Continuous Simplicial Neural Networks

Aref Einizade
LTCI, Télécom Paris
Institut Polytechnique de Paris
aref.einizade@telecom-paris.fr

Dorina Thanou
EPFL, Lausanne, Switzerland
dorina.thanou@epfl.ch

Fragkiskos D. Malliaros
CentraleSupélec, Inria
Université Paris-Saclay
fragkiskos.malliaros@centralesupelec.fr

Jhony H. Giraldo
LTCI, Télécom Paris
Institut Polytechnique de Paris
jhony.giraldo@telecom-paris.fr

Abstract

Simplicial complexes provide a powerful framework for modeling higher-order interactions in structured data, making them particularly suitable for applications such as trajectory prediction and mesh processing. However, existing simplicial neural networks (SNNs), whether convolutional or attention-based, rely primarily on discrete filtering techniques, which can be restrictive. In contrast, partial differential equations (PDEs) on simplicial complexes offer a principled approach to capture continuous dynamics in such structures. In this work, we introduce **co**ntinuous **sim**plicial neural netw**o**rk (COSIMO), a novel SNN architecture derived from PDEs on simplicial complexes. We provide theoretical and experimental justifications of COSIMO's stability under simplicial perturbations. Furthermore, we investigate the over-smoothing phenomenon—a common issue in geometric deep learning—demonstrating that COSIMO offers better control over this effect than discrete SNNs. Our experiments on real-world datasets demonstrate that COSIMO achieves competitive performance compared to state-of-the-art SNNs in complex and noisy environments. The implementation codes are available in https://github.com/ArefEinizade2/COSIMO.

1 Introduction

Graph representation learning provides a powerful framework for modeling structured data. In this context, graph neural networks (GNNs) have gained significant attention [1, 2, 3, 4], extending neural network architectures to graph-structured data. By capturing complex relationships between nodes, GNNs have been successfully applied to various domains, including computational/digital pathology [5], social network analysis [6], drug discovery [7], materials modeling [8], and computer vision [9, 10]. However, traditional GNNs primarily focus on pairwise interactions between nodes, limiting their ability to model higher-order relationships in complex systems such as biological networks [11]. To overcome this limitation, researchers have explored more expressive mathematical structures, such as *abstract simplicial complexes* [12], generalizing graphs by incorporating multi-way connections and their Laplacians by introducing the Hodge decomposition theory [12, 13].

An abstract simplicial complex is a combinatorial structure composed of sets that are closed under subset operations. For instance, a three-dimensional simplicial complex includes tetrahedrons (four-element sets), triangles (three-element sets), edges (two-element sets), and vertices (one-element sets), as illustrated in Fig. 1. Graphs correspond to simplicial 1-complexes, containing only nodes and edges, while point clouds (sets of unconnected nodes) can be seen as simplicial 0-complexes.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Building upon the mathematical foundation of abstract simplicial complexes, simplicial neural networks (SNNs) [14] have emerged as a powerful approach for learning on higher-order structures. Most existing SNNs rely on discrete simplicial filters [14, 15] and their variations [16, 17, 18] to process data defined on simplicial complexes. However, a fundamental question remains largely unexplored: *how can we design continuous SNNs?* Continuous models play a crucial role in capturing real-world dynamics evolving on structured data [19]. Compared to their discrete counterparts, continuous convolutional models offer several advantages, including: *i*) better control of over-smoothing [19], preventing excessive feature homogenization across the structure, and *ii*) greater robustness to structural perturbations [20, 21]. Despite these benefits, to the best of our knowledge, no continuous filtering method has been proposed for learning on simplicial complexes. To address this gap, we introduce the **co**ntinuous **sim**plicial neural netw**o**rk (COSIMO) model, the first method for modeling and learning over the continuous dynamics of simplicial complexes with higher-order connections. We analyze COSIMO both theoretically and empirically, demonstrating its effectiveness in learning from simplicial complex data. Our main contributions can be summarized as follows:

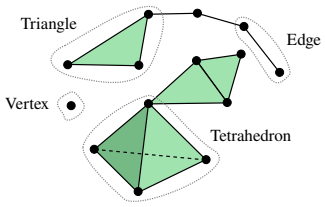

Figure 1: Example of an abstract simplicial complex.

- We introduce COSIMO, a new approach that uses partial differential equations (PDEs) on simplicial complexes to enable continuous information flow through these structures.
- We establish theoretical stability guarantees for COSIMO, demonstrating its robustness to structural perturbations.
- We provide a detailed analysis of over-smoothing in both discrete and continuous SNNs, showing that COSIMO achieves a better control on the rate of convergence to the over-smoothing state.
- We validate COSIMO through experiments on both synthetic and real-world datasets, showing its competitive performance against state-of-the-art (SOTA) methods.

2 Related Work

The introduction of topological signal processing over simplicial complexes [22, 13] has significantly advanced topological methods in machine learning, highlighting the benefits of these data structures [12], and contributing to the emerging field of topological deep learning [23, 24]. The evolution of the SNN architectures has followed a trajectory similar to that of GNNs, with the following stages: *i*) the establishment of principles in *topological signal processing over simplicial complexes* [22], *ii*) the formulation of *simplicial filters* [12], and *iii*) the application of these filters to create *SNNs* [14].

Research on learning methods for simplicial complexes has explored various approaches. Roddenberry and Segarra [25] were among the first to develop neural networks that operate on the graph edges using simplicial representations. Building on this, Ebli *et al*. [14] incorporated the Hodge Laplacian, a generalization of the graph Laplacian, to extend GNNs to higher-dimensional simplices. Other studies [26, 17] refined this approach by separating the lower and upper Laplacians, two components of the Hodge Laplacian that capture connections between simplices of different dimensions. Keros *et al*. [27] further extended the framework in [26] to detect topological holes, while Chen *et al*. [28] integrated node and edge operations for link prediction. More recently, attention mechanisms have been incorporated into simplicial networks [29, 11, 30].

Most of these studies focus on learning within individual simplicial levels without explicitly modeling the incidence relations (interactions between simplices of different dimensions) inherent in simplicial complexes [18]. The inclusion of these relations was explored in [18, 17, 31], which proposed convolutional-like architectures that were later unified under the simplicial complex convolutional neural network (SCCNN) framework [18]. Simultaneously, other studies extended message-passing from GNNs [32] to simplicial complexes by leveraging both adjacency and incidence structures [16, 33].

Unlike GNNs, the theoretical understanding of SNNs is still developing. For example, Roddenberry *et al*. [26] analyzed how permutation-symmetric neural networks preserve equivariance under permutation and orientation changes—an important property also supported by SCCNNs [18]. Another study examined message-passing in simplicial complexes through the lens of the Weisfeiler-Lehman test, applied to simplicial complexes derived from clique expansions of graphs [16]. A

spectral formulation based on the simplicial complex Fourier transform was also proposed in [17]. Other critical yet underexplored theoretical aspects of SNNs, including their stability and over-smoothing behaviors, have been analyzed only in the context of SCCNNs [18]. For example, Yang *et al.* [18] analyzed the steady state solution of the diffusion PDE on simplicial complexes, showing that SCCNNs are more robust to over-smoothing compared to their non-Hodge-aware counterparts.

In contrast to previous methods, COSIMO leverages continuous dynamics in both the lower and upper Hodge Laplacians. Although the discrete SCCNN also decouples the Hodge Laplacians, it faces two challenges: *i*) the Hodge filters' order must be manually tuned, and *ii*) the model has limited control over over-smoothing, a common issue in deep GNNs and SNNs. COSIMO addresses these challenges by introducing lower and upper Hodge-aware PDEs on simplicial complexes, enabling differentiability of the convolutional operation *w.r.t.* the simplicial receptive fields, providing greater flexibility and robustness in learning.

3 Preliminaries

3.1 Notation and Simplicial Complexes

Notation. Calligraphic letters like $\mathcal{X}$ denote sets with $|\mathcal{X}|$ being their cardinality. Uppercase boldface letters such as $\mathbf{B}$ represent matrices, and lowercase boldface letters like $\mathbf{x}$ denote vectors. Similarly, $\mathrm{tr}(\cdot)$ represents the trace of a matrix, $\|\cdot\|$ is the ℓ_2-norm of a vector, and $(\cdot)^\top$ designates transposition.

Simplicial complex. A simplicial complex is a set $\mathcal{X}$ of finite subsets of another set $\mathcal{V}$ that is closed under restriction, *i.e.*, $\forall\, s^k \in \mathcal{X}$, if $s^{k'} \subseteq s^k$, then $s^{k'} \in \mathcal{X}$. Each element of $\mathcal{X}$ is called a *simplex*. Particularly, if $|s^k| = k+1$, we call s^k a k-simplex. A *face* of s^k is a subset with cardinality k, while a *coface* of s^k is a $k+1$-simplex that has s^k as a face [22]. We refer to the 0-simplices as nodes, the 1-simplices as edges, and the 2-simplices as triangles. For higher-order simplices, we use the term k-simplices. The notation $\mathcal{X}_k$ represents the collection of k-simplices of $\mathcal{X}$. If $\mathcal{X}_c = \emptyset\ \forall\ c > d$, we say $\mathcal{X}$ is a simplicial complex of dimension d. For example, a simple graph is a simplicial complex of dimension one and can be represented as $G = (\mathcal{X}_0, \mathcal{X}_1)$, *i.e.*, the set of nodes and edges.

We use incidence matrices $\mathbf{B}_k \in \{-1, 0, 1\}^{|\mathcal{X}_{k-1}| \times |\mathcal{X}_k|}$, to describe the incidence relationships between $k-1$-simplices (faces) and k-simplices. For example, $\mathbf{B}_1$ and $\mathbf{B}_2$ are node-to-edge and edge-to-triangle incidence matrices, respectively. Simplicial complexes are defined with some orientation, and therefore the value $\mathbf{B}_k(i,j)$ is either -1 or 1 if the k-simplex i is incident to the $k-1$-simplex j, depending on the orientation, and 0 otherwise. Please notice that $\mathbf{B}_0$ is not defined. We define the *k-Hodge Laplacians* as $\mathbf{L}_k = \mathbf{B}_k^\top \mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$, where $\mathbf{L}_{k,d} = \mathbf{B}_k^\top \mathbf{B}_k$ is the *lower Laplacian*, $\mathbf{L}_{k,u} = \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$ is the *upper Laplacian*, $\mathbf{L}_0 = \mathbf{B}_1\mathbf{B}_1^\top$ is the *graph Laplacian*. Discrete SNNs [18, 17] define their convolution operations as matrix polynomials of the Hodge Laplacians over *simplicial signals*, *i.e.*, signals defined over the simplicial complex.

Simplicial signal. We define a *k-simplicial signal* as a function in $\mathcal{X}_k$ as $x_k : \mathcal{X}_k \to \mathbb{R}$. Therefore, we can define a one-dimensional k-simplicial signal as $\mathbf{x}_k \in \mathbb{R}^{|\mathcal{X}_k|}$. We can calculate how $\mathbf{x}_k$ varies *w.r.t.* the faces and cofaces of k-simplices by $\mathbf{B}_k\mathbf{x}_k$ and $\mathbf{B}_{k+1}^\top\mathbf{x}_k$ [18]. For example, in a node signal $\mathbf{x}_0$, $\mathbf{B}_1^\top\mathbf{x}_0$ computes its *gradient* as the difference between adjacent nodes, and in an edge signal $\mathbf{x}_1$, $\mathbf{B}_1\mathbf{x}_1$ computes its *divergence* [18].

Dirichlet energy. The Dirichlet energy $E(\cdot)$ quantifies the smoothness of a simplicial signal with respect to the k-Hodge Laplacian [18], where a lower energy value indicates a smoother signal.

Definition 3.1 (from [18])**.** The Dirichlet energy of a simplicial signal $\mathbf{x}_k$ can be stated as:

$$E(\mathbf{x}_k) := \mathbf{x}_k^\top \mathbf{L}_k \mathbf{x}_k = \|\mathbf{B}_k\mathbf{x}_k\|_2^2 + \|\mathbf{B}_{k+1}^\top\mathbf{x}_k\|_2^2. \tag{1}$$

This definition generalizes the Dirichlet energy from graphs to simplicial complexes, measuring how similar the values assigned to adjacent simplices are, with higher energy indicating larger variation.

Simplicial filters. For a k-simplicial signal $\mathbf{x}_k$, a simplicial filter is a function $f : \mathbb{R}^{|\mathcal{X}_k|} \to \mathbb{R}^{|\mathcal{X}_k|}$ as:

$$f(\mathbf{x}_k) = \left(\sum_{i=0}^{T_d} \alpha_i \mathbf{L}_{k,d}^i + \sum_{i=0}^{T_u} \beta_i \mathbf{L}_{k,u}^i\right)\mathbf{x}_k, \tag{2}$$

where $\{\alpha_0, \ldots, \alpha_{T_d}\}$ and $\{\beta_0, \ldots, \beta_{T_u}\}$ are the parameters of the polynomials, and T_d, T_u are the order of the polynomials [34]. Please notice that the well-known graph filter [35] is a specific case of (2). In this case, the graph signal is given by $\mathbf{x}_0 \in \mathbb{R}^{|\mathcal{X}_0|}$ and since $\mathbf{B}_0$ is not defined, we have $f(\mathbf{x}_0) = \sum_{i=0}^{T_u} \beta_i \mathbf{L}_0^i \mathbf{x}_i$, *i.e.*, the classical graph filter with the graph Laplacian as the shift operator.

