ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096934

TANGENT BUNDLE FILTERS AND NEURAL NETWORKS: FROM MANIFOLDS TO CELLULAR SHEAVES AND BACK

C. Battiloro[1,2], *Z. Wang*[1], *H. Riess*[3], *P. Di Lorenzo*[2], *A. Ribeiro*[1]

[1] ESE Department, University of Pennsylvania, Philadelphia, USA
[2] DIET Department, Sapienza University of Rome, Rome, Italy
[3] ECE Department, Duke University, Durham, USA
E-mail: claudio.battiloro@uniroma1.it, zhiyangw@seas.upenn.edu

ABSTRACT

In this work we introduce a convolution operation over the tangent bundle of Riemannian manifolds exploiting the Connection Laplacian operator. We use this convolution operation to define tangent bundle filters and tangent bundle neural networks (TNNs), novel continuous architectures operating on tangent bundle signals, i.e. vector fields over manifolds. We discretize TNNs both in space and time domains, showing that their discrete counterpart is a principled variant of the recently introduced Sheaf Neural Networks. We formally prove that this discrete architecture converges to the underlying continuous TNN. We numerically evaluate the effectiveness of the proposed architecture on a denoising task of a tangent vector field over the unit 2-sphere.

Index Terms— Geometric Deep Learning, Tangent Bundle Signal Processing, Tangent Bundle Neural Networks, Cellular Sheaves

1. INTRODUCTION

The success of deep learning is mostly the success of Convolutional Neural Networks (CNNs) [1]. CNNs have achieved impressive performance in a wide range of applications showing good generalization ability. Based on shift operators in the space domain, one (but not the only one) key attribute is that the convolutional filters satisfy the property of shift equivariance. Nowadays, data defined on irregular (non-Euclidean) domains are pervasive, with applications ranging from detection and recommendation in social networks processing [2], to resource allocations over wireless networks [3], or point clouds for shape segmentation [4], just to name a few. For this reason, the notions of shifts in CNNs have been adapted to convolutional architectures on graphs (GNNs) [5, 6] as well as a plethora of other structures, e.g. simplicial complexes [7–10], cell complexes [11, 12], and manifolds [13]. In [14], a framework for algebraic neural networks has been proposed exploiting commutative algebras. In this work we focus on tangent bundles, a formal tool for describing and processing vector fields on manifolds, which are key elements in tasks such as robot navigation or flocking modeling.

Related Works. The renowned manifold assumption states that high dimensional data examples are sampled from a low-dimensional Riemannian manifold. This assumption is the fundamental block of manifold learning, a class of methods for non-linear dimensionality reduction. Some of these methods approximate manifolds with k-NN or geometric graphs via sampling points, i.e., for a fine enough sampling resolution, the graph Laplacian of the approximating graph "converges" to the Laplace-Beltrami operator of the manifold [15]. These techniques rely on the eigenvalues and eigenvectors of the graph Laplacian [16], and they give rise to a novel perspective on manifold learning. In particular, the above approximation leads to important transferability results of graph neural networks (GNNs) [17, 18], as well as to the introduction of Graphon and Manifold Neural Networks, continuous architectures shown to be limit objects of GNNs [19, 20]. However, most of the previous works focus on scalar signals, e.g. one or more scalar values attached to each node of graphs or point of manifolds; recent developments [21] show that processing vector data defined on tangent bundles of manifolds or discrete vector bundles [22, 23] comes with a series of benefits. Moreover, the work in [24] proves that it is possible to approximate both manifolds and their tangent bundles with certain cellular sheaves obtained from a point cloud via k-NN and Local PCA, such that, for a fine enough sampling resolution, the Sheaf Laplacian of the approximating sheaf "converges" to the Connection Laplacian operator. Finally, the work in [25] generalizes the result of [24] by proving the spectral convergence of a large class of Laplacian operators via the Principal Bundle set up.

Contributions. In this work we define a convolution operation over the tangent bundles of Riemannian manifolds with the Connection Laplacian operator. Our definition is consistent, i.e. it reduces to manifold convolution [19] in the one-dimensional bundle case, and to the standard convolution if the manifold is the real line. We introduce tangent bundle convolutional filters to process tangent bundle signals (i.e. vector fields over manifolds), we define a frequency representation for them and, by cascading layers consisting of tangent bundle filterbanks and nonlinearities, we introduce Tangent Bundle Neural Networks (TNNs). We then discretize the TNNs in the space domain by sampling points on the manifold and building a cellular sheaf [26] representing a legit approximation of both the manifold and its tangent bundle [24]. We formally prove that the 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, showing that space-time discretized TNNs are a principled variant of the very recently introduced Sheaf Neural Networks [23, 27, 28], discrete architectures operating on cellular sheaves and generalizing graph neural networks. Finally, we numerically evaluate the performance of TNNs on a denoising task of a tangent vector field of the unit 2-sphere.

