OSCILLATORY NEURAL DYNAMICS OVER SHEAVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective long-range propagation remains a central challenge in graph neural networks, as increasing a model's propagation depth does not guarantee that distant nodes effectively influence each other. Sheaf neural networks enrich graph propagation through matrix-valued transport between stalks; still, this expressivity alone does not automatically imply effective long-range communication. We introduce ONDA, a long-range graph learning framework based on *operator-valued information waves*. Stalk-valued representations evolve through second-order dynamics governed by learned sheaf transport operators, combining wave-like propagation with expressive local geometry. We characterize long-range influence through a stalk-wise sensitivity analysis and show that the cross-influence never vanishes. Across long-range propagation, severe graph bottlenecks, graph transfer, and heterophilic benchmarks, ONDA consistently improves over scalar wave propagation, diffusive sheaf baselines, and state-of-the-art models, demonstrating the benefit of coupling wave dynamics with matrix-valued transport.

1 INTRODUCTION

Effectively modeling long-range dependencies remains a central challenge in deep learning for graphs (Shi et al., 2023; Arroyo et al., 2025). Graph Neural Networks (GNNs) (Micheli, 2009; Scarselli et al., 2009; Defferrard et al., 2016; Gilmer et al., 2017; Kipf & Welling, 2017) usually rely on local message-passing to exchange information between nodes, meaning that dependencies between non-adjacent regions must be mediated through repeated propagation steps. However, while increasing depth expands the receptive field (i.e., the region from which information can be gathered), it does not guarantee that distant nodes communicate effectively. Indeed, challenges such as over-smoothing (Cai & Wang, 2020; Oono & Suzuki, 2020; Rusch et al., 2023), over-squashing (Alon & Yahav, 2021; Topping et al., 2022; Di Giovanni et al., 2023; Mishayev et al., 2025), and, more generally, vanishing gradients (Arroyo et al., 2025) prevent relevant information exchange between distant nodes. For this reason, *learning how to propagate information effectively over long graph distances* has become a critical problem in graph representation learning.

Existing approaches address this problem from several directions. Rewiring methods modify the graph to shorten communication paths (Topping et al., 2022; Gutteridge et al., 2023), while graph Transformers allow direct message passing between any pair of nodes via attention mechanisms (Shi et al., 2021a; Rampášek et al., 2022). A complementary line of work instead changes the dynamics of propagation while retaining sparse local interactions. In particular, differential-equation-inspired GNNs (DE-GNNs) have explored stable and non-dissipative dynamics as a way to maintain information over depth (Rusch et al., 2022; Gravina et al., 2023; Heilig et al., 2025; Hariri et al., 2025). Oscillatory DE-GNNs (Rusch et al., 2022; Trenta et al., 2025) follow this direction by modeling information propagation through second-order wave dynamics on the original graph.

While these approaches primarily focus on improving *how far* information can propagate, another important aspect is *how* information is exchanged between neighboring nodes. Many widely used GNNs generally propagate features through local aggregation operators based on scalar edge weights and shared feature transformations. Cellular sheaves provide a richer framework by associating local vector spaces (or stalks) with nodes and edges, together with linear restriction maps that relate node representations within a common edge space (Curry, 2014; Hansen & Ghrist, 2019; Hansen & Gebhart, 2020). Sheaf Neural Networks (SNNs) learn these maps, inducing matrix-valued couplings between neighboring node stalks and substantially enlarging the class of graph-supported propagation operators (Bodnar et al., 2022). This additional expressivity, however, does not by itself

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.

guarantee effective long-range communication between nodes. Indeed, neural sheaf models still rely on repeated dissipative local propagation, while information originating from distant regions must therefore traverse multiple successive sheaf transformations before reaching a target node. Consequently, maintaining informative signals over depth remains a largely underexplored challenge in SNNs, complementary to the problem of learning richer local transport. Recent works have started investigating how to extend neural sheaf architectures through mechanisms like large-scale diffusion (Bamberger et al., 2025) and directional routing (Ribeiro et al., 2026) to broaden the radius of information propagation across the graph.

Rather than extending the spatial range of individual updates or controlling which messages traverse the graph, in this work, we seek to design a principled sheaf neural network whose dynamics enable useful stalk-valued information to remain active as it traverses the graph. This motivates ONDA (**o**scillatory **n**eural **d**ynamics on she**a**ves), a framework for long-range information propagation based on *operator-valued information waves*. In ONDA, stalk-valued representations evolve through an oscillatory second-order dynamical system governed by a learned sheaf Laplacian. The learned sheaf shapes the propagation of these information waves: its restriction maps determine how information is transformed between stalks and how different propagated contributions reinforce or cancel when they interact. In this sense, the learned sheaf acts as a propagation medium whose local properties govern how information waves travel across the graph. This couples the persistence of wave propagation with the expressive local geometry of cellular sheaves, allowing information to remain influential over distance while being non-trivially transformed along the graph.

Our contributions are threefold: **(i)** we introduce ONDA, a new class of sheaf neural networks based on second-order oscillatory dynamics, providing a non-trivial extension of wave-like graph propagation to matrix-valued sheaf operators; **(ii)** we theoretically investigate the long-range propagation properties of ONDA, showing that its oscillatory dynamics preserve the influence of distant information; **(iii)** we demonstrate strong and consistent empirical performance across a diverse suite of tasks, including long-range propagation, over-squashing mitigation, and heterophilic graph learning.

2 Related Work

Long-Range Propagation on Graphs. Although stacking message-passing layers progressively enlarges the receptive field, information from distant nodes can rapidly lose influence due to phenomena such as over-smoothing (Cai & Wang, 2020; Oono & Suzuki, 2020; Rusch et al., 2023) and over-squashing (Alon & Yahav, 2021; Topping et al., 2022; Di Giovanni et al., 2023; Mishayev et al., 2025), both closely related to vanishing gradients (Arroyo et al., 2025). Various approaches aim to improve long-range information propagation. Graph rewiring methods (Topping et al., 2022; Gasteiger et al., 2019; Gutteridge et al., 2023; Black et al., 2023) modify the graph topology to create more favorable communication paths, whereas Graph Transformers (Kreuzer et al., 2021; Rampášek et al., 2022; Shi et al., 2021a; Ying et al., 2021; Ma et al., 2023) use attention to learn direct interactions between nodes. Other methods modify the propagation dynamics through weight-space regularization (Gravina et al., 2023; 2025), port-Hamiltonian dynamics (Heilig et al., 2025), message filtering (Errica et al., 2024; Finkelshtein et al., 2024), or virtual nodes (Southern et al., 2025). Oscillatory GNNs (Rusch et al., 2022; Trenta et al., 2025) exploit second-order oscillatory propagation dynamics to facilitate communication between distant nodes, but operate within conventional GNN frameworks and therefore rely on standard graph-based interactions. In contrast, ONDA introduces oscillatory dynamics into sheaf neural networks, combining persistent wave-like propagation with learned matrix-valued transport operators that can control how information is transformed and transmitted along individual edges. Together, these mechanisms allow ONDA to combine effective long-range propagation with richer, geometry-aware information transport.

Sheaf Neural Networks. Cellular sheaves are used to extend graph-based learning by associating vector spaces with graph cells and linear restriction maps with their incidences, giving rise to matrix-valued generalizations of classical graph operators (Hansen & Gebhart, 2020). Building on this framework, Bodnar et al. (2022) introduced learnable sheaf structures into GNNs, showing that the resulting diffusion can provide favourable properties in heterophilic settings and alleviate over-smoothing. Subsequent works have explored different ways of parametrizing and learning the underlying sheaf, including connection-Laplacian constructions (Barbero et al., 2022), polynomial filters (Borgi et al., 2025), inductive biases inspired by joint diffusion processes (Hernandez Car-

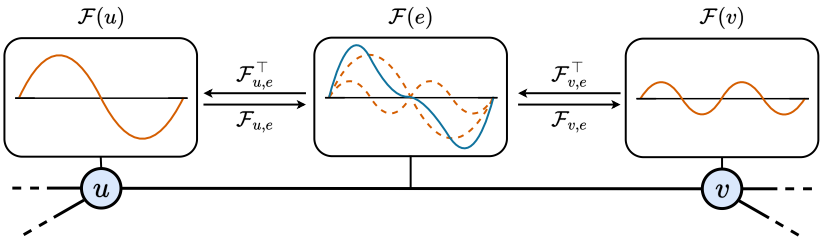


Figure 1: Schematic view of ONDA on a single edge. Orange curves show the oscillatory evolution of node states in the stalks $\mathcal{F}(u)$ and $\mathcal{F}(v)$; learned restriction maps express them in the shared edge stalk $\mathcal{F}(e)$ (dashed orange). The blue curve shows their superposition, highlighting constructive and destructive interference between the mapped signals.

alt et al., 2024), and adaptive nonlinear sheaf Laplacians (Zaghen et al., 2024). More recently, Bamberger et al. (2025) proposed using continuous diffusion over flat vector bundles to operate over larger spatial scales, while other approaches consider directional routing (Ribeiro et al., 2026) specifically designed to better exploit deep sheaf propagation. Nevertheless, most existing SNNs remain centered around diffusion-like dynamics, which are inherently dissipative and struggle to capture long-range interactions. In contrast, ONDA introduces second-order oscillatory dynamics into sheaf neural networks, interpreting learned matrix-valued restriction maps as the wave propagation medium whose local properties govern how information waves travel across the graph.

3 ONDA

We introduce Oscillatory Neural Dynamics on Sheaves (ONDA), a new class of sheaf neural networks driven by second-order oscillatory dynamics, allowing stalk-valued information to remain relevant as it traverses the graph. A schematic illustration of ONDA is provided in Figure 1.

3.1 Notation and preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the finite set of n nodes (or vertices) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges. Given two nodes $u, v \in \mathcal{V}$ we denote an edge between them with the unordered pair $\{u, v\}$. Graph connectivity is encoded by the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, with degree matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ defined by $\boldsymbol{D}_{ii} = \sum_{j=1}^{n} \boldsymbol{A}_{ij}$. A *cellular sheaf on* $\mathcal{G}$ (Shepard, 1985; Curry, 2014) assigns a finite-dimensional vector space $\mathcal{F}(v)$ to each vertex $v \in \mathcal{V}$, a finite-dimensional vector space $\mathcal{F}(e)$ to each edge $e \in \mathcal{E}$, and a linear restriction map $\mathcal{F}_{v,e} : \mathcal{F}(v) \to \mathcal{F}(e)$ to each incidence $v \in e$. We refer to $\mathcal{F}(v)$ and $\mathcal{F}(e)$ as the *vertex* and *edge stalks*, respectively. In this paper, all stalks share a common dimension $d \geq 1$. The space of *0-cochains*, or *stalk-valued node states* is $C^0(\mathcal{G}; \mathcal{F}) := \bigoplus_{v \in \mathcal{V}} \mathcal{F}(v)$. We write $\boldsymbol{X} = (\boldsymbol{X}_v)_{v \in \mathcal{V}} \in C^0(\mathcal{G}; \mathcal{F})$, where $\boldsymbol{X}_v \in \mathcal{F}(v)$. For c feature channels, node features lie in $C^0(\mathcal{G}; \mathcal{F}) \otimes \mathbb{R}^c$. The *Sheaf Laplacian* (Hansen & Ghrist, 2019; Hansen, 2020) $L_{\mathcal{F}} : C^0(\mathcal{G}; \mathcal{F}) \to C^0(\mathcal{G}; \mathcal{F})$ acts nodewise as:

$$(L_{\mathcal{F}} \boldsymbol{X})_v = \sum_{e = \{v,u\} \in \mathcal{E}} \mathcal{F}_{v,e}^{*} \big(\mathcal{F}_{v,e} \boldsymbol{X}_v - \mathcal{F}_{u,e} \boldsymbol{X}_u \big) \tag{1}$$

where each stalk is equipped with an inner product and $\mathcal{F}_{v,e}^{*}$ denotes the adjoint of $\mathcal{F}_{v,e}$. The resulting sheaf Laplacian is self-adjoint and positive semidefinite (Hansen & Ghrist, 2019). Along each edge, the restriction maps place neighbouring node representations in a common space, where their discrepancy is computed; the adjoint then maps this discrepancy back to the incident node, where it is aggregated with the remaining neighbour contributions. For $d = 1$, Euclidean inner products, and identity restriction maps, $L_{\mathcal{F}}$ reduces to the standard graph Laplacian $\boldsymbol{L}_{\mathcal{G}} = \boldsymbol{D} - \boldsymbol{A}$ (Brouwer & Haemers, 2012). We also use the symmetrically normalized sheaf Laplacian $\Delta_{\mathcal{F}} := \mathcal{D}_{\mathcal{F}}^{-1/2} L_{\mathcal{F}} \mathcal{D}_{\mathcal{F}}^{-1/2}$, where $\mathcal{D}_{\mathcal{F}}$ is the block diagonal part of $L_{\mathcal{F}}$ (Hansen & Ghrist, 2019).

Given its oscillatory dynamics, ONDA is inherently a differential-equation-based architecture, where node representations are interpreted as the evolving states of a dynamical system defined

over the graph (Poli et al., 2019; Gravina et al., 2023; Han et al., 2024). Under this view, we denote the collection of all node states at time t as $\boldsymbol{X}(t) \in C^0(\mathcal{G}; \mathcal{F}) \otimes \mathbb{R}^c$, stacked as an $nd \times c$ matrix. As in DE-GNNs, the continuous dynamics are instantiated as a neural architecture through numerical discretization (Gravina et al., 2023). Therefore, given an initial condition $\boldsymbol{X}(0) = \overline{\boldsymbol{X}}$ and a step size $h > 0$, we denote by $\boldsymbol{X}^{(\ell)}$ the node features after ℓ integration steps, initialized with $\boldsymbol{X}^{(0)} = \overline{\boldsymbol{X}}$. Thus, $\boldsymbol{X}^{(\ell)} \approx \boldsymbol{X}(t_\ell)$, where $t_\ell = h\ell$ and the parenthesized superscript indexes the discrete steps.

3.2 Oscillatory Neural Dynamics on Sheaves

Neural sheaf models replace scalar graph interactions with matrix-valued transport between local stalks, transforming information as it crosses each edge. Yet, expressive transport alone does not guarantee that information remains influential over long propagation ranges. ONDA addresses both aspects jointly: the learned sheaf controls how information moves between neighbouring stalks, while second-order oscillatory dynamics govern how it evolves with depth.

Specifically, we model the evolution of the stalk-valued representations $\boldsymbol{X}(t) \in C^0(\mathcal{G}; \mathcal{F}) \otimes \mathbb{R}^c$ through the following second-order differential equation

$$\ddot{\boldsymbol{X}}(t) = -L_{\mathcal{F}}(\boldsymbol{I}_n \otimes \boldsymbol{W}_1)\boldsymbol{X}(t)\boldsymbol{W}_2 - R_\theta(\boldsymbol{X}(t)) \odot \dot{\boldsymbol{X}}(t) + F_\theta(\boldsymbol{X}(t)), \quad \boldsymbol{X}(0) = \overline{\boldsymbol{X}} \tag{2}$$

where $L_{\mathcal{F}}$ is the sheaf Laplacian from Equation (1), $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix, $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$ is the weight matrix acting on stalk coordinates, and $\boldsymbol{W}_2 \in \mathbb{R}^{c \times c}$ transforms feature channels. The terms $R_\theta(\boldsymbol{X}(t)) \in \mathbb{R}^{nd \times c}_{\geq 0}$ and $F_\theta(\boldsymbol{X}(t)) \in \mathbb{R}^{nd \times c}$ provide state-dependent dissipation and external forcing, respectively, while $\odot$ denotes the Hadamard product.