3.2 Discrete Simplicial Neural Network

Analogous to a convolutional neural network, an SNN can be defined for simplicial complexes using simplicial filters [17]. However, notice that relying only on filters like in (2) ignores the connections among the adjacent simplices modeled by $\mathbf{L}_k$. Since different simplicial signals influence each other via the simplicial complex localities, previous works have defined simplicial filter banks [34]. The following discrete Hodge-aware filtering model is adapted from SCCNNs [18].

One-dimensional case. Let the lower and upper projections of a simplicial signal $\mathbf{x}_k^l$ at layer l be $\mathbf{x}_{k,d}^l = \mathbf{B}_k^\top \mathbf{x}_{k-1}^l \in \mathbb{R}^{|\mathcal{X}_k|}$ and $\mathbf{x}_{k,u}^l = \mathbf{B}_{k+1} \mathbf{x}_{k+1}^l \in \mathbb{R}^{|\mathcal{X}_k|}$, respectively[1]. We define a simplicial layer (with parameters θ and ψ [18]) as a function $g : \mathbb{R}^{|\mathcal{X}_k|} \times \mathbb{R}^{|\mathcal{X}_k|} \times \mathbb{R}^{|\mathcal{X}_k|} \to \mathbb{R}^{|\mathcal{X}_k|}$ given as:

$$\mathbf{x}_k^l = \sigma\left(\mathbf{H}_{k,d}^l \mathbf{x}_{k,d}^{l-1} + \mathbf{H}_k^l \mathbf{x}_k^{l-1} + \mathbf{H}_{k,u}^l \mathbf{x}_{k,u}^{l-1}\right), \tag{3}$$

where $\mathbf{H}_{k,d}^l := \sum_{i=0}^{T_d} \theta_{k,d,i}^l \mathbf{L}_{k,d}^i$, $\mathbf{H}_{k,u}^l := \sum_{i=0}^{T_u} \theta_{k,u,i}^l \mathbf{L}_{k,u}^i$ and $\mathbf{H}_k^l := \sum_{i=0}^{T_d} \psi_{k,d,i}^l \mathbf{L}_{k,d}^i + \sum_{i=0}^{T_u} \psi_{k,u,i}^l \mathbf{L}_{k,u}^i$.

Multi-dimensional case. Let $\{\mathbf{X}_k^l, \mathbf{X}_{k,d}^l, \mathbf{X}_{k,u}^l\}$ be the F_{l-1}-dimensional simplicial signal and its lower and upper projections at layer l. Let $\mathbf{\Theta}_{k,d,i}^l$, $\mathbf{\Theta}_{k,u,i}^l$, $\mathbf{\Psi}_{k,d,i}^l$, and $\mathbf{\Psi}_{k,u,i}^l$ be learnable linear projections in $\mathbb{R}^{F_{l-1} \times F_l}$ corresponding to the α and ϕ parameters for the unidimensional case in (3). Using (2) and (3), we can define an SNN layer for the multidimensional case as follows [18]:

$$\mathbf{X}_k^l = \sigma\left(\sum_{i=0}^{T_d} \mathbf{L}_{k,d}^i \mathbf{X}_{k,d}^{l-1} \mathbf{\Theta}_{k,d,i}^l + \sum_{i=0}^{T_d} \mathbf{L}_{k,d}^i \mathbf{X}_k^{l-1} \mathbf{\Psi}_{k,d,i}^l + \sum_{i=0}^{T_u} \mathbf{L}_{k,u}^i \mathbf{X}_k^{l-1} \mathbf{\Psi}_{k,u,i}^l + \sum_{i=0}^{T_u} \mathbf{L}_{k,u}^i \mathbf{X}_{k,u}^{l-1} \mathbf{\Theta}_{k,u,i}^l\right). \tag{4}$$

The discrete SNN in (4) is analogous to the GNN case, where discrete powers of the Hodge Laplacians capture multi-hop diffusions in the simplicial signal and its lower and upper projections.

All the proofs of theorems, propositions, and lemmas of the paper are provided in Appendices A-H.

4 Continuous Simplicial Neural Network

Discrete SNNs provide flexibility in filtering simplicial signals through lower and upper projections. However, their information propagation remains fixed for each polynomial order, limiting adaptability. In this section, we introduce COSIMO, which enables a dynamic receptive field in each convolutional operation. We begin by formulating the PDEs that govern physics-informed dynamics over simplicial complexes. Next, we define the fundamental operations of COSIMO as the solutions to these PDEs. Finally, we provide a rigorous stability analysis of COSIMO, showing its robustness to topological perturbations in simplicial complexes.

4.1 PDEs in Simplicial Complexes

Our set of PDEs is inspired by heat diffusion on simplicial complexes, providing a natural extension of discrete SNNs. Intuitively, performing heat diffusion over the decoupled Hodge Laplacians enables information propagation at different rates within the continuous domain of the simplicial complex. This parallels the case in graphs [19], where continuous GNN formulations have been shown to generalize certain discrete GNNs [36], opening new possibilities for architectural design.

In our framework, considering both joint diffusion on s^k and independent diffusion on s^{k-1} and s^{k+1} allows for greater flexibility in modeling complex relationships. By enabling these dynamics to evolve at different rates, we can better adapt to the underlying topology. Motivated by these considerations, we model simplicial heat diffusion using a system of PDEs on the Hodge Laplacians. Let t_d and t_u

[1] In this work, the superscript l refers to the layer index and should not be confused with exponentiation.

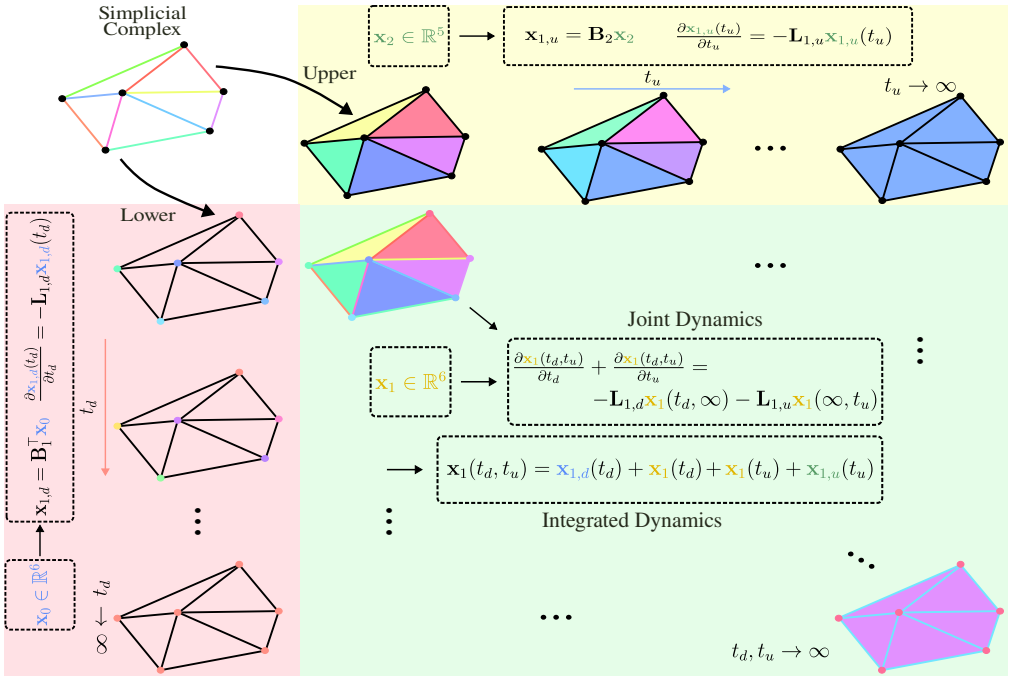

Figure 2: The PDE-based signal evolution on a simplicial complex, governed by independent diffusion processes on the lower and upper Hodge Laplacians and a coupled process integrating both spaces. The colors in the simplicial complexes represent the values of the underlying simplicial signals.

be the time variables governing the dynamics in the lower and upper Laplacians, respectively. We define these dynamics through the following system of PDEs:

Independent lower dynamics: The signal evolution in the lower space follows a heat diffusion process:

$$\frac{\partial \mathbf{x}_{k,d}(t_d)}{\partial t_d} = -\mathbf{L}_{k,d}\mathbf{x}_{k,d}(t_d). \tag{5}$$

Independent upper dynamics: Similarly, the signal in the upper space evolves according to:

$$\frac{\partial \mathbf{x}_{k,u}(t_u)}{\partial t_u} = -\mathbf{L}_{k,u}\mathbf{x}_{k,u}(t_u). \tag{6}$$

Joint dynamics: The interaction between the lower and upper spaces is captured by:

$$\frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_d} + \frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_u} = -\mathbf{L}_{k,d}\mathbf{x}_k(t_d,\infty) - \mathbf{L}_{k,u}\mathbf{x}_k(\infty,t_u), \tag{7}$$

where $\mathbf{x}_k(t_d,\infty) = \lim_{t_u\to\infty}\mathbf{x}_k(t_d,t_u)$ and $\mathbf{x}_k(\infty,t_u) = \lim_{t_d\to\infty}\mathbf{x}_k(t_d,t_u)$ state marginal stable solutions in upper and lower subspaces.

Integrated dynamics: The final solution by integration of the independent and joint dynamics is as:

$$\mathbf{x}_k(t_d,t_u) = \mathbf{x}_{k,d}(t_d) + \mathbf{x}_k(t_d) + \mathbf{x}_k(t_u) + \mathbf{x}_{k,u}(t_u). \tag{8}$$

These dynamics describe the information flow across different simplicial levels, ensuring a principled integration of independent and coupled dynamics. Refer to Fig. 2 for a visual representation on s^2.

Remark 4.1. With $t_d = t_u = t$, the joint dynamic PDE in (7) turns into $\frac{\partial \mathbf{x}_k(t)}{\partial t} = -\mathbf{L}_{k,d}\mathbf{x}_k(t) - \mathbf{L}_{k,u}\mathbf{x}_k(t)$, and its steady state solution, $\mathbf{L}_k\mathbf{x}_k = \mathbf{0}$, lies in the kernel space of $\mathbf{L}_k$ [18], justifying the need for independent lower and upper Hodge-aware PDEs.

4.2 COSIMO as a Solution to the Simplicial PDEs

We propose COSIMO as a solution to the descriptive sets of PDEs introduced in Section 4.1.

Proposition 4.2. *The solution to the descriptive sets of PDEs in Section 4.1 is given by:*

$$\mathbf{x}'_k(t_d,t_u) = \overbrace{e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_{k,d}(0)}^{\mathbf{x}_{k,d}(t_d)} + \overbrace{e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_{k,u}(0)}^{\mathbf{x}_{k,u}(t_u)} + \overbrace{e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_k(0,0) + e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_k(0,0)}^{\mathbf{x}_k(t_d,t_u)}, \tag{9}$$

where $\mathbf{x}_{k,d}(0)$*,* $\mathbf{x}_{k,u}(0)$*, and* $\mathbf{x}_k(0,0)$ *are the initial conditions for the PDEs.*

Using (9) and extending the solution to the multidimensional case, we propose the l-th layer of COSIMO as (considering $\sigma(\cdot)$ as a nonlinearity and $\{\mathbf{X}^0_k, \mathbf{X}^0_{k,d}, \mathbf{X}^0_{k,u}\}$ as the initial conditions):

$$\mathbf{X}^l_k = \sigma\Big(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^{l-1}_{k,d}\mathbf{\Theta}^l_{k,d} + e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^{l-1}_{k,u}\mathbf{\Theta}^l_{k,u} + e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^{l-1}_{k}\mathbf{\Psi}^l_{k,d} + e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^{l-1}_{k}\mathbf{\Psi}^l_{k,u}\Big), \tag{10}$$

where $\{\mathbf{\Theta}^l_{k,d}, \mathbf{\Theta}^l_{k,u}, \mathbf{\Psi}^l_{k,d}, \mathbf{\Psi}^l_{k,u}\}$ are learnable linear projections in $\mathbb{R}^{F_{l-1}\times F_l}$.

