{\alpha}(\mathbf{H}, t)^\top \mathbf{d}_i^{(L)}$.

Proposition 1 (Topology-conditioned field separation)**.** *Consider two node configurations with different closed-walk signatures, $\mathbf{d}_i^{(L)} \neq \mathbf{d}_j'^{(L)}$. Then there exists a common admissible walk-length distribution $\boldsymbol{\alpha} \in \operatorname{int} \Delta^L$ for which their topology scores differ. Moreover, if the two configurations have the same local state and the same nonzero transported activation, then a gate with $a \neq 0$ yields different vector-field updates. Hence the selective topology-aware node gate can distinguish structural differences even when the transported message itself is unchanged. Proof: Appendix C.7.*

Interpretation. The proposition shows that $\boldsymbol{\alpha}$ selects propagation scales while also weighting the closed-walk statistics through which $\boldsymbol{\gamma}$ can selectively modulate node updates. This is an existence result rather than a claim that every trained model must rely strongly on the structural channel. As a canonical witness, the gate separates C_6 from $2K_3$, which are 1-WL-indistinguishable under constant node features (Appendix C.8). Section 5.1 tests whether this structural signal is functionally used in learned dynamics.

5 EXPERIMENTS

We evaluate PIANO around three questions: **Q1: state-adaptive propagation scale**—does the walk-length distribution adapt to changing interaction range and is it functionally used; **Q2: selective topology-aware modulation**—when does the node gate provide useful structural information; and **Q3: transfer and scalability**—do these mechanisms transfer to real spatiotemporal and large-graph problems with favourable accuracy–efficiency trade-offs?

Our Graph-wave-Scale benchmark (protocol in Appendix D.1) isolates Q1 by varying the hidden interaction range on a fixed graph, while Springs (Kipf et al., 2018) isolates Q2 through changing structural node roles. For Q3, Ocean ΔSLA (AVISO & NOAA NMFS SWFSC ERD, 2012) and SEVIR (Veillette et al., 2020) test transfer to natural spatiotemporal dynamics with different spatial scales, while static graph benchmarks test whether the learned propagation mechanism extends beyond forecasting; ogbn-arxiv additionally stresses scalability on a large sparse graph. Further experimental details are in Appendix B.

5.1 CONTROLLED VALIDATION OF THE ADAPTIVE MECHANISMS

Graph-wave-Scale provides a controlled instantiation of Figure 1(a), isolating **adaptive propagation-scale weighting** (Q1) by varying the hidden interaction range $\rho \in \{1, 2, 4\}$ on a fixed degree-4 circulant graph with the node gate disabled, under the shared $(L, T) = (8, 1)$ setting. Springs isolates **selective topology-aware modulation** (Q2) by resampling the interaction graph across trajectories, thereby changing node structural roles, at weak ($\kappa = 0.1$) and strong ($\kappa = 1.0$) coupling. Table 2 reports a compact forecasting comparison, while Figure 2 diagnoses the primary walk-length mechanism. Full baseline results and graph-access grouping are given in Table 7; controller and gate ablations appear in Appendix D.

Table 2: **Controlled forecasting comparison.** *Graph-wave-Scale parameter counts. Full comparisons and gate/controller ablations are in Appendix D.

Model	Params*	Graph-wave-Scale ($\times 10^{-5}$)	Springs $\kappa = 0.1$ ($\times 10^{-4}$)	Springs $\kappa = 1.0$ ($\times 10^{-3}$)
STGODE	136.8K	14.3 ± 1.5	14.3 ± 0.6	69.1 ± 1.5
GRAND-nl	32.6K	105 ± 4	11.7 ± 0.2	55.0 ± 0.7
HOPE	28.6K	71.7 ± 4.2	10.4 ± 0.9	6.64 ± 0.26
GG-ODE	149.8K	28.5 ± 9.4	349 ± 61	137 ± 7
CSG-ODE	553.7K	3.26 ± 0.58	22.4 ± 1.5	48.6 ± 4.4
PIANO	33.4K	**3.79** ± 0.65	**5.28** ± 0.75	**3.79** ± 0.13

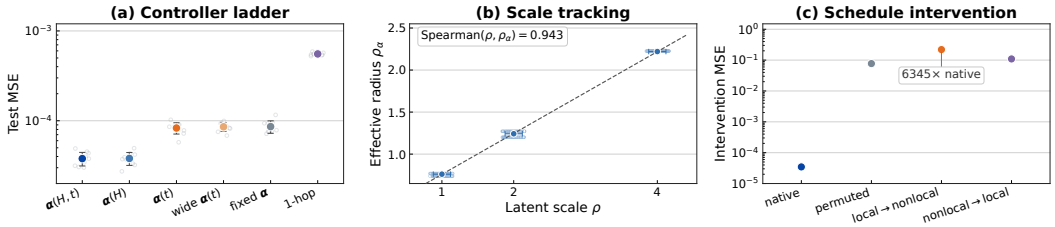


Figure 2: **Controlled validation of adaptive propagation scale.** **(a)** Graph-wave-Scale controller ladder. **(b)** Learned effective propagation radius tracks the latent interaction scale over 200 held-out families for the seed-42 diagnostic checkpoint ($\rho_s = 0.943$). **(c)** Permuting or cross-swapping the learned walk-length schedule increases MSE by orders of magnitude.

Q1. State-adaptive propagation scale. PIANO achieves $3.79\pm0.65 \times 10^{-5}$ MSE, improving over the learned fixed-$\boldsymbol{\alpha}$ baseline ($8.61\pm1.37 \times 10^{-5}$) and Graph WaveNet ($5.01\pm1.15 \times 10^{-5}$); CSG-ODE is lower ($3.26\pm0.58 \times 10^{-5}$) but uses $16.6\times$ more parameters. The controller ladder attributes the gain primarily to conditioning on the evolving state rather than explicit time input or controller capacity (Figure 2a). More importantly, the radius inferred from $\boldsymbol{\alpha}(\mathbf{H}, t)$ tracks the hidden interaction scale for the seed-42 diagnostic checkpoint ($\rho_s = 0.943$), while mismatched walk-length schedules increase error by orders of magnitude (Figure 2b–c). An $L \times T$ sweep further shows complementary gains from spatial order and integration horizon (Appendix D.6), consistent with Theorem 2.

Q2. Selective topology-aware modulation. At weak coupling ($\kappa = 0.1$), PIANO obtains 5.28×10^{-4} MSE, whereas feature-only, shuffled-c_i, and no-gate variants yield 8.85–9.12×10^{-4} (Figure 5). Thus correctly aligned closed-walk information helps in this controlled regime, supporting Proposition 1; at stronger coupling, the gate effect becomes small. Full gate decompositions and ablations are reported in Appendix D.5.

5.2 Transfer and Scalability

Table 3 evaluates Q3 across Ocean ΔSLA, SEVIR, and static graph learning.

Ocean ΔSLA. Among the continuous-time graph models compared in Tables 3 and 12, PIANO attains the highest Mean ACC (0.509 ± 0.001), the strongest week-$+1$ and week-$+4$ ACC, and the second-lowest peak memory; the learned-fixed $\boldsymbol{\alpha}$ control is marginally faster to train and infer. Against broader forecasting families, ConvLSTM retains the highest absolute Mean ACC and MgNO the lowest raw runtime, while PIANO occupies a favourable compact accuracy–resource regime. Qualitative forecast maps for an illustrative held-out window are in Appendix E.4.

SEVIR. Under the primary window-split $5\to2$ protocol, Graph WaveNet attains the lowest displayed MSE and the highest global CSI, while PIANO ranks second with 69.9K parameters and improves upon the matched latent CNN-only backbone, GRAND-nl, HOPE, and the matched one-hop Graph ODE on MSE and both CSI thresholds (Table 3). An event-split $5\to2$ audit changes the exact ordering but retains the broader transfer conclusion (Table 18); the completed window-split $5\to10$ experiment is reported only as a long-horizon stress test in Appendix F.2. Replacing the native walk-length schedule by a batch permutation or validation-set mean worsens $4/5$ checkpoints, whereas permuting the node-wise topology score c_i has negligible effect. Thus SEVIR provides additional evidence for functional use of the walk-length schedule, but not of topology-aware gating on the nearly regular latent radar graph. Full robustness and CSI audits are in Appendix F.

Static graph learning. Across the three homophilic citation benchmarks, PIANO is competitive on Cora and Citeseer and obtains the highest PubMed accuracy (0.895 ± 0.003). On ogbn-arxiv (169,343 nodes), PIANO reaches 0.666 ± 0.007, significantly exceeding matched GPR-GNN and GCN ($p = 0.027$ and 0.004, paired Wilcoxon). Full-graph PIANO uses four vector-field evaluations, requiring 100 ms inference and 1.34 GB peak memory; complete resource comparisons are in Appendix G.

Table 3: **Transfer and scalability across complementary regimes.** For Ocean, bold marks the best result within the continuous-time graph family; contextual forecasters are references. Completed SEVIR rows use ten matched window-split seeds; CSI is globally pooled over five audited seeds. ‡GREAD on ogbn-arxiv uses five seeds; others use ten unless noted. Additional results: Tables 12–14 (Ocean), 15 and 16 (SEVIR short-horizon 5→2), 20 and 22 (static graphs).

Ocean ΔSLA (8→4; ACC ↑, efficiency ↓)

		Forecast skill			Efficiency		
Model	Params	ACC_{+1}	ACC_{+4}	Mean ACC	T_{ep} (s)	T_{inf} (ms)	Peak MB
Contextual forecasters							
ConvLSTM	112K	0.767 ±0.004	0.369 ±0.006	0.547 ±0.004	1.33	0.69	503
MgNO	736.6K	0.763 ±0.004	0.320 ±0.005	0.508 ±0.003	0.39	0.16	192
Graph WaveNet	82.9K	0.779 ±0.001	0.322 ±0.003	0.515 ±0.002	2.06	1.23	1078
Continuous-time graph models							
STGODE	115.7K	0.773 ±0.000	0.279 ±0.001	0.486 ±0.001	1.23	0.65	536
GRAND-nl	26.4K	0.752 ±0.001	0.234 ±0.001	0.440 ±0.001	3.41	2.92	662
HOPE	**22.4K**	0.766 ±0.001	0.278 ±0.001	0.481 ±0.000	2.05	2.24	137
GG-ODE	149.6K	0.738 ±0.011	0.320 ±0.007	0.498 ±0.007	13.03	7.92	6550
CSG-ODE	1.47M	0.760 ±0.002	0.298 ±0.009	0.489 ±0.004	6.41	3.05	1954
PIANO	43.1K	**0.776** ±0.000	**0.323** ±0.001	**0.509** ±0.001	**0.77**	**0.41**	**127**

SEVIR VIL (5→2, window split; MSE ↓, global CSI ↑)

Model	Params	MSE	ΔMSE vs. PIANO	CSI@0.5	CSI@0.7
Latent CNN-only	69.6K	0.00580 ±0.00004	+0.00014	0.7190 ±0.0006	0.7418 ±0.0008
Graph WaveNet	171.2K	**0.00559** ±0.00006	−0.00006	**0.7288** ±0.0015	**0.7711** ±0.0124
GRAND-nl	69.5K	0.00582 ±0.00003	+0.00016	0.7192 ±0.0012	0.7393 ±0.0055
HOPE	65.4K	0.00580 ±0.00004	+0.00014	0.7202 ±0.0019	0.7420 ±0.0015
One-hop Graph ODE	65.4K	0.00579 ±0.00005	+0.00013	0.7202 ±0.0028	0.7397 ±0.0073
PIANO	69.9K	<u>0.00566</u> ±0.00008	–	<u>0.7273</u> ±0.0050	<u>0.7423</u> ±0.0063

Static node classification (accuracy ↑)

	Homophilic			Large-scale
Model	Cora	Citeseer	PubMed	ogbn-arxiv
GCN	0.874 ±0.013	**0.761** ±0.011	0.871 ±0.007	0.655 ±0.002
GPR-GNN	**0.878** ±0.013	0.760 ±0.013	0.881 ±0.007	0.660 ±0.003
GREAD‡	0.871 ±0.013	**0.761** ±0.012	0.870 ±0.006	0.573 ±0.006
PIANO	0.873 ±0.014	0.760 ±0.010	**0.895** ±0.003	**0.666** ±0.007

6 Conclusion

We introduced PIANO, a continuous-time graph model whose instantaneous spatial operator is a state-adaptive path-integral propagator under a state-conditioned distribution over walk lengths. An optional topology-aware gate provides auxiliary node-wise modulation. Nested sparse recursion evaluates the operator in $\mathcal{O}(L|E|d)$ time; each realised spatial operator is pointwise non-expansive and the nonlinear flow is well-posed, while a frozen linear analysis characterises how L and T jointly determine a sufficient finite-hop approximation radius.

Taken together, the theory and controlled interventions support the central design hypothesis that a continuous graph vector field should redistribute mass across spatial orders as the state evolves, rather than fix that mass a priori or conflate it with temporal order or neighbour selection. The empirical study goes beyond aggregate predictive performance to validate the intended behaviour of PIANO: adaptive walk-length weighting tracks the underlying interaction range, is functionally used by the learned dynamics, and shows complementary gains with integration horizon. Transfer experiments show that this inductive bias applies across spatiotemporal forecasting and sparse large-graph learning, but gains are not uniform: ConvLSTM retains the highest Ocean Mean ACC, SimVP the higher pooled CSI on the 5→10 stress test, and the ogbn-arxiv gain incurs additional resource cost. These results support a domain-dependent inductive bias rather than universal superiority, while null and regime-dependent results delimit the utility of the optional gate.

AI USE STATEMENT

In this work, we used generative AI tools to draft parts of the manuscript, edit the paper for readability, and identify and retrieve relevant literature. The authors reviewed and revised all AI-assisted text and independently verified retrieved sources and associated claims against the original publications. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The model definition and computational complexity are given in Section 3; assumptions and theoretical statements appear in Section 4, with complete proofs in Appendix C. Appendix B provides pseudocode, dataset processing and split details, training protocols, and the configurations used for the reported experiments.

REFERENCES

Juan A. Acebrón, Luis L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, 77(1):137–185, 2005. doi: 10.1103/RevModPhys.77.137.

AVISO and NOAA NMFS SWFSC ERD. Sea surface height, absolute, Aviso, 0.25 degrees, global, 1992–2012, science quality (1 day composite). NOAA CoastWatch ERDDAP dataset `erdTAssh1day`, 2012. URL `https://coastwatch.pfeg.noaa.gov/erddap/info/erdTAssh1day/index.html`. Weekly North Atlantic ΔSLA derived from absolute SSH.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Martin T. Barlow. *Random Walks and Heat Kernels on Graphs*, volume 438 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2017. doi: 10.1017/9781107415690.

Alessio Borgi, Fabrizio Silvestri, and Pietro Liò. Polynomial neural sheaf diffusion: A spectral filtering approach on cellular sheaves. *arXiv preprint arXiv:2512.00242*, 2025.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-Unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision – ECCV 2022 Workshops*, volume 13803 of *Lecture Notes in Computer Science*, pp. 205–218. Springer, 2023. doi: 10.1007/978-3-031-25066-8_9.

Benjamin Paul Chamberlain, James Rowbottom, Maria Gorinova, Stefan Webb, Emanuele Rossi, and Michael M. Bronstein. GRAND: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418, 2021.

Ashesh Chattopadhyay, Michael Gray, Tianning Wu, Anna B. Lowe, and Ruoying He. OceanNet: a principled neural operator-based digital twin for regional oceans. *Scientific Reports*, 14:21181, 2024. doi: 10.1038/s41598-024-72145-0.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6572–6583, 2018.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pp. 6367–6374. AAAI Press, 2022.

Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. GREAD: Graph neural reaction-diffusion networks. In *Proceedings of the 40th International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5722–5747, 2023.

Fan Chung. The heat kernel as the PageRank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007. doi: 10.1073/pnas.0708838104.

Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997. ISBN 9780821803158.

Federico Errica, Henrik Christiansen, Viktor Zaverkin, Takashi Maruyama, Mathias Niepert, and Francesco Alesiani. Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15490–15515. PMLR, 2025. URL https://proceedings.mlr.press/v267/errica25a.html.

Ernesto Estrada and Naomichi Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

Ernesto Estrada and Juan A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 364–373. ACM, 2021.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Benjamin Chamberlain, Michael Bronstein, and Federico Monti. SIGN: Scalable inception graph neural networks. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3170–3180, 2022a.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022b.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Juncai He, Xinliang Liu, and Jinchao Xu. MgNO: Efficient parameterization of linear operators via multigrid. In *International Conference on Learning Representations*, 2024.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Zijie Huang, Yizhou Sun, and Wei Wang. Generalizing graph ODE for learning complex system dynamics across environments. In *KDD '23: The 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, August 6-10, 2023*, pp. 798–809. ACM, 2023.

Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural ODEs. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, 2023. doi: 10.1109/TKDE.2022.3221989.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pp. 2693–2702, 2018.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*, 2017.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 338–348, 2020. doi: 10.1145/3394486.3403076.

Zibo Liu, Parshin Shojaee, and Chandan K. Reddy. Graph-based multi-ODE neural networks for spatio-temporal traffic forecasting. *Transactions on Machine Learning Research (TMLR)*, 2023.

Qingqing Long, Zheng Fang, Chen Fang, Chong Chen, Pengfei Wang, and Yuanchun Zhou. Unveiling delay effects in traffic forecasting: A perspective from spatial-temporal delay differential equations. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*, pp. 1035–1044. ACM, 2024. doi: 10.1145/3589334.3645688.

Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. HOPE: high-order graph ODE for modeling interacting dynamics. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23124–23139. PMLR, 2023.

Zheng Ma, Junyu Xuan, Yu Guang Wang, Ming Li, and Pietro Liò. Path integral based convolution and pooling for graph neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-coupled oscillator networks. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*, 2022.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 802–810, 2015.

Geoffrey K. Vallis. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. Cambridge University Press, 2 edition, 2017.

N. M. van Dijk, S. P. J. van Brummelen, and R. J. Boucherie. Uniformization: Basics, extensions and applications. *Performance Evaluation*, 118:8–32, 2018. doi: 10.1016/j.peva.2017.09.008.

Mark Veillette, Siddharth Samsi, and Chris Mattioli. SEVIR : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, volume 33, pp. 22009–22019, 2020.

Zhiqiang Wang, Xiaoyi Wang, and Jianqing Liang. CSG-ODE: ControlSynth graph ODE for modeling complex evolution of dynamic graphs. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64670–64689. PMLR, 2025.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 1907–1913, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 753–763. ACM, 2020.

Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, Proceedings of Machine Learning Research, pp. 10432–10441. PMLR, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 3634–3640, 2018.

Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4560–4570, 2022. doi: 10.1145/3534678.3539121.

Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I. Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with NowcastNet. *Nature*, 619 (7970):526–532, 2023.

A Appendix Guide and Claim–Evidence Map

The appendix is organised around the main claims of the paper.

Appendix B. Implementation and reproducibility details.

Appendix C. Proofs of non-expansiveness, well-posedness, and the joint L–T reach bound.

Appendix D. Controlled tests of walk-length adaptation and the optional gate.

Appendix E. Ocean ΔSLA transfer experiments.

Appendix F. SEVIR radar-nowcasting transfer experiments.

Appendix G. Citation-graph transfer, ogbn-arxiv scaling, and gate checks.

Appendix H. Extended comparison with adaptive-hop, graph-ODE, and sheaf methods.

The central empirical claim is that the state-conditioned walk-length distribution $\boldsymbol{\alpha}(\mathbf{H}, t)$ parameterises a state-adaptive path-integral propagator, the primary mechanism of PIANO. The selective topology-aware node gate $\boldsymbol{\gamma}$ is an auxiliary control whose utility appears when node roles carry useful heterogeneous structure. Null gate results—a nearly open ocean gate and weak SEVIR/ogbn interventions—are retained as scope rather than omitted.

Table 4: Claim-to-evidence map for the appendix. Proofs appear in Appendix C; controlled mechanism experiments in Appendix D.

Main-paper claim	Appendix evidence	What it establishes
State-adaptive flow is well-posed; the realised spatial operator is non-expansive	Lemma 1, Theorem 1	pointwise non-expansiveness and well-posed flow
L and T play complementary roles	Theorem 2, Appendix D.6	theory/empirical consistency
$\boldsymbol{\alpha}(\mathbf{H}, t)$ adapts walk-length weighting	Graph-wave-Scale controller ladder	state conditioning drives the gain
Learned walk-length schedule is functionally used	ρ_α diagnostic and walk-length schedule intervention	not merely visualization
Selective topology-aware node gate can distinguish topology	Proposition 1, $C_6/2K_3$ witness	representational capability
Selective node gate helps when structural modulation is useful	Springs $\kappa = 0.1$	selective learned utility
Method transfers beyond controlled systems	Ocean ΔSLA and SEVIR	practical relevance
Full-graph scalability	ogbn-arxiv	feasible full-graph propagation on a large benchmark

Table 5: Main-benchmark statistics. Split entries count forecasting windows. SEVIR distinguishes input pixels from the 32^2 latent graph nodes. NRI Springs samples a new interaction graph per trajectory.

Dataset	Input / graph N	Graph	Hist.$\rightarrow$Hor.	Train/Val/Test
Graph-wave	64 / 64	$C_{64}(1, 2)$	$10 \rightarrow 20$	7998/999/999
NRI Springs	10 / 10	per-trajectory	$10 \rightarrow 20$	8000/1000/1000
Ocean ΔSLA	$\sim$944 / $\sim$944	k-NN (k=8)	$8 \rightarrow 4$ wk	chrono. split
SEVIR (VIL)	128^2 px / 32^2	k-NN	$5 \rightarrow 2$; $5 \rightarrow 10$	308K / 17K / 353K

B Implementation and Reproducibility

Unless a dataset section specifies otherwise, training follows Algorithms 1–3 and the configuration in Table 6. For every canonical PIANO run, the raw adjacency A_0 is symmetrised, one unit self-loop is added per node, and propagation uses $\hat{A} = D^{-1/2}(A_0 + I)D^{-1/2}$. Unless stated otherwise, checkpoints are selected by validation performance. Quantitative results are mean $\pm$ sample standard deviation over ten seeds unless a table notes a smaller seed count.

B.1 End-to-End Training Procedure

This appendix spells out the training pipeline summarised in Section 3. Each ODE solver call to f_θ is Algorithm 1; Algorithms 2 and 3 wrap it in encode–integrate–decode and the outer optimisation loop.

Forecasting experiments use AdamW, whereas static-graph experiments use Adam; exact configurations are given in Table 6.