Paper Outline. The paper is organized as follows. We start with some preliminary concepts in Section 2. We define the tangent bundle convolution and filters in Section 3, and Tangent Bundle Neural Networks (TNNs) in Section 4. In Section 5, we discretize TNNs in space and time domains, showing that discretized TNNs are Sheaf Neural Networks and proving the convergence result. Numerical results are in Section 6 and conclusions are in Section 7.

Authorized licensed use limited to: Universita degli Studi di Roma La Sapienza. Downloaded on September 13,2023 at 09:12:36 UTC from IEEE Xplore. Restrictions apply.

2. PRELIMINARY DEFINITIONS

Manifolds and Tangent Bundles. We consider a compact and smooth $d-$dimensional manifold $\mathcal{M}$ isometrically embedded in $\mathbb{R}^p$. Each point $x \in \mathcal{M}$ is endowed with a $d-$dimensional tangent (vector) space $\mathcal{T}_x\mathcal{M} \cong \mathbb{R}^d$, $\mathbf{v} \in \mathcal{T}_x\mathcal{M}$ is said to be a tangent vector at x and can be seen as the velocity vector of a curve over $\mathcal{M}$ passing through the point x (formal definitions can be found in [29]). The disjoint union of the tangent spaces is called the tangent bundle $\mathcal{TM} = \bigsqcup_{x\in\mathcal{M}} \mathcal{T}_x\mathcal{M}$. The embedding induces a Riemann structure on $\mathcal{M}$; in particular, it equips each tangent space $\mathcal{T}_x\mathcal{M}$ with an inner product, called Riemann metric, given, for each $\mathbf{v}$,$\mathbf{w} \in \mathcal{T}_x\mathcal{M}$, by

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\mathcal{T}_x\mathcal{M}} = \imath\mathbf{v} \cdot \imath\mathbf{w}, \tag{1}$$

where $\imath\mathbf{v} \in \mathcal{T}_x\mathbb{R}^p$ is the embedding of $\mathbf{v} \in \mathcal{T}_x\mathcal{M}$ in $\mathcal{T}_x\mathbb{R}^p \subset \mathbb{R}^p$ (the d-dimensional subspace of $\mathbb{R}^p$ which is the embedding of $\mathcal{T}_x\mathcal{M}$ in $\mathbb{R}^p$), with $\imath : \mathcal{TM} \to \mathcal{T}_x\mathbb{R}^p$ being an injective linear mapping referred to as differential [29], and $\cdot$ is the dot product. The Riemann metric induces also a probability measure μ over the manifold.

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. An inner product for tangent bundle signals $\mathbf{F}$ and $\mathbf{G}$ is

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

and the induced norm is $||\mathbf{F}||^2_{\mathcal{TM}} = \langle \mathbf{F}, \mathbf{F} \rangle_{\mathcal{TM}}$. We denote with $\mathcal{L}^2(\mathcal{TM})$ the Hilbert Space of finite energy (w.r.t. $||\cdot||_{\mathcal{TM}}$) tangent bundle signals. In the following we denote $\langle \cdot, \cdot \rangle_{\mathcal{TM}}$ with $\langle \cdot, \cdot \rangle$ when there is no risk of confusion.

Connection Laplacian. The Connection Laplacian is a (second-order) operator $\Delta : \mathcal{L}^2(\mathcal{TM}) \to \mathcal{L}^2(\mathcal{TM})$, given by the trace of the second covariant derivative defined (for this work) via the Levi-Cita connection [24]. The connection Laplacian Δ has some desirable properties: it is negative semidefinite, self-adjoint and elliptic. The Connection Laplacian characterizes the heat diffusion equation

$$\frac{\partial \mathbf{U}(x,t)}{\partial t} - \Delta \mathbf{U}(x,t) = 0, \tag{3}$$

