# Tangent Bundle Convolutional Learning: From Manifolds to Cellular Sheaves and Back

Claudio Battiloro, Zhiyang Wang, *Graduate Student Member, IEEE*, Hans Riess, *Member, IEEE*, Paolo Di Lorenzo, *Senior Member, IEEE*, and Alejandro Ribeiro, *Senior Member, IEEE*

***Abstract*—In this work we introduce a convolution operation over the tangent bundle of Riemann manifolds in terms of exponentials of the Connection Laplacian operator. We define tangent bundle filters and tangent bundle neural networks (TNNs) based on this convolution operation, which are novel continuous architectures operating on tangent bundle signals, i.e. vector fields over the manifolds. Tangent bundle filters admit a spectral representation that generalizes the ones of scalar manifold filters, graph filters and standard convolutional filters in continuous time. We then introduce a discretization procedure, both in the space and time domains, to make TNNs implementable, showing that their discrete counterpart is a novel principled variant of the very recently introduced sheaf neural networks. We formally prove that this discretized architecture converges to the underlying continuous TNN. Finally, we numerically evaluate the effectiveness of the proposed architecture on various learning tasks, both on synthetic and real data, comparing it against other state-of-the-art and benchmark architectures.**

***Index Terms*—Tangent bundle signal processing, tangent bundle neural networks, cellular sheaves, sheaf neural networks, graph signal processing.**

## I. Introduction

DURING the last few years, the development of deep learning techniques has led to state-of-the-art results in various fields. More and more sophisticated architectures have promoted significant improvements from both theoretical and practical perspectives. Although it is not the only reason, the success of deep learning is in part due to Convolutional Neural Networks (CNNs) [2]. CNNs have achieved excellent performances in a wide range of applications, spanning from image recognition [3] to speech analysis [4] while, at the same time, lightening the computational load of feedforward fully-connected neural networks and integrating features in different spatial resolutions with pooling operators. CNNs are based on shift operators in the space domain that induce desirable properties in the convolutional filters, among which the most relevant one is the property of shift equivariance. CNNs naturally leverage the regular (often metric) structure of the signals they process, such as spatial or temporal structure. However, data defined on irregular (non-Euclidean) domains are pervasive, with applications ranging from detection and recommendation in social networks [5], to resource allocations over wireless networks [6], and point clouds for shape segmentation [7], just to name a few. Structured data is modeled via the more varied mathematical objects, among which graphs and manifolds are notable examples. For this reason, the notions of shifts in CNNs have been adapted to convolutional architectures on graphs (GNNs) [8], [9] as well as a plethora of other structures, e.g. simplicial complexes [10], [11], [12], cell complexes [13], [14], homogeneous spaces [15], order lattices [16], and manifolds [17], [18], [19]. In [20], a framework for algebraic neural networks has been proposed exploiting commutative algebras. However, none of these studies consider convolutional filtering of vector fields over manifolds. Therefore, in this work we focus on tangent bundles, manifolds constructed from the tangent spaces of a domain manifold. Tangent bundles are a specialization of vector bundles which are a specialization of sheaves, all three of which, in increasing levels of generality, mathematically characterize both (1) when local data extends globally and (2) topological obstructions thereof. Our present focus is on tangent bundles as they are a tool for describing and processing vector fields, ubiquitous data structures critical in tasks such as robot navigation and flocking modeling, as well as in climate science [21] and astrophysics [22]. Moreover, to make the proposed procedures implementable, we formally describe and leverage the link between tangent bundles and orthogonal cellular sheaves (also called discrete vector bundles), a mathematical structure that generalizes connection graphs and matrix-weighted graphs.

Manuscript received 25 March 2023; revised 1 August 2023, 11 November 2023, and 7 February 2024; accepted 4 March 2024. Date of publication 20 March 2024; date of current version 16 April 2024. This work was funded by NSF CCF 1934960. An earlier version of this paper was presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023 [DOI: 10.1109/ICASSP49357.2023.10096934]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bo Chen. *(Corresponding author: Claudio Battiloro.)*

Claudio Battiloro is with the Biostatistics Department, Harvard University, Boston, MA 02115 USA, and also with the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: cbattiloro@hsph.harvard.edu).

Zhiyang Wang and Alejandro Ribeiro are with the Electrical and Systems Engineering Department, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: zhiyangw@seas.upenn.edu).

Hans Riess is with the Electronics and Communication Engineering Department, Duke University, Durham, NC 27708 USA.

Paolo Di Lorenzo is with the Information Engineering, Electronics, and Telecommunications Department, Sapienza University of Rome, Rome 00184 Italy.

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2024.3379862, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3379862

### A. Related Works

The well-known manifold hypothesis [23] states that high-dimensional data examples are sampled from one (or more) low-dimensional (Riemann) manifolds. This assumption is the fundamental block of manifold learning, a class of methods for non-linear dimensionality reduction. The Laplacian Eigenmap framework is based on the approximation of manifolds by weighted undirected graphs constructed with $k$-nearest neighbors or proximity radius heuristics, with the key assumption being that a set of sampled points of the manifold is available [24], [25], [26]. Formal connections between GNNs and Manifold Neural Networks (MNNs) are established in [27], [28]. Most of the previous works focused on scalar signals, e.g. one or more scalar values attached to each node of graphs or point of manifolds; however, recent developments [29], [30], [31], [32] showed that processing vector data defined on tangent bundles of manifolds or discrete vector bundles comes with a series of benefits. The work in [29] introduced a method for computing parallel transport of vector-valued data on a curved manifold by extending a vector field defined over any region to the rest of the manifold via geodesic curves. The work in [22] presented an algorithm to reconstruct the magnetopause surfaces from tangent vector observations. Pioneering works on sheaf theory can be found in [33], [34], [35]. Discrete versions of sheaves, called cellular sheaves, were first introduced in [36] and were later rediscovered in [37]. In [36], [37], these sheaves were first defined over regular cell complexes, hence the term "cellular" sheaves. Often, as in this work, cellular sheaves are defined over tamer objects, here graphs. In [30], the authors studied the problem of learning cellular sheaves from (assumed) smooth graph signals. The work in [31], [38], [39], [40] introduced a novel class of diffusion dynamics on cellular sheaves as a model for network dynamics. In [32], [41], [42], neural networks operating on discrete vector bundles are presented, generalizing GNNs: additionally, the work in [32] exploited cellular sheaf theory to show that the underlying geometry of the graph gives rise to oversmoothing behavior of GNNs. Finally, the most important works for us are [43], [44]. In particular, in [43], the authors introduced an algorithmic generalization of non-linear dimensionality reduction methods based on the Connection Laplacian operator and proved that both manifolds and their tangent bundles can be approximated with certain cellular sheaves constructed from sampled points of the manifolds. The work in [44] further generalized the result of [43] by presenting a framework for approximating Connection Laplacians over manifolds via their principal bundle structure, and by proving the spectral convergence of the approximating sheaf Laplacians.

### B. Contributions

In this work, we first define a *convolution operation over the tangent bundle* of Riemann manifolds via the Connection Laplacian operator. Our definition is derived from the vector diffusion equation over manifolds, and generalizes convolutions on manifolds [27], graphs [8], [45], as well as standard time convolutions. Leveraging this operation, we introduce *Tangent Bundle Convolutional Filters* to process tangent bundle signals (vector fields). We define the *frequency representation* of tangent bundle signals and the *frequency response* of tangent bundle filters using the spectral properties of the Connection Laplacian. By cascading layers consisting of tangent bundle filter banks and pointwise non-linearities, we introduce *Tangent Bundle Neural Networks* (TNNs). The proposed convolutional processing framework can be also seen as a novel instantiation of the general theory of algebraic signal processing [20], [46]. However, tangent bundle filters and tangent bundle neural networks are continuous architectures that cannot be directly implemented in practice. For this reason, we provide a principled way of discretizing them, both in time and space domains, making convolutions on them computable. In particular, we discretize the TNNs in the space domain by sampling points on the manifold and building a cellular sheaf [38] that represents a legitimate approximation of both the manifold and its tangent bundle [43]. We *prove that the space discretized architecture over the cellular sheaf converges to the underlying TNN* as the number of sampled points increases. Moreover, we further discretize the architecture in the time domain by sampling the filter impulse function in discrete and finite time steps, notably showing that space-time discretized TNNs (DD-TNNs) are a novel principled variant of the very recently introduced Sheaf Neural Networks [32], [41], [42], and thus shedding further light, from a theoretical point of view, on the deep connection between algebraic topology and differential geometry. Finally, we evaluate the performance of TNNs on both synthetic and real data; in particular, we design a denoising task of a synthetic tangent vector field on the torus, a manifold classification task, a reconstruction task, and a forecasting task of the daily Earth wind field, tackled via a recurrent version of our architecture. We empirically demonstrate the advantage of incorporating the tangent bundle structure into our model by comparing TNNs against Manifold Neural Networks from [27] (architectures taking into account the manifold structure, but not the tangent spaces), Multi-Layer Perceptrons [47], and Recurrent Neural Networks (the latter two do not consider any geometric information).

### C. Paper Outline

The paper is organized as follows. We introduce some preliminary concepts in Section II. We define tangent bundle convolution, filters and neural networks in Section III. In Section IV, we illustrate the proposed discretization procedure for TNNs and we prove the convergence result. We discuss the consistency of the proposed convolution in Section V. Numerical results are in Section VI, and conclusions in Section VII.

## II. Preliminary Definitions

In this section, we review some concepts from Riemann geometry that will be useful to introduce the convolution operation over tangent bundles.

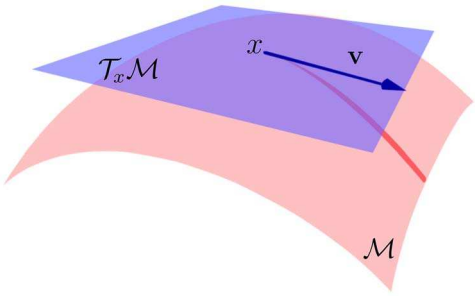



Fig. 1. An example of tangent vector.

TABLE I
NOTATION

| | |
|---|---|
| Manifold | $\mathcal{M}$ |
| Tangent Space at point $x$ | $\mathcal{T}_x\mathcal{M}$ |
| Tangent Bundle | $\mathcal{TM}$ |
| Tangent Bundle Signal | $\mathbf{F}:\mathcal{M}\to\mathcal{TM}$ |
| Differential | $d\iota:\mathcal{T}_x\mathcal{M}\to\mathcal{T}_x\mathbb{R}^p$ |
| Riemann Metric | $\langle\ ,\ \rangle_{\mathcal{T}_x\mathcal{M}}:\mathcal{T}_x\mathcal{M}\times\mathcal{T}_x\mathcal{M}\to\mathbb{R}$ |

### A. Manifolds and Tangent Bundles

We consider a compact, smooth, and orientable $d$-dimensional manifold $\mathcal{M}$ smoothly embedded in $\mathbb{R}^p$. Each point $x\in\mathcal{M}$ is endowed with a $d$-dimensional tangent space $\mathcal{T}_x\mathcal{M}$ isomorphic to $\mathbb{R}^d$, whose elements $\mathbf{v}\in\mathcal{T}_x\mathcal{M}$ are said to be tangent vectors at $x$. For explicit construction of tangent spaces on a manifold, consult an introductory textbook on differential topology [48]. Informally, tangent vectors can be seen as a generalization of the velocity vector of a curve constrained to $\mathcal{M}$ passing through the point $x$. An example of a tangent vector is depicted in Fig. 1.

*Definition 1 (Tangent Bundle):* The tangent bundle is the disjoint union of the tangent spaces $\mathcal{TM}=\bigsqcup_{x\in\mathcal{M}}\mathcal{T}_x\mathcal{M}$ together with the projection map $\pi:\mathcal{TM}\to\mathcal{M}$ given by $\pi(x,\mathbf{v})=x$.

Moreover, the tangent bundle has a natural topology which makes it a smooth $2d$-manifold and makes $\pi$ a smooth map [49]. In abuse of language, we often refer to the tangent bundle as simply the space $\mathcal{TM}$. The embedding induces a Riemann structure on $\mathcal{M}$ which allows to equip each tangent space $\mathcal{T}_x\mathcal{M}$ with an inner product.

*Definition 2 (Riemann Metric):* A Riemann Metric on a compact and smooth $d$ -dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^p$ is a (smoothly chosen) inner product $\langle\ ,\ \rangle_{\mathcal{T}_x\mathcal{M}}:\mathcal{T}_x\mathcal{M}\times\mathcal{T}_x\mathcal{M}\to\mathbb{R}$ on each of the tangent spaces $\mathcal{T}_x\mathcal{M}$ of $\mathcal{M}$ given, for each $\mathbf{v}$, $\mathbf{w}\in\mathcal{T}_x\mathcal{M}$, by

$$\langle\mathbf{v},\mathbf{w}\rangle_{\mathcal{T}_x\mathcal{M}}=\langle d\iota\mathbf{v},d\iota\mathbf{w}\rangle_{\mathbb{R}^p},\tag{1}$$

where $d\iota\mathbf{v}\in\mathcal{T}_x\mathbb{R}^p$ is called the differential of $\mathbf{v}\in\mathcal{T}_x\mathcal{M}$ in $\mathcal{T}_x\mathbb{R}^p\subset\mathbb{R}^p$, $\mathcal{T}_x\mathbb{R}^p$ is the $d$ -dimensional subspace of $\mathbb{R}^p$ being the embedding of $\mathcal{T}_x\mathcal{M}$ in $\mathbb{R}^p$, the differential $d\iota:\mathcal{T}_x\mathcal{M}\to\mathcal{T}_x\mathbb{R}^p$ is an injective linear mapping (also referred to as push-forward, as it pushes tangent vectors on $\mathcal{M}$ forward to tangent vectors on $\mathbb{R}^p$) [48], and $\langle,\rangle_{\mathbb{R}^p}$ is the usual dot product.

The Riemann metric induces also a uniform probability measure $\mu$ over the manifold, simply given by the considered region scaled by the volume of the manifold.

### B. Tangent Bundle Signals

A tangent bundle signal is a vector field over the manifold, thus a mapping $\mathbf{F}:\mathcal{M}\to\mathcal{TM}$ that associates to each point of the manifold a vector in the corresponding tangent space. In the theory of vector bundles, a bundle signal is a section. An example of a (sparse) tangent vector field over the unit 2-sphere is depicted in Fig. 3 [1].

*Remark 1:* The choice of employing the terminology "tangent bundle signal" and not the standard "vector fields" or "sections" aims to further underline the strong signal processing perspective of this work, and to facilitate the understanding of its generalization properties, as highlighted in Section V.

We can define an inner product for tangent bundle signals in the following way.

*Definition 3 (Tangent Bundle Inner Product):* Given tangent bundle signals $\mathbf{F}$ and $\mathbf{G}$, their inner product is given by

$$\langle\mathbf{F},\mathbf{G}\rangle_{\mathcal{TM}}=\int_{\mathcal{M}}\langle\mathbf{F}(x),\mathbf{G}(x)\rangle_{\mathcal{T}_x\mathcal{M}}\mathrm{d}\mu(x),\tag{2}$$

and the induced norm is $||\mathbf{F}||^2_{\mathcal{TM}}=\langle\mathbf{F},\mathbf{F}\rangle_{\mathcal{TM}}$.

We denote with $\Gamma(\mathcal{TM})$ the space of tangent bundle signals. Note that tangent bundle signals have finite energy with respect to $||\cdot||_{\mathcal{TM}}$, because they are (continuous) sections of the tangent bundle. Therefore, the length of all the vectors in a vector field is bounded because the image of a continuous function on a compact set is bounded. Hence, integrating a bounded function on (compact) $\mathcal{M}$ is always well-defined. In the following, we denote $\langle\cdot,\cdot\rangle_{\mathcal{TM}}$ with $\langle\cdot,\cdot\rangle$ when there is no risk of confusion.

## III. Tangent Bundle Convolutional Filters

Linear filtering operations are historically synonymous (under appropriate assumptions) with convolution. Time signals are filtered by computing the continuous-time convolution of the input signal and the filter impulse response [17]; images are filtered by computing multidimensional convolutions [34]; graph signals are filtered by computing graph convolutions [5]; scalar manifold signals are filtered by computing manifold convolutions [27]. In this paper, we define a tangent bundle filter as the convolution of the filter impulse response $\widetilde{h}$ and the tangent bundle signal $\mathbf{F}$. To do so, we exploit the Connection Laplacian Operator.