B.2 Computational Complexity

Nested SpMM evaluation. The recursion in Equation (7) is obtained from

$$\begin{aligned} M_\theta(\mathbf{H}, t)\mathbf{H} &= \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\hat{A}^k\mathbf{H} \\ &= \alpha_0\mathbf{H} + \hat{A}\Big(\alpha_1\mathbf{H} + \hat{A}(\cdots + \hat{A}\alpha_L\mathbf{H})\Big), \end{aligned} \tag{11}$$

Algorithm 1 PIANO vector field $f_\theta(\mathbf{H}, t)$ (one ODE-solver evaluation)

Require: state $\mathbf{H} \in \mathbb{R}^{N\times d}$, time t, normalised sparse adjacency $\hat{A}$, maximum explicit order L, precomputed diagonals $\{\mathrm{diag}(\hat{A}^k)\}_{k=0}^{L}$
Require: parameters: affine channel map $\mathcal{W}$, walk-length MLP MLP_ω, gate scalars a, b, b_0, leak δ

$\bar{\mathbf{h}} \leftarrow \mathrm{pool}(\mathbf{H})$ {permutation-invariant global summary}
$\boldsymbol{\alpha} \leftarrow \mathrm{softmax}\big(\mathrm{MLP}_\omega[\,\bar{\mathbf{h}} \,\|\, \tau(t)\,]\big)$ {walk-length weights on the simplex Δ^L}
$\mathbf{G} \leftarrow \alpha_L\,\mathbf{H}$ {nested init at order L}
for $k = L-1$ **to** 0 **do**
  $\mathbf{G} \leftarrow \alpha_k\,\mathbf{H} + \hat{A}\,\mathbf{G}$ {one SpMM, $\mathcal{O}(|E|\,d)$}
end for
// now $\mathbf{G} = M_\theta(\mathbf{H}, t)\,\mathbf{H}$
$\mathbf{Z} \leftarrow \sigma(\mathcal{W}(\mathbf{G}))$ {channel mixing, $\sigma = \mathrm{GELU}$}
$c_i \leftarrow \sum_{k=0}^{L} \alpha_k\,(\hat{A}^k)_{ii}$ **for all** i {$\boldsymbol{\alpha}$-weighted centrality, reuses diagonals, $\mathcal{O}(LN)$}
$\gamma_i \leftarrow \mathrm{sigmoid}\big(a\,c_i + b\,\|\mathbf{h}_i\|_2 + b_0\big)$ {per-node centrality gate}
return $\boldsymbol{\gamma} \odot \mathbf{Z} - \delta\,\mathbf{H}$ {$= d\mathbf{H}/dt$; graph propagation $\mathcal{O}(L\,|E|\,d)$}

Algorithm 2 PIANO forward pass on one minibatch (predict $\hat{\mathbf{y}}$ from $\mathbf{x}$)

Require: input window $\mathbf{x}$ (task-specific shape), encoder $\mathcal{E}_\phi$, decoder $\mathcal{D}_\psi$, vector field f_θ (Alg. 1), integration times $\{0, T\}$, solver method / tolerances (`method`, `rtol`, `atol`)
Require: graph: sparse $\hat{A}$, precomputed $\{\mathrm{diag}(\hat{A}^k)\}_{k=0}^{L}$ (static adjacency, or kNN on the latent grid rebuilt when resolution changes)
$\mathbf{H}(0) \leftarrow \mathcal{E}_\phi(\mathbf{x})$ {shape $[B, N, d]$ after flattening spatial/latent nodes}
bind f_θ to $(\hat{A}, \{\mathrm{diag}(\hat{A}^k)\})$; reset NFE counter
if adjoint training **then**
  $\mathbf{H}(T) \leftarrow \mathrm{ODESolve}_{\mathrm{adj}}\big(f_\theta, \mathbf{H}(0), [0, T]\big)$
else
  $\mathbf{H}(T) \leftarrow \mathrm{ODESolve}\big(f_\theta, \mathbf{H}(0), [0, T]\big)$
end if {each internal step evaluates Alg. 1}
$\hat{\mathbf{y}} \leftarrow \mathcal{D}_\psi\big(\mathbf{H}(T)\big)$
return $\hat{\mathbf{y}}$

where the dependence of α_k on $(\mathbf{H}, t)$ is omitted for readability. This requires one running feature buffer and L SpMMs.

Let C_α denote the cost of computing $\boldsymbol{\alpha}(\mathbf{H}, t)$ for one sample. For one vector-field evaluation, the recursion performs L SpMMs and therefore costs $\mathcal{O}(L|E|d)$. The affine channel map costs $\mathcal{O}(Nd^2)$; pooling, norms, and gating contribute $\mathcal{O}(Nd)$; and forming the closed-walk score from stored diagonals costs $\mathcal{O}(LN)$. Thus the complete unbatched vector-field cost is

$$\mathcal{O}\big(L|E|d + Nd^2 + C_\alpha + LN + Nd\big)\,.$$

For batch size B and F vector-field evaluations chosen by the ODE solver, this becomes $\mathcal{O}\big(BF(L|E|d + Nd^2 + C_\alpha + LN + Nd)\big)$, excluding the task-specific encoder and decoder.

The gate stores $\{\mathrm{diag}(\hat{A}^k)\}_{k=0}^{L}$, requiring $\mathcal{O}(LN)$ memory per graph. Our exact chunked routine propagates columns of the identity through $\hat{A}$ and requires $\mathcal{O}(L|E|N)$ preprocessing work without forming dense powers. This cost is paid once for a static graph and once per graph when topology varies across samples.

B.3 Experimental configuration

Table 6 records the canonical PIANO setting used for each primary reported table. Values are taken from frozen seed-42 run configurations, with trainer defaults only when a JSON key is omitted. All reported checkpoints, including Ocean, are selected by validation performance. Graph-wave, Springs, Ocean, and SEVIR use MSE training loss; Cora, Citeseer, PubMed, and ogbn-arxiv use

Algorithm 3 Training PIANO by minimising the task-specific loss

Require: training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$, parameters (ϕ, θ, ψ), task loss ℓ_{task} (MSE / MAE / Huber), optional mass weight $\lambda_{\text{mass}} \geq 0$, optimiser hyperparameters, gradient clip $c > 0$, max epochs $E_{\max}$, early-stopping patience P

initialise (ϕ, θ, ψ); build optimiser and optional LR scheduler
precompute $\{\text{diag}(\hat{A}^k)\}_{k=0}^{L}$ for each fixed graph
$\mathcal{L}_{\text{best}} \leftarrow +\infty$; `wait` $\leftarrow 0$
for $e = 1$ **to** $E_{\max}$ **do**
  // — training epoch —
  for each minibatch $(\mathbf{x}, \mathbf{y})$ in $\mathcal{D}_{\text{train}}$ **do**
    $\hat{\mathbf{y}} \leftarrow \text{PIANOFORWARD}(\mathbf{x})$ {Alg. 2}
    $\mathcal{L}_{\text{batch}} \leftarrow \ell_{\text{task}}(\hat{\mathbf{y}}, \mathbf{y})$
    if $\lambda_{\text{mass}} > 0$ **then**
      $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \lambda_{\text{mass}}\, \mathbb{E}\big[\, |\text{Mass}(\hat{\mathbf{y}}) - \text{Mass}(\mathbf{y})| / (|\text{Mass}(\mathbf{y})| + \epsilon) \,\big]$
    end if
    zero gradients; back-propagate $\mathcal{L}_{\text{batch}}$ (adjoint through ODE if enabled)
    clip gradient norm of (ϕ, θ, ψ) to c
    optimiser step on (ϕ, θ, ψ)
  end for
  // — validation (no gradients) —
  $\bar{\mathcal{L}}_{\text{val}} \leftarrow$ average ℓ_{task} over $\mathcal{D}_{\text{val}}$
  log task metrics on $\mathcal{D}_{\text{val}}$ (e.g. CSI / SSIM for radar)
  if LR scheduler active **then**
    scheduler step (on $\bar{\mathcal{L}}_{\text{val}}$ or per epoch)
  end if
  if $\bar{\mathcal{L}}_{\text{val}} < \mathcal{L}_{\text{best}}$ **then**
    $\mathcal{L}_{\text{best}} \leftarrow \bar{\mathcal{L}}_{\text{val}}$; save checkpoint; `wait` $\leftarrow 0$
  else
    `wait` $\leftarrow$ `wait` $+ 1$
  end if
  if `wait` $\geq P$ **then**
    break
  end if
end for
return best checkpoint (ϕ, θ, ψ)

cross-entropy. All reported runs set $\lambda_{\text{mass}} = 0$ and use direct backpropagation through the solver rather than the adjoint method.

Table 6: PIANO canonical configuration for primary reported runs, from frozen seed-42 configs (trainer defaults only where a JSON key is omitted). Graph-wave-Scale uses cosine annealing ($T_{\max}$=40, $\eta_{\min}$=10^{-6}), weight decay 10^{-5}, and clip 5.0. Ocean uses the same cosine schedule and clip 5.0, with weight decay 10^{-5} from the trainer default, k=8 kNN, and `normalize_adj`. SEVIR uses cosine annealing, clip 1.0, weight decay 10^{-5}, and early stopping with patience 8. ogbn-arxiv additionally uses dropout 0.5, leak δ=0.1, and early-stopping patience 50. Small citation graphs use Geom-GCN splits, patience 50, clip 5.0, weight decay 5×10^{-4}, no scheduler, and the same compact d=64, L=3 encoder as ogbn-arxiv. Springs shares the Graph-wave-Scale AdamW recipe (learning rate 3×10^{-3}, cosine annealing, clip 5.0).

Benchmark	d	L	T	Solver/NFE	Optimiser / η	Batch	Epochs	Seeds
Graph-wave-Scale	64	8	1	RK4 (4)	AdamW / 3×10^{-3}	64	40	10
Springs	64	8	0.5	RK4 (4)	AdamW / 3×10^{-3}	64	40	10
Ocean ΔSLA	64	3	1	RK4 (4)	AdamW / 10^{-3}	16	40	10
SEVIR (window)	64	3	1	RK4 (4)	AdamW / 10^{-3}	32	30	10
ogbn-arxiv	64	3	1	RK4 (4)	Adam / 10^{-2}	full graph	500	10
Cora / Citeseer / PubMed	64	3	1	RK4 (4)	Adam / 10^{-2}	full graph	200	10 splits

B.4 Statistical conventions and metrics

Statistical conventions. Unless otherwise stated, uncertainty is the sample standard deviation across independent seeds. Comparisons with matched seeds or splits use paired tests; the exact test is reported with the corresponding experiment. Confidence intervals are paired across matched seeds where applicable. No claim of statistical superiority is made from displayed-mean ordering alone.

Evaluation metrics. For SEVIR, at threshold q, the critical success index is

$$\mathrm{CSI@}\, q = \frac{\mathrm{TP}_q}{\mathrm{TP}_q + \mathrm{FP}_q + \mathrm{FN}_q},$$

where thresholding is applied to the normalised VIL values. Whenever CSI is reported in this paper, it is computed from a single globally pooled TP/FP/FN over the official test set (Appendix F.3). MSE remains the principal SEVIR score.

For ocean forecasting, ACC is the Pearson correlation between predicted and target ΔSLA values over graph nodes for each sample and lead:

$$\mathrm{ACC} = \frac{\sum_i (\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}.$$

The reported value is averaged over test samples and the four forecast leads.

C Theoretical Analysis and Complete Proofs

This appendix collects the definitions and complete proofs supporting Section 4. The frozen-linear receptive-field bound is a sufficient finite-hop approximation radius for a frozen linear surrogate, not the receptive field of the full nonlinear state-adaptive ODE.

C.1 Mathematical Preliminaries

This appendix collects the matrix-analysis and ODE tools used in the proofs. Throughout, $\mathbf{H} \in \mathbb{R}^{N\times d}$ is the latent state and $\hat{A} \in \mathbb{R}^{N\times N}$ is the symmetrically normalised adjacency.

(P1) Two matrix norms. For a matrix X we use two sizes. The *Frobenius norm* $\|X\|_F = \big(\sum_{ij} X_{ij}^2\big)^{1/2}$ is just the Euclidean length of X read as a long vector. The *spectral (operator) norm* $\|X\|_2 = \max_{\mathbf{v}\neq 0} \|X\mathbf{v}\|_2/\|\mathbf{v}\|_2$ is the largest factor by which X can stretch any vector. For compatible matrices X, Y we use the sub-multiplicative inequalities

$$\|XY\|_F \le \min\big\{\|X\|_2\,\|Y\|_F,\ \|X\|_F\,\|Y\|_2\big\}, \qquad \|XY\|_2 \le \|X\|_2\,\|Y\|_2, \tag{12}$$

which say "the output is at most the stretch factor times the input." Both Frobenius bounds are needed: the first form controls $M_\theta\mathbf{H}$ (left-multiply by an operator), and the second controls the linear part of channel mixing.

We also repeatedly use the Frobenius Cauchy–Schwarz inequality

$$|\langle X, Y\rangle_F| \le \|X\|_F \|Y\|_F,$$

and the vector form $|\mathbf{x}^\top \mathbf{y}| \le \|\mathbf{x}\|_2\|\mathbf{y}\|_2$.

Affine channel map used in the implementation. The compact notation $\mathcal{W}$ in Eq. (8) denotes

$$\mathcal{W}(X) := XW + \mathbf{1}_N \mathbf{q}^\top, \qquad W \in \mathbb{R}^{d\times d}, \quad \mathbf{q} \in \mathbb{R}^d, \tag{13}$$

which is the implemented `Linear` layer with bias. If $\|W\|_2 \le B_W$ and $\|\mathbf{q}\|_2 \le B_q$, set $B_0 := \sqrt{N}\, B_q$. Since $\|\mathbf{1}_N \mathbf{q}^\top\|_F = \sqrt{N}\,\|\mathbf{q}\|_2$,

$$\begin{aligned} \|\mathcal{W}(X) - \mathcal{W}(Y)\|_F &\le B_W \|X - Y\|_F, \\ \|\mathcal{W}(X)\|_F &\le B_W \|X\|_F + B_0. \end{aligned} \tag{14}$$

Thus the bias cancels in state differences and affects only the additive growth constants.

(P2) Symmetric matrices and the spectral norm. A real symmetric matrix S admits an eigendecomposition $S = U\Lambda U^\top$, where U is orthogonal and Λ contains the real eigenvalues. Hence

$$S^k = U\Lambda^k U^\top, \qquad \|S\|_2 = \max_i |\lambda_i|.$$

In our setting, $A = A^\top$ is entrywise nonnegative, includes self-loops, and has positive degrees $D_{ii} = \sum_j A_{ij}$. Define $\hat{A} = D^{-1/2}AD^{-1/2}$. Then $\hat{A}$ is real symmetric and

$$\mathbf{x}^\top(I \pm \hat{A})\mathbf{x} = \frac{1}{2}\sum_{i,j} A_{ij}\left(\frac{x_i}{\sqrt{D_{ii}}} \pm \frac{x_j}{\sqrt{D_{jj}}}\right)^2 \geq 0. \tag{15}$$

Hence $-I \preceq \hat{A} \preceq I$, so every eigenvalue lies in $[-1, 1]$ and $\|\hat{A}\|_2 \leq 1$. Writing $P = D^{-1}A$, we have

$$\hat{A} = D^{1/2}PD^{-1/2}, \qquad \hat{A}^k = D^{1/2}P^kD^{-1/2},$$

which relates the symmetric and random-walk propagators (Chung, 1997; Barlow, 2017).

(P3) Lipschitz continuity. A map F is L-Lipschitz if $\|F(x) - F(y)\| \leq L\|x - y\|$. Lipschitz constants multiply under composition.

For softmax, with

$$\mathbf{s} = \mathrm{softmax}(\mathbf{z}), \qquad J(\mathbf{z}) = \mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top,$$

the Jacobian satisfies $\|J(\mathbf{z})\|_2 \leq 1/2$. Hence softmax is globally L_{sm}-Lipschitz in the Euclidean norm, where one may take

$$L_{\mathrm{sm}} \leq \frac{1}{2}. \tag{16}$$

We occasionally use the looser choice $L_{\mathrm{sm}} = 1$ for simpler constants.

For an MLP MLP_ω with weight matrices $W_1, \ldots, W_m$ and activation Lipschitz constants $L_1, \ldots, L_{m-1}$, let L_ω denote its Euclidean Lipschitz constant. One valid choice is

$$L_\omega \leq \left(\prod_{\ell=1}^{m} \|W_\ell\|_2\right)\left(\prod_{\ell=1}^{m-1} L_\ell\right). \tag{17}$$

The implemented GELU satisfies $\sup_z |\sigma'(z)| \leq C_{\mathrm{GELU}} \approx 1.13$. Hence Assumption **(A3)** holds with $L_\sigma = C_{\mathrm{GELU}}$ and $s_0 = 0$.

Node-wise mean pooling satisfies

$$\|\mathrm{pool}(\mathbf{H}_1) - \mathrm{pool}(\mathbf{H}_2)\|_2 \leq \frac{1}{\sqrt{N}}\|\mathbf{H}_1 - \mathbf{H}_2\|_F.$$

The Euclidean norm and sigmoid satisfy

$$|\|\mathbf{x}\|_2 - \|\mathbf{y}\|_2| \leq \|\mathbf{x} - \mathbf{y}\|_2, \qquad |\,\mathrm{sigmoid}(u) - \mathrm{sigmoid}(v)| \leq \frac{1}{4}|u - v|.$$

These facts are combined in Appendix C.3 to establish the regularity implied by Assumption **(A4)**.

(P4) Grönwall's inequality. This is the workhorse for bounding how fast a quantity can grow along a trajectory. If $u : [0, T] \to [0, \infty)$ is absolutely continuous and satisfies the differential inequality $\dot{u}(t) \leq \beta\, u(t) + c$ for almost every t, where $\beta > 0$ and $c \geq 0$, then

$$u(t) \;\leq\; \Big(u(0) + \tfrac{c}{\beta}\Big)e^{\beta t} - \tfrac{c}{\beta} \;\leq\; \Big(u(0) + \tfrac{c}{\beta}\Big)e^{\beta t}. \tag{18}$$

When $\beta = 0$ the bound simplifies to $u(t) \leq u(0) + ct$. Intuitively, the worst case is the matching equality $\dot{u} = \beta u + c$, whose solution is the right-hand side; the inequality version can only be smaller. (Proof: let $v(t) = u(t) + c/\beta$; then $\dot{v} \leq \beta v$, so $\frac{d}{dt}(e^{-\beta t}v) \leq 0$, hence $e^{-\beta t}v(t) \leq v(0)$, which rearranges to Eq. (18).)

(P5) Local existence, uniqueness, and continuation. Consider $\dot{\mathbf{H}} = f(\mathbf{H}, t)$ with $\mathbf{H}(0) = \mathbf{H}_0$. If f is continuous in t and locally Lipschitz in $\mathbf{H}$, uniformly on compact time intervals, then a unique maximal local solution exists. In finite dimensions, if the maximal solution remains bounded on every finite subinterval of $[0, T]$, it extends uniquely to all of $[0, T]$. Lemma 4 supplies the required bound, while Appendix C.5 verifies local Lipschitz continuity.

(P6) Linear propagators and matrix exponentials. For a linear ODE $\dot{\mathbf{h}}(t) = M\mathbf{h}(t)$, where $\mathbf{h}$ may represent a vectorised latent state, with a constant matrix M, the solution is $\mathbf{h}(T) = \exp(TM)\,\mathbf{h}(0)$, where the matrix exponential is the convergent series $\exp(TM) = \sum_{m=0}^{\infty} \frac{(TM)^m}{m!} = I + TM + \frac{T^2}{2}M^2 + \cdots$. We call $\Pi_T := \exp(TM)$ the time-T *propagator*: it maps the initial state to the state at time T. Truncating the series at degree κ (retaining $m = 0, \ldots, \kappa$) gives a polynomial in M of degree at most κ, and the remaining tail is small when κ is large enough (Appendix C.6).

(P7) Walks, powers, and graph distance. For an entrywise nonnegative adjacency-type matrix, $(\hat{A}^p)_{ij}$ is the total weight of length-p walks from i to j. Hence

$$p < d_G(i, j) \quad \Longrightarrow \quad (\hat{A}^p)_{ij} = 0,$$

where d_G is measured in the off-diagonal support graph. Self-loops do not alter this implication. Consequently, a polynomial in $\hat{A}$ of degree at most D can couple only pairs with $d_G(i, j) \le D$.

(P8) Closed walks on the diagonal. For an entrywise nonnegative adjacency-type matrix, $(\hat{A}^k)_{ii}$ is the total weight of length-k closed walks rooted at node i. Hence the sequence

$$\mathbf{d}_i^{(L)} = \big((\hat{A}^0)_{ii}, \ldots, (\hat{A}^L)_{ii}\big)^\top$$

provides an order-L structural signature of node i. For a simple unweighted loop-free adjacency A, $(A^3)_{ii} = 2\tau_i$, where τ_i is the number of triangles containing i. Because the model uses a self-loop-normalised $\hat{A}$, its diagonal powers also include self-loop and backtracking walks; triangle walks contribute additional mass beyond this baseline. The topology score $c_i = \boldsymbol{\alpha}^\top \mathbf{d}_i^{(L)}$ therefore forms a state-dependent projection of these closed-walk statistics, as used in Proposition 1.

C.2 Proof of Lemma 1 (Spectral Form and Pointwise Non-expansiveness)

Lemma 1 (Spectral form and pointwise non-expansiveness)**.** *Under (A1)–(A2), let $\hat{A} = U\Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ and $\lambda_i \in [-1, 1]$. For every realised $(\mathbf{H}, t)$, the operator in Equation* (4) *satisfies*

$$M_\theta(\mathbf{H}, t) = U p_{\boldsymbol{\alpha}}(\Lambda) U^\top, \qquad p_{\boldsymbol{\alpha}}(\lambda) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\lambda^k,$$

and $\|M_\theta(\mathbf{H}, t)\|_2 = \max_i |p_{\boldsymbol{\alpha}}(\lambda_i)| \le 1$. Consequently, for every $X \in \mathbb{R}^{N\times d}$, $\|M_\theta(\mathbf{H}, t)X\|_F \le \|X\|_F$. Thus, for each realised state and time, the simplex-constrained spatial operator is non-expansive in its input.

Proof. Under *(A1)*, the symmetric normalised adjacency admits the eigendecomposition

$$\hat{A} = U\Lambda U^\top, \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N), \qquad \lambda_i \in [-1, 1],$$

where U is orthogonal. Hence $\hat{A}^k = U\Lambda^k U^\top$ for every $k \ge 0$. Substituting into Equation (4) gives

$$\begin{aligned} M_\theta(\mathbf{H}, t) &= \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\hat{A}^k \\ &= U\left(\sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\Lambda^k\right) U^\top \\ &= U\, p_{\boldsymbol{\alpha}}(\Lambda)\, U^\top, \end{aligned}$$

where $p_{\boldsymbol{\alpha}}(\lambda) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\lambda^k$.

Thus $M_\theta(\mathbf{H}, t)$ is symmetric with eigenvalues $p_{\boldsymbol{\alpha}}(\lambda_i)$. By *(A2)*, $\alpha_k(\mathbf{H}, t) \geq 0$ and $\sum_{k=0}^{L} \alpha_k(\mathbf{H}, t) = 1$, so

$$\begin{aligned}\|M_\theta(\mathbf{H}, t)\|_2 &= \max_i |p_{\boldsymbol{\alpha}}(\lambda_i)| \\ &= \max_i \left| \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\lambda_i^k \right| \\ &\leq \max_i \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)|\lambda_i|^k \\ &\leq \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t) = 1.\end{aligned}$$

The first inequality is the triangle inequality, and the second uses $|\lambda_i| \leq 1$.

Finally, Equation (12) gives, for every $X \in \mathbb{R}^{N\times d}$,

$$\|M_\theta(\mathbf{H}, t)X\|_F \leq \|M_\theta(\mathbf{H}, t)\|_2 \|X\|_F \leq \|X\|_F.$$

Therefore, each realised spatial operator is pointwise non-expansive. □

Interpretation. For fixed $(\mathbf{H}, t)$, the coefficients can reweight short- and long-walk components without introducing spectral-norm amplification in the spatial mixing step. Stability of the full ODE additionally requires controlling how $M_\theta(\mathbf{H}, t)$ changes with the state, as established in Theorem 1.

