ManifoldNet: A Deep Neural Network for Manifold-Valued Data With Applications

Rudrasis Chakraborty, Jose Bouza, Jonathan H. Manton, *Fellow, IEEE*, and Baba C. Vemuri, *Fellow, IEEE*

Abstract—Geometric deep learning is a relatively nascent field that has attracted significant attention in the past few years. This is partly due to the availability of data acquired from non-euclidean domains or features extracted from euclidean-space data that reside on smooth manifolds. For instance, pose data commonly encountered in computer vision reside in Lie groups, while covariance matrices that are ubiquitous in many fields and diffusion tensors encountered in medical imaging domain reside on the manifold of symmetric positive definite matrices. Much of this data is naturally represented as a grid of manifold-valued data. In this paper we present a novel theoretical framework for developing deep neural networks to cope with these grids of manifold-valued data inputs. We also present a novel architecture to realize this theory and call it the ManifoldNet. Analogous to vector spaces where convolutions are equivalent to computing weighted sums, manifold-valued data 'convolutions' can be defined using the weighted Fréchet Mean (wFM). (This requires endowing the manifold with a Riemannian structure if it did not already come with one.) The hidden layers of ManifoldNet compute wFMs of their inputs, where the weights are to be learnt. This means the data remain manifold-valued as they propagate through the hidden layers. To reduce computational complexity, we present a provably convergent recursive algorithm for computing the wFM. Further, we prove that on non-constant sectional curvature manifolds, each wFM layer is a contraction mapping and provide constructive evidence for its non-collapsibility when stacked in layers. This captures the two fundamental properties of deep network layers. Analogous to the equivariance of convolution in euclidean space to translations, we prove that the wFM is equivariant to the action of the group of isometries admitted by the Riemannian manifold on which the data reside. To showcase the performance of ManifoldNet, we present several experiments using both computer vision and medical imaging data sets.

Index Terms—Weighted fréchet mean, equivariance, group action, riemannian manifolds

1 Introduction

Convolutional neural networks (CNNs) have attracted enormous attention in the past decade due to their significant success in Computer Vision, Speech Analysis and other fields. CNNs pioneered by [1] have gained much popularity since their significant success on Imagenet data reported in [2]. CNNs have traditionally been restricted to dealing with data residing in vector spaces. There has been a growing interest in the past several years to generalize CNNs and deep networks in general to data that reside in smooth non-euclidean spaces. Before embarking on a literature review, it would be useful to categorize the data space into the following classes: (i) Data that are samples of real-valued functions defined on a manifold and (ii) data that are manifold-valued and hence are sample points on a manifold. In this paper we will consider problems involving the latter category, namely, when the input data are sample points on known Riemannian manifolds, e.g., the manifold of symmetric positive definite (SPD) matrices, $\mathrm{SPD}(n)$, the n-sphere, $\mathbf{S}^n$, the special orthogonal group, $\mathbf{SO}(n)$, and the Grassmannian, $\mathbf{Gr}(p,n)$. More precisely, the domain of interest is an n-dimensional grid of manifold-valued data: a function of the form $f : U \rightarrow \mathcal{M}$ where $U \subset \mathbf{Z}^n$ is the image domain and $\mathcal{M}$ is a smooth Riemannian manifold.

We are not aware of much prior work on deep neural networks (DNNs) that can cope with the data-type described in (ii) above with the exception of [3], [4], [5], [6]. In [3], authors presented a deep network architecture for classification of hand-crafted features residing on a Grassmann manifold that form the input to the network. In [4], the authors presented a DNN architecture for data on $\mathrm{SPD}(n)$. In both of these works [3], [4], authors are not dealing with manifold-valued images as input data but simply a collection of features (derived from images) which are manifold-valued. Thus, the architecture does not involve the use of any convolution or equivalent operations for $\mathbf{Gr}(p,n)$ or $\mathrm{SPD}(n)$. Further, they do not use the natural invariant metric or intrinsic operations on the Grassmannian or the $\mathrm{SPD}(n)$ in the network blocks. Using intrinsic operations within the layers guarantees that the result remains on the manifold and hence one does not require any projection operations to ensure the result lies in the same space. Work in [5] addresses the issue of generalizing the concept of batch normalization to neural network architectures described in [3], [4]. Although, their batch-normalization generalization can be applied to our situation of manifold-valued fields/images as well. In [6], authors develop a local convolution layer which constraints the weight mask to be SPD by

- *R. Chakraborty is with the University of California, Berkeley, Berkeley, CA 94720 USA. E-mail: rudrasischa@gmail.com.*
- *J. Bouza and B.C. Vemuri are with the University of Florida, Gainesville, FL 32611 USA. E-mail: josejbouza@gmail.com, vemuri@cise.ufl.edu.*
- *J. Manton is with the University of Melbourne, Parkville, VIC 3010, Australia. E-mail: j.manton@ieee.org.*

Manuscript received 17 Sept. 2019; revised 16 May 2020; accepted 4 June 2020. Date of publication 22 June 2020; date of current version 7 Jan. 2022. (Corresponding author: Baba C. Vemuri.)
Recommended for acceptance by S. Zafeiriou, M. Bronstein, T. Cohen, O. Vinyals, L. Song, J. Leskovec, P. Liò, J. Bruna, and M. Gori.
Digital Object Identifier no. 10.1109/TPAMI.2020.3003846

learning unconstrained weights and taking the matrix inner product of such weight matrices and adding a fudge factor (scaled identity matrix) to guarantee the SPD property of the weight mask. Their convolution operation is the standard euclidean space convolution and doesn't involve proving equivariance to symmetry group actions admitted by the $\mathrm{SPD}(n)$ manifold.

There are several deep networks reported in the literature to deal with cases when data reside on 2-manifolds encountered in Computer Vision and Graphics for modeling shapes of objects. Some of these are based on graph-based representations of points on the surfaces in 3D and a generalization of CNNs to graphs [7], [8]. For more on Graph CNNs, we refer the reader to a recent comprehensive survey [9]. There is also recent work in [10] where the authors presented a deep network called geodesic CNN (GCNN), where convolutions are performed in local geodesic polar charts constructed on the manifold. These approaches fall in the category of functions on a manifold (the first category above) and hence are fundamentally distinct from our work reported here.

In this paper, we present a novel DNN framework called ManifoldNet. This is a potential analog of a CNN that can cope with input data are manifold-valued images i.e., the value set belongs to a Riemannian manifold. The motivation in defining the analog relies on the equivariance property. Note that convolution of functions in vector spaces are equivariant to translations. Further, it is easy to show that traditional convolutions of functions are equivalent to computing the weighted mean [11]. For the case of manifold-valued data, we can define the analogous operation of a weighted Fréchet mean (wFM) and prove that it is equivariant to the action of $I(\mathcal{M})$. This will be presented in a subsequent section. A preliminary conference version of this work was published in [12]. In [12], we presented manifold operations that allows for the generalization of CNN's to non-constant sectional curvature manifold valued data. In this article, we extend the framework to the case of constant sectional curvature manifolds as well and present a new architecture for the same. In comparison to our preliminary work in [12], in addition to the aforementioned significant extension, this paper contains an expanded theory section with more detailed analysis along with a detailed section on the network architecture and many more experiments pertinent to both computer vision and medical imaging.

Our key contributions in this work are: (i) we define the analog of convolution operations for manifold-valued data to be one of computing the wFM for which we present a provably convergent, efficient and recursive estimator. (ii) A proof of equivariance of wFM to the natural action of $I(\mathcal{M})$, generalizing a fundamental property of CNN's. (iii) We prove that on non-constant sectional curvature manifolds, each wFM layer is a contraction mapping and provide constructive evidence for its non-collapsibility when stacked in layers. Further, a proof of collapsibility for the case of constant sectional curvature manifolds. (iv) A novel deep architecture involving the Riemannian counterparts to the conventional CNN units. (v) Several experiments involving the application of ManifoldNet to both computer vision and medical imaging data sets. In computer vision, we present experiments on video classification and image reconstruction (using an auto-encoder +decoder setting). In medical imaging, we present experiments on (a) regression between changes in diffusional structure—captured in the Cauchy deformation tensor obtained via non-rigid registration of the ensemble average propagator (EAP) field computed from the patient scan to the EAP control atlas—and function in movement disorder patients. (b) An experiment on classification of Parkinson Disease (PD) patients and Controls (normal subjects) from diffusion magnetic resonance brain scans.

2 Group Action Equivariant Network for Manifold-Valued Data

In this section we will define the primary operations for extending deep learning architectures to manifold-valued images. Input data will be of the form $f : U \to \mathcal{M}$ for $U \subset \mathbf{Z}^N$ the image domain and $\mathcal{M}$ a Riemannian manifold, i.e., a field of $\mathcal{M}$-valued data. We replace the key blocks of a standard CNN architecture as follows: (a) Standard convolution replaced by a moving window of wFM (Section 2.1). (b) ReLU replaced by **G-transport/ G-expansion** (see Section 2.2). (c) Standard fully connected layer is replaced by an invariant final layer (see Section 2.3). In subsequent subsections, we will present a detailed description of each of these basic operations before moving on to a detailed description of the architecture we propose.

