Dilated Convolutional Neural Networks for Sequential Manifold-valued Data

Xingjian Zhen[†*] Rudrasis Chakraborty[‡*] Nicholas Vogt[†] Barbara B. Bendlin[†] Vikas Singh[†]
[†]University of Wisconsin Madison [‡]University of California, Berkeley
[*]Equal contribution

Abstract

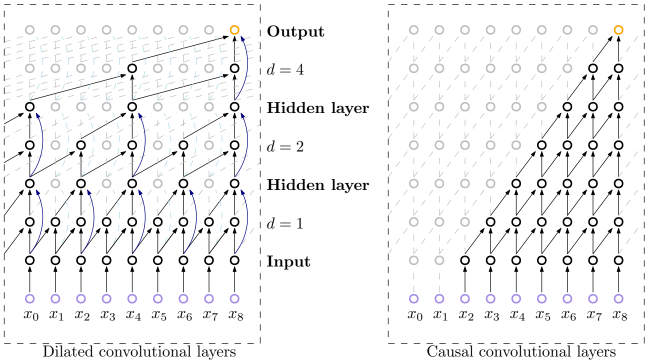
Efforts are underway to study ways via which the power of deep neural networks can be extended to non-standard data types such as structured data (e.g., graphs) or manifold-valued data (e.g., unit vectors or special matrices). Often, sizable empirical improvements are possible when the geometry of such data spaces are incorporated into the design of the model, architecture, and the algorithms. Motivated by neuroimaging applications, we study formulations where the data are sequential manifold-valued measurements. *This case is common in brain imaging, where the samples correspond to symmetric positive definite matrices or orientation distribution functions. Instead of a recurrent model which poses computational/technical issues, and inspired by recent results showing the viability of dilated convolutional models for sequence prediction, we develop a dilated convolutional neural network architecture for this task. On the technical side, we show how the modules needed in our network can be derived while explicitly taking the Riemannian manifold structure into account. We show how the operations needed can leverage known results for calculating the weighted Fréchet Mean (wFM). Finally, we present scientific results for group difference analysis in Alzheimer's disease (AD) where the groups are derived using AD pathology load: here the model finds several brain fiber bundles that are related to AD even when the subjects are all still cognitively healthy.*

1. Introduction

The classical definition of convolution assumes that the data are scalar or vector-valued and lie on discrete equally spaced intervals. This assumption is ideal for natural images and central to how we use convolutional filters in deep neural networks but is far less appropriate for other domains where the data are structured such as meshes, graphs or measurements on a manifold. In computer vision and machine learning, these problems that need deep learning models for structured data are studied under the topic called geometric deep learning [9], which has led to a number of elegant approaches including convolutional neural networks (CNN) on non-Euclidean data [14, 33]. The reason this is important is that mathematically, non-Euclidean data violates a number of key properties of Euclidean spaces such as a global linear structure and coordinate system, as well as shift invariance/equivariance. As a result, the core operations we use in classical statistics and machine learning as well as within deep neural network architectures often need to be tailored based on the geometry and specifics of the data at hand. When such adjustments are made in modern deep learning architectures, a number of authors have reported sizable improvements in the performance of the learning algorithms [11, 10, 34, 14, 27, 26, 15].

We should note that specializing learning methods to better respect or exploit the structure (or geometry) of the data are not a new development. Time series data are common in finance [49], and as a result, has been analyzed using specialized methods in statistics for decades. Surface normal vectors on the unit sphere have been widely used in graphics [48], and probability density functions, as well as covariance matrices, are common in both machine learning and computer vision [45, 16]. In neuroimaging, which is a key focus of our paper, the structured measurement at a voxel of an image may capture water diffusion [6, 53, 36, 31, 2, 13] or local structural change [25, 59, 32]. The latter example is commonly known as the Cauchy deformation tensor (CDT) [32] and has been utilized to achieve improvements over brain imaging methods such as tensor-based morphometry [37, 43, 4]. When the mathematical properties of such data are exploited, one often needs new loss functions and specialized optimization schemes. This step often involves first defining an intrinsic metric for the underlying geometry (structure) of the data. It is important to note that within geometric deep learning for *manifolds*, two types of settings are often considered. The **first** case is where the data are functions on a manifold. The **second** case corresponds to the setting where data are sample points on a manifold, such as a Riemannian manifold. In this paper, we study the second setting, which is not covered in the form described here in existing works including [9].

When the structure or geometry of the data informs the formulation of the learning task (or algorithm), we obtain differential geometry inspired algorithms where the role of the extrinsic or intrinsic metric induced by the data is explicit. Many datasets do *not* have a temporal or sequential component associated with each sample. However, the analysis of temporal (or sequential) data is an important area of machine learning and vision, e.g., within action recognition [1, 7, 50] and video segmentation [20], the study of analogous geometric ideas in this regime, especially within deep learning, is limited. Specifically, there are few existing proposals describing deep neural network models for structured (or manifold-valued) *sequential* data. Recently in [12], the authors proposed a *recurrent* model for the manifold of symmetric positive definite (SPD) matrices. This work is interesting and replaces a number of blocks within a recurrent model with the "statistical recurrent units". But it is known that training recurrent models is more involved than convolutional architectures – shortly, our experiments will show that a $2\times$ speed-up (by using a convolutional instead of a recurrent model) can be achieved. While the current consensus, within the community, is that sequential data should involve a recurrent network [17], as noted by [5], emerging results indicate that convolutional architectures often perform superior to recurrent networks on "sequential" applications such as audio synthesis. In fact, even historically, convolutional models were used for 1-D *sequential* data [22, 35]. Now, given that most use-cases of learning sequential models on manifold-valued data will *not* require the infinite memory capabilities offered by a recurrent model, it seems natural to investigate the extent to which convolutional models may suffice. Notice that in order to get the long effective memory from a CNN model, one needs to increase the depth and/or increase the receptive field: this is provided by extensions such as dilated convolutions. We find that the two key ingredients in [5] to achieve similar or better performance than a recurrent model for sequential tasks involves **(a)** using dilations to increase the receptive field of each convolution and **(b)** using residual connections to design a deeper but stable network. It seems logical that these developments should be an ideal starting point in designing models and algorithms for **sequential manifold-valued data** – the goal of this work. Our key **contribution** is the design of a Dilated CNN model for sequential manifold-valued data and showing its applicability in performing statistical analysis of brain images, specifically, diffusion-weighted MR images. To do so, we **(a)** define dilation for the convolution operator on the manifold of interest **(b)** define residual connections for our architecture **(c)** define weight normalization/dropout to add regularization/stability for the deeper network. We show that this yields an efficient formulation for sequential manifold-valued data, where few exist in the literature at this time. On the scientific side, we show that such a construction gives us the ability to identify structural connectivity changes in asymptomatic individuals who are at risk for developing Alzheimer's disease (AD) but are otherwise cognitively healthy.