C.3 Regularity of the Adaptive Operator and Gate

Lemma 2 (Lipschitz walk-length weights)**.** *Under* (A4)*, for every* $t \in [0, T]$*,*

$$\|\boldsymbol{\alpha}(\mathbf{H}_1, t) - \boldsymbol{\alpha}(\mathbf{H}_2, t)\|_2 \leq L_\alpha \|\mathbf{H}_1 - \mathbf{H}_2\|_F, \qquad L_\alpha = \frac{L_{\mathrm{sm}} L_\omega}{\sqrt{N}}, \tag{19}$$

where L_{sm} *denotes a Euclidean Lipschitz constant for softmax; one may take* $L_{\mathrm{sm}} = 1/2$*. The bound is uniform over* $t \in [0, T]$*. Using the looser choice* $L_{\mathrm{sm}} = 1$ *gives* $L_\alpha = L_\omega/\sqrt{N}$*.*

Proof. For node-wise mean pooling,

$$\mathrm{pool}(\mathbf{H}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i.$$

Hence, by Cauchy–Schwarz,

$$\begin{aligned}\|\mathrm{pool}(\mathbf{H}_1) - \mathrm{pool}(\mathbf{H}_2)\|_2 &\leq \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_{1,i} - \mathbf{h}_{2,i}\|_2 \\ &\leq \frac{1}{\sqrt{N}} \|\mathbf{H}_1 - \mathbf{H}_2\|_F.\end{aligned}$$

For fixed t, define the concatenated inputs and logits

$$\mathbf{z}_j(t) = \big[\mathrm{pool}(\mathbf{H}_j) \| \tau(t)\big], \qquad \boldsymbol{\ell}_j(t) = \mathrm{MLP}_\omega(\mathbf{z}_j(t)), \qquad j \in \{1, 2\}.$$

Because the same time encoding $\tau(t)$ is appended to both pooled states,

$$\|\mathbf{z}_1(t) - \mathbf{z}_2(t)\|_2 = \|\mathrm{pool}(\mathbf{H}_1) - \mathrm{pool}(\mathbf{H}_2)\|_2 .$$

Using Equation (5), $\boldsymbol{\alpha}(\mathbf{H}_j, t) = \mathrm{softmax}(\boldsymbol{\ell}_j(t))$, and therefore

$$\begin{aligned}\|\boldsymbol{\alpha}(\mathbf{H}_1, t) - \boldsymbol{\alpha}(\mathbf{H}_2, t)\|_2 &\leq L_{\mathrm{sm}} \|\boldsymbol{\ell}_1(t) - \boldsymbol{\ell}_2(t)\|_2 \quad \text{(Eq. (16))} \\ &\leq L_{\mathrm{sm}} L_\omega \|\mathbf{z}_1(t) - \mathbf{z}_2(t)\|_2 \quad \text{(Eq. (17))} \\ &= L_{\mathrm{sm}} L_\omega \|\mathrm{pool}(\mathbf{H}_1) - \mathrm{pool}(\mathbf{H}_2)\|_2 \\ &\leq \frac{L_{\mathrm{sm}} L_\omega}{\sqrt{N}} \|\mathbf{H}_1 - \mathbf{H}_2\|_F.\end{aligned}$$

The first two inequalities use the Lipschitz bounds of softmax and MLP_ω, respectively. Since the constants do not depend on t, the bound is uniform over $[0, T]$. □

The previous lemma controls the adaptive coefficients. The well-posedness analysis, however, requires a bound on how the resulting spatial operator $M_\theta(\mathbf{H}, t)$ itself changes with the state.

Corollary 1 (Lipschitz spatial operator)**.** *For every $t \in [0, T]$,*

$$\|M_\theta(\mathbf{H}_1, t) - M_\theta(\mathbf{H}_2, t)\|_2 \le \sqrt{L+1}\, L_\alpha \|\mathbf{H}_1 - \mathbf{H}_2\|_F. \tag{20}$$

Proof. Using Equation (4), $\|\hat{A}^k\|_2 \le 1$, and $\|\mathbf{v}\|_1 \le \sqrt{L+1}\|\mathbf{v}\|_2$,

$$\begin{aligned}
\|M_\theta(\mathbf{H}_1, t) - M_\theta(\mathbf{H}_2, t)\|_2 &= \left\| \sum_{k=0}^{L} \big(\alpha_k(\mathbf{H}_1, t) - \alpha_k(\mathbf{H}_2, t)\big) \hat{A}^k \right\|_2 \\
&\le \sum_{k=0}^{L} |\alpha_k(\mathbf{H}_1, t) - \alpha_k(\mathbf{H}_2, t)| \\
&= \|\boldsymbol{\alpha}(\mathbf{H}_1, t) - \boldsymbol{\alpha}(\mathbf{H}_2, t)\|_1 \\
&\le \sqrt{L+1}\, \|\boldsymbol{\alpha}(\mathbf{H}_1, t) - \boldsymbol{\alpha}(\mathbf{H}_2, t)\|_2 .
\end{aligned}$$

Applying Lemma 2 proves the result. □

The preceding results control the walk-length weights and the path-integral propagator they define. It remains to verify the regularity of the selective topology-aware node gate appearing in the full vector field.

Lemma 3 (Lipschitz selective topology-aware node gate)**.** *Under* (A1) *and* (A4)*, for every $t \in [0, T]$, the selective topology-aware node gate satisfies*

$$\|\boldsymbol{\gamma}(\mathbf{H}_1, t) - \boldsymbol{\gamma}(\mathbf{H}_2, t)\|_\infty \le L_{\gamma,\infty} \|\mathbf{H}_1 - \mathbf{H}_2\|_F, \tag{21}$$

where

$$L_{\gamma,\infty} = L_{\mathrm{sig}} \left(|a| \sqrt{L+1}\, L_\alpha + |b| \right), \qquad L_{\mathrm{sig}} = \frac{1}{4}.$$

The constant is independent of t. Consequently, $\mathbf{H} \mapsto \boldsymbol{\gamma}(\mathbf{H}, t)$ is globally Lipschitz. Moreover, for every fixed $\mathbf{H}$, $t \mapsto \boldsymbol{\gamma}(\mathbf{H}, t)$ is continuous.

Proof. For each node i, collect the diagonal walk masses into

$$\mathbf{d}_i = \big((\hat{A}^0)_{ii}, \dots, (\hat{A}^L)_{ii}\big)^\top.$$

Under *(A1)*, $\|\hat{A}\|_2 \le 1$. Hence, for every $k = 0, \dots, L$,

$$|(\hat{A}^k)_{ii}| = |\mathbf{e}_i^\top \hat{A}^k \mathbf{e}_i| \le \|\hat{A}^k\|_2 \le \|\hat{A}\|_2^k \le 1,$$

and therefore

$$\|\mathbf{d}_i\|_2 \le \sqrt{L+1}.$$

The topology score in Equation (9) is

$$\begin{aligned}
c_i(\mathbf{H}, t) &= \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t) (\hat{A}^k)_{ii} \\
&= \mathbf{d}_i^\top \boldsymbol{\alpha}(\mathbf{H}, t).
\end{aligned}$$

Thus, by Cauchy–Schwarz and Lemma 2,

$$\begin{aligned}
|c_i(\mathbf{H}_1, t) - c_i(\mathbf{H}_2, t)| &\le \|\mathbf{d}_i\|_2 \, \|\boldsymbol{\alpha}(\mathbf{H}_1, t) - \boldsymbol{\alpha}(\mathbf{H}_2, t)\|_2 \\
&\le \sqrt{L+1}\, L_\alpha \|\mathbf{H}_1 - \mathbf{H}_2\|_F.
\end{aligned}$$

The reverse triangle inequality gives

$$|\|\mathbf{h}_{1,i}\|_2 - \|\mathbf{h}_{2,i}\|_2| \le \|\mathbf{h}_{1,i} - \mathbf{h}_{2,i}\|_2 \le \|\mathbf{H}_1 - \mathbf{H}_2\|_F.$$

Recall that

$$\gamma_i(\mathbf{H}, t) = \mathrm{sigmoid}\left(a\, c_i(\mathbf{H}, t) + b\, \|\mathbf{h}_i\|_2 + b_0\right).$$

Since sigmoid is globally L_{sig}-Lipschitz with $L_{\mathrm{sig}} = 1/4$, and b_0 cancels in state differences,

$$\begin{aligned}|\gamma_i(\mathbf{H}_1, t) - \gamma_i(\mathbf{H}_2, t)| &\le L_{\mathrm{sig}}\Big(|a|\,|c_i(\mathbf{H}_1, t) - c_i(\mathbf{H}_2, t)| \\ &\qquad + |b|\,\big|\|\mathbf{h}_{1,i}\|_2 - \|\mathbf{h}_{2,i}\|_2\big|\Big) \\ &\le L_{\mathrm{sig}}\left(|a|\sqrt{L+1}\,L_\alpha + |b|\right)\|\mathbf{H}_1 - \mathbf{H}_2\|_F.\end{aligned}$$

Taking the maximum over i yields

$$\|\boldsymbol{\gamma}(\mathbf{H}_1, t) - \boldsymbol{\gamma}(\mathbf{H}_2, t)\|_\infty \le L_{\gamma,\infty}\|\mathbf{H}_1 - \mathbf{H}_2\|_F,$$

with

$$L_{\gamma,\infty} = L_{\mathrm{sig}}\left(|a|\sqrt{L+1}\,L_\alpha + |b|\right).$$

This constant is finite and independent of t.

Finally, under *(A4)*, $\tau(t)$ is continuous. For fixed $\mathbf{H}$, continuity of the MLP and softmax implies continuity of $t \mapsto \boldsymbol{\alpha}(\mathbf{H}, t)$, and therefore of $t \mapsto c_i(\mathbf{H}, t)$. Since the Euclidean norm and sigmoid are continuous, $t \mapsto \gamma_i(\mathbf{H}, t)$ is continuous for every node i. Hence $t \mapsto \boldsymbol{\gamma}(\mathbf{H}, t)$ is continuous. □

The walk-length weights and path-integral propagator are therefore globally Lipschitz in the state, while the selective topology-aware node gate admits the uniform ℓ_∞ bound in Equation (21). These estimates are precisely the ingredients used to establish local Lipschitz continuity of the full vector field in Appendix C.5.

C.4 A Priori Trajectory Bound

Lemma 4 (A priori trajectory bound)**.** *Under* (A1)–(A5)*, define*

$$\beta := L_\sigma B_W + \delta, \qquad c_0 := L_\sigma B_0 + s_0\sqrt{Nd},$$

and

$$\psi_\beta(t) := \begin{cases} \dfrac{e^{\beta t} - 1}{\beta}, & \beta > 0, \\ t, & \beta = 0. \end{cases}$$

Then any local solution of Equation (2) *defined on* $[0, t_*] \subseteq [0, T]$ *satisfies*

$$\|\mathbf{H}(t)\|_F \le e^{\beta t}\|\mathbf{H}(0)\|_F + c_0\psi_\beta(t) \le R_T, \tag{22}$$

for every $t \in [0, t_*]$*, where*

$$R_T := e^{\beta T}\|\mathbf{H}(0)\|_F + c_0\psi_\beta(T).$$

Proof. Step 1 (start from the full vector field). From Equation (8),

$$f_\theta(\mathbf{H}, t) = \boldsymbol{\gamma}(\mathbf{H}, t) \odot \sigma(\mathcal{W}(M_\theta(\mathbf{H}, t)\mathbf{H})) - \delta\mathbf{H},$$

where each scalar gate value γ_i is broadcast across the feature channels of node i. Since the gate is produced by a sigmoid,

$$0 \le \gamma_i(\mathbf{H}, t) \le 1.$$

Therefore, for every $\mathbf{X} \in \mathbb{R}^{N\times d}$,

$$\begin{aligned}\|\boldsymbol{\gamma} \odot \mathbf{X}\|_F^2 &= \sum_{i=1}^N \gamma_i^2\|\mathbf{x}_i\|_2^2 \\ &\le \sum_{i=1}^N \|\mathbf{x}_i\|_2^2 = \|\mathbf{X}\|_F^2.\end{aligned}$$

Thus the gate cannot increase the Frobenius norm. Applying this fact and the triangle inequality to the vector field gives

$$\|f_\theta(\mathbf{H},t)\|_F \le \|\sigma(\mathcal{W}(M_\theta(\mathbf{H},t)\mathbf{H}))\|_F + \delta\|\mathbf{H}\|_F. \tag{23}$$

Step 2 (bound the activation). By *(A3)*, the entrywise activation σ is L_σ-Lipschitz and satisfies $|\sigma(0)| = s_0$. Hence, for every scalar z,

$$|\sigma(z)| \le |\sigma(z) - \sigma(0)| + |\sigma(0)| \le L_\sigma|z| + s_0.$$

For any $\mathbf{Z} = (z_{ij}) \in \mathbb{R}^{N\times d}$, the Euclidean triangle inequality therefore gives

$$\begin{aligned}
\|\sigma(\mathbf{Z})\|_F &= \left(\sum_{i=1}^{N}\sum_{j=1}^{d}|\sigma(z_{ij})|^2\right)^{1/2} \\
&\le \left(\sum_{i=1}^{N}\sum_{j=1}^{d}\big(L_\sigma|z_{ij}| + s_0\big)^2\right)^{1/2} \\
&\le L_\sigma\left(\sum_{i=1}^{N}\sum_{j=1}^{d}|z_{ij}|^2\right)^{1/2} + s_0\left(\sum_{i=1}^{N}\sum_{j=1}^{d}1\right)^{1/2} \\
&= L_\sigma\|\mathbf{Z}\|_F + s_0\sqrt{Nd}.
\end{aligned} \tag{24}$$

For the implemented GELU, $\sigma(0) = 0$, so $s_0 = 0$. Applying Equation (24) with

$$\mathbf{Z} = \mathcal{W}(M_\theta(\mathbf{H},t)\mathbf{H})$$

yields

$$\|\sigma(\mathcal{W}(M_\theta(\mathbf{H},t)\mathbf{H}))\|_F \le L_\sigma\,\|\mathcal{W}(M_\theta(\mathbf{H},t)\mathbf{H})\|_F + s_0\sqrt{Nd}. \tag{25}$$

Step 3 (bound graph propagation and affine channel mixing). By the affine growth bound in *(A5)*, Lemma 1, and Equation (12),

$$\begin{aligned}
\|\mathcal{W}(M_\theta(\mathbf{H},t)\mathbf{H})\|_F &\le B_W\|M_\theta(\mathbf{H},t)\mathbf{H}\|_F + B_0 \\
&\le B_W\|M_\theta(\mathbf{H},t)\|_2\|\mathbf{H}\|_F + B_0 \\
&\le B_W\|\mathbf{H}\|_F + B_0.
\end{aligned} \tag{26}$$

Step 4 (assemble a linear-growth bound). Combining Equations (23), (25), and (26) gives

$$\begin{aligned}
\|f_\theta(\mathbf{H},t)\|_F &\le L_\sigma\big(B_W\|\mathbf{H}\|_F + B_0\big) + s_0\sqrt{Nd} + \delta\|\mathbf{H}\|_F \\
&= \big(L_\sigma B_W + \delta\big)\|\mathbf{H}\|_F + L_\sigma B_0 + s_0\sqrt{Nd}.
\end{aligned}$$

Thus, with

$$\beta = L_\sigma B_W + \delta, \qquad c_0 = L_\sigma B_0 + s_0\sqrt{Nd},$$

we obtain

$$\|f_\theta(\mathbf{H},t)\|_F \le \beta\|\mathbf{H}\|_F + c_0. \tag{27}$$

By *(A3)* and *(A5)*, all terms are nonnegative, so $\beta, c_0 \ge 0$.

Step 5 (regularise the state norm). The Frobenius norm is not differentiable at $\mathbf{H} = 0$. For $\varepsilon > 0$, define

$$u_\varepsilon(t) := \sqrt{\|\mathbf{H}(t)\|_F^2 + \varepsilon^2}. \tag{28}$$

Along the local solution, $\dot{\mathbf{H}}(t) = f_\theta(\mathbf{H}(t),t)$. Using

$$\frac{d}{dt}\|\mathbf{H}(t)\|_F^2 = 2\langle\mathbf{H}(t),\dot{\mathbf{H}}(t)\rangle_F,$$

the chain rule gives

$$\begin{aligned}
\dot{u}_\varepsilon(t) &= \frac{1}{2u_\varepsilon(t)}\frac{d}{dt}\|\mathbf{H}(t)\|_F^2 \\
&= \frac{\langle \mathbf{H}(t), \dot{\mathbf{H}}(t)\rangle_F}{u_\varepsilon(t)} \\
&\le \frac{\|\mathbf{H}(t)\|_F}{u_\varepsilon(t)}\|\dot{\mathbf{H}}(t)\|_F \qquad \text{(Cauchy–Schwarz)} \\
&\le \|f_\theta(\mathbf{H}(t), t)\|_F \qquad (\|\mathbf{H}(t)\|_F \le u_\varepsilon(t)) \\
&\le \beta\|\mathbf{H}(t)\|_F + c_0 \qquad \text{(Equation (27))} \\
&\le \beta u_\varepsilon(t) + c_0 \qquad (\beta \ge 0,\ \|\mathbf{H}(t)\|_F \le u_\varepsilon(t)).
\end{aligned}$$

Step 6 (apply Grönwall and remove the regularisation). We have shown that

$$\dot{u}_\varepsilon(t) \le \beta u_\varepsilon(t) + c_0.$$

When $\beta > 0$, Grönwall's inequality in *(P4)* gives

$$\begin{aligned}
u_\varepsilon(t) &\le e^{\beta t}u_\varepsilon(0) + \frac{c_0}{\beta}\big(e^{\beta t} - 1\big) \\
&= e^{\beta t}u_\varepsilon(0) + c_0\psi_\beta(t).
\end{aligned}$$

When $\beta = 0$, direct integration gives

$$u_\varepsilon(t) \le u_\varepsilon(0) + c_0 t = u_\varepsilon(0) + c_0\psi_0(t).$$

Thus, in both cases,

$$u_\varepsilon(t) \le e^{\beta t}u_\varepsilon(0) + c_0\psi_\beta(t).$$

Letting $\varepsilon \downarrow 0$ yields

$$\|\mathbf{H}(t)\|_F \le e^{\beta t}\|\mathbf{H}(0)\|_F + c_0\psi_\beta(t).$$

Since $t \le T$, $\beta \ge 0$, and both $t \mapsto e^{\beta t}$ and $t \mapsto \psi_\beta(t)$ are nondecreasing,

$$\|\mathbf{H}(t)\|_F \le e^{\beta T}\|\mathbf{H}(0)\|_F + c_0\psi_\beta(T) = R_T.$$

This holds for every $t \in [0, t_*]$. □

Interpretation. The estimate prevents finite-time escape of the latent state: every local solution remains inside the ball of radius R_T for as long as it exists within $[0, T]$. Combined with continuity in t and local Lipschitz continuity in $\mathbf{H}$, the continuation result in *(P5)* extends the unique local solution to the full interval $[0, T]$. Thus R_T provides an explicit finite-horizon bound used in the proof of Theorem 1.

C.5 Proof of Theorem 1 (Well-Posed Continuous Flow)

Theorem 1 (Well-posed continuous flow)**.** *Under (A1)–(A5), Equation* (2) *admits a unique solution on every finite interval* $[0, T]$ *from every initial state* $\mathbf{H}_0$*. Its time-*T *flow map*

$$\Phi_T : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}, \qquad \mathbf{H}_0 \mapsto \mathbf{H}(T; \mathbf{H}_0),$$

is a homeomorphism. Moreover, its restriction to every bounded initial-state set is bi-Lipschitz onto its image. Specifically, for

$$\mathcal{B}_{R_0} = \{\mathbf{H}_0 : \|\mathbf{H}_0\|_F \le R_0\},$$

there exists a finite $C_{T,R_0} \ge 1$ *such that, for all* $\mathbf{H}_{0,1}, \mathbf{H}_{0,2} \in \mathcal{B}_{R_0}$*,*

$$C_{T,R_0}^{-1}\|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F \le \|\Phi_T(\mathbf{H}_{0,1}) - \Phi_T(\mathbf{H}_{0,2})\|_F \le C_{T,R_0}\|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F.$$

The upper bound controls sensitivity to initial conditions, while the lower bound prevents distinct initial states from collapsing under the exact flow.

Explicit constants. On a state ball

$$\mathcal{B}_R := \{\mathbf{H} : \|\mathbf{H}\|_F \le R\},$$

define

$$L_\Psi(R) := 1 + \sqrt{L+1}\, L_\alpha R,$$
$$L_s(R) := L_\sigma B_W L_\Psi(R), \qquad S(R) := L_\sigma(B_W R + B_0) + s_0\sqrt{Nd},$$

and

$$L_{\gamma,\infty} := L_{\mathrm{sig}}\left(|a|\sqrt{L+1}\, L_\alpha + |b|\right), \qquad L_{\mathrm{sig}} = \frac{1}{4}.$$

Then an admissible Lipschitz constant for $f_\theta(\cdot, t)$ on $\mathcal{B}_R$, uniformly over $t \in [0, T]$, is

$$L_f(R) := L_s(R) + S(R)L_{\gamma,\infty} + \delta. \tag{29}$$

Proof. Fix $t \in [0, T]$ and $\mathbf{H}_1, \mathbf{H}_2 \in \mathcal{B}_R$, and write

$$M_j := M_\theta(\mathbf{H}_j, t), \qquad \Delta\mathbf{H} := \mathbf{H}_1 - \mathbf{H}_2.$$

Step 1 (state-adaptive propagation). Define

$$\Psi_t(\mathbf{H}) := M_\theta(\mathbf{H}, t)\mathbf{H}.$$

Adding and subtracting $M_1\mathbf{H}_2$ gives

$$\Psi_t(\mathbf{H}_1) - \Psi_t(\mathbf{H}_2) = M_1\Delta\mathbf{H} + (M_1 - M_2)\mathbf{H}_2.$$

By Lemma 1, $\|M_1\|_2 \le 1$, while Corollary 1 gives

$$\|M_1 - M_2\|_2 \le \sqrt{L+1}\, L_\alpha \|\Delta\mathbf{H}\|_F.$$

Since $\|\mathbf{H}_2\|_F \le R$,

$$\begin{aligned}\|\Psi_t(\mathbf{H}_1) - \Psi_t(\mathbf{H}_2)\|_F &\le \|M_1\Delta\mathbf{H}\|_F + \|(M_1 - M_2)\mathbf{H}_2\|_F \\ &\le \left(1 + \sqrt{L+1}\, L_\alpha R\right)\|\Delta\mathbf{H}\|_F \\ &= L_\Psi(R)\|\Delta\mathbf{H}\|_F.\end{aligned}$$

Step 2 (channel mixing and activation). Define

$$s_t(\mathbf{H}) := \sigma(\mathcal{W}(\Psi_t(\mathbf{H}))).$$

By *(A3)* and the affine Lipschitz bound in *(A5)*,

$$\begin{aligned}\|s_t(\mathbf{H}_1) - s_t(\mathbf{H}_2)\|_F &\le L_\sigma\|\mathcal{W}(\Psi_t(\mathbf{H}_1)) - \mathcal{W}(\Psi_t(\mathbf{H}_2))\|_F \\ &\le L_\sigma B_W\|\Psi_t(\mathbf{H}_1) - \Psi_t(\mathbf{H}_2)\|_F \\ &\le L_s(R)\|\Delta\mathbf{H}\|_F.\end{aligned}$$

Moreover, by Equation (24), *(A5)*, and Lemma 1,

$$\begin{aligned}\|s_t(\mathbf{H})\|_F &\le L_\sigma\,\|\mathcal{W}(M_\theta(\mathbf{H}, t)\mathbf{H})\|_F + s_0\sqrt{Nd} \\ &\le L_\sigma\big(B_W\|\mathbf{H}\|_F + B_0\big) + s_0\sqrt{Nd} \\ &\le S(R).\end{aligned}$$

Step 3 (selective topology-aware node gate). By Lemma 3,

$$\|\boldsymbol{\gamma}(\mathbf{H}_1, t) - \boldsymbol{\gamma}(\mathbf{H}_2, t)\|_\infty \le L_{\gamma,\infty}\|\Delta\mathbf{H}\|_F,$$

uniformly over $t \in [0, T]$.