In Equation (2), the sheaf operator shapes the information waves: restriction maps determine how signals transform across edges and how incoming wave contributions combine within each stalk. This mirrors classical wave equations (Evans, 2010), where the spatial Laplacian operator encoded the propagation properties of the underlying medium. ONDA therefore couples the propagation bias of wave dynamics with learned matrix-valued transport.

With zero dissipation and forcing and identity weight matrices, Equation (2) reduces to a cellular-sheaf generalization of the graph wave equation $\ddot{\boldsymbol{X}}(t) = -\boldsymbol{L}_{\mathcal{G}}\boldsymbol{X}(t)$. Here, $\boldsymbol{L}_{\mathcal{G}}$ is the graph Laplacian which governs spatial coupling in the graph wave equation and induces conservative oscillatory dynamics.

Purely conservative wave dynamics can nevertheless be restrictive for downstream learning, because not all information should necessarily be preserved indefinitely. We therefore augment the oscillatory term with the state-dependent dissipation $R_\theta(\boldsymbol{X}(t))$ and external forcing $F_\theta(\boldsymbol{X}(t))$. Dissipation adaptively attenuates propagated information, while forcing supplies an additional learned drive. Together, they let ONDA balance conservative and non-conservative behavior and adapt its propagation dynamics to the task at hand.

Discretization and Implementation Details. We obtain a neural architecture from Equation (2) by numerically discretizing its continuous dynamics. Introducing the velocity $\boldsymbol{V}(t) = \dot{\boldsymbol{X}}(t)$ and applying a finite-difference scheme (Ferziger & Peric, 2001) yields the update

$$\begin{aligned} \boldsymbol{V}^{(\ell+1)} &= \boldsymbol{V}^{(\ell)} - hL_{\mathcal{F}}\big((\boldsymbol{I}_n \otimes \boldsymbol{W}_1)\boldsymbol{X}^{(\ell)}\boldsymbol{W}_2\big) + hF_\theta(\boldsymbol{X}^{(\ell)}) - hR_\theta(\boldsymbol{X}^{(\ell)}) \odot \boldsymbol{V}^{(\ell)}, \\ \boldsymbol{X}^{(\ell+1)} &= \boldsymbol{X}^{(\ell)} + h\boldsymbol{V}^{(\ell+1)} \end{aligned} \tag{3}$$

where $h > 0$ is the integration step size, and $\ell = 0, \ldots, \mathcal{L} - 1$.

To increase expressivity, we stack multiple instances of the propagation dynamics in Equation (3) and interleave them with nonlinear feature transformations, following the general principle of separating recurrent propagation from channel-wise nonlinear mixing (Gu et al., 2021; Orvieto et al., 2023; Ceni et al., 2025; Trenta et al., 2025). A *block* consists of $\mathcal{L}$ propagation steps from Equation (3), followed by an MLP transformation. In block b, the features after $\mathcal{L}$ propagation steps initialize block $(b+1)$ through:

$$\boldsymbol{X}^{(b+1,0)} = \mathrm{MLP}_b(\boldsymbol{X}^{(b,\mathcal{L})}), \qquad b = 0, \ldots, B-1 \tag{4}$$

The velocity of the new block is again initialized from its input representation and the final block output is passed to a task-specific decoder. Propagation within each block can be either fixed or

adaptive. In the fixed setting, restriction maps are inferred once from the block input and held constant for all $\mathcal{L}$ steps: each block first learns the sheaf governing transport, then evolves node representations according to Equation (3). In the adaptive setting, restriction maps are recomputed at each step from the current features, allowing the evolving node states to reshape the propagation medium. The number of blocks B, the number of propagation steps $\mathcal{L}$, and the integration step size h are treated as architectural hyperparameters.

Following standard SNN practice (Bodnar et al., 2022; Bamberger et al., 2025; Ribeiro et al., 2026), we parametrize restriction maps with neural networks. We consider both symmetrically normalized sheaf Laplacians and a directed variant. In the latter, restriction maps are learned independently for each edge orientation, and each discrepancy is applied only at the receiving endpoint. For simplicity, we retain the notation $L_{\mathcal{F}}$. Further details on sheaf normalization and the directed variant are provided in Appendix A.2; the restriction-map parametrizations are described in Appendix B.1.

4 Theoretical Analysis

In this section, we study the evolution of node states in a single ONDA block, with particular focus on providing theoretical grounding for long-range influence across nodes. We start from the continuous definition of the model in Equation (2) without dissipation and external forcing, showing that the oscillatory dynamics provide explicit guarantees on the cross-influence between nodes. In Theorem 1, we translate these results to the discrete case.

To isolate the benefits of the oscillatory dynamics, we fix the restriction maps within one block and set the weights to the identity. For simplicity, we also work with a single feature channel and write $\mathcal{H} := C^0(\mathcal{G};\mathcal{F}) = \bigoplus_{v\in\mathcal{V}} \mathcal{F}(v) \cong \mathbb{R}^{nd}$ for the feature space. With a fixed sheaf Laplacian $L_{\mathcal{F}} : \mathcal{H} \to \mathcal{H}$, the conservative form of ONDA is given by:

$$\ddot{\boldsymbol{X}}(t) + L_{\mathcal{F}}\boldsymbol{X}(t) = 0, \qquad \boldsymbol{X}(0) = \overline{\boldsymbol{X}}, \qquad \dot{\boldsymbol{X}}(0) = \overline{\boldsymbol{V}} \tag{5}$$

where $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{V}}$ are respectively the initial node features and their velocities. We now define the equivalent of the sine and cosine functions for the sheaf Laplacian through the corresponding power series expansion (Higham, 2008):

$$\mathcal{C}_{\mathcal{F}}(t) = \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k}}{(2k)!}L_{\mathcal{F}}^k, \qquad \mathcal{S}_{\mathcal{F}}(t) = \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k+1}}{(2k+1)!}L_{\mathcal{F}}^k \tag{6}$$

Analogously to the wave equation in physics (Evans, 2010), the general solution to Equation (5) is given by $\boldsymbol{X}(t) = \mathcal{C}_{\mathcal{F}}(t)\overline{\boldsymbol{X}} + \mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}}$, as shown in Appendix A.3. This describes the node features at time t through a combination of the sine and cosine operators.

We now seek to measure how different nodes influence each other through the common framework of sensitivity analysis (Di Giovanni et al., 2023). Concretely, we measure how changes in the features of node u affect a target node v. For each $u \in \mathcal{V}$, we define $\iota_u : \mathcal{F}(u) \to \mathcal{H}$ and $\pi_u : \mathcal{H} \to \mathcal{F}(u)$ as the inclusion and projection operators, respectively. We define $\boldsymbol{W} \in \mathcal{F}(u)$ as a perturbation of the initial state at node u. Thus, the perturbed initial condition of the whole graph is $\overline{\boldsymbol{X}} + \iota_u\boldsymbol{W}$, where the components relative to u are shifted by a vector $\boldsymbol{W}$. Given the closed-form solution of Equation (5), we can calculate the difference between the perturbed and unperturbed trajectories as:

$$\Delta\boldsymbol{X}(t) = (\mathcal{C}_{\mathcal{F}}(t)(\overline{\boldsymbol{X}} + \iota_u\boldsymbol{W}) + \mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}}) - (\mathcal{C}_{\mathcal{F}}(t)\overline{\boldsymbol{X}} + \mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}}) = \mathcal{C}_{\mathcal{F}}(t)\iota_u\boldsymbol{W} \tag{7}$$

This equation already shows that changes in the stalk of a single node u affect the whole trajectory indefinitely, as $\mathcal{C}_{\mathcal{F}}(t)$ does not vanish in the limit of $t \to \infty$. We refer to the evolving response $\Delta\boldsymbol{X}(t)$ generated by a perturbation at node u as its *information wave*. Projecting this wave onto a target node v gives $\Delta\boldsymbol{X}_v(t) = \pi_v\mathcal{C}_{\mathcal{F}}(t)\iota_u\boldsymbol{W}$. Independent of the particular perturbation $\boldsymbol{W}$, we therefore define the *source-to-target influence* at time t as:

$$J^{\mathcal{F}}_{v\leftarrow u}(t) := \pi_v\mathcal{C}_{\mathcal{F}}(t)\iota_u = \big(\mathcal{C}_{\mathcal{F}}(t)\big)_{vu} : \mathcal{F}(u) \to \mathcal{F}(v) \tag{8}$$

In contrast to the scalar case, the map $J^{\mathcal{F}}_{v\leftarrow u}(t)$ is a matrix-valued transformation between the source and target stalks, describing both the strength of the interaction and how directions in $\mathcal{F}(u)$ are

transformed in $\mathcal{F}(v)$. The fact that the operators $\mathcal{C}_{\mathcal{F}}, \iota_u, \pi_v$ are all linear allows for the linear superposition of waves and their effects, similar to how physical waves linearly compose and decompose (Evans, 2010). In the small perturbation limit, the source-to-target response in Equation (8) can be thought of as the equivalent of the Jacobian matrix in sensitivity analysis on standard GNNs (Di Giovanni et al., 2023) to the sheaf case.

To understand whether the non-vanishing influence of node u on the whole trajectory enables long-range propagation, we first examine the behaviour of Equation (8) depending on the target node v in terms of the graph structure. Since $(L_{\mathcal{F}})_{vu} = 0$ whenever distinct nodes u, v are not adjacent, applying $L_{\mathcal{F}}$ once only couples the stalk of node u to itself or adjacent ones. Subsequent powers $L_{\mathcal{F}}^k$ in the definition of the cosine operator $\mathcal{C}_{\mathcal{F}}$ can couple only nodes connected by a path of length at most k. For $k \geq 0$, we formalise this idea (see also Appendix A.4) by defining $\mathcal{W}_k(u, v)$ as the set of length-k walks $\omega = (v_0 = u, v_1, \ldots, v_k = v)$, where each step either follows an edge or remains at the current node. From this we can construct an operator $(L_{\mathcal{F}})_\omega$ as the ordered composition of Laplacian blocks along ω, that is, $(L_{\mathcal{F}})_\omega := (L_{\mathcal{F}})_{v_k v_{k-1}} \ldots (L_{\mathcal{F}})_{v_1 v_0} : \mathcal{F}(u) \to \mathcal{F}(v)$.

For $k = 0$, we define the empty product as the identity on $\mathcal{F}(u)$. In particular, this operator describes the stalk-to-stalk transformation accumulated along the walk ω and lets us decompose the source-to-target response across all different walks that connect u to v.

Proposition 1 (Pathwise source-to-target response)**.** *Let $u, v \in \mathcal{V}$ be two nodes connected by a shortest path of length r. Then:*

$$J^{\mathcal{F}}_{v \leftarrow u}(t) = \sum_{k=r}^{\infty} (-1)^k \frac{t^{2k}}{(2k)!} \sum_{\omega \in \mathcal{W}_k(u,v)} (L_{\mathcal{F}})_\omega \tag{9}$$

If there is no path connecting u and v in the graph, then $J^{\mathcal{F}}_{v \leftarrow u}(t) = 0$ for every t.

The proof is provided in Appendix A.4. Thus, if u and v are r edges apart, the first possible contribution comes from shortest paths of length r, with longer walks contributing through higher powers of $L_{\mathcal{F}}$. A key observation is that the propagation time t weighs the contribution of different walks through the coefficients $\frac{(-1)^k t^{2k}}{(2k)!}$. Hence, the contribution of walks of different lengths to the source-to-target response depends on the propagation time t, and graph distance does not prevent communication between connected nodes. At early times, shorter walks dominate the response, while increasing t allows contributions from progressively longer walks to become relevant. Intuitively, the longer the wave evolves, the more opportunities it has to propagate through the graph and influence the target through different routes.

While Proposition 1 describes how the source-to-target response is assembled from local interactions, we now describe its evolution as time varies. For the self-adjoint sheaf Laplacian, this temporal behaviour can be characterised through its spectral theory (Hansen & Ghrist, 2019; Hansen, 2020). By the finite-dimensional spectral theorem (Halmos, 1974), we can decompose $L_{\mathcal{F}} = \sum_\lambda \lambda P_\lambda$, where the sum runs over distinct eigenvalues $\lambda \geq 0$ and P_λ is the orthogonal projector onto the corresponding eigenspace. As before, we write $(P_\lambda)_{vu} := \pi_v P_\lambda \iota_u : \mathcal{F}(u) \to \mathcal{F}(v)$ for the associated source-to-target block.

Proposition 2 (Spectral source-to-target response)**.** *For the sheaf Laplacian $L_{\mathcal{F}}$, the source-to-target sensitivity satisfies:*

$$J^{\mathcal{F}}_{v \leftarrow u}(t) = (P_0)_{vu} + \sum_{\lambda > 0} \cos(t\sqrt{\lambda})(P_\lambda)_{vu} \tag{10}$$

The proof is provided in Appendix A.5. This statement makes the oscillatory nature of the response explicit. Each block $(P_\lambda)_{vu}$ gives the contribution of the corresponding wave frequency and eigenspace, while the factor $\cos(t\sqrt{\lambda})$ determines how that contribution changes over time. The term $(P_0)_{vu}$ remains constant, whereas the positive-eigenvalue contributions oscillate. Waves at different frequencies can reinforce or cancel, so a non-trivial source-to-target response can be weak or strong depending on time, but cannot decay to zero. This fundamentally distinguishes ONDA from sheaf models based on heat-flow dynamics, where positive-eigenvalue components decay exponentially (Bodnar et al., 2022; Bamberger et al., 2025).

The continuous-time analysis shows that injecting oscillatory biases into feature dynamics sustains long-range influence on sheaves. We now explain how the above results directly translate into the

discrete updates from Equation (3), under the same assumptions of no dissipation or external forcing. Starting from the same perturbation $\boldsymbol{W} \in \mathcal{F}(u)$ at the source node u, let:

$$\Delta \boldsymbol{X}^{(\ell)} := \boldsymbol{X}^{(\ell)}(\overline{\boldsymbol{X}} + \iota_u \boldsymbol{W}) - \boldsymbol{X}^{(\ell)}(\overline{\boldsymbol{X}}) \tag{11}$$

denote the resulting change in the node features after ℓ discrete updates. Its component at a target node v is $\Delta \boldsymbol{X}_{v,\ell} := \pi_v \Delta \boldsymbol{X}^{(\ell)} \in \mathcal{F}(v)$. As in the continuous case, we define the *discrete source-to-target sensitivity* $J^{\mathcal{F}}_{v \leftarrow u}[\ell, h]$ by the linear map satisfying $\Delta \boldsymbol{X}_{v,\ell} = J^{\mathcal{F}}_{v \leftarrow u}[\ell, h]\boldsymbol{W}$. Thus, $J^{\mathcal{F}}_{v \leftarrow u}[\ell, h]$ describes how a perturbation in the source stalk is transformed into a response at the target after ℓ discrete updates with step size h. The following result gives this discrete source-to-target sensitivity a closed-form expression in terms of the spectral projectors, under a mild stability condition. In particular, it applies to the symmetrically normalized sheaf Laplacian $\Delta_{\mathcal{F}}$ for $0 < h < \sqrt{2}$, since its spectrum is contained in $[0, 2]$, as shown in Appendix A.2.