2.1 wFM on $\mathcal{M}$ as a Generalization of Convolution

We will begin by defining a convolution type operation for inputs sampled from a Riemannian manifold $\mathcal{M}$. This operation will slide a moving window of weights over the input points but replace the usual weighted sum (i.e., inner product) operation with a weighted Fréchet mean (wFM [13]).

Let $\{w_i\}_{i=1}^N$ be weights satisfying a *convexity constraint*, i.e., $w_i > 0$ for all i and $\sum_i w_i = 1$. Then, the wFM is defined as,

$$\mathsf{wFM}(\{X_i\}, \{w_i\}) = \arg\min_{M \in \mathcal{M}} \sum_{i=1}^{N} w_i d^2(X_i, M). \tag{1}$$

This definition works in any metric space, although the existence and uniqueness properties will vary. For Riemannian manifolds, these properties have been well characterized, and will introduce some relevant limitations on our architecture. Specifically, [14] has shown that the input points must lie in a ball of radius $r_{\mathrm{cvx}}(\mathcal{M})$ to ensure the wFM exists and is unique. Defining $r_{\mathrm{CVX}}(\mathcal{M})$ requires some background, which we detail in appendix (a), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.3003846, for completeness.

We can use this operation for our goal of constructing a convolution operation on manifold-valued fields. Suppose $f : \mathbf{Z}^N \to \mathcal{M}$ and $w : \mathbf{Z}^N \to \mathbf{R}$ are a manifold-valued field and a weight filter, respectively. Then we can define a convolution operation as

$$(f * w)(\mathbf{y}) := \mathbf{wFM}_{\mathbf{x} \in \mathbf{Z}^N}(f(\mathbf{x}), w(\mathbf{x} - \mathbf{y})). \tag{2}$$

For the rest of the paper, we will assume that the input samples on $\mathcal{M}$ lie inside an open ball $U = \mathcal{B}_{r_{\mathrm{cvx}}}(\mathcal{M})$. This will ensure the existence and uniqueness of the wFM as defined in Equation (1). We specify the value of $r_{\mathrm{CVX}}(\mathcal{M})$ for specific manifolds of interest in Section 3.

What About Equivariance?. Clearly Equation (1) generalizes the notion of euclidean mean to manifolds, but this is hardly a justification for its utility inside a deep network architecture. We will now show that the wFM operation generalizes a fundamental property of convolutions in euclidean space i.e., isometry equivariance. To this end we will now show that the wFM is equivariant to the action of the natural group of isometries of $\mathcal{M}$. First we formally define equivariance and the isometry group of a manifold $\mathcal{M}$.

Definition 1 (Equivariance). *Let X and Y be sets acted upon by G, i.e., G-sets [15]. Then, $F : X \to Y$ is said to be* G-equivariant *if $\forall g \in G, \forall x \in X, F(g.x) = g.F(x)$.*

Definition 2 (Group of isometries of $\mathcal{M}$). *We say a diffeomorphism $\phi : \mathcal{M} \to \mathcal{M}$ is an isometry if $d(\phi(x), \phi(y)) = d(x, y)$ for all $x, y \in \mathcal{M}$. Note that d is the distance induced by the Riemannian metric (see appendix (a)), available in the online supplemental material. The isometries of $\mathcal{M}$ form a group under composition. We denote this group by $I(\mathcal{M})$ and for $g \in I(\mathcal{M})$ we denote the result of applying g to $x \in \mathcal{M}$ by $g \cdot x$.*

Clearly $\mathcal{M}$ is a G-set, where $G = I(\mathcal{M})$. Now we are ready to prove the main theorem of this subsection, the equivariance of the wFM operation to the action of the isometry group $I(\mathcal{M})$. This result is intuitive, since, as can be noted from Equation (1), the wFM operation is fundamentally a metric space operation, and thus should be equivariant to isometric transformations.

Theorem 1. *Given $\{w_i\}$ satisfying the convex constraint, let $F : P \to U$ be a function defined by $\{X_i\} \mapsto \mathsf{wFM}(\{X_i\}, \{w_i\})$. Then, F is $I(\mathcal{M})$-equivariant.*

Proof. Let $g \in I(\mathcal{M})$ and $\{X_i\}_{i=1}^N \in P$. We want to show that,

$$g \cdot \mathsf{wFM}(\{X_i\}, \{w_i\}) = \mathsf{wFM}(\{g \cdot X_i\}, \{w_i\}). \tag{3}$$

Set $\widetilde{M} = \mathsf{wFM}(\{g \cdot X_i\}, \{w_i\})$. Then,

$$\sum_{i=1}^{N} w_i d^2\Big(g \cdot X_i, \widetilde{M}\Big) = \sum_{i=1}^{N} w_i d^2(X_i, g^{-1} \cdot \widetilde{M}\Big).$$

So that $\mathsf{wFM}(\{X_i\}, \{w_i\}) = g^{-1} \cdot \widetilde{M}$. Applying g to both sides we get Equation (3), completing the proof. □

Now that we have defined the convolution type operation as the wFM for manifold-valued data and have shown its equivariance to the natural isometry group action admitted by the manifold $\mathcal{M}$, we present a *computationally efficient estimator* for the wFM that will allow us to use it many times over within the wFM layers of a deep ManifoldNet.

How to Compute wFM *Efficiently?.* We will now define an efficient estimator of the wFM and state a statistical consistency theorem. To state and prove the statistical consistency we will need to interpret the samples X_i as being drawn from an unknown distribution over the continuous, manifold valued random variable X. Further, we will use the continuous counterpart of wFM, i.e., the weighted Frechet expectation (wFE) in the statement of Proposition 1. Given $\{X_i\}_{i=1}^N \subset U$ and $\{w_i := w(X_i)\}_{i=1}^N$ such that $\forall i, w_i > 0$, the nth estimate, M_n of $\mathsf{wFM}(\{X_i\}, \{w_i\})$ is given by the following recursion:

$$M_1 = X_1 \qquad M_n = \Gamma_{M_{n-1}}^{X_n}\left(\frac{w_n}{\sum_{j=1}^{n} w_j}\right). \tag{4}$$

Where $\Gamma_X^Y : [0, 1] \to U$ is the shortest geodesic curve from X to Y.[1] This gives us an efficient inductive/recursive way to define convolution operation on $\mathcal{M}$ who's complexity is linear in the number of points.

We have the following theorem, showing statistical consistency of the proposed estimator (4). See appendix (b), available in the online supplemental material, for the definition of the wFE and the proof.

Proposition 1. *Let $\{X_i\}_{i=1}^N$ be i.i.d. samples drawn from p_X on $\mathcal{M}$. Let the* wFE *be finite. Then, M_N converges a.s. to* wFE *as $N \to \infty$.*

Coming back to the proposed convolution operation in Equation (2), we will use this estimator M_n to compute an approximate wFM within each window. Proposition 1 means that as the kernel size increases (or equivalently, as the sampling rate of the underlying continuous image increases), the estimator converges to the true weighted Frechet Expectation of the sample distribution in the window (see appendix 5, available in the online supplemental material).

We will henceforth denote the above estimator the inductive wFM estimator (iFME). Note that in [16], [17], [18], the authors present recursive algorithms for FM computation on the hyper-sphere, Stiefel and $\mathrm{SPD}(n)$ manifolds respectively. These specific algorithms are special cases of our formulation since the wFM approach presented is applicable to any Riemannian manifold.

Take-Home Message. To summarize, the wFM (computed using the above recursive estimator) naturally generalizes the traditional convolution operation in vector spaces to smooth manifolds and possesses the fundamental group-equivariance property of convolutions. We extend this to a complete moving-window convolution-type operation in Section 3.

2.2 Nonlinear Operations Between Layers

Traditional deep network models use intermediate pointwise non-linear functions between convolutional layers (e.g., ReLU). There are at least two properties shared by all such functions: they are all (a) non-linear and (b) contractive. In light of the first property these functions are commonly called "non-linearities". The need for the first property is obvious: without a non-linear intermediate operation there is no "deep" learning, since the composition of linear layers will collapse to a single linear layer. The reasons for the second property are more complicated [19] but also important. This section will address both properties of non-linearities between layers. We will show that the wFM defined above is actually a contractive operation in its own right. This leaves us with the problem of non-linearity, which will be addressed on a case by case basis for non-constant and constant sectional curvature manifolds respectively.

The Contraction Property. We will begin by addressing the second point, namely, the contraction property. Formally, let F be a mapping from U to V. Assume U and V are metric spaces equipped with metrics d_U and d_V respectively. Then F is a contraction mapping *iff* $\exists c < 1$ such that $\forall x, y \in U$,

1. Observe that, in general wFM is defined with $\sum_{i=1}^N w_i = 1$, but in above definition, $\sum_{i=1}^N w_i \neq 1$. We can normalize $\{w_i\}$ to get $\{\widetilde{w}_i\}$ by $\widetilde{w}_i = w_i/(\sum_i w_i)$, but then Eq. (4) will not change as $\widetilde{w}_n/(\sum_{j=1}^n \widetilde{w}_j) = w_n/(\sum_{j=1}^n w_j)$

$d_V(F(x), F(y)) \leq c\, d_U(x, y)$, and F is a non-expansive mapping [19] *iff* $d_V(F(x), F(y)) \leq d_U(x, y)$.