Step 4 (gated product and leak). Using

$$f_\theta(\mathbf{H}, t) = \boldsymbol{\gamma}(\mathbf{H}, t) \odot s_t(\mathbf{H}) - \delta\mathbf{H},$$

write

$$\boldsymbol{\gamma}_1 \odot s_1 - \boldsymbol{\gamma}_2 \odot s_2 = \boldsymbol{\gamma}_1 \odot (s_1 - s_2) + (\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2) \odot s_2.$$

Since $0 \le \gamma_{1,i} \le 1$,

$$\begin{aligned}\|\boldsymbol{\gamma}_1 \odot s_1 - \boldsymbol{\gamma}_2 \odot s_2\|_F &\le \|s_1 - s_2\|_F + \|\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2\|_\infty \|s_2\|_F \\ &\le (L_s(R) + S(R)L_{\gamma,\infty}) \, \|\Delta\mathbf{H}\|_F.\end{aligned}$$

Therefore,

$$\begin{aligned}\|f_\theta(\mathbf{H}_1, t) - f_\theta(\mathbf{H}_2, t)\|_F &\le (L_s(R) + S(R)L_{\gamma,\infty} + \delta) \, \|\Delta\mathbf{H}\|_F \\ &= L_f(R)\|\Delta\mathbf{H}\|_F.\end{aligned}$$

Thus $f_\theta(\cdot, t)$ is Lipschitz on every bounded state ball, with a constant uniform over $t \in [0, T]$.

Step 5 (continuity in time). Under *(A4)*, $\tau(t)$ is continuous, while the MLP, softmax, matrix products, activation, node norms, and sigmoid are continuous. Hence

$$(\mathbf{H}, t) \mapsto \boldsymbol{\alpha}(\mathbf{H}, t), \quad M_\theta(\mathbf{H}, t), \quad \boldsymbol{\gamma}(\mathbf{H}, t), \quad f_\theta(\mathbf{H}, t)$$

are continuous. In particular, $t \mapsto f_\theta(\mathbf{H}, t)$ is continuous for every fixed $\mathbf{H}$.

Step 6 (existence and uniqueness on $[0, T]$). Steps 1–5 show that f_θ is continuous in t and locally Lipschitz in $\mathbf{H}$, uniformly on compact time intervals. Therefore, by *(P5)*, a unique maximal local solution exists from every initial state.

Lemma 4 gives

$$\|\mathbf{H}(t)\|_F \le e^{\beta t}\|\mathbf{H}_0\|_F + c_0\psi_\beta(t)$$

for as long as the solution exists in $[0, T]$. Hence the solution cannot escape to infinity in finite time. The continuation result in *(P5)* therefore extends the unique solution to all of $[0, T]$.

Step 7 (invertibility by reversed dynamics). Fix a terminal state $\mathbf{Y}$ and define the reversed-time vector field

$$\widetilde{f}(\mathbf{X}, s) := -f_\theta(\mathbf{X}, T - s), \qquad s \in [0, T].$$

It has the same continuity, local-Lipschitz, and linear-growth properties as f_θ. Hence the reversed IVP

$$\dot{\mathbf{X}}(s) = \widetilde{f}(\mathbf{X}(s), s), \qquad \mathbf{X}(0) = \mathbf{Y},$$

has a unique solution on $[0, T]$.

Setting

$$\mathbf{H}(t) = \mathbf{X}(T - t)$$

gives the unique forward solution satisfying $\mathbf{H}(T) = \mathbf{Y}$. Thus every terminal state has exactly one preimage, and Φ_T is bijective.

Step 8 (bi-Lipschitz control on bounded initial sets). Fix $R_0 < \infty$. Lemma 4 gives the common trajectory radius

$$R_{T,R_0} := e^{\beta T}R_0 + c_0\psi_\beta(T)$$

for every trajectory starting in $\mathcal{B}_{R_0}$. Define

$$L_{f,R_0} := L_f(R_{T,R_0}), \qquad C_{T,R_0} := e^{L_{f,R_0}T}.$$

For two trajectories $\mathbf{H}_j(t) = \mathbf{H}(t; \mathbf{H}_{0,j})$ with $\mathbf{H}_{0,j} \in \mathcal{B}_{R_0}$, write $\Delta\mathbf{H}(t) = \mathbf{H}_1(t) - \mathbf{H}_2(t)$. Both trajectories remain in the common ball of radius R_{T,R_0}, so the Lipschitz estimate of Step 4 applies with constant L_{f,R_0}. The standard continuous-dependence / Grönwall argument then yields

$$\|\Delta\mathbf{H}(T)\|_F \le e^{L_{f,R_0}T}\|\Delta\mathbf{H}(0)\|_F.$$

Running the same trajectories backward via Step 7 traverses the same common region from terminal separation $\|\Delta\mathbf{H}(T)\|_F$ to initial separation $\|\Delta\mathbf{H}(0)\|_F$, and the same estimate gives

$$\|\Delta\mathbf{H}(0)\|_F \le e^{L_{f,R_0}T}\|\Delta\mathbf{H}(T)\|_F.$$

Therefore,

$$C_{T,R_0}^{-1}\|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F \le \|\Phi_T(\mathbf{H}_{0,1}) - \Phi_T(\mathbf{H}_{0,2})\|_F \le C_{T,R_0}\|\mathbf{H}_{0,1} - \mathbf{H}_{0,2}\|_F.$$

Hence the restriction of Φ_T to every bounded initial-state set is bi-Lipschitz onto its image.

Step 9 (continuity of the inverse). Fix $Q_0 < \infty$. The reversed linear-growth bound implies that every reversed trajectory starting from $\|\mathbf{Y}\|_F \le Q_0$ remains in the common ball

$$R^{\leftarrow}_{T,Q_0} = e^{\beta T} Q_0 + c_0 \psi_\beta(T).$$

Hence the reversed vector field has a finite Lipschitz constant on this ball, say $L^{\leftarrow}_{f,Q_0}$. Applying the preceding continuous-dependence estimate to two reversed trajectories yields

$$\|\Phi_T^{-1}(\mathbf{Y}_1) - \Phi_T^{-1}(\mathbf{Y}_2)\|_F \le e^{L^{\leftarrow}_{f,Q_0} T} \|\mathbf{Y}_1 - \mathbf{Y}_2\|_F.$$

Thus Φ_T^{-1} is continuous on $\mathbb{R}^{N\times d}$. Since Φ_T is also continuous and bijective, Φ_T is a homeomorphism. □

Why this matters. The theorem shows that the state adaptivity introduced by PIANO does not sacrifice the basic stability of the latent dynamics. Although both the walk-length weights $\boldsymbol{\alpha}(\mathbf{H}, t)$ and the selective topology-aware node gate $\boldsymbol{\gamma}(\mathbf{H}, t)$ change with the evolving state, the resulting vector field remains locally Lipschitz and has controlled growth. Hence every initial latent state generates a unique trajectory through the full integration horizon.

The exact continuous flow is also invertible: evolving backward from a terminal latent state recovers a unique initial state. Moreover, on every bounded set of initial conditions,

$$C_{T,R_0}^{-1} \|\Delta\mathbf{H}(0)\|_F \le \|\Delta\mathbf{H}(T)\|_F \le C_{T,R_0} \|\Delta\mathbf{H}(0)\|_F,$$

so nearby latent states cannot undergo uncontrolled expansion or collapse under the exact flow. This provides a bounded-set stability guarantee for the state-adaptive dynamics, rather than merely for each frozen spatial operator.

Importantly, the maximum explicit order L changes how far each realised operator can propagate information, while $\|M_\theta(\mathbf{H}, t)\|_2 \le 1$ remains independent of L. Thus increasing the available walk length does not by itself introduce spectral-norm amplification in the spatial mixing step. The explicit constants in the proof quantify how maximum explicit order, integration horizon, and architectural Lipschitz constants enter the overall flow sensitivity. These bounds are conservative sufficient guarantees for the exact latent ODE, rather than predictions of numerical solver or gradient behaviour.

Having established stability of the state-adaptive flow, we next study its non-local reach. To isolate the roles of maximum explicit order L and integration horizon T, Theorem 2 considers a frozen linear surrogate and derives a sufficient finite-hop approximation bound for its exact propagator.

C.6 Proof of Theorem 2 (Frozen Linear Receptive-Field Bound)

Recall the frozen linear surrogate

$$\dot{\mathbf{h}}(t) = M_\theta \mathbf{h}(t), \qquad M_\theta = \sum_{k=0}^{L} \alpha_k \hat{A}^k, \qquad \boldsymbol{\alpha} \in \Delta^L,$$

where M_θ is fixed along the trajectory. Its exact propagator is

$$\Pi_T := \exp(T M_\theta), \qquad \mathbf{h}(T) = \Pi_T \mathbf{h}(0).$$

Here L is the maximum explicit order, $T \ge 0$ is the integration horizon, and $d_G(i, j)$ is the graph distance defined in *(P7)*. For clarity, the analysis is written for a single feature channel; the same operator bound applies column-wise to matrix-valued states.

Theorem 2 (Receptive-field bound for frozen linear propagation)**.** *Under (A1)–(A2), let* $\Pi_{T,\kappa} := \sum_{m=0}^{\kappa} (T M_\theta)^m / m!$. *Then* $\Pi_{T,\kappa}$ *is a polynomial in* $\hat{A}$ *of degree at most* κL, *so* $[\Pi_{T,\kappa}]_{ij} = 0$ *whenever* $d_G(i, j) > \kappa L$, *and*

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le e^T \frac{T^{\kappa+1}}{(\kappa+1)!}.$$

Hence, for every $\varepsilon \in (0,1)$, there exists $\kappa(\varepsilon) = \mathcal{O}(T + \log(1/\varepsilon))$ such that $\|\Pi_T - \Pi_{T,\kappa(\varepsilon)}\|_2 \le \varepsilon$. A sufficient radius bound is therefore $r(\varepsilon) := L\kappa(\varepsilon) = \mathcal{O}(L(T + \log(1/\varepsilon)))$. This is a sufficient upper bound for the frozen linear surrogate, not a necessary radius or an exact equivalence between L and T.

Proof. Step 1 (polynomial degree of M_θ^m). For $m = 0$,

$$M_\theta^0 = I = \hat{A}^0.$$

Set $c_{0,0} := 1$. For $m \ge 1$,

$$\begin{aligned} M_\theta^m &= \left(\sum_{k=0}^{L} \alpha_k \hat{A}^k \right)^m \\ &= \sum_{k_1=0}^{L} \cdots \sum_{k_m=0}^{L} \left(\prod_{\ell=1}^{m} \alpha_{k_\ell} \right) \hat{A}^{k_1 + \cdots + k_m}. \end{aligned}$$

Since each $k_\ell \in \{0, \dots, L\}$, every exponent $p = k_1 + \cdots + k_m$ satisfies $0 \le p \le mL$. Grouping terms with the same total exponent gives

$$M_\theta^m = \sum_{p=0}^{mL} c_{m,p} \hat{A}^p, \qquad c_{m,p} := \sum_{\substack{k_1, \dots, k_m \in \{0,\dots,L\} \\ k_1 + \cdots + k_m = p}} \prod_{\ell=1}^{m} \alpha_{k_\ell}, \tag{30}$$

for $m \ge 1$, together with $c_{0,0} = 1$. We use the convention $c_{m,p} = 0$ whenever $p > mL$. Hence M_θ^m is a polynomial in $\hat{A}$ of degree at most mL for every $m \ge 0$.

Step 2 (support of the truncated propagator). The exact propagator has the matrix-exponential expansion

$$\Pi_T = e^{TM_\theta} = \sum_{m=0}^{\infty} \frac{T^m}{m!} M_\theta^m. \tag{31}$$

Its order-κ Taylor truncation is therefore

$$\Pi_{T,\kappa} := \sum_{m=0}^{\kappa} \frac{T^m}{m!} M_\theta^m.$$

By Step 1, each M_θ^m is a polynomial in $\hat{A}$ of degree at most mL. Substituting Equation (30) gives

$$\begin{aligned} \Pi_{T,\kappa} &= \sum_{m=0}^{\kappa} \frac{T^m}{m!} \left(\sum_{p=0}^{mL} c_{m,p} \hat{A}^p \right) \\ &= \sum_{p=0}^{\kappa L} \left(\sum_{m=0}^{\kappa} \frac{T^m}{m!} c_{m,p} \right) \hat{A}^p \\ &= \sum_{p=0}^{\kappa L} b_p \hat{A}^p, \end{aligned}$$

where

$$b_p := \sum_{m=0}^{\kappa} \frac{T^m}{m!} c_{m,p}.$$

Thus $\Pi_{T,\kappa}$ is a polynomial in $\hat{A}$ of degree at most κL.

By the walk-support property *(P7)*, $[\hat{A}^p]_{ij} = 0$ whenever $p < d_G(i,j)$. Hence, if $d_G(i,j) > \kappa L$, then for every $p \in \{0, \dots, \kappa L\}$,

$$p < d_G(i,j),$$

and therefore $[\hat{A}^p]_{ij} = 0$. Consequently,

$$[\Pi_{T,\kappa}]_{ij} = 0 \qquad \text{whenever } d_G(i,j) > \kappa L. \tag{32}$$

Hence the truncated propagator has support radius at most κL.

Step 3 (exponential-tail error). Subtracting the order-κ truncation from Equation (31) gives

$$\Pi_T - \Pi_{T,\kappa} = \sum_{m=\kappa+1}^{\infty} \frac{T^m}{m!} M_\theta^m.$$

By Lemma 1, $\|M_\theta\|_2 \le 1$, and therefore $\|M_\theta^m\|_2 \le \|M_\theta\|_2^m \le 1$. Hence

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le \sum_{m=\kappa+1}^{\infty} \frac{T^m}{m!}. \tag{33}$$

Re-indexing $m = \kappa + 1 + j$,

$$\begin{aligned}\sum_{m=\kappa+1}^{\infty} \frac{T^m}{m!} &= \frac{T^{\kappa+1}}{(\kappa+1)!} \sum_{j=0}^{\infty} T^j \frac{(\kappa+1)!}{(\kappa+1+j)!} \\ &\le \frac{T^{\kappa+1}}{(\kappa+1)!} \sum_{j=0}^{\infty} \frac{T^j}{j!} \\ &= e^T \frac{T^{\kappa+1}}{(\kappa+1)!},\end{aligned}$$

where

$$\frac{(\kappa+1)!}{(\kappa+1+j)!} = \frac{1}{(\kappa+2)\cdots(\kappa+1+j)} \le \frac{1}{j!}.$$

Combining this with Equation (33) proves

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le e^T \frac{T^{\kappa+1}}{(\kappa+1)!}. \tag{34}$$

Consequently, for every initial state,

$$\|\Pi_T \mathbf{h}(0) - \Pi_{T,\kappa} \mathbf{h}(0)\|_2 \le e^T \frac{T^{\kappa+1}}{(\kappa+1)!} \|\mathbf{h}(0)\|_2. \tag{35}$$

Step 4 (truncation order for error ε). Fix $\varepsilon \in (0,1)$ and set $n = \kappa + 1$. Since

$$\log(n!) = \sum_{j=1}^{n} \log j \ge \int_1^n \log x \, dx = n \log n - n + 1,$$

we have

$$n! \ge \left(\frac{n}{e}\right)^n.$$

Therefore Equation (34) gives

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le e^T \left(\frac{eT}{n}\right)^n. \tag{36}$$

If $n \ge 2eT$, then $eT/n \le 1/2$, so

$$\|\Pi_T - \Pi_{T,\kappa}\|_2 \le e^T 2^{-n}.$$

It is therefore sufficient to require

$$n \ge 2eT, \qquad n \ge \frac{T + \log(1/\varepsilon)}{\log 2}.$$

Choose

$$\kappa(\varepsilon) := \left\lceil \max\left\{ 2eT,\ \frac{T + \log(1/\varepsilon)}{\log 2} \right\} \right\rceil - 1. \tag{37}$$

Then

$$\|\Pi_T - \Pi_{T,\kappa(\varepsilon)}\|_2 \le \varepsilon.$$

Moreover,

$$\begin{aligned}\kappa(\varepsilon) &\le \max\left\{2eT,\ \frac{T+\log(1/\varepsilon)}{\log 2}\right\}\\ &\le 2eT + \frac{T+\log(1/\varepsilon)}{\log 2},\end{aligned}$$

and hence

$$\kappa(\varepsilon) = \mathcal{O}\left(T + \log\frac{1}{\varepsilon}\right). \tag{38}$$

The operator bound immediately implies the corresponding output bound

$$\|\Pi_T \mathbf{h}(0) - \Pi_{T,\kappa(\varepsilon)}\mathbf{h}(0)\|_2 \le \varepsilon\|\mathbf{h}(0)\|_2.$$

Step 5 (sufficient finite-hop radius). By Equation (32), $\Pi_{T,\kappa(\varepsilon)}$ has support radius at most $L\kappa(\varepsilon)$. A sufficient radius bound is therefore

$$r(\varepsilon) := L\kappa(\varepsilon) = \mathcal{O}\left(L\left(T + \log\frac{1}{\varepsilon}\right)\right). \tag{39}$$

This quantity is an upper bound on the support radius of the truncated propagator, not a claim that it equals the minimal or effective receptive field.

Thus L bounds the spatial degree contributed per operator application, while T controls how many Taylor composition depths are sufficient for a prescribed approximation error. Cancellations, vanishing coefficients, or special graph structure may yield a smaller effective support.

The theorem applies only to the frozen linear surrogate. It does not claim that $r(\varepsilon)$ is the receptive field of the full state-adaptive nonlinear ODE. Appendix D.6 instead examines empirically whether varying L and T produces the corresponding qualitative interaction in the trained model. □

C.7 Proof of Proposition 1 (Topology-Conditioned Field Separation)

For two node configurations, possibly associated with different graph operators $\hat{A}$ and $\hat{A}'$, define

$$\mathbf{d}_i^{(L)} := \left((\hat{A}^0)_{ii}, \dots, (\hat{A}^L)_{ii}\right)^\top, \qquad \mathbf{d}_j'^{(L)} := \left((\hat{A}'^0)_{jj}, \dots, (\hat{A}'^L)_{jj}\right)^\top.$$

For a common walk-length distribution $\boldsymbol{\alpha}$, their topology scores are

$$c_i = \boldsymbol{\alpha}^\top \mathbf{d}_i^{(L)}, \qquad c_j' = \boldsymbol{\alpha}^\top \mathbf{d}_j'^{(L)}.$$

Proposition 1 (Topology-conditioned field separation)**.** *Consider two node configurations with different closed-walk signatures, $\mathbf{d}_i^{(L)} \neq \mathbf{d}_j'^{(L)}$. Then there exists an admissible common walk-length distribution $\boldsymbol{\alpha} \in \operatorname{int}\Delta^L$ for which their topology scores differ. Moreover, if the two configurations have the same local state and the same nonzero transported activation, then a gate with $a \neq 0$ yields different vector-field updates. Hence the selective topology-aware node gate can distinguish structural differences even when the transported message itself is unchanged.*

Proof. Step 1 (different signatures induce different topology scores). Let

$$\mathbf{v} := \mathbf{d}_i^{(L)} - \mathbf{d}_j'^{(L)}.$$

By assumption, $\mathbf{v} \neq \mathbf{0}$. Since

$$(\hat{A}^0)_{ii} = (\hat{A}'^0)_{jj} = 1,$$

we have $v_0 = 0$. Hence, whenever the premise is non-vacuous, there exists some $q \in \{1, \dots, L\}$ such that $v_q \neq 0$.

For $\eta > 0$, define

$$\alpha_q^{(\eta)} := 1 - L\eta, \qquad \alpha_k^{(\eta)} := \eta \quad (k \neq q).$$

For $0 < \eta < 1/L$, $\boldsymbol{\alpha}^{(\eta)} \in \operatorname{int} \Delta^L$, and

$$\lim_{\eta \downarrow 0} (\boldsymbol{\alpha}^{(\eta)})^\top \mathbf{v} = v_q \neq 0.$$

Hence, for sufficiently small $\eta > 0$,

$$(\boldsymbol{\alpha}^{(\eta)})^\top \mathbf{v} \neq 0.$$

Therefore there exists a common admissible $\boldsymbol{\alpha} \in \operatorname{int} \Delta^L$ such that

$$\boldsymbol{\alpha}^\top \mathbf{d}_i^{(L)} \neq \boldsymbol{\alpha}^\top \mathbf{d}_j'^{(L)},$$

or equivalently,

$$c_i \neq c_j'.$$

Any such interior-simplex vector can be realised by finite softmax logits, so this choice is compatible with Equation (5).

Step 2 (different topology scores produce different gates). Suppose the two configurations have the same local state,

$$\mathbf{h}_i = \mathbf{h}_j'.$$

Their node-feature norms therefore agree. Since

$$\gamma_i = \operatorname{sigmoid}\left(a\, c_i + b\|\mathbf{h}_i\|_2 + b_0\right),$$

and analogously for γ_j', the conditions $a \neq 0$, $c_i \neq c_j'$, and strict monotonicity of sigmoid imply

$$\gamma_i \neq \gamma_j'.$$

Step 3 (different gates separate the vector-field updates). Let the transported activations of the two configurations be

$$\mathbf{s}_i := \left[\sigma(\mathcal{W}(M_\theta(\mathbf{H}, t)\mathbf{H}))\right]_i,$$

and

$$\mathbf{s}_j' := \left[\sigma(\mathcal{W}(M_\theta'(\mathbf{H}', t)\mathbf{H}'))\right]_j.$$

Suppose

$$\mathbf{s}_i = \mathbf{s}_j' =: \mathbf{s} \neq \mathbf{0}.$$

Since the local states are also identical, their leak terms coincide. Consequently,

$$\begin{aligned} f_{\theta,i} - f_{\theta,j}' &= \gamma_i \mathbf{s} - \gamma_j' \mathbf{s} \\ &= (\gamma_i - \gamma_j')\mathbf{s} \neq \mathbf{0}. \end{aligned}$$

Therefore

$$f_{\theta,i} \neq f_{\theta,j}',$$

which proves that the selective topology-aware node gate can separate structural differences even when the local state and transported activation are held fixed. □

C.8 A 1-WL-Indistinguishable Canonical Witness

Corollary 2 (Separation beyond 1-WL)**.** *Let C_6 denote the six-cycle and $2K_3$ the disjoint union of two triangles. With constant initial node features, the two graphs are indistinguishable by 1-WL. After adding one unit self-loop to every node and applying symmetric normalisation, however, their order-3 closed-walk signatures differ. In particular, for every node i,*

$$(\hat{A}_{C_6}^3)_{ii} = \frac{7}{27}, \qquad (\hat{A}_{2K_3}^3)_{ii} = \frac{1}{3} = \frac{9}{27}.$$

Hence Proposition 1 implies that the selective topology-aware node gate can separate this canonical 1-WL-indistinguishable pair for suitable parameters.