Remark 4.3. One possible approach to make our model more expressive is the aggregation of M learnable branches. Let $f^{(m)}_{k,l}(\mathbf{X}; t, \mathbf{\Theta}, \mathbf{\Psi})$ be the m-th branch of layer l as the right-hand side of the COSIMO model in (10), where $\mathbf{X} = \{\mathbf{X}^0_k, \mathbf{X}^0_{k,d}, \mathbf{X}^0_{k,u}\}$, $t^{(m)} = \{t^{(m)}_d, t^{(m)}_u\}$, $\mathbf{\Theta}^{(m)} = \{\mathbf{\Theta}^{l,(m)}_{k,d}, \mathbf{\Theta}^{l,(m)}_{k,u}\}$, and $\mathbf{\Psi}^{(m)} = \{\mathbf{\Psi}^{l,(m)}_{k,d}, \mathbf{\Psi}^{l,(m)}_{k,u}\}$. Considering $\texttt{AGG}(.)$ as a well-defined aggregation function, *e.g.*, a multilayer perceptron, the output for the l-th layer can be stated as:

$$\mathbf{X}^l_k = \texttt{AGG}(\{f^{(m)}_{k,l}(\mathbf{X}; t, \mathbf{\Theta}, \mathbf{\Psi})\}^M_{m=1}). \tag{11}$$

4.3 Computational Complexity of COSIMO

For the efficient implementation of exponential Hodge filters, we benefit from the eigenvalue decomposition (EVD) of the Hodge Laplacians [37, 21]. Precisely, for a Laplacian $\mathbf{L} \in \mathbb{R}^{N\times N}$ with the eigenvalues $\lambda_0 = 0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$, after performing EVD, $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, the exponential Laplacian filtering on input $\mathbf{X} \in \mathbb{R}^{N\times F}$ and learnable weight $\mathbf{W} \in \mathbb{R}^{F\times F'}$ can be implemented as:

$$e^{-t\mathbf{L}}\mathbf{X}\mathbf{W} \approx \mathbf{V}^{(K)}(\overbrace{[\tilde{\boldsymbol{\lambda}}^{(K)}|\ldots|\tilde{\boldsymbol{\lambda}}^{(K)}]}^{F \text{ times}} \odot (\mathbf{V}^{(K)\top}\mathbf{X}))\mathbf{W}, \tag{12}$$

where $\tilde{\boldsymbol{\lambda}} := e^{-t\,\boldsymbol{\lambda}^{(K)}} = (e^{-t\,\lambda_{N-1}}, \ldots, e^{-t\,\lambda_{N-K}})^\top$, and $\mathbf{V}^{(K)} \in \mathbb{R}^{N\times K}$ is built by choosing the K most dominant eigenvalue-eigenvector pairs of $\mathbf{L}$, and $\odot$ states the element-wise multiplication. As K gets closer to N, the estimation is more accurate. Using the implementation in (12), the computational complexity of the EVD decreases from $\mathcal{O}(N^3)$ to $\mathcal{O}(KN^2)$. Thus, the computational complexity of calculating the Hodge-aware exponential Hodge filters in (9) can be reduced from $\mathcal{O}(|\mathcal{X}_k|^3)$ to $\mathcal{O}(|\mathcal{X}_k|^2(K^{(d)}_k + K^{(u)}_k + K_k))$, where $K^{(d)}_k$, $K^{(u)}_k$, and K_k are the most dominant eigenvalue-eigenvectors of $\mathbf{L}_{k,d}$, $\mathbf{L}_{k,u}$, and $\mathbf{L}_k$, respectively. The empirical analysis regarding the trade-off on runtime, computational complexity, and performance is presented in the Appendix I.

4.4 Stability Analysis

Here, we study the robustness of our model against simplicial perturbations. We model these perturbations as structural inaccuracies in the incidence matrices, given by the additive error models $\tilde{\mathbf{B}}_k = \mathbf{B}_k + \mathbf{E}_k$ and $\tilde{\mathbf{B}}_{k+1} = \mathbf{B}_{k+1} + \mathbf{E}_{k+1}$, where $\|\mathbf{E}_k\| \leq \epsilon_k$ and $\|\mathbf{E}_{k+1}\| \leq \epsilon_{k+1}$ represent the perturbation errors. Building upon these additive models on the continuous simplicial filtering operations in (9), the following theorem bounds the COSIMO's stability:

Theorem 4.4. *Given the additive simplicial perturbation models* $\tilde{\mathbf{B}}_k = \mathbf{B}_k + \mathbf{E}_k$ *and* $\tilde{\mathbf{B}}_{k+1} = \mathbf{B}_{k+1} + \mathbf{E}_{k+1}$*, where* $\|\mathbf{E}_k\| \leq \epsilon_k$ *and* $\|\mathbf{E}_{k+1}\| \leq \epsilon_{k+1}$*, the error between true and perturbed targets in* (9)*, i.e.,* $\delta_{\mathbf{X}_k} := \|\tilde{\mathbf{X}}_k(t_d,t_u) - \mathbf{X}_k(t_d,t_u)\|$*, is bounded as:*

$$\delta_{\mathbf{X}_k} \leq t_d\delta_{k,d}e^{t_d\delta_{k,d}}\left(\|\mathbf{x}_{k,d}(0)\| + \|\mathbf{x}_k(0,0)\|\right) + t_u\delta_{k,u}e^{t_u\delta_{k,u}}\left(\|\mathbf{x}_{k,u}(0)\| + \|\mathbf{x}_k(0,0)\|\right), \tag{13}$$

where $\delta_{k,d} := 2\sqrt{\lambda_{max}(\mathbf{L}_{k,d})}\epsilon_k + \epsilon^2_k$, $\delta_{k,u} := 2\sqrt{\lambda_{max}(\mathbf{L}_{k,u})}\epsilon_{k+1} + \epsilon^2_{k+1}$.

Theorem 4.4 shows how the robustness of the model is influenced by the maximum eigenvalues of $\mathbf{L}_{k,d}$ and $\mathbf{L}_{k,u}$, as well as by the Hodge receptive fields t_d and t_u. Furthermore, the error bounds of ϵ_k and ϵ_{k+1} play a critical role in determining $\delta_{k,d}$ and $\delta_{k,u}$, which ultimately control the stability. The following corollary simplifies Theorem 4.4 under the assumption of sufficiently small error bounds.

Corollary 4.5. *When the error bounds ϵ_k and ϵ_{k+1} are sufficiently small, the error between the true and perturbed targets is given by $\delta_{\mathbf{X}_k} = \mathcal{O}(\epsilon_k) + \mathcal{O}(\epsilon_{k+1})$.*

Corollary 4.5 demonstrates stability of the proposed network against small simplicial perturbations, generalizing the stability results on the continuous GNNs [20].

5 Understanding Over-smoothing in SNNs

We first comprehensively analyze the over-smoothing problem in discrete SNNs. Next, we study over-smoothing in COSIMO highlighting the key differences with discrete SNNs. In both cases, we focus on the Dirichlet energy convergence to zero, making the simplicial signals non-discriminative. In this section, we set $F_l = F$ for all l.

5.1 Over-smoothing in Discrete SNNs

Based on the discrete SNN in (4) with $T_d = T_u = 1$ (and zeroing weights for $i = 0$) and Definition 3.1, the following theorem characterizes the over-smoothing properties of the discrete SNN.

Theorem 5.1. *In the discrete SNN in* (4) *and nonlinearity functions* $\mathrm{ReLU}(\cdot)$ *or* $\mathrm{LeakyReLU}(\cdot)$*, the Dirichlet energy of the simplicial signals at the $l+1$-th layer is bounded by the Dirichlet energy of the l-th layer and some structural and architectural characteristics as:*

$$E(\mathbf{X}_k^{l+1}) \leq \\ s\tilde{\lambda}_{max}^2 E(\mathbf{X}_k^l) + s\tilde{\lambda}_{max}^3 \left(E_{k-1}(\mathbf{X}_{k-1}^l) + E_{k+1}(\mathbf{X}_{k+1}^l)\right) + 2Fs\tilde{\lambda}_{max}^{3.5}\|\mathbf{X}_k^l\| \left(\|\mathbf{X}_{k-1}^l\| + \|\mathbf{X}_{k+1}^l\|\right), \tag{14}$$

where $\tilde{\lambda}_{max} := \max_k \{\lambda_{max}(\mathbf{L}_{k,d}), \lambda_{max}(\mathbf{L}_{k,u})\}$, *and* $s := \sqrt{\max_{k,l,i} \{\|\mathbf{\Theta}_{k,d,i}^l\|, \|\mathbf{\Theta}_{k,u,i}^l\|, \|\mathbf{\Psi}_{k,d,i}^l\|, \|\mathbf{\Psi}_{k,u,i}^l\|\}}$.

The upper bound in (14) is composed of three terms. Unlike GNN counterparts that depend solely on $E(\mathbf{X}_k^l)$ [38], this bound includes two additional terms, the second and third, which help prevent the upper bound from vanishing exponentially. This robustness has been previously analyzed in the context of SCCNNs [18] (more details in Appendix M.2). However, under some practically justified conditions, the upper bound in (14) can converge to zero, *i.e.*, the over-smoothing phenomenon, as described in the next corollary:

Corollary 5.2. *In* (14)*, if* $\tilde{\lambda}_{max} < \min\{s^{-\frac{1}{3}}, (2Fs)^{-\frac{1}{3.5}}, s^{-\frac{1}{2}}\}$*, then* $\lim_{l\to\infty} E(\mathbf{X}_k^l) \to 0$.

With the assumption in Corollary 5.2, we should modify $\tilde{\lambda}_{\max}$ to control the upper bound in (14). This involves modifying the structural properties of the simplicial complex. Therefore, preventing discrete SNNs from converging to the over-smoothing state is not straightforward.

5.2 Over-smoothing in COSIMO

The next theorem characterizes the counterpart of Theorem 5.1 in the continuous settings.

Theorem 5.3. *Considering the continuous Hodge-aware framework in* (10) *and nonlinearity functions* $\mathrm{ReLU}(\cdot)$ *or* $\mathrm{LeakyReLU}(\cdot)$*, and defining* $\varphi := \min_k \{t_d\lambda_{min}(\mathbf{L}_{k,d}), t_u\lambda_{min}(\mathbf{L}_{k,u})\}$*, it holds:*

$$E(\mathbf{X}_k^{l+1}) \leq s.(e^{-2\varphi}+1).E(\mathbf{X}_k^l) + s.e^{-2\varphi}.\tilde{\lambda}_{max}.(E(\mathbf{X}_{k-1}^l) + E(\mathbf{X}_{k+1}^l)) \\ + 2F.s.(e^{-\varphi}+e^{-2\varphi}).\tilde{\lambda}_{max}^{1.5}.\|\mathbf{X}_k^l\|.(\|\mathbf{X}_{k-1}^l\| + \|\mathbf{X}_{k+1}^l\|) + 2F.s.e^{-\varphi}.\tilde{\lambda}_{max}.\|\mathbf{X}_k^l\|^2, \tag{15}$$

where $s := \sqrt{\max_{k,l} \{\|\mathbf{\Theta}_{k,d}^l\|, \|\mathbf{\Theta}_{k,u}^l\|, \|\mathbf{\Psi}_{k,d}^l\|, \|\mathbf{\Psi}_{k,u}^l\|\}}$.

We observe that the first two terms in (15) tend to converge to zero as in (14) when stacking multiple layers. However, the third term might have a different behavior described in the following corollary.

Corollary 5.4. *The upper bound in* (15) *exponentially converges to zero by stacking layers if* $\ln(s) < \min\left\{-\ln(1+e^{-2\varphi}), 2\varphi - \ln(\tilde{\lambda}_{max}), \varphi - \ln(2F(1+e^{-\varphi})\tilde{\lambda}_{max}^{1.5}), \varphi - \ln(2F\tilde{\lambda}_{max})\right\}$.

Table 1: Accuracies in trajectory prediction on synthetic and `ocean-drifts` datasets.

Method	`synthetic` ↑	`ocean-drifts` ↑
SNN [14]	0.655 ± 0.02	0.525 ± 0.06
SCoNe [26]	0.631 ± 0.03	0.490 ± 0.08
SCNN [17]	$\mathbf{0.677 \pm 0.02}$	0.530 ± 0.08
Bunch [31]	0.623 ± 0.04	0.460 ± 0.06
SCCNN [18]	0.652 ± 0.04	$\underline{0.545 \pm 0.08}$
COSIMO	$\underline{0.659 \pm 0.04}$	$\mathbf{0.550 \pm 0.06}$

Table 2: MSE in regression on partial deformable shapes on the `Shrec-16` dataset.

Method	`small` ↓	`full` ↓
HSN [43]	0.138 ± 0.001	0.133 ± 0.001
SCACMPS [24]	0.137 ± 0.011	0.432 ± 0.001
SAN [11]	0.052 ± 0.011	0.075 ± 0.002
SCCNN [18]	$\underline{0.020 \pm 0.003}$	$\underline{0.063 \pm 0.003}$
COSIMO	$\mathbf{0.010 \pm 0.004}$	$\mathbf{0.027 \pm 0.007}$

Assuming $t_d = t_u = t$ in (9) (then $\varphi = t\lambda_{\min}(\mathbf{L}_k)$) and considering t as a hyperparameter, one heuristic to prevent over-smoothing in COSIMO is stated in the next proposition.

Proposition 5.5. *If* $\ln(s) > 2\varphi - \ln(\tilde{\lambda}_{max})$ *(violating one of the conditions in Corollary 5.4), then* $t < \frac{\ln(s\tilde{\lambda}_{max})}{2\lambda_{min}(\mathbf{L}_k)} + k_f(\mathbf{L}_k)$*, where* $k_f(\mathbf{L}_k)$ *is the finite condition number [39] of the* k*-simplex.*

Theorem 5.3 aligns with the main takeaways in the GNN literature [40, 21], where increasing the graph receptive field leads to an increase in the mixing rate of the node features, leading to a faster convergence to the over-smoothing state. We observe from Proposition 5.5 that decreasing the simplicial receptive field t can alleviate over-smoothing, which is a key difference from the discrete case discussed in Section 5.1. We experimentally validate this claim in Section 6. Besides stability and over-smoothing, we show the permutation equivariance property of COSIMO in Appendix J.

Remark 5.6. Due to the differentiability of our model in (10), the simplicial receptive fields $\{t_d, t_u\}$ can be treated as learnable parameters [37], which is the case with all of our experiments except Section 6.2. This provides a significant advantage over the discrete SNN formulation in (3), where the graph filter orders $\{T_d, T_u\}$ must be manually tuned, leading to additional computational cost.

6 Experiments and Results

We evaluate COSIMO against SOTA methods in applications of trajectory prediction, mesh regression, node and graph classification. We compare COSIMO with a wide range of graph and simplicial models, including GCN [3], GraphSAGE [41], GIN [32], GAT [42], SNN [14], SCoNe [26], SCNN [17], Bunch [31], HSN [43], SCACMPS [24], SAN [11], SaNN [44], GSAN [45] and SCCNN [18]. Then, we experimentally validate the theoretical claims in this paper. More details on the datasets and implementation are outlined in the Appendix K and L, respectively.