### A. Connection Laplacian

The Connection Laplacian is a (second-order) operator $\Delta:\Gamma(\mathcal{TM})\to\Gamma(\mathcal{TM})$, given by the trace of the second covariant derivative defined (for this work) via the Levi-Civita connection [43] (the unique connection compatible with the Riemann metric). The Connection Laplacian $\Delta$ has some desirable properties: it is negative semidefinite, self-adjoint, elliptic, and, furthermore, has a negative spectrum $\{-\lambda_i,\boldsymbol{\phi}_i\}_{i=1}^{\infty}$ with eigenvalues $\lambda_i$ and corresponding eigenvector fields $\boldsymbol{\phi}_i$ satisfying

$$\Delta\boldsymbol{\phi}_i=-\lambda_i\boldsymbol{\phi}_i,\tag{3}$$

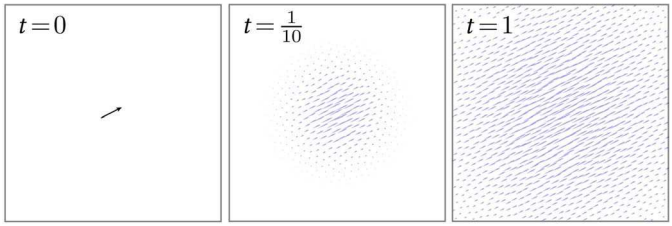



Fig. 2. Vector diffusion.

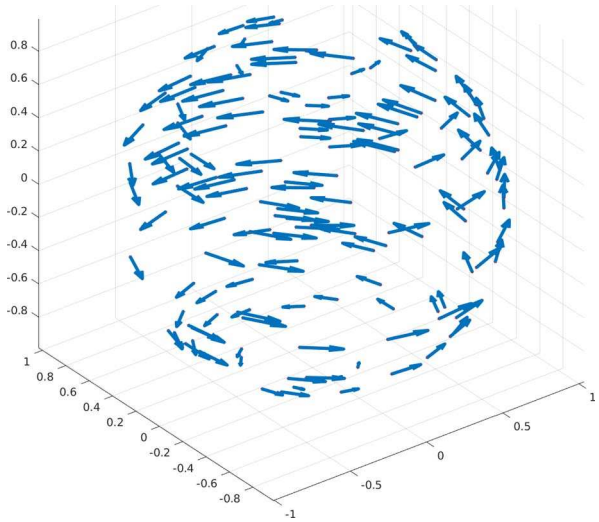


Fig. 3. An example of tangent bundle signal.

with $0 < \lambda_1 \leq \lambda_2 \leq \cdots$. The only possible accumulation (limit) point is $-\infty$ [43] We can use the Connection Laplacian to fathom a heat equation for vector diffusion:

$$\frac{\partial \mathbf{U}(x,t)}{\partial t} - \Delta \mathbf{U}(x,t) = 0, \tag{4}$$

where $\mathbf{U} : \mathcal{M} \times \mathbb{R}_0^+ \to \mathcal{TM}$ and $\mathbf{U}(\cdot,t) \in \Gamma(\mathcal{TM})\, \forall t \in \mathbb{R}_0^+$; we denote the initial condition condition with $\mathbf{U}(x,0) = \mathbf{F}(x)$. As reported in [29] and in Fig. 2 (obtained from Fig. 4 of [29]), an intuitive interpretation of (4) is imagining the evolution of the vector field $\mathbf{U}(x,t)$ over time as a "smearing out" of the initial vector field $\mathbf{F}(x)$. In this interpretation, the role of the Connection Laplacian can be understood as a means to diffuse vectors from one tangent space to another, because it encodes when tangent vectors are parallel (via the connection), and how to "move" them keeping them parallel (via the induced parallel transport). On scalar functions on Euclidean domains, it agrees with the classical Laplace operator. (Indeed, in the flat case it is sufficient to independently diffuse each scalar component, however, this approach fails for curved space.) The solution of (4) is given by

$$\mathbf{U}(x,t) = e^{t\Delta}\mathbf{F}(x), \tag{5}$$

which provides a way to construct tangent bundle convolution, as explained in the following section.

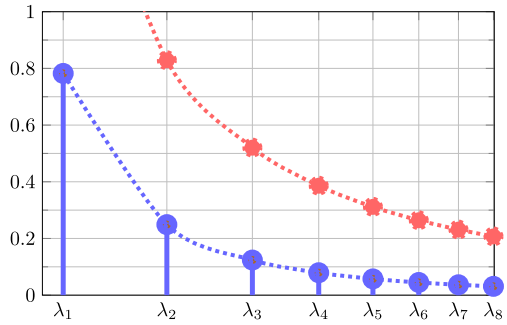



Fig. 4. Illustration of a lowpass, non-amplifying, Lipschitz continuous tangent bundle filter. The $x$-axis stands for the spectrum with each sample representing an eigenvalue. Here the eigenvalues increase at a logarithmic rate. The red dotted line is $\lambda_i^{-2}$ and the blue dotted line is the filter, obtained with impulse response $\widetilde{h}(t) = t^2/6$, thus $\hat{h}(\lambda) = \frac{1}{3}\lambda^{-3}$, from (10).

### B. Tangent Bundle Filters

We are now in the condition of defining a convolution operation and tangent bundle convolutional filters leveraging the heat diffusion dynamics in (4).

*Definition 4 (Tangent Bundle Filter):* Let $\widetilde{h} : \mathbb{R}^+ \to \mathbb{R}$ and let $\mathbf{F} \in \Gamma(\mathcal{TM})$ be a tangent bundle signal. The tangent bundle filter with impulse response $\widetilde{h}$, denoted with $\mathbf{h}$, is given by

$$\mathbf{G}(x) = \left(\widetilde{h} \star_{\mathcal{TM}} \mathbf{F}\right) = \int_0^\infty \widetilde{h}(t)\mathbf{U}(x,t)\mathrm{d}t, \tag{6}$$

where $\star_{\mathcal{TM}}$ is the *tangent bundle convolution*, and $\mathbf{U}(x,t)$ is the solution of the heat equation in (4) with $\mathbf{U}(x,0) = \mathbf{F}(x)$.

In the following, we will use the terms tangent bundle filter and tangent bundle convolution interchangeably. One cannot explicity compute the output $\mathbf{G}$ directly from the input $\mathbf{F}$ in Definition 4. However, this is remedied by injecting the solution of the heat equation (5) into (6). In this way, we can derive a closed-form expression for $\mathbf{h}$ that is parametric on the Connection Laplacian, as shown in the following proposition.

*Proposition 1 (Parametric Filter):* Any tangent bundle filter $\mathbf{h}$ defined as in (6) is a parametric map $\mathbf{h}(\Delta)$ of the Connection Laplacian operator $\Delta$, given by

$$\mathbf{G}(x) = \mathbf{h}\mathbf{F}(x) = \int_0^\infty \widetilde{h}(t)e^{t\Delta}\mathbf{F}(x)\mathrm{d}t = \mathbf{h}(\Delta)\mathbf{F}(x). \tag{7}$$

We can make several considerations starting from Proposition 1: we can state that tangent bundle filters are spatial operators, since they operate directly on points $x \in \mathcal{M}$; moreover, they are local operators, because they are parametrized by $\Delta$ which is itself a local operator.

*Remark 2:* The exponential term $e^{t\Delta}$ can be seen as a diffusion or shift operator similar to a time delay in a linear time-invariant (LTI) filter [50], or to a graph shift operator in a linear shift-invariant (LSI) graph filter [51], or to a manifold shift operator based on the Laplace-Beltrami operator [27]. The resemblance is due to the fact that tangent bundle filters are linear combinations of the elements of the tangent bundle diffusion sequence, such as graph filters are linear combinations of the elements of the graph diffusion sequence and manifold filters are linear combinations of the elements of the manifold

diffusion sequence. These considerations are further useful to validate the consistency of the proposed convolution operation, discussed in detail in Section V.

### C. Frequency Representation of Tangent Bundles Filters

The spectral properties of the Connection Laplacian $\Delta$ allow us to introduce the notion of a frequency domain. Following the approach historically common to many signal processing frameworks, we define the frequency representation of a tangent bundle signal $\mathbf{F} \in \Gamma(\mathcal{TM})$ as its projection onto the eigenbasis of the Connection Laplacian

$$\left[\hat{F}\right]_i = \langle \mathbf{F}, \boldsymbol{\phi}_i \rangle = \int_{\mathcal{M}} \langle \mathbf{F}(x), \boldsymbol{\phi}_i(x) \rangle_{\mathcal{T}_x\mathcal{M}} \mathrm{d}\mu(x). \tag{8}$$

*Proposition 2 (Frequency Representation):* Given a tangent bundle signal $\mathbf{F}$ and a tangent bundle filter $\mathbf{h}(\Delta)$ as in Definition 4, the frequency representation of the filtered signal $\mathbf{G} = \mathbf{h}(\Delta)\mathbf{F}$ is given by

$$\left[\hat{G}\right]_i = \int_0^\infty \widetilde{h}(t) e^{-t\lambda_i} \mathrm{d}t \left[\hat{F}\right]_i. \tag{9}$$

*Proof:* See Section B of Supplemental Material. □

Therefore, we can characterize the frequency response of a tangent bundle filter in the following way.

*Definition 5 (Frequency Response):* The frequency response $\hat{h}(\lambda)$ of the filter $\mathbf{h}(\Delta)$ is defined as

$$\hat{h}(\lambda) = \int_0^\infty \widetilde{h}(t) e^{-t\lambda} \mathrm{d}t. \tag{10}$$

This leads to $\left[\hat{G}\right]_i = \hat{h}(\lambda_i)\left[\hat{F}\right]_i$, meaning that the tangent bundle filter is point-wise in the frequency domain. We can finally write the frequency representation of the filter as

$$\mathbf{G} = \mathbf{h}(\Delta)\mathbf{F} = \sum_{i=1}^{\infty} \hat{h}(\lambda_i) \langle \mathbf{F}, \boldsymbol{\phi}_i \rangle \boldsymbol{\phi}_i. \tag{11}$$

*Remark 3:* The frequency response $\hat{h}(\lambda)$ in Definition 5 is the Laplace transform of $\widetilde{h}(t)$ if we let $\lambda$ be an arbitrary complex argument. The effect of a tangent bundle filter on a tangent bundle signal in the frequency domain is determined by evaluating $\hat{h}(\lambda)$ at the eigenvalues $\lambda_i$ of the Connection Laplacian. This interpretation is analogous to the interpretation of the Fourier transform as an instantiation of the Laplace transform restricted to $\lambda = j\omega$ [50]. This analogy can be furthered by observing that $j\omega$ are eigenvalues of the derivative operator (see Section V). This interpretation is also consistent with the interpretation of the frequency response of manifold filters—also a Laplace transform which is instantiated at the eigenvalues of the Laplace-Beltrami operator [27]—and the frequency response of graph filters – a $z$-transform which is instantiated at the eigenvalues of the graph shift operator [45].

### D. Lowpass Tangent Bundle Filters

The spectrum of the Connection Laplacian $\Delta$ is infinite-dimensional, i.e., there is an infinite (though countable) number of eigenvalues that need to be taken into account. However, we can design lowpass filters to tackle this problem. This design, although not mandatory for practical purposes, is crucial in proving the convergence result of the discretized filters and neural networks to the underlying continuous filters and TNNs, respectively, stated in Theorem 1.

*Definition 6 (Lowpass Tangent Bundle Filters):* A tangent bundle filter $\mathbf{h}(\Delta)$ is a lowpass filter if its frequency response function $\hat{h}$ is $\mathcal{O}(\lambda_i^{-2})$, i.e. if $\limsup_{i\to\infty} \hat{h}(\lambda_i)\lambda_i^2 < \infty$.

In other words, lowpass filters asymptotically decay at least as fast as $\lambda_i^2$, thus progressively suppressing high frequencies. Finally, we define Lipshitz continuous and non-amplifying tangent bundle filters.

*Definition 7 (Tangent Bundle Filters with Lipschitz Continuity):* A tangent bundle filter is $C$-Lispchitz if its frequency response is Lipschitz continuous with constant $C$, i.e if $|\hat{h}(a) - \hat{h}(b)| \leq C|a-b|$ for all $a, b \in (0, \infty)$.

*Definition 8 (Non-Amplifying Tangent Bundle Filters):* A tangent bundle filter is non-amplifying if for all $\lambda \in (0, \infty)$, its frequency response $\hat{h}$ satisfies $|\hat{h}(\lambda)| \leq 1$.

The Lipschitz continuity is a standard assumption, while the non-amplifying assumption is perfectly reasonable, as any (finite-energy) filter function $\hat{h}(\lambda)$ can be normalized. An example of a lowpass, non-amplifying, Lipschitz continuous tangent bundle filter is depicted in Fig. 4.

### E. Tangent Bundle Neural Networks

We define a layer of a Tangent Bundle Neural Network (TNN) as a bank of tangent bundle filters followed by a pointwise non-linearity. In this setting, pointwise informally means "pointwise in the ambient space". We introduce the notion of differential-preserving non-linearity to formalize this concept in a consistent way.

*Definition 9 (Differential-preserving Non-Linearity):* Denote with $U_x \subset \mathcal{T}_x\mathbb{R}^p$ the image of the injective differential $d\iota$ in $\mathcal{T}_x\mathbb{R}^p$. A mapping $\sigma : \Gamma(\mathcal{TM}) \to \Gamma(\mathcal{TM})$ is a differential-preserving non-linearity if it can be written as $\sigma(\mathbf{F}(x)) = d\iota^{-1}\widetilde{\sigma}_x(d\iota\mathbf{F}(x))$, where $\widetilde{\sigma}_x : U_x \to U_x$ is a point-wise non-linearity in the usual (Euclidean) sense.

Furthermore, we assume that $\widetilde{\sigma}_x = \widetilde{\sigma}$ for all $x \in \mathcal{M}$.

*Definition 10 (Tangent Bundle Neural Networks):* The $l$-th layer of a TNN with $F_l$ input signals $\{\mathbf{F}_l^q\}_{q=1}^{F_l}$, $F_{l+1}$ output signals $\{\mathbf{F}_{l+1}^u\}_{u=1}^{F_{l+1}}$, and non-linearity $\sigma(\cdot)$ is defined as

$$\mathbf{F}_{l+1}^u(x) = \sigma\left(\sum_{q=1}^{F_l} \mathbf{h}(\Delta)_l^{u,q} \mathbf{F}_l^q(x)\right), \; u = 1, ..., F_{l+1}. \tag{12}$$

Therefore, a TNN of depth $L$ with input signals $\{\mathbf{F}^q\}_{q=1}^{F_0}$ is built as the stack of $L$ layers defined as in (12), where $\mathbf{F}_0^q = \mathbf{F}^q$. An additional task-dependent readout layer (e.g sum for classification) can be appended to the final layer.

To globally represent the TNN, we collect all the filter impulse responses in a function set $\mathcal{H} = \left\{\widetilde{h}_l^{u,q}\right\}_{l,u,q}$ and we describe the TNN $u$-th output as a mapping $\mathbf{F}_L^u = \boldsymbol{\Psi}_u\left(\mathcal{H}, \Delta, \{\mathbf{F}^q\}_{q=1}^{F_0}\right)$ to emphasize that at TNN is parameterized by both $\mathcal{H}$ and the Connection Laplacian $\Delta$.

## IV. Discretization in Space and Time

Tangent Bundle Filters and Tangent Bundle Neural Networks operate on tangent bundle signals, thus they are continuous architectures that cannot be directly implemented in practice. Here we provide a procedure for discretizing tangent bundle signals, both in time and spatial domains; the discretized counterpart of TNNs is an instantiation of the recently introduced Sheaf Neural Networks [41]. For this reason, in this section we first provide a brief review of cellular sheaves over undirected graphs, and then we explain the proposed discretization procedure.

### A. Cellular Sheaves

A cellular sheaf over an (undirected) graph consists of a vector space for each node and edge and a collection of linear transformations indexed by node-edge incidence pairs of the graph. Formally, it is a functor on a partially ordered set of node-edge incidence relations into the category of vector spaces and linear transformations. We introduce the following non-standard notation to emphasize the role that sheaves play in approximating tangent bundles as the number of nodes increases.

*Definition 11 (Cellular Sheaf over a Graph):* Suppose $\mathcal{M}_n = (\mathcal{V}_n, \mathcal{E}_n)$ is an undirected graph with $n = |\mathcal{V}_n|$ nodes. A cellular sheaf over $\mathcal{M}_n$ is the tuple $\mathcal{TM}_n = (\mathcal{M}_n, \mathcal{F})$, i.e.:

- A vector space $\mathcal{F}(v)$ for each $v \in \mathcal{V}_n$. We refer to these vector spaces as node stalks.
- A vector space $\mathcal{F}(e)$ for each $e \in \mathcal{E}_n$. We refer to these vector spaces as edge stalks.
- A linear mapping $V_{v,e} : \mathcal{F}(v) \to \mathcal{F}(e)$ represented by a matrix $\mathbf{V}_{v,e}$ for each pair $(v, e) \in \mathcal{V}_n \times \mathcal{E}_n$ with incidence $v \trianglelefteq e$. These mappings are called restriction maps.