Output
$d = 4$
Hidden layer
$d = 2$
Hidden layer
$d = 1$
Input
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
Dilated convolutional layers
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
Causal convolutional layers

Figure 1: Schematic diagram of dilated CNN and causal CNN (see [5] for definition and additional description).

2. Preliminaries

The motivation of this work is the analysis of sequential manifold-valued data, using deep architectures. As described above, our architecture utilizes ideas presented earlier in the context of dilated convolutional neural networks (DCNN) on Euclidean spaces [5]. To set up our formulation, we review the standard DCNN formulation and then describe our proposed manifold-valued DCNN framework.

Dilated Convolutions [5]: Given a 1-D input sequence $\mathbf{x} : \mathbf{N} \rightarrow \mathbf{R}^n$ and a kernel $w : \{0, \cdots, k-1\} \rightarrow \mathbf{R}$, the dilated convolution function $(\mathbf{x} \star_d w) : \mathbf{N} \rightarrow \mathbf{R}^n$ is:

$$(\mathbf{x} \star_d w)(s) = \sum_{i=0}^{k-1} w(i)\mathbf{x}(s - id), \tag{1}$$

where $\mathbf{N}$ is the set of natural numbers, and k and d are the kernel size and the dilation factor respectively. Notice that with $d = 1$, we get the normal convolution operator. In a dilated CNN, the receptive field size will depend on the depth of the network as well as on the choice of k and d. Thus, the authors in [5] suggested the use of *residual connections* [21] – this was found to provide stability for deeper networks. Notice that, unlike the standard residual network connection, here the authors used a 1×1 convolution layer in order to match the width of the input and the output. Additionally, in order to regularize the network, the authors used *weight normalization* [44] and *dropout* [46]. The weight normalization was applied to the kernel of the dilated convolution layer. The dropout was implemented by randomly zeroing out an entire output channel of a dilated convolution layer. Finally, as an activation function, the authors used ReLU non-linearity. A schematic diagram of a standard dilated CNN is given in Fig. 1.

Next, we discuss generalizing the operations needed within a DCNN so that they can operate on manifold-valued data. Specifically, we will generalize the following operations: **(1)** Dilated convolution **(2)** Residual connection **(3)** Weight Normalization **(4)** ReLU and **(5)** Dropout, to the setting where data are manifold-valued.

Recently in [11], the authors proposed a CNN architecture for manifolds and/or manifold-valued data. We can utilize some of these ideas towards deriving the dilated convolution operation. Before discussing the details of the definition of dilated CNN for manifold-valued data, we will first introduce some notations, concepts, and terminology.

Assumptions: We use $(\mathcal{M}, g)$ to denote a Riemannian manifold $\mathcal{M}$ with the Riemannian metric g and $d_{\mathcal{M}} : \mathcal{M} \times \mathcal{M} \to [0, \infty)$ denotes the distance induced by the metric g. We assume that the samples on $\mathcal{M}$ lie inside a regular geodesic ball of radius r centered at p, $\mathcal{B}_r(p)$, for some $p \in \mathcal{M}$ and $r = \min\{r_{\text{cvx}}(\mathcal{M}), r_{\text{inj}}(\mathcal{M})\}$. Here, r_{cvx} and r_{inj} are the convexity and injectivity radius of $\mathcal{M}$ [19].

Weighted Fréchet mean (wFM): Let $\{X_i\}_{i=1}^N$ be samples on $\mathcal{M}$. The authors in [11] define the convolution operation using the weighted Fréchet mean (wFM) [39] of $\{X_i\}$. Consider a one dimensional kernel $\{w(i)\}_{i=1}^N$ satisfying the convexity constraint, i.e., **(a)** $\forall i, w(i) > 0$ **(b)** $\sum_i w(i) = 1$. Then, the wFM (uniqueness is guaranteed by the statement above) is defined as:

$$\mathsf{wFM}(\{X_i\}, \{w\}) = \arg\min_M \sum_{i=1}^N w(i) d_{\mathcal{M}}^2(X_i, M), \quad (2)$$

Group of isometries: The set $I(\mathcal{M})$ of all isometries of $\mathcal{M}$ forms a group with respect to function composition. We will use G to denote this group and for $g \in G$, and $X \in \mathcal{M}$, let $g.X$ denote the result of applying the isometry g to point X ('.' simply denotes the group action).

Key Application focus: Diffusion-weighted imaging (DWI) is a magnetic resonance imaging (MRI) technique that measures the diffusion of water molecules to generate contrast in MRI, and has been widely applied to measure the loss of structural connectivity in the brain. At each voxel in the image, water diffusion can be variously represented: two common options are using an elliptical approximation (see Fig. 2(a)) where a 3×3 covariance matrix expresses the diffusivity properties or an orientation distribution function where one represents the probability densities of water diffusion over different orientations. One can divide the 3D image into anatomically meaningful parcels in Fig. 2(b) and then run standard tractography routines to estimate the strength of connectivity between each pair of anatomical parcels [42]. The fiber bundles, hence estimated, are shown in Fig. 2(c). For analysis, one often focuses on certain important fiber bundles instead of analyzing the full set of fibers. Notice that if we specify a starting and ending anatomical region for a fiber bundle, we can consider the corresponding covariance matrices encountered on this "path" as multi-variate manifold-valued measurements of this function. This is precisely the type of sequential manifold-valued data that we will seek to model in this paper.