6.1 Real-world Applications

Trajectory prediction. Trajectory prediction involves forecasting paths within simplicial complexes. To evaluate the effectiveness of COSIMO, we assess its performance on two datasets: a synthetic simplicial complex and the `ocean-drifts` dataset from [26, 46]. As shown in Table 1, COSIMO, SCCNN, and Bunch, which incorporate inter-simplicial couplings, do not outperform SCNN on the synthetic dataset. This is likely because the input data assigned to nodes and triangles is zero, as noted in [26], making inter-simplicial couplings ineffective. However, in the `ocean-drifts` dataset, where higher-order information plays a more significant role, incorporating higher-order convolutions—as in COSIMO and SCCNN—improves the average accuracy.

Regression on partial deformable shapes. The `Shrec-16` benchmark [47] extends prior mesh classification datasets to meshes with missing parts, where each class has a full template in a neutral pose for evaluation. To increase complexity, all shapes were sampled to 10K vertices before introducing missing parts in two regular and irregular ways. This results in a dataset of 599 shapes across eight classes (humans and animals). The dataset is divided into a training set (199 shapes) and a test set (400 shapes). The main task here is to regress the correct mesh class under missing parts. We compare COSIMO against SOTA methods on two *small* and *full* versions of the dataset. As shown in Table 2, COSIMO achieves the lowest mean square error (MSE) in mesh regression on both dataset versions, outperforming all baselines. Notably, its superior performance on both small and full versions highlights its adaptability to different amounts of data, *e.g.*, even limited data.

Table 3: Accuracy results on node classification (NC) and graph classification (GC) tasks.

Method	`high-school` ↑ (NC)	`senate-bills` ↑ (NC)	`proteins` ↑ (GC)
GCN [3]	0.40 ± 0.04	$\underline{0.67 \pm 0.06}$	0.58 ± 0.05
GraphSAGE [41]	0.27 ± 0.05	0.54 ± 0.03	0.61 ± 0.03
GIN [32]	0.18 ± 0.04	0.53 ± 0.04	0.61 ± 0.03
GAT [42]	0.34 ± 0.05	0.50 ± 0.04	0.57 ± 0.06
SCNN [17]	0.81 ± 0.01	0.62 ± 0.05	0.61 ± 0.03
SCCNN [18]	$\underline{0.88 \pm 0.04}$	0.64 ± 0.09	0.69 ± 0.06
SAN [29, 11]	0.86 ± 0.04	0.53 ± 0.09	0.64 ± 0.05
SaNN [44]	0.83 ± 0.03	0.61 ± 0.08	$\underline{0.77 \pm 0.02}$
GSAN [45]	0.88 ± 0.05	*OOM*	0.77 ± 0.04
COSIMO	$\mathbf{0.90 \pm 0.05}$	$\mathbf{0.69 \pm 0.08}$	$\mathbf{0.79 \pm 0.01}$

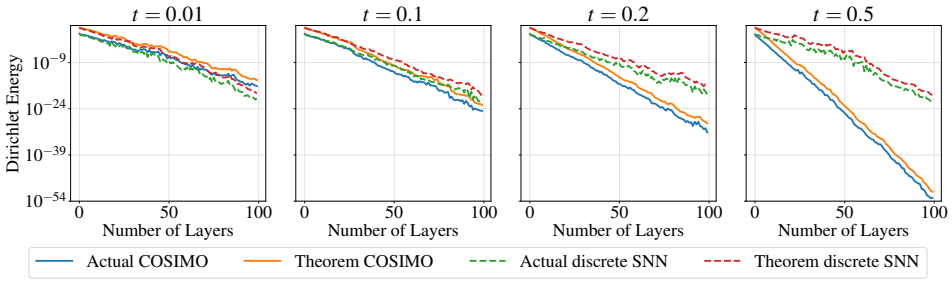

Figure 3: Over-smoothing results of discrete SNNs and COSIMO across different layer depths.

Node classification (NC). In this task, the objective is to infer categorical labels associated with 0-dimensional simplices (nodes) embedded in a simplicial complex on two benchmark datasets: `high-school` [48, 49], and `senate-bills` [48, 50, 51]. We partition each dataset chronologically, reserving the initial 80% for training the encoder and the final 20% for evaluation, as in the literature [44, 52]. Table 3 shows the results for node classification. We observe that COSIMO outperforms previous simplicial and graph-based SOTA methods.

Graph classification (GC). This task entails assigning discrete labels to entire graphs that result from a clique-lifting process applied to simplicial complexes conducted on the standard `proteins` dataset [53]. Model performance is assessed by computing the average classification accuracy over ten stratified folds [44, 52, 45] in Table 3. Similar to the NC task, COSIMO obtains superior results by leveraging the higher-order information. In fact, the Hodge-aware SNNs are mostly effective in cases of existing higher-order information (*e.g.*, on the edges, triangles, etc.). But, the poor performance of GNNs is probably due to relying only on the data over 0-order simplices, *i.e.*, the nodes, and neglecting the other simplices like edge flows or triangular dynamics.

6.2 Over-smoothing Analysis

The goals of this section are twofold: (i) to validate Theorems 5.1 and 5.3, and (ii) to study the behavior of discrete SNNs in (4) and COSIMO in (10) when facing over-smoothing. For the discrete SNN, we consider $T_d = T_u = 1$ ($i = 1$) in (4). For COSIMO in (10), we explore different scenarios by setting the receptive fields $t_d = t_u = t$ where $t \in \{10^{-2}, 10^{-1}, 0.2, 0.5\}$. In both cases, the linear projections with hidden units $F_{l-1} = F_l = 4$ are generated from normal distributions. Figure 3 shows the left-hand side (LHS) and right-hand side (RHS) of Theorems 5.1 and 5.3 averaged over 50 random realizations with number of layers varying from 1 to 100. These results validate Theorems 5.1 and 5.3, confirming that the LHSs are upper bounded by the RHSs. We also observe that adjusting t in COSIMO provides control over the over-smoothing rate, *i.e.*, how quickly the output of the SNN converges to zero Dirichlet energy. Specifically, setting $t = 10^{-2}$ results in a slower over-smoothing rate in COSIMO compared to the discrete SNN. In contrast, increasing t leads to a faster over-smoothing rate in COSIMO than in the discrete SNN. This shows that variations in the continuous receptive fields in (10) directly influence the rate of convergence to the over-smoothing state. Additional analysis has been provided in the Appendix M.

6.3 Stability Analysis

We generate 2-order simplicial complexes, following the approach in [26]: *i*) we uniformly sample $N = 30$ random points from the unit square and construct the Delaunay triangulation, *ii*) we remove triangles contained within predefined disk regions, and *iii*) we use the generative model in (9) with $k = 1$, $t_d = 1$, and $t_u = 2$, generating $\{\mathbf{X}_k^0 \in \mathbb{R}^{|\mathcal{X}_k| \times 1}\}_{k=0}^2$ from normal probability distributions. After extracting the incidence matrices $\mathbf{B}_1$ and $\mathbf{B}_2$, we include noise by varying their respective signal-to-noise ratios (SNRs) in $\{-5, 0, 10, 20\}$ dB. For each setting, we train COSIMO and evaluate its prediction performance, averaged over 30 random realizations. Figure 4 presents the results, including standard deviations. The results confirm that the model's overall stability arises from the stability on upper and lower subspaces, validating theoretical findings of Theorem 4.4.

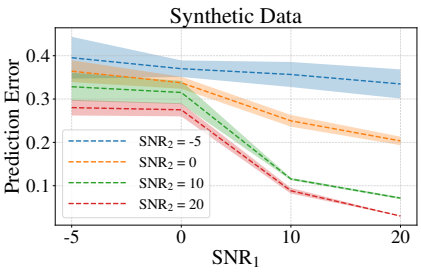

Figure 4: Stability analysis under varying SNRs.

7 Conclusion and Limitations

In this paper, we have introduced COSIMO, a novel Hodge-aware model for filtering simplicial signals that addresses the limitations of discrete SNNs by incorporating dynamic receptive fields. We provided rigorous theoretical analyses of the stability and over-smoothing behavior of our model, offering new insights into its performance. Through extensive experiments, we validated our theoretical findings, demonstrating that COSIMO is not only stable but also allows for effective control over the over-smoothing rate through its continuous receptive fields. Our experimental results highlight the superiority of COSIMO over existing SOTA SNNs, particularly in challenging trajectory prediction, regression of partial shapes, node and graph classification tasks.

Regarding future work, although we discussed in detail how to reduce the computational complexity of EVD operations, we will seek to alleviate the need for performing EVDs and potential alternatives to ease this requirement, like non-negative matrix factorization and Cholesky decomposition. Precisely, we will explore the direct implicit Euler method [54] for matrix exponential approximations to reduce the EVD computational complexity at the expense of higher runtime.

Acknowledgments

This research was supported by DATAIA Convergence Institute as part of the «Programme d'Investissement d'Avenir», (ANR-17-CONV-0003) operated by the center Hi! PARIS. This work was also partially supported by the EuroTech Universities Alliance, and the ANR French National Research Agency under the JCJC projects DeSNAP (ANR-24-CE23-1895-01) and GraphIA (ANR-20-CE23-0009-01).

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[2] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[5] S. Brussee, G. Buzzanca, A. M. Schrader, and J. Kers, "Graph neural networks in histopathology: Emerging trends and future directions," *Medical Image Analysis*, p. 103444, 2025.

[6] W. Uwents, G. Monfardini, H. Blockeel, M. Gori, and F. Scarselli, "Neural networks for relational learning: An experimental comparison," *Machine Learning*, vol. 82, no. 3, pp. 315–349, 2011.

[7] P. Gainza, F. Sverrisson, F. Monti, E. Rodola, D. Boscaini, M. M. Bronstein, and B. E. Correia, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nature Methods*, 2020.

[8] A. A. Duval, V. Schmidt, A. Hernández-Garcıa, S. Miret, F. D. Malliaros, Y. Bengio, and D. Rolnick, "FAENet: Frame averaging equivariant GNN for materials modeling," in *International Conference on Machine Learning*, 2023.

[9] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[10] J. H. Giraldo, S. Javed, and T. Bouwmans, "Graph moving object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2485–2503, 2020.

[11] L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa, "Simplicial attention neural networks," *arXiv preprint arXiv:2203.07485*, 2022.

[12] E. Isufi, G. Leus, B. Beferull-Lozano, S. Barbarossa, and P. Di Lorenzo, "Topological signal processing and learning: Recent advances and future challenges," *Signal Processing*, p. 109930, 2025.

[13] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin'on the edge... and beyond," *Signal Processing*, vol. 187, p. 108149, 2021.

[14] S. Ebli, M. Defferrard, and G. Spreemann, "Simplicial neural networks," in *Advances in Neural Information Processing Systems - Workshops*, 2020.

[15] H. Wu, A. Yip, J. Long, J. Zhang, and M. K. Ng, "Simplicial complex neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 561–575, 2023.

[16] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lio, and M. Bronstein, "Weisfeiler and Lehman go topological: Message passing simplicial networks," in *International Conference on Machine Learning*, 2021.

[17] M. Yang, E. Isufi, and G. Leus, "Simplicial convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[18] M. Yang, G. Leus, and E. Isufi, "Hodge-aware convolutional learning on simplicial complexes," *Transactions on Machine Learning Research*, 2025.

[19] A. Han, D. Shi, L. Lin, and J. Gao, "From continuous dynamics to graph neural networks: Neural diffusion and beyond," *Transactions on Machine Learning Research*, 2024.

[20] Y. Song, Q. Kang, S. Wang, K. Zhao, and W. P. Tay, "On the robustness of graph neural diffusion to topology perturbations," in *Advances in Neural Information Processing Systems*, 2022.

[21] A. Einizade, F. D. Malliaros, and J. H. Giraldo, "Continuous product graph neural networks," in *Advances in Neural Information Processing Systems*, 2024.

[22] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[23] T. Papamarkou, T. Birdal, M. M. Bronstein, G. E. Carlsson, J. Curry, Y. Gao, M. Hajij, R. Kwitt, P. Lio, P. D. Lorenzo, V. Maroulas, N. Miolane, F. Nasrin, K. N. Ramamurthy, B. Rieck, S. Scardapane, M. T. Schaub, P. Veličković, B. Wang, Y. Wang, G. Wei, and G. Zamzmi, "Position: Topological deep learning is the new frontier for relational learning," in *International Conference on Machine Learning*, 2024.

[24] M. Papillon, S. Sanborn, M. Hajij, and N. Miolane, "Architectures of topological deep learning: A survey of message-passing topological neural networks," *arXiv preprint arXiv:2304.10031*, 2023.

[25] T. M. Roddenberry and S. Segarra, "HodgeNet: Graph neural networks for edge data," in *Asilomar Conference on Signals, Systems, and Computers*, 2019.

[26] T. M. Roddenberry, N. Glaze, and S. Segarra, "Principled simplicial neural networks for trajectory prediction," in *International Conference on Machine Learning*, 2021.

[27] A. D. Keros, V. Nanda, and K. Subr, "Dist2Cycle: A simplicial neural network for homology localization," in *AAAI Conference on Artificial Intelligence*, 2022.

[28] Y. Chen, Y. R. Gel, and H. V. Poor, "BScNets: Block simplicial complex neural networks," in *AAAI Conference on Artificial Intelligence*, 2022.