One can easily see that the popular choices for nonlinear operations like ReLU and sigmoid are indeed non-expansive mappings. We will now show that the function wFM as defined in Equation (1), is a contraction mapping for any non-trivial choice of weights. Let $\{X_i\}_{i=1}^N$ and $\{Y_j\}_{j=1}^M$ be two sets of samples on $\mathcal{M}$. Without loss of generality assume $N \leq M$. We consider the set $\mathcal{U}^M = \underbrace{U \times \cdots \times U}_{M \text{ times}}$. Clearly $\{Y_j\}_{j=1}^M \in \mathcal{U}^M$ and we embed $\{X_i\}_{i=1}^N$ in $\mathcal{U}^M$ as follows: we construct $\{\widetilde{X}_i\}_{i=1}^M$ from $\{X_i\}_{i=1}^N$ by defining $\widetilde{X}_i = X_{(i-1) \bmod N+1}$. Let us denote the embedding by ι. Now, define the distance on $\mathcal{U}^M$ as $d(\{\widetilde{X}_i\}_{i=1}^M, \{Y_j\}_{j=1}^M) = \max_{i,j} d(X_i, Y_j)$. We say the choice of weights for wFM is trivial if one of the weights is 1 (hence all others are 0).

Proposition 2. *Assume that all $\{X_i\}_{i=1}^N$ and $\{Y_j\}_{j=1}^M$ are not same. Then, for all nontrivial choices of $\{\alpha_i\}_{i=1}^N$ and $\{\beta_j\}_{j=1}^M$ satisfying the convexity constraint, $\exists c < 1$ such that,*

$$d\Big(\mathsf{wFM}\Big(\{X_i\}_{i=1}^N, \{\alpha_i\}_{i=1}^N\Big), \mathsf{wFM}\Big(\{Y_j\}_{i=1}^M, \{\beta_j\}_{i=1}^M\Big)\Big) \leq c\, d\Big(\iota\Big(\{X_i\}_{i=1}^N\Big), \{Y_j\}_{j=1}^M\Big). \tag{5}$$

Note that, the above proposition holds for the particular choice of d.

Necessity for Non-Linearity. As mentioned before, the non-linearities between layers prevent the deep neural networks from collapsing to a single fully connected layer. The analogous question in the ManifoldNet framework is: *do composition of* wFM *operations collapse to a single* wFM *operation (possibly with different weights)?* The answer depends on the geometry of the manifold. In the case of constant sectional curvature manifolds, the answer is yes, and in the case of non-constant sectional curvature manifolds the answer is most likely no (see conjecture below). We now state both results here and provide a rigorous proof for the former and then present a numerical experiment why the latter might be true in appendix (c), available in the online supplemental material.

Theorem 2. *The multi-layer ManifoldNet is equivalent to the single layer Manifold-Net for data on Riemannian manifolds with constant sectional curvature.*

Conjecture 1. *The multi-layer ManifoldNet is not equivalent to the single layer ManifoldNet for data on Riemannian manifolds with non-constant sectional curvature.*

The above statements have important implications. An implication of Theorem 2 is that the wFM operation is not sufficient on its own for the constant curvature manifold case. This is not necessarily a limitation, since it is analogous to the euclidean convolution case, but it requires the development of some intermediate non-linearities. To overcome this, we propose several choices of non-collapsible pointwise operations that could be used in between the wFM layers. We emphasize that these operations are only necessary in the case where the manifold of interest is of constant curvature and in general these operations will not maintain equivariance across layers.

In contrast, Conjecture 1 states that for non-constant sectional curvature manifolds, the wFM is not only a contraction (as was shown before) but also a "non-linearity". Note that the term "non-linearity" is being abused here. It is possible in theory for the wFM to be non-linear yet be collapsible, meaning that several wFM layers are equivalent to a single one. The important property demonstrated by the Conjecture 1 is that wFM layers are most likely *non-collapsible* on non-constant sectional curvature manifolds. Moving forward, we occasionally use the term "non-linearity" regardless to maintain the analogy with the standard CNN case. Note that the degree of non-linearity provided by the wFM operation will depend on the curvature of the manifold, so that manifolds with non-constant but slowly varying sectional curvature may not provide much non-linearity in the wFM. A future avenue of work is to find the explicit relationship between the rate of change of sectional curvature and the degree of non-linearity in the wFM operation.

Choices of Non-Linearities. We now discuss some non-linear operators on $\mathcal{M}$.

e-Transport. After each layer of convolution (wFM), we can learn an element $g \in G$ to transport on $\mathcal{M}$. Given N as the output of a wFM layer, we define the G-transport operator G_{tr} as a learnable function defined as follows: $G_{\text{tr}}(N; g) = g.N$, where, $g \in G$ is learnable. This operator is equivariant to the action of G. But notice that, for manifolds with constant sectional curvature, this layer *does not* prevent the collapsibility issue as mentioned in Theorem 2. This motivates a more general non-linear operator defined below.

G-Expansion Operator. Let $\{X_i\}_{i=1}^N \subset \mathcal{M}$ be the points to which we want to apply the convolution operation. We define the expansion operator G_{ex} as a learnable function defined as follows:

$$G_{\text{ex}}(\{X_i\}; \{w_i\}, \{g_i\}) = \mathsf{wFM}(\{g_i.X_i\}, \{w_i\}), \tag{6}$$

where, $\{g_i\} \subset G$ are learnable. Notice that, this expansion operator *does not* preserve the equivariance but *does* prevent the collapsibility problem for manifolds with constant sectional curvature. An important point to note is that this operation may map the points to a geodesic ball greater than the convexity radius of $\mathcal{M}$. This would be an issue since the next wFM would not be well defined. To prevent this, we can explicitly check that the result of the G-expansion operation lies within a ball of convexity radius, and if not, revert to the initial input. A more principled approach to this issue will be addressed in future work.

Tangent ReLU. Let $\{X_i\}_{i=1}^N \subset \mathcal{M}$ be input points and set $\mu = \text{FM}(\{X_i\})$, the unweighted Fréchet mean. Then we define the TReLU operation by

$$\text{tReLU}(X_i) = \text{Exp}_\mu\Big(\iota^{-1}\Big(\text{ReLU}(\iota\Big(\text{Exp}_\mu^{-1}(X_i)\Big)\Big)\Big),$$

where Exp_μ and Exp_μ^{-1} are the Riemannian exponential and inverse exponential (log) maps centered at μ, respectively (see Appendix (a), available in the online supplemental material, for definitions). Let $\iota : T_\mu\mathcal{M} \to \mathbf{R}^m$ be an isomorphism from tangent space at μ to $\mathbf{R}^m$, where m is the dimension of $\mathcal{M}$. Explicitly, we lift the data points to the tangent space at the Fréchet mean, apply ReLU in the tangent space, and then map back to the manifold using the Riemannian

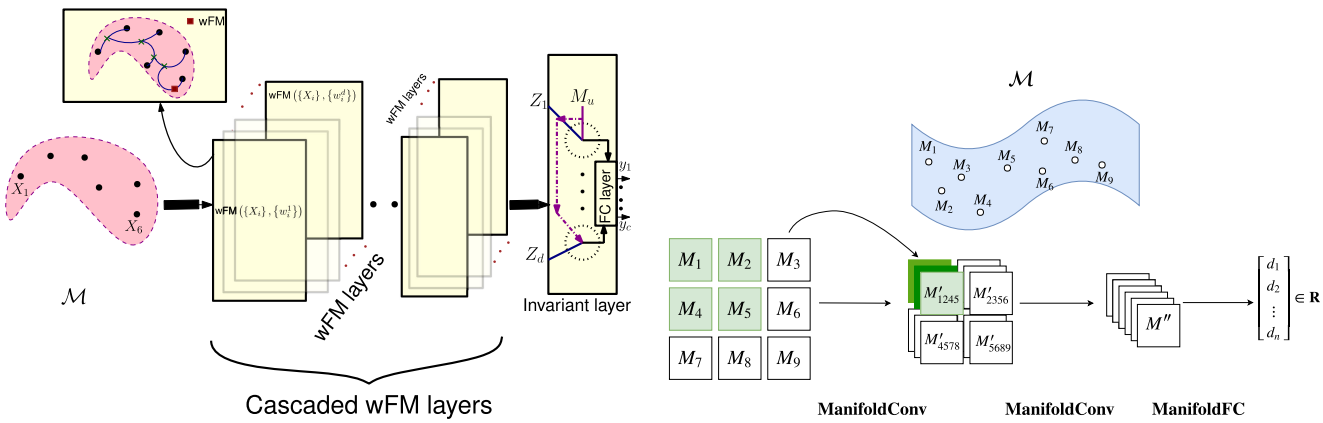

Fig. 1. *Left:* Schematic diagram of a ManifoldNet; *Right:* (2×2) ManifoldNet conv. example.

exponential map. This expansion operator *does not* preserve equivariance but *does* prevents the collapsibility problem for manifolds with constant sectional curvature.