3. Dilated convolutions for manifold-valued measurements

We now describe how to obtain the specific components needed in our architecture for manifold-valued data.

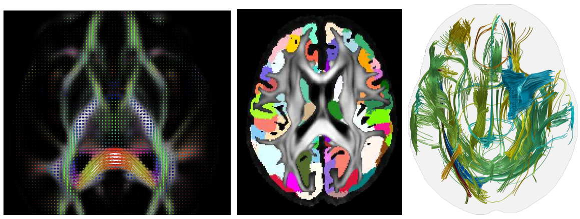


Figure 2: *(Left-Right)* (a) diffusion MRI, (b) Parcels, (c) Fiber bundles

Dilated convolution operator: Given a 1-D input sequence $X : \mathbf{N} \to \mathcal{M}$ and a kernel $w : \{0, \cdots, k-1\} \to \mathbf{R}$ satisfying the convexity constraint, the dilated convolution function $(X \star_d w) : \mathbf{N} \to \mathcal{M}$ is defined as:

$$(X \star_d w)(s) = \arg\min_M \sum_{i=0}^{k-1} w(i) d_{\mathcal{M}}^2(X(s-id), M), \quad (3)$$

where as before, k and d are the kernel size and dilation factor respectively. Observe that the convexity constraint on the kernel is merely to ensure that the result also lies on the manifold. We will use the weighted Fréchet mean (wFM) as a dilated convolution operator. This choice is mathematically justified because **(1)** Eq. (1) is the minimizer of the weighted variance which is wFM, if the choice of distance is the ℓ_2 distance. **(2)** We will show in Proposition (1) that the dilated convolution operator is equivariant to the action of G. This is a direct analog of its Euclidean counterpart. Notice that the dilated convolution operator defined in (1) is equivariant to translations, i.e., if $\mathbf{x}$ is translated by some amount $\mathbf{t}$, so is the result $(\mathbf{x} \star_d w)$. On the manifold $\mathcal{M}$, the analog of translation is the action of G, hence the equivariance of $(X \star_d w)$ with respect to G is a desirable property.

Proposition 1. *Using notations in (3) and given w satisfying the convexity constraint, let $F : X \mapsto (X \star_d w)$. Then, F is G-equivariant, i.e., F is equivariant to the action of G.*

Proof. Observe that, if $g \in G$ acts on X, then, $X(s-id) \mapsto g.X(s-id)$, for all s, d, i. Since g is an element of isometry group, therefore, $d_{\mathcal{M}}(g.X(s-id), g.M) = d_{\mathcal{M}}(X(s-id), M)$, for all $M \in \mathcal{M}$. So, $g.M = (g.X \star_d w)(s)$ iff $M = (X \star_d w)(s)$, which concludes our proof. □

In (3), since $(X \star_d w)$ is a $\mathcal{M}$ valued function, we will use M as a manifold-valued function, i.e., $M(s) = (X \star_d w)(s)$. Similar to the Euclidean dilated convolution layer, we learn multiple dilated kernels (given by the number of output channels) for a dilated convolutional layer.

Residual connection: Let X and F be the input and output of a dilated convolutional layer where the numbers of channels are c_{in} and c_{out}. Then, analogous to the Euclidean residual connection, we define the residual connection using two steps: **(a)** First, concatenate X and $F(X)$ to get $(c_{in} + c_{out})$ number of channels. **(b)** Use wFM to extract c_{out} number of outputs. More formally, let $R(X, F_X)$ be the output of the residual connection, then the k^{th} channel of the residual connection, $R_k(X, F(X))$ is given by:

$$R_k(X,F(X))(s) \overset{def}{\equiv} \arg\min_{M} \left(\sum_{i=1}^{c_{in}} w_k(i) d^2_{\mathcal{M}}(X_i(s), M) + \sum_{j=1}^{c_{out}} w_k(j + c_{in}) d^2_{\mathcal{M}}(F_j(s), M) \right),$$
$$\text{s.t.} \sum\nolimits_i w_k(i) = 1, \forall w_k(i) > 0, \qquad (4)$$

where, $k \in \{1, \cdots, c_{out}\}$ and X_i and F_j denotes the i^{th} and j^{th} channel of X and F respectively.

Weight normalization, ReLU, and Dropout: The weight normalization in the standard Euclidean convolutional network is not needed here since we impose a convexity constraint on the kernel. We argue that since Dropout is a regularizer, we will not use dropout for our manifold-valued DCNN implementation because of the implicit regularization due to the convexity constraint. As argued in [11], wFM is both **(a)** a contraction mapping [11] and **(b)** a nonlinear mapping and hence ReLU or any other non-linearity is not strictly necessary. Here, similar reasoning explains why a ReLU is not needed (since the contraction and non-linear mapping are provided directly by wFM).

Equivariance and Invariance: A few reasons why convolutional networks are so powerful are **(a)** translational equivariance of a convolution layer and so, weights can be shared across an image **(b)** translational invariance property of the entire convolutional network which is the property of the fully connected last layer. As we showed above, the way we defined our dilated convolution operator leads to equivariance to the action of G. But we still have not shown that the *last layer* can be designed in a way that the output of the network does not change with respect to the action of G. So, we still need an analogous G-invariant last layer.

