CHRONOSHEAF: JOINT SPATIO-TEMPORAL SPECTRAL FILTERING ON CELLULAR SHEAVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Sheaf neural networks generalize graph neural networks by assigning each edge a learned linear restriction map that specifies how the local representations of adjacent nodes relate to one another, thereby mitigating oversmoothing. Existing sheaf architectures, however, define the sheaf Laplacian as an instantaneous spatial operator and therefore cannot explicitly model how signals evolve over time on a graph. We first introduce a spectral sheaf Laplacian, defined jointly over space and time, such that a single operator regulates signal propagation across both the spatial graph and the temporal domain. Directly extending this operator's receptive field in space or time would require increasingly complex sheaf structures. Instead, we propose a spectral filter that expands the effective receptive field along both dimensions without enlarging the underlying sheaf. Building on this operator, we introduce **ChronoSheaf**, a spatio-temporal sheaf network with two key components. First, Transitional Restriction Maps (TRM) decompose each node's restriction map into a spatial term and two temporal frames that independently control its relation to neighboring nodes at the preceding and following timestep, enabling asymmetric temporal influence. Second, the Dual Spectral Filter (DSF) applies a single bounded-degree polynomial filter with an adaptively estimated spectral radius to the joint spatio-temporal sheaf operator. We validate ChronoSheaf on a diverse set of spatio-temporal forecasting benchmarks, where it outperforms state-of-the-art baselines. The code is available at: `https://anonymous.4open.science/r/ChronoSheaf-1D7D/`.

1 INTRODUCTION

Sheaf neural networks generalize graph neural networks by assigning a vector space (stalk) to each node and edge and learning linear restriction maps between them, which lets them model heterogeneous relationships. The ordinary graph Laplacian is recovered as a special case of the constant sheaf, where every stalk is identical, and every restriction map is the identity. By letting restriction maps be learned and non-trivial, Bodnar et al. (2022) showed that a node's local frame need not agree with its neighbors' outright, giving sheaf neural networks a proven advantage over conventional graph neural networks on heterophilic graphs and against oversmoothing. In every such construction, however, the sheaf is defined purely over the spatial graph. Even when restriction maps are learned from node features (Bodnar et al., 2022), the base space contains no notion of time, so the sheaf Laplacian only learns from nodes observed at the same instant, and the associated diffusion relaxes toward a static equilibrium, the space of harmonic cochains (Hansen & Ghrist, 2019). As a consequence, increasing depth or filter order does not let a spatial sheaf Laplacian reach across time. Hence, temporal context must come from a separate module outside the sheaf that does not benefit from its learned local frames.

This restriction has a direct architectural consequence for spatio-temporal graph learning. The dominant paradigm thereby depends on conventional GNN architectures which are prone to oversmoothing and compose two separate operators, a spatial graph operator, and a recurrent, convolutional, or attention-based module for temporal dependence (Li et al., 2018b; Yu et al., 2018; Wu et al., 2019; Bai et al., 2020; Zheng et al., 2020; Wu et al., 2020). Marisca et al. (2025) recently showed this composition has a measurable cost; because causal temporal convolutions propagate information only forward, deeper stacks accumulate disproportionately more paths from early timesteps

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.

than from recent ones, so convolutional spatio-temporal GNNs counterintuitively favor temporally distant information over nearby steps.

We address this gap by defining a single spectral sheaf operator jointly over space and time, so that one filtering recursion controls both spatial and temporal dependence. This operator extends the sheaf domain from the spatial graph alone to a space-time graph, and edges connect spatial neighbors at a shared time as well as each node to its neighbors at the adjacent timestep. The network architecture built around this operator is ChronoSheaf, consisting of two components. Transitional restriction maps decompose each node's restriction map into a learned spatial frame that determines how spatial neighbors are compared, plus two further frames, F^{in} and F^{out}, that independently weight a node's interaction with its neighbors at the preceding and following timestep. The dual spectral filter then applies a single spectral filter to this joint operator, giving ChronoSheaf a receptive field that grows jointly in graph hops and temporal lag from one spectral recursion (Figure 1). Our contributions are as follows:

- We introduce a new class of sheaf GNNs, called ChronoSheaf, in which we fundamentally rethink the three core components of sheaf GNNs:
  1. The *base space* of the sheaf is a joint space-time graph, so that each node's stalk is compared not only with its spatial neighbors but also with its own state and its neighbors' states at adjacent timesteps.
  2. The *restriction maps* allow each node to respond to its neighbors' past and future states with distinct strengths to capture asymmetric temporal influence.
  3. The *sheaf Laplacian*'s spatial hops and temporal lags are controlled by a spectral filter that acts on the Laplacian's frequency spectrum without needing eigendecomposition. We further provide a theoretical analysis of the resulting operator, characterizing its spectral properties and its joint spatio-temporal receptive field.
- We conduct extensive experiments on six real-world datasets spanning diverse domains. The results show that ChronoSheaf resists oversmoothing, captures long-range dependencies more effectively, and outperforms state-of-the-art methods by a notable margin.

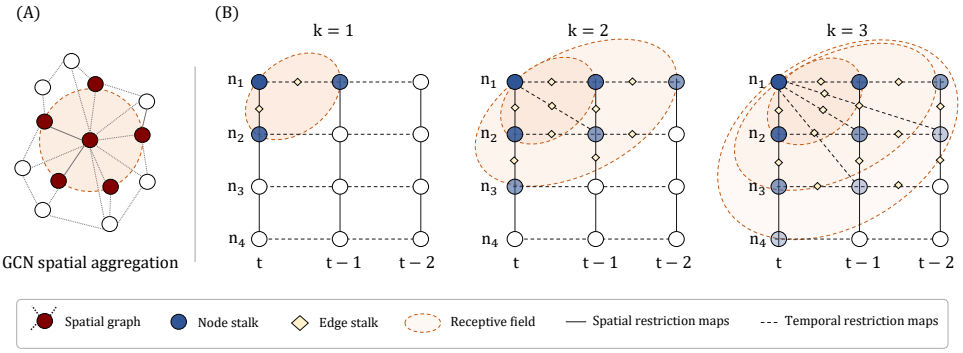


Figure 1: (A) State-of-the-art graph forecasting models aggregate spatial information from neighboring nodes and enlarge their receptive field by stacking additional layers. (B) Our model performs spectral filtering directly on joint space-time cellular sheaves. Higher-order polynomial filters extend the receptive field to larger neighborhoods and capture deeper temporal context.

2 Related Work

Spectral graph convolution and its limits. Spectral filtering on graphs originates in Bruna et al. (2013) and is made scalable via polynomial truncation in Defferrard et al. (2016), whose first-order simplification underlies the method proposed in Kipf & Welling (2017). Veličković et al. (2017) replace the fixed spectral filter with data-dependent attention weights. Known failure modes of this design, oversmoothing (Rusch et al., 2023; Li et al., 2018a) and oversquashing (Alon & Yahav, 2021), motivate architectures that enrich what a node can represent. This is the main gap targeted by sheaf neural architectures.