Theorem 1 (Discrete source-to-target response)**.** *Suppose $L_{\mathcal{F}}$ is self-adjoint and positive semidefinite and $h^2 \lambda_{\max}(L_{\mathcal{F}}) < 4$. Then, for every integer $\ell \geq 0$:*

$$J^{\mathcal{F}}_{v \leftarrow u}[\ell, h] = \sum_{\lambda} \frac{\cos\big((\ell + \frac{1}{2})\theta_h(\lambda)\big)}{\cos\big(\theta_h(\lambda)/2\big)} (P_\lambda)_{vu} \tag{12}$$

where $\theta_h(\lambda) = 2 \arcsin\left(\frac{h\sqrt{\lambda}}{2}\right)$. Moreover, $J^{\mathcal{F}}_{v \leftarrow u}[\ell, h] = 0$ whenever there is no path from u to v of length at most ℓ. Finally, for each node pair u, v, the response is bounded and, if it is non-zero at some step, it does not converge to zero as $\ell \to \infty$.

The proof is provided in Appendix A.6. Thus, both features and properties on long-range propagation and non-vanishing of responses we discussed previously still hold for the discretized dynamics that define ONDA. First, the local nature of message-passing updates enables communication between nodes at distance r only after $\ell \geq r$ iterations. At the same time, each positive-eigenvalue contribution remains oscillatory across depth and time, and does not vanish, making the discrete dynamics consistent with the continuous ones. Indeed, as shown in Appendix A.6, for each fixed eigenvalue λ, the coefficient multiplying $(P_\lambda)_{vu}$ in Equation (12) converges to $\cos(T\sqrt{\lambda})$ as $\ell \to \infty$ and $h \to 0$ with $\ell h = T$ fixed. Taking this limit in the spectral expansion therefore recovers the continuous-time expression at time T. Thus, for sufficiently small step sizes, ONDA inherits the persistent oscillatory behaviour of the continuous dynamics, leveraging matrix-valued sheaf operators for long-range propagation.

5 Experiments

We evaluate ONDA across complementary settings that isolate distinct aspects of effective information propagation on graphs. Section 5.1 examines long-range propagation across structurally challenging topologies with severe information bottlenecks. Section 5.2 provides a controlled test of short-range over-squashing, requiring densely connected regions to exchange aggregated information through a narrow communication channel. Section 5.3 evaluates transfer across varying source-to-target distances and graph topologies. We further test ONDA beyond explicitly propagation-oriented tasks on heterophilic benchmarks in Appendix C.2. Appendix B details the sheaf learners and hyperparameter search space, Appendix C.1 reports an ablation of the restriction-map parametrization, and Appendix D analyzes computational cost and runtime.

5.1 Long-range Propagation: ECHO Benchmark

We first evaluate ONDA on ECHO (Miglior et al., 2026), a benchmark specifically designed to assess long-range propagation capabilities. Its synthetic suite, ECHO-Synth, comprises single-source shortest path (SSSP), node eccentricity, and graph diameter prediction, tasks whose solutions depend on information travelling across distant regions of the graph. Models with limited propagation range therefore incur high error. We follow data splits, hyperparameters search space, and experimental protocol of (Miglior et al., 2026).

Results. As shown in Table 1, ONDA achieves the lowest error on both SSSP and eccentricity, substantially outperforming the strongest competing methods, while remaining competitive on diameter

Table 1: **ECHO-Synth benchmark results.** Performance comparison on MAE (↓). Baseline values are reported directly from Miglior et al. (2026). For each column, the **best**, **second-best**, and **third-best** results are highlighted.

Model	sssp ↓	ecc ↓	diam ↓
A-DGN	$1.176_{\pm 0.140}$	$4.981_{\pm 0.037}$	$\mathbf{1.151_{\pm 0.038}}$
DRew	$1.279_{\pm 0.011}$	$4.651_{\pm 0.020}$	$1.243_{\pm 0.047}$
GCN	$2.102_{\pm 0.094}$	$5.233_{\pm 0.034}$	$3.832_{\pm 0.262}$
GCNII	$2.128_{\pm 0.429}$	$5.241_{\pm 0.030}$	$2.005_{\pm 0.093}$
GIN/GINE	$2.234_{\pm 0.271}$	$4.869_{\pm 0.092}$	$1.630_{\pm 0.161}$
GPS	$0.472_{\pm 0.050}$	$4.758_{\pm 0.021}$	$2.160_{\pm 0.098}$
GraphCON	$5.734_{\pm 0.011}$	$5.474_{\pm 0.001}$	$2.969_{\pm 0.189}$
GRIT	$\mathbf{0.121_{\pm 0.013}}$	$5.091_{\pm 0.158}$	$\mathbf{1.014_{\pm 0.046}}$
PH-DGN	$1.323_{\pm 0.485}$	$5.068_{\pm 0.126}$	$1.627_{\pm 0.398}$
SWAN	$0.896_{\pm 0.232}$	$4.840_{\pm 0.045}$	$\mathbf{1.121_{\pm 0.070}}$
SONAR	$0.527_{\pm 0.254}$	$\mathbf{2.393_{\pm 0.808}}$	$1.239_{\pm 0.151}$
NSD	$\mathbf{0.231_{\pm 0.038}}$	$4.672_{\pm 0.315}$	$1.448_{\pm 0.097}$
CSNN	$0.421_{\pm 0.165}$	$\mathbf{4.522_{\pm 0.048}}$	$\mathbf{1.134_{\pm 0.094}}$
ONDA	$\mathbf{0.085_{\pm 0.019}}$	$\mathbf{0.885_{\pm 0.158}}$	$1.183_{\pm 0.020}$

prediction. It also consistently surpasses SONAR and GraphCON, the two most closely related oscillatory GNN baselines, as well as established sheaf models, including NSD and CSNN. These results show that ONDA's oscillatory sheaf dynamics are particularly effective on tasks designed to stress long-range information propagation.

5.2 Graph Bottleneck Information Transfer: Barbell Benchmark

Following (Bamberger et al., 2025), we consider the Barbell benchmark to assess robustness to over-squashing. This synthetic node-regression task consists of two cliques connected by a single bridge edge and requires information to be transferred between them. Node features are independently sampled from two disjoint distributions, one per clique, and each node must predict the empirical feature mean of the opposite subgraph. Despite the graph diameter being three, every cross-clique signal must traverse the same bridge edge. The benchmark therefore isolates whether a model can aggregate a clique-level statistic and transport it through a severe communication bottleneck. We report nodewise mean-squared error, averaged over test graphs. Under the standard benchmark scaling, an MSE near 1 indicates no effective cross-clique transfer, values between 0.4 and 0.6 indicate partial transfer, and values below 0.25 are considered to solve the task (Bamberger et al., 2025; Hariri et al., 2025).

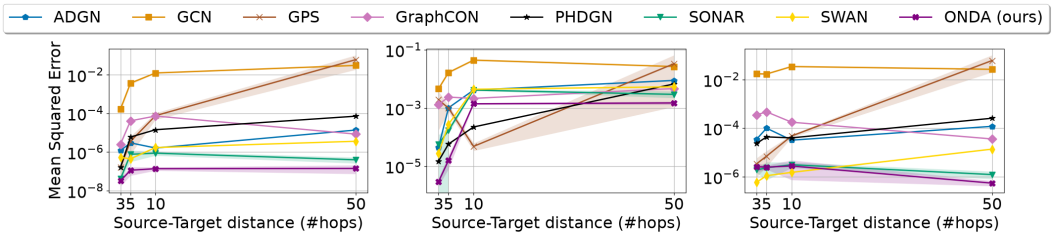
Results. Table 6 summarizes Barbell results, with an extended comparison in Appendix C.1. Standard NSD variants fail to achieve effective cross-clique transfer at any tested graph sizes, underscoring the need for an explicit propagation mechanism inn SNNs. Even CSNN and SONAR, the closest competitors from the sheaf and oscillatory perspectives, respectively, achieve only partial transfer. By contrast, ONDA solves the task near-perfectly at $N = 10$ and $N = 20$. The gap is most revealing at $N = 50$, where ONDA is the only method to satisfy the solving criterion, while all baselines exhibit only limited information transfer.

5.3 Graph Transfer Task

We consider the graph transfer task introduced by (Di Giovanni et al., 2023), following the regression setup of (Gravina et al., 2025), with the aim of assessing model capabilities to propagate information at increasing distances with different topologies and bottlenecks. Nodes are initialized with random features, except for the source and target nodes, whose values are set to 1 and 0, respectively. These nodes are separated by $\kappa \in \{3, 5, 10, 50\}$ hops, and the objective is to swap their values while leaving all other node features unchanged. The task considers line, ring, and crossed-ring graphs, which induce increasing levels of propagation difficulty. We use the same data, hyperparameter search space, and experimental protocol as in (Gravina et al., 2025).

Table 2: **Barbell-clique Results.** Test MSE ($\times 10^{-3}$; $\downarrow$) Results are reported as mean$_{\pm \text{std}}$ over 5 seeds using the best configuration selected by validation. For each column, the **best**, **second-best**, and **third-best** results are highlighted.

Model	$N = 10 \downarrow$	$N = 20 \downarrow$	$N = 50 \downarrow$
GCN	$884.200_{\pm 6.900}$	$961.100_{\pm 12.300}$	$860.000_{\pm 35.000}$
GAT	**$7.900_{\pm 2.100}$**	**$487.000_{\pm 342.200}$**	$852.000_{\pm 29.000}$
GraphSAGE	$455.500_{\pm 199.600}$	$998.200_{\pm 6.500}$	$876.000_{\pm 32.000}$
GIN	$900.500_{\pm 47.900}$	$970.600_{\pm 3.900}$	$985.000_{\pm 123.000}$
GraphTransformer	$670.300_{\pm 306.800}$	$1041.200_{\pm 46.900}$	$831.000_{\pm 13.000}$
ChebNet	**$10.600_{\pm 1.200}$**	$983.600_{\pm 5.100}$	$922.000_{\pm 47.000}$
SONAR	$462.800_{\pm 250.300}$	**$717.500_{\pm 249.600}$**	**$809.000_{\pm 68.000}$**
NSD	$937.200_{\pm 11.400}$	$1005.100_{\pm 7.800}$	$825.000_{\pm 32.000}$
CSNN	$473.900_{\pm 106.200}$	$890.800_{\pm 265.400}$	**$742.000_{\pm 168.000}$**
ONDA	**$0.700_{\pm 0.300}$**	**$0.900_{\pm 0.500}$**	**$83.000_{\pm 85.000}$**



(a) Line (b) Ring (c) Crossed-Ring

Figure 2: Information transfer performance on (a) Line, (b) Ring, and (c) Crossed-Ring graphs.

Results. Figure 2 reports the mean test MSE and standard deviation over three seeds as a function of the distance between source and target nodes. ONDA achieves the lowest error at almost all evaluated distances across the three topologies, while most competing methods progressively degrade as the propagation distance increases. In particular, ONDA consistently improves over non-dissipative GNNs such as A-DGN, SWAN, and PH-DGN, while also outperforming GraphCON and SONAR, which share its oscillatory inductive bias but lack the richer matrix-valued transport provided by the sheaf structure. It also compares favorably with the graph Transformer GPS, despite relying only on local propagation. Overall, these results show that ONDA maintains effective information transfer as the distance between source and target nodes increases.

6 Conclusion

We introduced ONDA, a sheaf neural network that couples learned matrix-valued transport with second-order oscillatory dynamics for effective long-range information propagation. Rather than relying on richer sheaf transport alone, ONDA uses the learned sheaf to determine how information is transformed between stalks, while wave-like dynamics allow these transformed signals to remain influential as they propagate across the graph. Our theoretical analysis shows that connected nodes can influence one another regardless of their distance, and our experiments demonstrate the effectiveness of ONDA across a large variety of tasks, including long-range propagation and short-range over-squashing, where our model sets a new bar for state-of-the-art performance, as well as heterophilic benchmarks.

AI use statement

In this work, we used generative AI tools to aid and polish the writing, to improve the clarity of mathematical exposition, and for routine coding and debugging. We did not use generative AI tools to generate research ideas or make any scientific claims. All AI-assisted material was carefully reviewed, revised, and verified by the authors to ensure accuracy and consistency with the underlying

research. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

Ethics statement

This work does not involve human subjects, personally identifiable information, or the collection of sensitive data. The datasets used in our experiments are publicly available research benchmarks and are used in accordance with their respective licenses and intended research purposes. We are not aware of any significant ethical concerns arising from the proposed methodology or its experimental evaluation. As with other machine learning methods, downstream applications should nevertheless consider potential issues related to bias, fairness, privacy, and misuse. We take responsibility for ensuring that the research is conducted and reported in accordance with established standards of research integrity.

Reproducibility statement

The main paper provides a complete description of the proposed method and experimental setting, while additional architectural details, implementation choices, hyperparameters, dataset information, and experimental configurations are reported in the appendix. We also provide the details required to reproduce the reported baselines, ablations, and evaluation procedures. The source code and configuration files used to produce the experimental results will be publicly released upon acceptance.

References

Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Álvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael M. Bronstein, and Pierre Vandergheynst. On Vanishing Gradients, Over-Smoothing, and Over-Squashing in GNNs: Bridging Recurrent and Graph Learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural network for message diffusion on graphs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=scI9307PLG.

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.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5): 3950–3957, May 2021. doi: 10.1609/aaai.v35i5.16514. URL https://ojs.aaai.org/index.php/AAAI/article/view/16514.

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.

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

Alessio Borgi, Luke Braithwaite, Mario Severino, Emanuele Mule, Fabrizio Silvestri, and Pietro Liò. sheaf-mpnn: A pytorch implementation of sheaf neural networks as message passing. https://github.com/alessioborgi/sheaf-mpnn, 2026. GitHub repository, accessed 22 June 2026.

Xavier Bresson and Thomas Laurent. Residual Gated Graph ConvNets. *arXiv preprint arXiv:1711.07553*, 2018.

Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer, New York, NY, 2012. ISBN 978-1-4614-1939-6. doi: 10.1007/978-1-4614-1939-6.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Andrea Ceni, Alessio Gravina, Claudio Gallicchio, Davide Bacciu, Carola-Bibiane Schonlieb, and Moshe Eliasof. Message-Passing State-Space Models: Improving Graph Learning with Modern Sequence Modeling. *arXiv preprint arXiv:2505.18728*, 2025.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8KYeilT3Ow.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and Deep Graph Convolutional Networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, 13–18 Jul 2020.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

John B Conway. *A course in functional analysis*. Graduate Texts in Mathematics. Springer, New York, NY, 2 edition, 2019. ISBN 978-1-4757-4383-8. doi: 10.1007/978-1-4757-4383-8.

Justin Michael Curry. *Sheaves, cosheaves and applications*. PhD thesis, University of Pennsylvania, 2014.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Liò, and Michael Bronstein. On over-squashing in message passing neural networks: the impact of width, depth, and topology. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 1550–1558, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512201. URL https://doi.org/10.1145/3485447.3512201.

Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

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. *arXiv preprint arXiv:2312.16560*, 2024.

Lawrence C Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, Providence, RI, 2 edition, March 2010.

Joel H Ferziger and Milovan Peric. *Computational methods for fluid dynamics*. Springer, Berlin, Germany, 3 edition, November 2001.