Invariant last layer: Analogous to the Euclidean recurrent model/ dilated CNN, in the last layer we will only consider the output of the last time point of a sequence, i.e., if X is the output of the last dilated convolutional layer with c number of channels, then the input of our last layer is $\{X_i(N)\}_{i=1}^{c}$, where $X(N) \in \mathcal{M}$ is the value of the last time point. We know already that $\{X_i(N)\}$ are G-equivariant. So, in order to make the entire dilated convolutional network G invariant, we need an invariant last layer. This is analogous to the translational invariant property of a fully connected (FC) layer in the traditional (Euclidean) dilated CNN. We design our last invariant layer as follows: **(a)** We will first learn nC number of wFM (let denoted by $\{\mu_i\}_{i=1}^{nC}$) of $\{X_i(N)\}_{i=1}^{c}$ using (2), where nC is a hyperparameter. **(b)** For all $i \in \{1, \cdots, c\}$, and for all $j \in \{1, \cdots, nC\}$, we compute the distance between $X_i(N)$ and μ_j, denoted by d_{ij}. **(c)** Thus, for each $X_i(N)$, we get nC number of feature representations. **(d)** We will use a standard fully connected (FC) layer with $c \times nC$ features as input and the desired number of outputs.

Proposition 2. *The last layer is G-invariant.*

Algorithm 1: A basic i^{th} DCNN building block with two convolution layers

```
function DCNN
VARIABLES($N, c^1_{in}, c^1_{out}, c^2_{out}, c_{res}, k_1, d_1, k_2, d_2, nC, c$)
    $x^{i-1} \leftarrow \mathrm{Input}(c^1_{in}, N)$
    $y_1 \leftarrow \mathrm{Dilated\_Conv}(x^{i-1}, c^1_{in}, c^1_{out}, k_1, d_1)$
    $y_1 \leftarrow \mathrm{Dilated\_Conv}(y_1, c^1_{out}, c^2_{out}, k_2, d_2)$
    $x^i \leftarrow \mathrm{Residual}(x^{i-1}, y_1, c^1_{in}, c^2_{out}, c_{res})$
    $y_o \leftarrow \mathrm{Inv}(x^i, nC, c)$ (For last DCNN block)
end function
```

Proof. Observe that $d_{ij} = d_{\mathcal{M}}(X_i(N), \mu_j)$. From Proposition 1, we know that μ_j is G-equivariant, hence, $\mu_j \mapsto g.\mu_j$, for some $g \in G$ if $\forall i, X_i(N) \mapsto g.X_i(N)$. But, $d_{\mathcal{M}}(X_i(N), \mu_j) = d_{\mathcal{M}}(g.X_i(N), g.\mu_j)$, which concludes the proof. □

In order to reduce the number of parameters in the last layer, we propose a parameter efficient last layer which is defined as using a FC layer on the tangent space, i.e., input $\{\mathsf{Log}(X_i(N))\}_{i=1}^{c}$ as input to the FC layer, where Log is the Riemannian inverse exponential map.

Now, we have all components of our dilated CNN on manifold-valued data. A schematic of our model is shown in Fig. 3. The building block for a 2-layer manifold DCNN is shown in Alg. 1. Note that the network parameters are scalar-valued, with a convexity constraint. In order to enforce the convexity constraint, i.e., $\{w(i)\} \geq 0$ and $\sum_i w(i) = 1$, we will learn $\left\{\sqrt{w(i)}\right\}$, which can be any real value. We will enforce the sum constraint by normalization. Thus we will use SGD to learn $\left\{\sqrt{w(i)}\right\}$.

4. Experiments

In this section, we apply the manifold DCNN to answer the following questions: **(1)** By replacing a RNN with our

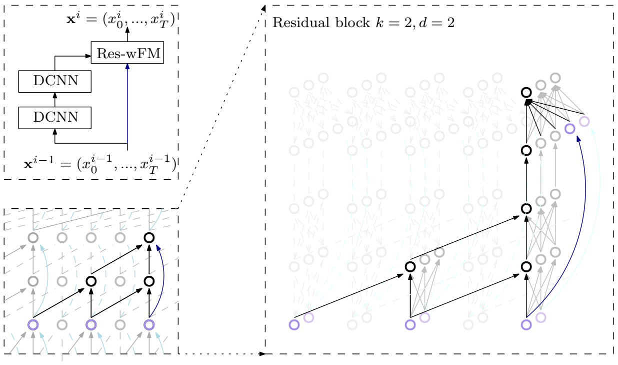

$\mathbf{x}^i = (x^i_0, ..., x^i_T)$
Res-wFM
DCNN
DCNN
$\mathbf{x}^{i-1} = (x^{i-1}_0, ..., x^{i-1}_T)$
Residual block $k = 2, d = 2$

Figure 3: Schematic diagram of the residual block of manifold DCNN. There're two DCNN blocks and one residual connection in one block. wFM is used to extract the $c_{out} = 3$ channels from the concatenation.

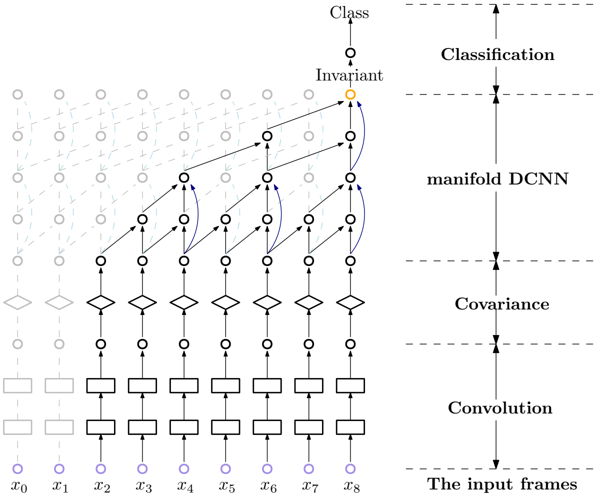


Figure 4: Schematic diagram of the network architecture for vision datasets. We use two CNN to extract the features. And we calculate the covariance between feature channels to get the SPD matrices. In the last layer, we use G-invariant and a fully connected layer to do the classification.

DCNN with a manifold constraint, what improvement in terms of the number of parameters/time can we achieve, without sacrificing performance? **(2)** For computer vision applications, how much improvement can we get? **(3)** When using our method for scientific analysis of neuroimaging data, can we obtain promising results that show that such models can enable discoveries beyond current capabilities?