Sheaf neural networks. Hansen & Ghrist (2019) generalize the graph Laplacian to a Hodge Laplacian over a cellular sheaf using linear restriction maps defined between per-node stalks. Bodnar et al. (2022) instantiate this as a neural network, with a fully parametric diffusion architecture and provable control over Dirichlet-energy decay, showing a learned non-trivial sheaf mitigates oversmoothing where the trivial sheaf cannot. Follow-up work restricts the space of restriction maps: with orthogonal Laplacians (Barbero et al., 2022) and $O(d)$-bundle maps with continuous transport (Bamberger et al., 2025). Most recently, Hajij et al. (2025) subsume sheaf diffusion, GNNs, and attention into a single copresheaf-on-combinatorial-complex formalism. Ribeiro et al. (2026) define cellular sheaves over directed graphs, allowing the diffusion operator to selectively listen to distant nodes and mitigate oversquashing. Fiorini et al. (2026) also propose a directed cellular sheaf and its associated directed sheaf Laplacian, which encodes both graph topology and edge orientation. These extensions, however, address spatial connectivity specifically. From a different angle, we address two questions: how time can be encoded within the sheaf and how to learn the sheaf Laplacian's spectrum.

Spatio-temporal graph learning. The dominant paradigm combines a spatial graph operator with a separate temporal mechanism, including recurrent units (Li et al., 2018b), gated temporal convolutions (Yu et al., 2018; Wu et al., 2019), and self-attention (Zheng et al., 2020). Other approaches enrich the spatial or dynamical component through adaptive graph learning (Bai et al., 2020), zigzag-persistence representations (Chen et al., 2021), neural ODEs (Fang et al., 2021), or controlled differential equations (Choi et al., 2022). Recent work continues to push the adaptive graph side toward efficiency: Ma et al. (2025) replace quadratic-cost adaptive graph learning with a linear, mixture-of-experts formulation. Across these methods, spatial and temporal dependence are usually modeled by two distinct operators, composed architecturally, so receptive fields grow additively with depth. A few works operate on a joint space-time domain; STSGCN (Song et al., 2020) connects each node to itself and its neighbors at adjacent timesteps within a localized space-time graph and graph-time convolutional networks (Sabbaqi & Isufi, 2023) filter over parametric product graphs. These methods demonstrate the value of a joint domain, but they weight its edges with scalars: every node shares a single feature space with its past, its future, and its neighbors, which is precisely the reason behind oversmoothing.

3 ChronoSheaf

This section describes ChronoSheaf. Section 3.1 establishes the sheaf-theoretic and spectral preliminaries. Section 3.2 introduces the transitional restriction maps, the learned frames that control the space-time domain. Section 3.3 assembles these frames into the three-part sheaf Laplacian and proves it is a real, symmetric, positive semi-definite operator. Section 3.4 introduces the dual spectral filter, with the adaptive spectral-radius bound. Finally, Section 3.5 describes the proposed layer and the full network.

3.1 Preliminaries

Spatial graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, undirected graph with $|\mathcal{V}| = N$ nodes, and adjacency matrix $A \in \{0,1\}^{N\times N}$. We work with the symmetrically normalized adjacency $\tilde{A} = D^{-1/2}AD^{-1/2}$, where $D = \mathrm{diag}(A\mathbf{1})$, so that the normalized graph Laplacian is $L_{\mathrm{sym}} = I - \tilde{A}$.

Cellular sheaves. A cellular sheaf $\mathcal{F}$ on $\mathcal{G}$ assigns to each node v a stalk $\mathcal{F}(v) \cong \mathbb{R}^r$ and, for each incident node-edge pair $v \triangleleft e$, a linear restriction map $\mathcal{F}_{v\triangleleft e} : \mathcal{F}(v) \to \mathcal{F}(e)$ into a shared edge stalk (Hansen & Ghrist, 2019). The associated sheaf Laplacian

$$(L_{\mathcal{F}}x)_v = \sum_{e=(u,v)\in\mathcal{E}} \mathcal{F}^{\top}_{v\trianglelefteq e}\big(\mathcal{F}_{v\trianglelefteq e}x_v - \mathcal{F}_{u\trianglelefteq e}x_u\big), \tag{1}$$

is a symmetric, positive semi-definite operator whose quadratic form $x^{\top}L_{\mathcal{F}}x = \sum_e \|\mathcal{F}_{v\triangleleft e}x_v - \mathcal{F}_{u\triangleleft e}x_u\|_2^2$ measures disagreement between adjacent nodes after each node's local frame is applied (Bodnar et al., 2022). Setting every restriction map to the identity on $\mathbb{R}^1$ recovers the ordinary, unnormalized combinatorial graph Laplacian $D - A$. The sheaf with identity restriction maps is called the trivial sheaf.

Spectral graph theory. For a symmetric operator L with eigendecomposition $L = U\Lambda U^\top$, a spectral graph filter applies a function $g(\Lambda)$ of the eigenvalues before returning to the node domain, $g(L)x = Ug(\Lambda)U^\top x$.

In this work, we do not compute this eigendecomposition explicitly, because of the computationally expensive overhead it adds to the network. We use its polynomial (Chebyshev) approximation (Defferrard et al., 2016): with $\tilde{L}$ rescaled to $[-1, 1]$, the order-K Chebyshev filter is

$$g_\theta(L)x = \sum_{k=0}^{K} \theta_k \, T_k(\tilde{L})\, x, \qquad T_0(z) = 1,\ T_1(z) = z,\ T_k(z) = 2zT_{k-1}(z) - T_{k-2}(z), \tag{2}$$

which is exactly K-localized (depends on nodes within K hops) and requires no eigendecomposition.

Joint space-time sheaf domain. ChronoSheaf's central construction is a cellular sheaf defined not over $\mathcal{G}$ alone, but over a joint space-time graph $\mathcal{G}^\star = (\mathcal{V}^\star, \mathcal{E}^\star)$. Nodes are pairs (n, t) for $n \in \mathcal{V}$ and $t \in \{1, \ldots, P\}$, where P is the number of observed timesteps. $\mathcal{E}^\star$ contains a spatial edge between (n, t) and (m, t) whenever $(n, m) \in \mathcal{E}$, and a temporal edge between (n, t) and $(m, t-1)$ for every m that is either n itself (a self-loop) or a spatial neighbor of n. Sections 3.2- 3.3 add to $\mathcal{G}^\star$ node-homogeneous frames and assemble its sheaf Laplacian; Section 3.4 shows that a single Chebyshev recursion over this one operator is what lets ChronoSheaf's receptive field grow jointly in graph hops and temporal lag.

Forecasting Task. The model takes input signal $x \in \mathbb{R}^{N \times P \times d_{\text{in}}}$, obtained from the raw observations after standard preprocessing (segmentation into P timesteps of d_{in} channels). Given a history of such signals, the goal is to learn a mapping F that predicts the next H steps of the underlying scalar measurement at each node.

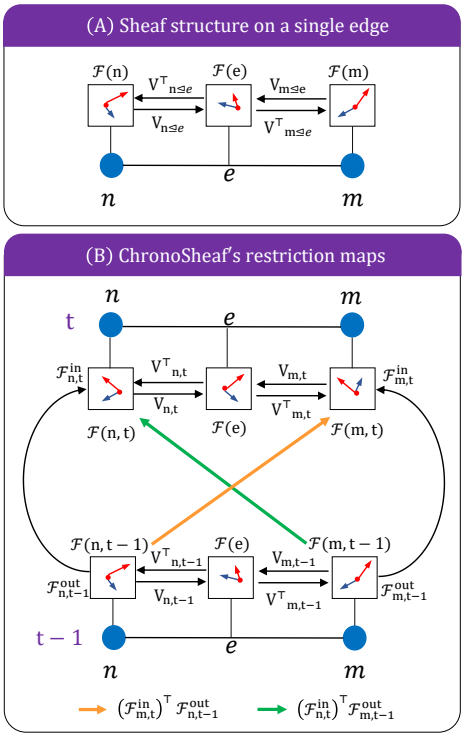


Figure 2: (A) Neural sheaf structure Bodnar et al. (2022). (B) ChronoSheaf's joint space-time sheaf structure.