The space $\mathcal{L}^2(\mathcal{TM}_n) = \bigoplus_{v \in \mathcal{V}} \mathcal{F}(v)$ formed by the direct sum of vector spaces associated with the nodes of the graph is commonly called the space of 0-cochains, which we refer to as sheaf signals on $\mathcal{TM}_n$. We write a sheaf signal on $\mathcal{M}_n$ as $\mathbf{f}_n \in \mathcal{L}^2(\mathcal{TM}_n)$.

*Definition 12 (Sheaf Laplacian):* The (non-normalized) Sheaf Laplacian of a sheaf $\mathcal{TM}_n$ is a linear mapping $\Delta_n : \mathcal{L}^2(\mathcal{TM}_n) \to \mathcal{L}^2(\mathcal{TM}_n)$ defined node-wise

$$(\Delta_n \mathbf{f}_n)(v) = \sum_{v \trianglelefteq e \trianglerighteq u} \mathbf{V}_{v,e}^T (\mathbf{V}_{v,e} \mathbf{f}_n(v) - \mathbf{V}_{u,e} \mathbf{f}_n(u)). \quad (13)$$

While in general, the dimensions of the stalks may be arbitrary, this work focuses on discrete $\mathcal{O}(d)$-bundles, or orthogonal sheaves. In an orthogonal sheaf, we have $\mathbf{V}_{v,e}^{-1} = \mathbf{V}_{v,e}^T$ for all $v \trianglelefteq e$ and $\mathcal{F}(v) \cong \mathbb{R}^d$ for all $v$. Note, that this does not mean every stalk is equal, but has the same dimension.

### B. Discretization in the Space Domain

The manifold $\mathcal{M}$, the tangent bundle $\mathcal{TM}$, and the Connection Laplacian $\Delta$ can be approximated from a set of sampled points $\mathcal{X} \subset \mathbb{R}^p$. Knowing the coordinates of the sampled points, we construct an orthogonal cellular sheaf over an undirected geometric graph such that its normalized Sheaf Laplacian converges to the manifold Connection Laplacian as the number of sampled points (nodes) increases [44]. Formally, we assume that a set of $n$ points $\mathcal{X} = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ are sampled i.i.d. from measure $\mu$ over $\mathcal{M}$. We build a cellular sheaf $\mathcal{TM}_n$ via the Vector Diffusion Maps procedure whose details are listed in [43] and which we briefly review here.

We start by building a weighted (geometric) graph $\mathcal{M}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with nodes $\mathcal{V}_n = \{1, 2, \dots, n\}$ and weights $w_{ij}$ for nodes $i$ and $j$ as follows. Set a scale $\epsilon_n > 0$. For each pair $i, j \in \mathcal{V}_n \times \mathcal{V}_n$, if $\|x_i - x_j\|_2^2 \leq \epsilon_n$, then $ij \in \mathcal{E}_n$ with weight

$$w_{i,j} = \exp\left(\frac{||x_i - x_j||_2}{\sqrt{\epsilon_n}}\right); \quad (14)$$

otherwise, $ij \notin \mathcal{E}_n$ and $w_{i,j} = 0$ [43] (Eq. 2.5, page 6). The neighborhood $\mathcal{N}_i$ of each point $x_i$ contains the points $x_j \in \mathcal{X}$ lying in a ball of radius $\sqrt{\epsilon_n}$ centered at $x_i$. Using a local PCA procedure, we assign to each node $i$ an orthogonal transformation $\mathbf{O}_i \in \mathbb{R}^{p \times \hat{d}}$, that is an approximation of a basis of the tangent space $\mathcal{T}_{x_i}\mathcal{M}$, with $\hat{d}$ being an estimate of $d$ obtained from the same procedure (or $d$ itself, if known). In particular, we fix another scale parameter $\epsilon_{\text{PCA}}$ (different from the graph kernel scale parameter $\epsilon_n$) and we define the PCA neighborhood $\mathcal{N}_i^{\text{P}}$ of each point $x_i$ as the points $x_j \in \mathcal{X}$ lying in a ball of radius $\sqrt{\epsilon_{\text{PCA}}}$ centered at $x_i$. We define $\mathbf{X}_i \in \mathbb{R}^{p \times |\mathcal{N}_i^{\text{P}}|}$ for each point to be a matrix whose $j$-th column is the vector $x_j - x_i$, with $x_j \in \mathcal{N}_i^{\text{P}}$; equivalently, it is possible to shift each neighbor by the mean $1/|\mathcal{N}_i^{\text{P}}| \sum_{x_j \in \mathcal{N}_i^{\text{P}}} x_j$. At this point, we compute for each point a matrix $\mathbf{B}_i = \mathbf{X}_i \mathbf{C}_i$, where $\mathbf{C}_i$ is a diagonal matrix whose entry are defined as $\mathbf{C}(i,i) = \sqrt{K(||x_i - x_j||_2/\sqrt{\epsilon_{\text{PCA}}})}$, with $K(\cdot)$ being any twice differentiable positive monotone function supported on $[0, 1]$ (this scaling is useful to emphasize nearby points over far away points). We now perform the actual Local PCA by computing, per each point, the following covariance matrix and its eigendecomposition

$$\mathbf{R}_i = \mathbf{B}_i^T \mathbf{B}_i = \mathbf{M}_i \Sigma_i \mathbf{M}_i^T. \quad (15)$$

*Definition 13 (Approximated Tangent Space [43] (Eq. 2.1, page 5)):* For each point $x_i \in \mathcal{X} \subset \mathcal{M}$, the approximated basis $\mathbf{O}_i$ of its tangent space $\mathcal{T}_{x_i}\mathcal{M}$ is given by the $\hat{d}$ largest left eigenvectors of the covariance matrix $\mathbf{R}_i$ from (15), where $\hat{d}$ is an estimate of $\dim(\mathcal{M})$ or $\dim(\mathcal{M})$ itself, if known.

When the true manifold dimension $d$ is not known, it is possible to estimate it directly from the sampled points. In the ideal case of neighboring points in $\mathcal{N}_i^P$ being located exactly on $\mathcal{T}_{x_i}\mathcal{M}$, it holds that $\text{rank}(\mathbf{X}_i) = \text{rank}(\mathbf{B}_i) = d$, therefore only $d$ singular values are non-vanishing. In this ideal case, $d$ can be obviously estimated as the number of singular values different from zero. However, there may usually be more than $d$ non-vanishing singular values due to the curvature effect. In this case, it is possible to estimate the dimension $d$ as the number of singular values accounting for a certain (high) percentage of the variability of the displacements in $\mathbf{B}_i$. In practice, denoting the singular values of $\mathbf{B}_i$ with $\beta_{i,1} \geq \beta_{i,2} \geq \cdots \geq \beta_{i,|\mathcal{N}_i^P|}$, a threshold $0 < \gamma \leq 1$ (possibly close to 1) is chosen and a local dimension $\hat{d}_i$ is estimated per each point $x_i$ as the smallest number of

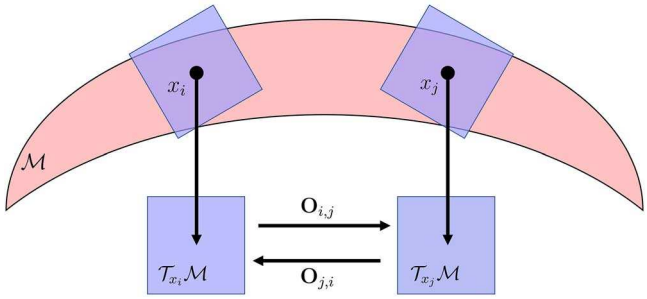



Fig. 5. Pictorial view of discrete parallel transport.

singular values for which $\sum_{j=1}^{\hat{d}_i}\beta_{i,j}/\sum_{j=1}^{|\mathcal{N}_i^P|}\beta_{i,j}\geq\gamma$. For example, setting $\gamma=0.8$ means that $\hat{d}_i$ singular values account for at least $80\%$ variability of the displacements. At this point, the estimate $\hat{d}$ of the dimension $d$ of the manifold is obtained as the (integer) mean or the median of the local estimated dimensions $\{\hat{d}_i\}_{i=1}^n$ [43]. Definition 13 is equivalent to say that $\mathbf{O}_i$ is built with the first $\hat{d}$ columns of $\mathbf{M}_i$ from (15). Moreover, as usual, $\mathbf{O}_i$ can be equivalently (and efficiently) computed as the first $\hat{d}$ left singular vectors of $\mathbf{B}_i$, without explicitly computing the covariance matrix $\mathbf{R}_i$. The local PCA procedure is summarized in Algorithm 1 in Section A of the Supplemental Material. Now, a discrete approximation of the parallel transport operator [48], that is a linear transformation from $\mathcal{T}_{x_i}\mathcal{M}$ to $\mathcal{T}_{x_j}\mathcal{M}$, is needed. In the discrete domain, this translates to associating a matrix to each edge of the above graph. For $\epsilon_n$ small enough, $\mathcal{T}_{x_i}\mathcal{M}$ and $\mathcal{T}_{x_j}\mathcal{M}$ are close, meaning that the column spaces of $\mathbf{O}_i$ and $\mathbf{O}_j$ are similiar. If the column spaces coincide, then the matrices $\mathbf{O}_i$ and $\mathbf{O}_j$ are related by an orthogonal transformation $\widetilde{\mathbf{O}}_{i,j}=\mathbf{O}_i{}^T\mathbf{O}_j$. However, if $\mathcal{M}$ is curved, the column spaces of $\mathbf{O}_i$ and $\mathbf{O}_j$ will not coincide. For this reason, the transport operator approximation $\mathbf{O}_{i,j}$ is defined as the closest orthogonal matrix to $\widetilde{\mathbf{O}}_{i,j}$ [43] (Eq. 2.4, page 6), i.e.:

$$\mathbf{O}_{i,j}=\underset{\mathbf{O}:\mathbf{O}^T\mathbf{O}=\mathbf{I}}{\operatorname{argmin}}\|\mathbf{O}-\widetilde{\mathbf{O}}_{i,j}\|_{HS}, \tag{16}$$

where $\|\cdot\|_{HS}$ is the Hilbert-Schmidt norm. The solution of problem (16) is given by $\mathbf{O}_{i,j}=\mathbf{M}_{i,j}\mathbf{V}_{i,j}^T\in\mathbb{R}^{\hat{d}\times\hat{d}}$, where $\mathbf{M}_{i,j}$ and $\mathbf{V}_{i,j}$ are the SVD of $\widetilde{\mathbf{O}}_{i,j}=\mathbf{M}_{i,j}\mathbf{\Sigma}_{i,j}\mathbf{V}_{i,j}^T$ (and the restriction maps of the approximating sheaf); a pictorial view of this discrete approximating transport is presented in Fig. 5. We now build a block matrix $\mathbf{S}\in\mathbb{R}^{n\hat{d}\times n\hat{d}}$ and a diagonal block matrix $\mathbf{D}\in\mathbb{R}^{n\hat{d}\times n\hat{d}}$ with $\hat{d}\times\hat{d}$ blocks defined as

$$\mathbf{S}_{i,j}=w_{i,j}\widetilde{\mathbf{D}}_i^{-1}\mathbf{O}_{i,j}\widetilde{\mathbf{D}}_j^{-1},\quad \mathbf{D}_{i,i}=\text{ndeg }(i)\mathbf{I}_{\hat{d}}, \tag{17}$$

where $\widetilde{\mathbf{D}}_i=\deg(i)\mathbf{I}_{\hat{d}}$, $\deg(i)=\sum_j w_{i,j}$ is the degree of node $i$, and $\text{ndeg}(i)=\sum_j w_{i,j}/(\deg(i)\deg(j))$ is the normalized degree of node $i$. Finally, we define the (normalized) Sheaf Laplacian as the following matrix

$$\Delta_n=\epsilon_n^{-1}\left(\mathbf{D}^{-1}\mathbf{S}-\mathbf{I}\right)\in\mathbb{R}^{n\hat{d}\times n\hat{d}}, \tag{18}$$

which is the approximated Connection Laplacian of the underlying manifold $\mathcal{M}$ [43] (page 13). The procedure to build the Sheaf Laplacian is summarized in Algorithm 2 in Section A of the Supplemental Material. A sheaf $\mathcal{TM}_n$ with this (orthogonal) structure represents a discretized version of $\mathcal{TM}$. Further details in [43].

At this point, we introduce a linear sampling operator $\mathbf{\Omega}_n^{\mathcal{X}}:\Gamma(\mathcal{TM})\to\Gamma(\mathcal{TM}_n)$ to discretize a tangent bundle signal $\mathbf{F}$ as a sheaf signal $\mathbf{f}_n\in\mathbb{R}^{n\hat{d}}$ such that (refer to Appendix A for the rigorous definition of $\Gamma(\mathcal{TM}_n)$):

$$\mathbf{f}_n=\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \tag{19}$$

$$\mathbf{f}_n(x_i):=[\mathbf{f}_n]_{((i-1)\hat{d}+1):(i+1)\hat{d}}=\mathbf{O}_i{}^T d\iota\mathbf{F}(x_i)\in\mathbb{R}^{\hat{d}}, \tag{20}$$

where $((i-1)\hat{d}+1):(i+1)\hat{d}$ indicates all the components of $\mathbf{f}_n$ from the $((i-1)\hat{d}+1)$-th to the $(i+1)\hat{d}$-th component. In words, the sampling operator $\mathbf{\Omega}_n^{\mathcal{X}}$ in (19) takes the embedded tangent signal $d\iota\mathbf{F}$ as input, evaluates it on each point $x_i$ in the sampling set $\mathcal{X}$, projects the evaluated signals $d\iota_{x_i}(\mathbf{F}(x_i))\in\mathbb{R}^p$ over the $d$-dimensional subspaces spanned by the $\mathbf{O}_i$s from Definition 13 and, finally, sequentially collects the $n$ projections $\mathbf{O}_i{}^T d\iota\mathbf{F}(x_i)\in\mathbb{R}^{\hat{d}}$ in the vector $\mathbf{f}_n\in\mathbb{R}^{n\hat{d}}$, representing the discretized tangent bundle signal. We are now in the condition of plugging the discretized operator from (18) and signal from (19) in the definition of tangent bundle filter from (7), obtaining:

$$\mathbf{g}_n=\int_0^\infty\widetilde{h}(t)e^{t\Delta_n}\mathbf{f}_n\mathrm{d}t=\mathbf{h}(\Delta_n)\mathbf{f}_n\in\mathbb{R}^{n\hat{d}}. \tag{21}$$

Following the same considerations of Section III-E, we can define a discretized space tangent bundle neural network (D-TNN) as the stack of $L$ layers of the form

$$\mathbf{f}_{n,l+1}^u=\sigma\left(\sum_{q=1}^{F_l}\mathbf{h}(\Delta_n)_l^{u,q}\mathbf{f}_{n,l}^q\right),\ u=1,...,F_{l+1}, \tag{22}$$

where (with a slight abuse of notation) $\sigma$ has the same point-wise law of $\widetilde{\sigma}$ in Definition 9. As in the continuous case, we describe the $u$th output of a D-TNN as a mapping $\mathbf{\Psi}_u\big(\mathcal{H},\Delta_n,\{\mathbf{x}_n^q\}_{q=1}^{F_0}\big)$ to emphasize that it is parameterized by filters $\mathcal{H}$ and the Sheaf Laplacian $\Delta_n$. The D-TNN architecture comes with desirable theoretical properties. As the number of sampling points goes to infinity, the Sheaf Laplacian $\Delta_n$ converges to the Connection Laplacian $\Delta$ [43] and the sheaf signal $\mathbf{x}_n$ consequently converges to the tangent bundle signal $\mathbf{F}$. Combining these results, we prove in the next theorem that the output of a D-TNN converges to the output of the corresponding underlying TNN as the sample size increases, validating the approximation fitness of a D-TNN. To the best of our knowledge, this is the first result to *formally* connect Sheaf Neural Networks to tangent bundles of Riemann manifolds. Let us denote the injectivity radius and the condition number [44], [48] of the manifold $\mathcal{M}$ with $\kappa$ and $\tau$, respectively.