Next, we will answer the questions above by analyzing the comparative performance of manifold DCNN via four experiments: **(1)** two computer vision applications of classifying videos and **(2)** two neuroimaging experiments for scientific discoveries related to Alzheimer's disease.

4.1. Improvement in terms of parameters/time on synthetic and real computer vision datasets

In this section, we organize two sets of experiments: **(1)** Classification of different moving patterns on the Moving MNIST data **(2)** Classification of 11 actions on the UCF-11 data. Both these experiments serve as empirical evidence of the efficiency of manifold DCNN in terms of the number of parameters and time per epoch. We compared our method with five state-of-the-art sequential models: SPD-SRU [12], LSTM [24], SRU [40], TT-GRU and TT-LSTM [54]. For all methods except TT-GRU and TT-LSTM, before the sequence process module, we used a convolution block. For manifold DCNN and SPD-SRU (also for manifold-valued data), between the convolution block and the sequence process unit, we include a covariance block analogous to [58]. The architecture of this experiment is shown in Fig. 4.

As one of the key operations of DCNN is wFM, below we will use an efficient recursive provably consistent estimator of wFM on the space of covariance matrices (SPD with some added small noise along diagonal). Let $X(s)$ be an SPD matrix for all $s \in \mathbf{N}$, and then the n^{th} recursive wFM estimator, M_n is given as:

$$M_0 = X(s) \qquad M_n = \Gamma_{M_{n-1}}^{X(s-n*d)}\left(\frac{w(n)}{\sum_{j=0}^{n} w(j)}\right), \quad (5)$$

where Γ is the shortest geodesic on the manifold of SPD matrices equipped with the canonical affine invariant Riemannian metric [23].

4.1.1 Moving MNIST: Moving pattern classification

We generated the Moving MNIST data according to the algorithm proposed in [47]. In this experiment, we classify the moving patterns of different digits. For each moving pattern, we generated 1000 sequences with length 20 showing 2 digits moving in the same pattern in a 64×64 frame. The moving speed and the direction are fixed inside each class, but the digits are chosen randomly. In this experiment, the difference in the moving angle from two sequences across different classes is at least $5°$.

Results: In Table 1, the results show that our method not only achieves the best test accuracy with the smallest number of parameters but is also 1.5 times faster than the SPD-SRU which has the second smallest # of parameters. The kernel of CNN we use has size 5×5 with the input channel and output channel set to 5 and 10 respectively. All parameters are chosen in a way to use the fewest number of parameters without deteriorating the test accuracy.

Scalability: We assess the running time (training and testing) of manifold DCNN with respect to the SPD matrix size. From Fig. 5(a), we can see that as the matrix size increases, the training time increases, while the testing time remains almost the same. This is a desirable property as it indicates that inference time does not depend on matrix size. Also, for different orientations differences, manifold DCNN gives almost perfect classification accuracy with very small standard deviation, as shown in Fig. 5(b).

4.1.2 UCF-11: Action classification

The UCF-11 dataset [38] contains 1600 video clips of 11 different classes, such as basketball shooting, diving, etc. The video lengths (frame sequences) vary from 204 to 1492, with the resolution of each frame being 320×240. We sample every 3 frames, resize each frame to 160×120, and clip the frame sequences to have the length of 50. For our method, we chose two convolution layers with kernels 7×7 and output channels 4 and 6 before the DCNN block. Hence, the dimension of the covariance matrices is 7×7. For the manifold DCNN block, we use three residual blocks, with

Model	**# params.**	**time (s) / epoch**	**Test acc.**		
			30°versus 60°	10°versus 15°	10°versus 15°versus 20°
DCNN	$\mathbf{1517}$	~ 4.3	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.01}$	$\mathbf{0.95 \pm 0.01}$
SPD-SRU	1559	~ 6.2	$\mathbf{1.00 \pm 0.00}$	0.96 ± 0.02	0.94 ± 0.02
TT-GRU	2240	$\sim \mathbf{2.0}$	$\mathbf{1.00 \pm 0.00}$	0.52 ± 0.04	0.47 ± 0.03
TT-LSTM	2304	$\sim \mathbf{2.0}$	$\mathbf{1.00 \pm 0.00}$	0.51 ± 0.04	0.37 ± 0.02
SRU	159862	~ 3.5	$\mathbf{1.00 \pm 0.00}$	0.75 ± 0.19	0.73 ± 0.14
LSTM	252342	~ 4.5	0.97 ± 0.01	0.71 ± 0.07	0.57 ± 0.13

Table 1: Comparative results on Moving MNIST. Our model achieves the highest accuracy (in blue) with the least # of parameters in all setups.

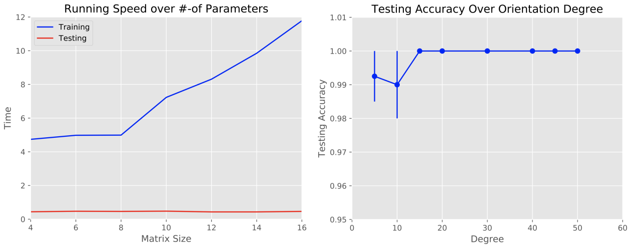


Figure 5: Left: time versus matrix size. As the matrix size increases, the training time inevitably increases but the testing time consistently remains extremely small. Right: accuracy versus degree difference of orientation in the dataset. Beyond the degree difference as small as 15°, the error bar becomes negligible implying our model quickly becomes very robust.

channels set to be $[1, 3, 3]$; $[3, 3, 4]$ and $[4, 4, 4]$ respectively. The kernel size is 5 for each residual block with the initial dilation number being 1 (if not specified, the initial dilated number is always 1 in this paper.). For TT-GRU and TT-LSTM, we follow the same setting as given in [54]. For SPD-SRU, SRU, and LSTM, we use the same parameters as in [12]. All models achieve $> 90\%$ training accuracy.