3.2 Transitional Restriction Maps (TRM)

ChronoSheaf's first component assigns learned frames to the joint graph $\mathcal{G}^\star$. Let $B \in \mathbb{R}^{3 \times N \times r \times d}$ be a learnable base tensor, holding one candidate frame of each of three kinds (spatial, outgoing-temporal, incoming-temporal frames) per node. At each timestep, a learned gate rescales the three candidates for node n as a function of the layer input $h_{n,t} \in \mathbb{R}^d$:

$$g_{n,t} = \mathbf{1} + \tfrac{1}{2} \tanh(W_g h_{n,t}), \qquad W_g \in \mathbb{R}^{3 \times d}, \tag{3}$$

$$V_{n,t} = g_{n,t,1} B_{1,n}, \qquad F^{\text{out}}_{n,t} = g_{n,t,2} B_{2,n}, \qquad F^{\text{in}}_{n,t} = g_{n,t,3} B_{3,n}, \tag{4}$$

so that $V_{n,t}, F^{\text{in}}_{n,t}, F^{\text{out}}_{n,t} \in \mathbb{R}^{r \times d}$ are node n's three restriction maps at time t. Since $\tanh(\cdot) \in (-1, 1)$, each component of $g_{n,t} \in \mathbb{R}^3$ lies in $(0.5, 1.5)$ to bound every frame's scale for numerical stability.

We call F^{in} and F^{out} *transitional restriction maps*. $V_{n,t}$ plays the role of an ordinary spatial sheaf frame, learning how node n's representation is compared with its spatial neighbors at the same time. $F^{\text{out}}_{n,t}$ and $F^{\text{in}}_{n,t}$ learn how n's representation is compared across a single-step transition in time: $F^{\text{out}}_{n,t}$ sets how strongly n projects its current signal onto the following timestep, and $F^{\text{in}}_{n,t}$ sets how strongly n accepts a neighbor's projection from the preceding one. Because these are two independently learned maps, a node can weight its sensitivity to the past differently from its sensitivity to the future.

3.3 THE CHRONOSHEAF LAPLACIAN

Spatial term. We instantiate the sheaf Laplacian of Eq. 1 to node-homogeneous restriction maps, in which every spatial edge incident to (n,t) uses the same spatial frame, $\mathcal{F}_{(n,t)\trianglelefteq e} = V_{n,t}$, and weight each edge by the normalized adjacency $\tilde{A}$:

$$(L_{sp}x)_{n,t} = \sum_m \tilde{A}(n,m) V_{n,t}^\top \left(V_{n,t}x_{n,t} - V_{m,t}x_{m,t}\right). \tag{5}$$

Because $V_{n,t} \in \mathbb{R}^{r\times d}$, the restriction maps project the d-dimensional feature stalks into a shared r-dimensional edge space before measuring disagreement.

Temporal terms. Let $\hat{A} = D^{-1/2}(A+I)D^{-1/2}$ be the self-loop-augmented normalized adjacency, and let $\hat{d}(n) := \sum_m \hat{A}(n,m)$ denote its weighted row sum at node n. The two transitional terms are:

$$(L_{in}x)_{n,t} = \hat{d}(n)(F_{n,t}^{in})^\top F_{n,t}^{in} x_{n,t} - \sum_m \hat{A}(n,m)(F_{n,t}^{in})^\top F_{m,t-1}^{out} x_{m,t-1}, \tag{6}$$

$$(L_{out}x)_{n,t} = \hat{d}(n)(F_{n,t}^{out})^\top F_{n,t}^{out} x_{n,t} - \sum_m \hat{A}(n,m)(F_{n,t}^{out})^\top F_{m,t+1}^{in} x_{m,t+1}, \tag{7}$$

each masked to zero at the first and last timestep, where no earlier or later patch exists. The full Laplacian is then:

$$L = L_{sp} + L_{in} + L_{out}, \tag{8}$$

a linear operator on $\mathbb{R}^{N\times P\times d}$ that we equivalently view as a block matrix with one $d\times d$ block $L_{(n,t),(m,s)}$ per pair of joint-graph nodes, acting on $x_{m,s} \in \mathbb{R}^d$ (Figure 2).

Proposition 1. *L is a real, symmetric operator on $\mathcal{G}^\star$: $L_{(n,t),(m,s)}^\top = L_{(m,s),(n,t)}$ for every pair of joint-graph nodes, even though $F^{\text{in}} \neq F^{\text{out}}$ in general.*

Proof. Every diagonal block is a sum of Gram matrices, hence symmetric. For $s = t$ (spatial neighbors), Eq. 5 gives $L_{(n,t),(m,t)} = -\tilde{A}(n,m)V_{n,t}^\top V_{m,t}$; since $(V_{n,t}^\top V_{m,t})^\top = V_{m,t}^\top V_{n,t}$ and $\tilde{A}(n,m) = \tilde{A}(m,n)$, this equals $L_{(m,t),(n,t)}^\top$. For $s = t-1$, Eq. 6 gives $L_{(n,t),(m,t-1)} = -\hat{A}(n,m)(F_{n,t}^{\text{in}})^\top F_{m,t-1}^{\text{out}}$, while Eq. 7 evaluated at node m, time $t-1$, gives $L_{(m,t-1),(n,t)} = -\hat{A}(m,n)(F_{m,t-1}^{\text{out}})^\top F_{n,t}^{\text{in}}$. Transposing the first block, $\big(-\hat{A}(n,m)(F_{n,t}^{\text{in}})^\top F_{m,t-1}^{\text{out}}\big)^\top = -\hat{A}(n,m)(F_{m,t-1}^{\text{out}})^\top F_{n,t}^{\text{in}}$, which equals $L_{(m,t-1),(n,t)}$ since $\hat{A}(n,m) = \hat{A}(m,n)$. Symmetry holds block by block, and therefore for L as a whole; it follows from L_{in}'s backward-looking mapping and L_{out}'s forward-looking mapping being adjoint to one another. □

Proposition 2 (Positive semi-definiteness)**.** *$L = L_{\text{sp}} + L_{\text{in}} + L_{\text{out}}$ is positive semi-definite.*

Proof. *Spatial term.* Fix a patch t and let $y_{n,t} := V_{n,t}x_{n,t} \in \mathbb{R}^r$, with components $y_{n,t,k}$, $k = 1,\dots,r$. Expanding Eq. 5,

$$x^\top L_{\text{sp}} x = \sum_{t,k} y_{\cdot,t,k}^\top \big(\text{diag}(\tilde{A}\mathbf{1}) - \tilde{A}\big) y_{\cdot,t,k} = \sum_{t,k} \tfrac{1}{2} \sum_{n,m} \tilde{A}(n,m)\big(y_{n,t,k} - y_{m,t,k}\big)^2 \geq 0,$$

using that $\text{diag}(W\mathbf{1}) - W$ is positive semi-definite for any symmetric $W \geq 0$.