*Theorem 1:* Let $\mathcal{X}=\{x_1,\ldots,x_n\}\subset\mathbb{R}^p$ be a set of $n$ i.i.d. sampled points from measure $\mu$ over $\mathcal{M}\subset\mathbb{R}^p$ and $\mathbf{F}$ a tangent bundle signal. Let $\mathcal{TM}_n$ be the cellular sheaf built from $\mathcal{X}$ as explained above with $\hat{d}=d$ and $0<\epsilon_n\leq\min\{\kappa,\tau^{-1}\}$. Let $\mathbf{\Psi}_u\big(\mathcal{H},\cdot,\cdot\big)$ be the $u$th output of a neural network of $L$ layers parameterized by the operator $\Delta$ of $\mathcal{TM}$ or by the discrete operator $\Delta_n$ of $\mathcal{TM}_n$. If:

TABLE II
NOTATION AND SUMMARY OF THE CONTINUOUS (ON TANGENT BUNDLES) FRAMEWORK AND ITS DISCRETIZATION (ON CELLULAR SHEAVES)

| | Tangent Bundle $\mathcal{TM}$ | Cellular Sheaf $\mathcal{TM}_n$ |
|---|---|---|
| Signal | $\mathbf{F}$ | $\mathbf{f}_n$ |
| Laplacian | $\Delta$ | $\Delta_n$ |
| Inner product | $\langle \mathbf{F}, \mathbf{G} \rangle_{\mathcal{TM}} = \int_{\mathcal{M}} \langle \mathbf{F}(x), \mathbf{G}(x) \rangle_{\mathcal{T}_x\mathcal{M}} \mathrm{d}\mu(x)$ | $\langle \mathbf{f}_n, \mathbf{g}_n \rangle_{\mathcal{TM}_n} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{f}_n(x_i) \cdot \mathbf{u}_n(x_i)$ |
| Filter | $\mathbf{G} = \int_0^\infty \widetilde{h}(t) e^{t\Delta} \mathbf{F} \mathrm{d}t = \mathbf{h}(\Delta)\mathbf{F}$ | $\mathbf{g}_n = \sum_{k=0}^{K-1} h_k e^{k\Delta_n} \mathbf{f}_n = \mathbf{h}(\Delta_n)\mathbf{f}$ |
| Neural Network | $\mathbf{F}_{l+1}^u = \sigma\Big( \sum_{q=1}^{F_l} \mathbf{h}(\Delta)_l^{u,q} \mathbf{F}_l^q \Big)$ | $\mathbf{f}_{n,l+1}^u = \sigma\Big( \sum_{q=1}^{F_l} \mathbf{h}(\Delta_n)_l^{u,q} \mathbf{f}_{n,l}^q \Big)$ |

**A1** the frequency response of filters in $\mathcal{H}$ are non-amplifying Lipschitz continuous;
**A2** Each filter $\widetilde{h}(\cdot) \in \mathcal{H}$ is a lowpass filter;
**A3** $\widetilde{\sigma}$ from Definition 9 is point-wise normalized Lipschitz continuous,

then there exists a sequence of scales $\epsilon_n \to 0$ as $n \to \infty$ s.t.

$$\lim_{n\to\infty} ||\mathbf{\Psi}_u\big(\mathcal{H}, \Delta_n, \mathbf{\Omega}_n^{\mathcal{X}} \mathbf{F}\big) - \mathbf{\Omega}_n^{\mathcal{X}} \mathbf{\Psi}_u\big(\mathcal{H}, \Delta, \mathbf{F}\big)||_{\mathcal{TM}_n} = 0, \tag{23}$$

with the limit in probability, for each $u = 1, 2, \ldots, F_L$.

*Proof:* See Appendix A. □

*Remark 4:* Denoting the Sheaf Laplacian with $\Delta_n$ is an abuse of notation, because Theorem 1 is a condition both on $\epsilon_n \to 0$ and $n \to \infty$. For this reason, we should employ a notation such as $\Delta_{n,\epsilon_n}$; however, we will keep $\Delta_n$ in the following for the sake of exposition and consistency.

Theorem 1 requires the filters to be lowpass. This can be challenging in a learning context because the filters are learned end-to-end and they may or may not satisfy this hypothesis. Thus, the practical implication of Theorem 1 is that it is possible to train TNNs on sampled manifolds although we do not offer an explicit method to guarantee that this is indeed attained. A first important point to make is that this condition is not spurious, as it is a minimal condition imposed in the proof of Theorem 1 to guarantee convergence. A second important point is that filters can be forced to be lowpass by constraining the filters coefficients during training, if needed. Here we do not advocate the use of these constraints.

### C. Discretization in the Time Domain

The discretization in space introduced in the previous section is still not enough for implementing TNNs in practice. Indeed, learning the continuous time function $\widetilde{h}(t)$ is in general infeasible. For this reason, we discretize $\widetilde{h}(t)$ in the continuous time domain by fixing a sampling interval $T_s > 0$. In this way, we can replace the filter response function with a series of coefficients $h_k = \widetilde{h}(kT_s)$, $k = 0, 1, 2 \ldots$. Fixing $T_s = 1$ and taking $K$ samples over the time horizon, the discrete-time version of the convolution in (6) is given by

$$\mathbf{h}(\Delta_n)\mathbf{F}(x) = \sum_{k=0}^{\infty} h_k e^{k\Delta_n} \mathbf{F}(x), \tag{24}$$

which can be seen as a finite impulse response (FIR) filter with shift operator $e^{\Delta_n}$. We are now in the condition of injecting the space discretization from Section IV in the finite-time architecture in (24), thus finally obtaining an implementable tangent bundle filter that exploits the approximating cellular sheaf $\mathcal{TM}_n$ as

$$\mathbf{g}_n = \mathbf{h}(\Delta_n)\mathbf{f}_n = \sum_{k=0}^{K-1} h_k e^{k\Delta_n} \mathbf{f}_n. \tag{25}$$

The discretized manifold filter of order $K$ can be seen as a generalization of graph convolution to the orthogonal cellular sheaf domain. Thus, we refer $e^{\Delta_n}$ as a sheaf shift operator. At this point, by replacing the filter $\mathbf{h}_l^{pq}(\Delta_n)$ in (22) with (25), we obtain the following architecture:

$$\mathbf{f}_{n,l+1}^u = \sigma\left( \sum_{q=1}^{F_l} \sum_{k=0}^{K-1} h_{k,l}^{u,q} \big(e^{\Delta_n}\big)^k \mathbf{f}_{n,l}^q \right), \; u = 1, ..., F_{l+1}, \tag{26}$$

that we refer to as discretized space-time tangent bundle neural network (DD-TNN). DD-TNNs are a novel principled variant of the recently proposed Sheaf Neural Networks [32], [41], [42], with $e^{\Delta_n}$ as (sheaf) shift operator and order $K$ diffusion. To better enhance this similarity, we rewrite the layer in (26) in matrix form by introducing the matrices $\mathbf{X}_{n,l} = \{\mathbf{f}_{n,l}^u\}_{u=1}^{F_l} \in \mathbb{R}^{n\hat{d} \times F_l}$, and $\mathbf{H}_{l,k} = \{h_{k,l}^{u,q}\}_{q=1,u=1}^{F_l, F_{l+1}} \in \mathbb{R}^{F_l \times F_{l+1}}$, as

$$\mathbf{X}_{n,l+1} = \sigma\left( \sum_{k=0}^{K-1} \big(e^{\Delta_n}\big)^k \mathbf{X}_{n,l} \mathbf{H}_{l,k} \right) \in \mathbb{R}^{n\hat{d} \times F_{l+1}}, \tag{27}$$

where the filter weights $\{\mathbf{H}_{l,k}\}_{l,k}$ are learnable parameters. Finally, we have completed the process of building TNNs from (orthogonal) cellular sheaves and back. The proposed methodology also shows that manifolds and their tangent bundles can be seen as the limits of graphs and (orthogonal) cellular sheaves on top of them. A summary of the proposed continuous framework on tangent bundles and its discretization on orthogonal cellular sheaves is presented in Table II. Please notice that, when $T_s = 1$ and $K = 1$ in (27), the standard Sheaf Neural Network from [41] (up to an additional channel mixing matrix) with the exponential of the sheaf Laplacian as shift operator is recovered.

## V. CONSISTENCY OF TANGENT BUNDLE CONVOLUTIONS

The tangent bundle convolution in Definition 4 provides a definition of a convolution that is compatible with convolutions on manifold scalar fields, convolutions on graphs, and (standard) convolutions for signals in time.

The manifold convolution from [27] is recovered when the bundle is a scalar bundle, i.e. when scalar functions $f : \mathcal{M} \to \mathbb{R}$

over the manifold are considered. In this case, the Connection Laplacian $\Delta$ reduces to the usual Laplace-Beltrami operator [49], here denoted with $\Delta_{\mathcal{M}} : \mathcal{L}_2(\mathcal{M}) \to \mathcal{L}_2(\mathcal{M})$, and the resulting convolution, given a filter $\widetilde{h} : \mathbb{R}^+ \to \mathbb{R}$, is

$$g(x) = \big(\widetilde{h} \star_{\mathcal{M}} f\big)(x) = \int_0^\infty \widetilde{h}(t) e^{-t\Delta_{\mathcal{M}}} f(x) \mathrm{d}t. \tag{28}$$

This expression is both the manifold convolution from [27] and a particular case of (7). The negative sign comes from the convention to define the standard Laplacian as a positive semidefinite operator whereas the Connection Laplacian is defined to be negative semidefinite.

Given a set of $n$ points $\mathcal{X} \subset \mathbb{R}^p$ sampled from the manifold, we further recover a form of graph convolution [8], [45]. In particular, if the manifold $\mathcal{M}$ is discretized as a geometric graph $\mathcal{M}_n$ whose nodes are the sampled points, the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ is discretized as a graph Laplacian $\Delta_{\mathcal{M},n} \in \mathbb{R}^{n \times n}$ whose entries are the weights $w_{i,j}$ of equation (14). If we further discretize $f : \mathcal{M} \to \mathbb{R}$ as a graph signal $\mathbf{f}_n : \mathcal{M}_n \to \mathbb{R}$, the resulting convolution is

$$\mathbf{g}_n = \big(\widetilde{h} \star_{\mathcal{M}_n} \mathbf{f}_n\big) = \int_0^\infty \widetilde{h}(t) e^{-t\Delta_{\mathcal{M},n}} \mathbf{f}_n \mathrm{d}t. \tag{29}$$

This is a particular case of (21) and can be interpreted as an exponential form of a graph convolution. Further discretizing the filter across the index $t$ as we do in Section IV-C. yields the graph convolution

$$\mathbf{g}_n = \sum_{k=0}^{K-1} h_k (e^{-\Delta_{\mathcal{M},n}})^k \mathbf{f}_n. \tag{30}$$

This is a FIR graph filter with $e^{-\Delta_{\mathcal{M},n}}$ used as a shift operator. The expression can be made more familiar if we approximate the exponential by $e^{-\Delta_{\mathcal{M},n}} \approx \mathbf{I}_n - \Delta_{\mathcal{M},n}$.

The standard time convolution is recovered when the manifold is the real line $\mathbb{R}$, the functions $f : \mathbb{R} \to \mathbb{R}$ are scalar functions, and the operator employed in the heat equation in (4) is replaced by the derivative operator $\partial/\partial x$. In particular, due to the fact that the exponential of the derivative operator is a time shift operator, we can write $e^{-t\partial/\partial x} f(x) = f(x - t)$. In this case, the resulting convolution is

$$\begin{aligned} g(x) = \big(\widetilde{h} \star_{\mathbb{R}} f\big)(x) &= \int_0^\infty \widetilde{h}(t) e^{-t\partial/\partial x} f(x) \mathrm{d}t \\ &= \int_0^\infty \widetilde{h}(t) f(x - t)\, \mathrm{d}t, \end{aligned} \tag{31}$$

This is the (standard) time convolution and also a particular case of (7). An additional amenable theoretical feature of our tangent bundle convolution is its consistency with the framework of Algebraic Signal Processing (ASP) [20], [46]. An ASP model is made of four components: (i) A vector space $\mathbb{V}$ where the signals of interest live. (ii) The space End($\mathbb{V}$) of endomorphisms of $\mathbb{V}$ containing the linear maps that can be applied to the signals in $\mathbb{V}$. (iii) An Algebra $\mathbb{A}$ that defines abstract convolutional filters. (iv) A homomorphism $\rho$ that maps filters in $\mathbb{A}$ to endomorphisms that can be applied to signals. In our case, the vector space $\mathbb{V}$ is made of tangent bundle signals, and the algebra $\mathbb{A}$ is the algebra $(\mathcal{L}_1(\mathbb{R}_+), \star_{\mathbb{R}})$ of absolute integrable functions in $\mathbb{R}_+$ with the standard convolution $\star_{\mathbb{R}}$ as the product. The homomorphism $\rho$ maps the filter $\widetilde{h}(t)$ to the tangent bundle filter $\rho(\widetilde{h})$ whose action on a signal $\mathbf{F}$ is

$$\rho(\widetilde{h}) \circ \mathbf{F}(s) = \int_0^\infty \widetilde{h}(t) e^{t\Delta} \mathbf{F}(s)\, dt. \tag{32}$$

This is clearly the definition we obtain in (7) by combining (5) and (6). It is trivial to verify that $\rho(\widetilde{h})$ is a homomorphism.

An alternative definition of tangent bundle convolution would be obtained if we replace the algebra $\mathbb{A}$ with the algebra of polynomials. Thus, filters would be polynomials $\widetilde{h}(t) = \sum_{k=0}^K h_k t^k$ and the tangent bundle filters would be

$$\rho(\widetilde{h}) \circ \mathbf{F}(s) = \sum_{k=0}^K h_k \Delta^k \mathbf{F}(s). \tag{33}$$

In this latter case, discretizing the manifold would give rise to graph filters defined as polynomials of the graph Laplacian. In this paper we prefer to work with (32) rather than (33) because it leads to the connection with convolutions in continuous time stated in (31). This connection can't be made if we adopt (33) as a definition of tangent bundle filter. It is important to remark that if we adopt (33) as a definition a similar convergence theorem holds. We just need to change the definition of the filter's frequency response to the polynomial $\sum_{k=0}^K h_k \lambda^k$ and proceed to adapt assumptions and derivations.

## VI. Numerical Results

In this section, we assess the performance of Tangent Bundle Neural Networks on four tasks: denoising of a tangent vector field on the torus (synthetic data), reconstruction from partial observations of the Earth wind field (real data), forecasting of the Earth wind field (real data), obtained via a recurrent version of the proposed architecture, and binary manifold classification (synthetic data). In this work, we are interested in showing the advantage of including information about the tangent bundle structure for processing tangent bundle signals. For this reason, in the following experiments we always use the vanilla DD-TNN architecture in (27) without any additional modules (e.g. readout MLP layers), and we compare our architectures against vanilla Manifold Neural Networks (MNNs) from [27], convolutional architectures built in a similar way to ours but taking into account only the manifold structure. MNNs are implemented as GNNs with the exponential of the normalized cloud Laplacian [27], [52]. Moreover, we also compare DD-TNNs against Multi-Layer Perceptrons (MLPs) [47] in the denoising and reconstruction tasks, against Recurrent Neural Networks (RNNs) [53] in the forecasting task, and against 3D-CNN in the classification task. Therefore, from a discrete point of view, we present a comparison between a specific (novel and principled) Sheaf Neural Networks class (DD-TNNs, which introduce a relational inductive bias [54] given by the tangent bundle/sheaf structure), a specific Graph Neural Networks class (MNNs, which introduce a relational inductive bias given by the manifold/graph structure), and Multi-Layer Perceptrons/Recurrent Neural Networks (MLPs/RNNs, which introduce no relational inductive biases). It is clear that the employed classes of architectures could be enriched with many additional components

TABLE III
MSE ON THE TORUS DENOISING TASK

| | | $\tau = 10^{-2}$ | $\tau = 10^{-1}$ | $\tau = 3 \cdot 10^{-1}$ |
|---|---|---|---|---|
| $\mathrm{E}\{n\} = 100$ | DD-TNN | $\mathbf{2.02 \cdot 10^{-4}} \pm 1.88 \cdot 10^{-5}$ | $\mathbf{1.78 \cdot 10^{-2}} \pm 1.96 \cdot 10^{-3}$ | $\mathbf{1.35 \cdot 10^{-1}} \pm 1.42 \cdot 10^{-2}$ |
| | MNN | $7.33 \cdot 10^{-4} \pm 4.61 \cdot 10^{-4}$ | $2.45 \cdot 10^{-2} \pm 4.26 \cdot 10^{-3}$ | $2.19 \cdot 10^{-1} \pm 3.56 \cdot 10^{-2}$ |
| | MLP | $2.34 \cdot 10^{-4} \pm 2.88 \cdot 10^{-5}$ | $1.83 \cdot 10^{-2} \pm 2.48 \cdot 10^{-3}$ | $1.52 \cdot 10^{-1} \pm 2.15 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 200$ | DD-TNN | $\mathbf{2.06 \cdot 10^{-4}} \pm 1.46 \cdot 10^{-5}$ | $\mathbf{1.82 \cdot 10^{-2}} \pm 1.18 \cdot 10^{-3}$ | $\mathbf{1.36 \cdot 10^{-1}} \pm 1.05 \cdot 10^{-2}$ |
| | MNN | $7.78 \cdot 10^{-4} \pm 5.76 \cdot 10^{-4}$ | $2.50 \cdot 10^{-2} \pm 3.90 \cdot 10^{-3}$ | $2.11 \cdot 10^{-1} \pm 3.30 \cdot 10^{-2}$ |
| | MLP | $2.28 \cdot 10^{-4} \pm 3.52 \cdot 10^{-5}$ | $1.88 \cdot 10^{-2} \pm 2.88 \cdot 10^{-3}$ | $1.55 \cdot 10^{-1} \pm 2.06 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 300$ | DD-TNN | $\mathbf{2.05 \cdot 10^{-4}} \pm 1.07 \cdot 10^{-5}$ | $\mathbf{1.80 \cdot 10^{-2}} \pm 1.01 \cdot 10^{-3}$ | $\mathbf{1.31 \cdot 10^{-1}} \pm 7.91 \cdot 10^{-3}$ |
| | MNN | $6.64 \cdot 10^{-4} \pm 4.13 \cdot 10^{-4}$ | $2.43 \cdot 10^{-2} \pm 4.01 \cdot 10^{-3}$ | $2.05 \cdot 10^{-1} \pm 3.06 \cdot 10^{-2}$ |
| | MLP | $2.36 \cdot 10^{-4} \pm 2.87 \cdot 10^{-5}$ | $1.85 \cdot 10^{-2} \pm 2.25 \cdot 10^{-3}$ | $1.51 \cdot 10^{-1} \pm 1.87 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 400$ | DD-TNN | $\mathbf{2.00 \cdot 10^{-4}} \pm 9.60 \cdot 10^{-6}$ | $\mathbf{1.80 \cdot 10^{-2}} \pm 8.99 \cdot 10^{-4}$ | $\mathbf{1.35 \cdot 10^{-1}} \pm 8.03 \cdot 10^{-3}$ |
| | MNN | $6.84 \cdot 10^{-4} \pm 6.28 \cdot 10^{-4}$ | $3.45 \cdot 10^{-2} \pm 5.88 \cdot 10^{-2}$ | $2.55 \cdot 10^{-1} \pm 9.50 \cdot 10^{-2}$ |
| | MLP | $2.26 \cdot 10^{-4} \pm 3.27 \cdot 10^{-5}$ | $1.86 \cdot 10^{-2} \pm 2.28 \cdot 10^{-3}$ | $1.58 \cdot 10^{-1} \pm 1.90 \cdot 10^{-2}$ |