where $\mathbf{U} : \mathcal{M} \times \mathbb{R}_0^+ \to \mathcal{TM}$ and $\mathbf{U}(\cdot, t) \in \mathcal{L}^2(\mathcal{TM}) \, \forall t \in \mathbb{R}_0^+$ (see [21] for a simple interpretation of (3)). With initial condition set as $\mathbf{U}(x,0) = \mathbf{F}(x)$, the solution of (3) is given by

$$\mathbf{U}(x,t) = e^{t\Delta}\mathbf{F}(x), \tag{4}$$

which provides a way to construct tangent bundle convolution, as explained in the following section. The Connection Laplacian Δ has a negative spectrum $\{-\lambda_i, \boldsymbol{\phi}_i\}_{i=1}^{\infty}$ with eigenvalues λ_i and corresponding eigenvector fields $\boldsymbol{\phi}_i$ satisfying

$$\Delta \boldsymbol{\phi}_i = -\lambda_i \boldsymbol{\phi}_i, \tag{5}$$

with $0 < \lambda_1 \leq \lambda_2 \leq \dots$. The λ_is and the $\boldsymbol{\phi}_i$s can be interpreted as the canonical frequencies and oscillation modes of $\mathcal{TM}$.

3. TANGENT BUNDLE CONVOLUTIONAL FILTERS

In this section we define the tangent bundle convolution of a filter impulse response $\widetilde{h}$ and a tangent bundle signal $\mathbf{F}$.

Definition 1. (Tangent Bundle Filter) Let $\widetilde{h} : \mathbb{R}^+ \to \mathbb{R}$ and let $\mathbf{F} \in \mathcal{L}^2(\mathcal{TM})$ be a tangent bundle signal. The manifold filter with impulse response $\widetilde{h}$, denoted with $\mathbf{h}$, is given by

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

where $\mathbf{U}(x,t)$ is the solution of the heat equation in (3) with $\mathbf{U}(x,0) = \mathbf{F}(x)$. Injecting (4) in (6), we obtain

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

The convolution in Definition 1 is consistent, i.e. it generalizes the manifold convolution [19] and the standard convolution in Euclidean domains (see Appendix A.4 in [30]). The frequency representation $\hat{F}$ of $\mathbf{F}$ can be obtained by projecting $\mathbf{F}$ onto the $\boldsymbol{\phi}_i$s basis

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

Definition 2. (Bandlimited Tangent Bundle Signals) A tangent bundle signal is said to be λ_M-bandlimited with $\lambda_M > 0$ if $\big[\hat{F}\big]_i = 0$ for all i such that $\lambda_i > \lambda_M$.

Proposition 1. Given a tangent bundle signal $\mathbf{F}$ and a tangent bundle filter $\mathbf{h}(\Delta)$ as in Definition 1, the frequency representation of the filtered signal $\mathbf{G} = \mathbf{h}(\Delta)\mathbf{F}$ is given by

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

Proof. See Appendix A.1 in [30].

Definition 3. (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 $\big[\hat{G}\big]_i = \hat{h}(\lambda_i)\big[\hat{F}\big]_i$, meaning that the tangent bundle filter is point-wise in the frequency domain. Therefore, we can write the frequency representation of the tangent bundle filter as

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

We note that the frequency response of the tangent bundle filter generalizes the frequency response of a standard time filter as well as a graph filter [31].

4. 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.

Definition 4. (Differential-preserving Non-Linearity) Denote with $U_x \subset \mathcal{T}_x\mathbb{R}^p$ the image of the injective differential $\imath$ in x. A mapping $\sigma : \mathcal{L}^2(\mathcal{TM}) \to \mathcal{L}^2(\mathcal{TM})$ is a differential-preserving non-linearity if it can be written as $\sigma(\mathbf{F}(x)) = \imath^{-1} \circ \widetilde{\sigma}_x \circ \imath\mathbf{F}(x)$, 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 $x \in \mathcal{M}$. Thus, 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 point-wise non linearity $\sigma(\cdot)$ is written as

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

A TNN of depth L with input signals $\{\mathbf{F}^q\}_{q=1}^{F_0}$ is built as the stack of L layers defined in (12), where $\mathbf{F}_0^q = \mathbf{F}^q$. To globally represent the TNN, we collect all the filter impulse responses in a function set $\mathcal{H} = \big\{\hat{h}_l^{u,q}\big\}_{l,u,q}$ and we describe the TNN $u-$th output as a mapping $\mathbf{F}_L^u = \boldsymbol{\Psi}_u\big(\mathcal{H}, \Delta, \{\mathbf{F}^q\}_{q=1}^{F_0}\big)$ to enhance that it is parameterized by filters $\mathcal{H}$ and Connection Laplacian Δ.