Temporal terms. Let $u_{n,t} := F_{n,t}^{\text{in}} x_{n,t}$ and $w_{n,t} := F_{n,t}^{\text{out}} x_{n,t}$, both in $\mathbb{R}^r$. Expanding Eq. 6-Eq.7,

$$x^\top L_{\text{in}} x = \sum_{n,t} \hat{d}(n)\|u_{n,t}\|^2 - \sum_{n,t,m} \hat{A}(n,m)\, u_{n,t}^\top w_{m,t-1},$$

$$x^\top L_{\text{out}} x = \sum_{n,t} \hat{d}(n)\|w_{n,t}\|^2 - \sum_{n,t,m} \hat{A}(n,m)\, w_{n,t}^\top u_{m,t+1}.$$

Reindexing the second cross term ($n \leftrightarrow m$, $t \mapsto t-1$) and using $\hat{A}(n,m) = \hat{A}(m,n)$ shows the two cross terms are equal; reindexing $\sum_{n,t} \hat{d}(n)\|w_{n,t}\|^2$ the same way aligns every diagonal term with a cross term at the same (n,t,m). Completing the square gives the exact identity

$$x^\top (L_{in} + L_{out}) x = \sum_{n,t,m} \hat{A}(n,m)\|F_{n,t}^{in} x_{n,t} - F_{m,t-1}^{out} x_{m,t-1}\|_2^2,$$

the sum ranging over all valid (n, t, m) with $t \geq 2$; every term is nonnegative since $\hat{A}(n, m) \geq 0$. Summing the two nonnegative quadratic forms gives $x^\top Lx \geq 0$ for all x. Figure 3 illustrates how a node's past affect its neighbor's present. □

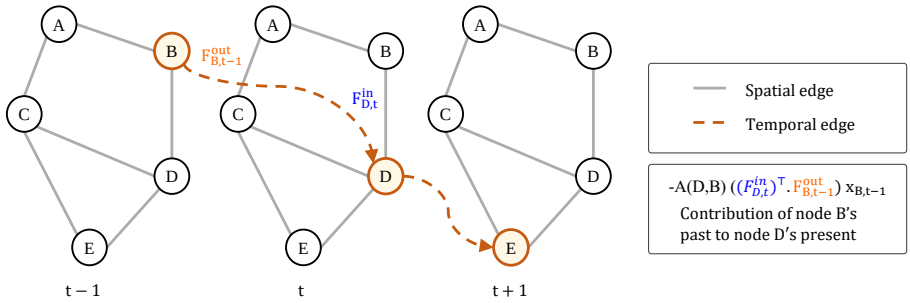


Figure 3: Node B's signal at $t-1$ contributes $-A(D,B)(F^{\text{in}}_{D,t})^\top F^{\text{out}}_{B,t-1}\, x_{B,t-1}$ to $(Lx)_{D,t}$, transporting B's signal from its outgoing frame into D's incoming frame. The analogous term one step later carries D's present into E's future.

3.4 Dual Spectral Filter (DSF)

ChronoSheaf's second component filters L with the Chebyshev polynomial of Section 3.1, using per-channel coefficients $\theta_{c,0:K}$: the output is $\sum_k \big(T_k(\tilde{L})x\big)\,\Theta_k$, where $\Theta_k = \operatorname{diag}(\theta_{1,k}, \ldots, \theta_{d,k})$ scales the d channels, so one order-K recursion over L filters jointly across the spatial and temporal edges of $\mathcal{E}^\star$. We call this the *dual spectral filter*: a single Chebyshev recursion that is simultaneously a spatial and a temporal filter, since Propositions 1 and 2 let L carry both kinds of edges as one real, symmetric, positive semi-definite operator.

Applying this filter requires an estimate of the sheaf Laplacian's largest eigenvalue $\lambda_{\max}(L)$. Because $V, F^{\text{in}}, F^{\text{out}}$ are learned and unconstrained in norm, no fixed bound holds for L in general, and an eigendecomposition at every forward pass is too costly. We instead adaptively bound $\lambda_{\max}(L)$: for a real symmetric block matrix with $d \times d$ blocks $L_{(n,t),(m,s)}$,

$$\lambda_{\max}(L) \;\leq\; \max_{(n,t)} \Big[\big\| L_{(n,t),(n,t)} \big\|_2 + \sum_{(m,s)\neq(n,t)} \big\| L_{(n,t),(m,s)} \big\|_2 \Big], \tag{9}$$

where $\|\cdot\|_2$ is the spectral norm. This is a generalization of the known scalar Gershgorin bound (entrywise absolute value) to the $d \times d$ blocks of L. We bound Eq. 9 directly from the frames of Section 3.2, using the triangle and submultiplicative inequalities for the spectral norm ($\|A+B\|_2 \leq \|A\|_2 + \|B\|_2$, $\|AB\|_2 \leq \|A\|_2\|B\|_2$, $\|A^\top A\|_2 = \|A\|_2^2$):

$$\begin{aligned} \hat{\lambda}_{\max} := \max_{(n,t)} \Big[& s_n \|V_{n,t}\|_2^2 + \hat{d}(n)\big(\|F^{\text{in}}_{n,t}\|_2^2 + \|F^{\text{out}}_{n,t}\|_2^2\big) \\ & + \sum_m \tilde{A}(n,m)\|V_{n,t}\|_2\|V_{m,t}\|_2 \\ & + \sum_m \hat{A}(n,m)\big(\|F^{\text{in}}_{n,t}\|_2\|F^{\text{out}}_{m,t-1}\|_2 + \|F^{\text{out}}_{n,t}\|_2\|F^{\text{in}}_{m,t+1}\|_2\big)\Big], \end{aligned} \tag{10}$$

with $s_n = \sum_m \tilde{A}(n,m)$. Every quantity in Eq. 10 is a spectral norm of one $r \times d$ frame, so $\hat{\lambda}_{\max}$ is tracked once per forward pass at negligible cost and detached from the computation graph (no backpropagation through $\hat{\lambda}_{\max}$).

Proposition 3 (Receptive field)**.** *The output at (n,t) after M stacked ChronoSheaf layers depends on the input at (m,s) with $d_{\mathcal{G}}(n,m) + |t-s| \leq 2MK$, where $d_{\mathcal{G}}$ is graph-hop distance in $\mathcal{G}$ and t, s index patches.*

Proof. Every nonzero block of $L = L_{\text{sp}} + L_{\text{in}} + L_{\text{out}}$ connects (n,t) to some (m,s) with $d_{\mathcal{G}}(n,m) + |t-s| \leq 2$ because L_{sp} contributes same-patch spatial neighbors ($d_{\mathcal{G}}(n,m) \leq 1$, $|t-s| = 0$); L_{in} and L_{out} each contribute a self term one patch earlier or later ($d_{\mathcal{G}}(n,m) = 0$, $|t-s| = 1$) and a

spatial-neighbor term one patch earlier or later ($d_{\mathcal{G}}(n,m) = 1$, $|t-s| = 1$, since $\hat{A}(n,m) \neq 0$ whenever $m = n$ or m is a spatial neighbor of n).