(biases, layer normalization, dropout, gating, just to name a few), and it is also clear that a huge number of other architectures could be tailored to the proposed tasks, but testing them is beyond the scope of this paper.[1]

### A. Torus Denoising

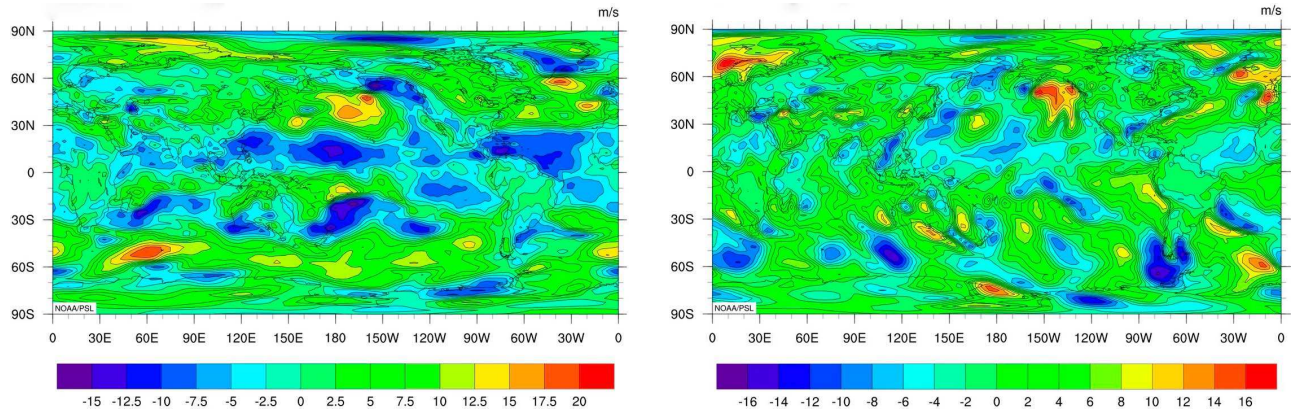
We design a denoising task on a 2-dimensional torus ($\mathcal{M} = \mathcal{T}_2$) and its tangent bundle. It is well known that the 2-torus, the 2-sphere, the real projective plane, together with their connected sums completely classify closed 2-dimensional manifolds, thus it is a good manifold to test our architecture. A parameterization of the 2-dimensional torus is obtained by revolving a circle in three-dimensional space about an axis that is coplanar with the circle: $[x, y, z] = [(b + a\cos\theta)\cos\phi, (b + a\cos\theta)\sin\phi, r\sin\theta]$, where $\phi, \theta \in [0, 2\pi)$, $a$ is the radius of the tube, and $b$ is the distance from the center of the tube to the center of the torus; $b/a$ is called the aspect ratio. In this experiment, we work on a ring torus, thus a torus with aspect ratio greater than one (in particular, we choose $b = 0.3$, $a = 0.1$). We uniformly sample the torus on $n$ points $\mathcal{X} = \{x_1, \ldots, x_n\}$, and we compute the corresponding cellular sheaf $\mathcal{TM}_n$, Sheaf Laplacian $\Delta_n$ and signal sampler $\mathbf{\Omega}_n^{\mathcal{X}}$ as explained in Section IV-B, with $\epsilon_{\text{PCA}} = 0.8$ and $\epsilon_n = 0.5$. We consider the tangent vector field over the torus given by $d\iota\mathbf{F}(x, y, z) = (-\sin\theta, \cos\theta, 0) \in \mathbb{R}^3$. At this point, we add AWGN with variance $\tau^2$ to $d\iota\mathbf{F}$ obtaining a noisy field $\widetilde{d\iota\mathbf{F}}$, then we use $\mathbf{\Omega}_n^{\mathcal{X}}$ to sample it, obtaining $\widetilde{\mathbf{f}}_n \in \mathbb{R}^{2n}$. We test the performance of the TNN architecture by evaluating its ability to denoising $\widetilde{\mathbf{f}}_n$. We exploit a 3 layers architecture with 8 and 4 hidden features, and 1 output feature (the denoised signal), using $K = 2$ in each layer, with Tanh() non-linearities in the hidden layers and a linear activation on the output layer; the architecture hyperparameters have been chosen with hyperparameters sweeps. We train the architecture to minimize the square error $\|\widetilde{\mathbf{f}}_n - \mathbf{f}_n^o\|^2$ between the noisy signal $\widetilde{\mathbf{f}}_n$ and the output of the network $\mathbf{f}_n^o$ via the ADAM optimizer [55] and a patience of 5 epochs, with hyperparameters set to obtain the best results. We compare our architecture with a 3 layers MNN (implemented via a GNN as explained in [27]) with same hyperparameters; to make the comparison fair, $\widetilde{d\iota\mathbf{F}}$ evaluated on $\mathcal{X}$ is given as input to the MNN, organized in a matrix $\widetilde{\mathbf{F}}_n \in \mathbb{R}^{n\times 3}$. We train the MNN to minimize the square error $\|\widetilde{\mathbf{F}}_n - \mathbf{F}_n^o\|_F^2$, where $\|\|_F$ is the Frobenius Norm and $\mathbf{F}_n^o$ is the network output. It is trivial to see that the "two" MSEs used for TNN and MNN are completely equivalent due to the orthogonality of the projection matrices $\mathbf{O}_i$. In Table III, we evaluate TNNs and MNNs for four different expected sample sizes ($\mathrm{E}\{n\} = 100$, $\mathrm{E}\{n\} = 200$, $\mathrm{E}\{n\} = 300$, and $\mathrm{E}\{n\} = 400$), for three different noise standard deviation ($\tau = 10^{-2}$, $\tau = \cdot 10^{-1}$ and $\tau = 3 \cdot 10^{-1}$), showing the MSEs $\frac{1}{n}\|\mathbf{f}_n - \mathbf{f}_n^o\|^2$ and $\frac{1}{n}\|\mathbf{F}_n - \mathbf{F}_n^o\|_F^2$, where $\mathbf{f}_n$ is the sampling via $\mathbf{\Omega}_n^{\mathcal{X}}$ of the clean field and $\mathbf{F}_n$ is the matrix collecting the clean field evaluated on $\mathcal{X}$. 8 sampling realizations and 8 mask realizations per each of them are tested; to make the results consistent, divergent or badly trained runs are discarded if present, and then the results are averaged (on average about 2 runs are discarded per each sampling realization). As the reader can notice from Table I, TNNs always perform better than MNNs and MLPs, due to their "bundle-awareness", i.e. the sheaf structure.

### B. Wind Field Reconstruction

We design a reconstruction task on real-world data. We use daily average measurements (the tangent bundle signal) of Earth surface wind field collected by NCEP/NCAR[2]; in particular, we use the data corresponding to the wind field of the 1st of January 2016, consisting of regularly spaced observations covering the whole Earth surface. The observations are localized in terms of latitude and longitude, thus we convert them in 3-dimensional coordinates by using the canonical spherical approximation for the Earth with nominal radius $R = 6356.8$. The wind field is a 2-dimensional tangent vector field made of a zonal component, following the local parallel of latitude, and a meridional component, following the local meridian of longitude. A visualization of the wind field is shown in Fig. 6 (figures taken from the official data repository). We preprocess the data by scaling the observations to be in the range $[-1, 1]$. We first randomly sample $n$ points to obtain the sampling set $\mathcal{X}$, the cellular sheaf $\mathcal{TM}_n$, and the Sheaf Laplacian $\Delta_n$ again with $\epsilon_{\text{PCA}} = 0.8$ and $\epsilon_n = 0.5$; at this point, we mask $\widetilde{n} < n$ of these points, we collect them in a set $\widetilde{\mathcal{X}}^C \subset \mathcal{X}$, and we aim to infer their corresponding measurements exploiting the remaining available $n - \widetilde{n}$ measurements, collected in the set $\widetilde{\mathcal{X}} \subset \mathcal{X}$. This reconstruction problem can be equivalently seen as a semi-supervised regression problem. To tackle it, we first organize the data corresponding to the point in $\mathcal{X}$ in a

[1] Our implementation of TNNs & datasets available at https://github.com/clabat9/Tangent-Bundle-Neural-Networks

[2] https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.html




(a) Zonal Wind (b) Meridional Wind

Fig. 6. Visualization of Earth wind field on 1st of January 2016 (a) zonal component. (b) meridional component.

TABLE IV
MSE ON THE WIND FIELD RECONSTRUCTION TASK

| | | $\mathrm{E}\{\widetilde{n}\} = 0.5n$ | $\mathrm{E}\{\widetilde{n}\} = 0.3n$ | $\mathrm{E}\{\widetilde{n}\} = 0.1n$ |
|---|---|---|---|---|
| $\mathrm{E}\{n\} = 100$ | DD-TNN | $\mathbf{1.93 \cdot 10^{-2}} \pm 3.64 \cdot 10^{-3}$ | $\mathbf{1.15 \cdot 10^{-2}} \pm 2.75 \cdot 10^{-3}$ | $\mathbf{3.31 \cdot 10^{-3}} \pm 1.62 \cdot 10^{-3}$ |
| | MNN | $4.20 \cdot 10^{-2} \pm 3.05 \cdot 10^{-2}$ | $3.12 \cdot 10^{-2} \pm 1.86 \cdot 10^{-2}$ | $2.82 \cdot 10^{-2} \pm 2.32 \cdot 10^{-2}$ |
| | MLP | $2.00 \cdot 10^{-2} \pm 3.99 \cdot 10^{-3}$ | $1.21 \cdot 10^{-2} \pm 2.50 \cdot 10^{-3}$ | $3.61 \cdot 10^{-3} \pm 1.70 \cdot 10^{-3}$ |
| $\mathrm{E}\{n\} = 200$ | DD-TNN | $\mathbf{1.99 \cdot 10^{-2}} \pm 2.30 \cdot 10^{-3}$ | $\mathbf{1.18 \cdot 10^{-2}} \pm 1.62 \cdot 10^{-3}$ | $\mathbf{3.67 \cdot 10^{-3}} \pm 1.23 \cdot 10^{-3}$ |
| | MNN | $3.19 \cdot 10^{-2} \pm 1.31 \cdot 10^{-2}$ | $2.74 \cdot 10^{-2} \pm 1.55 \cdot 10^{-2}$ | $2.58 \cdot 10^{-2} \pm 1.82 \cdot 10^{-2}$ |
| | MLP | $2.03 \cdot 10^{-2} \pm 2.28 \cdot 10^{-3}$ | $1.20 \cdot 10^{-2} \pm 1.68 \cdot 10^{-3}$ | $3.69 \cdot 10^{-3} \pm 1.17 \cdot 10^{-3}$ |
| $\mathrm{E}\{n\} = 300$ | DD-TNN | $\mathbf{1.88 \cdot 10^{-2}} \pm 1.72 \cdot 10^{-3}$ | $\mathbf{1.13 \cdot 10^{-2}} \pm 1.54 \cdot 10^{-3}$ | $\mathbf{3.96 \cdot 10^{-3}} \pm 1.00 \cdot 10^{-3}$ |
| | MNN | $2.68 \cdot 10^{-2} \pm 7.64 \cdot 10^{-3}$ | $2.41 \cdot 10^{-2} \pm 1.05 \cdot 10^{-2}$ | $2.09 \cdot 10^{-2} \pm 1.76 \cdot 10^{-2}$ |
| | MLP | $1.95 \cdot 10^{-2} \pm 1.74 \cdot 10^{-3}$ | $1.18 \cdot 10^{-2} \pm 1.56 \cdot 10^{-3}$ | $4.00 \cdot 10^{-3} \pm 8.85 \cdot 10^{-4}$ |
| $\mathrm{E}\{n\} = 400$ | DD-TNN | $\mathbf{1.95 \cdot 10^{-2}} \pm 1.66 \cdot 10^{-3}$ | $\mathbf{1.14 \cdot 10^{-2}} \pm 1.38 \cdot 10^{-3}$ | $\mathbf{3.70 \cdot 10^{-3}} \pm 8.55 \cdot 10^{-4}$ |
| | MNN | $2.48 \cdot 10^{-2} \pm 6.55 \cdot 10^{-3}$ | $2.52 \cdot 10^{-2} \pm 1.20 \cdot 10^{-2}$ | $8.16 \cdot 10^{-2} \pm 1.87 \cdot 10^{-1}$ |
| | MLP | $2.01 \cdot 10^{-2} \pm 1.66 \cdot 10^{-3}$ | $1.19 \cdot 10^{-2} \pm 1.24 \cdot 10^{-3}$ | $3.81 \cdot 10^{-3} \pm 8.46 \cdot 10^{-4}$ |

matrix $\mathbf{F}_n \in \mathbb{R}^{n \times 2}$, where the first column collects the zonal components and the second column collects the meridional components. At this point, we build the matrix $\widetilde{\mathbf{F}}_n \in \mathbb{R}^{n \times 2}$, that is a copy of $\mathbf{F}$ except for the rows of $\mathbf{F}$ corresponding to the masked points in $\widetilde{\mathcal{X}}^C$, replaced with the mean of the measurements of the available points in $\widetilde{\mathcal{X}}$. We then vectorize $\widetilde{\mathbf{F}}_n$ to obtain $\widetilde{\mathbf{f}}_n \in \mathbb{R}^{2n}$, the input tangent bundle signal. We now exploit the same DD-TNN architecture from Section VI-A, with the same hyperparameters, to perform the reconstruction task by training it to minimize the reconstruction square error

$$\sum_{i \in \widetilde{\mathcal{X}}} \|\widetilde{\mathbf{f}}_n(i) - \mathbf{f}_n^o(i)\|^2 \tag{34}$$

between the available measurements $\mathbf{f}_n(i)$ and the output of the network corresponding to them $\mathbf{f}_n^o(i)$, $i \in \widetilde{\mathcal{X}}$. Again, we compare our architecture with the same MNN from Section VI-A, to which we give as input the matrix $\widetilde{\mathbf{F}}$ and we train it to minimize $\sum_{i \in \widetilde{\mathcal{X}}} \|\widetilde{\mathbf{F}}_n(i) - \mathbf{F}_n^o(i)\|^2$, where $\mathbf{F}_n^o$ is the network output and $\widetilde{\mathbf{F}}_n(i)$ indicates the $i-$th row of $\widetilde{\mathbf{F}}_n(i)$; being $\widetilde{\mathbf{f}}_n$ the vectorization of $\widetilde{\mathbf{F}}_n$, also in this case it is trivial to check the equivalence of the two MSEs. As evaluation metric, we use the reconstruction MSE on the measurements corresponding to the masked nodes $\frac{1}{n}\sum_{i \in \widetilde{\mathcal{X}}^C} \|\mathbf{f}_n(i) - \mathbf{f}_n^o(i)\|^2$. In Table IV we evaluate TNNs and MNNs for four different expected sample sizes ($\mathrm{E}\{n\} = 100$, $\mathrm{E}\{n\} = 200$, $\mathrm{E}\{n\} = 300$, and $\mathrm{E}\{n\} = 400$), for three different masking probabilities ($\mathrm{E}\{\widetilde{n}\} = 0.5n$, $\mathrm{E}\{\widetilde{n}\} = 0.3n$, and $\mathrm{E}\{\widetilde{n}\} = 0.1n$) per each of them (the probability of a node to being masked). As the reader can notice, TNNs are always able to perform better than MNNs and MLPs, keeping the performance stable with the number of samples and, of course, improving with more observations available.