Take-Home Message. For manifolds with constant sectional curvature, the presence of an intermediate non-linearity is essential to prevent collapsibility while for manifolds with non-constant sectional curvature, it is not strictly required.

2.3 The Invariant (Last) Layer

We will form a deep network by cascading multiple sliding wFM windows each of which acts as a convolution-type layer, possibly with a point-wise non-linearity operation in-between in the case of manifolds with constant sectional curvature. Each convolutional-type layer is equivariant to the group action, and hence at the end of the cascaded convolutional layers, the output is equivariant to the group action applied to the input of the network. Let d be the number of output channels. Each channel will be equivariant to the isometry group action. But in order to build a network that yields an output which is *invariant* to the group action we would like the last layer (i.e., the analogue of a linear classifier) to be invariant to the group action. This is accomplished in traditional CNNs using a combination of pooling layers and the last fully connected (FC) layer. We define a final layer which is explicitly invariant.

Construction of the Last Layer. The last layer is thus constructed as follows: Let $\{Z_i\}_{i=1}^d \subset \mathcal{M}$ be the output of d channels and $M_u = \mathsf{FM}(\{Z_i\}_{i=1}^d) = \mathsf{wFM}(\{Z_i\}_{i=1}^d, \{{}^1/_d\}_1^d)$ be the unweighted FM of the outputs $\{Z_i\}_{i=1}^d$. Then, we construct a layer with d outputs whose ith output $o_i = d(M_u, Z_i)$. Let c be the number of classes for the classification task, then, an FC layer with inputs $\{o_i\}$ and c output nodes is used. Finally, a softmax operation is then used at the c output nodes to obtain the outputs $\{y_i\}_{i=1}^c$.

Invariance of the Last Layer. In the following proposition, we claim that this last layer with $\{Z_i\}_{i=1}^d$ inputs and $\{y_i\}_{i=1}^c$ outputs is group invariant.

Proposition 3. *The last layer with $\{Z_i\}_{i=1}^d$ inputs and $\{y_i\}_{i=1}^c$ outputs is group invariant.*

Proof. Using the above construction, let $W \in \mathbf{R}^{c \times d}$ and $\mathbf{b} \in \mathbf{R}^c$ be the weight matrix and bias respectively of the FC layer. Then,

$$\mathbf{y} = F\big(W^T \mathbf{o} + \mathbf{b}\big) = F\big(W^T d(M_u, Z) + \mathbf{b}\big), \tag{7}$$

where, F is the softmax function. In the above equation, we treat $d(M_u, Z)$ as the vector $[d(M_u, Z_1), \ldots, d(M_u, Z_d)]^t$. Observe that, $g.M_u = \mathsf{FM}(\{g.Z_i\}_{i=1}^d)$. As each of the d channels is group equivariant, Z_i becomes $g.Z_i$. Because of the invariance property of the distance under group action, $d(g.M_u, g.Z_i) = d(M_u, Z_i)$. Hence, one can see that if we change the inputs $\{Z_i\}$ to $\{g.Z_i\}$, the output $\mathbf{y}$ will remain invariant. □

Take-Home Message. Analogous to the standard CNN, the presence of an invariant last layer is crucial to make our proposed ManifoldNet invariant to the action of G.

In Fig. 1 we present a self-explanatory schematic of the ManifoldNet depicting the different layers of processing the manifold-valued data as described above in Sections 2.1, 2.2, and 2.3. A self-explanatory schematic diagram to explain the concepts of equivariance and invariance is shown in Fig. 2.

3 Architecture

We now present the basic building blocks of the ManifoldNet architecture for both the non-constant and constant sectional curvature cases. Note that in both cases we can describe the input as an N-dimensional finite grid of manifold valued points, explicitly, a function $f : U \to \mathcal{M}$ where $U \subset \mathbf{Z}^N$ and $\mathcal{M}$ is a suitable manifold. For the purposes of exposition we will consider the case of a *manifold-valued image*, i.e., $N = 2$.

3.1 ManifoldConv and ManifoldFC

We begin by defining the *ManifoldConv* layer, a generalization of the convolutional layers in CNNs. In direct correspondence to the convolutional layer in CNNs, this layer involves moving/sliding a learnable weight kernel over the spatial dimensions of the image, but replaces weighted sums with

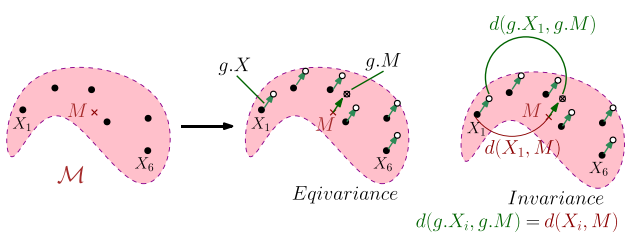

Fig. 2. Schematic of equivariance and invariance where $\{X_i\} \subset \mathcal{M}$, M is the wFM and $g \in G$ is the group element.

weighted Fréchet Means. Explicitly, if $f : \mathbf{Z}^2 \to \mathcal{M}$ is the layer input and $w : \mathbf{Z}^2 \to \mathbf{R}$ is the learned weight filter, then the *ManifoldConv* layer maps f to

$$(f * w)(\mathbf{y}) := \mathbf{wFM}_{\mathbf{x}\in\mathbf{Z}^2}(f(\mathbf{x}), w(\mathbf{x}-\mathbf{y})). \tag{8}$$

We will henceforth use the notation $f * w$ to denote *ManifoldConv* convolutions, where the manifold $\mathcal{M}$ is implicit in the functions. Note that traditional convolution layers in CNNs are (up to a scale factor) obtained as a special case when $\mathcal{M} = \mathbf{R}$. Hence, the *ManifoldConv* layer is a direct generalization of traditional convolution layers in CNNs.

From an implementation perspective there are several important points to mention:

1) As in the traditional convolution layers in CNNs, the weight kernel $w : \mathbf{Z}^2 \to \mathbf{R}$ is taken to be non-zero only in some neighborhood of the origin, i.e., the *kernel size*.
2) The weight kernel needs to satisfy the convexity constraint, i.e., $\sum_{\mathbf{x}\in\mathbf{Z}^2} w(\mathbf{x}) = 1$ and w must be strictly positive. We enforce this by learning an unconstrained set of underlying weights and then normalizing them to satisfy the convexity constraint before each forward pass.
3) The *ManifoldConv* layer can deal with multiple input and output channels using the same methods as traditional CNN's. This can be used to increase model capacity by learning multiple weight masks within each layer.
4) The *wFM* exists and is unique if all input points reside inside a geodesic ball of radius $r_{\text{inj}}(\mathcal{M})$, the injectivity radius of the manifold (see Section 2). We will address this condition on a case by case basis for each manifold below.
5) On many manifolds there exists no closed form expression for the *wFM*, so its naive computation involving gradient descent applied to the weighted Fr'echet functional can be computationally rather expensive. A network with just a few layers may have to compute millions of *wFM*'s just for one forward pass. To make this efficient we use the inductive Fréchet Mean estimator presented in Section 2. The inductive Fréchet Mean estimator transforms the *wFM*calculation into a series of geodesic function evaluations, so assuming $\mathcal{M}$ has a tractable closed form geodesic, this computation will be fast. Again, we elaborate on this for some specific manifolds below.

Each *ManifoldConv* layer also inherits some important properties from the *wFM* operation.

1) The *ManifoldConv* layers inherit equivariance to the natural action of $I(\mathcal{M})$, the isometry group associated with the manifold. Specifically, if $Z \in I(\mathcal{M})$ then $(Z \cdot f) * w = Z \cdot (f * w)$, where the action Z on a function is defined by the pointwise action on the output points, i.e., $(Z \cdot f)(x) = Z \cdot (f(x))$. This extends a key property that has made convolution layers in traditional CNNs so powerful: equivariance to an underlying group of isometries. Further, just like in the traditional CNNs, the ManifoldConv layer is equivariant to translations of the domain. In CNNs, it is well known that the network layers are equivariant to translations in the domain as well as range (for example adding a constant brightness to boost all of the image pixel values). In analogy to adding a constant brightness to all the pixel-values of an image we have, an application of the same group action to all the manifold-valued pixels in the manifold-valued image setting. Thus, what we need in this case is equivariance to the isometry group admitted by the manifold where the pixels take their values from. This group action equivariance was what was shown in Theorem-1 for the ManifoldNet. We can summarize these two properties by saying that $w * (Z \cdot f \circ T) = Z \cdot (w * f) \circ T$ where $T : \mathbf{Z^n} \to \mathbf{Z}^n$ is a translation in the domain.
2) Since the *wFM* operation is a contraction and, in the case of non-constant sectional curvature manifolds is non-collapsible, the *ManifoldConv* layer also inherits these properties. These are the two fundamental motives of the non-linearities in traditional CNNs. Therefore, the *ManifoldConv* layer acts as its own "non-linearity" in the case of non-constant sectional curvature manifolds, and, crucially, *ManifoldConv* layers can be stacked without intermediate ReLU type layers. Note that for the case of constant sectional curvature manifolds, we do have collapasibility and thus will require a "non-linearity" between the layers.