The order-K Chebyshev recursion of the dual spectral filter is a degree-K polynomial in L, so one ChronoSheaf layer's output at (n,t) depends on (m,s) only if $d_{\mathcal{G}}(n,m)+|t-s| \leq 2K$. Composing M layers sums this bound M times. □

3.5 ChronoSheaf Layer

One ChronoSheaf layer applies the dual spectral filter of Section 3.4 to the operator L of Section 3.3, wrapped in a residual connection,

$$\text{ChronoSheaf}(h) = \text{LayerNorm}\big(h + \text{Dropout}(W\, g_\theta(L)\, h)\big), \qquad W \in \mathbb{R}^{d\times d}, \tag{11}$$

with the frames $V, F^{\text{in}}, F^{\text{out}}$ (hence L) and the coefficients $\theta_{c,0:K}$ learned independently at every layer. The full network stacks M such layers. Writing x for the patchified, preprocessed input and $e \in \mathbb{R}^{N\times d}$ for a learned per-node embedding broadcast across the batch and every patch, the first layer's input is $h^{(0)} = \text{PE}\big(xW_{\text{in}} + e\big)$, where W_{in} projects into the model dimension d and PE adds a fixed sinusoidal encoding along the patch axis to mark the order in which the temporal edges of $\mathcal{G}^\star$ connect. Each of the M layers then maps $h^{(l-1)} \mapsto h^{(l)}$ as in Eq. 11.

4 Experiments

We evaluate the performance of ChronoSheaf on several spatio-temporal forecasting benchmarks. In Section 4.1, we assess ChronoSheaf's ability to mitigate oversmoothing in spatio-temporal learning tasks. In Section 4.2, we include the results of traffic, weather, air quality forecasting on six benchmark datasets compared to SOTA baselines. Section 4.3 examines the effect of spectral filtering on the space-time sheaf Laplacian and how different values of polynomial orders affect performance. Finally, in Section 4.4, we examine the network's ability to retain local representations given heterogeneous features from another modality. Extended results and complete dataset details are provided in the Appendices.

4.1 Oversmoothing Analysis

We examine whether TRM preserves local representations as the network deepens. On PEMS08, we stack $M \in \{1,\dots,8\}$ layers and track both test Root Mean Square Error (RMSE) and the normalized Dirichlet energy:

$$E(\mathbf{H}) = \frac{N}{|\mathcal{E}|} \cdot \frac{\displaystyle\sum_{(i,j)\in\mathcal{E}} \sum_{t=1}^{T} \big\|\mathbf{h}_{i,t} - \mathbf{h}_{j,t}\big\|_2^2}{\displaystyle\sum_{i=1}^{N}\sum_{t=1}^{T} \big\|\mathbf{h}_{i,t}\big\|_2^2}, \tag{12}$$

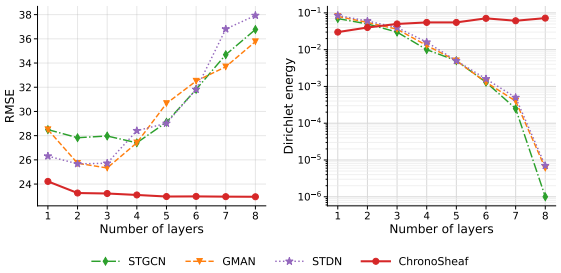


Figure 4: Test RMSE (left) and Dirichlet energy (right, log scale) versus network depth. Baseline energy collapses with depth, while ChronoSheaf's remains stable.

of the hidden node representations; energy that decays toward zero means node features are becoming indistinguishable. We compare against three representative baselines: STGCN (convolution-based), GMAN (attention-based), and STDN (the current state of the art in spatio-temporal forecasting). Results are shown in Figure 4.

All three baselines degrade sharply with depth, tracking a Dirichlet energy that collapses by four to six orders of magnitude by the eighth layer. This is exactly the known behavior of oversmoothing under repeatedly applied propagation discussed in Li et al. (2019) and Arroyo et al. (2026). ChronoSheaf's energy stays within roughly one order of magnitude across the same range, and its RMSE saturates as depth increases.

Table 1: Traffic forecasting results on LargeST (SD and CA), PEMS04, and PEMS08, averaged over 12 forecasting steps. Best results are in bold and second-best are underlined. "OOM" denotes out of memory.

Model	LargeST (SD)			LargeST (CA)			PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN Yu et al. (2018)	19.67	34.14	13.86	21.33	36.39	16.53	22.70	35.55	<u>14.59</u>	18.02	27.83	11.40
DCRNN Li et al. (2018b)	21.03	33.37	14.13	21.87	34.41	17.06	24.70	38.12	17.12	17.86	27.83	11.45
GW-Net Wu et al. (2019)	<u>17.74</u>	<u>29.62</u>	<u>11.88</u>	21.72	34.20	17.40	25.45	39.70	17.29	19.13	31.05	12.68
GMAN Zheng et al. (2020)	19.63	30.01	12.20	OOM	OOM	OOM	20.23	<u>32.17</u>	17.06	16.47	25.72	<u>10.48</u>
ASTGCN Guo et al. (2019)	23.70	37.63	15.65	OOM	OOM	OOM	22.80	35.82	16.56	18.63	28.27	13.08
CrossGNN Huang et al. (2023)	23.09	39.04	15.87	26.16	41.25	20.25	25.11	40.32	20.88	21.91	35.69	16.73
FITS Xu et al. (2024)	21.34	38.01	14.36	20.12	35.22	15.81	20.02	33.31	16.49	<u>15.43</u>	26.34	10.75
SGP Cini et al. (2023)	20.25	35.06	12.33	OOM	OOM	OOM	21.90	34.27	16.15	18.06	27.33	13.55
CITRUS Einizade et al. (2024)	18.22	30.67	12.23	<u>19.66</u>	33.50	15.72	22.64	35.08	16.83	17.89	27.81	11.21
SparseTSF Lin et al. (2024)	25.90	46.77	18.30	23.75	43.03	18.87	22.24	38.65	17.25	16.61	30.84	11.54
TimeBase Huang et al. (2025)	19.15	32.21	13.22	20.23	<u>32.43</u>	<u>14.42</u>	23.21	41.59	16.18	17.52	35.99	11.98
STDN Cao et al. (2025)	19.01	31.25	14.10	OOM	OOM	OOM	**19.15**	34.37	23.51	18.26	<u>25.67</u>	20.93
ChronoSheaf	**16.42**	**28.33**	**10.72**	**17.52**	**31.66**	**14.22**	<u>19.95</u>	**29.11**	**13.11**	**14.32**	**23.26**	**10.21**

4.2 Spatio-temporal Forecasting

Setup. For traffic forecasting, we use PEMS04 and PEMS08 (Song et al., 2020), in addition to the San Diego (SD) and full California (CA) subsets of LargeST (Liu et al., 2023). All four datasets consist of traffic flow readings from loop detectors aggregated into 5-minute intervals, with sensors as nodes and edges derived from road-network connectivity. Graph sizes vary from 170 to 8,600 nodes. Following the experimental setup of Liu et al. (2023), we use 12 historical timesteps as input to predict the next 12 steps. Further details and hyperparameter configurations are provided in Appendix C. Additionally, we use the Molene dataset (Girault, 2015) for weather forecasting, which contains hourly temperature readings from 32 stations in Brittany, France, with 10 historical timesteps as input to predict the next 5 steps. We further evaluate ChronoSheaf on the Air Quality dataset (Microsoft Research, 2026), which contains hourly $PM_{2.5}$ concentration readings from 437 monitoring stations across 43 Chinese cities. The input spans 48 hours and predicts the next 48 hours. Full dataset details are provided in Appendix C.