### C. Wind Field Forecasting With Recurrent TNNs

We design a forecasting task on the same wind field data from Section VI-B. In particular, we use daily observation corresponding to the wind field from the 1st of January 2016 to 7 September 2016 to train the model and we use observations from the 1st of January 2017 to 7 September 2017 to test it. We, again, randomly sample $n$ points to obtain the sampling set $\mathcal{X}$, the cellular sheaf $\mathcal{TM}_n$, and the Sheaf Laplacian $\Delta_n$; at this point, we organize the data corresponding to the sampled point in $\mathcal{X}$ in a sequence $\{\mathbf{F}_{n,t}\}_t$ indexed by time $t$ (daily interval), with each $\mathbf{F}_{n,t} \in \mathbb{R}^{n \times 2}$. As in Section VI-B, we vectorize $\{\mathbf{F}_{n,t}\}_t$ to obtain $\{\mathbf{f}_{n,t}\}_t$, the input tangent bundle signals, with each $\mathbf{f}_{n,t} \in \mathbb{R}^{2n}$. We now introduce a hyperparameter $T_f > 0$ representing the length of the predictive time window of the model, i.e., given in input a subsequence $\{\mathbf{f}_{n,t}\}_{t=T_s}^{t=T_s+T_f}$ starting at time $T_s$ of length $T_f$, the model outputs a sequence $\{\mathbf{f}_{n,t}^o\}_{t=1}^{t=T_f}$ of length $T_f$ aiming at estimating the next $T_f$ element $\{\mathbf{f}_{n,t}\}_{t=T_s+T_f+1}^{t=T_s+2T_f+1}$ of the input sequence.

TABLE V
MSE ON THE WIND FIELD FORECASTING TASK

| | | $T_f = 20$ | $T_f = 50$ | $T_f = 80$ |
|---|---|---|---|---|
| $\mathrm{E}\{n\} = 100$ | DD-TNN | $\mathbf{8.39 \cdot 10^{-2}} \pm 1.62 \cdot 10^{-2}$ | $1.20 \cdot 10^{-1} \pm 7.13 \cdot 10^{-2}$ | $1.36 \cdot 10^{-1} \pm 5.05 \cdot 10^{-2}$ |
| | MNN | $3.76 \cdot 10^{-1} \pm 2.74 \cdot 10^{-1}$ | $6.18 \cdot 10^{-1} \pm 2.09 \cdot 10^{-1}$ | $9.63 \cdot 10^{-1} \pm 1.55 \cdot 10^{-2}$ |
| | RNN | $8.89 \cdot 10^{-2} \pm 3.06 \cdot 10^{-2}$ | $\mathbf{8.69 \cdot 10^{-2}} \pm 3.19 \cdot 10^{-2}$ | $\mathbf{7.24 \cdot 10^{-2}} \pm 2.63 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 200$ | DD-TNN | $\mathbf{8.14 \cdot 10^{-2}} \pm 3.58 \cdot 10^{-2}$ | $\mathbf{7.03 \cdot 10^{-2}} \pm 1.40 \cdot 10^{-2}$ | $1.61 \cdot 10^{-1} \pm 7.38 \cdot 10^{-2}$ |
| | MNN | $3.49 \cdot 10^{-1} \pm 2.03 \cdot 10^{-1}$ | $6.70 \cdot 10^{-1} \pm 2.30 \cdot 10^{-1}$ | $5.03 \cdot 10^{-1} \pm 1.39 \cdot 10^{-1}$ |
| | RNN | $9.78 \cdot 10^{-2} \pm 3.58 \cdot 10^{-2}$ | $7.54 \cdot 10^{-2} \pm 4.92 \cdot 10^{-2}$ | $\mathbf{1.20 \cdot 10^{-1}} \pm 5.99 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 300$ | DD-TNN | $\mathbf{6.43 \cdot 10^{-2}} \pm 1.13 \cdot 10^{-2}$ | $\mathbf{7.03 \cdot 10^{-2}} \pm 3.16 \cdot 10^{-2}$ | $2.49 \cdot 10^{-1} \pm 1.74 \cdot 10^{-1}$ |
| | MNN | $6.69 \cdot 10^{-1} \pm 2.12 \cdot 10^{-1}$ | $6.09 \cdot 10^{-1} \pm 3.66 \cdot 10^{-1}$ | $8.83 \cdot 10^{-1} \pm 8.60 \cdot 10^{-2}$ |
| | RNN | $7.95 \cdot 10^{-2} \pm 3.43 \cdot 10^{-2}$ | $8.68 \cdot 10^{-2} \pm 4.06 \cdot 10^{-2}$ | $\mathbf{1.34 \cdot 10^{-1}} \pm 4.66 \cdot 10^{-2}$ |
| $\mathrm{E}\{n\} = 400$ | DD-TNN | $8.93 \cdot 10^{-2} \pm 2.78 \cdot 10^{-2}$ | $\mathbf{8.47 \cdot 10^{-2}} \pm 1.67 \cdot 10^{-2}$ | $1.34 \cdot 10^{-1} \pm 4.35 \cdot 10^{-2}$ |
| | MNN | $4.06 \cdot 10^{-1} \pm 2.47 \cdot 10^{-1}$ | $7.50 \cdot 10^{-1} \pm 2.86 \cdot 10^{-1}$ | $2.35 \cdot 10^{-1} \pm 9.49 \cdot 10^{-2}$ |
| | RNN | $\mathbf{6.29 \cdot 10^{-2}} \pm 2.66 \cdot 10^{-2}$ | $1.25 \cdot 10^{-1} \pm 3.89 \cdot 10^{-2}$ | $\mathbf{5.19 \cdot 10^{-2}} \pm 3.38 \cdot 10^{-2}$ |

To do so, we introduce a recurrent version of the proposed DD-TNNs, which, to the best of our knowledge, is also the first recurrent architecture working on cellular sheaves. The building block of the proposed recurrent architecture is a layer made of three components: a tangent bundle filter processing the current sequence element $\mathbf{f}_{n,t}$, a tangent bundle filter processing the current hidden state $\mathbf{z}_{t-1}$, i.e., the output of the layer computed on the previous sequence element, and a pointwise non-linearity. Formally, the layer reads as:

$$\mathbf{z}_t = \sigma\left( \sum_{k=0}^{K-1} h_k \left(e^{\Delta_n}\right)^k \mathbf{f}_{n,t} + \sum_{k=0}^{K-1} w_k \left(e^{\Delta_n}\right)^k \mathbf{z}_{t-1} \right), \quad (35)$$

with $t = T_s, ..., T_s + T_f$, and $\mathbf{z}_0 = \mathbf{0}$. To obtain the required estimates, we can set $\{\mathbf{f}^o_{n,t}\}_{t=1}^{t=T_f} = \{\mathbf{z}_t\}_{t=1}^{t=T_f}$. This architecture can be used also in a Multi-Layer fashion: in this case, at layer $l$ and at time $t$, the first filter takes $\mathbf{z}_{l-1,t}$ (the current time $t$ hidden state of the previous layer $l-1$) as input, and the second filter takes $\mathbf{z}_{l,t-1}$ (the previous time $t-1$ hidden state of the current layer $l$) as input. Therefore, the resulting $L-$layers architecture is:

$$\mathbf{z}_{l,t} = \sigma\left( \sum_{k=0}^{K-1} h_{k,l} \left(e^{\Delta_n}\right)^k \mathbf{z}_{l-1,t} + \sum_{k=0}^{K-1} w_{k,l} \left(e^{\Delta_n}\right)^k \mathbf{z}_{l,t-1} \right), \quad (36)$$

with $l = 1, ..., L$, $t = T_s, ..., T_s + T_f$, and $\mathbf{z}_{0,t} = \mathbf{f}_{n,t}$. In this case, to obtain the required estimates, we can set $\{\mathbf{f}^o_{n,t}\}_{t=1}^{t=T_f} = \{\mathbf{z}_{L,t}\}_{t=1}^{t=T_f}$. For the wind field forecasting task, the training set is made of all the possible $m = 250 - 2T_f$ subsequences of length $2T_f$ of the 2016 data, we use a 3-layers Recurrent DD-TNN with $K = 2$ and Tanh non-linearities, and we train it to minimize the square error $\sum_{t=1}^{m} \sum_{\tilde{t}=t}^{t+T_f} \|\mathbf{f}_{n,\tilde{t}} - \mathbf{f}^o_{n,\tilde{t}-t+1}\|_2^2$. To have a fair comparison, we set up the corresponding recurrent version of MNNs (RMNNs, a recurrent graph neural network) with the same structure, same hyperparameters, same loss but with inputs $\{\mathbf{F}_{n,t}\}_t$. As evaluation metric, we compute the MSE on the 2017 data after training. In Table V we evaluate RTNNs and RMNNs for four different expected sample sizes ($\mathrm{E}\{n\} = 100, \mathrm{E}\{n\} = 200, \mathrm{E}\{n\} = 300$, and $\mathrm{E}\{n\} = 400$), and for three different time window lengths ($T_f = 20$, $T_f = 50$, and $T_f = 80$) per each of them. Also in this case, the bundle "awareness" of (recurrent) TNNs allows to reach significantly better results in all the tested scenarios w.r.t. (recurrent) MNNs, outperforming RNNs too in most of the cases except for the long-term predictions cases ($T_f = 80$), probably due to the absence of gating or memory mechanisms, useful also to improve training. Moreover, in all the experiments, TNNs have fewer parameters than MNNs, due to the different organization of the data in the input layer.

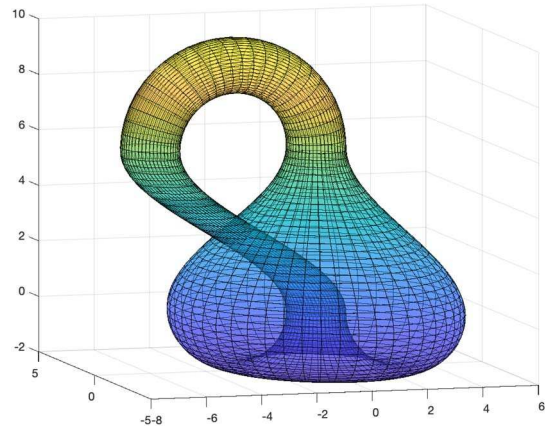



Fig. 7. An immersed Klein bottle.

### D. Manifold Classification

We design an ill-conditioned binary manifold classification task. The goal is to assess whether TNNs are able to learn to distinguish between a torus and an immersed Klein bottle, given uninformative features, i.e. a constant vector field. The Klein bottle is a 2-dimensional non-orientable manifold (i.e. with no "inside" or "outside"). The Klein bottle has to be properly embedded in $\mathbb{R}^4$ since it must pass through itself without the presence of a hole. However, it can be immersed in $\mathbb{R}^3$, as depicted in Fig. 7. Working on the immersion makes the classification task ill-conditioned, because an immersed Klein Bottle is not even a proper manifold. The choice of giving as input uninformative features is made in order to evaluate if the network is able to solely leverage the geometric information contained in the approximated Connection Laplacian (the Sheaf Laplacian) to infer the manifold, resembling [56], [57]. We opted to employ the immersed Klein bottle as one of the target manifolds to heuristically evaluate how our method tackles ill-conditioned tasks that do not match the theoretical justification of our architecture. We create datasets of 2000 data points, where each datapoint is computed with the following steps:

TABLE VI
ACCURACY ON THE MANIFOLD CLASSIFICATION TASK

| | Id() | Tanh() |
|---|---|---|
| DD-TNN | $76.2\% \pm 4.8$ | $87.5\% \pm 0.9$ |
| MNN | $56.3\% \pm 5.3$ | $97.7\% \pm 0.3$ |
| 3D-CNN | $\mathbf{86.3\% \pm 2.3}$ | $\mathbf{98.5\% \pm 0.6}$ |

i) choose if sampling the torus or the Klein bottle tossing a fair coin; ii) uniformly sample the manifold on $\mathrm{E}\{n\} = 100$ points and compute the corresponding cellular sheaf $\mathcal{TM}_n$, and Sheaf Laplacian $\Delta_n$ (both the manifolds are normalized to be contained in a cube of unitary side length, such that the scale is not a discriminating feature); iii) associate to each sampled point the projection via the $\mathbf{O}_i$ s of the constant vector field given by $d\iota\mathbf{F}(x, y, z) = (1, 1, 1) \in \mathbb{R}^3$. We exploit the same DD-TNN architecture from Section VI-A, with the same hyperparameters, plus a final 2-layers MLP classifier with a ReLU() non-linearity after the first layer and a Softmax() non-linearity after the second layer. We train the architecture to minimize the usual binary cross-entropy. We compare our architecture with the same MNN from Section VI-A, where obviously the Sheaf Laplacian is replaced by the graph Laplacian. Unlike the experiments of the previous section, we found that in this case the employed non-linearity impacts the classification accuracy; in particular, we observed changes when the "non-linearity" is the identity function Id() (thus when a cascade of discretized tangent bundle filters is used) or the Tanh(). For this reason we evaluate TNNs and MNNs using both Id() and Tanh(). Moreover, we also report the results of a simple 3D-CNN. To feed the data in the 3D-CNN, we follow the approach from [58], i.e. we first convert each data point to the volumetric representation as an occupancy grid with resolution $6 \times 6 \times 6$. The choice of the hyperparameters is made to keep the number of learnable parameters similar to DD-TNN and MNN. The results are averaged over 8 realizations of the datasets, and per each of them the training dataset is obtained by sampling $80\%$ of the datapoints, and the test set is obtained with the remaining $20\%$. As the reader can notice in Table VI, the 3D-CNN performs better than both MNN and DD-TNN. This is something that we could expect: as described above, this task is ill-conditioned for DD-TNN and MNN, and it is trivially easier to solve for an architecture designed for point clouds, the 3D-CNN w.r.t. architectures based on manifold diffusion operators. However, the performance of DD-TNN is still competitive even if the setting is disadvantaged. Moreover, TNNs significantly perform better than MNNs when Id() is employed, i.e. tangent bundle filters significantly perform better than manifold filters, while MNNs perform better than TNNs when an actual non-linearity, the Tanh(), is introduced.

## VII. CONCLUSION

In this work, we introduced Tangent Bundle Filters and Tangent Bundle Neural Networks (TNNs), novel continuous architectures operating on tangent bundle signals, i.e. manifold vector fields. We made TNNs implementable by discretization in space and time domains, showing that their discrete counterpart is a principled variant of Sheaf Neural Networks. We proved that discretized TNNs asymptotically converge to their continuous counterparts, and we assessed the performance of TNNs on both synthetic and real data. This work gives a multifaceted contribution: on the methodological side, it is the first work to introduce a signal processing framework for signals defined on tangent bundles of Riemann manifolds via the Connection Laplacian; on the theoretical side, the presented discretization procedure and convergence result explicitly link the manifold domain with cellular sheaves, formalizing intuitions presented in works like [42]. In future work, we will investigate more general classes of cellular sheaves that approximate unions of manifolds (perhaps representing multiple classes) or, more generally, stratified spaces [59], [60]. We believe our perspective on tangent bundle neural networks could shed further light on challenging problems in graph neural networks such as heterophily [32], over-squashing [61], or transferability [28]. Finally, we plan to tackle more sophisticated tasks with our proposed architectures.