Authorized licensed use limited to: Universita degli Studi di Roma La Sapienza. Downloaded on September 13,2023 at 09:12:36 UTC from IEEE Xplore. Restrictions apply.

5. 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 principled way of discretizing them both in time and space domains.

Discretization in the Space Domain. The manifold $\mathcal{M}$, the tangent bundle $\mathcal{TM}$, and the Connection Laplacian Δ can be approximated starting from a set of sampled points (point-cloud). Knowing the coordinates of the sampled points, it is indeed possible to build a specific (orthogonal) cellular sheaf over an undirected geometric graph (see Appendix A.3 in [30]) such that its Sheaf Laplacian converges to the manifold Connection Laplacian as the number of sampled points (nodes) increases [25]. 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 μ over $\mathcal{M}$. We build a cellular sheaf $\mathcal{TM}_n$ following the Vector Diffusion Maps procedure whose details are listed in [24]. In particular, we build a geometric graph $\mathcal{M}_n$, with weights for nodes i and j set as

$$w_{i,j} = \exp\left(\frac{||x_i - x_j||^2}{\sqrt{\epsilon}}\right)\mathbb{I}\Big(0 < ||x_i - x_j||^2 \leq \sqrt{\epsilon}\Big), \quad (13)$$

where ϵ controls the chosen Gaussian Kernel. We then assign to each node i an orthogonal transformation $\mathbf{O}_i \in \mathbb{R}^{p \times \hat{d}}$ computed via a local PCA procedure, that is an approximation of a basis of the tangent space $\mathcal{T}_{x_i}\mathcal{M}$, where $\hat{d}$ is an estimate of d obtained from the same procedure. At this point, an approximation of the transport operator [29] from $\mathcal{T}_{x_i}\mathcal{M}$ to $\mathcal{T}_{x_j}\mathcal{M}$ is also needed. In the discrete domain, this translates to associating a matrix to each edge of the above graph (the restriction maps of the sheaf). For ϵ 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 they were coinciding, then the matrices $\mathbf{O}_i$ and $\mathbf{O}_j$ would have been the same up to an orthogonal transformation $\widetilde{\mathbf{O}}_{i,j}$ satisfying $\widetilde{\mathbf{O}}_{i,j} = {\mathbf{O}_i}^T\mathbf{O}_j$. However, the subspaces are not coinciding due to curvature. For this season, the transport operator approximation $\mathbf{O}_{i,j}$ is defined as the closest orthogonal matrix [24] to $\widetilde{\mathbf{O}}_{i,j}$, and it is computed as $\mathbf{O}_{i,j} = \mathbf{M}_{i,j}\mathbf{V}_{i,j}^T \in \mathbb{R}^{\hat{d} \times \hat{d}}$, where $\mathbf{M}_{i,j}$ and $\mathbf{V}_{i,j}$ are the SVD of $\widetilde{\mathbf{O}}_{i,j} = \mathbf{M}_{i,j}\mathbf{\Sigma}_{i,j}\mathbf{V}_{i,j}^T$. 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

$$\mathbf{S}_{i,j} = w_{i,j}\widetilde{\mathbf{D}}_i^{-1}\mathbf{O}_{i,j}\widetilde{\mathbf{D}}_j^{-1}, \quad \mathbf{D}_{i,i} = \text{ndeg(i)}\mathbf{I}_{\hat{d}}, \quad (14)$$

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

$$\Delta_n = \epsilon^{-1}\big(\mathbf{D}^{-1}\mathbf{S} - \mathbf{I}\big) \in \mathbb{R}^{n\hat{d} \times n\hat{d}}, \quad (15)$$

which is the approximated Connection Laplacian of the discretized manifold. A sheaf $\mathcal{TM}_n$ with this (orthogonal) structure is also said to be a discrete $\mathcal{O}\big(\hat{d}\big)-$bundle and represents a discretized version of $\mathcal{TM}$. We introduce a linear sampling operator $\mathbf{\Omega}_n^{\mathcal{X}} : \mathcal{L}^2(\mathcal{TM}) \to \mathcal{L}^2(\mathcal{TM}_n)$ to discretize a tangent bundle signal $\mathbf{F}$ as a sheaf signal $\mathbf{f}_n \in \mathbb{R}^{n\hat{d}}$ (a 0-cochain of the sheaf) such that

$$\mathbf{f}_n = \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}, \quad (16)$$

$$\mathbf{f}_n(x_i) := [\mathbf{f}_n]_{((i-1)\hat{d}+1):(i+1)\hat{d}} = {\mathbf{O}_i}^T i\mathbf{F}(x_i). \quad (17)$$