Ben Finkelshtein, Xingyue Huang, Michael M. Bronstein, and Ismail Ilkan Ceylan. Cooperative Graph Neural Networks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ZQcqXCuoxD.

Johannes Gasteiger, Stefan Weiß enberger, and Stephan Günnemann. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for Quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *ICML'17*, pp. 1263–1272. JMLR.org, 2017.

Alessio Gravina, Davide Bacciu, and Claudio Gallicchio. Anti-Symmetric DGN: a stable architecture for Deep Graph Networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=J3Y7cgZOOS.

Alessio Gravina, Moshe Eliasof, Claudio Gallicchio, Davide Bacciu, and Carola-Bibiane Schönlieb. On Oversquashing in Graph Neural Networks Through the Lens of Dynamical Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):16906–16914, Apr. 2025. doi: 10.1609/aaai.v39i16.33858. URL https://ojs.aaai.org/index.php/AAAI/article/view/33858.

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

Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pp. 12252–12267. PMLR, 2023.

Paul R. Halmos. *Finite-Dimensional Vector Spaces*. Undergraduate Texts in Mathematics. Springer, New York, NY, 1974. ISBN 978-1-4612-6387-6. doi: 10.1007/978-1-4612-6387-6.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035. Curran Associates Inc., 2017. ISBN 9781510860964.

Andi Han, Dai Shi, Lequan Lin, and Junbin Gao. From Continuous Dynamics to Graph Neural Networks: Neural Diffusion and Beyond. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=fPQSxjqa2o. Survey Certification.

Jakob Hansen. *Laplacians of cellular sheaves: Theory and applications*. PhD thesis, University of Pennsylvania, 2020.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. In *NeurIPS 2020 Workshop on TDA and Beyond*, 2020. URL https://arxiv.org/abs/2012.06333.

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

Ali Hariri, Alvaro Arroyo, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, Davide Bacciu, Xiaowen Dong, Kamyar Azizzadenesheli, and Pierre Vandergheynst. Return of ChebNet: Understanding and Improving an Overlooked GNN on Long Range Tasks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=oLyfML1Qze.

Simon Heilig, Alessio Gravina, Alessandro Trenta, Claudio Gallicchio, and Davide Bacciu. Port-Hamiltonian Architectural Bias for Long-Range Propagation in Deep Graph Networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=03EkqSCKuO.

Ferran Hernandez Caralt, Guillermo Bernárdez Gil, Iulia Duta, Pietro Liò, and Eduard Alarcón Cot. Joint diffusion processes as an inductive bias in sheaf neural networks. In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pp. 249–263. PMLR, 29 Jul 2024.

Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008. ISBN 978-0-89871-777-8. doi: 10.1137/1.9780898717778.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C. Bayan Bruss, and Tom Goldstein. GOAT: A global transformer on large-scale graphs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17375–17390. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kong23a.html.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13242–13256. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/li22ad.html.

Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, Stan Z. Li, Jian Tang, Guy Wolf, and Stefanie Jegelka. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23321–23337. PMLR, 23–29 Jul 2023.

Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, 62:101695, 2022. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2022.101695. URL https://www.sciencedirect.com/science/article/pii/S1877750322000990.

Alessio Micheli. Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

Luca Miglior, Matteo Tolloso, Alessio Gravina, and Davide Bacciu. Can You Hear Me Now? A Benchmark for Long-Range Graph Propagation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DgkWFPZMPp.

Yaaqov Mishayev, Yonatan Sverdlov, Tal Amir, and Nadav Dym. Short-range oversquashing. In *The Fourth Learning on Graphs Conference*, 2025. URL https://openreview.net/forum?id=rmX8Jamnyg.

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=HhbqHBBrfZ.

Kenta Oono and Taiji Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

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

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35, 2022.

André Ribeiro, Ana Luiza Tenorio, Juan Belieni, Amauri H Souza, and Diego Mesquita. Cooperative sheaf neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AHpexliCTM.

T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, pp. 18888–18909. PMLR, 2022.

T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A Survey on Oversmoothing in Graph Neural Networks. *arXiv preprint arXiv:2303.10993*, 2023.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Allen Dudley Shepard. *A cellular description of the derived category of a stratified space*. Brown University, 1985.

Dai Shi, Andi Han, Lequan Lin, Yi Guo, and Junbin Gao. Exposition on over-squashing problem on GNNs: Current Methods, Benchmarks and Challenges, 2023.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021a. doi: 10.24963/ijcai.2021/214. URL https://doi.org/10.24963/ijcai.2021/214.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021b. doi: 10.24963/ijcai.2021/214. URL https://doi.org/10.24963/ijcai.2021/214. Main Track.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pp. 31613–31632. PMLR, 2023.

Joshua Southern, Francesco Di Giovanni, Michael M. Bronstein, and Johannes F. Lutzeyer. Understanding virtual nodes: Oversquashing and node heterogeneity. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NmcOAwRyH5.

Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Soc., 2012. doi: 10.1090/gsm/140.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Alessandro Trenta, Alessio Gravina, and Davide Bacciu. SONAR: Long-Range Graph Propagation Through Information Waves. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 145054–145087. Curran Associates, Inc., 2025. doi: 10.52202/085713-4854.

A. Vaswani et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23341–23362. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wang22am.html.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

Olga Zaghen, Antonio Longa, Steve Azzolin, Lev Telyatnikov, Andrea Passerini, and Pietro Liò. Sheaf diffusion goes nonlinear: Enhancing gnns with adaptive sheaf laplacians. In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pp. 264–276. PMLR, 29 Jul 2024.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7793–7804. Curran Associates, Inc., 2020.

Jiong Zhu, Ryan A. Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K. Ahmed, and Danai Koutra. Graph neural networks with heterophily. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11168–11176, May 2021. doi: 10.1609/aaai.v35i12.17332. URL https://ojs.aaai.org/index.php/AAAI/article/view/17332.

A Mathematical Background and Proofs

This appendix provides the mathematical background and proofs for the theoretical results presented throughout Section 4. First, Appendix A.1 gathers some additional notation and conventions on inner products. Section A.2 establishes the spectral bound for the normalized sheaf Laplacian and includes a description of the directed sheaf operator. Appendix A.3 derives the closed-form solution of the continuous sheaf wave equation, while Appendices A.4 and A.5 prove the pathwise and

spectral representations of the source-to-target response, respectively. Finally, Appendix A.6 proves the discrete-time result from Theorem 1, and Corollary 1 shows that the discrete response converges to the continuous response as the step size tends to zero. Throughout, all vector spaces are over $\mathbb{R}$.

A.1 Preliminaries on inner products and norms

We recall basic definitions and facts concerning inner products and norms on finite-dimensional spaces. This material can be found in standard introductory textbooks to functional analysis, see for instance (Conway, 2019). For each vertex $u \in \mathcal{V}$, denote by $\langle \cdot, \cdot \rangle_{\mathcal{F}(u)}$ a fixed real inner product on $\mathcal{F}(u)$ and write $\|\boldsymbol{X}\|^2_{\mathcal{F}(u)} = \langle \boldsymbol{X}, \boldsymbol{X} \rangle_{\mathcal{F}(u)}$ for $\boldsymbol{X} \in \mathcal{F}(u)$. The cochain space $\mathcal{H} = C^0(\mathcal{G}; \mathcal{F})$ carries the direct sum inner product, defined by $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_{\mathcal{H}} := \sum_{u \in \mathcal{V}} \langle \boldsymbol{X}_u, \boldsymbol{Y}_u \rangle_{\mathcal{F}(u)}$ and $\|\boldsymbol{X}\|^2_{\mathcal{H}} = \sum_{u \in \mathcal{V}} \|\boldsymbol{X}_u\|^2_{\mathcal{F}(u)}$. All matrix representations are taken with respect to fixed orthonormal bases of the vertex and edge stalks.

For a linear map $A : F \to G$ between finite-dimensional inner-product spaces, we use the *induced operator norm*, defined as

$$\|A\| := \sup_{\|\boldsymbol{X}\|_F \leq 1} \|A\boldsymbol{X}\|_G.$$

Since we consider finite-dimensional vector spaces, this is a finite quantity and satisfies $\|A\boldsymbol{X}\|_G \leq \|A\|\|\boldsymbol{X}\|_F$. Additionally, for $c \in \mathbb{R}$, $A, B \in \mathrm{Hom}_{\mathbb{R}}(F, G)$, $D \in \mathrm{Hom}_{\mathbb{R}}(E, F)$, the following properties hold:

$$\|cA\| = |c|\|A\|, \qquad \|A + B\| \leq \|A\| + \|B\|, \qquad \|AD\| \leq \|A\|\|D\|.$$

Recall that a sequence $A_n : F \to G$ *converges to* A *in operator norm* if $\|A_n - A\| \to 0$. An operator series $\sum_{n=0}^{\infty} A_n$ *converges absolutely* in operator norm if $\sum_{n=0}^{\infty} \|A_n\| < \infty$. Since F and G are finite-dimensional, the space of linear maps $F \to G$ is complete in the operator norm. Thus, absolute convergence guarantees convergence of the partial sums in operator norm.

For $A, B \in \mathrm{Hom}_{\mathbb{R}}(F, G)$, we also use the *Hilbert–Schmidt inner product*

$$\langle A, B \rangle_{\mathrm{HS}} := \mathrm{tr}(A^* B), \qquad \|A\|^2_{\mathrm{HS}} := \langle A, A \rangle_{\mathrm{HS}}.$$

Since $\mathrm{Hom}_{\mathbb{R}}(F, G)$ is finite-dimensional, all norms on this space are equivalent. Thus, convergence of operator series is the same in the operator and Hilbert–Schmidt norms, as is absolute convergence.

A.2 Sheaf Laplacian and Spectral bounds

Recall that we denote by $\mathcal{D}_{\mathcal{F}}$ the block-diagonal part of $L_{\mathcal{F}}$, given by $(\mathcal{D}_{\mathcal{F}})_{vv} = \sum_{e \ni v} \mathcal{F}^*_{v,e} \mathcal{F}_{v,e}$. The symmetrically normalized sheaf Laplacian is given by $\Delta_{\mathcal{F}} = \mathcal{D}_{\mathcal{F}}^{-1/2} L_{\mathcal{F}} \mathcal{D}_{\mathcal{F}}^{-1/2}$. Here, $\mathcal{D}_{\mathcal{F}}^{-1/2}$ denotes the Moore–Penrose pseudoinverse of $\mathcal{D}_{\mathcal{F}}^{1/2}$. It acts by $\mu^{-1/2}$ on each eigenspace of $\mathcal{D}_{\mathcal{F}}$ with eigenvalue $\mu > 0$, and by zero on $\ker \mathcal{D}_{\mathcal{F}}$. We recall the following well-known fact, presented here with a self-contained elementary proof. A more general version of this result can be found in (Hansen & Ghrist, 2019).

Lemma 1. *The symmetrically normalized sheaf Laplacian $\Delta_{\mathcal{F}}$ is self-adjoint and all of its eigenvalues lie in* $[0, 2]$.

Proof. Since $L_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{F}}^{-1/2}$ are self-adjoint, also $\Delta^*_{\mathcal{F}} = \mathcal{D}_{\mathcal{F}}^{-1/2} L^*_{\mathcal{F}} \mathcal{D}_{\mathcal{F}}^{-1/2} = \Delta_{\mathcal{F}}$.

Let $(q_j)_j$ be an orthonormal eigenbasis of $\mathcal{D}_{\mathcal{F}}$, with eigenvalues $\mu_j \geq 0$. For $\boldsymbol{X} = \sum_j \alpha_j q_j$, we have

$$\boldsymbol{Y} = \mathcal{D}_{\mathcal{F}}^{-1/2} \boldsymbol{X} = \sum_{\mu_j > 0} \frac{\alpha_j}{\sqrt{\mu_j}} q_j, \qquad \langle \boldsymbol{Y}, \mathcal{D}_{\mathcal{F}} \boldsymbol{Y} \rangle = \sum_{\mu_j > 0} |\alpha_j|^2 \leq \sum_j |\alpha_j|^2 = \|\boldsymbol{X}\|^2.$$

Substituting $\boldsymbol{Y} = \mathcal{D}_{\mathcal{F}}^{-1/2}\boldsymbol{X}$, we obtain

$$\begin{aligned} 0 \leq \langle \boldsymbol{X}, \Delta_{\mathcal{F}}\boldsymbol{X}\rangle = \langle \boldsymbol{Y}, L_{\mathcal{F}}\boldsymbol{Y}\rangle &= \sum_{e=\{u,v\}\in\mathcal{E}} \|\mathcal{F}_{u,e}\boldsymbol{Y}_u - \mathcal{F}_{v,e}\boldsymbol{Y}_v\|^2 \\ &\leq 2\sum_{e=\{u,v\}\in\mathcal{E}} \left(\|\mathcal{F}_{u,e}\boldsymbol{Y}_u\|^2 + \|\mathcal{F}_{v,e}\boldsymbol{Y}_v\|^2\right) \\ &= 2\sum_{v\in\mathcal{V}}\sum_{e\ni v}\langle \boldsymbol{Y}_v, \mathcal{F}_{v,e}^*\mathcal{F}_{v,e}\boldsymbol{Y}_v\rangle = 2\langle \boldsymbol{Y}, \mathcal{D}_{\mathcal{F}}\boldsymbol{Y}\rangle \leq 2\|\boldsymbol{X}\|^2, \end{aligned}$$

where the inequality in the second line uses $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, and the final inequality uses the bound on $\langle \boldsymbol{Y}, \mathcal{D}_{\mathcal{F}}\boldsymbol{Y}\rangle$ established above. Hence, for any eigenvalue λ with non-zero eigenvector $\boldsymbol{X}$, we find $0 \leq \lambda = \frac{\langle \boldsymbol{X}, \Delta_{\mathcal{F}}\boldsymbol{X}\rangle}{\|\boldsymbol{X}\|^2} \leq 2$. □

Directed variant. Here we give the precise description of the directed version of the sheaf Laplacian as mentioned in Section 3.2. For each edge $e = \{u, v\}$, consider two oriented copies $e_u = (u \leftarrow v)$ and $e_v = (v \leftarrow u)$. Each copy e_z, $z \in \{u, v\}$, carries an edge stalk $\mathcal{F}(e_z)$ and independently learned restriction maps $\mathcal{F}_{w,e_z} : \mathcal{F}(w) \to \mathcal{F}(e_z)$, $w \in \{u, v\}$. The nodewise expression is given by

$$(L_{\mathcal{F}}^{\mathrm{dir}}\boldsymbol{X})_v = \sum_{e=\{u,v\}\in\mathcal{E}} \mathcal{F}_{v,e_v}^*\left(\mathcal{F}_{v,e_v}\boldsymbol{X}_v - \mathcal{F}_{u,e_v}\boldsymbol{X}_u\right).$$

This operator is not guaranteed to be self-adjoint. Identifying $\mathcal{F}(e_u) = \mathcal{F}(e_v) = \mathcal{F}(e)$ and forcing $\mathcal{F}_{w,e_u} = \mathcal{F}_{w,e_v} = \mathcal{F}_{w,e}$ for $w \in \{u, v\}$ recovers $L_{\mathcal{F}}$.

A.3 Solution of the sheaf wave equation

We derive the solution of Equation (5) by rewriting the second-order dynamics as a first-order linear system and computing its operator exponential.