Proof. Step 1 (1-WL equivalence). Initialise every node with the same colour. Before adding self-loops, every node in both C_6 and $2K_3$ has exactly two neighbours. Hence every node receives the same multiset of neighbour colours at the first 1-WL iteration and is assigned the same updated colour. Repeating the argument inductively, all nodes in both graphs retain identical 1-WL colours at every iteration. Since both graphs contain six nodes, their colour histograms are identical, and 1-WL cannot distinguish them from constant initial features (Xu et al., 2019). Adding the same unit self-loop to every node preserves this conclusion.

Step 2 (normalised length-3 closed-walk mass of C_6). Let A_{C_6} denote the adjacency matrix of the six-cycle before self-loop augmentation. After adding one unit self-loop to every node, the graph is 3-regular, so

$$\hat{A}_{C_6} = \frac{I_6 + A_{C_6}}{3}.$$

Therefore,

$$\hat{A}^3_{C_6} = \frac{I_6 + 3A_{C_6} + 3A^2_{C_6} + A^3_{C_6}}{27}.$$

For every node i,

$$(A_{C_6})_{ii} = 0.$$

There are exactly two length-2 walks from i back to itself, one through each neighbour, so

$$(A^2_{C_6})_{ii} = 2.$$

Since C_6 contains no triangles, there is no length-3 closed walk from i, and hence

$$(A^3_{C_6})_{ii} = 0.$$

Consequently,

$$(\hat{A}^3_{C_6})_{ii} = \frac{1 + 3 \cdot 2}{27} = \frac{7}{27}.$$

Step 3 (normalised length-3 closed-walk mass of $2K_3$). Consider one triangle component K_3. After adding one unit self-loop to every node, its augmented adjacency is the all-ones matrix J_3, and every augmented degree equals 3. Hence

$$\hat{A}_{K_3} = \frac{J_3}{3}.$$

Since

$$J_3^2 = 3J_3,$$

we have

$$\hat{A}^2_{K_3} = \left(\frac{J_3}{3}\right)^2 = \frac{J_3}{3} = \hat{A}_{K_3}.$$

Thus $\hat{A}_{K_3}$ is idempotent and

$$\hat{A}^k_{K_3} = \hat{A}_{K_3}, \qquad k \geq 1.$$

Because $2K_3$ is the disjoint union of two such components, $\hat{A}_{2K_3}$ is block diagonal with two $J_3/3$ blocks. Therefore,

$$\hat{A}^3_{2K_3} = \hat{A}_{2K_3},$$

and every diagonal entry satisfies

$$(\hat{A}^3_{2K_3})_{ii} = \frac{1}{3} = \frac{9}{27}.$$

Step 4 (the closed-walk signatures differ). For completeness, the two graphs have identical diagonal path statistics at orders $k = 0, 1, 2$. For every node in either graph,

$$(\hat{A}^0)_{ii} = 1, \qquad (\hat{A})_{ii} = \frac{1}{3}, \qquad (\hat{A}^2)_{ii} = \frac{1}{3}.$$

Their first differing diagonal statistic therefore occurs at order $k = 3$:

$$(\hat{A}^3_{C_6})_{ii} = \frac{7}{27}, \qquad (\hat{A}^3_{2K_3})_{ii} = \frac{9}{27}.$$

Hence

$$\mathbf{d}_{C_6}^{(3)} = \left(1, \frac{1}{3}, \frac{1}{3}, \frac{7}{27}\right)^\top \neq \left(1, \frac{1}{3}, \frac{1}{3}, \frac{9}{27}\right)^\top = \mathbf{d}_{2K_3}^{(3)}.$$

Step 5 (conditions of Proposition 1). Let

$$\mathbf{H}_0 = \mathbf{1}_6 \mathbf{h}_0^\top$$

be the same constant node state on both graphs. Thus corresponding nodes have identical local states.

After self-loop augmentation, both graphs are 3-regular, and hence

$$\hat{A}_{\mathcal{G}} \mathbf{1}_6 = \mathbf{1}_6, \qquad \mathcal{G} \in \{C_6, 2K_3\}.$$

Therefore

$$\hat{A}_{\mathcal{G}}^k \mathbf{H}_0 = \mathbf{H}_0, \qquad k \geq 0.$$

For any common $\boldsymbol{\alpha} \in \Delta^3$,

$$\begin{aligned} M_\theta^{\mathcal{G}} \mathbf{H}_0 &= \sum_{k=0}^{3} \alpha_k \hat{A}_{\mathcal{G}}^k \mathbf{H}_0 \\ &= \left(\sum_{k=0}^{3} \alpha_k \right) \mathbf{H}_0 \\ &= \mathbf{H}_0. \end{aligned}$$

Hence both graphs produce the same transported activation

$$\mathbf{U}_0 := \sigma(\mathcal{W}(\mathbf{H}_0)).$$

Choose $\mathbf{h}_0$ and the affine channel map such that $\mathbf{U}_0 \neq \mathbf{0}$.

Since

$$\mathbf{d}_{C_6}^{(3)} \neq \mathbf{d}_{2K_3}^{(3)},$$

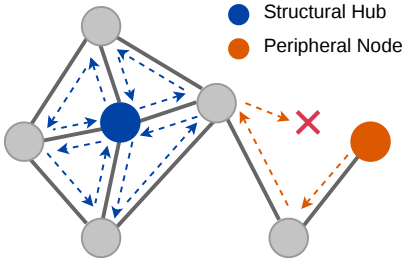
Proposition 1 provides a common admissible $\boldsymbol{\alpha} \in \operatorname{int} \Delta^3$ for which the topology scores differ. The two graphs therefore satisfy all conditions of the proposition, and a selective topology-aware node gate with $a \neq 0$ produces different vector-field updates. Thus the mechanism separates this 1-WL-indistinguishable pair. □

C.9 Topological Intuition of the Centrality Gate

Figure 1(b) sketches how $\boldsymbol{\gamma}$ distinguishes a structural hub from a peripheral branch; here we make that pictorial contrast *computable*. On a minimal *kite* graph (Figure 3)—a central **Blue Node** in a 4-cycle "body," connected by spokes to all four cycle vertices, with a 2-edge tail ending at a degree-1 **Orange Node**—we first tabulate nontrivial closed walks in the loop-free graph. These counts isolate motif contributions to the diagonal statistics in $c_i = \sum_k \alpha_k (\hat{A}^k)_{ii}$ (Eq. (9)); the implementation's self-loops add further baseline walks.

Closed walks as local motifs. For any node i, the diagonal $(\hat{A}^k)_{ii}$ counts weighted closed walks of length k that return to i—the same local motif statistics that enter subgraph centrality (Estrada & Rodríguez-Velázquez, 2005) and path-integral graph filters (Ma et al., 2020). Summing these diagonals with state-adaptive walk-length weights $\alpha_k(\mathbf{H}, t)$ yields our score c_i in Eq. (9). This is a finite-order analogue of the matrix-exponential subgraph-centrality score $\left[e^{\hat{A}}\right]_{ii}$ used in subgraph centrality (Estrada & Hatano, 2008): we never materialise $e^{\hat{A}}$, but the gate can learn to privilege nodes that sit in dense, loop-rich motifs during continuous integration.

Hub vs. tail on the kite. In the loop-free kite, the **Blue Node** closes 4 walks at k=2 and 8 at k=3 (Figure 3(b); each geometric motif in panel (a) is counted in both orientations). The **Orange Node** closes only one backtracking walk at k=2 and none at k=3. With the model's self-loops, both nodes acquire additional closed walks, so the tip's length-3 diagonal is generally nonzero; the relevant signal is the hub's additional motif mass, not a zero-versus-nonzero dichotomy. The learned gate can use this contrast to downweight thin peripheral branches when useful for the task.



(a) Kite topology.

(b) Closed walks (loop-free; $\times 2$).

Walk length k	1	2	3
Structural hub	0	4	8
Peripheral node	0	1	0

Figure 3: Quantitative complement to Figure 1(b). **Blue Node**: dense 4-cycle body; **Orange Node**: acyclic tail tip. Panel (b) counts nontrivial closed walks in the loop-free graph ($\times 2$, undirected). Self-loops used by the model add baseline walks at both nodes but do not erase the hub's additional motif contributions.

Why a gate is needed in continuous time. Fixed one-hop graph ODEs (Poli et al., 2019; Fang et al., 2021) and attention-diffusion models with state-dependent *edge* weights but fixed one-hop reach (Chamberlain et al., 2021) scale the same vector-field template at every node. Over an integration horizon T, noise from peripheral nodes can accumulate on equal footing with hub nodes. PIANO couples the path-integral operator to a per-node gate (Eqs. (8) and (9)). The gate can learn to upweight nodes embedded in active motifs and suppress isolated branches; when $a, b > 0$, larger c_i or $\|\mathbf{h}_i\|_2$ increases γ_i. The kite is an illustrative structural example. Learned node modulation is tested on Springs (Section 5.1); SEVIR is not used as causal evidence that the gate isolates storm cores (Appendix F).

D Controlled Mechanism Experiments

The central mechanism of PIANO is the state-adaptive path-integral propagator $M_\theta(\mathbf{H}, t)$, parameterised by a state-conditioned walk-length distribution. The selective topology-aware node gate is an auxiliary channel: it helps when node roles carry useful heterogeneous structure, and is not assumed to be universally necessary.

We use controlled forecasting benchmarks to isolate two questions: (i) can the walk-length weights adapt to state-dependent spatial interaction range, and (ii) can selective topology-aware node modulation exploit heterogeneous structural roles? Graph-wave-Scale isolates the first mechanism, NRI Springs with observed graphs isolates the second, and Graph-wave-Speed controls for changes in temporal propagation rate.

D.1 Controlled-benchmark protocols

NRI Springs: selective topology-aware node modulation. Each trajectory follows the NRI Springs protocol (Kipf et al., 2018) with $N{=}10$ particles, features $(x, y, \dot{x}, \dot{y})$, ten observed frames, twenty forecast frames, and an $8000/1000/1000$ split. A new interaction graph is sampled for every trajectory. Our mechanism-matched experiments supply the observed ground-truth adjacency to PIANO, One-hop Graph ODE, GRAND-nl, HOPE, and the learned-fixed and Boltzmann controls; broader fixed/learned-graph baselines are reported only as contextual comparisons. The three PIANO settings share the same benchmark-specific backbone and gate setting and differ only in their walk-length controller:

- **PIANO**: uses the adaptive $\boldsymbol{\alpha}(\mathbf{H}, t)$.
- **PIANO** (learned-fixed $\boldsymbol{\alpha}$): uses $\boldsymbol{\alpha} = \mathrm{softmax}(\mathbf{z})$, equivalently $\alpha_\ell = e^{z_\ell} / \sum_{j=0}^{L} e^{z_j}$, as one learned global state- and time-independent mixture.

- **PIANO** (Boltzmann-fixed $\boldsymbol{\alpha}$): uses the non-trainable short-walk prior $\alpha_\ell = e^{-\ell}/\sum_{j=0}^{L} e^{-j}$, $\ell = 0, \ldots, L$. It is evaluated only on Springs; for Graph-wave-Scale, whose latent factor changes spatial range, learned-fixed is the relevant state-independent control.

The controlled factor is spring stiffness $\kappa \in \{0.1, 1.0\}$, which scales the spring contribution of each edge while leaving the graph-generation process unchanged. Springs therefore evaluates the structural statistic $c_i(\mathbf{H}, t)$ in Eq. (9) and its node-wise modulation through $\boldsymbol{\gamma}$ in Eq. (8).

Figure 4 summarises the common Graph-wave topology and the controlled distinction between spatial scale and propagation rate.

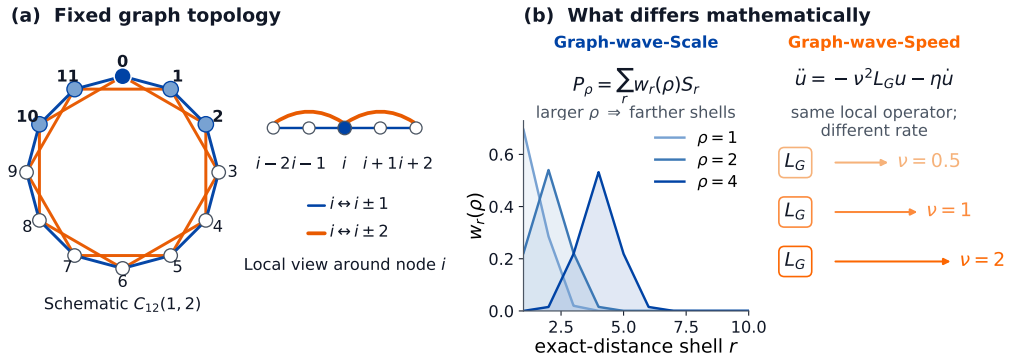


Figure 4: **Topology and generative distinction of Graph-wave.** **(a)** Both variants use the same frozen degree-4 circulant lattice. The displayed $C_{12}(1, 2)$ graph is a readable schematic of the $C_{64}(1, 2)$ benchmark: blue edges connect i to $i \pm 1$, and orange edges connect i to $i \pm 2$ modulo the number of nodes. **(b)** Graph-wave-Scale changes the spatial operator by shifting direct interaction mass towards more distant exact-distance shells as ρ increases. Graph-wave-Speed retains the same local Laplacian L_G and changes only its coefficient ν^2. Scale therefore changes which graph regions directly influence a node, whereas Speed changes how rapidly the same local interaction propagates.

Graph-wave: state-adaptive path-integral propagation. Both Graph-wave variants use the same fixed degree-4 circulant graph $C_{64}(1, 2)$ with N=64, node features $(u, \dot{u})$, ten observed frames, twenty forecast frames, and a 7998/999/999 split (Figure 4a). Feature normalisation is estimated from the training set only. The forecasting model observes the graph and trajectory history, but not the latent simulator parameter.

Graph-wave-Scale changes the direct spatial interaction range $\rho \in \{1, 2, 4\}$ by shifting interaction mass across exact graph-distance shells (Figure 4b, left). It is the primary test of whether the state-conditioned walk-length weights $\boldsymbol{\alpha}(\mathbf{H}, t)$ adapt the effective propagation order in $M_\theta(\mathbf{H}, t)$.

Graph-wave-Speed retains the same local Laplacian and varies only the temporal rate $\nu \in \{0.5, 1, 2\}$ (Figure 4b, right). It serves as a control distinguishing adaptation to spatial interaction range from sensitivity to faster local propagation.

Unless stated otherwise, Graph-wave comparisons use the gate-off setting $\boldsymbol{\gamma} = \mathbf{1}$ with $(L, T) = (8, 1)$. Full generator mathematics appear in Appendix D.3.

Training protocol. Except in the dedicated $L \times T$ analysis, Springs uses the gate-enabled controller $\boldsymbol{\alpha}(\mathbf{H}, t)$ with L=8 and T=0.5 (41.1K parameters); Graph-wave uses the corresponding gate-off Scale setting (33.4K). Models are trained for forty epochs with batch size 64, and the best validation checkpoint is selected. ODE integration uses RK4 with four vector-field evaluations, matching the frozen Graph-wave-Scale recipe in Table 6. All reported comparisons use ten fixed seeds unless noted otherwise.

Table 7: **Controlled forecasting MSE.** Results are mean ± sample SD over ten seeds. Bold and underlined entries denote the best and second-best mechanism-matched results in each column; contextual rows use a shared or learned graph and are unhighlighted. †Graph-wave-Scale with $\boldsymbol{\gamma} = \mathbf{1}$; ‡Springs with the node-wise gate enabled where applicable. "—" denotes not evaluated.

Model	Springs κ=0.1 ($\times 10^{-4}$)	Springs κ=1.0 ($\times 10^{-3}$)	Graph-wave-Scale ($\times 10^{-5}$)	Trainable params (Scale / Springs)
Contextual discrete / recurrent forecasters				
Graph WaveNet (Wu et al., 2019)	18.0 ±1.3	62.2 ±1.0	5.01 ±1.15	62.3K† / 63.9K‡
AGCRN (Bai et al., 2020)	14.2 ±1.4	48.4 ±5.7	42.3 ±4.7	752.0K† / 761.8K‡
LSTM (Hochreiter & Schmidhuber, 1997)	21.4 ±0.6	156 ±2	59.3 ±5.6	53.3K† / 56.4K‡
Contextual continuous-time graph models				
STGODE (Fang et al., 2021)	14.3 ±0.6	69.1 ±1.5	14.3 ±1.5	136.8K† / 142.1K‡
STG-NCDE (Choi et al., 2022)	16.0 ±3.6	80.7 ±15.1	46.3 ±26.4	378.2K† / 388.2K‡
GG-ODE (Huang et al., 2023)	349 ±61	137 ±7	28.5 ±9.4	149.8K† / 150.1K‡
CSG-ODE (Wang et al., 2025)	22.4 ±1.5	48.6 ±4.4	3.26 ±0.58	553.7K† / 548.3K‡
Mechanism-matched models and controls				
One-hop Graph ODE (Poli et al., 2019)	9.35 ±0.57	4.31 ±0.16	55.5 ±2.0	28.6K† / 36.3K‡
GRAND-nl (Chamberlain et al., 2021)	11.7 ±0.2	55.0 ±0.7	105 ±4	32.6K† / 40.3K‡
HOPE (Luo et al., 2023)	10.4 ±0.9	6.64 ±0.26	71.7 ±4.2	28.6K† / 36.3K‡
PIANO (learned-fixed $\boldsymbol{\alpha}$)	7.56 ±0.74	<u>3.98</u> ±0.16	<u>8.61</u> ±1.37	28.6K† / 36.3K‡
PIANO (Boltzmann-fixed $\boldsymbol{\alpha}$)	**3.81** ±0.67	7.07 ±0.40	—	— / 36.3K‡
PIANO	<u>5.28</u> ±0.75	**3.79** ±0.13	**3.79** ±0.65	33.4K† / 41.1K‡

D.2 Main controlled results

We first ask whether the proposed mechanisms translate into forecasting accuracy. Table 7 compares PIANO with discrete, recurrent, and continuous-time graph baselines. The Springs columns intentionally separate mechanism-matched models, which receive the per-trajectory adjacency, from contextual baselines whose implementations use a shared or learned graph.

Graph-wave-Scale uses one fixed adjacency. On Springs, PIANO, One-hop Graph ODE, GRAND-nl, HOPE, and both fixed-$\boldsymbol{\alpha}$ controls receive the observed trajectory graph. The contextual baselines instead use a mean-training graph (Graph WaveNet, STGODE, GG-ODE, and CSG-ODE), learn a graph (AGCRN and STG-NCDE), or are graph-free (LSTM); cross-group comparisons are therefore contextual. NRI variants are omitted because they infer latent graphs.

On Springs with κ=1.0, PIANO attains the lowest displayed MSE. At κ=0.1, the hand-designed Boltzmann prior is strongest and PIANO is second, indicating that adaptive walk-length selection is not uniformly beneficial under weak coupling. On Graph-wave-Scale, CSG-ODE obtains the lowest mean MSE, followed by PIANO and Graph WaveNet. Relative to PIANO, CSG-ODE is better on seven of ten matched seeds, with paired mean difference

$$\mathrm{MSE}_{\text{CSG-ODE}} - \mathrm{MSE}_{PIANO} \approx -5.3 \times 10^{-6},$$

but uses approximately $16.6\times$ as many trainable parameters. The result therefore supports strong controlled forecasting with a favourable accuracy–parameter trade-off rather than absolute superiority on Scale MSE.

D.3 Construction and interpretation of Graph-wave

This subsection records the exact generator mathematics behind the Graph-wave protocols in Appendix D.1 and Figure 4. Both variants use the same fixed circulant lattice

$$G = C_{64}(1, 2), \qquad i \longleftrightarrow i \pm 1,\ i \pm 2 \pmod{64},$$

which has degree four, 128 undirected edges, and diameter 16. The forecasting model observes G and the trajectory history, but not the latent simulator parameter ρ or ν. The node-wise gate is disabled ($\boldsymbol{\gamma} = \mathbf{1}$) so that both variants isolate $M_\theta(\mathbf{H}, t)$.

Graph-wave-Scale: changing spatial interaction range. For each graph distance r, define the shell matrix

$$S_r(i,j) = \mathbf{1}[d_G(i,j) = r]\,,$$

and let $\widetilde{S}_r$ denote its row-normalised form. Thus, $(\widetilde{S}_r\mathbf{u})_i$ is the average displacement of nodes lying at exact graph distance r from node i. With $R = \mathrm{diam}(G) = 16$, we construct the scale-dependent interaction operator

$$P_\rho = \sum_{r=1}^{R} w_r(\rho)\widetilde{S}_r, \qquad w_r(\rho) = \frac{\exp\!\big[-(r-\rho)^2/(2\sigma_\rho^2)\big]}{\sum_{q=1}^{R}\exp\!\big[-(q-\rho)^2/(2\sigma_\rho^2)\big]},$$

where $\sigma_\rho = 0.75$ and $\rho \in \{1, 2, 4\}$. The displacement field evolves according to

$$\ddot{\mathbf{u}}(t) = -\omega^2\big(I - P_\rho\big)\mathbf{u}(t) - \eta\dot{\mathbf{u}}(t), \qquad \omega = 1.$$

Small ρ concentrates interaction mass on nearby shells; larger ρ shifts that mass outward, so ρ changes which graph distances enter the instantaneous acceleration. The latent scale ρ and operator P_ρ are used only by the simulator. Trajectories are organised into matched families that share the graph, source node, pulse sign, and initial condition, and differ only in ρ.

Graph-wave-Speed: changing temporal propagation rate. Graph-wave-Speed uses the same fixed local graph Laplacian in every regime:

$$\ddot{\mathbf{u}}(t) = -\nu^2 L_G\mathbf{u}(t) - \eta\dot{\mathbf{u}}(t), \qquad \nu \in \{0.5, 1, 2\}.$$

For a fixed state $\mathbf{u}(t)$, changing ν rescales the local restoring acceleration by ν^2: one quarter as large at $\nu = 0.5$ and four times as large at $\nu = 2$, relative to $\nu = 1$. Larger ν therefore makes the same local interaction accumulate through more successive hops within the same physical time, without introducing direct coupling to new distance shells.

D.4 What drives adaptive walk-length selection?