[29] C. W. J. Goh, C. Bodnar, and P. Lio, "Simplicial attention networks," in *International Conference on Learning Representations - Workshop*, 2022.

[30] S. H. Lee, F. Ji, and W. P. Tay, "SGAT: Simplicial graph attention network," in *International Joint Conference on Artificial Intelligence*, 2022.

[31] E. Bunch, Q. You, G. Fung, and V. Singh, "Simplicial 2-complex convolutional neural networks," in *Advances in Neural Information Processing Systems - Workshops*, 2020.

[32] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *International Conference on Learning Representations*, 2019.

[33] M. Hajij, G. Zamzmi, T. Papamarkou, V. Maroulas, and X. Cai, "Simplicial complex representation learning," in *International Conference on Learning Representations - Workshops*, 2021.

[34] E. Isufi and M. Yang, "Convolutional filtering in simplicial complexes," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[35] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[36] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, "Grand: Graph neural diffusion," in *International conference on machine learning*, pp. 1407–1418, PMLR, 2021.

[37] M. Behmanesh, M. Krahn, and M. Ovsjanikov, "TIDE: Time derivative diffusion for deep learning on graphs," in *International Conference on Machine Learning*, 2023.

[38] C. Cai and Y. Wang, "A note on over-smoothing for graph neural networks," International Conference on Machine Learning - Workshops, 2020.

[39] D. A. Spielman, "Algorithms, graph theory, and linear equations in laplacian matrices," in *International Congress of Mathematicians*, 2010.

[40] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *International Conference on Learning Representations*, 2020.

[41] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017.

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[43] M. Hajij, K. N. Ramamurthy, A. Guzmán-Sáenz, and G. Za, "High skip networks: A higher order generalization of skip connections," in *International Conference on Learning Representations - Workshops*, 2022.

[44] S. Gurugubelli and S. P. Chepuri, "SaNN: Simple yet powerful simplicial-aware neural networks," in *International Conference on Learning Representations*, 2024.

[45] C. Battiloro, L. Testa, L. Giusti, S. Sardellitti, P. Di Lorenzo, and S. Barbarossa, "Generalized simplicial attention neural networks," *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[46] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, "Random walks on simplicial complexes and the normalized Hodge 1-laplacian," *SIAM Review*, vol. 62, no. 2, pp. 353–391, 2020.

[47] L. Cosmo, E. Rodola, M. M. Bronstein, A. Torsello, D. Cremers, Y. Sahillioğlu, *et al.*, "Shrec'16: Partial matching of deformable shapes," in *Eurographics Workshop on 3D Object Retrieval*, 2016.

[48] P. S. Chodrow, N. Veldt, and A. R. Benson, "Generative hypergraph clustering: From blockmodels to modularity," *Science Advances*, vol. 7, no. 28, p. eabh1303, 2021.

[49] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, and J. Kleinberg, "Simplicial closure and higher-order link prediction," *Proceedings of the National Academy of Sciences*, vol. 115, no. 48, pp. E11221–E11230, 2018.

[50] J. H. Fowler, "Connecting the congress: A study of cosponsorship networks," *Political analysis*, vol. 14, no. 4, pp. 456–487, 2006.

[51] J. H. Fowler, "Legislative cosponsorship networks in the us house and senate," *Social networks*, vol. 28, no. 4, pp. 454–465, 2006.

[52] H. Madhu and S. P. Chepuri, "TopoSRL: topology preserving self-supervised simplicial representation learning," in *Advances in Neural Information Processing Systems*, 2023.

[53] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[54] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, "DiffusionNet: Discretization agnostic learning on surfaces," *ACM Transactions on Graphics*, vol. 41, no. 3, pp. 1–16, 2022.

[55] C. Van Loan, "The sensitivity of the matrix exponential," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 971–981, 1977.

[56] J. Von Neumann, "Some matrix-inequalities and metrization of matrix-space," *Tomsk Univ. Rev.*, vol. 1, pp. 286–300, 1937. Reprinted in *Collected Works*, Vol. IV, Pergamon Press, 1962, pp. 205–219.

[57] M. Hajij, G. Zamzmi, T. Papamarkou, N. Miolane, A. Guzmán-Sáenz, K. N. Ramamurthy, T. Birdal, T. K. Dey, S. Mukherjee, S. N. Samaga, *et al.*, "Topological deep learning: Going beyond graph data," *arXiv preprint arXiv:2206.00606*, 2022.

[58] M. Hajij, M. Papillon, F. Frantzen, J. Agerberg, I. AlJabea, R. Ballester, C. Battiloro, G. Bernárdez, T. Birdal, A. Brent, *et al.*, "TopoX: a suite of python packages for machine learning on topological domains," *Journal of Machine Learning Research*, vol. 25, no. 374, pp. 1–8, 2024.

[59] M. De Domenico and J. Biamonte, "Spectral entropies as information-theoretic tools for complex network comparison," *Physical Review X*, vol. 6, no. 4, p. 041062, 2016.

[60] J. H. Giraldo, K. Skianis, T. Bouwmans, and F. D. Malliaros, "On the trade-off between over-smoothing and over-squashing in deep graph neural networks," in *ACM International Conference on Information and Knowledge Management*, 2023.

A Proof of Proposition 4.2

Proof. By considering the proposed solution in (9) as follows:

$$\mathbf{x}'_k(t_d,t_u) = \overbrace{e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_{k,d}(0)}^{\mathbf{x}_{k,d}(t_d)} + \overbrace{e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_{k,u}(0)}^{\mathbf{x}_{k,u}(t_u)} \quad + \overbrace{e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_k(0,0) + e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_k(0,0)}^{\mathbf{x}_k(t_d,t_u)}, \tag{16}$$

the simplified generative PDE of this formulation can be expressed as:

$$\begin{aligned}
&\frac{\partial \mathbf{x}_{k,d}(t_d)}{\partial t_d} = -\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_{k,d}(0) = -\mathbf{L}_{k,d}\mathbf{x}_{k,d}(t_d),\\
&\frac{\partial \mathbf{x}_{k,u}(t_u)}{\partial t_u} = -\mathbf{L}_{k,u}e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_{k,u}(0) = -\mathbf{L}_{k,u}\mathbf{x}_{k,u}(t_u),\\
&\mathbf{x}_k(t_d,t_u) = e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_k(0,0) + e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_k(0,0),\\
&\to \lim_{t_d\to\infty}\mathbf{x}_k(t_d,t_u) = e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_k(0,0), \quad \lim_{t_u\to\infty}\mathbf{x}_k(t_d,t_u) = e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_k(0,0),\\
&\to \frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_d} = -\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{x}_k(0,0), \quad \frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_u} = -\mathbf{L}_{k,u}e^{-t_u\mathbf{L}_{k,u}}\mathbf{x}_k(0,0).
\end{aligned} \tag{17}$$

Therefore,

$$\frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_d} + \frac{\partial \mathbf{x}_k(t_d,t_u)}{\partial t_u} = -\mathbf{L}_{k,d}\overbrace{\mathbf{x}_k(t_d,\infty)}^{\lim_{t_u\to\infty}\mathbf{x}_k(t_d,t_u)} - \mathbf{L}_{k,u}\overbrace{\mathbf{x}_k(\infty,t_u)}^{\lim_{t_d\to\infty}\mathbf{x}_k(t_d,t_u)}, \tag{18}$$

which concludes the proof. □

B Proof of Theorem 4.4

Proof. Using the additive perturbation models and the triangular inequality principle, one can write:

$$\begin{aligned}
&\|\tilde{\mathbf{L}}_{k,d} - \mathbf{L}_{k,d}\| \le 2\|\mathbf{B}_k\|\,\|\mathbf{E}_k\| + \|\mathbf{E}_k^\top\mathbf{E}_k\| \le 2\epsilon_k\sqrt{\lambda_{\max}(\mathbf{L}_{k,d})} + \epsilon_k^2,\\
&\|\tilde{\mathbf{L}}_{k,u} - \mathbf{L}_{k,u}\| \le 2\|\mathbf{B}_{k+1}\|\,\|\mathbf{E}_{k+1}\| + \|\mathbf{E}_{k+1}^\top\mathbf{E}_{k+1}\| \le 2\epsilon_{k+1}\sqrt{\lambda_{\max}(\mathbf{L}_{k,u})} + \epsilon_{k+1}^2.
\end{aligned} \tag{19}$$

Now, for a Laplacian perturbation model on $\mathbf{L}$, one can write [55, 20]

$$\|e^{-t\tilde{\mathbf{L}}} - e^{-t\mathbf{L}}\| \le t\|\tilde{\mathbf{L}} - \mathbf{L}\|\,\|e^{-t\mathbf{L}}\|e^{\|t(\tilde{\mathbf{L}}-\mathbf{L})\|}, \tag{20}$$

for a positive constant ρ. Then, by the definitions $\delta_{k,d} := 2\epsilon_k\sqrt{\lambda_{\max}(\mathbf{L}_{k,d})} + \epsilon_k^2$ and $\delta_{k,u} := 2\epsilon_{k+1}\sqrt{\lambda_{\max}(\mathbf{L}_{k,u})} + \epsilon_{k+1}^2$, the bound on the exponential Hodge filters can be obtained as:

$$\begin{aligned}
&\|e^{-t_d\tilde{\mathbf{L}}_{k,d}} - e^{-t_d\mathbf{L}_{k,d}}\| \le t_d\overbrace{\|\tilde{\mathbf{L}}_{k,d} - \mathbf{L}_{k,d}\|}^{\delta_{k,d}}\overbrace{\|e^{-t_d\mathbf{L}_{k,d}}\|}^{e^{-t_d(0)}=1}e^{\|t_d(\tilde{\mathbf{L}}_{k,d}-\mathbf{L}_{k,d})\|} \le t_d\delta_{k,d}e^{t_d\delta_{k,d}},\\
&\|e^{-t_u\tilde{\mathbf{L}}_{k,u}} - e^{-t_u\mathbf{L}_{k,u}}\| \le t_u\overbrace{\|\tilde{\mathbf{L}}_{k,u} - \mathbf{L}_{k,u}\|}^{\delta_{k,u}}\overbrace{\|e^{-t_u\mathbf{L}_{k,u}}\|}^{e^{-t_u(0)}=1}e^{\|t_u(\tilde{\mathbf{L}}_{k,u}-\mathbf{L}_{k,u})\|} \le t_u\delta_{k,u}e^{t_u\delta_{k,u}}.
\end{aligned} \tag{21}$$

Next, by considering the solution in (9), the perturbation bound can be expressed as:

$$\begin{aligned}
&\|\tilde{\mathbf{x}}_k(t_d,t_u) - \mathbf{x}_k(t_d,t_u)\|\\
&\le \|e^{-t_d\tilde{\mathbf{L}}_{k,d}} - e^{-t_d\mathbf{L}_{k,d}}\|\,\|\mathbf{x}_{k,d}(0)\| + \|e^{-t_u\tilde{\mathbf{L}}_{k,u}} - e^{-t_u\mathbf{L}_{k,u}}\|\,\|\mathbf{x}_{k,u}(0)\|\\
&+ \big(\|e^{-t_d\tilde{\mathbf{L}}_{k,d}} - e^{-t_d\mathbf{L}_{k,d}}\| + \|e^{-t_u\tilde{\mathbf{L}}_{k,u}} - e^{-t_u\mathbf{L}_{k,u}}\|\big)\,\|\mathbf{x}_k(0,0)\|\\
&\le t_d\delta_{k,d}e^{t_d\delta_{k,d}}(\|\mathbf{x}_{k,d}(0)\| + \|\mathbf{x}_k(0,0)\|) + t_u\delta_{k,u}e^{t_u\delta_{k,u}}(\|\mathbf{x}_{k,u}(0)\| + \|\mathbf{x}_k(0,0)\|).
\end{aligned} \tag{22}$$

□

C Proof of Corollary 4.5

Proof. Based on the proof stated in Appendix B, we can bound $\|\tilde{\mathbf{L}}_k - \mathbf{L}_k\|$ as:

$$\begin{aligned}\|\tilde{\mathbf{L}}_k - \mathbf{L}_k\| &\leq \|\tilde{\mathbf{L}}_{k,d} - \mathbf{L}_{k,d}\| + \|\tilde{\mathbf{L}}_{k,u} - \mathbf{L}_{k,u}\| \\ &\leq 2\|\mathbf{B}_k\|\,\|\mathbf{E}_k\| + \|\mathbf{E}_k^\top \mathbf{E}_k\| + 2\|\mathbf{B}_{k+1}\|\,\|\mathbf{E}_{k+1}\| + \|\mathbf{E}_{k+1}\mathbf{E}_{k+1}^\top\| \\ &\leq 2\epsilon_k\sqrt{\lambda_{\max}(\mathbf{L}_{k,d})} + \epsilon_k^2 + 2\epsilon_{k+1}\sqrt{\lambda_{\max}(\mathbf{L}_{k,u})} + \epsilon_{k+1}^2 \\ &\overset{\text{if } \epsilon_k \text{ and } \epsilon_{k+1} \text{ are small}}{\approx} \mathcal{O}(\epsilon_k) + \mathcal{O}(\epsilon_{k+1}).\end{aligned} \tag{23}$$