Results: Test accuracy with the number of parameters and time per epoch is shown in Table 2. We can see the number of parameters for our method is comparable with SPD-SRU with higher test accuracy ($\approx 4\%$ improvement) and much faster runtime ($\approx 2.5\times$). Note that without residual connections, the accuracy drops to 0.809 ± 0.044: in other words, residual connections are useful.

Take-home message: *With the above two experiments, we can conclude that manifold DCNN* ***(i)*** *is faster,* ***(ii)*** *uses fewer parameters and* ***(iii)*** *gives better or comparable classification accuracy compared to the state-of-the-art.*

4.2. Group effects in Preclinical Alzheimer's disease

Cardinal features of Alzheimer's disease (AD) include the development of beta-amyloid plaques (amyloid), neurofibrillary tangles (tau), and progressive neurodegeneration (characterized by MRI) [30]. Autopsy studies among individuals with AD dementia indicate that degeneration of myelinated axons in the context of amyloid and tau pathology is a defining feature of dementia status [41]. Techniques for measuring axonal degeneration in vivo include analysis of cerebrospinal fluid, as well as diffusion-weighted imaging; however, few studies have tested the extent to which early amyloid accumulation may be associated with neural injury. Our goal is to utilize our method to **identify white matter fiber bundles that are affected *early* in the preclinical disease process**. Positron emission tomography (PET) imaging with Pittsburgh compound B (PiB), which identifies amyloid deposition, can be used as an indicator of AD pathology [28]. Thus, we compared healthy individuals who were positive for AD pathology (PiB+) to healthy individuals who were negative for pathology (PiB-). Additionally, we compared individuals who carried a risk gene for AD (APOE+) to non-carriers (APOE-).

Model	# params.	time (s)/ epoch	Test acc.
manifold DCNN	3393	$\sim$ **33**	**0.823 ± 0.018**
SPD-SRU	**3337**	$\sim$ 76	0.784 ± 0.014
TT-GRU	6048	$\sim$ 42	0.78
TT-LSTM	6176	$\sim$ **33**	0.78
SRU	2535630	$\sim$ 50	0.75
LSTM	14626425	$\sim$ 57	0.70

Table 2: Comparative results on UCF-11 data. Our model achieves the best accuracy and the fastest speed with a small number of parameters.

4.2.1 Diffusion-weighted imaging (DWI)

Data acquisition: Diffusion-weighted imaging was completed on a General Electric (GE) 3 Tesla scanner with a 32-channel head coil and a spin-echo echo-planar imaging pulse sequence among participants who are asymptomatic. Multi-shell DWI data were collected using b-values $b = 0$, $b = 500$, $b = 800$, $b = 2000$, with $2 \times 2 \times 2mm$ resolution. The signal was corrected using MRTrix3[51] and FSL's 'eddy'[3]. Diffusion tensor imaging (DTI) and the orientation distribution functions (ODF), which were used as the representative of the DWI, were performed using the Diffusion Imaging in Python (DIPY) toolbox[18]. To generate fiber bundles of interest, the data was processed using TRACULA[56, 55, 57]. With this pipeline, we generated 18 major fiber bundles [52], as shown in Fig. 2(c). Regions of interest (ROI) in the template space, were inversely warped back to the subject space to generate the fiber bundles and each data point used in the analysis for each participant.

Analysis: From the previous experiments, we can see that manifold DCNN performs well on classification problems with faster computation speed and fewer parameters. Due to the fast runtime *and* the small number of parameters, we can use permutation testing to perform group analysis. The statistical testing is performed on each fiber bundle between the two groups, to determine if the DCNN model between the two groups is different. To summarize, the setup is: **(1)** Group 1 (PiB+) versus Group 2 (PiB-), **(2)** Group 1 (APOE+) versus Group 2 (APOE-). Now, we will give some details of the DCNN models for DTI and ODF representations before the statistical analysis.

(i) Diffusion tensor imaging (DTI): Diffusion tensor imaging (DTI) is a method to represent the Diffusion imaging with SPD matrices. Since all of the data samples lie on the SPD manifold, the model is similar to the classification model above. The only difference between classification model and this group analysis model is that instead of the prediction of the classes, we are fitting the two groups of data into two trainable models, θ_1 and θ_2 and assessing if the distributions of θ_1 and θ_2 are statistically different.

(ii) Orientation distribution function (ODF): Orientation distribution function (ODF) represents the probability densities of water diffusion over different orientations. In order to perform the statistical analysis, we discretized the space of orientations, i.e., $\mathbf{S}^2$. We sampled 724 equally spaced points on the sphere $\mathbf{S}^2$ to represent the ODF. Let the ODF be denoted by $\mathbf{x}_t$, then after the discretization, we have $\sum_{i=1}^{724} \mathbf{x}_t^i = 1$. As ODF is a probability density function, we use square root parameterization [8, 45] to represent ODF. Using the square root parameterization, we map $\mathbf{x}_t$ onto the positive orthant of the unit hypersphere of dimension 723, i.e., $\mathbf{S}^{723}$. As in Section 3, a key component of DCNN is the definition of wFM, which we can define on $\mathbf{S}^n$:

$$\begin{aligned}\mathbf{y}(s) &= \mathsf{wFM}\left(\{w(i)\}, \{\mathbf{x}(s - d*(k-1) : d : s)\}\right) \\ &= \arg\min_M \sum_{i=0}^{k-1} w(i) d_{\mathbf{S}}^2\left(\mathbf{x}(s - d*i), M\right),\end{aligned} \quad (6)$$

Here $d_{\mathbf{S}}$ is the rotation invariant geodesic distance on $\mathbf{S}^{723}$ and $\mathbf{x}(s)$ is a sample on $\mathbf{S}^{723}$ for $s \in \mathbf{N}$. Analogous to the SPD manifold, we can define a recursive wFM estimator $\mathbf{m}_n$:

$$\mathbf{m}_0 = \mathbf{x}(s) \qquad \mathbf{m}_n = \Gamma_{\mathbf{m}_{n-1}}^{\mathbf{x}(s-n*d)}\left(\frac{w(n)}{\sum_{j=0}^{n} w(j)}\right), \quad (7)$$

where Γ is the shortest geodesic on $\mathbf{S}^{723}$. Using the above-defined estimator of wFM, we can define DCNN on $\mathbf{S}^{723}$ as in Section 3. **Note**: Our baseline model, SPD-SRU cannot deal with the S^n manifold as we do here.