To convert walk order into graph distance, we precompute on the regular Graph-wave base graph

$$\rho_k^2 = \frac{1}{N}\sum_{i,j} d_G(i,j)^2[\hat{A}^k]_{ij}, \qquad \rho_\alpha(\mathbf{H},t) = \left(\sum_{k=0}^{L}\alpha_k(\mathbf{H},t)\rho_k^2\right)^{1/2}.$$

Thus ρ_α is the root-mean-square graph displacement implied by the learned walk mixture, not an estimator constrained to equal the simulator parameter ρ. Self-loops, backtracking, and repeated visits make ρ_k smaller than k, explaining why ρ_α remains below 4 even when the latent regime is $\rho = 4$ and $L = 8$.

Table 8 expands the controller comparison summarised in Figure 2a. We keep the gate off and fix $(L, T) = (8, 1)$ so that differences isolate walk-length-weight conditioning. Uniform $\boldsymbol{\alpha}$ fixes $\alpha_\ell = 1/(L+1)$ for every walk length, with no learned controller. Wide $\boldsymbol{\alpha}(t)$ is a time-only controller with a widened MLP, providing an approximately parameter-matched capacity control for state conditioning.

As summarised in Figure 2b–c, the learned effective radius tracks the hidden scale ($\mathrm{Spearman}(\rho, \rho_\alpha) = 0.943$ over 200 matched held-out families for seed 42), and intervening on $\boldsymbol{\alpha}$ alone destroys the forecast. Concretely, at each matched solver time we replace only the simplex-normalised walk-length-weight vector while retaining the recipient state, network parameters, and remaining vector-field computations. For seed 42, native MSE 3.44×10^{-5} increases to 8.44×10^{-2} after same-family walk-length schedule permutation and to 1.33–2.30×10^{-1} under local–nonlocal cross-swaps. Across independently trained seeds 42, 1, and 2, the median native MSE is 3.46×10^{-5} versus 2.20×10^{-1} under the local-to-nonlocal intervention.

D.5 When does selective topology-aware node modulation help?

The gate decomposition in Figure 5 shows that structural information matters selectively. Table 9 gives the corresponding controller and gate-on/off comparison.

Table 8: Graph-wave-Scale gate-off controller ladder with $(L, T) = (8, 1)$. Results are mean $\pm$ sample SD; relative MSE is normalised by $\boldsymbol{\alpha}(H, t)$.

Controller	Scale MSE ($\times 10^{-5}$) $\downarrow$	Params	Relative MSE	Seeds
Other mechanism-matched methods				
One-hop Graph ODE (Poli et al., 2019)	$55.5_{\pm 2.0}$	28.6K	14.6	10
GRAND-nl (Chamberlain et al., 2021)	$105_{\pm 4}$	32.6K	27.7	10
HOPE (Luo et al., 2023)	$71.7_{\pm 4.2}$	28.6K	18.9	10
PIANO controller variants				
$\boldsymbol{\alpha}(H, t)$	$\mathbf{3.79}_{\pm 0.65}$	33.4K	1.00	10
$\boldsymbol{\alpha}(H)$	$3.81_{\pm 0.62}$	33.3K	1.01	10
$\boldsymbol{\alpha}(t)$	$8.30_{\pm 1.19}$	29.3K	2.19	10
Wide $\boldsymbol{\alpha}(t)$	$8.57_{\pm 0.93}$	31.4K	2.26	10
Learned fixed $\boldsymbol{\alpha}$	$8.61_{\pm 1.37}$	28.6K	2.27	10
Uniform $\boldsymbol{\alpha}$	$11.7_{\pm 1.5}$	28.6K	3.09	10

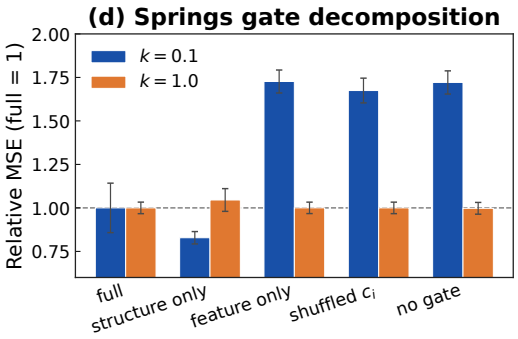


Figure 5: **Springs gate decomposition.** MSE ratios relative to the full model at weak ($\kappa = 0.1$) and strong ($\kappa = 1.0$) coupling (mean $\pm$ SD, ten seeds). Correctly aligned closed-walk information helps under weak coupling, whereas gate variants are similar at stronger coupling.

Table 9: Springs walk-length controller and gate ablation over ten seeds. The learned fixed controller uses a trainable global mixture $\boldsymbol{\alpha} = \mathrm{softmax}(\mathbf{z})$ independent of state and time. The Boltzmann controller uses the non-trainable prior $\alpha_\ell \propto e^{-\ell}$.

Walk-length weights	Gate	$\kappa = 0.1$ ($\times 10^{-4}$)	$\kappa = 1.0$ ($\times 10^{-3}$)
Adaptive $\boldsymbol{\alpha}(H, t)$	On	$5.28_{\pm 0.75}$	$3.79_{\pm 0.13}$
Adaptive $\boldsymbol{\alpha}(H, t)$	Off	$9.09_{\pm 0.35}$	$3.78_{\pm 0.13}$
Learned fixed $\boldsymbol{\alpha}$	On	$7.56_{\pm 0.74}$	$3.98_{\pm 0.16}$
Learned fixed $\boldsymbol{\alpha}$	Off	$7.52_{\pm 0.74}$	$3.98_{\pm 0.16}$
Boltzmann fixed $\boldsymbol{\alpha}$	On	$3.81_{\pm 0.67}$	$7.07_{\pm 0.40}$
Boltzmann fixed $\boldsymbol{\alpha}$	Off	$3.79_{\pm 0.64}$	$7.06_{\pm 0.39}$

The expanded ablation distinguishes a trainable static controller from a hand-specified low-order prior. At $\kappa = 0.1$, the full adaptive model improves upon the learned-fixed gate-on controller by 2.28×10^{-4} MSE, with a paired 95% interval of approximately $\pm 4.5 \times 10^{-5}$. This difference should not be attributed to walk-length adaptation alone: removing the gate increases the adaptive controller's MSE to 9.09×10^{-4}, whereas the learned-fixed result remains essentially unchanged at 7.52×10^{-4}. The weak-coupling gain therefore reflects an interaction between state-adaptive walk-length weighting and selective topology-aware node modulation. The Boltzmann walk-length distribution remains strongest overall, indicating that its predefined low-order bias is particularly well matched to this regime.

At $\kappa = 1.0$, the gate has negligible effect for all three path controllers. The adaptive controller improves over the learned fixed mixture by 1.90×10^{-4}, with a paired 95% interval of approximately $\pm 5.4 \times 10^{-5}$, while both substantially outperform the Boltzmann prior. This regime therefore provides the cleaner evidence that state-adaptive walk-length selection improves upon a trainable trajectory-independent mixture.

D.6 Path order and integration horizon

Table 10: Graph-wave-Scale test MSE ($\times 10^{-5}$) across maximum explicit order L and integration horizon T for $\boldsymbol{\alpha}(H, t)$ with the gate disabled. Cell $(8, 1)$ uses ten seeds; all other cells use three seeds.

$L \backslash T$	0.5	1.0	2.0	4.0
1	45.4	27.0	19.6	17.2
2	36.9	20.8	11.1	9.28
4	9.72	8.16	6.17	4.57
8	5.42	3.79	2.65	**2.56**

Table 10 shows that on Graph-wave-Scale, increasing either L or T reduces the displayed mean MSE, with diminishing returns at larger values. The validation-selected canonical setting $(8, 1)$ remains close to the best-performing region while retaining a shorter integration horizon than $(8, 2)$ or $(8, 4)$. The grid supports complementary, task-dependent effects of L and T; it does not establish interchangeability.

E Ocean ΔSLA Details

E.1 Protocol and quantitative context

We forecast weekly North Atlantic ΔSLA derived from the AVISO absolute sea-surface height product hosted on NOAA CoastWatch ERDDAP (`erdTAssh1day`) (AVISO & NOAA NMFS SWFSC ERD, 2012), restricted to a Gulf Stream domain spanning 25–45°N and 280–330°E after longitude wrapping (Figure 6). All methods share the same upstream weeks, region crop, coarsening (c=4), ocean mask, week-over-week ΔSLA with train-only normalisation, chronological $70/10/20$ split, and $8 \to 4$ windows. Approximately N=944 ocean cells are retained; ACC is always computed on these cells. Graph models receive the corresponding node tensor and a symmetric spherical 8-NN adjacency (mean edge length $\approx 120\,\mathrm{km}$). Native-grid models receive the coarsened raster with land zeroed and are scored by restricting predictions to the same mask.

For all reported canonical graph results, propagation uses the self-loop-augmented, symmetrically normalised operator

$$\hat{A} = D^{-1/2}(A + I)D^{-1/2}.$$

Consequently, $\hat{A}^k$ in Eq. (4) represents a normalised length-k walk operator on the geographic graph. The state-dependent coefficients $\boldsymbol{\alpha}(\mathbf{H}, t)$ determine the relative contribution of these propagation orders, while $\boldsymbol{\gamma}(\mathbf{H}, t)$ modulates the resulting nonlinear update node-wise.

Native-grid baselines use compact default configurations (not a full hyperparameter search) and reload validation-selected checkpoints. They comprise ConvLSTM (Shi et al., 2015) and forecasting adaptations of OceanNet (FNO–PEC) (Chattopadhyay et al., 2024), MgNO (He et al., 2024), and

Algorithm 4 Raster ΔSLA $\to$ spherical k-NN graph (Ocean protocol).

Require: weekly SLA field $S \in \mathbb{R}^{T\times H_0\times W_0}$; region box; coarsen c=4; ocean-fraction threshold τ=0.5; k=8; chrono split ratios $0.7/0.1/0.2$

Ensure: node series $X \in \mathbb{R}^{T\times N\times 1}$; binary adj. $A \in \{0,1\}^{N\times N}$; mask $M \in \{0,1\}^{H\times W}$ with $N = \sum_{ij} M_{ij}$

1: Crop S to the North Atlantic box; keep weeks in $[1993, 2011]$
2: Invalid / land: set $|S|$>50 to NaN
3: Block-mean coarsen by c on (lat, lon): $S \leftarrow \mathrm{BlockMean}(S, c)$; obtain coarse coords (φ, λ) and shape (H, W)
4: Ocean fraction $f_{ij} \leftarrow$ time-mean of $\mathbf{1}[S_{\cdot,ij}\ \text{finite}]$
5: Retain cells $M_{ij} \leftarrow \mathbf{1}[f_{ij} > \tau] \wedge \mathbf{1}[S_{\cdot,ij}\ \text{all finite}]$
6: Flatten retained cells to nodes $i = 1, \ldots, N$ with coords (φ_i, λ_i) and series $Y_{t,i} \leftarrow S_{t,ij}$
7: Target ΔSLA: $Y_{t,i} \leftarrow Y_{t,i} - Y_{t-1,i}$ (first week zero); clip $\pm 3\sigma$ on *train* weeks only
8: Normalise by train std: $X \leftarrow Y/\sigma_{\text{train}}$; land never enters X
9: Spherical distances $d_{ij} \leftarrow \mathrm{Haversine}((\varphi_i, \lambda_i), (\varphi_j, \lambda_j))$
10: For each i: connect to k nearest $j \neq i$; set $A \leftarrow \max(A, A^\top)$ (symmetric binary k-NN)
11: Chrono split weeks $70/10/20$; build $8\to 4$ sliding windows
12: **Optional (PIANO / matched ODE):** $\hat{A} \leftarrow D^{-1/2}(A + I)D^{-1/2}$
13: **Native-grid branch:** keep full field $G_{t,:,:} \leftarrow$ scatter $X_{t,\cdot}$ onto M; land=0; models consume $(B, 8, 1, H, W)$ and are scored via $\widehat{Y}[\ldots, M{=}1]$

Table 11: Shared upstream preprocessing with family-specific tensor layout and spatial inductive bias. All methods are scored on the same $N{\approx}944$ retained ocean cells.

Family	Models	Input	Spatial structure	Land handling
Native-grid	ConvLSTM, OceanNet, MgNO, Swin-UNet	$B{\times}8{\times}1{\times}H{\times}W$	raster operators; optional static mask channel	land=0; masked loss; ACC on $M{=}1$
Discrete / recurrent forecasters	STGCN, Graph WaveNet, AGCRN, LSTM	$B{\times}8{\times}N{\times}1$	physical k-NN A (Graph WaveNet: +learned adj.; AGCRN: learned only; LSTM: none)	nodes = ocean cells only
Continuous-time graph	STGODE, STG-NCDE, GG-ODE, CSG-ODE, GRAND-nl, HOPE, PIANO	$B{\times}8{\times}N{\times}1$	physical k-NN; PIANO/matched use $\hat{A}$; STG-NCDE adaptive; CSG-ODE physical+latent mix	nodes = ocean cells only

Swin-UNet (Cao et al., 2023); tables use the short names OceanNet, MgNO, and Swin-UNet for the three adaptations under the shared $8\to 4$ ΔSLA protocol.

Runtime columns in Tables 12 and 14 use the same GPU (Quadro RTX 8000) on the seed-42 checkpoint at each model's native batch size (B=16, except CSG-ODE at B=8). Train time is mean $\pm$ sample SD over three timed epochs after one warmup epoch (forward + backward + optimiser). Inference is milliseconds per batch and per sample over the full test loader after five warmup batches with CUDA synchronisation; compare per-sample times across the two batch sizes. Peak_{tr} reports peak training memory. Native-grid rows use the raster datamodule with ocean masking. Published STG-NCDE accuracy was trained at B=8; its runtime row uses B=16.

E.2 Accuracy, forecast horizon, and resource efficiency

Table 12 reports the complete comparison of native-grid, discrete or recurrent, and continuous-time graph models. Native-grid rows are evaluated on the same $N{\approx}944$ ocean cells as the graph models. GRAND-nl and HOPE use the same normalised geographic operator $\hat{A}$ as PIANO. One-hop Graph ODE and learned-fixed $\boldsymbol{\alpha}$ isolate multi-hop propagation and state-dependent walk-length adaptation.

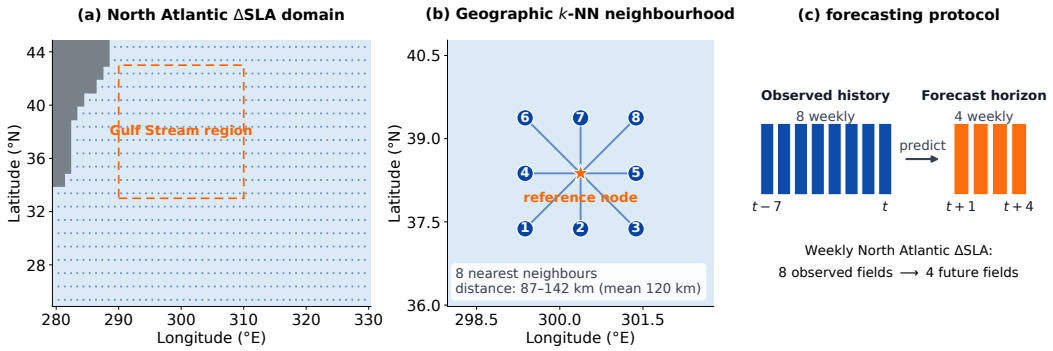


Figure 6: **North Atlantic ΔSLA forecasting setup.** **(a)** Forecast domain and highlighted Gulf Stream region; dots denote the approximately 944 valid ocean nodes. **(b)** Geographic 8-nearest-neighbour connectivity around one reference node, with neighbour distances ranging from approximately 87 to 142 km. **(c)** Eight observed weekly fields are used to forecast the following four weeks.

Table 12: Weekly North Atlantic ΔSLA forecasting and efficiency (8→4 weeks). ACC is mean ± sample SD over ten seeds. Bold and underlined values are the best and second-best results among continuous-time graph models; other families are contextual references. †CSG-ODE reimplementation. ‡Forecasting adaptations; see Appendix E.1.

		Forecast skill			Efficiency		
Model	Params	ACC_{+1} ↑	ACC_{+4} ↑	Mean ACC↑	T_{ep} (s)↓	T_{inf} (ms)↓	M^{tr}_{peak} (MB)↓
Native-grid forecasters							
ConvLSTM (Shi et al., 2015)	112.0K	0.767 ±0.004	0.369 ±0.006	0.547 ±0.004	1.33 ±0.10	0.69	503
OceanNet‡ (Chattopadhyay et al., 2024)	1.05M	0.724 ±0.004	0.273 ±0.002	0.458 ±0.002	2.92 ±0.08	1.63	325
MgNO‡ (He et al., 2024)	736.6K	0.763 ±0.004	0.320 ±0.005	0.508 ±0.003	0.39 ±0.12	0.16	192
Swin-UNet‡ (Cao et al., 2023)	1.64M	0.567 ±0.003	0.251 ±0.005	0.378 ±0.004	1.06 ±0.05	1.05	354
Discrete / recurrent forecasters							
STGCN (Yu et al., 2018)	189.8K	0.743 ±0.003	0.225 ±0.006	0.441 ±0.005	1.91 ±0.04	1.13	781
Graph WaveNet (Wu et al., 2019)	82.9K	0.779 ±0.001	0.322 ±0.003	0.515 ±0.002	2.06 ±0.06	1.23	1078
AGCRN (Bai et al., 2020)	754.7K	0.758 ±0.001	0.298 ±0.002	0.484 ±0.001	6.60 ±0.02	2.90	3571
LSTM (Hochreiter & Schmidhuber, 1997)	50.7K	0.747 ±0.000	0.235 ±0.001	0.435 ±0.000	0.79 ±0.01	0.75	1113
Continuous-time graph models							
STGODE (Fang et al., 2021)	115.7K	0.773 ±0.000	0.279 ±0.001	0.486 ±0.001	1.23 ±0.10	0.65	536
GRAND-nl (Chamberlain et al., 2021)	26.4K	0.752 ±0.001	0.234 ±0.001	0.440 ±0.001	3.41 ±0.40	2.92	662
STG-NCDE (Choi et al., 2022)	389.1K	0.694 ±0.019	0.189 ±0.036	0.381 ±0.034	26.40 ±0.06	11.47	1106
HOPE (Luo et al., 2023)	**22.4K**	0.766 ±0.001	0.278 ±0.001	0.481 ±0.000	2.05 ±0.06	2.24	137
GG-ODE (Huang et al., 2023)	149.6K	0.738 ±0.011	0.320 ±0.007	0.498 ±0.007	13.03 ±0.01	7.92	6550
CSG-ODE† (Wang et al., 2025)	1.47M	0.760 ±0.002	0.298 ±0.009	0.489 ±0.004	6.41 ±0.01	3.05	1954
One-hop Graph ODE (Poli et al., 2019)	**22.4K**	0.771 ±0.001	0.308 ±0.001	0.499 ±0.001	1.77 ±0.09	2.09	**101**
PIANO (learned-fixed α)	38.6K	0.775 ±0.001	0.312 ±0.002	0.505 ±0.001	**0.71** ±0.01	**0.39**	128
PIANO	43.1K	**0.776** ±0.000	**0.323** ±0.001	**0.509** ±0.001	0.77 ±0.01	0.41	127

Figure 7 complements the numerical comparisons in Tables 12, 13, and 14. Within the continuous-time graph family, PIANO has the highest Mean ACC (0.509 ± 0.001) and second-lowest peak memory among the evaluated continuous-time graph models; the learned-fixed $\boldsymbol{\alpha}$ control is slightly faster to train and infer. Across all families, ConvLSTM retains the highest Mean ACC and MgNO the lowest runtime; the trade-off panels therefore position PIANO as a compact continuous-time alternative rather than a universal winner. Panels (a) and (b) omit the one-hop and learned-fixed controls. Among the plotted models, PIANO has the lowest peak memory, and HOPE, PIANO, Graph WaveNet, and ConvLSTM lie on the accuracy–parameter frontier. Lead-wise, PIANO is strongest among continuous-time models at weeks $+1$, $+2$, and $+4$, and shares the highest displayed week-$+3$ mean with GG-ODE.

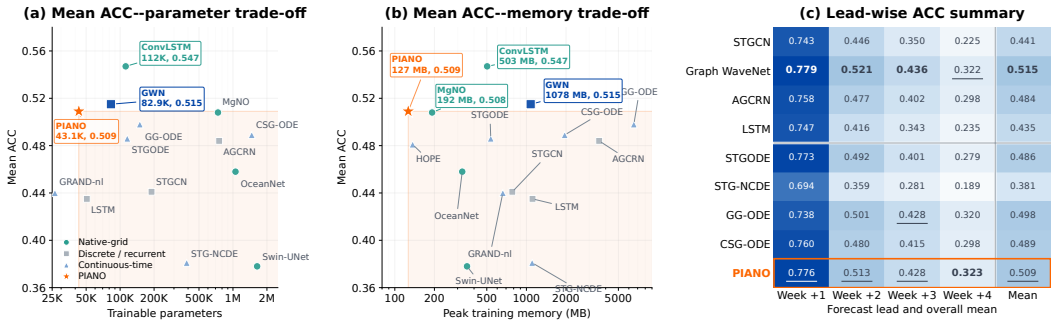


Figure 7: **Forecast accuracy and resource efficiency on weekly North Atlantic ΔSLA.** Means over ten seeds. **(a)** Mean ACC versus parameters. Shading marks models with at least as many parameters as PIANO and no higher Mean ACC. **(b)** Mean ACC versus peak training memory, shaded analogously. **(c)** Lead-wise ACC. Bold and underline mark the best and second-best continuous-time graph models; other rows are contextual.

Table 13: Lead-wise ACC on weekly North Atlantic ΔSLA ($8 \to 4$). Values are mean ± sample SD over the same ten-seed protocol as Table 12. Bold and underlined values denote the best and second-best results, respectively, among continuous-time graph models; the native-grid and discrete / recurrent families are included as contextual references. GG-ODE and PIANO have the same displayed ACC_{+3} mean and are therefore both marked best. Native-grid rows use compact default configurations.