We are now in the condition of plugging the discretized operator and signal in the definition of tangent bundle filter in (7), obtaining

$$\mathbf{g}_n = \int_0^\infty \widetilde{h}(t)e^{t\Delta_n}\mathbf{f}_n\text{d}t = \mathbf{h}(\Delta_n)\mathbf{f}_n \in \mathbb{R}^{n\hat{d}}. \quad (18)$$

Following the same considerations of Section 4, we can define a discretized space tangent bundle neural network (D-TNN) as the stack of L layers of the form

$$\mathbf{x}_{n,l+1}^u = \sigma\left(\sum_{q=1}^{F_l}\mathbf{h}(\Delta_n)_l^{u,q}\mathbf{x}_{n,l}^q\right), \ u = 1, ..., F_{l+1}, \quad (19)$$

where (with a slight abuse of notation) σ has the same point-wise law of $\widetilde{\sigma}$ in Definition 4. 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 enhance that it is parameterized by filters $\mathcal{H}$ and the Sheaf Laplacian Δ_n. As the number of sampling points goes to infinity, the Sheaf Laplacian Δ_n converges to the Connection Laplacian Δ and the sheaf signal $\mathbf{x}_n$ converges to the tangent bundle signal $\mathbf{F}$. Combining these results, we prove in the next proposition that the output of a D-TNN converges to the output of the corresponding TNN as the sample size increases.

Theorem 1. Let $\mathcal{X} = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ be a set of n i.i.d. sampled points from measure μ over $\mathcal{M} \subset \mathbb{R}^p$ and $\mathbf{F}$ a bandlimited tangent bundle signal. Let $\mathcal{TM}_n$ be a cellular sheaf built from $\mathcal{X}$ as explained above, with $\epsilon = n^{-2/(\hat{d}+4)}$. Let $\mathbf{\Psi}_u\big(\mathcal{H}, \cdot, \cdot\big)$ be the $u-th$ output of a neural network with L layers parameterized by the operator Δ of $\mathcal{TM}$ or by the discrete operator Δ_n of $\mathcal{TM}_n$. If:

- the frequency response of filters in $\mathcal{H}$ are non-amplifying Lipschitz continuous;
- the non-linearities are differential-preserving;
- $\widetilde{\sigma}$ from Definition 4 is point-wise normalized Lipschitz continuous,
- $\mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}$ is a bandlimited sheaf signal

then it holds for each $u = 1, 2, \dots, F_L$ that:

$$\lim_{n\to\infty} ||\mathbf{\Psi}_u\big(\mathcal{H}, \Delta_n, \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{F}\big) - \mathbf{\Omega}_n^{\mathcal{X}}\mathbf{\Psi}_u\big(\mathcal{H}, \Delta, \mathbf{F}\big)||_{\mathcal{TM}_n} = 0, \quad (20)$$

with the limit taken in probability.

Proof. See Appendix A.2 in [30].

Discretization in the Time Domain. The discretization in space introduced in the previous section is still not enough for implementing TNNs in practice. Indeed, from Definition 1, we should learn the continuous time function $\widetilde{h}(t)$, and this is generally infeasible. To make TNNs and their training implementable, we discretize the function $\widetilde{h}(t)$ in the continuous time domain with a fixed sampling interval T_s. We replace the filter response function with a series of coefficients $h_k = \widetilde{h}(kT_s)$, $k = 0, 1, 2 \dots$. With $T_s = 1$ and fixing K samples over the time horizon, the discrete-time version of the convolution in (6) can be thus written as

$$\mathbf{h}(\Delta_n)\mathbf{F}(x) = \sum_{k=0}^{\infty} h_k e^{k\Delta}\mathbf{F}(x), \quad (21)$$

which corresponds to the form of a finite impulse response (FIR) filter with shift operator e^{Δ}. We can now inject the space discretization in the finite-time architecture in (21), obtaining an implementable manifold filter on the discretized manifold (cellular sheaf) $\mathcal{TM}_n$ as

$$\mathbf{g}_n = \mathbf{h}(\Delta_n)\mathbf{f}_n = \sum_{k=0}^{K-1} h_k e^{k\Delta_n}\mathbf{f}_n, \quad (22)$$

where $e^{k\Delta_n}$ is the matrix exponential. The discretized manifold filter of order K can be seen as a generalization of graph convolution [5] to the (orthogonal) cellular sheaf domain, thus we refer

Authorized licensed use limited to: Universita degli Studi di Roma La Sapienza. Downloaded on September 13,2023 at 09:12:36 UTC from IEEE Xplore. Restrictions apply.