In classification tasks we would like the the output class to be *invariant* to some natural group action on the inputs. The *ManifoldConv* layers defined above give us equivariance to the isometry group action, so if we define a final layer that is invariant to the isometry group action then the entire network will be invariant. To achieve this we use the invariant final layer defined in Section 2.3. This layer maps the output activation of our *ManifoldConv* layer $g : U \to \mathcal{M}$ to a real valued vector, which is then easily fed through one or several (traditional) fully connected layers and finally through a softmax function for classification. We call this entire map from final activation to classification the *ManifoldFC* layer.

We now present some specific instances of the general theory and architecture that has been developed for manifolds of interest.

3.2 Non-Positive Sectional Curvature Example: $\mathcal{M} = \mathrm{SPD}(n)$

The manifold $\mathrm{SPD}(n)$ of symmetric positive define matrices with the $\mathrm{GL}(n)$-invariant metric is commonly encountered in computer vision and medical imaging applications. For e.g., in the former, for metric learning problems, covariance tracking etc. and in the latter, diffusion tensor imaging, elastography etc. For more on the use of covariance matrices in computer vision, we refer the reader to [20]. The space of SPD matrices is a Riemannian symmetric space with non-constant sectional curvature [21], and thus (Section 2) we can design *ManifoldNet* classifiers of the form

$$\mathbf{ManifoldConv} \to \ldots \to \mathbf{ManifoldConv} \to \mathbf{ManifoldFC}.$$

For $\mathrm{SPD}(n)$ with the $\mathrm{GL}(n)$-invariant metric the geodesic $\Gamma_X^Y : [0,1] \to \mathrm{SPD}(n)$ between two points X and Y on $SPD(n)$ is given by [22]

$$\Gamma_X^Y(t) = X^{\frac{1}{2}}\left(X^{-\frac{1}{2}}YX^{-\frac{1}{2}}\right)^t X^{\frac{1}{2}}. \tag{9}$$

$\mathrm{SPD}(n)$ is geodesically complete. Moreover, since $\mathrm{SPD}(n)$ has non-positive section curvature, a theorem of E. Cartan (see 6.1.5 in [23]) gives that the *wFM* is a global operation on $\mathrm{SPD}(n)$, i.e., for any points $\{X_i\} \subset \mathrm{SPD}(n)$ and weights $\{w_i\}$ satisfying the convexity constraint, $\mathbf{wFM}(\{X_i\}, \{w_i\})$ exists and is unique.

Note that the real valued powers of matrices in Equation (9) require eigen-decompositions for computation. We implement the *ManifoldConv* and *ManifoldFC* operations for $\mathrm{SPD}(n)$ in PyTorch.[2]

3.3 Constant Curvature Example: $\mathcal{M} = \mathbf{S}^{n-1}$

The hypersphere $\mathbf{S}^{n-1}$ is ubiquitous across a multitude of applications. In information geometry, the hypersphere arises as a parametrization of statistical manifolds, i.e., *manifolds of probability densities*, endowed with the Fisher-Rao metric. This metric is natural to the manifold of densities: it is invariant to reparametrizations of the density functions [24].

The unit Hilbert sphere identification with a statistical manifold is obtained using the so called *square root density parametrization*. For a comprehensive study of the square root parametrization the reader is referred to [24]. Formally, the square root parametrization is a map $\sqrt{\cdot} : \mathcal{X} \to \mathbf{S}^{n-1}$, $3 < n \leq \infty$, from a statistical manifold $\mathcal{X}$ to the unit hypersphere. Under the square root parametrization the natural $\mathrm{SO}(n)$-
invariant geodesic metric on $\mathbf{S}^{n-1}$ is equivalent to the Fisher-Rao metric, i.e., the square root parametrization $\sqrt{\cdot}$ is an isometry.

Note that $\mathbf{S}^{n-1}$ is a manifold with constant sectional curvature. We must thus use a "non-linearity" between successive layers (Section 2.2). We will later experiment with several of the non-linearities presented in (Section 2). On $\mathbf{S}^{n-1}$, geodesics take the simple form

$$\Gamma_{\mathbf{x}}^{\mathbf{y}}(t) = \frac{1}{\sin(\theta)}(\mathbf{x}\sin((1-t)\theta + \mathbf{y}\sin(t\theta)), \tag{10}$$

where, $\theta = \arccos(\mathbf{x}^t\mathbf{y})$ and $r_{\mathrm{inj}}(\mathbf{S}^{n-1}) = \pi/2$, meaning that the input points to the **wFM** on $\mathbf{S}^{n-1}$ must lie within distance π of each other. The square root parametrization maps points only onto the positive orthant of the hypersphere, which is contained within a ball of radius $\pi/2$. Hence the **wFM** is a global operation on square root parametrized densities, i.e., if $\{\sqrt{\mathbf{p}_i}\} \subset \mathbf{S}^{n-1}$ are the images of some densities under the square root parametrization and $\{w_i\}$ are weights satisfying the convexity constraint then $\mathbf{wFM}(\sqrt{\mathbf{p}_i}, \{w_i\})$ exists and is unique. We implement the *ManifoldConv*, the *ManifoldFC* operation for $\mathbf{S}^{n-1}$, and several choices of non-linearities in PyTorch.[3]

4 Experiments

We now evaluate the *ManifoldNet* framework on two each of medical imaging and vision tasks respectively: 1) Diffusion Tensor field classification, 2) nonlinear regression between structure and function, 3) Video Reconstruction and 4) Video Classification.

2. $\mathrm{SPD}(n)$ implementation
3. $\mathbf{S}^{n-1}$ implementation

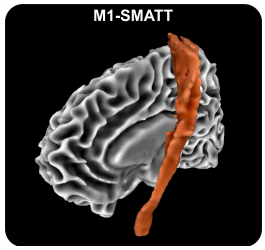

Fig. 3. M1 template.

TABLE 1
Comparison Results on Diffusion MRI Classification

Model	Non-linearity	# params.	time (s) / sample	Accuracy	
				Training Accuracy	Test Accuracy
DTI-ManifoldNet	None	$\sim$ **30K**	~ 0.3	$\mathbf{0.973 \pm 0.02}$	$\mathbf{0.948 \pm 0.03}$
ODF-ManifoldNet	Tangent-ReLU	$\sim 153K$	~ 0.02	0.951 ± 0.03	$\mathbf{0.942 \pm 0.02}$
ODF-ManifoldNet	G-expansion	$\sim 153K$	~ 0.02	0.934 ± 0.02	0.928 ± 0.01
ResNet-34	ReLU	$\sim 30M$	$\sim$ **0.008**	$\mathbf{0.984 \pm 0.04}$	0.713 ± 0.02
CapsuleNet	ReLU	$\sim 30M$	~ 0.009	0.63 ± 0.02	0.62 ± 0.04

4.1 Classification of Diffusion Tensor Images From Parkinson Disease Patients and Controls

In this experiment, we use a dataset consisting of diffusion weighted magnetic resonance (MR) images from 355 subjects diagnosed with Parkinson's disease and 356 control (healthy) subjects acquired at the University of Florida. This data is available for research use by request via the National Institute of Neurological Disorders (NINDS) Parkinson's Disease Biomarker Program (PDBP). All images were collected using a 3.0 T MR scanner (Philips Achieva) and 32-channel quadrature volume head coil. The parameters of the diffusion magnetic resonance image acquisition sequence were as follows: gradient directions = 64, b-values = 0 and 1000 s/mm2, repetition time = 7748 ms, echo time = 86 ms, flip angle = 90°, field of view = 224×224 mm, matrix size = 112×112, number of contiguous axial slices = 60, slice thickness = 2 mm, and SENSE factor $P = 2$. Eddy current correction was applied to each data set by using the widely used and publicly available FSL software pacakage [25].

From these raw diffusion weighted MR images we segment 12 regions of interest (ROIs) in the sensorimotor tracts, regions known to be affected by PD. This segmentation is achieved by registering to SMATT [26], a probabalistic atlas of the human sensorimotor tracts. For an example tract (M1) in the SMATT template, see Fig. 3. Our goal is to classify PD/Control directly from the diffusion MRI data. We test two different ManifoldNet based approaches for doing this, along with a traditional CNN model.