Model	ACC_{+1}	ACC_{+2}	ACC_{+3}	ACC_{+4}	Mean ACC↑
Native-grid forecasters					
ConvLSTM (Shi et al., 2015)	$0.767_{\pm 0.004}$	$0.569_{\pm 0.005}$	$0.482_{\pm 0.004}$	$0.369_{\pm 0.006}$	$0.547_{\pm 0.004}$
OceanNet (Chattopadhyay et al., 2024)	$0.724_{\pm 0.004}$	$0.446_{\pm 0.002}$	$0.388_{\pm 0.003}$	$0.273_{\pm 0.002}$	$0.458_{\pm 0.002}$
MgNO (He et al., 2024)	$0.763_{\pm 0.004}$	$0.515_{\pm 0.007}$	$0.436_{\pm 0.003}$	$0.320_{\pm 0.005}$	$0.508_{\pm 0.003}$
Swin-UNet (Cao et al., 2023)	$0.567_{\pm 0.003}$	$0.374_{\pm 0.007}$	$0.321_{\pm 0.007}$	$0.251_{\pm 0.005}$	$0.378_{\pm 0.004}$
Discrete / recurrent forecasters					
STGCN (Yu et al., 2018)	$0.743_{\pm 0.003}$	$0.446_{\pm 0.005}$	$0.350_{\pm 0.007}$	$0.225_{\pm 0.006}$	$0.441_{\pm 0.005}$
Graph WaveNet (Wu et al., 2019)	$0.779_{\pm 0.001}$	$0.521_{\pm 0.002}$	$0.436_{\pm 0.002}$	$0.322_{\pm 0.003}$	$0.515_{\pm 0.002}$
AGCRN (Bai et al., 2020)	$0.758_{\pm 0.001}$	$0.477_{\pm 0.001}$	$0.402_{\pm 0.002}$	$0.298_{\pm 0.002}$	$0.484_{\pm 0.001}$
LSTM (Hochreiter & Schmidhuber, 1997)	$0.747_{\pm 0.000}$	$0.416_{\pm 0.001}$	$0.343_{\pm 0.001}$	$0.235_{\pm 0.001}$	$0.435_{\pm 0.000}$
Continuous-time graph models					
STGODE (Fang et al., 2021)	$0.773_{\pm 0.000}$	$0.492_{\pm 0.001}$	$0.401_{\pm 0.001}$	$0.279_{\pm 0.001}$	$0.486_{\pm 0.001}$
GRAND-nl (Chamberlain et al., 2021)	$0.752_{\pm 0.001}$	$0.429_{\pm 0.001}$	$0.343_{\pm 0.001}$	$0.234_{\pm 0.001}$	$0.440_{\pm 0.001}$
STG-NCDE (Choi et al., 2022)	$0.694_{\pm 0.019}$	$0.359_{\pm 0.057}$	$0.281_{\pm 0.046}$	$0.189_{\pm 0.036}$	$0.381_{\pm 0.034}$
HOPE (Luo et al., 2023)	$0.766_{\pm 0.001}$	$0.484_{\pm 0.001}$	$0.395_{\pm 0.001}$	$0.278_{\pm 0.001}$	$0.481_{\pm 0.000}$
GG-ODE (Huang et al., 2023)	$0.738_{\pm 0.011}$	$0.501_{\pm 0.010}$	$\underline{0.428}_{\pm 0.007}$	$\underline{0.320}_{\pm 0.007}$	$0.498_{\pm 0.007}$
CSG-ODE (Wang et al., 2025)	$0.760_{\pm 0.002}$	$0.480_{\pm 0.004}$	$0.415_{\pm 0.004}$	$0.298_{\pm 0.009}$	$0.489_{\pm 0.004}$
One-hop Graph ODE (Poli et al., 2019)	$0.771_{\pm 0.001}$	$0.500_{\pm 0.001}$	$0.416_{\pm 0.001}$	$0.308_{\pm 0.001}$	$0.499_{\pm 0.001}$
PIANO (learned-fixed $\boldsymbol{\alpha}$)	$\underline{0.775}_{\pm 0.001}$	$\underline{0.510}_{\pm 0.001}$	$0.423_{\pm 0.002}$	$0.312_{\pm 0.002}$	$\underline{0.505}_{\pm 0.001}$
PIANO	$\mathbf{0.776}_{\pm 0.000}$	$\mathbf{0.513}_{\pm 0.001}$	$\mathbf{0.428}_{\pm 0.002}$	$\mathbf{0.323}_{\pm 0.001}$	$\mathbf{0.509}_{\pm 0.001}$

Table 14: Same-GPU runtime on weekly North Atlantic ΔSLA ($N \approx 944$). Bold and underlined values are the best and second-best among continuous-time graph models (lower better); other families are contextual references.

Model	Params	B	Train (s/ep)	Infer (ms/b)	Infer (ms/sam)	Peak_{tr} (MB)
Native-grid forecasters						
ConvLSTM (Shi et al., 2015)	112.0K	16	$1.33_{\pm 0.10}$	11.0	0.69	503
OceanNet (Chattopadhyay et al., 2024)	1.05M	16	$2.92_{\pm 0.08}$	26.1	1.63	325
MgNO (He et al., 2024)	736.6K	16	$0.39_{\pm 0.12}$	2.6	0.16	192
Swin-UNet (Cao et al., 2023)	1.64M	16	$1.06_{\pm 0.05}$	16.8	1.05	354
Discrete / recurrent forecasters						
STGCN (Yu et al., 2018)	189.8K	16	$1.91_{\pm 0.04}$	18.0	1.13	781
Graph WaveNet (Wu et al., 2019)	82.9K	16	$2.06_{\pm 0.06}$	19.7	1.23	1078
AGCRN (Bai et al., 2020)	754.7K	16	$6.60_{\pm 0.02}$	46.4	2.90	3571
LSTM (Hochreiter & Schmidhuber, 1997)	50.7K	16	$0.79_{\pm 0.01}$	11.9	0.75	1113
Continuous-time graph models						
STGODE (Fang et al., 2021)	115.7K	16	$1.23_{\pm 0.10}$	10.4	0.65	536
GRAND-nl (Chamberlain et al., 2021)	<u>26.4K</u>	16	$3.41_{\pm 0.40}$	46.7	2.92	662
STG-NCDE (Choi et al., 2022)	389.1K	16	$26.40_{\pm 0.06}$	183.6	11.47	1106
HOPE (Luo et al., 2023)	**22.4K**	16	$2.05_{\pm 0.06}$	35.8	2.24	137
GG-ODE (Huang et al., 2023)	149.6K	16	$13.03_{\pm 0.01}$	126.7	7.92	6550
CSG-ODE (Wang et al., 2025)	1.47M	8	$6.41_{\pm 0.01}$	24.4	3.05	1954
One-hop Graph ODE (Poli et al., 2019)	**22.4K**	16	$1.77_{\pm 0.09}$	33.4	2.09	**101**
PIANO (learned-fixed $\boldsymbol{\alpha}$)	38.6K	16	$\mathbf{0.71}_{\pm 0.01}$	**6.3**	**0.39**	128
PIANO	43.1K	16	<u>$0.77_{\pm 0.01}$</u>	<u>6.6</u>	<u>0.41</u>	<u>127</u>

E.3 State-adaptive spatial computation

Figure 9 provides mechanism-level diagnostics of the two adaptive controls introduced by PIANO: $\boldsymbol{\alpha}(\mathbf{H}, t)$ selects the relative contribution of different propagation orders, whereas $\boldsymbol{\gamma}(\mathbf{H}, t)$ modulates the nonlinear update node-wise.

All panels use the canonical ocean checkpoint trained with

$$\hat{A} = D^{-1/2}(A + I)D^{-1/2},$$

where $A = A^\top$ is the symmetrized binary geographic k-NN adjacency and D is the degree matrix of $A + I$. Thus the diagnostics correspond directly to the normalised operator in Eq. (4) and the closed-walk score in Eq. (9).

Adaptive walk-length selection. Figure 9(a) shows the complete walk-length-weight trajectory for the high-activity case $\#157$. At the beginning of integration, the learned operator assigns approximately 0.73 mass to the one-hop term and 0.27 to the three-hop term. As the latent state evolves, the three-hop contribution decreases continuously, while the one-hop contribution increases toward approximately 0.98. The zero- and two-hop weights remain numerically close to zero and therefore largely overlap the horizontal axis.

The corresponding pointwise spatial operator therefore changes from a mixture of local and non-local propagation to an almost purely one-hop operator. Because the coefficients remain on the probability simplex,

$$M_\theta(\mathbf{H}, t) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}, t)\hat{A}^k$$

is, for each fixed $(\mathbf{H}, t)$, a convex mixture of the normalised powers. Hence the pointwise operator remains non-expansive as established by Lemma 1. This operator-level property does not by itself imply that the complete nonlinear vector field or its ODE flow is contractive.

Figure 9(b) summarizes each mixture through the effective walk length

$$\bar{k}(\mathbf{H}, t) = \sum_{k=0}^{L} k\, \alpha_k(\mathbf{H}, t).$$

We compare three cases selected from their data characteristics rather than their prediction errors: median activity, high activity, and high temporal change. Their initial effective orders are approximately 1.27, 1.54, and 1.71, respectively, and decrease to approximately 1.01, 1.04, and 1.05 by the end of integration.

The high-activity and high-change cases therefore begin with broader spatial mixtures than the median-activity case, before all three trajectories move toward predominantly one-hop propagation. Since the cases share the same learned network, integration interval, and time encoding, their different values at identical integration times provide direct empirical evidence that the state-dependent input in Eq. (5) affects the selected propagation scale. This is a diagnostic of instantaneous adaptive scale selection, not an empirical test of the frozen linear-flow approximation theorem.

Structural gate diagnostics. Let the pre-sigmoid gate logit implicit in Eq. (9) be

$$\ell_i(t) := a\, c_i(\mathbf{H}(t), t) + b\, \|\mathbf{h}_i(t)\|_2 + b_0, \qquad \gamma_i(t) = \mathrm{sigmoid}\big(\ell_i(t)\big). \tag{40}$$

To separate relative node-wise modulation from the graph-wide logit level, we inspect the centred quantity

$$\widetilde{\ell}_i(t) := \ell_i(t) - \overline{\ell}(t), \qquad \overline{\ell}(t) := \frac{1}{N}\sum_{j=1}^{N} \ell_j(t). \tag{41}$$

This centring is used only for visualization and does not modify the gate evaluated by the model. Positive $\widetilde{\ell}_i(t)$ indicates an above-average gate input at the same integration time, while negative values indicate relatively weaker modulation.

Figure 9(c) relates the normalised closed-walk score

$$c_i(t) = \sum_{k=0}^{L} \alpha_k(\mathbf{H}(t), t)(\hat{A}^k)_{ii}$$

to $\widetilde{\ell}_i(t)$ at $t = 0$, $T/2$, and T. The displayed scores remain approximately within 0.075–0.11, consistently with the normalised formulation and the bound $0 \le c_i(t) \le 1$. Light points show the complete node distributions, and solid curves show binned medians.

The broad vertical spread at similar values of $c_i(t)$ is expected: the structural score is only one of the two node-dependent terms in Eq. (40). The evolving feature norm $\|\mathbf{h}_i(t)\|_2$ also contributes to the logit. Panel (c) therefore examines how the topology-derived score relates to the learned gate input; it does not imply that structural centrality alone determines the gate.

Relative spatial modulation. Figure 9(d) maps $\widetilde{\ell}_i(t)$ at the beginning, midpoint, and end of integration. Orange regions receive above-average gate logits at the corresponding integration time, whereas blue regions receive below-average logits. The titles retain the absolute graph-wide mean and complete node-wise range:

$$\overline{\ell}(0) = 4.17, \qquad \overline{\ell}(T/2) = 4.82, \qquad \overline{\ell}(T) = 6.41,$$

with ranges

$$[2.00, 22.21], \qquad [2.36, 23.90], \qquad [2.97, 28.41].$$

All displayed logits are positive. Consequently, the corresponding sigmoid gates are predominantly open: even the minimum logit is 2.00 at $t = 0$ and 2.97 at $t = T$. The centred maps should therefore be interpreted as visualizations of the relative pre-sigmoid spatial preference and node ordering, rather than as evidence of equally large differences in the final multiplicative gains $\gamma_i(t)$.

The centred patterns remain geographically structured and evolve during integration. Regions with above-average logits partially align with the high-$|\Delta\mathrm{SLA}|$ contours, suggesting that the learned gate responds to spatially organized activity. This relationship is descriptive rather than causal because both the normalised closed-walk score and the latent feature norm contribute to the gate.

Together, the four panels support the intended separation of roles:

$$\boldsymbol{\alpha}(\mathbf{H}, t) \quad \text{selects the relative spatial scale of propagation,}$$

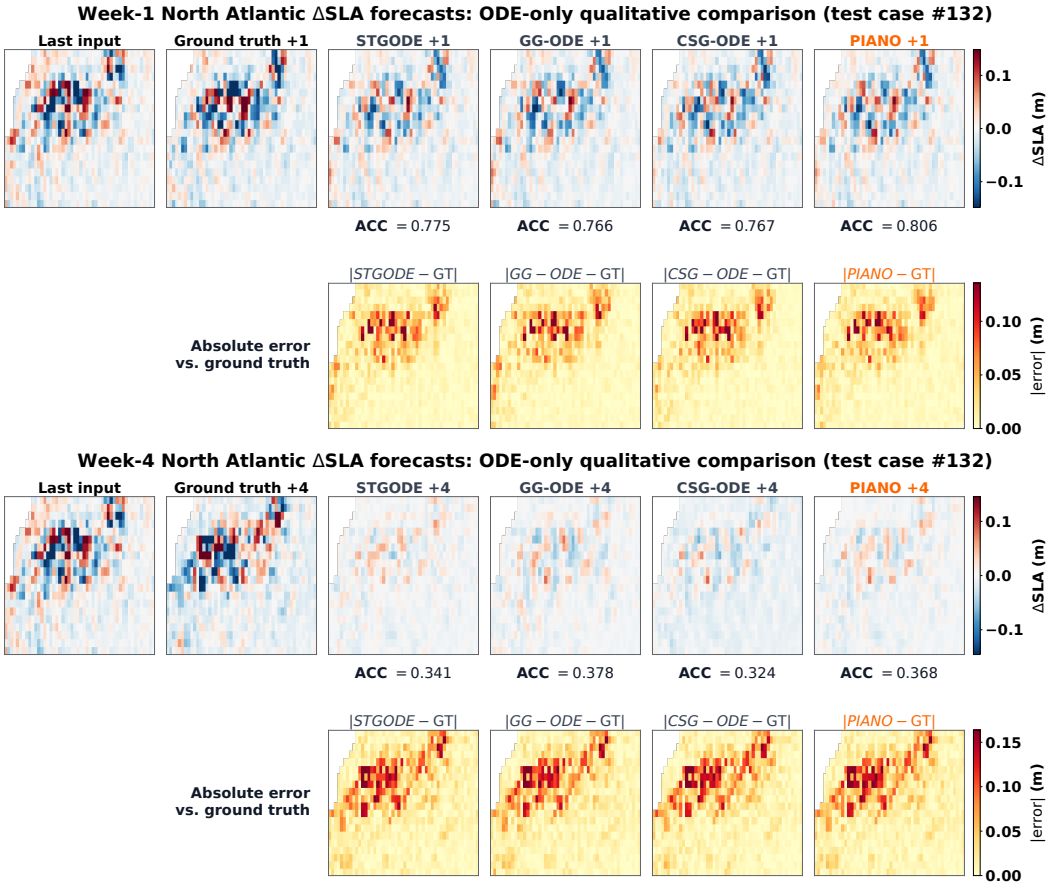


Figure 8: **ODE-only qualitative North Atlantic ΔSLA forecasts on illustrative test window #132.** Within each block, the upper row shows the final input, ground truth, and predictions from STGODE, GG-ODE, CSG-ODE, and PIANO, while the lower row shows absolute errors against the corresponding ground truth. Signed fields share one zero-centred colour scale within each lead, and error maps share one nonnegative scale. ACC and RMSE values are sample-specific to this example and are not used for ranking; aggregate performance over ten seeds is reported in Table 12.

whereas

$$\boldsymbol{\gamma}(\mathbf{H}, t) \quad \text{modulates the nonlinear update across nodes.}$$

The observed variation over integration time, input cases, and spatial locations demonstrates learned state-adaptive behaviour rather than repeated application of one fixed propagation profile. These panels are mechanism visualizations, not causal tests of component utility; the performance contribution of each adaptive component is assessed through the corresponding controlled ablations.

E.4 Qualitative forecasts across lead time

Figure 8 compares continuous-time graph forecasters on illustrative held-out case #132 at weeks +1 and +4. The sample-specific maps show short- and long-lead behaviour under a shared initial condition and are not used for model ranking or mechanism attribution. Aggregate conclusions rely on Table 12 and Figure 7.

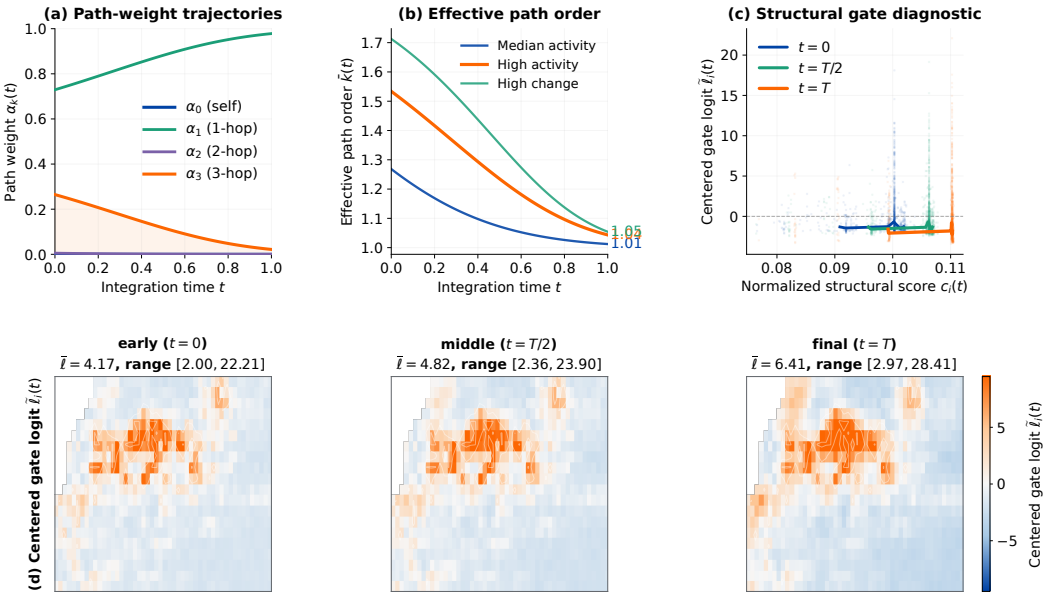


Figure 9: **State-adaptive spatial computation on North Atlantic ΔSLA.** All panels use the canonical self-loop-augmented, symmetrically normalised graph. **(a)** Walk-length-weight trajectories for high-activity case #157, shifting from mixed one- and three-hop propagation toward predominantly one-hop propagation. **(b)** Effective walk length for median-activity, high-activity, and high-change inputs, showing input-dependent scale selection. **(c)** Normalised structural score versus centred gate logit at $t = 0$, $T/2$, and T; curves show binned medians. **(d)** Centred gate-logit maps at the same times; white contours mark high-$|\Delta\text{SLA}|$ regions. Titles report the absolute logit mean and range, and the shared colour scale is percentile-clipped.

F SEVIR Radar-Nowcasting Experiments

SEVIR is used as a high-dimensional transfer, robustness, and functional-use study, not as the primary mechanism demonstration. Graph-wave-Scale and Springs remain the primary controlled tests of adaptive walk-length weighting and selective topology-aware node modulation. The default window-level train/validation split can place overlapping 2018 events in both subsets; the event-level robustness check is reported in Section F.3.

F.1 Dataset, split, and latent-graph protocol

The input is a 128×128 VIL raster. A convolutional encoder maps each frame to a 32×32 latent grid ($N = 1024$ nodes), on which we construct a fixed spatial k-NN scaffold ($k = 8$) from latent-cell coordinates using Euclidean distance. Directed k-NN edges are symmetrised to form an unweighted adjacency A; unit self-loops are then added and symmetrically normalised as $\hat{A} = D^{-1/2}(A + I)D^{-1/2}$. The scaffold is identical across samples and latent-graph models, and an architecture-identical decoder maps the latent prediction back to pixel space (Figure 10). All latent-backbone rows use architecture-identical encoder/decoder backbones, latent width, normalisation, optimiser, and training budget, and are trained end-to-end. Graph-based rows additionally share the same fixed k-NN scaffold. The primary compact-transfer task reported in the main text and appendix is $5\rightarrow2$. The completed window-split $5\rightarrow10$ comparison is retained only as a secondary long-horizon stress test. CSI thresholding is applied to normalised VIL.

The default train/validation protocol (`split_level=window`) permutes overlapping 2018 windows after extraction using seed 42, yielding 308,206/17,122 train/validation windows; the full 2019 set (353,073 windows) is reserved for testing. Because windows from the same 2018 event can enter both training and validation, this protocol makes validation windows statistically depen-

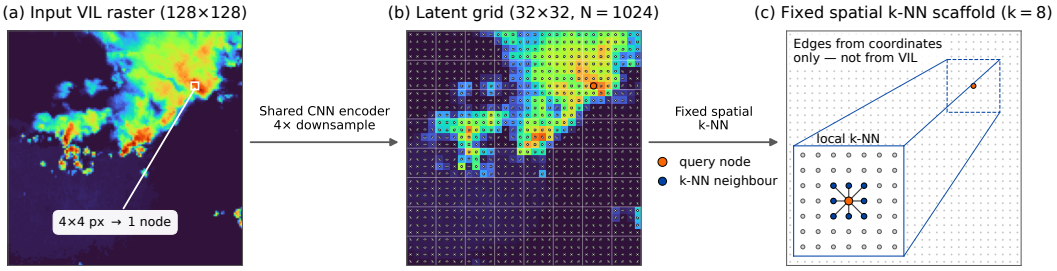


Figure 10: **Latent-graph protocol for SEVIR.** Each 128×128 VIL frame is mapped by a CNN encoder to a 32×32 latent grid with $N = 1024$ spatial nodes. A fixed k-NN scaffold ($k = 8$) is constructed from latent-cell coordinates only and shared across samples and latent-graph models. Thus the graph structure is sample-invariant, while encoded node states depend on the input VIL field. The latent heatmap is schematic and illustrates spatial correspondence; the learned latent representation is multi-channel.

dent on training windows from the same event and can affect checkpoint selection. We therefore additionally evaluate an event-level protocol (`split_level=event`) that partitions events before window extraction, yielding $7167/398$ train/validation events and $308{,}181/17{,}114$ windows, with the same 2019 test set (Section F.3). VIL values are normalised by 255, and the latent graph construction is fixed across models. Unless stated otherwise, matched $5\to2$ and $5\to10$ comparisons use seeds $42, 1, 2, 3, 7, 5, 11, 13, 17, 23$; the event-split and inference-time intervention diagnostics use five seeds.

F.2 Forecasting performance and efficiency

Table 15 reports the window-split $5\to2$ comparison. Graph WaveNet attains the strongest displayed predictive metrics, while PIANO achieves the second-lowest MSE and second-highest audited CSI@0.5/0.7 with 69.9K parameters. Relative to the matched latent CNN-only backbone, GRAND-nl, HOPE, and the matched one-hop Graph ODE, PIANO improves all three metrics with nearly unchanged parameter count. We therefore use SEVIR primarily to assess cross-domain transfer and parameter efficiency.

Table 15: SEVIR VIL $5\to2$. MSE is mean $\pm$ sample SD over ten window-split seeds; CSI is global TP/FP/FN over five seeds. Bold and underline mark the best and second-best displayed results.