		$\tau = 10^{-2}$	$\tau = 5 \cdot 10^{-2}$	$\tau = 1 \cdot 10^{-1}$
$n = 200$	DD-TNN	$\mathbf{2 \cdot 10^{-4}} \pm 1.6 \cdot 10^{-5}$	$\mathbf{4.9 \cdot 10^{-3}} \pm 2.4 \cdot 10^{-4}$	$\mathbf{1.9 \cdot 10^{-2}} \pm 1.3 \cdot 10^{-3}$
	MNN	$2.9 \cdot 10^{-4} \pm 1.5 \cdot 10^{-5}$	$7 \cdot 10^{-3} \pm 2.8 \cdot 10^{-4}$	$2.9 \cdot 10^{-2} \pm 1.5 \cdot 10^{-3}$
$n = 800$	DD-TNN	$\mathbf{2 \cdot 10^{-4}} \pm 5.7 \cdot 10^{-6}$	$\mathbf{5 \cdot 10^{-3}} \pm 1.2 \cdot 10^{-4}$	$\mathbf{1.9 \cdot 10^{-2}} \pm 4.6 \cdot 10^{-4}$
	MNN	$2.8 \cdot 10^{-4} \pm 8.7 \cdot 10^{-6}$	$7.3 \cdot 10^{-3} \pm 1.7 \cdot 10^{-4}$	$2.9 \cdot 10^{-2} \pm 6.9 \cdot 10^{-4}$

Table 1: MSE on the denoising task

e^{Δ_n} as a sheaf shift operator. At this point, by replacing the filter $\mathbf{h}_l^{pq}(\Delta_n)$ in (19) with (22), we obtain the following architecture:

$$\mathbf{f}_{n,l+1}^u = \sigma\left(\sum_{q=1}^{F_l}\sum_{k=1}^{K} h_{k,l}^{u,q}\left(e^{\Delta_n}\right)^k \mathbf{f}_{n,l}^q\right),\ u = 1, ..., F_{l+1}, \quad (23)$$

that we refer to as discretized space-time tangent bundle neural network (DD-TNN), which can be seen as a principled variant of the recently proposed Sheaf Neural Networks [23, 27, 28], with e^{Δ_n} as (sheaf) shift operator with order K diffusion. The layer in (23) can be rewritten 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

$$\mathbf{X}_{n,l+1} = \sigma\left(\sum_{k=1}^{K}\left(e^{\Delta_n}\right)^k \mathbf{X}_{n,l}\mathbf{H}_{l,k}\right) \in \mathbb{R}^{n\hat{d} \times F_{l+1}}, \quad (24)$$

where the filter weights $\{\mathbf{H}_{l,k}\}_{l,k}$ are learnable parameters. We have completed the process of building TNNs from cellular sheaves and back. Manifolds and their Tangent Bundles can be seen as the limits of graphs and cellular sheaves on them, making TNNs also a tool for analyzing large graphs with vector data.

6. NUMERICAL RESULTS

We assess the consistency of the proposed framework by designing a denoising task[1]. We work on the unit 2-sphere ($\mathcal{M} = \mathcal{S}_2$) and its tangent bundle. In particular, we uniformly sample the sphere on n points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and we compute the corresponding cellular sheaf $\mathcal{TM}_n$, Sheaf Laplacian Δ_n and signal sampler $\mathbf{\Omega}_n^{\mathcal{X}}$ as explained in Section 5 (also obtaining $\hat{d} = 2$). We consider the tangent vector field over the sphere given by