Using this direction, it can be easily seen that:

$$\begin{aligned}\|\tilde{\mathbf{L}}_{k,d} - \mathbf{L}_{k,d}\| &\leq 2\|\mathbf{B}_k\|\,\|\mathbf{E}_k\| + \|\mathbf{E}_k^\top \mathbf{E}_k\| \leq 2\epsilon_k\sqrt{\lambda_{\max}(\mathbf{L}_{k,d})} + \epsilon_k^2 \overset{\text{if } \epsilon_k \text{ is small}}{\approx} \mathcal{O}(\epsilon_k) \\ \|\tilde{\mathbf{L}}_{k,u} - \mathbf{L}_{k,u}\| &\leq 2\|\mathbf{B}_{k+1}\|\,\|\mathbf{E}_{k+1}\| + \|\mathbf{E}_{k+1}^\top \mathbf{E}_{k+1}\| \\ &\leq 2\epsilon_{k+1}\sqrt{\lambda_{\max}(\mathbf{L}_{k,u})} + \epsilon_{k+1}^2 \overset{\text{if } \epsilon_{k+1} \text{ is small}}{\approx} \mathcal{O}(\epsilon_{k+1}).\end{aligned} \tag{24}$$

Now, for a sample Laplacian $\mathbf{L}$ with $\|\tilde{\mathbf{L}} - \mathbf{L}\| = \mathcal{O}(\epsilon)$, one can write:

$$\|e^{-t\tilde{\mathbf{L}}} - e^{-t\mathbf{L}}\| \leq t\|\tilde{\mathbf{L}} - \mathbf{L}\|\,\|e^{-t\mathbf{L}}\|e^{\|t(\tilde{\mathbf{L}}-\mathbf{L})\|} = \mathcal{O}(\epsilon t e^{-\rho t}) = \mathcal{O}(\epsilon), \tag{25}$$

for a positive constant ρ [55, 20]. Therefore, by adapting (25), on can write:

$$\begin{aligned}\|e^{-t_d\tilde{\mathbf{L}}_{k,d}} - e^{-t_d\mathbf{L}_{k,d}}\| &\leq t_d\|\tilde{\mathbf{L}}_{k,d} - \mathbf{L}_{k,d}\|\,\|e^{-t_d\mathbf{L}_{k,d}}\|e^{\|t_d(\tilde{\mathbf{L}}_{k,d}-\mathbf{L}_{k,d})\|} \\ &= \mathcal{O}(\epsilon_k t_d e^{-\rho_d t_d}) = \mathcal{O}(\epsilon_k), \\ \|e^{-t_u\tilde{\mathbf{L}}_{k,u}} - e^{-t_u\mathbf{L}_{k,u}}\| &\leq t_u\|\tilde{\mathbf{L}}_{k,u} - \mathbf{L}_{k,u}\|\,\|e^{-t_u\mathbf{L}_{k,u}}\|e^{\|t_u(\tilde{\mathbf{L}}_{k,u}-\mathbf{L}_{k,u})\|} \\ &= \mathcal{O}(\epsilon_{k+1} t_u e^{-\rho_u t_u}) = \mathcal{O}(\epsilon_{k+1}).\end{aligned} \tag{26}$$

By considering (22) and (26), the proof is completed. □

D Proof of Theorem 5.1

Proof. First, by stating the Dirichlet with $E(.)$, we need the following lemmas from [38]:

Lemma D.1. *(Lemma 3.2 in [38]).* $E(\mathbf{XW}) \leq \|\mathbf{W}^\top\|_2^2 E(\mathbf{X})$.

Lemma D.2. *(Lemma 3.3 in [38]). For ReLU and Leaky-ReLU nonlinearities* $E(\sigma(\mathbf{X})) \leq E(\mathbf{X})$.

Lemma D.3. *(Von Neumann's trace inequality [56]) For two square matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of size* m *and singular values* $\sigma_i(\mathbf{A})$ *and* $\sigma_i(\mathbf{B})$*, respectively,* $tr(\mathbf{AB}) \leq \sum_{i=1}^m \sigma_i(\mathbf{A})\sigma_i(\mathbf{B}) \leq m\|\mathbf{A}\|_2\|\mathbf{B}\|_2$.

Lemma D.4. *Since* $\mathbf{B}_k\mathbf{B}_{k+1} = \mathbf{0}$*, one can obtain* $\mathbf{L}_{k,d}\mathbf{L}_{k,u} = \mathbf{0}$ *and* $\mathbf{L}_{k,d}e^{-t\mathbf{L}_{k,u}} = \mathbf{L}_{k,d}$*. Similar deductions can be obtained for* $\mathbf{L}_{k,u}$.

Lemma D.5. *Since* $\mathbf{X}_{k,d} = \mathbf{B}_k^\top\mathbf{X}_{k-1}$ *and* $\mathbf{L}_{k,d} = \mathbf{B}_k^\top\mathbf{B}_k$ *and* $\mathbf{L}_{k-1,u} = \mathbf{B}_k\mathbf{B}_k^\top$*, one can obtain* $E_d(\mathbf{X}_{k,d}) = \mathrm{tr}(\mathbf{X}_{k-1}^\top\mathbf{B}_k(\mathbf{B}_k^\top\mathbf{B}_k)\mathbf{B}_k^\top\mathbf{X}_{k-1}) \leq \lambda_{max}(\mathbf{L}_{k-1,u})E_u(\mathbf{X}_{k-1})$*. Similar deductions can be obtained for* $\mathbf{L}_{k,u}$.

Then,

$$
\begin{aligned}
&E(\mathbf{X}_k^{l+1}) = \mathrm{tr}({\mathbf{X}_k^{l+1}}^\top \mathbf{L}_{k,d}\mathbf{X}_k^{l+1}) + \mathrm{tr}({\mathbf{X}_k^{l+1}}^\top \mathbf{L}_{k,u}\mathbf{X}_k^{l+1}) \\
&\leq s\lambda_{\max}^2(\mathbf{L}_{k,d})E(\mathbf{X}_{k,d}^l) + 2Fs\lambda_{\max}^3(\mathbf{L}_{k,d})\|\mathbf{X}_{k,d}^l\|_2.\|\mathbf{X}_k^l\|_2 + s\lambda_{\max}^2(\mathbf{L}_{k,d})E_d(\mathbf{X}_k^l) \\
&+ s\lambda_{\max}^2(\mathbf{L}_{k,u})E(\mathbf{X}_{k,u}^l) + 2Fs\lambda_{\max}^3(\mathbf{L}_{k,u})\|\mathbf{X}_{k,u}^l\|_2.\|\mathbf{X}_k^l\|_2 + s\lambda_{\max}^2(\mathbf{L}_{k,u})E_u(\mathbf{X}_k^l) \\
&\leq s\lambda_{\max}^2(\mathbf{L}_{k,d})\lambda_{\max}(\mathbf{L}_{k-1,u})E_u(\mathbf{X}_{k-1}^l) + 2Fs\lambda_{\max}^{3.5}(\mathbf{L}_{k,d})\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k-1}^l\|_2 + s\lambda_{\max}^2(\mathbf{L}_{k,d})E_d(\mathbf{X}_k^l) \\
&+ s\lambda_{\max}^2(\mathbf{L}_{k,u})\lambda_{\max}(\mathbf{L}_{k+1,d})E_d(\mathbf{X}_{k+1}^l) + 2Fs\lambda_{\max}^{3.5}(\mathbf{L}_{k,u})\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k+1}^l\|_2 + s\lambda_{\max}^2(\mathbf{L}_{k,u})E_u(\mathbf{X}_k^l) \\
&\leq s\tilde{\lambda}_{\max}^3 E_u(\mathbf{X}_{k-1}^l) + 2Fs\tilde{\lambda}_{\max}^{3.5}\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k-1}^l\|_2 + s\tilde{\lambda}_{\max}^2 E_d(\mathbf{X}_k^l) \\
&+ s\tilde{\lambda}_{\max}^3 E_d(\mathbf{X}_{k+1}^l) + 2Fs\tilde{\lambda}_{\max}^{3.5}\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k+1}^l\|_2 + s\tilde{\lambda}_{\max}^2 E_u(\mathbf{X}_k^l) \\
&\leq s\tilde{\lambda}_{\max}^3(E(\mathbf{X}_{k-1}^l) + E(\mathbf{X}_{k+1}^l)) + 2Fs\tilde{\lambda}_{\max}^{3.5}\|\mathbf{X}_k^l\|.(\|\mathbf{X}_{k-1}^l\|_2 + \|\mathbf{X}_{k+1}^l\|_2) + s\tilde{\lambda}_{\max}^2 E(\mathbf{X}_k^l).
\end{aligned} \tag{27}
$$

□

E Proof of Corollary 5.2

Proof. Using the results of Theorem 5.1, if the constraints of $s\tilde{\lambda}_{\max}^3 < 1$, $2Fs\tilde{\lambda}_{\max}^{3.5} < 1$, and $s\tilde{\lambda}_{\max}^2 < 1$ are simultaneously satisfied, by stacking layers, their multiplications converge to zero making the RHS in Theorem 5.1 converge to zero as well. Holding the mentioned conditions together completes the proof. □

F Proof of Theorem 5.3

Proof. First, consider that $E(\mathbf{x})$ can be stated by $\tilde{\mathbf{x}}$, *i.e.*, the Graph Fourier Transform (GFT) [35] of $\mathbf{x}$ (where $\{\lambda_i\}_{i=1}^N$ eigenvalues of the Laplacian $\hat{\mathbf{L}}$), as follows [38]:

$$
E(\mathbf{x}) = \mathbf{x}^\top \hat{\mathbf{L}}\mathbf{x} = \sum_{i=1}^{N} \lambda_i \tilde{x}_i^2. \tag{28}
$$

Next, taking λ as the smallest nonzero eigenvalue of the Laplacian $\hat{\mathbf{L}}$, the following lemma describes the behavior of a heat kernel in the most basic scenario of over-smoothing.

Lemma F.1. *We have:*

$$
E(e^{-\hat{\mathbf{L}}}\mathbf{x}) \leq e^{-2\lambda}E(\mathbf{x}). \tag{29}
$$

Proof. By showing the EVD of $\hat{\mathbf{L}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ and $e^{-\hat{\mathbf{L}}} = \mathbf{V}e^{-\mathbf{\Lambda}}\mathbf{V}^\top$, we have:

$$
\begin{aligned}
&E(e^{-\hat{\mathbf{L}}}\mathbf{x}) \\
&= \mathbf{x}^\top \overbrace{e^{-\hat{\mathbf{L}}}}^{\mathbf{V}e^{-\mathbf{\Lambda}}\mathbf{V}^\top}{}^\top \overbrace{\hat{\mathbf{L}}}^{\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top} \overbrace{e^{-\hat{\mathbf{L}}}}^{\mathbf{V}e^{-\mathbf{\Lambda}}\mathbf{V}^\top} \mathbf{x} = \sum_{i=1}^{N} \lambda_i \tilde{x}_i^2 e^{-2\lambda_i} \leq e^{-2\lambda}\left(\sum_{i=1}^{N} \lambda_i \tilde{x}_i^2\right) = e^{-2\lambda}E(\mathbf{x}).
\end{aligned} \tag{30}
$$

Note that we excluded the zero eigenvalues because they do not engage in the calculation of the Dirichlet energy. □

Leveraging from Lemmas D.1- D.5, and also the triangle principle in inequalities, one can simply derive the following useful inequalities:

$$
\begin{aligned}
&\boxed{1}\quad E_d(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k,d}\mathbf{\Theta}^l_{k,d}) \le E_d(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k,d})\mathbf{\Theta}^l_{k,d}\|_2^2\\
&\boxed{2}\quad E_d(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,d}) \le E_d(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k})\|\mathbf{\Psi}^l_{k,d}\|_2^2\\
&\boxed{3}\quad E_d(e^{-t_d\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u}) \le E_d(\mathbf{X}^l_{k})\|\mathbf{\Psi}^l_{k,u}\|_2^2\\
&\boxed{4}\quad \mathrm{tr}({\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,d}{\mathbf{\Theta}^l_{k,d}}^\top)\\
&\qquad \le F.\|{\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k}\|_2\,\|\mathbf{\Psi}^l_{k,d}{\mathbf{\Theta}^l_{k,d}}^\top\|_2\\
&\boxed{5}\quad \mathrm{tr}({\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u}{\mathbf{\Theta}^l_{k,d}}^\top)\\
&\qquad \le F.\|{\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\|_2\,\|\mathbf{\Psi}^l_{k,u}{\mathbf{\Theta}^l_{k,d}}^\top\|_2\\
&\boxed{6}\quad \mathrm{tr}({\mathbf{X}^l_{k}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u}{\mathbf{\Theta}^l_{k,d}}^\top)\\
&\qquad \le F.\|{\mathbf{X}^l_{k}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\|_2\,\|\mathbf{\Psi}^l_{k,u}{\mathbf{\Theta}^l_{k,d}}^\top\|_2
\end{aligned}
\tag{31}
$$