Lemma 2. *Let $L_{\mathcal{F}}$ be fixed. The unique solution of*

$$\ddot{\boldsymbol{X}}(t) + L_{\mathcal{F}}\boldsymbol{X}(t) = 0,$$

with initial conditions $\boldsymbol{X}(0) = \overline{\boldsymbol{X}}$, $\dot{\boldsymbol{X}}(0) = \overline{\boldsymbol{V}}$, is given by $\boldsymbol{X}(t) = \mathcal{C}_{\mathcal{F}}(t)\overline{\boldsymbol{X}} + \mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}}$.

Proof. For a time-dependent node state $\boldsymbol{X}(t) \in C^0(\mathcal{G}; \mathcal{F})$ with velocity $\boldsymbol{V}(t) = \dot{\boldsymbol{X}}(t)$, we follow usual terminology of second-order dynamics (Teschl, 2012) and introduce the *phase-space state*

$$\boldsymbol{Z}(t) := \begin{pmatrix} \boldsymbol{X}(t) \\ \dot{\boldsymbol{X}}(t) \end{pmatrix} \in \mathcal{H} \oplus \mathcal{H}.$$

In this notation, our problem becomes equivalent to the first-order system

$$\dot{\boldsymbol{Z}}(t) = G_{\mathcal{F}}\boldsymbol{Z}(t), \qquad G_{\mathcal{F}} := \begin{pmatrix} 0 & I_{\mathcal{H}} \\ -L_{\mathcal{F}} & 0 \end{pmatrix}, \qquad \boldsymbol{Z}(0) = \begin{pmatrix} \overline{\boldsymbol{X}} \\ \overline{\boldsymbol{V}} \end{pmatrix}. \tag{13}$$

Since $\mathcal{H} \oplus \mathcal{H}$ is finite-dimensional, $G_{\mathcal{F}}$ is bounded. Its exponential is defined by the absolutely convergent power series $e^{tG_{\mathcal{F}}} := \sum_{n=0}^{\infty} \frac{t^n}{n!} G_{\mathcal{F}}^n$. Since

$$G_{\mathcal{F}}^2 = -\begin{pmatrix} L_{\mathcal{F}} & 0 \\ 0 & L_{\mathcal{F}} \end{pmatrix},$$

the even and odd powers in the above power series give

$$\begin{aligned} e^{tG_{\mathcal{F}}} &= \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k}}{(2k)!}\begin{pmatrix} L_{\mathcal{F}}^k & 0 \\ 0 & L_{\mathcal{F}}^k \end{pmatrix} + \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k+1}}{(2k+1)!}\begin{pmatrix} 0 & L_{\mathcal{F}}^k \\ -L_{\mathcal{F}}^{k+1} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{C}_{\mathcal{F}}(t) & \mathcal{S}_{\mathcal{F}}(t) \\ -L_{\mathcal{F}}\mathcal{S}_{\mathcal{F}}(t) & \mathcal{C}_{\mathcal{F}}(t) \end{pmatrix}. \end{aligned}$$

The power series defining $\mathcal{C}_{\mathcal{F}}$ and $\mathcal{S}_{\mathcal{F}}$ have infinite radius of convergence and can therefore be differentiated termwise. Hence

$$\mathcal{C}'_{\mathcal{F}}(t) = \sum_{k=1}^{\infty}(-1)^k \frac{t^{2k-1}}{(2k-1)!} L_{\mathcal{F}}^k = -L_{\mathcal{F}} \sum_{j=0}^{\infty}(-1)^j \frac{t^{2j+1}}{(2j+1)!} L_{\mathcal{F}}^j = -L_{\mathcal{F}}\mathcal{S}_{\mathcal{F}}(t),$$

$$\mathcal{S}'_{\mathcal{F}}(t) = \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k}}{(2k)!} L_{\mathcal{F}}^k = \mathcal{C}_{\mathcal{F}}(t).$$

Therefore, for $\boldsymbol{X}(t) := \mathcal{C}_{\mathcal{F}}(t)\overline{\boldsymbol{X}} + \mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}}$, we find that

$$\ddot{\boldsymbol{X}}(t) = -L_{\mathcal{F}}\mathcal{C}_{\mathcal{F}}(t)\overline{\boldsymbol{X}} - L_{\mathcal{F}}\mathcal{S}_{\mathcal{F}}(t)\overline{\boldsymbol{V}} = -L_{\mathcal{F}}\boldsymbol{X}(t).$$

Moreover, the above identities at $t = 0$ give that $\boldsymbol{X}(0) = \overline{\boldsymbol{X}}$, $\dot{\boldsymbol{X}}(0) = \overline{\boldsymbol{V}}$. Hence $\boldsymbol{X}(t)$ solves our problem. Uniqueness follows by the standard existence-and-uniqueness theorem for finite-dimensional linear ordinary differential equations. □

A.4 Proof of Proposition 1

We first express the blocks of $L_{\mathcal{F}}^k$ as sums of compositions along length-k walks.

Recall that, for $k \geq 0$ and vertices $u, v \in \mathcal{V}$, we denoted by $\mathcal{W}_k(u, v)$ the set of length-k walks $\omega = (v_0 = u, v_1, \ldots, v_k = v)$, where either $v_j = v_{j-1}$ or $\{v_{j-1}, v_j\} \in \mathcal{E}$ at each step. To each such $\omega \in \mathcal{W}_k(u, v)$, we associated

$$(L_{\mathcal{F}})_{\omega} = (L_{\mathcal{F}})_{v_k v_{k-1}} \ldots (L_{\mathcal{F}})_{v_1 v_0} : \mathcal{F}(u) \to \mathcal{F}(v).$$

Allowing steps with $v_j = v_{j-1}$ accounts for the diagonal blocks of $L_{\mathcal{F}}$ in these compositions. For $k = 0$, we put $\mathcal{W}_0(u, u) = \{(u)\}$, with empty composition equal to $I_{\mathcal{F}(u)}$.

Lemma 3. *For every $k \geq 0$ and $u, v \in \mathcal{V}$,*

$$(L_{\mathcal{F}}^k)_{vu} = \sum_{\omega \in \mathcal{W}_k(u,v)} (L_{\mathcal{F}})_{\omega} : \mathcal{F}(u) \to \mathcal{F}(v).$$

Proof. We proceed by induction on k.

For $k = 0$, the result is clear. Suppose the result holds for some $k \geq 0$. Then

$$(L_{\mathcal{F}}^{k+1})_{vu} = \sum_{a \in \mathcal{V}} (L_{\mathcal{F}})_{va} (L_{\mathcal{F}}^k)_{au} = \sum_{a \in \mathcal{V}} \sum_{\omega \in \mathcal{W}_k(u,a)} (L_{\mathcal{F}})_{va} (L_{\mathcal{F}})_{\omega}.$$

Since $(L_{\mathcal{F}})_{va} = 0$ whenever $a \neq v$ and $\{a, v\} \notin \mathcal{E}$, only those a for which $a = v$ or $\{a, v\} \in \mathcal{E}$ can contribute. For every such term, if $\omega = (v_0 = u, \ldots, v_k = a) \in \mathcal{W}_k(u, a)$, appending v gives the walk $\widetilde{\omega} = (v_0 = u, \ldots, v_k = a, v_{k+1} = v) \in \mathcal{W}_{k+1}(u, v)$, with associated composition

$$(L_{\mathcal{F}})_{\widetilde{\omega}} = (L_{\mathcal{F}})_{va} (L_{\mathcal{F}})_{\omega}.$$

Hence, $(L_{\mathcal{F}}^{k+1})_{vu} = \sum_{\omega \in \mathcal{W}_{k+1}(u,v)} (L_{\mathcal{F}})_{\omega}$, completing the induction. □

For two different nodes v, u, we define their *distance*, denoted by $\mathrm{dist}_{\mathcal{G}}(u, v)$, as the minimum $k \geq 0$ such that $\mathcal{W}_k(u, v) \neq \emptyset$ if they lie in the same connected component, and $\mathrm{dist}_{\mathcal{G}}(u, v) = \infty$ otherwise. Also, $\mathrm{dist}_{\mathcal{G}}(u, u) = 0$.

Proof of Proposition 1 . For $r = \mathrm{dist}_{\mathcal{G}}(u, v) < \infty$, we have $\mathcal{W}_k(u, v) = \emptyset$ for $k < r$, and hence $(L_{\mathcal{F}}^k)_{vu} = 0$ for these values of k. Taking the (v, u)-block of the series defining $\mathcal{C}_{\mathcal{F}}(t)$ gives

$$J^{\mathcal{F}}_{v \leftarrow u}(t) = \sum_{k=0}^{\infty}(-1)^k \frac{t^{2k}}{(2k)!} (L_{\mathcal{F}}^k)_{vu} = \sum_{k=r}^{\infty}(-1)^k \frac{t^{2k}}{(2k)!} \sum_{\omega \in \mathcal{W}_k(u,v)} (L_{\mathcal{F}})_{\omega}.$$

At $k = r$, these walks are precisely the shortest paths. Indeed, a stationary step or a repeated vertex would allow a segment to be removed and thereby produce a shorter walk from u to v, contrary to the definition of r.

For the last statement, if u and v lie in different connected components, then $\mathcal{W}_k(u, v) = \emptyset$ for every k. Every block $(L_{\mathcal{F}}^k)_{vu}$ therefore vanishes, and $J^{\mathcal{F}}_{v \leftarrow u}(t) = 0$ for all t. □

A.5 Proof of Proposition 2

This result applies to the positive semidefinite self-adjoint sheaf Laplacian and uses the orthogonal projectors onto the eigenspaces P_λ indexed by distinct eigenvalues given by the finite-dimensional spectral theorem (Halmos, 1974). These projectors satisfy $L_{\mathcal{F}}P_\lambda = \lambda P_\lambda$, as well as the identities $P_\lambda^* = P_\lambda,\ P_\lambda P_\mu = \delta_{\lambda\mu}P_\lambda$, and $\sum_{\lambda\in\mathrm{spec}(L_{\mathcal{F}})} P_\lambda = I_{\mathcal{H}}$.

Proof of Proposition 2 . Substituting the above identities into the power series gives

$$\begin{aligned}\mathcal{C}_{\mathcal{F}}(t)P_\lambda &= \sum_{k=0}^{\infty}(-1)^k\frac{t^{2k}}{(2k)!}L_{\mathcal{F}}^k P_\lambda \\ &= \left(\sum_{k=0}^{\infty}(-1)^k\frac{t^{2k}\lambda^k}{(2k)!}\right)P_\lambda = \cos(t\sqrt{\lambda})P_\lambda.\end{aligned}$$

Since the spectral projectors sum to $I_{\mathcal{H}}$, also

$$\mathcal{C}_{\mathcal{F}}(t) = \mathcal{C}_{\mathcal{F}}(t)\sum_\lambda P_\lambda = P_0 + \sum_{\lambda>0}\cos(t\sqrt{\lambda})P_\lambda,$$

where P_0 is the projector onto $\ker L_{\mathcal{F}}$, with $P_0 = 0$ when $0 \notin \mathrm{spec}(L_{\mathcal{F}})$. Applying π_v on the left and ι_u on the right proves the result. □

A.6 Proof of Theorem 1

For the proof of this result, we start with two preliminary lemmas. First, Lemma 4 expresses the discrete response as a polynomial in $L_{\mathcal{F}}$. Afterwards, we solve the corresponding scalar recurrence to obtain the spectral formula and boundedness in Lemma 5. The spectral formula and boundedness in Theorem 1 then follow directly. To prove that a response which is non-zero at some step cannot converge to zero, we compute the Cesàro mean of its squared Hilbert–Schmidt norm. Finally, Corollary 1 shows that the discrete spectral coefficients converge to their continuous counterparts as $h \to 0$ with $\ell h = T$ fixed.

Let $(\boldsymbol{X}^{(\ell)}, \boldsymbol{V}^{(\ell)})$ denote the underlying trajectory of Equation (3) with fixed restriction maps, identity weights, and no dissipation or external forcing, i.e.,

$$\boldsymbol{V}^{(\ell+1)} = \boldsymbol{V}^{(\ell)} - hL_{\mathcal{F}}\boldsymbol{X}^{(\ell)}, \qquad \boldsymbol{X}^{(\ell+1)} = \boldsymbol{X}^{(\ell)} + h\boldsymbol{V}^{(\ell+1)}, \tag{14}$$

and let $(\widetilde{\boldsymbol{X}}^{(\ell)}, \widetilde{\boldsymbol{V}}^{(\ell)})$ denote the trajectory obtained by replacing $\overline{\boldsymbol{X}}$ with $\overline{\boldsymbol{X}} + \iota_u\boldsymbol{W}$, while keeping the initial velocity fixed. Write

$$\Delta\boldsymbol{X}^{(\ell)} := \widetilde{\boldsymbol{X}}^{(\ell)} - \boldsymbol{X}^{(\ell)}, \qquad \Delta\boldsymbol{V}^{(\ell)} := \widetilde{\boldsymbol{V}}^{(\ell)} - \boldsymbol{V}^{(\ell)},$$

so that $\Delta\overline{\boldsymbol{X}} = \iota_u\boldsymbol{W}$ and $\Delta\overline{\boldsymbol{V}} = 0$. The following lemma expresses the resulting response as a polynomial in $L_{\mathcal{F}}$.

Lemma 4. *Let $p_{0,h}(z) = 1$, $p_{1,h}(z) = 1 - h^2 z$, and, for $\ell \geq 1$,*

$$p_{\ell+1,h}(z) = (2 - h^2 z)p_{\ell,h}(z) - p_{\ell-1,h}(z).$$

Then, for every integer $\ell \geq 0$, $\Delta\boldsymbol{X}^{(\ell)} = p_{\ell,h}(L_{\mathcal{F}})\iota_u\boldsymbol{W}$, and $J^{\mathcal{F}}_{v\leftarrow u}[\ell, h] = \big(p_{\ell,h}(L_{\mathcal{F}})\big)_{vu}$, where $\deg p_{\ell,h} \leq \ell$. In particular, if $\mathrm{dist}_{\mathcal{G}}(u, v) > \ell$, then $J^{\mathcal{F}}_{v\leftarrow u}[\ell, h] = 0$.

Proof. Taking differences in the update Equation (14) gives

$$\Delta\boldsymbol{V}^{(\ell+1)} = \Delta\boldsymbol{V}^{(\ell)} - hL_{\mathcal{F}}\Delta\boldsymbol{X}^{(\ell)}, \qquad \Delta\boldsymbol{X}^{(\ell+1)} = \Delta\boldsymbol{X}^{(\ell)} + h\Delta\boldsymbol{V}^{(\ell+1)}.$$

For $\ell = 0$, this becomes $\Delta\boldsymbol{X}^{(1)} = (I_{\mathcal{H}} - h^2 L_{\mathcal{F}})\Delta\boldsymbol{X}^{(0)} = p_{1,h}(L_{\mathcal{F}})\iota_u\boldsymbol{W}$.