Baselines. We compare ChronoSheaf to established GCN/RNN methods (STGCN (Yu et al., 2018), DCRNN (Li et al., 2018b)), adaptive graph learning (Graph WaveNet (Wu et al., 2019), CrossGNN (Huang et al., 2023)), attention-based models (ASTGCN (Guo et al., 2019), GMAN (Zheng et al., 2020)), scalable architectures (SGP (Cini et al., 2023), SparseTSF (Lin et al., 2024), FITS (Xu et al., 2024)), spectral methods (CITRUS (Einizade et al., 2024)), and the most recent decomposition-based approaches (TimeBase (Huang et al., 2025), STDN (Cao et al., 2025)). Note on comparison with other Sheaf-based methods: to the best of our knowledge, there is currently no published literature on sheaf-based methods for spatio-temporal modeling; all identified sheaf networks operate exclusively in the spatial domain. Therefore, direct comparisons with analogous sheaf models are not applicable.

Results. The reported results in Table 1 for traffic forecasting are in Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), averaged over 12 forecasting steps. On the four datasets, ChronoSheaf achieves the best result in 11 of the 12 reported dataset–metric combinations. Table 2 reports MAE for weather and air quality forecasting across multiple prediction horizons. ChronoSheaf's advantage over the baselines is most pronounced at longer horizons, which is consistent with its ability to capture extended temporal context.

4.3 Spectral Filtering Analysis

We vary the order k of the polynomial filter on the space-time sheaf Laplacian, which sets the number of hops each layer aggregates over on the PEMS08 dataset (Table 3). Accuracy improves sharply up to $k = 4$ and then saturates. Training time grows almost linearly with k, as each order adds one Laplacian multiplication. It should be noted that the spectral filtering approach here inherits the same advantages of the sheaf Laplacian in being more expressive. We therefore select k per dataset to balance predictive accuracy against computational cost, taking into account both graph size and the marginal gain from higher orders. Accordingly, we set $k = 2$ for the LargeST dataset and $k = 4$ for all other datasets.

Table 2: Weather (Molene) and air quality ($PM_{2.5}$) forecasting results in MAE across multiple prediction horizons. Best results are in bold and second-best are underlined.

Model	Molene (weather)					Air Quality ($PM_{2.5}$)		
	step-1	*step-2*	*step-3*	*step-4*	*step-5*	*1-12 hr*	*13-24 hr*	*25-48 hr*
STGCN (Yu et al., 2018)	1.066	1.552	1.933	2.225	2.698	22.77	31.59	35.59
DCRNN (Li et al., 2018b)	0.396	0.635	<u>0.907</u>	<u>1.164</u>	<u>1.419</u>	24.28	32.95	36.11
GW-Net (Wu et al., 2019)	0.855	1.124	1.455	1.865	2.104	22.42	30.29	34.73
GMAN (Zheng et al., 2020)	0.962	1.424	1.882	2.022	2.417	24.11	32.80	36.25
ASTGCN (Guo et al., 2019)	0.981	1.342	1.877	1.906	2.125	25.51	33.98	37.42
CrossGNN (Huang et al., 2023)	0.952	1.244	1.432	1.880	2.025	25.63	33.74	36.96
FITS (Xu et al., 2024)	0.813	1.036	1.37	1.671	1.964	21.81	31.17	34.56
SGP (Cini et al., 2023)	0.522	0.848	1.148	1.399	1.650	26.42	33.38	35.99
CITRUS (Einizade et al., 2024)	0.497	0.757	1.048	1.332	1.600	27.40	32.66	35.56
SparseTSF (Lin et al., 2024)	0.882	1.137	1.269	1.566	1.897	23.44	32.65	35.51
TimeBase (Huang et al., 2025)	0.574	0.821	1.036	1.478	1.988	**21.20**	**29.88**	<u>32.97</u>
STDN (Cao et al., 2025)	<u>0.352</u>	<u>0.621</u>	0.924	1.336	1.566	22.17	30.16	34.62
ChronoSheaf	**0.312**	**0.524**	**0.806**	**1.010**	**1.242**	<u>21.31</u>	<u>29.90</u>	**32.41**

Table 3: Effect of the spectral filter order k on test RMSE and training time.

Polynomial Filter Order	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
RMSE	25.96	23.92	23.26	23.20	23.15	23.15	23.05	23.04	23.01	23.01
Training Time (s/epoch)	8.021	11.63	15.22	19.46	23.86	26.80	30.57	34.80	37.70	40.32

4.4 Performance under heterogeneous features

Real-world sensor networks rarely observe a single, homogeneous signal: traffic dynamics are modulated by exogenous factors, which are recorded by physically distinct stations on different scales and with different semantics. Standard graph diffusion treats all node features as residing in a common space and propagates them through a shared aggregation, which tends to wash out modality-specific information and over-smooth local representations when heterogeneous signals are mixed. To test this, we augment PEMS04 and PEMS08 with meteorological covariates from NOAA weather stations across the same regions (District 4 and District 8) during the same time intervals, and assess whether each model can exploit the additional modality. Baselines whose architectures do not support additional exogenous node features are omitted from this comparison. As shown in Table 4, ChronoSheaf achieves the lowest MAE and RMSE on both datasets, indicating that ChronoSheaf can effectively exploit heterogeneous input features without degrading forecasting performance.

Table 4: Traffic forecasting results on PEMS04 and PEMS08 augmented with NOAA weather covariates from the corresponding regions, averaged over 12 forecasting steps. Best results are in bold and second-best are underlined.

Model	Multimodal PEMS04			Multimodal PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN (Yu et al., 2018)	21.36	34.42	13.25	17.98	27.52	<u>11.10</u>
DCRNN (Li et al., 2018b)	21.67	35.91	14.32	18.44	27.93	13.62
GW-Net (Wu et al., 2019)	24.17	38.22	16.54	18.88	28.79	14.02
GMAN (Zheng et al., 2020)	23.63	37.67	15.50	17.08	28.41	12.66
CITRUS (Einizade et al., 2024)	20.34	30.11	14.89	15.56	27.31	12.10
STDN (Cao et al., 2025)	<u>18.33</u>	<u>29.76</u>	**11.10**	<u>14.31</u>	<u>24.01</u>	11.63
ChronoSheaf	**15.53**	**27.22**	<u>12.47</u>	**13.26**	**22.59**	**10.55**

5 Conclusion

In this work, we introduced ChronoSheaf, a cellular sheaf defined jointly over space and time, consisting of two main components. First, transitional restriction maps decompose each node's restriction map into a spatial frame and a pair of independently gated frames that map it to its neighbors at the preceding and following timestep, giving a node asymmetric sensitivity to past versus future influence. We proved the resulting joint Laplacian is real, symmetric, and positive semi-definite by construction. Second, the dual spectral filter applies a single polynomial recursion to this operator, with an adaptively estimated spectral radius, giving one filtering pass a receptive field that grows jointly in graph hops and temporal lag. Evaluated on several forecasting benchmarks, ChronoSheaf outperforms established spatio-temporal graph baselines.

6 Ethics and Reproducibility Statement

Ethics Statement: We do not foresee any immediate negative societal or ethical impacts of our work.

Reproducibility Statement: We are committed to ensuring the reproducibility of our results. Detailed experimental setup and dataset descriptions are provided in the appendices. The source code is available at: `https://anonymous.4open.science/r/ChronoSheaf-1D7D/`.