$$i\mathbf{F}(x, y, z) = (-y, x, 0) \in \mathbb{R}^3, \quad (25)$$

depicted in Fig. 1 for a realization of $\mathcal{X}$ with $n = 200$. At this point, we add AWGN with variance τ^2 to $i\mathbf{F}$ obtaining a noisy field $i\widetilde{\mathbf{F}}$, then we use $\mathbf{\Omega}_n^{\mathcal{X}}$ to sample it, obtaining $\widetilde{\mathbf{f}}_n \in \mathbb{R}^{2n}$. We test the perfomance of the TNN architecture (implemented with a DD-TNN as in (23)) by evaluating its ability of denoising $\widetilde{\mathbf{f}}_n$. We exploit a one layer architecture with 1 output feature (the denoised signal), and 5 filter taps. We train the architecture to minimize the MSE $\frac{1}{n}\|\widetilde{\mathbf{f}}_n - \mathbf{f}_{n,1}\|^2$ between the noisy signal $\widetilde{\mathbf{f}}_n$ and the output of the network $\mathbf{f}_{n,1}$ via the ADAM optimizer [32], with hyperparameters set to obtain the best results. We compare our architecture with a 1-layer Manifold Neural Network (MNN) architecture (implemented via a GNN as explained in [19]); to make the comparison fair, $i\widetilde{\mathbf{F}}$ evaluated on $\mathcal{X}$ is given as input to the MNN, organizing it in a matrix $\widetilde{\mathbf{F}}_n \in \mathbb{R}^{n \times 3}$. We train the MNN to minimize the MSE $\frac{1}{n}\|\widetilde{\mathbf{F}}_n - \mathbf{F}_{n,1}\|_F^2$, where $\|\|_F$ is the Frobenius Norm and $\mathbf{F}_{n,1}$ is the network output. It is easy 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 1 we evaluate TNNs and MNNs for two different sample sizes ($n = 200$ and $n = 800$), for three different noise standard deviation ($\tau = 10^{-2}$,$\tau = 5 \cdot 10^{-2}$ and $\tau = 10^{-1}$), showing the (again equivalent) MSEs $\frac{1}{n}\|\mathbf{f}_n - \mathbf{f}_{n,1}\|^2$ and $\frac{1}{n}\|\mathbf{F}_n - \mathbf{F}_{n,1}\|_F^2$, 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 $\mathcal{X}$. The results are averaged over 5 sampling realizations and 5 noise realizations per each of them. As the reader can notice from Table 1, TNNs always perform better than MNNs, due to their "bundle-awareness". Moreover, the mean performance remains stable as the number of points decreases, but the variances increase, meaning that having more sampling points (thus a better estimation of the Connection Laplacian) results in a more stable decision of the network. A real-world instance of this synthetic task could be denoising of Earth wind fields.

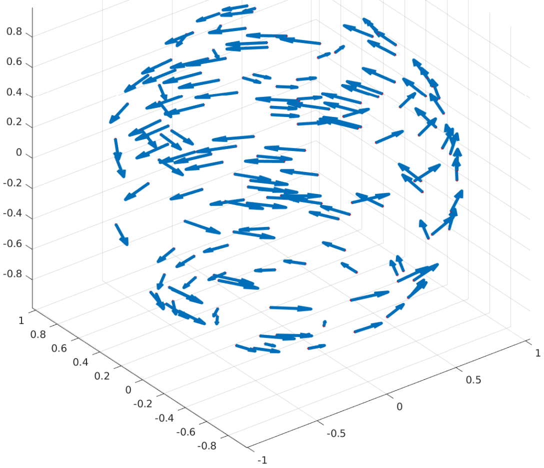

Fig. 1: Visualization of the embedded tangent vector field $i\mathbf{F}$

7. CONCLUSIONS

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. The results of this preliminary work, in addition to the introduction of a novel tool for processing manifold vector fields, could lead to a deeper understanding of topological neural architectures in terms of transferability and stability, with the opportunity of designing proper signal processing frameworks on tangent bundles and cellular sheaves. We plan to investigate these problems as well as applying TNNs to real-world complex tasks.

[1] `https://github.com/clabat9/Tangent-Bundle-Neural-Networks`

Authorized licensed use limited to: Universita degli Studi di Roma La Sapienza. Downloaded on September 13,2023 at 09:12:36 UTC from IEEE Xplore. Restrictions apply.

8. REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] M. Aggarwal and M. N. Murty, *Machine Learning in Social Networks: Embedding Nodes, Edges, Communities, and Graphs*. Springer Nature, 2020.

[3] Z. Wang, M. Eisen, and A. Ribeiro, “Learning decentralized wireless resource allocations with graph neural networks,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.

[4] Y. Xie, J. Tian, and X. X. Zhu, “Linking points with labels in 3d: A review of point cloud semantic segmentation,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, 2020.

[5] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, “Convolutional neural network architectures for signals supported on graphs,” *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2018.