The first approach utilizes diffusion tensors, which capture the local diffusion process within a voxel using a symmetric positive definite matrix [27]. The second approach utilizes orientation distribution functions (ODF's), a more sophisticated representation than the diffusion tensors that captures the radial projection of the ensemble average of the diffusion propagator (probabality density) function [28]. It is well known that ODFs can capture crossing fibers, a phenomenon which diffusion tensor representation is incapable of modeling [29]. The final approach naively trains a CNN directly on the raw data represented as a scalar field with several channels. Below we describe the data-processing pipeline, architecture choice and results for each of the approaches considered. These results are also summarized in Table 1.

- *DTI Representation.* After extracting 12 ROIs (corresponding to 6 sensorimotor tracts in each hemisphere of the brain) we fit diffusion tensors to the data in these ROIs. This gives us a diffusion tensor field with 12 channels (one for each ROI). Since diffusion tensors are symmetric and positive definite matrices, we can represent each channel as a field $f_i : U \rightarrow \mathrm{SPD}(3)$ for $U \subset \mathbf{R}^3$.

  We use a 7-layer ManifoldNet architecture consisting of 5 ManifoldConv layers followed by a ManifoldFC layer and finally a softmax. We utilize a cross entropy loss function and train for 100 epochs using an Adam optimizer with a learning rate of 0.005. Utilizing this training procedure over a 10-fold cross-validtaion gives a *mean classification accuracy of 97.3 percent on the training set and 94.8 percent on the test set*. Inference time for a single SMATT fiber collection is about 0.3 s on a GTX 1080 Ti GPU.
- *ODF Representation.* In this case we fit orientation distribution functions to diffusion data within each voxel using the DiPy implementation of DSI with deconvolution [30]. This gives us an ODF field with 12 channels which, using the square root density parametrization [31], gives us an image of (Hilbert) sphere valued data, i.e., $f_i : U \rightarrow \mathbf{S}^\infty$ for each channel (ROI), where $U \subset R^3$. Of course, the probability density at each voxel is discretized, so that we can actually represent an ODF field by $f_i : U \rightarrow \mathbf{S}^N$ and $N < \infty$.

  Recall that the sphere is a manifold with constant sectional curvature, so we need to utilize some non-linearity in between consecutive ManifoldConv layers as described in Section 2.2. For this purpose, we test both the G-expansion operator and the tangent ReLU operation defined earlier. Specifically, we use 9 layers of ManifoldConv layers, each second ManifoldConv layer followed by a non-linearity (to increase the receptive field before the non-linearity [32]). This is followed by a ManifoldFC layer and a softmax for classification. We use the same training and testing procedure as in the DTI representation case, i.e., cross entropy loss trained for 100 epochs using Adam with a learning rate of 0.005. Using the G-expansion operator gives us a *mean classification accuracy of 93.4 percent on the training set and 92.8 percent on the testing set*. Using the tangent ReLU operation gives us a *mean classification accuracy of 95.1 percent on the training set and 94.2 percent on the test set*. Inference time for a single SMATT fiber collection is about 0.02 s on a GTX 1080 Ti GPU. We note that the relative efficiency of the recursive FM estimator on the sphere allows us to build a larger network while lowering inference time. This larger network improves classification accuracy. For reference, a 5 layer ODF representation network (the same size as the DTI case) gives a test classification accuracy of 74 percent, significantly worse than the larger network.
- *Raw Signal Representation.* In this case we do not fit any specific model to the data. Each q-space sampling direction of the raw signal corresponds to a scalar diffusion weighed image along that direction. Thus we interpret each sampling direction as a channel of a multi-channel image which we feed into a traditional CNN and CapsuleNet respectively.

  For the CNN architecture we utilize a ResNet-34 architecture which is trained from scratch using the training procedure described in the original paper [33]. 10-fold cross-validation gives us a *mean training accuracy of 98.4 percent*. We noted significant overfitting late in training so we report both an *early stoppage test accuracy of 71.3 percent and a non-early stoppage test accuracy of 42 percent*. Note that early stoppage was not considered for the previous two experiments, the reported test accuracy's are simply those obtained at the end of the final epochs.

  We also compared the performance of the ManifoldNet with a CapsuleNet [34], [35] with dynamic routing [36], again trained from scratch using the same training procedure reported in [36] on the data representation described above. Ten-fold cross validation yields a *training accuracy of 63 percent and a test accuracy of 62 percent*.

Note that ManifoldNet significantly outperforms the traditional CNN architecture on generalization performance, suggesting that the ManifoldNet architecture encodes better inductive biases for the problem. Beyond this, the ODF representation gives approximately equal accuracy to the DTI representation. This is slightly unexpected, since the ODF representation is more informative than the DTI representation. We hypothesize that it is due to the differences in architectures between the constant sectional curvature manifold corresponding to the ODF representation and non-constant sectional curvature manifold corresponding to the DTI representation, although more work is needed to conclusively determine the reasoning.

4.2 Nonlinear Regression Between Structure and Function

This dataset contains high angular resolution diffusion magnetic resonance image (HARDI) scans from, 1) healthy controls, 2) patients with essential tremor (ET) and 3) Parkinson's disease patients. This data pool contains scans from 25 controls, 15 ET and 26 PD patients. This HARDI data was acquired using the same acquisition parameters as described in the previous experiment. The dimension of each image is $(112 \times 112 \times 60)$. From each of these images, we identify the region of interest (ROI) (40 voxels in size) containing the Substantia Nigra (a neuroanatomical structure known to be affected most by PD and ET). In morphometric analysis, it is common to use the Cauchy deformation tensor (CDT) field to capture changes in a patient scan with respect to a reference template/atlas. Thus, in order to capture changes in patient HARDI scans with respect to the control atlas, we first non-rigidly register (see [37]) each of the EAP (ensemble average propagator–a probability density function) fields estimated from the input HARDI scan (see [38]) to the computed EAP atlas and obtain the CDT at each voxel in the ROI, given by $\sqrt{J^T J}$, where, J is the Jacobian of the non-rigid transformation [37]. The CDT is an SPD matrix of dimension (3×3) in this case. Hence, for each patient we extract a CDT field of dimension $(3 \times 3 \times 40)$. In this experiment, we seek to find the relationship between structural information in the form of CDT and an important clinical measure using the Movement Disorder Society's revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [39].

The MDS-UPDRS score is widely used to follow the longitudinal course of PD. These scores are obtained via interviews and clinical observations by an expert. In this experiment, available to us are the MDS-UPDRS scores for all the 58 subjects in the population under consideration. This score is a nonnegative natural number, with smaller values indicating normality.

For these 58 patients, we used a 3 layer ManifoldNet to find the relation between CDT field and MDS-UPDRS scores. We used an MSE loss and *obtained an R^2 statistic of 0.93*, outperforming conventional manifold regression techniques such as [40].

4.3 Video Reconstruction Experiment

Here we present experiments demonstrating the applicability of the theory layed out in Section 2 to dimensionality reduction and representation learning. Autoencoder architectures are commonly used for these purposes. This field has seen several significant advances in the past few years, including the introduction of denoising autoencoders [41], variational autoencoders [42], autoregressive models (PixelCNN [43], PixelRNN [44]), and flow-based generative models [45]. Many of these architectures are modifications of the traditional autoencoder network, which attempts to learn an identity map through a smaller latent space. In this experiment we will modify the usual autoencoder architecture by adding a linear dimensionality reduction layer in the latent space, achieved using a wFM of points on a Grassman manifold.

4.3.1 Background

To compute a linear subspace using the ManifoldNet framework we use an intrinsic averaging scheme on the Grassmannian. A point on the Grassmannian $\mathsf{Gr}(k,n)$ correspond to k-dimensional subspaces of the vector space $\mathbf{R}^n$. The Grassmannian is a smooth Riemannian homogeneous space [21] and a point $\mathcal{X} \in \mathsf{Gr}(k,n)$ on the Grassmannian can be specified by an orthonormal basis X, i.e., an $(n \times k)$ orthonormal matrix. Hauberg *et al.* [46] showed that the one dimensional principal subspace can be computed as an average of all one dimensional subspaces spanned by normally distributed data [47]. Motivated by this result, Chakraborty *et al.* [48] proposed an efficient intrinsic averaging scheme on $\mathsf{Gr}(k,n)$ that converges to the k-dimensional principal subspace of a normally distributed dataset in $\mathbf{R}^n$ [48]. In the ManifoldNet framework, we can modify this technique to learn a wFM of points on the Grassmannian that corresponds to a subspace of the latent space.

A traditional convolutional autoencoder performs nonlinear dimensionality reduction by learning an identity function through a small latent space. A common technique used when the desired latent space is smaller than the output of the encoder is to apply a fully connected layer to match the dimensions. We replace this fully connected layer by a weighted subspace averaging and projection block, called the Grassmann averaging layer. Specifically, we compute the wFM of the output of the encoder to get a subspace in the encoder output space. We then project the encoder output onto this space to obtain a reduced dimensionality latent space. We call an autoencoder with the Grassmann averaging block an autoencoder+iFME network, as shown in Fig. 4. In the experiments, we compare this to other dimensionality reduction techniques, including regular autoencoders that use fully connected layers to match encoder and latent space dimensions.