7 AI use statement

In this work, we used generative AI tools for: aiding and polishing writing, grammar and language checks, reviewing the paper before submission, providing feedback on research methodology and experiments, assisting with translation, summarizing existing literature, and editing the research paper to improve readability. We have not used generative AI tools for: ideation, developing theoretical models or conceptual frameworks, formulating mathematical claims, generating synthetic data sets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work.

References

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations (ICLR)*, 2021.

Álvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael Bronstein, and Pierre Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in gnns: Bridging recurrent and graph learning. *Advances in Neural Information Processing Systems*, 38:74356–74393, 2026.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural network for message diffusion on graphs. In *International Conference on Learning Representations*, volume 2025, pp. 16788–16813, 2025.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection laplacians. In *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pp. 28–36. PMLR, 25 Feb–22 Jul 2022.

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

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Lingxiao Cao, Bin Wang, Guiyuan Jiang, Yanwei Yu, and Junyu Dong. Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11463–11471, 2025.

Yuzhou Chen, Ignacio Segovia, and Yulia R Gel. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In *International conference on machine learning*, pp. 1684–1694. PMLR, 2021.

Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 6367–6374, 2022.

Andrea Cini, Ivan Marisca, Filippo Maria Bianchi, and Cesare Alippi. Scalable spatiotemporal graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 7218–7226, 2023.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Aref Einizade, Fragkiskos Malliaros, and Jhony H Giraldo. Continuous product graph neural networks. *Advances in Neural Information Processing Systems*, 37:90226–90252, 2024.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 364–373, 2021.

Stefano Fiorini, Hakan Emre Aktas, Iulia Duta, Pietro Morerio, Alessio Del Bue, Pietro Lio, and Stefano Coniglio. Sheaves reloaded: A direction awakening. In *International Conference on Learning Representations*, volume 2026, pp. 5144–5168, 2026.

Benjamin Girault. Stationary graph signals using an isometric graph translation. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 1516–1520. IEEE, 2015.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Mustafa Hajij, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John Davenport, Balaji Cherukuri, Joseph Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, et al. Copresheaf topological neural networks: A generalized deep learning framework. *Advances in Neural Information Processing Systems*, 38:149759–149803, 2025.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves: J. hansen, r. ghrist. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36:46885–46902, 2023.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *Proceedings of the Forty-Second International Conference on Machine Learning*, 2025.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9267–9276, 2019.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018a.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018b.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with *1k* parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30211–30226. PMLR, 21–27 Jul 2024.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in neural information processing systems*, 36:75354–75371, 2023.

Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. *Advances in Neural Information Processing Systems*, 38:58382–58418, 2025.

Ivan Marisca, Jacob Bamberger, Cesare Alippi, and Michael M. Bronstein. Over-squashing in spatiotemporal graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 33969–33999. Curran Associates, Inc., 2025.

Microsoft Research. Urban air. `https://www.microsoft.com/en-us/research/project/urban-air/`, 2026. Accessed: 2026-07-04.

André Ribeiro, Ana Luiza Tenório, Juan Belieni, Amauri Souza, and Diego Mesquita. Cooperative sheaf neural networks. In *International Conference on Learning Representations*, volume 2026, pp. 144580–144596, 2026.

T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Mohammad Sabbaqi and Elvin Isufi. Graph-time convolutional neural networks: Architecture and theoretical analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 14625–14638, 2023.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 914–921, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *28th International Joint Conference on Artificial Intelligence (IJCAI-19)*. International Joint Conference on Artificial Intelligence (IJCAI), 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 2018.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, 2020.

SUPPLEMENTARY MATERIAL

A ADDITIONAL RESULTS

A.1 PREDICTIONS COMPARISON

Figure 5 complements the spectral filtering analysis of Section 4.3 by comparing 12-step-ahead predictions of ChronoSheaf with Chebyshev orders $K = 2$ and $K = 10$ on nine PEMS08 test nodes. On nodes with steep rush-hour ramps (nodes 89, 103, and 144), $K = 2$ visibly lags the rising and falling edges of the daily cycle and underestimates peak flow. $K = 10$ follows these transitions closely and reduces MAE by a big margin. A higher order therefore lets each node draw on a wider space-time neighborhood, which is most useful when abrupt changes in flow propagate from upstream sensors before they reach the target node.

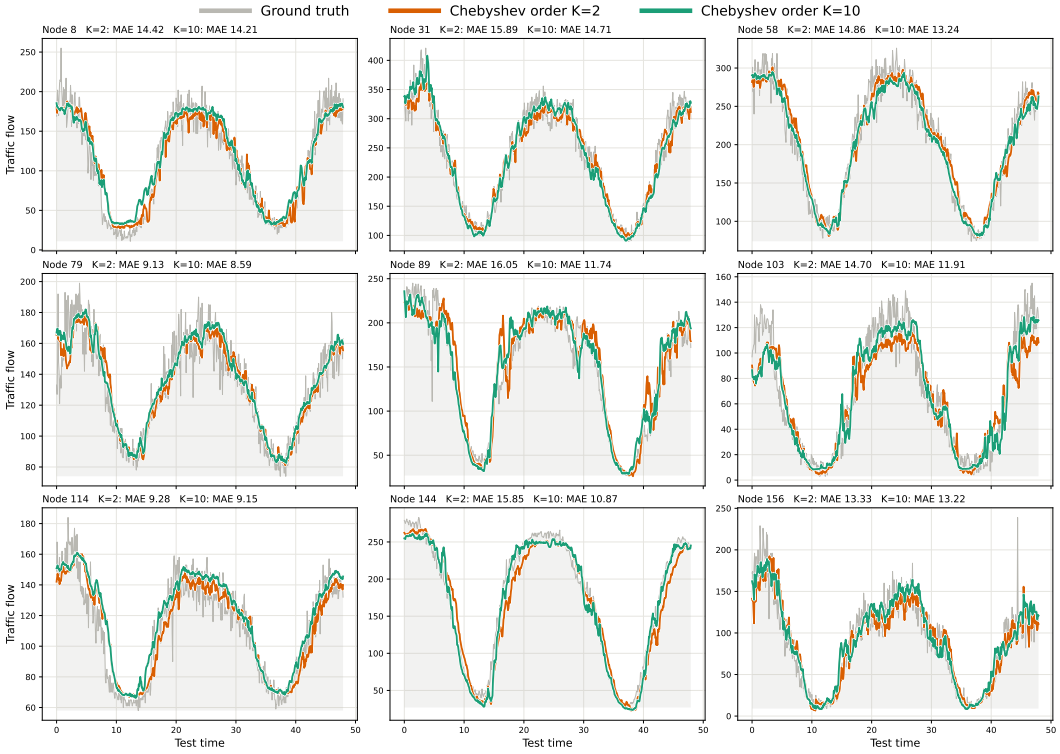


Figure 5: Ground truth versus 12-step-ahead ChronoSheaf predictions on nine PEMS08 test nodes, with Chebyshev order $K = 2$ and $K = 10$. Per-node MAE is shown above each panel.

A.2 ABLATION STUDIES

Table 5 evaluates four variants of ChronoSheaf, each altering one component while keeping all other settings fixed:

- *trivial sheaf*, with all restriction maps set to the identity.
- $F^{\text{in}} = F^{\text{out}}$, tying the two transitional maps.
- self-loop temporal edges only.
- *fixed* $\lambda_{\max}$, replacing the adaptive bound of Eq. equation 9 with a constant ($\lambda_{\max} = 2$).