For $\ell \geq 1$, the above update gives $h\Delta\boldsymbol{V}^{(\ell)} = \Delta\boldsymbol{X}^{(\ell)} - \Delta\boldsymbol{X}^{(\ell-1)}$. We substitute this into the next one, giving

$$\Delta\boldsymbol{X}^{(\ell+1)} = \Delta\boldsymbol{X}^{(\ell)} + h\Delta\boldsymbol{V}^{(\ell)} - h^2 L_{\mathcal{F}}\Delta\boldsymbol{X}^{(\ell)} = (2I_{\mathcal{H}} - h^2 L_{\mathcal{F}})\Delta\boldsymbol{X}^{(\ell)} - \Delta\boldsymbol{X}^{(\ell-1)}.$$

Together with the initial values, this recurrence implies by induction that $\Delta \boldsymbol{X}^{(\ell)} = p_{\ell,h}(L_{\mathcal{F}})\iota_u \boldsymbol{W}$. After applying π_v, and using that the identity holds for every $\boldsymbol{W} \in \mathcal{F}(u)$, this gives $J^{\mathcal{F}}_{v\leftarrow u}[\ell, h] = \pi_v p_{\ell,h}(L_{\mathcal{F}})\iota_u = \big(p_{\ell,h}(L_{\mathcal{F}})\big)_{vu}$.

The recurrence defining $p_{\ell,h}$ directly implies that $\deg p_{\ell,h} \le \ell$.

If $\mathrm{dist}_{\mathcal{G}}(u, v) > \ell$, then $\mathcal{W}_j(u, v) = \emptyset$ for every $j \le \ell$. By Lemma 3, $(L^j_{\mathcal{F}})_{vu} = 0$ for all such j, and hence $J^{\mathcal{F}}_{v\leftarrow u}[\ell, h] = \big(p_{\ell,h}(L_{\mathcal{F}})\big)_{vu} = 0$. □

We next solve the scalar recurrence defining $p_{\ell,h}$. For this and the subsequent proof, we use the following trigonometric identities, valid for $a, b, \theta \in \mathbb{R}$:

$$\cos(a + b) + \cos(a - b) = 2\cos(a)\cos(b), \tag{15}$$

$$\sin(a + b) - \sin(a - b) = 2\cos(a)\sin(b), \tag{16}$$

$$\cos\theta = 1 - 2\sin^2\left(\frac{\theta}{2}\right), \qquad \cos^2\theta = \frac{1 + \cos(2\theta)}{2}. \tag{17}$$

Lemma 5. *Let $h > 0$ and $\lambda \ge 0$ satisfy $h^2\lambda < 4$, and define $\theta_h(\lambda) := 2\arcsin\left(\frac{h\sqrt{\lambda}}{2}\right)$. Then, for every integer $\ell \ge 0$,*

$$p_{\ell,h}(\lambda) = \frac{\cos\left((\ell + \frac{1}{2})\theta_h(\lambda)\right)}{\cos\left(\theta_h(\lambda)/2\right)}.$$

Moreover, $|p_{\ell,h}(\lambda)| \le \frac{1}{\sqrt{1-h^2\lambda/4}}$.

Proof. Since $0 \le h\sqrt{\lambda}/2 < 1$, we have $\theta_h(\lambda) \in [0, \pi)$. By definition of $\theta_h(\lambda)$, also $\sin\left(\frac{\theta_h(\lambda)}{2}\right) = \frac{h\sqrt{\lambda}}{2}$. Using Equation (17), we find

$$2\cos\theta_h(\lambda) = 2\left(1 - 2\sin^2\left(\frac{\theta_h(\lambda)}{2}\right)\right) = 2 - h^2\lambda. \tag{18}$$

Define

$$q_\ell := \frac{\cos((\ell + \frac{1}{2})\theta_h(\lambda))}{\cos(\theta_h(\lambda)/2)}.$$

For $\ell = 0$, $q_0 = 1 = p_{0,h}(\lambda)$.

Applying Equation (15) to $a = \theta_h(\lambda),\ b = \theta_h(\lambda)/2$ and dividing by $\cos(\theta_h(\lambda)/2)$ gives

$$\frac{\cos(3\theta_h(\lambda)/2)}{\cos(\theta_h(\lambda)/2)} = 2\cos\theta_h(\lambda) - 1,$$

so that $q_1 = 2\cos\theta_h(\lambda) - 1 = 1 - h^2\lambda = p_{1,h}(\lambda)$, where we used Equation (18) and the definition of $p_{1,h}(\lambda)$.

For $\ell \ge 1$, again applying Equation (15) with $a = (\ell + \frac{1}{2})\theta_h(\lambda)$ and $b = \theta_h(\lambda)$ gives

$$\cos\left((\ell + \tfrac{3}{2})\theta_h(\lambda)\right) + \cos\left((\ell - \tfrac{1}{2})\theta_h(\lambda)\right) = 2\cos\theta_h(\lambda)\cos\left((\ell + \tfrac{1}{2})\theta_h(\lambda)\right).$$

Dividing this expression by $\cos(\theta_h(\lambda)/2)$, and using the definition of q_ℓ, we find

$$q_{\ell+1} + q_{\ell-1} = 2\cos\theta_h(\lambda)\, q_\ell.$$

By Equation (18), this can be rearranged to $q_{\ell+1} = 2\cos\theta_h(\lambda)q_\ell - q_{\ell-1} = (2 - h^2\lambda)q_\ell - q_{\ell-1}$.

In particular, q_ℓ has the same initial values and recurrence as $p_{\ell,h}(\lambda)$, and hence

$$p_{\ell,h}(\lambda) = \frac{\cos\left((\ell + \frac{1}{2})\theta_h(\lambda)\right)}{\cos\left(\theta_h(\lambda)/2\right)}.$$

For the final statement, we use Equations (17) and (18) to find

$$\cos^2(\theta_h(\lambda)/2) = \frac{1+\cos\theta_h(\lambda)}{2} = 1 - \frac{h^2\lambda}{4}. \tag{19}$$

Hence, $\cos\left(\frac{\theta_h(\lambda)}{2}\right) = \sqrt{1-\frac{h^2\lambda}{4}} > 0$, and $|p_{\ell,h}(\lambda)| \le \frac{1}{\cos(\theta_h(\lambda)/2)} = \frac{1}{\sqrt{1-h^2\lambda/4}}$. □

Proof of Theorem 1 . By Lemma 4 and the spectral decomposition of $L_{\mathcal{F}}$,

$$J^{\mathcal{F}}_{v\leftarrow u}[\ell,h] = \big(p_{\ell,h}(L_{\mathcal{F}})\big)_{vu} = \sum_{\lambda} p_{\ell,h}(\lambda)(P_\lambda)_{vu}.$$

Since $h^2\lambda_{\max}(L_{\mathcal{F}}) < 4$, Lemma 5 gives Equation (12).

The fact that $J^{\mathcal{F}}_{v\leftarrow u}[\ell,h] = 0$ whenever $\operatorname{dist}_{\mathcal{G}}(u,v) > \ell$ follows again from Lemma 4.

The bound in Lemma 5 gives

$$\big\|J^{\mathcal{F}}_{v\leftarrow u}[\ell,h]\big\| \le \sum_{\lambda} \frac{\|(P_\lambda)_{vu}\|}{\sqrt{1-h^2\lambda/4}}.$$

The right-hand side is finite under the step-size assumption and independent of ℓ, showing that the response is bounded as the propagation depth varies.

For the final statement, fix $u, v \in \mathcal{V}$ and write $J_\ell := J^{\mathcal{F}}_{v\leftarrow u}[\ell,h]$. To exclude that $J_\ell \longrightarrow 0$, it suffices to show that the Cesàro average $\frac{1}{N}\sum_{\ell=0}^{N-1}\|J_\ell\|^2_{\mathrm{HS}}$ has a strictly positive limit whenever the response is non-zero at some step.

We use the Hilbert–Schmidt norm as introduced in Section A.1 to expand the inner products

$$\begin{aligned}
\|J_\ell\|^2_{\mathrm{HS}} &= \|(P_0)_{vu}\|^2_{\mathrm{HS}} + 2\sum_{\lambda>0} p_{\ell,h}(\lambda)\,\langle (P_0)_{vu}, (P_\lambda)_{vu}\rangle_{\mathrm{HS}} \\
&\quad + \sum_{\lambda,\mu>0} p_{\ell,h}(\lambda)p_{\ell,h}(\mu)\,\langle (P_\lambda)_{vu}, (P_\mu)_{vu}\rangle_{\mathrm{HS}} \\
&= \|(P_0)_{vu}\|^2_{\mathrm{HS}} + 2\sum_{\lambda>0} p_{\ell,h}(\lambda)\,\langle (P_0)_{vu}, (P_\lambda)_{vu}\rangle_{\mathrm{HS}} + \sum_{\lambda>0} p_{\ell,h}(\lambda)^2\|(P_\lambda)_{vu}\|^2_{\mathrm{HS}} \\
&\quad + 2\sum_{0<\lambda<\mu} p_{\ell,h}(\lambda)p_{\ell,h}(\mu)\,\langle (P_\lambda)_{vu}, (P_\mu)_{vu}\rangle_{\mathrm{HS}}\,.
\end{aligned}$$

The last equality separates the diagonal terms $\lambda = \mu$ and uses symmetry of the real Hilbert–Schmidt inner product.

We analyze the three oscillatory sums separately, using a shared averaging identity. For fixed $\omega \notin 2\pi\mathbb{Z}$, applying Equation (16) with $a = (\ell+\frac{1}{2})\omega$ and $b = \omega/2$ gives

$$\sin((\ell+1)\omega) - \sin(\ell\omega) = 2\sin(\omega/2)\cos((\ell+\tfrac{1}{2})\omega).$$

Summing over $\ell = 0,\dots,N-1$, the left-hand side telescopes. Consequently,

$$T_N(\omega) := \frac{1}{N}\sum_{\ell=0}^{N-1}\cos((\ell+\tfrac{1}{2})\omega) = \frac{\sin(N\omega)}{2N\sin(\omega/2)} \longrightarrow 0. \tag{20}$$

Here $\sin(\omega/2) \neq 0$ is fixed, while $|\sin(N\omega)| \le 1$.

Term 1. For $\lambda > 0$, we have $\theta_h(\lambda) \in (0,\pi)$. Lemma 5 and Equation (20) therefore give

$$\frac{1}{N}\sum_{\ell=0}^{N-1} p_{\ell,h}(\lambda) = \frac{T_N(\theta_h(\lambda))}{\cos(\theta_h(\lambda)/2)} \longrightarrow 0.$$

Term 2. For $\lambda > 0$, using Equation (17), we obtain

$$\begin{aligned}
\frac{1}{N}\sum_{\ell=0}^{N-1} p_{\ell,h}(\lambda)^2 &= \frac{1}{N\cos^2(\theta_h(\lambda)/2)}\sum_{\ell=0}^{N-1}\cos^2((\ell+\tfrac{1}{2})\theta_h(\lambda)) \\
&= \frac{1+T_N(2\theta_h(\lambda))}{2\cos^2(\theta_h(\lambda)/2)} \longrightarrow \frac{1}{2\cos^2(\theta_h(\lambda)/2)}.
\end{aligned}$$

Here $T_N(2\theta_h(\lambda)) \to 0$ because of Equation (20) as $2\theta_h(\lambda) \in (0, 2\pi)$.

Term 3. For distinct positive eigenvalues λ, μ, applying Equation (15) gives

$$\frac{1}{N}\sum_{\ell=0}^{N-1} p_{\ell,h}(\lambda)p_{\ell,h}(\mu) = \frac{T_N(\theta_h(\lambda)-\theta_h(\mu)) + T_N(\theta_h(\lambda)+\theta_h(\mu))}{2\cos(\theta_h(\lambda)/2)\cos(\theta_h(\mu)/2)} \longrightarrow 0.$$

Indeed, as θ_h is strictly increasing,

$$\theta_h(\lambda) - \theta_h(\mu) \notin 2\pi\mathbb{Z}, \qquad \theta_h(\lambda) + \theta_h(\mu) \in (0, 2\pi).$$

So Equation (20) applies to both cosine terms.

Since the spectrum is finite, combining these limits gives

$$\lim_{N\to\infty} \frac{1}{N}\sum_{\ell=0}^{N-1} \|J_\ell\|_{\mathrm{HS}}^2 = \|(P_0)_{vu}\|_{\mathrm{HS}}^2 + \sum_{\lambda>0} \frac{\|(P_\lambda)_{vu}\|_{\mathrm{HS}}^2}{2\cos^2(\theta_h(\lambda)/2)}.$$

If $J_\ell \neq 0$ at some step, its spectral representation implies that at least one block $(P_\lambda)_{vu}$ is non-zero. The right-hand side is therefore strictly positive.

If J_ℓ converged to zero in operator norm, the equivalence of norms in Appendix A.1 would imply $\|J_\ell\|_{\mathrm{HS}}^2 \to 0$. Its averages would then also converge to zero, contradicting the positive limit above. Thus, a response that is non-zero at some step cannot converge to zero as $\ell \to \infty$. □

To conclude, we show that the discrete spectral response converges to its continuous counterpart when the step size tends to zero at fixed propagation time.

Corollary 1. *Let $L_{\mathcal{F}}$ be fixed, self-adjoint, and positive semidefinite. Fix $T > 0$ and set $h = T/\ell$. Then, for every eigenvalue λ of $L_{\mathcal{F}}$, $\lim_{\ell\to\infty} p_{\ell,h}(\lambda) = \cos(T\sqrt{\lambda})$. Consequently, for every pair of nodes u, v,*

$$\lim_{\ell\to\infty} \left\| J^{\mathcal{F}}_{v\leftarrow u}[\ell, T/\ell] - J^{\mathcal{F}}_{v\leftarrow u}(T) \right\| = 0.$$

Proof. Since $h = T/\ell \to 0$, the stability condition $h^2\lambda_{\max}(L_{\mathcal{F}}) < 4$ holds for all sufficiently large ℓ. We may therefore use the spectral coefficient from Lemma 5.

For $\lambda = 0$, the coefficient equals 1, so the limit statement holds.

For $\lambda > 0$, we have $\frac{\theta_h(\lambda)}{h} = \sqrt{\lambda}\frac{\arcsin(h\sqrt{\lambda}/2)}{h\sqrt{\lambda}/2} \longrightarrow \sqrt{\lambda}$, where the last step uses the identity $\lim_{x\to 0}\frac{\arcsin(x)}{x} = 1$. Since $\ell h = T$, it follows that $(\ell + \frac{1}{2})\theta_h(\lambda) = (T + \frac{h}{2})\frac{\theta_h(\lambda)}{h} \longrightarrow T\sqrt{\lambda}$. Using Equation (19), also $\cos\big(\theta_h(\lambda)/2\big) = \sqrt{1 - h^2\lambda/4} \longrightarrow 1$. Continuity of cosine therefore gives

$$p_{\ell,h}(\lambda) = \frac{\cos\big((\ell + \frac{1}{2})\theta_h(\lambda)\big)}{\cos\big(\theta_h(\lambda)/2\big)} \longrightarrow \cos(T\sqrt{\lambda}).$$

For the final statement, we have that $\left\| J^{\mathcal{F}}_{v\leftarrow u}[\ell, h] - J^{\mathcal{F}}_{v\leftarrow u}(T) \right\| \leq \sum_\lambda \left| p_{\ell,h}(\lambda) - \cos(T\sqrt{\lambda}) \right| \|(P_\lambda)_{vu}\|$. For $\ell \longrightarrow \infty$, this last term converges to zero by the above. □

B Additional Experimental Details

This section provides Section 5 with details on restriction-map parametrizations, training protocols, and hyperparameter selection.