Model	Params	MSE (n=10)↓	CSI@0.5 (n=5)↑	CSI@0.7 (n=5)↑
Native-raster forecasters				
SimVP (Gao et al., 2022a)	309K	0.00572 $\pm$0.00023	0.6653 $\pm$0.0371	0.6869 $\pm$0.0449
Earthformer (Gao et al., 2022b)	2.0M	0.00652 $\pm$0.00050	0.6406 $\pm$0.0438	0.6506 $\pm$0.0592
Matched latent-backbone models				
Graph WaveNet (Wu et al., 2019)	171.2K	**0.00559** $\pm$0.00006	**0.7288** $\pm$0.0015	**0.7711** $\pm$0.0124
CGNN-graph (Xhonneux et al., 2020)	61.3K	0.00589 $\pm$0.00014	0.7115 $\pm$0.0126	0.7382 $\pm$0.0050
STGODE-graph (Fang et al., 2021)	65.4K	0.00587 $\pm$0.00004	0.7171 $\pm$0.0016	0.7403 $\pm$0.0011
GRAND-nl (Chamberlain et al., 2021)	69.5K	0.00582 $\pm$0.00003	0.7192 $\pm$0.0012	0.7393 $\pm$0.0055
HOPE (Luo et al., 2023)	65.4K	0.00580 $\pm$0.00004	0.7202 $\pm$0.0019	0.7420 $\pm$0.0015
Latent CNN-only	69.6K	0.00580 $\pm$0.00004	0.7190 $\pm$0.0006	0.7418 $\pm$0.0008
One-hop Graph ODE (Poli et al., 2019)	65.4K	0.00579 $\pm$0.00005	0.7202 $\pm$0.0028	0.7397 $\pm$0.0073
PIANO (ours)	69.9K	<u>0.00566</u> $\pm$0.00008	<u>0.7273</u> $\pm$0.0050	<u>0.7423</u> $\pm$0.0063

Table 16 compares efficiency on $3\times$ Quadro RTX 8000, one process per GPU, batch size B=32, and seed-42 checkpoints. Train time is mean $\pm$ sample SD over three timed windows of 1500 batches after one warmup, extrapolated to the full 9631-batch loader. Inference is milliseconds per sample over 1500 timed batches after five warmup batches, with CUDA synchronisation. Peak memory is the maximum allocated CUDA memory per GPU. PIANO requires 69.9K parameters and 349 MB peak training memory, substantially below Graph WaveNet, SimVP, and Earthformer.

The matched one-hop Graph ODE is the fastest to train (397.3 ± 1.1 s/epoch) and has the lowest training memory (287 MB); the latent CNN-only backbone is second on both. HOPE is the fastest at inference (0.64 ms/sample) but slower to train (1060.0 ± 7.9 s/epoch). GRAND-nl is slower still (1417.7 ± 8.0 s/epoch, 2.64 ms/sample) and uses substantially more peak memory (1535 MB training). The result therefore supports a compact accuracy–efficiency trade-off rather than a universal runtime advantage.

Table 16: SEVIR $5\rightarrow2$ efficiency. Bold and underline mark the best and second-best displayed results (lower better).

Model	Params	Train (s/ep)	Infer (ms/sam)	Peak_{tr} (MB)	Peak_{inf} (MB)
Native-raster forecasters					
SimVP (Gao et al., 2022a)	309.2K	2911.2 ±0.7	2.64	9306	1186
Earthformer (Gao et al., 2022b)	1997.7K	2857.0 ±0.3	2.89	8728	654
Matched latent-backbone models					
Graph WaveNet (Wu et al., 2019)	171.2K	507.2 ±4.9	1.61	584	307
STGODE-graph (Fang et al., 2021)	65.4K	464.3 ±3.3	1.55	319	<u>154</u>
GRAND-nl (Chamberlain et al., 2021)	69.5K	1417.7 ±8.0	2.64	1535	512
HOPE (Luo et al., 2023)	65.4K	1060.0 ±7.9	**0.64**	319	207
One-hop Graph ODE (Poli et al., 2019)	65.4K	**397.3** ±1.1	<u>1.32</u>	**287**	<u>154</u>
Latent CNN-only	69.6K	<u>434.4</u> ±5.1	1.44	<u>315</u>	**150**
PIANO (learned-fixed $\boldsymbol{\alpha}$)	65.4K	560.8 ±2.5	1.71	349	158
PIANO	69.9K	539.9 ±3.3	1.79	349	158

Long-horizon forecasting. Table 17 reports the completed matched window-split $5\rightarrow10$ stress test. Paired-bootstrap 95% confidence intervals over ten seeds exclude zero for the MSE differences against the latent CNN-only backbone, Graph WaveNet, and SimVP (-0.00080, -0.00104, and -0.00157, respectively). We use MSE as the principal long-horizon comparison; globally pooled CSI does not support a CSI-winning claim for PIANO.

Table 17: SEVIR VIL window-split $5\rightarrow10$ stress test. MSE is mean $\pm$ sample SD over ten seeds; CSI is global TP/FP/FN over five seeds.

Model	Params	MSE↓	ΔMSE	CSI@0.5↑	CSI@0.7↑
Latent CNN-only	69.6K	0.01607 ±0.00006	$+0.00080$	0.4405 ±0.0011	0.3803 ±0.0042
Graph WaveNet (Wu et al., 2019)	171.2K	0.01631 ±0.00076	$+0.00104$	0.4485 ±0.0116	0.4275 ±0.0305
SimVP (Gao et al., 2022a)	309K	0.01684 ±0.00136	$+0.00157$	**0.4680** ±0.0090	**0.4628** ±0.0136
PIANO	69.9K	**0.01527** ±0.00010	–	0.4641 ±0.0019	0.3817 ±0.0083

F.3 Evaluation robustness

CSI aggregation. We audit batch-mean CSI against CSI computed after pooling TP/FP/FN over the full test set. The two procedures can differ substantially: for example, SimVP $5\rightarrow2$ changes from 0.710 ± 0.009 to 0.665 ± 0.037, while PIANO $5\rightarrow10$ changes from 0.312 ± 0.110 to 0.464 ± 0.002. All CSI values reported in the paper therefore use global TP/FP/FN pooling.

Event-level split. Retraining with an event-level train/validation partition changes the exact model ordering but preserves the broader short-horizon conclusion (Table 18): SimVP becomes strongest on MSE, while PIANO remains better than the matched latent CNN-only backbone and competitive with Graph WaveNet.

F.4 Component and functional-use analysis

Table 19 reports a ten-seed 2×2 study of walk-length controller adaptivity and node gating, together with the matched latent CNN-only control. Performance differences are small, on the order of 10^{-4} MSE. Additional controller and gate decompositions are similarly tightly clustered, compatible with

Table 18: Event-level train/validation split, 5→2 official-test MSE (mean ± sample SD over five seeds).

Model	MSE↓
SimVP (Gao et al., 2022a)	**0.00546** ± 0.00017
Graph WaveNet (Wu et al., 2019)	0.00566 ± 0.00017
Latent CNN-only	0.00581 ± 0.00005
PIANO	0.00572 ± 0.00021

the fixed, nearly regular latent grid. Accordingly, SEVIR is used here as a functional-use test; controlled mechanism evidence remains provided by Graph-wave-Scale and Springs (Appendix D.5).

Table 19: SEVIR 5→2 controller × gate study. Mean ± sample SD over ten seeds.

Variant	Params	MSE↓
Latent CNN-only	69.6K	0.005799 ± 0.000043
PIANO (learned-fixed, gate off)	65.4K	0.005759 ± 0.000079
PIANO (learned-fixed, gate on)	65.4K	0.005769 ± 0.000042
Adaptive × gate off	69.9K	0.005733 ± 0.000082
PIANO (adaptive × gate on)	69.9K	0.005656 ± 0.000085

To test functional use without retraining, we intervene on five trained full-PIANO checkpoints at inference time while holding the encoder, decoder, latent state, and all remaining parameters fixed. Evaluation uses a frozen 800-batch prefix of the official test set. Permuting the inferred walk-length schedules across samples within each batch, or replacing them by the validation-set mean walk-length schedule, increases prefix MSE on four of five checkpoints (Figure 11). Seed 1 is a collapsed-$\boldsymbol{\alpha}$ checkpoint for which the permuted and mean schedules reproduce the native walk-length schedule to approximately 10^{-12}. Excluding this checkpoint, the permutation and mean interventions increase MSE by $1.10 \pm 0.48 \times 10^{-4}$ and $1.36 \pm 0.52 \times 10^{-4}$, respectively. In contrast, shuffling c_i across nodes is numerically negligible ($\Delta\text{MSE} \sim 10^{-8}$). These interventions indicate that the inferred walk-length schedule is functionally used by most trained checkpoints, whereas node-wise topology alignment has negligible predictive effect on this fixed latent radar graph.

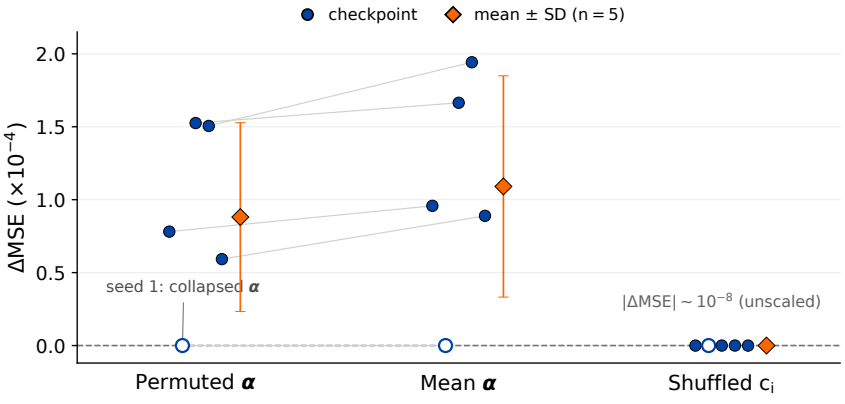


Figure 11: **Inference-time functional interventions on SEVIR.** Each point reports $\Delta\text{MSE} = \text{MSE}_{\text{intervention}} - \text{MSE}_{\text{native}}$ for one checkpoint; grey lines pair the two walk-length schedule interventions, and diamonds show mean ± SD over five checkpoints. Both $\boldsymbol{\alpha}$ interventions increase error on four of five checkpoints, whereas shuffling c_i has negligible effect. The hollow marker denotes seed 1, whose walk-length schedule collapsed.

F.5 Qualitative forecasts

We preselect validation cases using only input intensity and spatial extent, before inspecting prediction errors. Figure 12 shows four resulting convective regimes with the last input, ground truth, and PIANO forecasts at both leads; these examples are not used for ranking or mechanism claims.

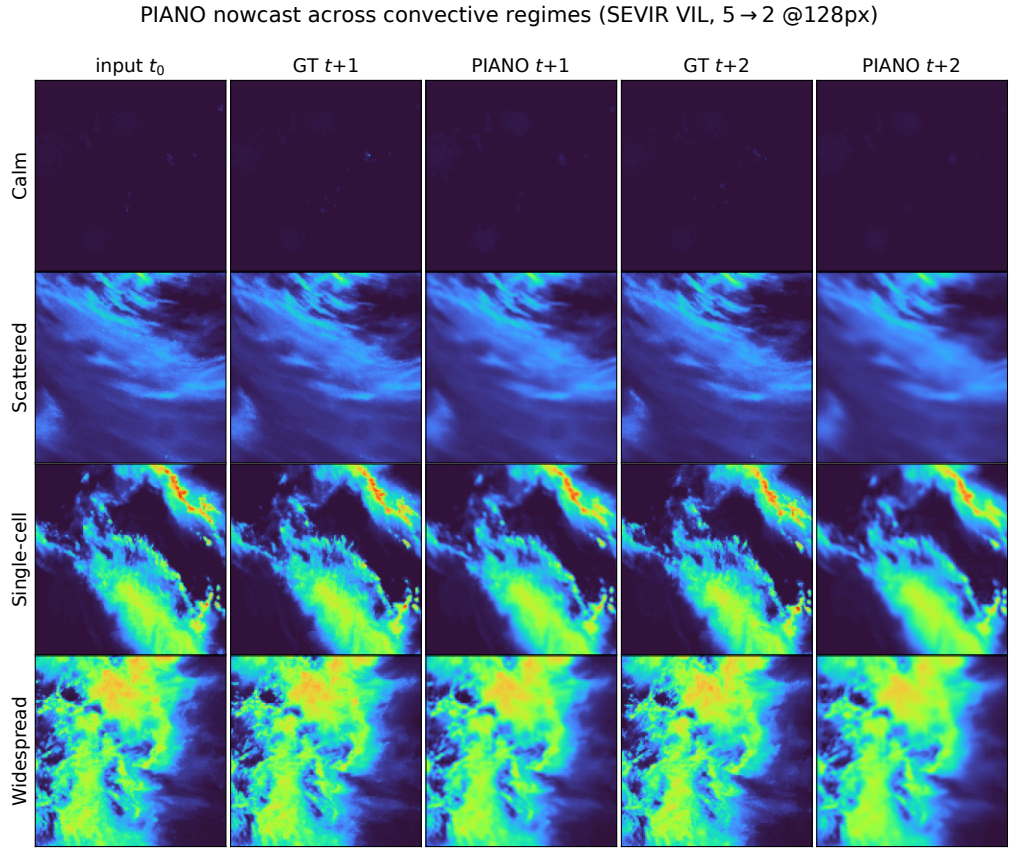


Figure 12: **PIANO nowcasts across preselected SEVIR convective regimes.** Rows show calm, scattered, single-cell, and widespread cases; columns show the last input, ground truth, and PIANO forecasts at $t+1$ and $t+2$ under a common VIL colour scale. Case selection uses validation inputs only and is fixed before inspecting prediction errors.

G Static Graph Learning and Scalability

We test whether PIANO transfers from spatiotemporal forecasting to transductive node classification and whether its sparse multi-order propagation remains practical at large graph scale. These experiments evaluate transfer and scalability rather than leaderboard-optimal configurations.

G.1 Datasets and protocol

We consider Cora, Citeseer, and PubMed using ten fixed train/validation/test splits and report mean $\pm$ sample SD of test accuracy. All methods use hidden dimension $d = 64$, validation-selected checkpoints, and matched optimisation budgets. Baselines include MLP, GCN, GPR-GNN, GRAND, GraphCON, and GREAD. Full PIANO uses the scalar gate in Equation (9); component variants selectively remove or restrict its adaptive mechanisms.

Table 20: Static node classification under the matched compact protocol. Values are mean ± sample SD over ten fixed splits. Bold denotes the highest displayed mean for each dataset.

Model	Cora	Citeseer	PubMed
MLP	0.753 $\pm$0.027	0.732 $\pm$0.017	0.863 $\pm$0.005
GCN (Kipf & Welling, 2017)	0.874 $\pm$0.013	**0.761** $\pm$0.011	0.871 $\pm$0.007
GPR-GNN (Chien et al., 2021)	**0.878** $\pm$0.013	0.760 $\pm$0.013	0.881 $\pm$0.007
GRAND (Chamberlain et al., 2021)	0.858 $\pm$0.011	0.745 $\pm$0.013	0.829 $\pm$0.004
GraphCON (Rusch et al., 2022)	0.841 $\pm$0.017	0.719 $\pm$0.018	0.841 $\pm$0.005
GREAD (Choi et al., 2023)	0.871 $\pm$0.013	0.761 $\pm$0.012	0.870 $\pm$0.006
PIANO	0.873 $\pm$0.014	0.760 $\pm$0.010	**0.895** $\pm$0.003

G.2 Small-graph node classification

PIANO is competitive across the three citation graphs and achieves the highest displayed PubMed accuracy (0.895 ± 0.003). Its PubMed improvement over GCN, GPR-GNN, and GREAD is significant under paired split-wise tests ($p = 0.002$).

G.3 Component analysis

Table 21: PIANO component analysis on static graphs. Cora, Citeseer, and PubMed report test accuracy over ten fixed splits; ogbn-arxiv reports validation and test accuracy over ten seeds. Bold denotes the highest displayed mean.

				ogbn-arxiv	
Variant	Cora	Citeseer	PubMed	Val	Test
One-hop Graph ODE	0.861 $\pm$0.016	0.750 $\pm$0.013	0.891 $\pm$0.006	0.665 $\pm$0.006	0.661 $\pm$0.006
Learned fixed $\boldsymbol{\alpha}$	0.868 $\pm$0.015	**0.761** $\pm$0.015	0.895 $\pm$0.005	0.671 $\pm$0.004	0.664 $\pm$0.005
$\boldsymbol{\alpha}(H, t)$, no gate	0.866 $\pm$0.016	0.759 $\pm$0.011	**0.897** $\pm$0.006	0.671 $\pm$0.007	0.665 $\pm$0.007
$\boldsymbol{\alpha}(H, t)$, + gate	**0.873** $\pm$0.014	0.760 $\pm$0.010	0.895 $\pm$0.003	**0.673** $\pm$0.006	**0.666** $\pm$0.007

The full model has higher displayed mean test accuracy than the one-hop variant on all four datasets. Learned fixed $\boldsymbol{\alpha}$ remains close to the adaptive controller, while the scalar gate provides no clear improvement over the no-gate variant. Static graph learning therefore supports the benefit of multi-order propagation, but does not show that state adaptation or topology-aware gating is universally necessary.

G.4 Large-graph evaluation on ogbn-arxiv

We use the official ogbn-arxiv temporal split with $90{,}941/29{,}799/48{,}603$ train/validation/test nodes. To satisfy *(A1)*, we symmetrise the directed citation graph, coalesce duplicate edges, add one self-loop per node, and construct $\hat{A} = D^{-1/2}(A + I)D^{-1/2}$, yielding $2{,}484{,}941$ nonzero entries.

The canonical PIANO configuration uses $L = 3$, $d = 64$, $T = 1$, and RK4 with four vector-field evaluations. Propagation is evaluated by the same sparse recursion as Equation (7), requiring $\mathcal{O}(L|E|d)$ work per vector-field evaluation and $\mathcal{O}(\mathrm{NFE}\, L|E|d)$ per ODE solve without materialising graph powers.

Across the same ten seeds, PIANO improves test accuracy over GCN by 0.0107 and over GPR-GNN by 0.0061, with paired Wilcoxon $p = 0.004$ and $p = 0.027$, respectively. It requires more inference time and memory than these discrete baselines, so this experiment demonstrates feasible full-graph continuous multi-order propagation rather than a runtime advantage.

Table 22: ogbn-arxiv accuracy and resource profile. Accuracy is mean ± SD over ten seeds for PIANO, GPR-GNN, and GCN, and over five seeds for GREAD. Runtime is full-graph inference.

Model	Test Acc.↑	Params	Infer. ms	Peak MB
GCN (Kipf & Welling, 2017)	$0.655_{\pm 0.002}$	10.9K	47	588
GPR-GNN (Chien et al., 2021)	$0.660_{\pm 0.003}$	10.9K	51	870
GREAD (Choi et al., 2023)	$0.573_{\pm 0.006}$	10.9K	119	1251
PIANO	$\mathbf{0.666}_{\pm 0.007}$	19.5K	100	1340

G.5 Topology-signal intervention

We test whether the trained scalar gate uses the closed-walk signature by intervening on five independently trained PIANO checkpoints. At inference, we compare the native signature with a fixed row permutation, its graph-wide mean, and removal of the structural term ac_i.

Table 23: Inference-time topology-signal intervention on ogbn-arxiv. The shuffled condition uses twenty fixed node permutations for each of five trained checkpoints.

Condition	Test Acc.↑	Δ vs. native
Native	$0.6655_{\pm 0.0057}$	–
Shuffled $\mathbf{d}^{(3)}_{\pi(i)}$	$0.6655_{\pm 0.0057}$	0.0000
Mean signature	$0.6655_{\pm 0.0057}$	0.0000
Structural term off	$0.6655_{\pm 0.0057}$	0.0000

All three interventions have negligible effect on test accuracy. Across the five checkpoints, $|a| \approx 0.02$, $\mathrm{std}_i(c_i) \approx 0.079$, $\mathrm{std}_i(\gamma_i) \approx 0.005$, and at least 99% of nodes satisfy $\gamma_i > 0.95$. Thus the topology statistic varies across nodes, but the learned structural coefficient is small and the gate is nearly fully open.

This does not contradict Proposition 1, which establishes that suitable parameters can produce topology-dependent updates. Rather, the trained ogbn-arxiv models choose not to exploit this channel. We therefore use ogbn-arxiv as evidence of static-graph transfer and scalability, while Springs provides the main learned-data evidence for selective topology-aware modulation.

H Extended Positioning and Related Work

H.1 Extended Related Work

Continuous-time graph models. Graph neural ODEs (Chen et al., 2018; Poli et al., 2019; Xhonneux et al., 2020) replace depth by integration time, with variants for traffic, reaction–diffusion, and physical systems (Fang et al., 2021; Choi et al., 2022; 2023; Luo et al., 2023; Huang et al., 2023; Jin et al., 2023; Wang et al., 2025). Most use one graph-interaction mechanism per vector-field evaluation: fixed diffusion, an integration-fixed learned graph, state-dependent one-hop edges, or adaptive interaction control. Integration can compose such local operators over time, whereas PIANO explicitly mixes orders $0, \ldots, L$ within each evaluation and reweights them from the evolving state.

Multi-hop polynomial filters. APPNP, GPR-GNN, and SIGN (Gasteiger et al., 2019; Chien et al., 2021; Frasca et al., 2020) are discrete layers with fixed or globally learned coefficients. PAN (Ma et al., 2020) uses a static path-length-weighted convolution related to subgraph centrality. PIANO instead places a state-conditioned expected finite-step propagator inside a continuous flow. PolyNSD (Borgi et al., 2025) also uses a high-order polynomial filter, but on a sheaf Laplacian with state-independent coefficients (Appendix H.3).

Spatiotemporal forecasting. Strong traffic GNNs (Yu et al., 2018; Wu et al., 2019; Bai et al., 2020; Wu et al., 2020) and radar/video models (Gao et al., 2022a;b; Zhang et al., 2023) use domain-

specific architectures; PIANO tests whether one compact adaptive continuous-time graph operator transfers across forecasting regimes.

H.2 On the "Order weights adapt?" Column

Table 1 asks whether relative weights over explicit propagation orders are recomputed from the evolving state. This differs from adapting edge weights, connectivity, interaction control, or temporal delay at fixed spatial order.

GRAM-ODE (Liu et al., 2023) uses state-dependent temporal-attention corrections, while spatial mixing remains fixed normalised-adjacency diffusion on a precomputed graph.

GRAND (Chamberlain et al., 2021) uses $\dot{\mathbf{X}} = (\mathbf{A}(\mathbf{X}) - I)\mathbf{X}$. GRAND-l holds $\mathbf{A}$ fixed during integration, whereas GRAND-nl recomputes it from the current state. Both reweight existing one-hop edges; broader propagation arises through integration rather than an explicit mixture of spatial orders.

GG-ODE (Huang et al., 2023) recomputes the interaction graph within the ODE from decoded latent positions, but messages remain local to the current radius-defined neighbours. It therefore adapts connectivity, not a distribution over explicit propagation orders.

CSG-ODE (Wang et al., 2025) uses GNN-generated state-dependent interaction control and an adaptive mixed graph during latent-state inference. Neither mechanism parameterizes a walk-length distribution over powers of one base adjacency.

What is distinctive about PIANO. PIANO re-predicts $\boldsymbol{\alpha}(\mathbf{H}, t)$ over explicit orders $0, \ldots, L$ at every vector-field evaluation. Its instantaneous support remains bounded by L, but its effective propagation scale changes with the state.

H.3 Relation to PolyNSD (Concurrent Work)

PolyNSD (Borgi et al., 2025) is a concurrent preprint using a high-order polynomial filter on the normalised sheaf Laplacian for heterophilic node classification. Its coefficients are globally learned and fixed during propagation. PIANO instead operates on the normalised graph adjacency inside a continuous vector field and recomputes simplex weights from the evolving state. The methods therefore differ in operator (sheaf Laplacian vs. adjacency), adaptivity (fixed global vs. state-conditioned coefficients), and evaluation regime.