4.2.2 Statistical analysis: permutation testing

Suppose we train our model for each of the two groups for each fiber bundle ^{fb} we have, with parameters θ_1^{fb} and θ_2^{fb}. Our goal is to test whether the fiber bundle ^{fb} is statistically different between the two groups. Thus, we model the manifold-valued data and perform statistical analysis in the parameters space. Since the models for each group lie in the same parameter space, the statistical analysis can be performed in the parameter space by bootstrapping. We can measure the distance between two models as $\sigma^{fb} = ||\theta_1^{fb} - \theta_2^{fb}||$ to represent the distance between the group-wise fitted models' distributions in parameter space. Then, we need to evaluate how statistically significant the distance is – and if the value is large enough, it is unlikely to happen by chance. A simple way to perform the test for statistical significance is via permutation testing. If we randomly shuffle (via a random permutation) the group information for all our samples (i.e., subjects) and run our model for both "random" groups, we will get new parameters $\hat{\theta}_1^{fb}$ and $\hat{\theta}_2^{fb}$. We define $\hat{\sigma}^{fb} = ||\hat{\theta}_1^{fb} - \hat{\theta}_2^{fb}||$ as a random variable. After permuting 5000 times, we can estimate the distribution of the $\hat{\sigma}^{fb}$ – this is the Null distribution (See Fig. 6 as examples). The p-value is defined as the ranking of the σ^{fb} among the distribution of the $\hat{\sigma}^{fb}$. If the p-value is less than the significance threshold $\alpha = 0.05$, we can conclude that this is **not** likely to happen by chance.

Since the length of different fiber bundles varies from 11 to 73, we construct the DCNN to have 3 layers of residual units, with channels being $1, 3, 3$; $3, 3, 5$ and $5, 8, 10$ respectively. And the 1-D kernel size is 3. We use all the data we have to pre-train the model. After pre-training, we fine tune the model during the permutation testing.

Experiments		PiB			APOE	
	Total	Positive	Negative	Total	Positive	Negative
Number	196	29	167	669	247	422
Age (years) (mean (SD))	62.40 (6.33)	66.29 (4.95)	61.75 (6.30)	65.61 (8.68)	64.55 (7.99)	66.23 (9.00)
Sex (female; %)	134 (68%)	21 (72%)	113 (68%)	426 (64%)	159 (64%)	267 (63%)

Table 3: Description of data/participant demographics used in the study.

4.2.3 Result 1: Group analysis: PiB+ versus PiB-

The study included imaging data acquired from 196 cognitively unimpaired (healthy) participants acquired in a local cohort at the University of Wisconsin. We provide demographic information from participants with PiB and APOE measures in Table 3. Initial analyses were run using single-shell data, where the model was run on all 18 fiber bundles, one by one, with the parameters mentioned above. We performed permutation tests for each fiber bundle individually.

Results for the 18 fibers are shown in Table 4 (column 2). We find that two of the 18 fibers satisfied the threshold of 0.05, which means that statistically these fiber bundles are different across the two groups. Since the sample sizes were small, the results presented are uncorrected p-values (multiple testing correction was not performed).

Fiber bundles evaluated in this analysis included those which are known **to be affected** in AD, including the superior longitudinal fasciculus and cingulum bundle, as well as control tracts that are **not likely to be affected** by AD, such as the corticospinal tract. We found significant differences between PiB+ and PiB- groups in fiber bundles that are likely to be affected by AD, including the superior longitudinal fasciculus and Corpus callosum - forceps minor.

When compared with the SPD-SRU model, which also reported brain imaging experiments in their paper, the results show only one out of 18 fibers survives. And also, we find that our model runs **much faster** (about 5×), which is very important when running permutation testing thousands of times. It takes 3.5 days to run permutation testing 5000 times using DCNN, while the SPD-SRU takes 18 days. When we keep the number of GPUs fixed, the difference between 3.5 and 18 will be even more sizable if we expand the number of permutation testing to 10000 or more.

4.2.4 Result 2: Group analysis: APOE+ versus APOE-

The APOE analysis was performed using data from 669 subjects with APOE information, with 247 of them being positive for APOE4 (a risk factor for AD). Analyses were also conducted using the multi-shell dMRI to generate ODF information. Similar to the preceding group difference analysis, the model was run on all 18 fiber bundles with the parameters described previously on both DTI and ODF.

The results for 18 fibers are shown in Table 4 in column 3. It is noteworthy that SPD-SRU can only deal with the SPD manifold. So for ODF, which lies on S^n, we can *only* run our DCNN model to do the group analysis.

Here, we found that four of the 18 fiber bundles met the significance threshold of 0.05 with DTI, while SPD-SRU only captured one. Five fiber bundles were identified when

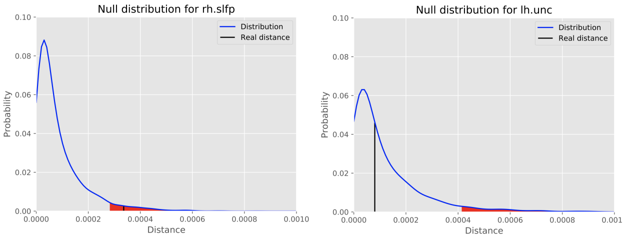


Figure 6: The Null distribution for one fiber bundle with $\alpha = 0.05$. If the real distance (black line) lies in the threshold (red area), that test is believed to not happen by chance.

using ODF. We found differences by APOE genotype in the forceps minor, cingulum projecting to parietal cortex, anterior thalamic projections, superior longitudinal fasciculus projecting to parietal cortex and inferior longitudinal fasciculus. We did not find differences in fiber bundles unlikely to be affected by AD, such as the corticospinal tract in both experiments. Fiber bundles that were consistently identified in both the DTI and ODF analyses included the inferior longitudinal fasciculus and the anterior thalamic projections.