[6] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The Graph Neural Network Model,” *IEEE Trans. on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[7] L. Giusti, C. Battiloro, P. D. Lorenzo, S. Sardellitti, and S. Barbarossa, “Simplicial attention neural networks,” *ArXiv*, vol. abs/2203.07485, 2022.

[8] C. Bodnar, F. Frasca, Y. G. Wang, N. Otter, G. Montufar, P. Liò, and M. M. Bronstein, “Weisfeiler and Lehman go topological: Message passing simplicial networks,” in *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[9] S. Barbarossa and S. Sardellitti, “Topological signal processing over simplicial complexes,” *IEEE Trans. on Signal Processing*, vol. 68, pp. 2992–3007, 2020.

[10] E. Isufi, F. Gama, and A. Ribeiro, “Edgenets: Edge varying graph neural networks,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[11] L. Giusti, C. Battiloro, L. Testa, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa, “Cell attention networks,” *arXiv:2209.08179*, 2022.

[12] C. Bodnar, F. Frasca, N. Otter, Y. Wang, P. Liò, G. F. Montufar, and M. Bronstein, “Weisfeiler and Lehman go cellular: Cw networks,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 2625–2640.

[13] Z. Wang, L. Ruiz, and A. Ribeiro, “Stability of neural networks on Riemannian manifolds,” in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1845–1849.

[14] A. Parada-Mayorga and A. Ribeiro, “Algebraic neural networks: Stability to deformations,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 3351–3366, 2021.

[15] M. Belkin and P. Niyogi, “Towards a theoretical foundation for laplacian-based manifold methods,” *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1289–1308, 2008.

[16] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[17] L. Ruiz, L. F. Chamon, and A. Ribeiro, “Transferability Properties of Graph Neural Networks,” *arXiv preprint arXiv:2112.04629*, 2021.

[18] R. Levie, W. Huang, L. Bucci, M. Bronstein, and G. Kutyniok, “Transferability of spectral graph convolutional neural networks,” *Journal of Machine Learning Research*, vol. 22, no. 272, pp. 1–59, 2021.

[19] Z. Wang, L. Ruiz, and A. Ribeiro, “Convolutional neural networks on manifolds: From graphs and back,” *arXiv preprint arXiv:2210.00376*, 2022.

[20] L. Ruiz, Z. Wang, and A. Ribeiro, “Graphon and graph neural network stability,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5255–5259.

[21] N. Sharp, Y. Soliman, and K. Crane, “The vector heat method,” *ACM Trans. Graph.*, vol. 38, no. 3, 2019.

[22] J. Hansen and R. Ghrist, “Learning sheaf Laplacians from smooth signals,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5446–5450.

[23] C. Bodnar, F. Di Giovanni, B. P. Chamberlain, P. Liò, and M. M. Bronstein, “Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs,” 2022. [Online]. Available: https://arxiv.org/abs/2202.04579

[24] A. Singer and H.-T. Wu, “Vector diffusion maps and the connection Laplacian,” *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21395

[25] A. Singer and H.-t. Wu, “Spectral convergence of the connection Laplacian from random samples,” 2013. [Online]. Available: https://arxiv.org/abs/1306.1587

[26] J. Hansen and R. Ghrist, “Toward a spectral theory of cellular sheaves,” *Journal of Applied and Computational Topology*, vol. 3, no. 4, pp. 315–358, Dec 2019. [Online]. Available: https://doi.org/10.1007/s41468-019-00038-7

[27] J. Hansen and T. Gebhart, “Sheaf neural networks,” 2020. [Online]. Available: https://arxiv.org/abs/2012.06333

[28] F. Barbero, C. Bodnar, H. S. d. O. Borde, M. Bronstein, P. Veličković, and P. Liò, “Sheaf neural networks with Connection Laplacians,” 2022. [Online]. Available: https://arxiv.org/abs/2206.08702

[29] J. M. Lee, *Riemannian manifolds: an introduction to curvature*. Springer Science & Business Media, 2006, vol. 176.

[30] C. Battiloro, Z. Wang, H. Riess, P. Di Lorenzo, and A. Ribeiro, “Tangent bundle filters and neural networks: From manifolds to cellular sheaves and back,” 2022. [Online]. Available: https://arxiv.org/abs/2210.15058

[31] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[32] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Authorized licensed use limited to: Universita degli Studi di Roma La Sapienza. Downloaded on September 13,2023 at 09:12:36 UTC from IEEE Xplore. Restrictions apply.