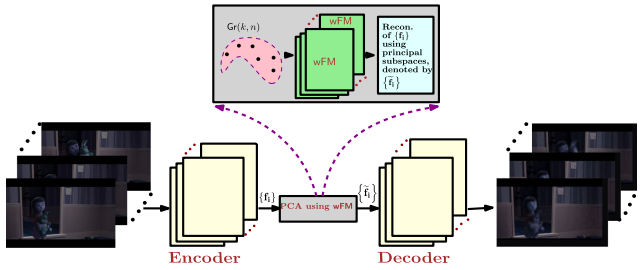

Fig. 4. Schematic description of autoencoder+iFME network.

4.3.2 Architecture and Results

We begin by testing on a 1000 frame color sample of video from the 1964 film "Santa Clause Conquers the Martians" of frame size 320×240. Here we use an 8 layer encoding-decoding architecture with $\mathbf{Conv} \rightarrow \mathbf{ELU} \rightarrow \mathbf{Batchnorm}$ layers, with the final layer applying a sigmoid activation to normalize the pixel values. The encoder returns a feature video consisting of 128 channels of size 120 for a dimension of (1000×15360). We compare a fully connected layer to a Grassmann averaging layer, both mapping to a desired latent space of dimension (1000×20). *The per pixel average reconstruction error for the Grassmann block network is 0.0110, compared to 0.0122 for the fully connected network, representing an improvement of 10.9 percent*. In general, the Grassmann averaging layer tends to do as well or better than the fully connected layer. Although in theory the fully connected layer can learn the same mapping as the Grassmann averaging layer, it has a much larger parameter space to search for this solution, implying that it is more likely to get trapped in local minima in the low loss regions of the loss function surface. *We also observe a parameter reduction of 46 percent*. In general the Grassmann averaging layer network is slower per iteration than the fully connected network, but also tends to exhibit faster convergence so that the time to reach the same reconstruction error is less for the Grassmann averaging layer. Overall, we see an improvement in all major performance categories.

4.4 Video Classification Experiment

Here we utilize the ManifoldNet architecture to design a low parameter video classifier. We start by using the method in [53] which we summarize here. Given a video with dimensions $(F \times 3 \times H \times W)$ of F frames, 3 color channels and a frame size of $(H \times W)$, we can apply a traditional convolution layer to obtain an output of size $(F \times C \times H' \times W')$ consisting of C channels of size $(H' \times W')$. Interpreting each channel as a feature map, we shift the features to have a zero mean and compute the covariance matrix of the convolution output to obtain a sequence of F symmetric positive (semi) definite (SPD) matrices of size $(C \times C)$. From here we can apply a series of temporal ManifoldNet wFMs to transform the $(F \times C \times C)$ input to a temporally shorter $(F' \times K \times C \times C)$ output, where K are the temporal wFM channels. We then reshape this to $(F'K \times C \times C)$ and pass it through an invariant final layer (Section 2.3) to obtain a vector of size $F'K$. Finally, a single FC+softmax layer is applied to produce a

TABLE 2
Comparison Results on Moving MNIST

Mode	# params.	time (s) / epoch	orientation (°)		
			30-60	10-15	10-15-20
SPD-TCN	**738**	~ 2.7	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.97 \pm 0.02}$
SPD-SRU [49]	1559	~ 6.2	$\mathbf{1.00 \pm 0.00}$	0.96 ± 0.02	0.94 ± 0.02
TT-GRU [50]	2240	$\sim \mathbf{2.0}$	$\mathbf{1.00 \pm 0.00}$	0.52 ± 0.04	0.47 ± 0.03
TT-LSTM [50]	2304	$\sim \mathbf{2.0}$	$\mathbf{1.00 \pm 0.00}$	0.51 ± 0.04	0.37 ± 0.02
SRU [51]	159862	~ 3.5	$\mathbf{1.00 \pm 0.00}$	0.75 ± 0.19	0.73 ± 0.14
LSTM [52]	252342	~ 4.5	0.97 ± 0.01	0.71 ± 0.07	0.57 ± 0.13

classification output. We call this the SPD temporal convolutional architecture SPD-TCN. In general, the SPD-TCN tends to perform very well on video classification tasks while *using very few parameters*, and runs efficiently.

We tested the ManifoldTCN on the Moving MNIST dataset [54]. In [49] authors developed a manifold valued recurrent network architecture, dubbed SPD-SRU, which produced state of the art classification results on a version of the Moving MNIST dataset in comparison to LSTM [52], SRU [51], TT-LSTM and TT-GRU [50] networks. For the LSTM and SRU networks, convolution layers are also used before the recurrent unit. We will compare directly with these results. For details of the various architectures used please see Section 5 of [49]. The Moving MNIST data generated in [54] consists of 1000 samples, each of 20 frames. Each sample shows two randomly chosen MNIST digits moving within a (64×64) frame, with the direction and speed of movement fixed across all samples in a class. The speed is kept the same across different classes, but the digit orientation differs across two different classes. For this experiment the SPD-TCN will consist of a single wFM layer with kernel size 5 and stride 3 returning 8 channels, leading to an (8×8) covariance matrix. We then apply three temporal SPD wFM layers of kernel size 3 and stride 2, with the following channels $1 \rightarrow 4 \rightarrow 8 \rightarrow 16$, i.e., after these three temporal SPD wFMs we have 16 temporal channels. This $(16 \times 8 \times 8)$ is used as an input to the invariant final layer to get a 16 dimensional output vector, which is transformed by a fully connected layer and softmax to obtain the output. We summarize the 10-fold cross validation results for several orientation differences between classes in Table 2. As evident from this table of comparisons, the SPD-TCN yields better results in comparison to the competing methods specifically in terms of the number of parameters as well as accuracy for the smaller angular orientation cases.

5 Discussion and Conclusion

In this paper, we presented a novel deep network suited for processing manifold-valued data sets. The key distinction between the work presented here and that presented in the literature under the umbrella of geometric deep learning is the type of input to the network. Here, we are interested in finite grids of manifold-valued data, such as a finite grid (thought of as an image) of (3,3) SPD matrices as was used in the diffusion tensor MRI based classification experiment presented earlier. One could easily apply standard CNNs to such data by ignoring the structure of the SPD matrices and simply vectorizing them. This however ignores the geometry underlying the data space and will in general lead to erroneous and inaccurate results. For instance, when we want to find the mean of two points on a sphere, if we ignore the geometry of the sphere and use the chordal distance between the two points to find the mean, this mean will not lie on the sphere. Thus, it is important to take the geometry of the data space into consideration and perform intrinsic operations admitted by the manifold on which the data lie. That said, in this paper we defined analogs of convolutional layers (in CNNs) called wFM layers that perform intrinsic operations on the manifold where the data reside. The existence and uniqueness of the wFM assumes that the data lie within a convexity radius specific to the manifold. This is usually the case in practice for the manifolds commonly encountered in applications namely, $\mathrm{SPD}(n)$, $\mathrm{Gr}(p,n)$, $\mathbf{S}^n$ and others. However, there might be special situations where the data does not satisfy this assumption and this matter needs closer examination in future work.

From a computational perspective, since the wFM operations need to be performed anywhere from several thousands of times for small networks, to several millions of times for larger networks, we presented an efficient recursive estimator of wFM. Currently, the rate of convergence of the recursive estimator is linear [16], [48] in the number of data points whose wFM is being computed. There is however scope to improve this rate of convergence by using the geometry of the manifold in weight selection within the recursive estimator. We will address this issue in our future work.

In summary, our key contributions in the work presented here are: (a) A novel deep network to be perceived as a generalization of the CNN to manifold-valued data inputs using purely intrinsic operations on the data manifold. (b) Analogous to convolutions in vector spaces–which can be computed using the weighted sums–we present wFM operations on the manifold and prove the equivariance of the wFM to natural group actions admitted by the manifold. (c) An efficient recursive wFM estimator that is provably convergent. (d) Constructive evidence on the non-collapsibility of stacked wFM layers (wihtout any intermediate nonlinearities such as the ReLU) for non-constant curvature manifolds and a theorem proving the collapsibility in the case of constant curvature manifolds. (e) Several experimental results demonstrating the efficacy of the ManifoldNet for applications in computer vision and medical imaging.

Acknowledgments

This work was supported in part by the NSF Grant IIS-1724174 to BCV. The authors would like to thank Dr. David Vaillancourt of UFL for providing the diffusion MRI data.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[3] Z. Huang, J. Wu, and L. Van Gool, "Building deep networks on Grassmann manifolds," *32nd AAAI Conf. Artif. Intell.*, 2018.
[4] Z. Huang and L. J. Van Gool, "A riemannian network for SPD matrix learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, vol. 1, pp. 2036–2042.
[5] D. Brooks, O. Schwander, F. Barbaresco, J.-Y. Schneider, and M. Cord, "Riemannian batch normalization for SPD neural networks," *Adv. Neural Inf. Proc. Syst.*, pp. 15489–15500, 2019.