B.1 Restriction-Map Parametrizations

We consider four restriction-map parametrizations for ONDA, following (Bodnar et al., 2022; Borgi et al., 2026). The maps $\mathcal{F}_{v,e} : \mathcal{F}(v) \to \mathcal{F}(e)$ act between d-dimensional node and edge stalks and take the following forms:

- *Diagonal:* The restriction map is constrained to $\mathcal{F}_{v,e} = \mathrm{diag}(\boldsymbol{\alpha}_{v,e})$. Each stalk coordinate is therefore scaled independently, without mixing information across coordinates. This requires d scalar coefficients per restriction map.
- *Orthogonal.* The restriction map satisfies $\mathcal{F}_{v,e}^{\top}\mathcal{F}_{v,e} = I_d$. These maps allow for coordinate mixing while preserving Euclidean norms and inner products.
- *Low-Rank.* The restriction map is represented in factorized form, $\mathcal{F}_{v,e} = U_{v,e}V_{v,e}^{\top}$, with $U_{v,e}, V_{v,e} \in \mathbb{R}^{d\times r}$ and $1 \leq r < d$. This permits coordinate mixing through an intermediate space of dimension r, with $\mathrm{rank}(\mathcal{F}_{v,e}) \leq r$.
- *General.* The restriction map is an unconstrained matrix $\mathcal{F}_{v,e} \in \mathbb{R}^{d\times d}$. This provides the most flexible local transport, allowing arbitrary linear transformations between stalks, at the cost of $\mathcal{O}(d^2)$ storage and application complexity.

We stress that these parametrizations only affect the restriction maps used to construct $L_{\mathcal{F}}$, while retaining the same ONDA update rule and oscillatory dynamics.

B.2 Employed Baselines

In our experiments, the performance of our method is compared with various state-of-the-art GNN and SNN baselines from the literature. Specifically, we consider:

- classical MPNN-based methods, i.e., ChebNet (Defferrard et al., 2016), GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), GatedGCN (Bresson & Laurent, 2018), GIN (Xu et al., 2019), GINE (Hu et al., 2020), GCNII (Chen et al., 2020), and CoGNN (Finkelshtein et al., 2024);
- heterophily-specific models, i.e., H2GCN (Zhu et al., 2020), CPGNN (Zhu et al., 2021), FAGCN (Bo et al., 2021), GPR-GNN (Chien et al., 2021), FSGNN (Maurya et al., 2022), GloGNN (Li et al., 2022), GBK-GNN (Du et al., 2022), and JacobiConv (Wang & Zhang, 2022);
- DE-GNNs, i.e., GraphCON (Rusch et al., 2022), A-DGN (Gravina et al., 2023), SWAN (Gravina et al., 2025), PH-DGN (Heilig et al., 2025), SONAR (Trenta et al., 2025);
- Graph Transformers, i.e., Transformer (Vaswani et al., 2017; Dwivedi & Bresson, 2021), GT (Shi et al., 2021b), GPS (Rampášek et al., 2022), GOAT (Kong et al., 2023), Exphormer (Shirzad et al., 2023), NAGphormer Chen et al. (2023), GRIT(Ma et al., 2023);
- Higher-Order GNNs, i.e., DRew (Gutteridge et al., 2023);
- Sheaf Neural Networks, i.e., NSD (Bodnar et al., 2022), BuNN (Bamberger et al., 2025), and CSNN (Ribeiro et al., 2026).

B.3 Hyperparameter Search Spaces for the Experiments in Section 5

The hyperparameter search spaces used for ONDA are reported in Table 3 for the ECHO benchmark, Table 4 for the Barbell benchmark, and Table 5 for the Graph Transfer benchmark. To select the best hyperparameters, we perform a Bayesian optimization in the chosen hyperparameter space.

C Additional Results

C.1 Restriction Map Parametrization Ablation

In this section, we investigate how the parametrization of the restriction maps affects information propagation on the Barbell benchmark from Section 5.2. In Table 2, we compare four parametrizations, i.e., diagonal, orthogonal, low-rank, and general maps, which differ in the expressivity of the learned transport between stalks. To disentangle the effect of the restriction-map parametrization from that of the oscillatory dynamics, we evaluate the same variants for both ONDA and NSD. This allows us to assess how the choice of sheaf transport affects information transfer across the severe topological bottleneck of the Barbell graphs. To focus on the propagation mechanism, we consider ONDA configurations with a single-layer MLP-based restriction-map learner, without forcing or dissipation.

Table 3: Hyperparameter search space for the ECHO Benchmark.

Hyperparameter	**Candidate values**
Hidden dimension H	$[36, 252]$
Stalk dimension d	$\{2, 3, 4\}$
ONDA Blocks B	$\{1, 2, 3, 4\}$
Inner iterations $\mathcal{L}$	$[1, 40]$
Step size h	$[10^{-4}, 0.5]$
Learning rate	$[10^{-5}, 10^{-2}]$
Operator	{classical, normalized, directional}
Sheaf type	{general, diagonal, orthogonal}
Dissipation	{False, True}
Forcing	{False, True}
Fixed Restriction maps	{False, True}

Table 4: Hyperparameter search space for the Barbell benchmark.

Hyperparameter	**Candidate values**
Hidden dimension H	$\{128, 192\}$
Stalk dimension d	$\{2, 6\}$
ONDA Blocks B	$\{1, 2, 4\}$
Inner iterations $\mathcal{L}$	$\{8, 10, 12, 14, 16, 20\}$
Step size h	$\{0.1, 0.2, 0.25, 0.33, 0.5, 0.6, 0.66, 0.75, 1.0\}$
Learning rate	$\{3 \times 10^{-4}\}$
Operator	{classical}
Sheaf type	{general, diagonal, orthogonal, low-rank}
Dissipation	{False}
Forcing	{False}
Fixed Restriction maps	{True}

Results. Across all restriction-map parametrizations and graph sizes, ONDA substantially improves over the corresponding NSD variant, showing that its effectiveness is robust to the particular choice of sheaf transport. All four ONDA parametrizations solve the task at $N = 10$, with the diagonal and orthogonal variants remaining below the solving threshold across all tested sizes. Among them, diagonal maps provide the most consistent behavior, achieving the lowest mean error at both $N = 20$ and $N = 50$. The remaining parametrizations exhibit a stronger dependence on graph size. Notably, the most expressive general maps are not systematically preferable to more constrained parametrizations. Overall, these results indicate that richer restriction-map expressivity alone is not sufficient for effective information transfer across severe bottlenecks, while the oscillatory propagation of ONDA consistently improves upon diffusion-based NSD under matched restriction-map parametrizations.

C.2 Heterophilic Tasks

Although ONDA is primarily designed to improve long-range information propagation, we also evaluate its effectiveness on heterophilic tasks. We consider the heterophilic benchmarks from Platonov et al. (2023), and compare ONDA against conventional MPNNs, graph Transformers, heterophily-specific architectures, and recent sheaf neural networks (see also Section B.2).

Hyperparameter Space. Table 7 reports the hyperparameter values considered for the heterophilic benchmarks. Hyperparameters were selected through sequential local searches over subsets of these candidate values as well as Bayesian optimization. Final configurations were selected according to validation performance. Similarly to Bamberger et al. (2025); Ribeiro et al. (2026), for these tasks we optionally learn restriction maps from node representations produced by a GNN encoder. Specifically, in Table 7, *GNN depth* denotes the number of message-passing layers, and *GNN*

Table 5: Hyperparameter search space for the graph transfer task.

Hyperparameter	**Candidate values**
Hidden dimension H	64 (Gravina et al., 2025)
Stalk dimension d	$\{1, 2, 4\}$
ONDA Blocks B	$\{1, 2\}$
Inner iterations $\mathcal{L}$	distance x $\{1, 2\}$ / B
Step size h	$\{0.05, 0.1, 0.5, 1.0\}$
Learning rate	10^{-3} (Gravina et al., 2025)
Operator	{classical, normalized, directional}
Sheaf type	{general, diagonal}
Dissipation	{False, True}
Forcing	{False, True}
Fixed Restriction maps	{False, True}

Table 6: **Barbell additional results.** Test MSE ($\times 10^{-3}$; $\downarrow$) on the Barbell task for graph sizes $N \in \{10, 20, 50\}$. All models report the results over the best configuration selected by validation, as mean$_{\pm \text{std}}$ over 5 seeds. For each column, we report the **best**, **second-best**, and **third-best** results.

Model	$N = 10 \downarrow$	$N = 20 \downarrow$	$N = 50 \downarrow$
GCN	$884.200_{\pm 6.900}$	$961.100_{\pm 12.300}$	$860.000_{\pm 35.000}$
GAT	$7.900_{\pm 2.100}$	$487.000_{\pm 342.200}$	$852.000_{\pm 29.000}$
GraphSAGE	$455.500_{\pm 199.600}$	$998.200_{\pm 6.500}$	$876.000_{\pm 32.000}$
GIN	$900.500_{\pm 47.900}$	$970.600_{\pm 3.900}$	$985.000_{\pm 123.000}$
GraphTransformer	$670.300_{\pm 306.800}$	$1041.200_{\pm 46.900}$	$831.000_{\pm 13.000}$
ChebNet	$10.600_{\pm 1.200}$	$983.600_{\pm 5.100}$	$922.000_{\pm 47.000}$
SONAR	$462.800_{\pm 250.300}$	$717.500_{\pm 249.600}$	$809.000_{\pm 68.000}$
Diag-NSD	$937.200_{\pm 11.400}$	$1006.600_{\pm 22.000}$	$858.000_{\pm 43.000}$
Ortho-NSD	$965.300_{\pm 23.000}$	$1005.100_{\pm 7.800}$	$891.000_{\pm 38.000}$
LowRank-NSD	$955.100_{\pm 6.300}$	$1008.100_{\pm 16.400}$	$825.000_{\pm 32.000}$
General-NSD	$938.100_{\pm 12.100}$	$1034.200_{\pm 23.500}$	$860.000_{\pm 31.000}$
CSNN	$473.900_{\pm 106.200}$	$890.800_{\pm 265.400}$	$742.000_{\pm 168.000}$
Diag-ONDA	**$1.200_{\pm 0.400}$**	**$0.900_{\pm 0.500}$**	**$83.000_{\pm 85.000}$**
Ortho-ONDA	**$0.700_{\pm 0.300}$**	**$68.900_{\pm 165.900}$**	**$92.000_{\pm 127.000}$**
LowRank-ONDA	$22.800_{\pm 19.500}$	$840.100_{\pm 307.900}$	**$128.000_{\pm 154.000}$**
General-ONDA	**$1.300_{\pm 0.300}$**	**$2.200_{\pm 1.200}$**	$616.000_{\pm 121.000}$

hidden dimension denotes the encoder's hidden width. At depth zero, an MLP learns the restriction maps directly from the endpoint node features, similar to (Bodnar et al., 2022).

Results. Table 8 reports the performance on the heterophilic benchmarks. ONDA achieves highly competitive performance across the heterophilic benchmarks despite not being specifically designed for these tasks, ranking third overall based on the average rank across the five datasets. Notably, ONDA consistently improves over SONAR across all five datasets, showing that extending oscillatory propagation to the sheaf setting is also beneficial beyond explicitly long-range tasks. Similarly, ONDA substantially outperforms NSD on every dataset, indicating that richer sheaf transport alone is not sufficient to match the performance obtained with the proposed oscillatory dynamics. Compared with more recent sheaf architectures, ONDA remains competitive with BuNN and CSNN. Overall, these results show that ONDA maintains strong performance on heterophilic graphs while improving over both its closest oscillatory and standard sheaf counterparts.

D Time and Memory Complexity Analysis

In this section, we provide both a theoretical and experimental analysis of the time and space complexity of ONDA. Let $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$, let d denote the stalk dimension, c the number of feature channels, $\mathcal{L}$ the number of internal propagation steps per block, and B the number of blocks.

Table 7: Hyperparameter search space for the heterophilic benchmarks.

Hyperparameter	**Candidate values**
Hidden dimension H	$\{64, 128, 256, 512\}$
Stalk dimension d	$\{2, 4, 8\}$
ONDA Blocks B	$\{2, 3, 4, 6, 8, 10, 11\}$
Inner iterations $\mathcal{L}$	$\{1, 2, 4, 6, 8, 10\}$
Step size ε	$\{0.01, 0.025, 0.04, 0.05, 0.075, 0.1\}$
Learning rate	$\{5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}, 2 \times 10^{-2}\}$
Operator	$\{\text{classical}, \text{normalized}, \text{directional}\}$
Sheaf type	$\{\text{general}\}$
Dissipation	$\{\text{False}, \text{True}\}$
Forcing	$\{\text{False}, \text{True}\}$
Fixed Restriction maps	$\{\text{True}\}$
GNN depth	$\{0, 1, 2, 3, 4, 5, 6\}$
GNN hidden dim	$\{64, 128, 256, 512\}$

Since ONDA operates through graph-local propagation, its computational cost is determined by the stalk dimension, the number of feature channels, and the parameterization of the restriction maps.

Time Complexity. Consider a single propagation step. The state and velocity $\boldsymbol{V}$ each contain ndc scalar entries. The stalk transformation $(\boldsymbol{I}_n \otimes \boldsymbol{W}_1)\boldsymbol{X}$ costs $\mathcal{O}(nd^2c)$, while the channel mixer $\boldsymbol{W}_2$ costs $\mathcal{O}(ndc^2)$. The cost of applying the sheaf operator depends on the restriction-map family (see Appendix B.1). Writing $q(d)$ for the cost of applying one restriction map to one feature channel, we have:

$$q(d) = \begin{cases} d, & \text{diagonal,} \\ d^2, & \text{general or orthogonal,} \\ rd, & \text{rank-}r \text{ low-rank,} \end{cases} \tag{21}$$

where the low-rank complexity assumes that the restriction maps are applied in factorized form. A sparse application of the sheaf operator therefore costs $\mathcal{O}(mq(d)c)$. Since the restriction maps may be recomputed from the current state at each propagation step, we denote the corresponding learner cost by $\mathcal{C}_{\text{map}}$. The velocity update uses the sheaf propagation term already counted above. The remaining updates of $\boldsymbol{V}$ and $\boldsymbol{X}$ involve only elementwise scaling and addition, contributing $\mathcal{O}(ndc)$ operations per step. This contribution is absorbed by the node-wise transformation costs, so the combined cost of one ONDA propagation step is:

$$T_{\text{ONDA}} = \mathcal{O}\big(\mathcal{C}_{\text{map}} + mq(d)c + nd^2c + ndc^2\big)\,. \tag{22}$$

Across B blocks, each with $\mathcal{L}$ propagation steps, the total propagation cost is:

$$T_{\text{ONDA}}^{\text{net}} = \begin{cases} \mathcal{O}\big(B\mathcal{C}_{\text{map}} + B\mathcal{L}(mq(d)c + nd^2c + ndc^2)\big)\,, & \text{when maps fixed within each block,} \\ \mathcal{O}\big(B\mathcal{L}\left(\mathcal{C}_{\text{map}} + mq(d)c + nd^2c + ndc^2\right)\big)\,, & \text{when maps recomputed at each step.} \end{cases}$$

In particular, for fixed stalk dimension d, feature dimension c, and restriction-map parameterization, ONDA retains linear scaling with the graph size $n + m$. Thus, its matrix-valued sheaf transport preserves the locality and sparse graph scaling of message-passing architectures (Gilmer et al., 2017). In contrast, Graph Transformers based on dense global self-attention typically incur quadratic complexity in the number of nodes, $\mathcal{O}(n^2)$, due to all-pairs interactions. Thus, despite its richer local transport, ONDA preserves the sparse scaling characteristic of MPNNs and avoids the quadratic dependence on n of dense Graph Transformers.