Table 5: Ablation study on model variants across prediction horizons in MAE (lower is better).

	Molene (weather)					Air Quality (PM2.5)		
Variant	step-1	step-2	step-3	step-4	step-5	1–12 hr	13–24 hr	25–48 hr
Trivial sheaf	0.36	0.61	0.95	1.32	1.52	23.44	32.91	36.06
$F^{\text{in}} = F^{\text{out}}$	0.32	0.55	0.85	1.07	1.34	21.71	30.83	33.72
Self-loop temporal edges only	0.33	0.56	0.88	1.12	1.40	22.06	31.4	34.76
Fixed $\lambda_{\max}$	0.32	0.55	0.84	1.05	1.29	22.05	30.88	33.41
ChronoSheaf (full)	**0.31**	**0.52**	**0.80**	**1.01**	**1.24**	**21.31**	**29.90**	**32.41**

B Complexity and Runtime Analysis

Let N, $|\mathcal{E}|$, P, d, r, K, and M denote the number of nodes, spatial edges, patches, the model and edge-stalk dimensions, the Chebyshev order, and the number of layers. ChronoSheaf never materializes L, which would require $\mathcal{O}(P(N + |\mathcal{E}|)d^2)$ memory. Instead, it computes Lx matrix-free in $\mathcal{O}(P(Nrd + |\mathcal{E}|r))$. It projects stalks into the r-dimensional edge space, aggregates along spatial and temporal edges, and projects back.

The dual spectral filter applies L K times. Because the gates are positive scalars, $\|V_{n,t}\|_2 = g_{n,t,1}\|B_{1,n}\|_2$ (and likewise for F^{in} and F^{out}), so the spectral-radius bound $\hat{\lambda}_{\max}$ needs only $3N$ base-frame norms per layer and adds no backward cost. An explicit eigendecomposition would instead cost $\mathcal{O}((NPd)^3)$. The total forward cost is $\mathcal{O}\big(MP[K(Nrd + |\mathcal{E}|r) + Nd^2]\big)$, which is linear in N for sparse graphs with $|\mathcal{E}| = \mathcal{O}(N)$, and also linear in K.

In Table 6, we include the training time per epoch using different values of the Chebyshev order K on different graph sizes N:

Table 6: Training time (seconds/epoch) for different polynomial filter orders k and different graph sizes N.

Polynomial Filter Order		**k=1**	**k=2**	**k=3**	**k=4**	**k=5**	**k=6**	**k=7**	**k=8**	**k=9**	**k=10**
Molene	**(N = 32)**	1.25	1.82	2.36	3.04	3.71	4.19	4.75	5.43	5.87	6.29
PEMS08	**(N = 170)**	8.021	11.63	15.22	19.46	23.86	26.80	30.57	34.80	37.70	40.32
PEMS04	**(N = 307)**	14.73	21.28	28.04	35.61	43.95	49.07	56.31	63.72	69.40	73.89
Air Quality	**(N = 437)**	23.60	34.31	44.65	57.38	70.04	79.02	89.77	102.56	110.71	118.85
LargeST (SD)	**(N = 716)**	34.89	50.47	66.38	84.51	103.96	116.42	133.18	151.14	164.23	175.20
LargeST (CA)	**(N = 8600)**	422.17	611.48	802.35	1022.91	1257.40	1408.76	1611.02	1829.35	1986.71	2119.84

C Experimental Details

This appendix describes the training protocol, evaluation metrics, and hyperparameters used for all experiments in Section 4. Unless stated otherwise, the same protocol is applied to ChronoSheaf and to all baselines.

C.1 Data Preprocessing

Each dataset is split chronologically into training, validation, and test sets. Following the protocols of the original benchmarks, PEMS04 and PEMS08 (Song et al., 2020) and the two LargeST subsets (Liu et al., 2023) use a 6:2:2 ratio. Molene and Air Quality use a 7:1:2 ratio. Samples are extracted with a stride-one sliding window of T_{in} input steps followed by T_{out} target steps. Windows are extracted separately within each split, so that no window crosses a split boundary. All input channels are z-score standardized independently, using the mean and standard deviation computed on the training split only. The forecast target is standardized with its own training-split statistics. Model outputs are mapped back to the original scale before the training loss and the evaluation metrics are computed. The same normalization is applied to all baselines.

C.2 TRAINING

We use four spatio-temporal blocks and set the rank of the restriction maps to 4. The model is trained using the Adam optimizer with an initial learning rate of 2.5×10^{-3}, $\epsilon = 10^{-4}$, and weight decay of 10^{-4}, with the global gradient norm clipped at 5. Training windows are drawn from the training split with a stride of one, reshuffled at the start of every epoch, and grouped into mini-batches of 16. The maximum number of training epochs is set to 250. After each epoch, the model is evaluated on the validation split using the validation RMSE, computed on the original scale with missing entries masked. The learning rate is multiplied by 0.7 whenever the validation RMSE does not improve for 10 consecutive epochs, with a lower bound of 10^{-5}.

C.3 EVALUATION

Let $y_{i,t}$ and $\hat{y}_{i,t}$ denote the ground truth and the prediction for node i at forecast step t, on the original scale. Let Ω be the set of observed target entries. We report MAE $= \frac{1}{|\Omega|} \sum_{(i,t)\in\Omega} |y_{i,t} - \hat{y}_{i,t}|$, RMSE $= \sqrt{\frac{1}{|\Omega|} \sum_{(i,t)\in\Omega} (y_{i,t} - \hat{y}_{i,t})^2}$, and MAPE $= \frac{100\%}{|\Omega|} \sum_{(i,t)\in\Omega} \left| \frac{y_{i,t} - \hat{y}_{i,t}}{y_{i,t}} \right|$. Unless a specific horizon is stated, metrics are averaged over all forecast steps and all test windows.

All experiments were run on a single NVIDIA A100 GPU (80 GB). Each configuration was trained and evaluated five times with random seeds $\{42, 43, 44, 45, 46\}$, and all reported numbers are the mean over the five runs.

C.4 DATASET DETAILS

Table 7 summarizes the six benchmarks.

Table 7: Summary of dataset statistics.

Dataset	Domain	Nodes	Time steps	Interval	Period	Target
PEMS04	Traffic	307	16,992	5 min	01/2018–02/2018	Flow
PEMS08	Traffic	170	17,856	5 min	07/2016–08/2016	Flow
LargeST (SD)	Traffic	716	525,888	5 min	01/2017–12/2021	Flow
LargeST (CA)	Traffic	8,600	525,888	5 min	01/2017–12/2021	Flow
Molene	Weather	32	744	1 h	01/2014	Temperature
Air Quality	Air quality	437	8,760	1 h	05/2014–04/2015	PM2.5

Multimodal PEMS04 and PEMS08. For the heterogeneous-feature experiment, we augment PEMS04 and PEMS08 with meteorological covariates from NOAA weather stations located in District 4 and District 8, over the same periods as the traffic data (January to February 2018 and July to August 2016, respectively). We retain four variables: dry-bulb temperature (°C), hourly precipitation (in), visibility (mi), and wind speed (mph). The hourly NOAA observations are aligned to the 5-minute traffic resolution by broadcasting. NOAA data did not have any missing values. All channels are standardized independently with training-split statistics, and traffic flow remains the sole forecast target.