[6] T. Zhang, W. Zheng, Z. Cui, and C. Li, "Deep manifold-to-manifold transforming network," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 4098–4102.

[7] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*.

[8] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[10] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 37–45.

[11] A. Goh, C. Lenglet, P. M. Thompson, and R. Vidal, "A nonparametric riemannian framework for processing high angular resolution diffusion images and its applications to ODF-based morphometry," *NeuroImage*, vol. 56, no. 3, pp. 1181–1201, 2011.

[12] R. Chakraborty, J. Bouza, J. Manton, and B. C. Vemuri, "A deep neural network for manifold-valued data with applications to neuroimaging," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2019, pp. 112–124.

[13] M. Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié," *Annales de l'I. H. P.*, vol. 10, no. 4, pp. 215–310, 1948.

[14] B. Afsari, "Riemannian L^p center of mass: Existence, uniqueness, and convexity," *Proc. Amer. Math. Soc.*, vol. 139, no. 02, pp. 655–655, 2011. [Online]. Available: http://www.ams.org/jourcgi/jour-getitem?pii=S0002–9939-2010-10541-5

[15] D. S. Dummit and R. M. Foote, *Abstract Algebra*, vol. 3. Hoboken, NJ, USA: Wiley, 2004.

[16] H. Salehian, R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri, "An efficient recursive estimator of the Fréchet mean on a hypersphere with applications to Medical Image Analysis," in *Proc. 5th MICCAI Workshop Math. Found. Comput. Anatomy*, 2015, pp. 143–154.

[17] R. Chakraborty *et al.*, "Statistics on the stiefel manifold: Theory and applications," *The Ann. Statist.*, vol. 47, no. 1, pp. 415–438, 2019.

[18] Y. Lim and M. Pálfia, "Weighted inductive means," *Linear Algebra Appl.*, vol. 453, pp. 59–83, 2014.

[19] S. Mallat, "Understanding deep convolutional networks," *Philos. Trans. A*, vol. 374, 2016, Art. no. 20150203. [Online]. Available: http://arxiv.org/abs/1601.04920

[20] H. Q. Minh, V. Murino, and H. Q. Minh, *Algorithmic Advances in Riemannian Geometry and Applications*. Berlin, Germany: Springer, 2016.

[21] S. Helgason, *Differential Geometry, Lie Groups, and Symmetric Spaces*, vol. 80. Cambridge, MA, USA: Academic Press, 1979.

[22] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIMAX*, vol. 26, no. 3, pp. 735–747, 2005.

[23] M. Berger, *A Panoramic View of Riemannian Geometry*. Berlin, Germany: Springer, 2012.

[24] A. Srivastava, I. Jermyn, and S. Joshi, "Riemannian analysis of probability density functions with applications in vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[25] S. M. Smith *et al.*, "Advances in functional and structural mr image analysis and implementation as FSL," *NeuroImage*, vol. 23, pp. S208–S219, 2004.

[26] D. Archer, D. Vaillancourt, and S. Coombes, "A template and probabilistic atlas of the human sensorimotor tracts using diffusion MRI," *Cerebral Cortex*, vol. 28, pp. 1–15, Mar. 2017.

[27] P. J. Basser, J. Mattiello, and D. LeBihan, "Mr diffusion tensor spectroscopy and imaging," *Biophysical J.*, vol. 66, no. 1, pp. 259–267, 1994.

[28] D. S. Tuch, T. G. Reese, M. R. Wiegell, and V. J. Wedeen, "Diffusion MRI of complex neural architecture," *Neuron*, vol. 40, no. 5, pp. 885–895, 2003.

[29] D. C. Alexander, "Multiple-fiber reconstruction algorithms for diffusion MRI," *White Matter Cogn. Neurosci. Advances Diffusion Tensor Imag. Appl.*, vol. 1064, pp. 113–133, 2005.

[30] E. Garyfallidis *et al.*, "Dipy, a library for the analysis of diffusion MRI data," *Front. Neuroinformatics*, vol. 8, 2014, Art. no. 8.

[31] A. Srivastava, I. Jermyn, and S. Joshi, "Riemannian analysis of probability density functions with applications in vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[32] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4898–4906.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[34] G. E. Hinton, Z. Ghahramani, and Y. W. Teh, "Learning to parse images," in *Proc. 12th Int. Conf. Neural Inf. Process. Syst.*, 2000, pp. 463–469.

[35] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw. Mach. Learn.*, 2011, pp. 44–51.

[36] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," *Adv. Neural Inf. Proc. Syst.*, pp. 3856–3866, 2017.

[37] G. Cheng, B. C. Vemuri, P. R. Carney, and T. H. Mareci, "Non-rigid registration of high angular resolution diffusion images represented by gaussian mixture fields," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2009, pp. 190–197.

[38] B. Jian and B. C. Vemuri, "A unified computational framework for deconvolution to reconstruct multiple fibers from DWMRI," *IEEE Trans. Med. Imag.*, vol. 26, no. 11, pp. 1464–1471, Nov. 2007.

[39] C. Ramaker, J. Marinus, A. M. Stiggelbout, and B. J. Van Hilten, "Systematic evaluation of rating scales for impairment and disability in parkinson's disease," *Movement Disorders: Official J. Movement Disorder Soc.*, vol. 17, no. 5, pp. 867–876, 2002.

[40] M. Banerjee, R. Chakraborty, E. Ofori, M. S. Okun, D. E. Viallancourt, and B. C. Vemuri, "A nonlinear regression technique for manifold valued data with applications to medical image analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4424–4432.

[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *2nd Int. Conf. Learn. Representations*, 2014.

[43] A. V. D. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4797–4805.

[44] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[45] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Adv. Neural Inf. Proc. Syst.*, pp. 10215–10224, 2018.

[46] S. Hauberg, A. Feragen, R. Enficiaud, and M. J. Black, "Scalable robust principal component analysis using grassmann averages," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2298–2311, Nov. 2016.

[47] S. Hauberg, A. Feragen, R. Enficiaud, and M. J. Black, "Scalable robust principal component analysis using grassmann averages," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2298–2311, Nov. 2016.

[48] R. Chakraborty, S. Hauberg, and B. C. Vemuri, "Intrinsic grassmann averages for online linear and robust subspace learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 801–809.

[49] R. Chakraborty *et al.*, "Statistical recurrent models on manifold valued data," *Adv. Neural Inf. Proc. Syst.*, pp. 8883–8894, May 2018.

[50] Y. Yang, D. Krompass, and V. Tresp, "Tensor-train recurrent neural networks for video classification," in *Proc. 34th Int. Conf. Mach. Learn.-Vol. 70*, Jul. 2017, pp. 3891–3900.

[51] J. B. Oliva, B. Poczos, and J. Schneider, "The statistical recurrent unit," in *Proc. 34th Int. Conf. Mach. Learn.-Vol. 70*, Mar. 2017, pp. 2671–2680.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[53] K. Yu and M. Salzmann, "Second-order convolutional neural networks," Mar. 2017, *ArXiv e-prints*.

[54] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 843–852. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045209

Rudrasis Chakraborty received the PhD degree in computer science from the University of Florida, Gainesville, Florida, in 2018. He is currently a postdoc at UC Berkeley, Berkeley, California. His research interests include the intersection of geometry, ML, and computer vision.

Jose Bouza is currently working toward the undergraduate degree in mathematics and computer science at the University of Florida, Gainesville, Florida. His primary research interests include computer vision and applied topology.

Jonathan H. Manton (Fellow, IEEE) received the bachelor of science (mathematics) and bachelor of engineering (electrical) degrees, in 1995, and the PhD degree, in 1998, from The University of Melbourne, Australia. He holds a distinguished chair at the University of Melbourne, Australia with the title future generation professor. He is also an adjunct professor with the Mathematical Sciences Institute, Australian National University, Australia, a fellow of the Australian Mathematical Society (FAustMS). In 2005 he became a full professor in the Research School of Information Sciences and Engineering (RSISE) at the Australian National University, Australia. From mid-2006 till mid-2008, he was on secondment to the Australian Research Council as executive director, mathematics, information and communication sciences. His principle fields of interest are mathematical systems theory (including signal processing and optimisation), geometry and topology (differential and algebraic), and learning and computation (including systems biology, systems neuroscience, and machine learning).

Baba C. Vemuri (Fellow, IEEE) received the PhD degree in electrical and computer engineering from the University of Texas at Austin, Austin, Texas. Currently, he holds the Wilson and Marie Collins professorship in engineering at the University of Florida, Gainesville, Florida. He is a professor at the Department of Computer and Information Sciences and Engineering, the Department of Statistics, ECE and BME at the University of Florida, Gainesville, Florida. His research interests include geometric statistics, computer vision, machine learning and medical imaging. In the past, he was an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Medical Imaging* and the *Computer Vision and Image Understanding*. Currently, he is an associate editor for the *International Journal of Computer Vision* and MedIA respectively. He received the IEEE Computer Society's Technical Achievement Award (2017) and is a fellow of the ACM (2009).