Memory Complexity. The instantaneous state $\boldsymbol{X}$ and velocity $\boldsymbol{V}$ together contain $2ndc$ scalar entries and therefore require $\mathcal{O}(ndc)$ memory. The learned restriction maps additionally require $\mathcal{O}(ms(d))$ storage, where:

$$s(d) = \begin{cases} d, & \text{diagonal,} \\ d^2, & \text{general or orthogonal,} \\ rd, & \text{rank-}r \text{ low-rank,} \end{cases} \tag{23}$$

Table 8: Mean test set score and std on heterophilic datasets. The higher, the better. **First**, **second**, and **third** best results for each task are color-coded. Baseline results are reported from Finkelshtein et al. (2024); Platonov et al. (2023); Müller et al. (2024); Luan et al. (2024); Ribeiro et al. (2026).

Model	Roman-empire Acc ↑	Amazon-ratings Acc ↑	Minesweeper AUC ↑	Tolokers AUC ↑	Questions AUC ↑
Luan et al. (2024)					
MLP-2	$66.04_{\pm 0.71}$	$49.55_{\pm 0.81}$	$50.92_{\pm 1.25}$	$74.58_{\pm 0.75}$	$69.97_{\pm 1.16}$
SGC-1	$44.60_{\pm 0.52}$	$40.69_{\pm 0.42}$	$82.04_{\pm 0.77}$	$73.80_{\pm 1.35}$	$71.06_{\pm 0.92}$
MLP-1	$64.12_{\pm 0.61}$	$38.60_{\pm 0.41}$	$50.59_{\pm 0.83}$	$71.89_{\pm 0.82}$	$70.33_{\pm 0.96}$
Graph-agnostic					
ResNet	$65.88_{\pm 0.38}$	$45.90_{\pm 0.52}$	$50.89_{\pm 1.39}$	$72.95_{\pm 1.06}$	$70.34_{\pm 0.76}$
ResNet+SGC	$73.90_{\pm 0.51}$	$50.66_{\pm 0.48}$	$70.88_{\pm 0.90}$	$80.70_{\pm 0.97}$	$75.81_{\pm 0.96}$
ResNet+adj	$52.25_{\pm 0.40}$	$51.83_{\pm 0.57}$	$50.42_{\pm 0.83}$	$78.78_{\pm 1.11}$	$75.77_{\pm 1.24}$
MPNNs					
GAT	$80.87_{\pm 0.30}$	$49.09_{\pm 0.63}$	$92.01_{\pm 0.68}$	$83.70_{\pm 0.47}$	$77.43_{\pm 1.20}$
GAT-sep	$88.75_{\pm 0.41}$	$52.70_{\pm 0.62}$	$93.91_{\pm 0.35}$	$83.78_{\pm 0.43}$	$76.79_{\pm 0.71}$
GAT (LapPE)	$84.80_{\pm 0.46}$	$44.90_{\pm 0.73}$	$93.50_{\pm 0.54}$	$84.99_{\pm 0.54}$	$76.55_{\pm 0.84}$
GAT (RWSE)	$86.62_{\pm 0.53}$	$48.58_{\pm 0.41}$	$92.53_{\pm 0.65}$	$85.02_{\pm 0.67}$	$77.83_{\pm 1.22}$
GAT (DEG)	$85.51_{\pm 0.56}$	$51.65_{\pm 0.60}$	$93.04_{\pm 0.62}$	$84.22_{\pm 0.81}$	$77.10_{\pm 1.23}$
GatedGCN	$74.46_{\pm 0.54}$	$43.00_{\pm 0.32}$	$87.54_{\pm 1.22}$	$77.31_{\pm 1.14}$	$76.61_{\pm 1.13}$
GCN	$73.69_{\pm 0.74}$	$48.70_{\pm 0.63}$	$89.75_{\pm 0.52}$	$83.64_{\pm 0.67}$	$76.09_{\pm 1.27}$
GCN (LapPE)	$83.37_{\pm 0.55}$	$44.35_{\pm 0.36}$	$94.26_{\pm 0.49}$	$84.95_{\pm 0.78}$	$77.79_{\pm 1.34}$
GCN (RWSE)	$84.84_{\pm 0.55}$	$46.40_{\pm 0.55}$	$93.84_{\pm 0.48}$	**$85.11_{\pm 0.77}$**	$77.81_{\pm 1.40}$
GCN (DEG)	$84.21_{\pm 0.47}$	$50.01_{\pm 0.69}$	$94.14_{\pm 0.50}$	$82.51_{\pm 0.83}$	$76.96_{\pm 1.21}$
CoGNN(Σ, Σ)	$91.57_{\pm 0.32}$	$51.28_{\pm 0.56}$	$95.09_{\pm 1.18}$	$83.36_{\pm 0.89}$	**$80.02_{\pm 0.86}$**
CoGNN(μ, μ)	$91.37_{\pm 0.35}$	**$54.17_{\pm 0.37}$**	$97.31_{\pm 0.41}$	$84.45_{\pm 1.17}$	$76.54_{\pm 0.95}$
GraphSAGE	$85.74_{\pm 0.67}$	**$53.63_{\pm 0.39}$**	$93.51_{\pm 0.57}$	$82.43_{\pm 0.44}$	$76.44_{\pm 0.62}$
SONAR	$89.82_{\pm 0.57}$	$52.22_{\pm 0.14}$	$96.29_{\pm 0.73}$	$83.57_{\pm 1.44}$	$74.96_{\pm 1.10}$
Graph Transformers					
Exphormer	$89.03_{\pm 0.37}$	$53.51_{\pm 0.46}$	$90.74_{\pm 0.53}$	$83.77_{\pm 0.78}$	$73.94_{\pm 1.06}$
NAGphormer	$74.34_{\pm 0.77}$	$51.26_{\pm 0.72}$	$84.19_{\pm 0.66}$	$78.32_{\pm 0.95}$	$68.17_{\pm 1.53}$
GOAT	$71.59_{\pm 1.25}$	$44.61_{\pm 0.50}$	$81.09_{\pm 1.02}$	$83.11_{\pm 1.04}$	$75.76_{\pm 1.66}$
GPS	$82.00_{\pm 0.61}$	$53.10_{\pm 0.42}$	$90.63_{\pm 0.67}$	$83.71_{\pm 0.48}$	$71.73_{\pm 1.47}$
$\text{GPS}_{\text{GCN+Performer}}$ (LapPE)	$83.96_{\pm 0.53}$	$48.20_{\pm 0.67}$	$93.85_{\pm 0.41}$	$84.72_{\pm 0.77}$	$77.85_{\pm 1.25}$
$\text{GPS}_{\text{GCN+Performer}}$ (RWSE)	$84.72_{\pm 0.65}$	$48.08_{\pm 0.85}$	$92.88_{\pm 0.50}$	$84.81_{\pm 0.86}$	$76.45_{\pm 1.51}$
$\text{GPS}_{\text{GCN+Performer}}$ (DEG)	$83.38_{\pm 0.68}$	$48.93_{\pm 0.47}$	$93.60_{\pm 0.47}$	$80.49_{\pm 0.97}$	$74.24_{\pm 1.18}$
$\text{GPS}_{\text{GAT+Performer}}$ (LapPE)	$85.93_{\pm 0.52}$	$48.86_{\pm 0.38}$	$92.62_{\pm 0.79}$	$84.62_{\pm 0.54}$	$76.71_{\pm 0.98}$
$\text{GPS}_{\text{GAT+Performer}}$ (RWSE)	$87.04_{\pm 0.58}$	$49.92_{\pm 0.68}$	$91.08_{\pm 0.58}$	$84.38_{\pm 0.91}$	$77.14_{\pm 1.49}$
$\text{GPS}_{\text{GAT+Performer}}$ (DEG)	$85.54_{\pm 0.58}$	$51.03_{\pm 0.60}$	$91.52_{\pm 0.46}$	$82.45_{\pm 0.89}$	$76.51_{\pm 1.19}$
GT	$86.51_{\pm 0.73}$	$51.17_{\pm 0.66}$	$91.85_{\pm 0.76}$	$83.23_{\pm 0.64}$	$77.95_{\pm 0.68}$
GT-sep	$87.32_{\pm 0.39}$	$52.18_{\pm 0.80}$	$92.29_{\pm 0.47}$	$82.52_{\pm 0.92}$	$78.05_{\pm 0.93}$
Heterophily-Designated GNNs					
CPGNN	$63.96_{\pm 0.62}$	$39.79_{\pm 0.77}$	$52.03_{\pm 5.46}$	$73.36_{\pm 1.01}$	$65.96_{\pm 1.95}$
FAGCN	$65.22_{\pm 0.56}$	$44.12_{\pm 0.30}$	$88.17_{\pm 0.73}$	$77.75_{\pm 1.05}$	$77.24_{\pm 1.26}$
FSGNN	$79.92_{\pm 0.56}$	$52.74_{\pm 0.83}$	$90.08_{\pm 0.70}$	$82.76_{\pm 0.61}$	**$78.86_{\pm 0.92}$**
GBK-GNN	$74.57_{\pm 0.47}$	$45.98_{\pm 0.71}$	$90.85_{\pm 0.58}$	$81.01_{\pm 0.67}$	$74.47_{\pm 0.86}$
GloGNN	$59.63_{\pm 0.69}$	$36.89_{\pm 0.14}$	$51.08_{\pm 1.23}$	$73.39_{\pm 1.17}$	$65.74_{\pm 1.19}$
GPR-GNN	$64.85_{\pm 0.27}$	$44.88_{\pm 0.34}$	$86.24_{\pm 0.61}$	$72.94_{\pm 0.97}$	$55.48_{\pm 0.91}$
H2GCN	$60.11_{\pm 0.52}$	$36.47_{\pm 0.23}$	$89.71_{\pm 0.31}$	$73.35_{\pm 1.01}$	$63.59_{\pm 1.46}$
JacobiConv	$71.14_{\pm 0.42}$	$43.55_{\pm 0.48}$	$89.66_{\pm 0.40}$	$68.66_{\pm 0.65}$	$73.88_{\pm 1.16}$
Sheaf Neural Networks					
NSD	$80.41_{\pm 0.72}$	$42.76_{\pm 0.54}$	$92.15_{\pm 0.84}$	$78.83_{\pm 0.76}$	$69.69_{\pm 1.46}$
BuNN	**$91.75_{\pm 0.39}$**	**$53.74_{\pm 0.51}$**	**$98.99_{\pm 0.16}$**	$84.78_{\pm 0.80}$	$78.75_{\pm 1.09}$
CSNN	**$92.63_{\pm 0.50}$**	$52.07_{\pm 1.00}$	**$99.07_{\pm 0.25}$**	**$85.45_{\pm 0.53}$**	**$79.31_{\pm 1.22}$**
Ours					
ONDA	**$91.86_{\pm 0.31}$**	$52.76_{\pm 0.48}$	**$98.95_{\pm 0.54}$**	**$85.21_{\pm 0.9}$**	$76.14_{\pm 1.30}$

Table 9: **Matched-parameter computational cost on Barbell.** All models use approximately 100k trainable parameters, four blocks, and 12 internal propagation steps per block. We report peak training memory, training time per epoch, and inference latency per graph for $N \in \{10, 20, 50\}$, corresponding to graphs with 20, 40, and 100 nodes, respectively. Values are means over five seeds; lower is better.

		$N = 10$			$N = 20$			$N = 50$		
Model	**#Params**	**Mem. MB**	**Train ms/ep.**	**Infer. ms/g**	**Mem. MB**	**Train ms/ep.**	**Infer. ms/g**	**Mem. MB**	**Train ms/ep.**	**Infer. ms/g**
SONAR	100,079	23.1	2400.1	10.245	35.1	2452.0	10.800	120.7	2675.2	11.138
Diag-ONDA	101,405	26.8	2186.3	6.716	50.9	2193.7	6.795	223.0	2270.4	6.722
Ortho-ONDA	103,700	24.5	2229.1	7.849	41.1	2237.4	7.962	159.9	2432.3	7.925
LowRank-ONDA	101,540	27.6	2464.3	8.371	53.9	2459.3	8.290	241.8	2736.4	8.610
General-ONDA	104,814	27.4	2404.6	8.150	53.1	2447.9	8.138	236.5	2729.5	8.604

up to constant factors associated with the chosen parameterization. Thus, excluding trainable parameters and common graph storage, inference requires:

$$M_{\mathrm{ONDA}}^{\mathrm{infer}} = \mathcal{O}(ndc + ms(d)) \,. \tag{24}$$

During training, backpropagation through the unrolled second-order dynamics requires retaining intermediate activations. Without activation checkpointing, this gives the leading-order bound:

$$M_{\mathrm{ONDA}}^{\mathrm{train}} = \mathcal{O}(B\mathcal{L}\,(ndc + ms(d))) \,. \tag{25}$$

For fixed d and restriction-map parameterization, the memory requirements of ONDA also scale linearly with $n + m$, as in MPNNs. This contrasts with dense Graph Transformers, whose attention matrices require $\mathcal{O}(n^2)$ memory per layer. Hence, both the computational and memory complexity of ONDA remain compatible with sparse graph processing, with the additional cost arising from the stalk-valued representation and matrix-valued restriction maps rather than from dense pairwise interactions.

Matched-parameter Empirical Cost. To complement the asymptotic analysis, we measure wall-clock time and peak GPU memory on the Barbell benchmark under a controlled matched-parameter setting. We compare ONDA with SONAR (Trenta et al., 2025), a message-passing architecture based on oscillatory dynamics, to assess the practical computational overhead introduced by the sheaf-based components of ONDA. All models use $B = 4$ blocks, $\mathcal{L} = 12$ internal propagation steps per block, and approximately 100k trainable parameters. Measurements are obtained with batch size one (to avoid confounding effects from batch-level computational optimizations) on an RTX 5080, with one exclusive GPU process at a time. Table 9 reports results on multiple settings of the problem, $N = 10, 20, 50$, showing results for different implementations of ONDA's restriction-map parametrization (see Appendix B.1). Despite the richer per-edge computation predicted by Equation (22), the measured runtime remains comparable to SONAR under a matched parameter budget. At $N = 50$, Diag-ONDA and Ortho-ONDA reduce training time by 15.1% and 9.1%, respectively, relative to SONAR, while their inference latency is lower by 39.6% and 28.8%. The more expressive LowRank and General variants incur only a 2.3% and 2.0% increase in training time, respectively, while still reducing inference latency by approximately 22.7%. Across $N \in \{10, 20, 50\}$, all four ONDA variants exhibit lower measured inference latency than SONAR. At $N = 50$, peak training memory ranges from 159.9 to 241.8 MB for the sheaf variants, compared with 120.7 MB for SONAR. This increase is consistent with Equation (25): ONDA maintains d-dimensional stalk states and learned edge-wise restriction maps throughout the unrolled dynamics. Importantly, this additional memory buys a strictly richer propagation mechanism while preserving sparse, linear-in-$n+m$ graph scaling. Hence, for the small stalk dimensions used in practice, ONDA retains the computational profile of local SONAR propagation while replacing scalar communication with expressive matrix-valued transport.