Then, building upon (31) and for obtaining an upper bound on $E(\mathbf{X}^{l+1}_k) = \mathrm{tr}({\mathbf{X}^{l+1}_k}^\top\mathbf{L}_{k,d}\mathbf{X}^{l+1}_k) + \mathrm{tr}({\mathbf{X}^{l+1}_k}^\top\mathbf{L}_{k,u}\mathbf{X}^{l+1}_k)$, we first elaborate on the first term:

$$
\begin{aligned}
&\mathrm{tr}({\mathbf{X}^{l+1}_k}^\top\mathbf{L}_{k,d}\mathbf{X}^{l+1}_k)\\
&=\overbrace{\mathrm{tr}({\mathbf{\Theta}^l_{k,d}}^\top{\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k,d}\mathbf{\Theta}^l_{k,d})}^{\boxed{1}}+\overbrace{\mathrm{tr}({\mathbf{\Psi}^l_{k,d}}^\top{\mathbf{X}^l_{k}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,d})}^{\boxed{2}}\\
&+\overbrace{\mathrm{tr}({\mathbf{\Psi}^l_{k,u}}^\top{\mathbf{X}^l_{k}}^\top e^{-t_u\mathbf{L}_{k,u}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u})}^{\boxed{3}}+\overbrace{2\,\mathrm{tr}({\mathbf{\Theta}^l_{k,d}}^\top{\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,d})}^{\boxed{4}}\\
&+\overbrace{2\,\mathrm{tr}({\mathbf{\Theta}^l_{k,d}}^\top{\mathbf{X}^l_{k,d}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u})}^{\boxed{5}}+\overbrace{2\,\mathrm{tr}({\mathbf{\Psi}^l_{k,d}}^\top{\mathbf{X}^l_{k}}^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{L}_{k,d}e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}^l_{k}\mathbf{\Psi}^l_{k,u})}^{\boxed{6}}\\
&\le e^{-2t_d\lambda^{(d)}_{\min}}E_d(\underbrace{\mathbf{B}^\top_k\mathbf{X}^l_{k-1}}_{\mathbf{X}^l_{k,d}})\|\mathbf{\Theta}^l_{k,d}\|_2^2+e^{-2t_d\lambda^{(d)}_{\min}}E_d(\mathbf{X}^l_k)\|\mathbf{\Psi}^l_{k,d}\|_2^2+E_d(\mathbf{X}^l_k)\|\mathbf{\Psi}^l_{k,u}\|_2^2\\
&+2.F.\lambda^{(d)}_{\max}e^{-2t_d\lambda^{(d)}_{\min}}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k,d}\|.\|\mathbf{\Theta}^l_{k,d}\|.\|\mathbf{\Psi}^l_{k,d}\|+2.F.\lambda^{(d)}_{\max}e^{-t_d\lambda^{(d)}_{\min}}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k,d}\|.\|\mathbf{\Theta}^l_{k,d}\|.\|\mathbf{\Psi}^l_{k,d}\|\\
&+2.F.\lambda^{(d)}_{\max}e^{-t_d\lambda^{(d)}_{\min}}\|\mathbf{X}_k\|^2\\
&\le s.e^{-2\varphi}\lambda^{(u)}_{\max}E_u(\mathbf{X}^l_{k-1})+s.e^{-2\varphi}E_d(\mathbf{X}^l_k)+s.E_d(\mathbf{X}^l_k)+2.F.s.e^{-2\varphi}.\lambda^{(d)}_{\max}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k,d}\|\\
&+2.F.s.e^{-\varphi}.\lambda^{(d)}_{\max}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k,d}\|+2.F.s.e^{-\varphi}.\lambda^{(d)}_{\max}.\|\mathbf{X}^l_k\|^2\\
&\le s.e^{-2\varphi}\tilde{\lambda}_{\max}E_u(\mathbf{X}^l_{k-1})+s.(e^{-2\varphi}+1).E_d(\mathbf{X}^l_k)+2.F.s.(e^{-\varphi}+e^{-2\varphi}).\tilde{\lambda}^{1.5}_{\max}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k-1}\|+2.F.s.e^{-\varphi}.\tilde{\lambda}_{\max}.\|\mathbf{X}^l_k\|^2
\end{aligned}
\tag{32}
$$

and similarly for the second term

$$
\begin{aligned}
&\mathrm{tr}({\mathbf{X}^{l+1}_k}^\top\mathbf{L}_{k,u}\mathbf{X}^{l+1}_k)\\
&\le s.e^{-2\varphi}\tilde{\lambda}_{\max}E_d(\mathbf{X}^l_{k+1})+s.(e^{-2\varphi}+1).E_u(\mathbf{X}^l_k)+2.F.s.(e^{-\varphi}+e^{-2\varphi}).\tilde{\lambda}^{1.5}_{\max}.\|\mathbf{X}^l_k\|.\|\mathbf{X}^l_{k+1}\|+2.F.s.e^{-\varphi}.\tilde{\lambda}_{\max}.\|\mathbf{X}^l_k\|^2.
\end{aligned}
\tag{33}
$$

Therefore, the final upper bound on $E(\mathbf{X}^{l+1}_k)$ can be expressed in a combined form as

$$
\begin{aligned}
E(\mathbf{X}^{l+1}_k) &\le s.e^{-2\varphi}\tilde{\lambda}_{\max}(E(\mathbf{X}^l_{k-1})+E(\mathbf{X}^l_{k+1}))\\
&+s.(e^{-2\varphi}+1)E(\mathbf{X}^l_k)+2Fs.(e^{-\varphi}+e^{-2\varphi})\tilde{\lambda}^{1.5}_{\max}\|\mathbf{X}^l_k\|(\|\mathbf{X}^l_{k-1}\|+\|\mathbf{X}^l_{k+1}\|)+2Fs.e^{-\varphi}\tilde{\lambda}_{\max}\|\mathbf{X}^l_k\|^2
\end{aligned}
\tag{34}
$$

where

$$
\begin{aligned}
\varphi &:= \min_k\{t_d\lambda_{\min}(\mathbf{L}_{k,d}), t_u\lambda_{\min}(\mathbf{L}_{k,u})\}\\
\tilde{\lambda}_{\max} &:= \max_k\{\lambda_{\max}(\mathbf{L}_{k,d}), \lambda_{\max}(\mathbf{L}_{k,u})\}\\
s &:= \sqrt{\max_{k,l}\{\|\mathbf{\Theta}^l_{k,d}\|, \|\mathbf{\Theta}^l_{k,u}\|, \|\mathbf{\Psi}^l_{k,d}\|, \|\mathbf{\Psi}^l_{k,u}\|\}}.
\end{aligned}
\tag{35}
$$

Therefore, the proof is completed. □

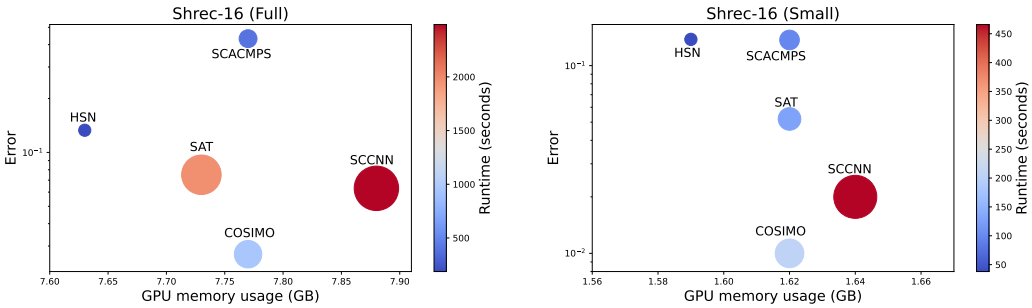

Figure 5: Comparison of the performance error and GPU memory usage (in GB) across runtime (in seconds) (in both color and circle size) on both the Small and Full versions of the `Shrec-16` dataset. The proposed COSIMO method has a good trade-off between runtime and memory usage while performing considerably better.

G Proof of Corollary 5.4

It follows similar to the justifications mentioned in Section E for the second and third terms of the results of Theorem 5.3.

H Proof of Proposition 5.5

Proof. We start from $\ln(s) > 2\varphi - \ln(\tilde{\lambda}_{\max})$. By considering the fact that $\lambda_{\min}(\mathbf{L}_{k,d}) \leq \min(\lambda_{\min}(\mathbf{L}_{k,u}), \lambda_{\min}(\mathbf{L}_k))$, assuming $t_d = t_u = t$, replacing $\varphi = t\lambda_{\min}(\mathbf{L}_k)$, and the definition $k_f(\mathbf{L}_k) := \frac{\lambda_{\max}(\mathbf{L}_k)}{\lambda_{\min}(\mathbf{L}_k)}$ (with $\lambda_{\min}(\mathbf{L}_k) \neq 0$) [39], one can write:

$$\ln(s) > 2\varphi - \ln(\tilde{\lambda}_{\max}) \rightarrow t < \frac{\ln(s\tilde{\lambda}_{\max})}{2\lambda_{\min}(\mathbf{L}_k)} < \frac{\ln(s\tilde{\lambda}_{\max}) + 2\lambda_{\max}(\mathbf{L}_k)}{2\lambda_{\min}(\mathbf{L}_k)} < \frac{\ln(s\tilde{\lambda}_{\max})}{2\lambda_{\min}(\mathbf{L}_k)} + k_f(\mathbf{L}_k). \tag{36}$$

□

I Runtime and Memory Usage Comparison in the Performance Trade-off

We provide additional comparison results of runtime (in seconds) and memory usage (in GB) on both Small and Full versions of the `Shrec-16` dataset compared to the SOTA in Fig. 5. In general and as observed from these results, the proposed COSIMO method enjoys a trade-off between runtime and memory usage while performing considerably better. More precisely, the proposed COSIMO method is ranked in the middle of the SOTA runtime and memory usage while providing significantly better accuracy performance. Note that, in COSIMO these metrics are heavily relying on the number of selected eigenvalue-eigenvector pairs in simplicial subspaces and therefore it can be optimized in a data-driven manner.

J Permutation Equivariance Property of COSIMO

Property (Permutation equivariance [26]). Consider a simplicial complex $\mathcal{X}$ characterized by boundary operators $\mathcal{B} = \{\mathbf{B}_k\}_{k=1}^K$. Let $\mathcal{P} = \{\mathbf{P}_k\}_{k=0}^K$ represent a sequence of permutation matrices, where each $\mathbf{P}_k$ is of size $|\mathcal{X}_k| \times |\mathcal{X}_k|$ and corresponds to the chain complex dimensions $\{C_k\}_{k=0}^K$, ensuring $\mathbf{P}_k \in \mathbb{R}^{|\mathcal{X}_k| \times |\mathcal{X}_k|}$. We define the permuted boundary operator as

$$[PB]_k := \mathbf{P}_{k-1}\mathbf{B}_k\mathbf{P}_k^\top.$$

A simplicial convolutional network (SCN) with the learnable weight matrix $\mathbf{W}$ is said to be permutation equivariant if, for any such transformation $\mathbf{P}$, the following holds:

$$\mathrm{SCN}_{\mathbf{W},\mathbf{B}}(\mathbf{c}_j) = \mathbf{P}_\ell\ \mathrm{SCN}_{\mathbf{W},PB}(\mathbf{P}_j\mathbf{c}_j). \tag{37}$$

Based on the above-mentioned properties, we show that COSIMO governs them in the following proposition.

Proposition J.1. *The COSIMO model stated in* (9) *exhibits the property of Permutation Equivariance.*

Proof. First, by considering $P\mathbf{L}_{k,d} = (\mathbf{P}_{k-1}\mathbf{B}_k\mathbf{P}_k)^\top(\mathbf{P}_{k-1}\mathbf{B}_k\mathbf{P}_k) = \mathbf{P}_k^\top\mathbf{L}_{k,d}\mathbf{P}_k$ and similarly $P\mathbf{L}_{k,u} = \mathbf{P}_k^\top\mathbf{L}_{k,u}\mathbf{P}_k$, and also $P\mathbf{X}_{k,d} = (\mathbf{P}_{k-1}\mathbf{B}_k\mathbf{P}_k^\top)^\top(\mathbf{P}_{k-1}\mathbf{X}_{k-1}) = \mathbf{P}_k\mathbf{B}_k^\top\mathbf{X}_{k-1} = \mathbf{P}_k\mathbf{X}_{k,d}$ and similarly $P\mathbf{X}_{k,u} = \mathbf{P}_k\mathbf{X}_{k,u}$, the permuted exponential expansion can be written as follows:

$$
\begin{aligned}
&\mathbf{P}_k\ \mathrm{COSIMO}_{\mathbf{W},PB}(\{\mathbf{P}_{k-1}\mathbf{c}_{k-1}, \mathbf{P}_k\mathbf{c}_k, \mathbf{P}_{k+1}\mathbf{c}_{k+1}\}) \\
&= \mathbf{P}_k\sigma\Big(\mathbf{P}_k^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}_{k,d}^{l-1}\mathbf{\Theta}_{k,d}^l + \mathbf{P}_k^\top e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}_{k,u}^{l-1}\mathbf{\Theta}_{k,u}^l + \mathbf{P}_k^\top e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}_k^{l-1}\mathbf{\Psi}_{k,d}^l \\
&+\mathbf{P}_k^\top e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}_k^{l-1}\mathbf{\Psi}_{k,u}^l\Big) \\
&= \sigma\Big(e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}_{k,d}^{l-1}\mathbf{\Theta}_{k,d}^l + e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}_{k,u}^{l-1}\mathbf{\Theta}_{k,u}^l + e^{-t_d\mathbf{L}_{k,d}}\mathbf{X}_k^{l-1}\mathbf{\Psi}_{k,d}^l + e^{-t_u\mathbf{L}_{k,u}}\mathbf{X}_k^{l-1}\mathbf{\Psi}_{k,u}^l\Big) \\
&= \mathrm{COSIMO}_{\mathbf{W},\mathbf{B}}(\{\mathbf{c}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k+1}\}),
\end{aligned}
\tag{38}
$$

which completes the proof. □

K More Details on the Datasets

Trajectory prediction. It is important to note that trajectory prediction in this context involves identifying a candidate node within the neighborhood of the target node, a process influenced by node degree. Given that the average node degree is 5.24 in the synthetic dataset and 4.81 in the `ocean-drifts` dataset, a random guess would achieve approximately 20% accuracy. The high standard deviations observed, particularly in the `ocean-drifts` dataset, may be attributed to its limited size. For more information, please refer to [26].