## APPENDIX

### A. Proof of Theorem 1

*Proof:* We define an inner product for sheaf signals $\mathbf{f}$ and $\mathbf{u}$ on a general cellular sheaf $\mathcal{TM}_n$ as

$$\langle \mathbf{f}, \mathbf{u} \rangle_{\mathcal{TM}_n} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{f}_n(x_i) \cdot \mathbf{u}_n(x_i), \tag{37}$$

and the induced norm $||\mathbf{f}||^2_{\mathcal{TM}_n} = \langle \mathbf{f}, \mathbf{f} \rangle_{\mathcal{TM}_n}$. Assuming that the points in $\mathcal{X}$ are sampled i.i.d. from the uniform probability measure $\mu$ given by the metric on $\mathcal{M}$ and that $\mathcal{TM}_n$ is built as in Section V, the inner product in (37) is equivalent to the following inner product for tangent bundle signals $\mathbf{F}$ and $\mathbf{U}$:

$$\begin{aligned} \langle \mathbf{F}, \mathbf{U} \rangle_{\mathcal{TM}_n} &= \int_{\mathcal{M}} \langle \mathbf{F}(x), \mathbf{U}(x) \rangle_{\mathcal{T}_x\mathcal{M}} \mathrm{d}\mu_n(x) \\ &= \frac{1}{n} \sum_{i=1}^{n} \langle \mathbf{F}(x_i), \mathbf{U}(x_i) \rangle_{\mathcal{T}_{x_i}\mathcal{M}}, \end{aligned} \tag{38}$$

and the induced norm $||\mathbf{F}||^2_{\mathcal{TM}_n} = \langle \mathbf{F}, \mathbf{F} \rangle_{\mathcal{TM}_n}$, where $\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ is the empirical measure corresponding to $\mu$. Indeed, from (1) and due to the orthogonality of the transformations $\mathbf{O}_i$ in Section V, (38) can be rewritten as

$$\begin{aligned} \langle \mathbf{F}, \mathbf{U} \rangle_{\mathcal{TM}_n} &= \frac{1}{n} \sum_{i=1}^{n} d\iota\mathbf{F}(x_i) \cdot d\iota\mathbf{U}(x_i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{O}_i{}^T d\iota\mathbf{F}(x_i) \cdot \mathbf{O}_i{}^T d\iota\mathbf{U}(x_i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{f}_n(x_i) \cdot \mathbf{u}_n(x_i) = \langle \mathbf{f}_n, \mathbf{u}_n \rangle_{\mathcal{TM}_n} \end{aligned} \tag{39}$$

where $\mathbf{f}_n = \mathbf{\Omega}^{\mathcal{X}}_n \mathbf{F}$ and $\mathbf{u}_n = \mathbf{\Omega}^{\mathcal{X}}_n \mathbf{U}$, respectively. We denote with $\Gamma(\mathcal{TM}_n)$ the space of tangent bundle signals w.r.t. the

empirical measure $\mu_n$ (or, equivalently, the space of sheaf signals w.r.t the norm induced by (37)). In the following, we will denote the norm $||\cdot||_{\mathcal{TM}_n}$ with $||\cdot||$ when there is no risk of confusion. In [44], the spectral convergence of the constructed Sheaf Laplacian in (18) to the Connection Laplacian of the underlying manifold has been proved, and we exploit that result for proving the following proposition.

*Proposition 3 (Consequence of Theorem 6.3 [44]):* Let $\mathcal{X} = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ be a set of $n$ i.i.d. sampled points from measure $\mu$ over $\mathcal{M} \subset \mathbb{R}^p$. Let $\mathcal{TM}_n$ be a cellular sheaf built from $\mathcal{X}$ as explained in Section V, with $\hat{d} = d$ and $0 < \epsilon_n \leq \min\{\kappa^{-1}, \iota\}$. Let $\Delta_n$ be the Sheaf Laplacian of $\mathcal{TM}_n$ and $\Delta$ be the Connection Laplacian operator of $\mathcal{M}$. Let $\lambda_i^n$ be the $i$-th eigenvalue of $\Delta_n$ and ${\boldsymbol{\phi}_i}^n$ the corresponding eigenvector. Let $\lambda_i$ be the $i$-th eigenvalue of $\Delta$ and $\boldsymbol{\phi}_i$ the corresponding eigenvector field of $\Delta$, respectively. Then there exists a sequence of scales $\epsilon_n \to 0$ as $n \to \infty$ such that:

$$\lim_{n\to\infty} \lambda_i^n = \lambda_i, \quad \lim_{n\to\infty} \|{\boldsymbol{\phi}_i}^n - \boldsymbol{\Omega}_n^{\mathcal{X}} \boldsymbol{\phi}_i\|_{\mathcal{TM}_n} = 0, \tag{40}$$

where the limits are taken in probability.

*Proof:* This proposition is a consequence of Theorem 6.3 in [25]. Indeed, we rely on the operator introduced in Definition 6.1 of [44] with $\alpha = 1$ ($h_n$ is our $\epsilon_n$), here denoted as $\Xi : \Gamma(\mathcal{TM}) \to \Gamma(\mathcal{TM})$, and on the operator $\widetilde{\Xi} = \epsilon_n^{-1}(\Xi - \mathrm{id})$, where id is the identity mapping. It is straightforward to check:

$$\widetilde{\Xi}\mathbf{F}(x_j) = d\iota^{-1}\mathbf{O}_j\big(\Delta_n \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}\big)(x_j), \tag{41}$$

for $j = 1, \dots, n$. We now show that the eigenvectors sampled on $\mathcal{X}$ and eigenvalues of $\widetilde{\Xi}$ correspond to the eigenvectors and eigenvalues of $\Delta_n$. Let us denote the the $i$-th eigenvector and eigenvalue of $\widetilde{\Xi}$ with $\widetilde{\boldsymbol{\phi}}_i^n$ and $-\widetilde{\lambda}_i^n$, respectively. We have:

$$\widetilde{\Xi}\widetilde{\boldsymbol{\phi}}_i^n(x_j) = -\widetilde{\lambda}_i^n \widetilde{\boldsymbol{\phi}}_i^n(x_j) = d\iota^{-1}\mathbf{O}_j\big(\Delta_n \boldsymbol{\Omega}_n^{\mathcal{X}} \widetilde{\boldsymbol{\phi}}_i^n\big)(x_j) \tag{42}$$

If we apply the mapping $i$ to the last two equalities of (42) and we exploit the orthoghonality of $\mathbf{O}_j$, we obtain:

$$\big(\Delta_n \boldsymbol{\Omega}_n^{\mathcal{X}} \widetilde{\boldsymbol{\phi}}_i^n\big)(x_j) = -\widetilde{\lambda}_i^n {\mathbf{O}_j}^T d\iota \widetilde{\boldsymbol{\phi}}_i^n = -\widetilde{\lambda}_i^n \boldsymbol{\Omega}_n^{\mathcal{X}} \widetilde{\boldsymbol{\phi}}_i^n(x_j) \tag{43}$$

where the second equality applies the definition of $\boldsymbol{\Omega}_n^{\mathcal{X}}$ in (19). Therefore, we have:

$$\lambda_i^n = \widetilde{\lambda}_i^n, \quad {\boldsymbol{\phi}_i}^n(x_j) = \boldsymbol{\Omega}_n^{\mathcal{X}} \widetilde{\boldsymbol{\phi}}_i^n(x_j), \tag{44}$$

$j = 1, \dots, n$. At this point, we can recall Theorem 6.3 in [44], that, in the setting of our Theorem 1, states that there exists a sequence of scales $\epsilon_n \to 0$ as $n \to \infty$ such that:

$$\lim_{n\to\infty} \widetilde{\lambda}_i^n = \lambda_i, \quad \lim_{n\to\infty} \|\widetilde{\boldsymbol{\phi}}_i^n - \boldsymbol{\phi}_i\|_{\mathcal{TM}} = 0, \tag{45}$$

with the limit taken in probability, $j = 1, \dots, n$. Injecting the empirical measure in (45) and exploiting the results in (39) and (44), we obtain:

$$\|\widetilde{\boldsymbol{\phi}}_i^n - \boldsymbol{\phi}_i\|_{\mathcal{TM}_n} = \|{\boldsymbol{\phi}_i}^n - \boldsymbol{\Omega}_n^{\mathcal{X}} \boldsymbol{\phi}_i\|_{\mathcal{TM}_n} \tag{46}$$

The results in (45) and (46) and the a.s. convergence of the empirical measure $\mu_n$ to the measure $\mu$ conclude the proof.

For the sake of clarity, in the following we will drop the dependence on the NNs output index $u$; from the definitions of TNNs in (12) and D-TNNS in (22), we can thus write:

$$\|\boldsymbol{\Psi}\big(\mathcal{H}, \Delta_n, \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}\big) - \boldsymbol{\Omega}_n^{\mathcal{X}}\boldsymbol{\Psi}\big(\mathcal{H}, \Delta, \mathbf{F}\big)\| = \Big\|\mathbf{x}_{n,L} - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_L\Big\|.$$

Further explicating the layers definitions, at layer $l$ we have:

$$\begin{aligned}&\Big\|\mathbf{x}_{n,l} - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_l\Big\| \\ &= \left\|\sigma\left(\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q\right) - \boldsymbol{\Omega}_n^{\mathcal{X}}\sigma\left(\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\right)\right\|\end{aligned} \tag{47}$$

with $\mathbf{x}_{n,0}^q = \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}^q$ for $q = 1, \dots, F_0$. Exploiting the normalized point-wise Lipschitz continuity of the non-linearities (**A3**) and the linearity of the sampling operator $\boldsymbol{\Omega}_n^{\mathcal{X}}$, we have:

$$\begin{aligned}\|\mathbf{x}_{n,l} - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_l\| &\leq \left\|\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\sum_{q=1}^{F_{l-1}} \mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\right\| \\ &\leq \sum_{q=1}^{F_{l-1}} \Big\|\mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\Big\|\end{aligned} \tag{48}$$

The difference term in the last LHS of (48) can be further decomposed for every $q = 1, \dots, F_{l-1}$ as

$$\begin{aligned}&\|\mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\| \\ &\quad\leq \|\mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q - \mathbf{h}_l^q(\Delta_n)\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_{l-1}^q \\ &\qquad + \mathbf{h}_l^q(\Delta_n)\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_{l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\| \\ &\quad\leq \Big\|\mathbf{h}_l^q(\Delta_n)\mathbf{x}_{n,l-1}^q - \mathbf{h}_l^q(\Delta_n)\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_{l-1}^q\Big\| \\ &\qquad + \Big\|\mathbf{h}_l^q(\Delta_n)\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_{l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{h}_l^q(\Delta)\mathbf{F}_{l-1}^q\Big\|\end{aligned} \tag{49}$$

The first term of the last inequality in (49) can be bounded as $\|\mathbf{x}_{n,l-1}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_{l-1}^q\|$ with the initial condition $\|\mathbf{x}_{n,0}^q - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}_0^q\| = 0$ for $q = 1, \dots, F_0$. Denoting the second term with $D_{l-1}^n$, and iterating the bounds derived above through layers and features, we obtain:

$$\|\boldsymbol{\Psi}(\mathcal{H}, \Delta_n, \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}) - \boldsymbol{\Omega}_n^{\mathcal{X}}\boldsymbol{\Psi}(\mathcal{H}, \Delta, \mathbf{F})\| \leq \sum_{l=0}^{L}\prod_{l'=l}^{L} F_{l'} D_l^n.$$

Therefore, we can focus on each difference term $D_l^n$ and omit the feature and layer indices to simplify the notations. We can write the convolution operation in the spectral domain as

$$\begin{aligned}&\|\mathbf{h}(\Delta_n)\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F} - \boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{h}(\Delta)\mathbf{F}\| \\ &\quad= \left\|\sum_{i=1}^{n} \hat{h}(\lambda_i^n)\langle\boldsymbol{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n \right. \\ &\qquad\left. - \sum_{i=1}^{\infty} \hat{h}(\lambda_i)\langle\mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}}\boldsymbol{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\right\|\end{aligned} \tag{50}$$

By adding and subtracting $\sum_{i=1}^{n}\hat{h}(\lambda_i)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n$, by coupling the terms with the same index and using the triangle inequality, we can then write

$$\begin{aligned}&\left\|\sum_{i=1}^{n}\hat{h}(\lambda_i^n)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n-\sum_{i=1}^{\infty}\hat{h}(\lambda_i)\langle\mathbf{F},\boldsymbol{\phi}_i\rangle_{\mathcal{TM}}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\right\|\\&\leq\left\|\sum_{i=1}^{n}\left(\hat{h}(\lambda_i^n)-\hat{h}(\lambda_i)\right)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right\|[T1]\\&+\left\|\sum_{i=1}^{n}\hat{h}(\lambda_i)\left(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n-\langle\mathbf{F},\boldsymbol{\phi}_i\rangle_{\mathcal{TM}}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\right)\right\|[T2]\\&+\left\|\sum_{p=n+1}^{\infty}\hat{h}(\lambda_p)\langle\mathbf{F},\boldsymbol{\phi}_p\rangle_{\mathcal{TM}}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_p\right\|[T3]\end{aligned}\tag{51}$$

We now proceed to prove that $[T1]$ converges to zero in probability as $n$ increases. Fixed a $M_{[T1]}\in\mathbb{N}$, we can always rewrite [T1] as

$$\begin{aligned}[T1]=&\left\|\sum_{i=1}^{\min\{n,M_{[T1]}\}}\left(\hat{h}(\lambda_i^n)-\hat{h}(\lambda_i)\right)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\left.+\sum_{i=M_{[T1]}+1}^{n}\left(\hat{h}(\lambda_i^n)-\hat{h}(\lambda_i)\right)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right\|\end{aligned}\tag{52}$$

Please notice that, when $n<M_{[T1]}$, the last sum is an empty sum. By using the triangle inequality, the orthonormality of the $\boldsymbol{\phi}_i^n$, the Cauchy-Schwartz inequality $|\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}|\leq\|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\|$, and the finiteness of $\|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\|$, we can further bound the RHS of (52), obtaining

$$\begin{aligned}[T1]\leq C_{[T1]}&\sum_{i=1}^{\min\{n,M_{[T1]}\}}|\hat{h}(\lambda_i^n)-\hat{h}(\lambda_i)|\\&+C_{[T1]}\sum_{i=M_{[T1]}+1}^{n}|\hat{h}(\lambda_i^n)-\hat{h}(\lambda_i)|,\end{aligned}\tag{53}$$

for some constant $C_{[T1]}>0$. At this point, by using the fact that $|a-b|\leq|b|$ and the Lipschitz continuity of $\hat{h}(\cdot)$ (**A1**), we can further bound the RHS of (53) as

$$[T1]\leq\underbrace{C_{[T1]}\sum_{i=1}^{\min\{n,M_{[T1]}\}}|\lambda_i^n-\lambda_i|}_{[T1.1]}+\underbrace{C_{[T1]}\sum_{i=M_{[T1]}+1}^{\infty}|\hat{h}(\lambda_i)|}_{[T1.2]}\tag{54}$$

It is clear that we can make $[T1.2]$ in (54) arbitrarily small by increasing $M_{[T1]}$ since it is the reminder of a convergent series with positive summands (**A2**). Therefore, for all $\gamma_{[T1]}>0$, we can always choose an $M_{[T1]}$ such that $[T1.2]$ is smaller than $\gamma_{[T1]}/2C$. Fixed $M_{[T1]}$, we can further bound $[T1.1]$ using the spectral convergence result in (40). In particular, using the definition of limit in probability, letting $0<\gamma_i\leq\gamma_{[T1]}/2CM$, for all $\delta_i>0$, there exist $N_i$ such that for all $n\geq N_i$, it holds

$$\mathbb{P}(|\lambda_i^n-\lambda_i|\leq\gamma_i)\geq1-\delta_i.\tag{55}$$

Therefore, for all $\gamma_{[T1]}>0$ and for all $n\geq\max_i N_i$, it holds

$$[T1.1]\leq C_{[T1]}\sum_{i=1}^{\min\{n,M_{[T1]}\}}\gamma_i\leq\gamma_{[T1]}/2\tag{56}$$

with probability at least $\prod_{i=1}^{\min\{n,M_{[T1]}\}}(1-\delta_i):=1-\delta_{[T1]}$. This allows us to state that for all $\gamma_{[T1]}>0$, for all $\delta_{[T1]}>0$, there exist an $N_{[T1]}$ such that, for all $n>N_{[T1]}$, we have

$$\mathbb{P}([T1]\leq\gamma_{[T1]})\geq1-\delta_{[T1]},\tag{57}$$

i.e. $[T1]$ converges in probability to zero. We now proceed to show that $[T2]$ in (51) converges to zero in probability as $n$ increases. By adding and subtracting $\sum_{i=1}^{n}\hat{h}(\lambda_i)\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i$, and by using the triangle inequality, we can write

$$\begin{aligned}&\left\|\sum_{i=1}^{n}\hat{h}(\lambda_i)(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n-\langle\mathbf{F},\boldsymbol{\phi}_i\rangle_{\mathcal{TM}}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i)\right\|\\&\leq\left\|\sum_{i=1}^{n}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\qquad\left.-\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right\|[T2.1]\\&+\left\|\sum_{i=1}^{n}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\right.\\&\qquad\left.-\langle\mathbf{F},\boldsymbol{\phi}_i\rangle_{\mathcal{TM}}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right\|[T2.2]\end{aligned}\tag{58}$$

We can use now the same approach of $[T1]$. In particular, fixed a $M_{[T2.1]}\in\mathbb{N}$, we can always rewrite $[T2.1]$ and then bound it using the triangle inequality as

$$\begin{aligned}[T2.1]=&\left\|\sum_{i=1}^{\min\{n,M_{[T2.1]}\}}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\qquad\left.-\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right.\\&\left.+\sum_{i=M_{[T2.1]}+1}^{n}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\qquad\left.-\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right\|\\&\leq\left\|\sum_{i=1}^{\min\{n,M_{[T2.1]}\}}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\qquad\left.-\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right\|\\&+\left\|\sum_{i=M_{[T2.1]}+1}^{n}\hat{h}(\lambda_i)\Big(\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\boldsymbol{\phi}_i^n\right.\\&\qquad\left.-\langle\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F},\boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\Big)\right\|\end{aligned}\tag{59}$$

We can now further bound the RHS of (59) by using the triangle inequality, the Cauchy-Schwarz inequality