Regression on partial deformable shapes. The sampling of the shapes were in regular cuts, where template shapes were sliced at six orientations, producing 320 partial shapes, and irregular holes, where surface erosion was applied based on area budgets (40%, 70%, and 90%), yielding 279 shapes. This creates to a dataset of 599 shapes across eight classes (humans and animals), with varying missing areas (10%–60%).

Node classification. To provide more details about the NC datasets, we have outlined their statistics in Table 4.

L Implementation Details

In certain cases, we mostly use TopoModelX [57, 58, 24] to implement previous SOTA methods. For accessing and processing real-world datasets, we employ Torch TopoNetX [58]. For the experiments on trajectory prediction, we use the aggregation of M branches discussed in Remark 4.3. We use cross-validation for tuning the possible hyperparameters with the selected values provided in Table 5. Detailed hyperparameter configurations for both synthetic and real-world datasets are provided in the code in https://github.com/ArefEinizade2/COSIMO. For experimental results on `synthetic` and `ocean-drifts`, we followed the experimental settings from reference papers [26, 18]. The experiments were conducted on an A100 NVIDIA GPU with 40 GB of memory.

Number of selected eigenvalue-eigenvector K. In our framework, the selection of an appropriate K can be approached in two ways: (1) supervised and (2) unsupervised. In the supervised setting — which serves as the primary approach in this work — K is determined through cross-validation over a reasonable range of values to identify the configuration yielding the best performance. For the unsupervised case, alternative strategies can be employed. For example, we explore the use of spectral entropy [59] as a proof of concept to assess its potential effectiveness, which is defined as follows:

$$
H := -\sum_{i=1}^{n} p_i \log p_i; \quad \text{where} \quad p_i = \frac{\lambda_i}{\sum_j \lambda_j}. \tag{39}
$$

Table 4: Node classification dataset statistics.

Dataset	Simplex	# 0-simplicies	# 1-simplicies	# 2-simplicies	Order
`high-school`	Group of people	327	5818	2370	3
`senate-bills`	Co-sponsors	294	6974	3013	3

Table 5: Hyperparameter details for each dataset; lr and n_{epochs} are the learning rate and number of epochs, respectively.

Hyperparam	`synthetic`	`ocean-drifts`	`Shrec-small`	`Shrec-full`	`high-school`	`senate-bills`	`proteins`
lr	5×10^{-3}	5×10^{-2}	10^{-2}	10^{-2}	10^{-3}	10^{-2}	10^{-3}
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM	ADAM
Batch size	100	100	256	512	256	256	256
n_{epochs}	1000	1000	100	100	700	100	30
M	3	3	1	1	1	1	1

Table 6: Node classification accuracy across selected optimal value of eigenvalue-eigenvector (K) pairs by unsupervised (Spectral entropy) and supervised (cross-validation) methods on the `high-school` dataset.

	Spectral entropy	Cross validation
K	14	10
Acc (%)	92.0	91.9

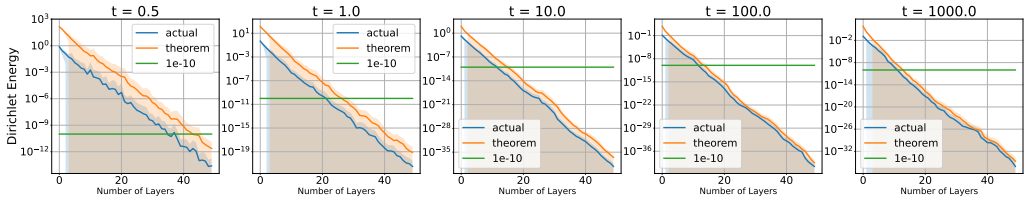

Figure 6: Over-smoothing results of COSIMO across different layer depths. Each subplot corresponds to a different parameter t in COSIMO, showing the evolution of the Dirichlet energy as a function of the number of layers.

One can choose K where the entropy contribution of the next eigenvalues becomes negligible.

We experimentally apply supervised and unsupervised methods for the selection of K on the `high-school` node classification dataset in Table 6. Note that the cross-validation method selects $\sim 3\%$ of eigenvalues for $\mathbf{L}_0$, which is fairly consistent with the spectral entropy methodology ($\sim 4.5\%$).

M A Deeper Look at Over-Smoothing

M.1 The effect of varying the Hodge receptive field t

To study the effect of varying the Hodge receptive field t on the effective number of layers, we first generate 100 random realizations of simplices with some random (filled) holes in them using the approach introduced in [26]. Then, by fixing the number of hidden features to 4, we varied the number of layers and monitored the actual and theoretical bounds in Theorem 5.3 averaged over the random realizations. The results are shown in Fig. 6. We considered a threshold of 10^{-10} as the over-smoothing occurrence threshold. As observed, increasing t reduces the effective number of layers from approximately 40 to 15. Besides, the difference between the actual and theoretical bounds gradually decreases and approaches zero, demonstrating the descriptive results of Theorem 5.3.

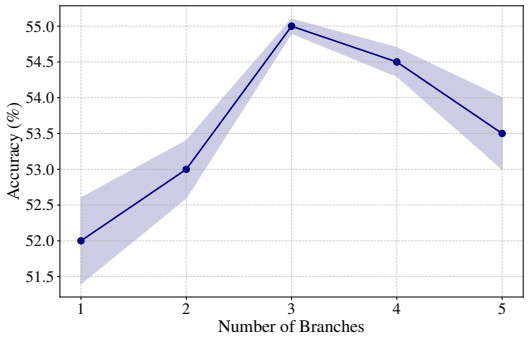

Figure 7: Sensitivity analysis on the number branches (M) of COSIMO.

Note that decreasing t too much can negatively impact the topology of the simplicial complex, potentially leading to issues like over-squashing of information [60], degrading performance. A thorough analysis of this variation is beyond the scope of the paper and is explored in future work.

M.2 Connection with related work [18]

If we consider a simplified filtering framework as $\mathbf{X}_k^{l+1} = (\mathbf{I} - \mathbf{L}_k)\mathbf{X}_k^l\mathbf{W}^l + \mathbf{X}_{k,d}^l\mathbf{W}_d^l + \mathbf{X}_{k,u}^l\mathbf{W}_u^l$ in (4), for $E(\mathbf{X}_k^{l+1}) = \mathrm{tr}(\mathbf{X}_k^{l+1^\top}\mathbf{L}_{k,d}\mathbf{X}_k^{l+1}) + \mathrm{tr}(\mathbf{X}_k^{l+1^\top}\mathbf{L}_{k,u}\mathbf{X}_k^{l+1})$, we can elaborate on the first term as:

$$\begin{aligned}
&\mathrm{tr}(\mathbf{X}_k^{l+1^\top}\mathbf{L}_{k,d}\mathbf{X}_k^{l+1}) = \mathrm{tr}([\mathbf{W}^{l^\top}\mathbf{X}_k^{l^\top}(\mathbf{I} - \mathbf{L}_{k,d})]\mathbf{L}_{k,d}[(\mathbf{I} - \mathbf{L}_{k,d})\mathbf{X}_k^l\mathbf{W}^l]) \\
&+ \mathrm{tr}(\mathbf{W}_d^{l^\top}\mathbf{X}_{k,d}^{l^\top}\mathbf{L}_{k,d}\mathbf{X}_{k,d}^l\mathbf{W}_d^l) + 2\,\mathrm{tr}([\mathbf{W}^{l^\top}\mathbf{X}_k^{l^\top}(\mathbf{I} - \mathbf{L}_{k,d})]\mathbf{L}_{k,d}\mathbf{X}_{k,d}^l\mathbf{W}_d^l) \\
&\leq s.\lambda_{max}^2(\mathbf{I} - \mathbf{L}_{k,d})E_d(\mathbf{X}_k^l) + F.s.\lambda_{max}(\mathbf{L}_{k,d}).\|\mathbf{X}_{k,d}^l\|^2 \\
&+ 2.F.s.\lambda_{max}(\mathbf{I} - \mathbf{L}_{k,d}).\lambda_{max}(\mathbf{L}_{k,d}).\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k,d}^l\|.
\end{aligned} \tag{40}$$

Similarly for the second term, we have:

$$\begin{aligned}
&\mathrm{tr}(\mathbf{X}_k^{l+1^\top}\mathbf{L}_{k,u}\mathbf{X}_k^{l+1}) \\
&\leq s.\lambda_{max}^2(\mathbf{I} - \mathbf{L}_{k,u})E_u(\mathbf{X}_k^l) + F.s.\lambda_{max}(\mathbf{L}_{k,u}).\|\mathbf{X}_{k,u}^l\|^2 \\
&+ 2.F.s.\lambda_{max}(\mathbf{I} - \mathbf{L}_{k,u}).\lambda_{max}(\mathbf{L}_{k,u}).\|\mathbf{X}_k^l\|.\|\mathbf{X}_{k,u}^l\|.
\end{aligned} \tag{41}$$

So, in a combined manner, we can write:

$$\begin{aligned}
&E(\mathbf{X}_k^{l+1}) \\
&\leq s.\lambda_{max}^2(\mathbf{I} - \mathbf{L}_k)E(\mathbf{X}_k^l) + F.s.\lambda_{max}(\mathbf{L}_{k,d}).\|\mathbf{X}_{k,d}^l\|^2 + F.s.\lambda_{max}(\mathbf{L}_{k,u}).\|\mathbf{X}_{k,u}^l\|^2 \\
&+ 2.F.s.\lambda_{max}(\mathbf{I} - \mathbf{L}_k).\lambda_{max}(\mathbf{L}_k).\|\mathbf{X}_k^l\|.(\|\mathbf{X}_{k,d}^l\| + \|\mathbf{X}_{k,u}^l\|).
\end{aligned} \tag{42}$$

If we consider the simplified framework with $F = 1$ as $\mathbf{x}_k^{l+1} = w_0(\mathbf{I} - \mathbf{L}_k)\mathbf{x}_k^l + w_1\mathbf{x}_{k,d} + w_2\mathbf{x}_{k,u}$, which was proposed in [31], the bound in (43) takes the form of:

$$\begin{aligned}
&E(\mathbf{x}_k^{l+1}) \\
&\leq s.\lambda_{max}^2(\mathbf{I} - \mathbf{L}_k)E(\mathbf{x}_k^l) + s.\lambda_{max}(\mathbf{L}_{k,d}).\|\mathbf{x}_{k,d}^l\|^2 + s.\lambda_{max}(\mathbf{L}_{k,u}).\|\mathbf{x}_{k,u}^l\|^2 \\
&+ 2.s.\lambda_{max}(\mathbf{I} - \mathbf{L}_k).\lambda_{max}(\mathbf{L}_k).\|\mathbf{x}_k^l\|.(\|\mathbf{x}_{k,d}^l\| + \|\mathbf{x}_{k,u}^l\|),
\end{aligned} \tag{43}$$

which partially coincides with the over-smoothing bound obtained in [18], with an additional fourth term missing in the previous work [18].

N Sensitivity on the Number of Branches (M) of COSIMO

Figure 7 depicts the accuracy results on the Ocean trajectory prediction task for different values of M. It is observed that increasing M up to 3 enhances the expressivity of COSIMO, as stated in Remark 4.3. However, continuing to increase beyond 3 results in performance degradation, likely due to overfitting.

Table 7: Experimental validation of the stability Theorem 4.4: the practical gap between left and right sides in (13) in Theorem 4.4.

	$\text{SNR}_1 = -5$	$\text{SNR}_1 = 0$	$\text{SNR}_1 = 10$	$\text{SNR}_1 = 20$
$\text{SNR}_2 = -5$	12.63	9.55	8.74	8.62
$\text{SNR}_2 = 0$	5.72	2.71	1.88	1.79
$\text{SNR}_2 = 10$	4.11	1.13	0.31	0.18
$\text{SNR}_2 = 20$	3.86	0.91	0.11	0.01

O Experimental Validation of the Stability Theorem 4.4

Following the SNR setting of Figure 4, Table 7 shows the difference between the right and left bounds in Theorem 4.4 in (13). We observe that, as the SNR_1 and SNR_2 increase, the gap tends to get tighter.

P Broader Impact

In the paper, we have shown the applicability of the proposed approach, *i.e.*, COSIMO, on a wide range of real-world societal applications, including ocean drift prediction, regression of the uncompleted mesh shapes, social data analysis, and protein structure classification. This can illustrate the societal impact of the COSIMO for improvement in different humanistic aspects like sea navigation, computer vision, social network development, and medical informatics.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have tried to include all the paper's contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed it at the end of the Conclusion and Limitations section as a separate paragraph for planning future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The Appendix provides all the needed proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 6 and the Appendix contain any needed detail for reproducing the results of the proposed approach.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have uploaded the reproducible codes alongside our paper and a Readme file to guide the reproduction of the main results. Besides, all of the real datasets we used are open-access benchmarks.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 and the Appendix mentioned all the needed details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our results are averaged over multiple random realizations, and we have reported the standard deviations alongside the main results whenever needed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported the used computing resources in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our paper, in every respect, meets the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed about Broader Impacts of our work in Appendix P.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the needed references and credits have been explicitly mentioned in our paper. Since the used datasets are open-access, no permission was needed.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new assets of our work are the implementation codes with detailed Readme files.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: There was no usage of LLMs in an important, original, or non-standard component of the core methods in this research, so the declaration is not required.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.