$|\langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n}| \leq \|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\|$, the non-amplifying frequency response (**A1**, for the first term), the finiteness of $\|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\|$, and the finiteness of $\|\boldsymbol{\phi}_i^n - \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\|$ (for the second term) as

$$[T2.1] \leq \underbrace{C_{[T2.1]} \sum_{i=1}^{\min\{n, M_{[T2.1]}\}} \|\boldsymbol{\phi}_i^n - \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\|}_{[T2.1.1]} + \underbrace{C_{[T2.1]} \sum_{i=M_{[T2.1]}+1}^{\infty} |\hat{h}(\lambda_i)|}_{[T2.1.2]}, \tag{60}$$

for some constant $C_{[T2.1]} > 0$. Leveraging the same arguments we used for $[T1.1]$ and $[T1.2]$ in (54) to bound $[T2.1.1]$ and $[T2.1.2]$ in (60), respectively, but using the convergence of the eigenvectors and not of the eigenvalues from (40), we can state that for all $\gamma_{[T2.1]} > 0$, for all $\delta_{[T2.1]} > 0$, there exist an $N_{[T2.1]}$ such that, for all $n > N_{[T2.1]}$, we have

$$\mathbb{P}([T2.1] \leq \gamma_{[T2.1]}) \geq 1 - \delta_{[T2.1]}, \tag{61}$$

i.e. $[T2.1]$ converges in probability to zero. Following the same procedure we used to obtain the bound in (59) for $[T2.1]$, we can obtain the following bound for $[T2.2]$:

$$\begin{aligned}[T2.2] \leq & \left\| \sum_{i=1}^{\min\{n, M_{[T2.2]}\}} \hat{h}(\lambda_i) \Big( \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i \right. \\ & \left. - \langle \mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}} \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i \Big) \right\| \\ & + \left\| \sum_{i=M_{[T2.2]}+1}^{n} \hat{h}(\lambda_i) \Big( \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i \right. \\ & \left. - \langle \mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}} \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i \Big) \right\| \end{aligned} \tag{62}$$

We further bound the RHS of (62) by using the triangle and Cauchy-Schwarz inequalities, the non-amplifying frequency response (for the first term), the finiteness of $\|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\|$ and $\|\mathbf{F}\|$, and the finiteness of $\|\boldsymbol{\phi}_i^n - \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\|$ (for the second term), as

$$[T2.2] \leq \underbrace{C_{[T2.2]} \sum_{i=1}^{\min\{n, M_{[T2.2]}\}} |\langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} - \langle \mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}}|}_{[T2.2.1]} + \underbrace{C_{[T2.2]} \sum_{i=M_{[T2.2]}+1}^{\infty} |\hat{h}(\lambda_i)|}_{[T2.2.2]}, \tag{63}$$

for some constant $C_{[T2.2]} > 0$. It is trivial, from the weak law of large numbers and from (38)-(39), that

$$\lim_{n\to\infty} \left| \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\rangle_{\mathcal{TM}_n} - \langle \mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}} \right| = 0, \tag{64}$$

with the limit taken in probability. By direct substitution and using the distributive law of the dot product, we can write

$$\begin{aligned} & \left| \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\rangle_{\mathcal{TM}_n} - \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} \right| \\ & = \left| \frac{1}{n} \sum_{i=1}^{n} \Big( \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}(x_i) \cdot \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i(x_i) - \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}(x_i) \cdot \boldsymbol{\phi}_i^n(x_i) \Big) \right| \\ & = \left| \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i - \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} \right| \leq \|\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\| \|\mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i - \boldsymbol{\phi}_i^n\|, \end{aligned} \tag{65}$$

where the last inequality is obtained using the Cauchy-Schwartz inequality. Therefore, using again the spectral convergence of eigenvectors from (40), we can write

$$\lim_{n\to\infty} \left| \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \mathbf{\Omega}_n^{\mathcal{X}}\boldsymbol{\phi}_i\rangle_{\mathcal{TM}_n} - \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} \right| = 0, \tag{66}$$

where the limit is taken in probability. As a direct consequence of the (64) and (66), we can directly state that

$$\lim_{n\to\infty} \left| \langle \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \boldsymbol{\phi}_i^n\rangle_{\mathcal{TM}_n} - \langle \mathbf{F}, \boldsymbol{\phi}_i\rangle_{\mathcal{TM}} \right| = 0, \tag{67}$$

again with the limit in probability. At this point, leveraging the same arguments we used for $[T1.1]$ and $[T1.2]$ in (54) (and for $[T2.1.1]$ and $[T2.1.2]$ in (60)) to bound $[T2.2.1]$ and $[T2.2.2]$ in (63), respectively, but using the convergence of the inner products in (67), we can state that for all $\gamma_{[T2.2]} > 0$, for all $\delta_{[T2.2]} > 0$, there exist an $N_{[T2.2]}$ such that, for all $n > N_{[T2.2]}$:

$$\mathbb{P}([T2.2] \leq \gamma_{[T2.2]}) \geq 1 - \delta_{[T2.2]}, \tag{68}$$

i.e. $[T2.2]$ converges in probability to zero. As a consequence, we can state that for all $\gamma_{[T2]} > 0$, for all $\delta_{[T2]} > 0$, there exist an $N_{[T2]}$ such that, for all $n > N_{[T2]}$, we have

$$\mathbb{P}([T2] \leq \gamma_{[T2]}) \geq 1 - \delta_{[T2]}, \tag{69}$$

i.e. $[T2]$ converges in probability to zero. We are now missing only the convergence in probability of $[T3]$ from (51). However, $[T3]$ is again the reminder of a convergent series with positive summands (**A2**), implying that it deterministically goes to zero as $n$ increases. Therefore, for all $\gamma_{[T3]} > 0$, there exist an $N_{[T3]}$ such that, for all $n > N_{[T3]}$, we have

$$[T3] \leq \gamma_{[T3]} \tag{70}$$

As a direct consequence of (57)-(69)-(70), we can state that for all $\gamma > 0$, for all $\delta > 0$, there exist a $N$ such that, for all $n > N$, we have

$$\mathbb{P}([T1] + [T2] + [T3] \leq \gamma) \geq 1 - \delta, \tag{71}$$

Combining (71) with (51), we can finally state that

$$\lim_{n\to\infty} D_l^n = \lim_{n\to\infty} \|\mathbf{h}(\Delta_n)\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F} - \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{h}(\Delta)\mathbf{F}\| = 0, \tag{72}$$

where the limit is taken in probability. The proof is concluded by combining (72) and (50).

## REFERENCES


[1] C. Battiloro et al., "Tangent bundle filters and neural networks: From manifolds to cellular sheaves and back," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[2] Y. LeCun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 84–90, 2012.

[4] O. Abdel-Hamid et al., "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 4277–4280.

[5] M. Aggarwal and M. N. Murty, *Machine Learning in Social Networks: Embedding Nodes, Edges, Communities, and Graphs*. New York, NY, USA: Springer-Verlag.

[6] Z. Wang, M. Eisen, and A. Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 1850–1863, 2022.

[7] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, Dec. 2020.

[8] F. Gama et al., "Convolutional neural network architectures for signals supported on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 1034–1049, Feb. 2019.

[9] F. Scarselli et al., "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[10] C. Battiloro et al., "Generalized simplicial attention neural networks," 2023, *arXiv:2309.02138*.

[11] C. Bodnar et al., "Weisfeiler and Lehman go topological: Message passing simplicial networks," in *Proc. Workshop Geometrical Topol. Representation Learn. (ICLR)*, 2021, pp. 1026–1037.

[12] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.

[13] L. Giusti et al., "Cell attention networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 1–8.

[14] C. Bodnar et al., "Weisfeiler and Lehman go cellular: CW networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34. Curran Associates, Inc., 2021, pp. 2625–2640.

[15] T. S. Cohen, M. Geiger, and M. Weiler, "A general theory of equivariant CNNs on homogeneous spaces," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[16] H. M. Riess and J. Hansen, "Multidimensional persistence module classification via lattice-theoretic convolutions," in *Proc. NeurIPS Workshop TDA Beyond*.

[17] Z. Wang, L. Ruiz, and A. Ribeiro, "Stability of neural networks on riemannian manifolds," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1845–1849.

[18] P. De Haan et al., "Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs," 2020, *arXiv:2003.05425*.

[19] S. C. Schonsheck et al., "Parallel transport convolution: A new tool for convolutional neural networks on manifolds," 2018, *arXiv:1805.07857*.

[20] A. Parada-Mayorga and A. Ribeiro, "Algebraic neural networks: Stability to deformations," *IEEE Trans. Signal Process.*, vol. 69, pp. 3351–3366, 2021.

[21] R. Bermejo et al., "Mathematical and numerical analysis of a nonlinear diffusive climate energy balance model," *Math. Comput. Model.*, vol. 49, no. 5, pp. 1180–1210, 2009.

[22] M. Collier and H. Connor, "Magnetopause surface reconstruction from tangent vector observations," *J. Geophys. Res. Space Phys.*, vol. 123, pp. 10189–10199, Nov. 2018.

[23] M. M. Bronstein et al., "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, 2017.

[24] M. Belkin and P. Niyogi, "Towards a theoretical foundation for Laplacian-based manifold methods," *J. Comput. Syst. Sci.*, vol. 74, no. 8, pp. 1289–1308, 2008.

[25] F. R. Chung, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1997.

[26] D. B. Dunson et al., "Spectral convergence of graph Laplacian and heat kernel reconstruction in $l_\infty$ from random samples," *Appl. Comput. Harmon. Anal.*, vol. 55, pp. 282–336, 2021.

[27] Z. Wang, L. Ruiz, and A. Ribeiro, "Convolutional neural networks on manifolds: From graphs and back," 2020, *arXiv:2210.00376*.

[28] R. Levie et al., "Transferability of spectral graph convolutional neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 272, pp. 1–59, 2021.

[29] N. Sharp, Y. Soliman, and K. Crane, "The vector heat method," *ACM Trans. Graph.*, vol. 38, no. 3, pp. 1–19, 2019.

[30] J. Hansen and R. Ghrist, "Learning sheaf Laplacians from smooth signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 5446–5450.

[31] J. Hansen and R. Ghrist, "Opinion dynamics on discourse sheaves," *SIAM J. Appl. Math.*, vol. 81, no. 5, pp. 2033–2060, 2021.

[32] C. Bodnar et al., "Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs," 2022.

[33] J. Leray and P. Malliavin, "*Selected Papers: Oeuvres Scientifiques*," Berlin, Germany: Springer, 1998.

[34] J.-P. Serre, "Faisceaux algébriques cohérents," *Ann. Math.*, pp. 197–278, 1955.

[35] A. Grothendieck, *A General Theory of Fibre Spaces With Structure Sheaf*. Dept. Math., Univ. Kansas,, 1955, no. 4.

[36] A. D. Shepard, *A Cellular Description of the Derived Category of a Stratified Space*. Brown Univ., 1985.

[37] J. M. Curry, *Sheaves, Cosheaves and Applications*. Univ. Pennsylvania, 2014.

[38] J. Hansen and R. Ghrist, "Toward a spectral theory of cellular sheaves," *J. Appl. Comput. Topol.*, vol. 3, no. 4, pp. 315–358, Dec. 2019, doi: 10.1007/s41468-019-00038-7.

[39] R. Ghrist and H. Riess, "Cellular sheaves of lattices and the Tarski Laplacian," *Homol. Homotopy Appl.*, vol. 24, no. 1, pp. 325–345, 2022.

[40] H. Riess and R. Ghrist, "Diffusion of information on networked lattices by gossip," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 5946–5952.

[41] J. Hansen and T. Gebhart, "Sheaf neural networks," 2020, *arXiv:2012.06333*.

[42] F. Barbero et al., "Sheaf neural networks with connection Laplacians," 2022, *arXiv:2206.08702*.

[43] A. Singer and H.-T. Wu, "Vector diffusion maps and the connection Laplacian," *Commun. Pure Appl. Math.*, vol. 65, no. 8, pp. 1067–1144, 2012.

[44] A. Singer and H. T. Wu, "Spectral convergence of the connection Laplacian from random samples," *Inf. Inference A J. IMA*, vol. 6, no. 1, pp. 58–123, 2017.

[45] D. I. Shuman et al., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[46] M. Puschel and J. M. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3572–3585, 2008.

[47] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1994.

[48] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, vol. 176. New York, NY, USA: Springer-Verlag, 2006.

[49] J. Lee, *Introduction to Smooth Manifolds*. New York, NY, USA: Springer-Verlag, 2000.

[50] A. V. Oppenheim et al., *Signals & Systems*. Pearson Educación, 1997.

[51] F. Gama et al., "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128–138, Nov. 2020.

[52] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001.

[53] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ. San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[54] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[56] K. Xu and W. Hu, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.

[57] C. Kanatsoulis and A. Ribeiro, "Graph neural networks are more powerful than we think," in *Proc. IEEE Trans. Acoust., Speech, Signal Process. (ICASSP)*, Piscataway, NJ, USA: IEEE, 2024, pp. 7550–7554. 2022.

[58] C. R. Qi et al., "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[59] B. J. Stolz et al., "Geometric anomaly detection in data," in *Proc. Nat. Acad. Sci.*, vol. 117, no. 33, pp. 19664–19669, 2020.

[60] V. Nanda, "Local cohomology and stratification," *Found. Comput. Math.*, vol. 20, pp. 195–222, 2020.
[61] F. D. Giovanni et al., "On over-squashing in message passing neural networks: The impact of width, depth, and topology," in *Int. Conf. Mach. Learn.*, 2023, pp. 7865–7885, 2023.

**Claudio Battiloro** received the Ph.D. degree in information and communication technologies from Sapienza University of Rome. He is a Former Visiting Associate with the SEAS of University of Pennsylvania. He is a Postdoctoral Fellow with T.H. Chan School of Public Health, Harvard University. His research interests include theory and methods for topological and algebraic signal processing, topological deep learning, and distributed optimization. He received different awards such as the IEEE SPS Italian Chapter Best M.Sc. Thesis Award (2020). In 2020, he graduated with distinction in the M.Sc. in data science with a (university-overall) Top 400 Students Award at Sapienza University.

**Zhiyang Wang** (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 2016 and 2019, respectively. She is currently working toward the Ph.D. degree with the Electrical and Systems Engineering Department, University of Pennsylvania. Her research interests include manifold signal processing, geometric deep learning, and wireless communication networks. She received the Best Student Paper Award at the 29th European Signal Processing Conference.

**Hans Riess** (Member, IEEE) received the B.S. degree in pure mathematics from Duke University, and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, in 2022. He is currently a Postdoctoral Associate with the Autonomous Systems Lab, Duke University where he leads efforts in the development and analysis of networked autonomous systems. He developed a novel approach of extracting global insights into complex systems using algebraic lattices and sheaves, with Robert Ghrist. He employs algebraic and topological methods to pioneer advancements in machine learning, autonomy, and optimization.

**Paolo Di Lorenzo** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunication engineering from Sapienza University of Rome, Rome, Italy, in 2008 and 2012, respectively. Currently, he is an Associate Professor with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Rome, Italy. He is the Technical Manager with the SNS-JU European Project 6G-GOALS, and the Principal Investigator with CNIT-Sapienza Research Unit in the H2020 European Project RISE 6G. His research interests include topological signal processing, AI-native communications, distributed optimization, and machine learning. He held a visiting research appointment with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA, USA. He is the recipient of the 2022 EURASIP Early Career Award, and of three Best Student Paper Awards at IEEE SPAWC10, EURASIP EUSIPCO11, and IEEE CAMSAP11, respectively. He is also the recipient of the 2012 GTTI (Italian National Group on Telecommunications and Information Theory) Award for the Best Ph.D. Thesis in communication engineering. He is currently an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING.

**Alejandro Ribeiro** (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the Universidad de la República Oriental del Uruguay, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, University of Minnesota, in 2005 and 2007, respectively. Currently, he is a Professor of electrical and systems engineering with the University of Pennsylvania (Penn), in 2008. His research interests include wireless autonomous networks, machine learning on network data, and distributed collaborative learning. Papers co-authored by him received the 2022 IEEE Signal Processing Society Best Paper Award, the 2022 IEEE Brain Initiative Student Paper Award, the 2021 Cambridge Ring Publication of the Year Award, the 2020 IEEE Signal Processing Society Young Author Best Paper Award, the 2014 O. Hugo Schuck Best Paper Award, and Paper Awards at EUSIPCO 2021, ICASSP 2020, EUSIPCO 2019, CDC 2017, SSP Workshop 2016, SAM Workshop 2016, Asilomar SSC Conference 2015, ACC 2013, ICASSP 2006, and ICASSP 2005. His teaching has been recognized with the 2017 Lindback Award for distinguished teaching and the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching. He received an Outstanding Researcher Award from Intel University Research Programs in 2019. He is a Penn Fellow Class of 2015 and a Fulbright Scholar Class of 2003.