4.2.5 Discussion of preclinical AD analysis results

While amyloid and tau pathology are defining features of AD, methods are also needed to detect AD-associated neurodegeneration [29]. Neurodegeneration may signal future cognitive decline. However, methods for detecting early and subtle neurodegeneration, particularly of myelinated axons, are not yet available, especially in preclinical AD. This is why our results here seem promising.

The results suggest significant differences in underlying fiber bundle microstructure among individuals who meet biological criteria for AD (based on PiB status) as well as differences by APOE genotype. Of note, our algorithm identified significant differences in the cingulum bundle by PiB status; this white matter fiber bundle connects medial temporal lobe and parietal cortices as part of a memory network that is impacted by AD, and is vulnerable to degeneration in the early stages of AD. Differences in the cingulum bundle were also apparent among carriers of the APOE4 allele, a genetic risk factor for sporadic AD. Likewise, superior longitudinal fasciculus differed by AD biomarker status and APOE genotype. Projections identified as being significantly different included fiber bundles projecting to parietal cortices. Parietal cortices are significantly impacted by AD pathology and are among the first to show amyloid accumulation. The results presented here may suggest that amyloid accumulation negatively impacts adjacent white matter fiber bundles. It may also be possible that degeneration of fiber bundles is a function of AD pathology spreading to anatomically linked brain regions via white matter fiber bundles, although further longitudinal evaluation is needed to test the hypothesis. In summary, statistical analysis enabled by our proposed algorithm was capable of identifying differences in biologically meaningful brain regions.

	p-value				
Fiber Name	**Experiment 1**		**Experiment 2**		
	PiB+ versus PiB-		APOE+ versus APOE-		
	DCNN on DTI	SPD-SRU	DCNN on DTI	SPD-SRU	DCNN on ODF
fmajor_PP	0.443	0.923	0.207	0.600	0.778
fminor_PP	**0.008**	0.158	**0.035**	**0.025**	N/A
lh.atr_PP	0.323	0.632	0.30	0.991	**0.028**
rh.atr_PP	0.295	0.143	0.86	0.271	0.563
lh.cab_PP	0.276	0.363	0.76	0.644	0.500
rh.cab_PP	0.311	0.263	0.78	0.848	0.444
lh.ccg_PP	0.230	0.267	**0.042**	0.609	**0.043**
rh.ccg_PP	0.093	0.087	**0.048**	0.532	**0.048**
lh.cst_AS	0.561	0.143	0.58	0.350	0.800
rh.cst_AS	0.629	0.278	0.35	0.667	0.769
lh.ilf_AS	0.309	0.895	0.47	0.977	**0.042**
rh.ilf_AS	0.405	0.889	0.46	0.563	0.857
lh.slfp_PP	0.482	0.615	0.68	0.107	0.192
rh.slfp_PP	0.571	0.941	**0.047**	0.154	**0.050**
lh.slft_PP	**0.005**	**0.041**	0.92	0.649	0.556
rh.slft_PP	0.790	0.462	0.53	0.947	0.333
lh.unc_AS	0.623	0.158	0.23	0.860	0.933
rh.unc_AS	0.298	0.895	0.34	0.324	0.182

[*] N/A: This ODF fiber bundle did not pass Quality Check (QC) after pre-processing. Therefore, we left it out of the analysis to avoid inconsistencies in the parameters used for pre-processing the full set of fiber bundles.

Table 4: p-values (uncorrected) for all fibers in different groups. The highlights are the fiber bundles that satisfy the significance threshold. Runtime for DCNN is $5\times$ times faster than SPD-SRU (not included here).

Take-home message: *Our DCNN model was able to capture more fiber differences with significant effects compared to the SPD-SRU. It is also noteworthy that our model is much more efficient: only* $60s$ *for one realization of the permutation test* ($\times\#$ *of realizations), while the SPD-SRU model* $> 5\times$ *times slower. Compared with the SPD-SRU, which can only handle DTI (SPD), our method is more general: handles both DTI (SPD) and ODF (*$\mathbf{S}^n$*) data.*

5. Conclusions

We present a new Dilated CNN formulation to model sequential and spatio-temporal manifold data, where few alternatives are available. Compared with the standard sequential model (RNN), our method can improve the performance when evaluated on the number of parameters and runtime. We show that when using wFM, Weight normalization, ReLU, and Dropout are no longer needed in this formulation. On the experimental side, for video analysis, we show that improvements can be obtained with fewer parameters and shorter running time. Importantly, we show that our algorithmic contributions facilitate scientific discovery relevant to AD, and may facilitate early disease detection at the preclinical stage. The analysis enabled by our formulation revealed subtle neurodegeneration of white matter fiber bundles affected by AD pathology, in brain regions implicated in prior studies of AD. The code is available at https://github.com/zhenxingjian/DCNN.

Acknowledgments

This research was supported in part by grants R01EB022883, R01AG059312, RF1AG027161 (WRAP study), R01AG021155, P50AG033514 (Wisconsin ADRC), R01 AG037639, P30 AG062715, R01 AG059312, RF1 AG027161, and NSF CAREER award RI 1252725.

References

[1] Bijan Afsari, Rizwan Chaudhry, Avinash Ravichandran, and René Vidal. Group action induced distances for averaging and clustering linear dynamical systems with applications to the analysis of dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2208–2215. IEEE, 2012. 2

[2] Iman Aganj, Christophe Lenglet, and Guillermo Sapiro. ODF reconstruction in q-ball imaging with solid angle consideration. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 1398–1401. IEEE, 2009. 1

[3] Jesper LR Andersson and Stamatios N Sotiropoulos. An integrated approach to correction for off-resonance effects and subject movement in diffusion mr imaging. *Neuroimage*, 125:1063–1078, 2016. 6

[4] J Ashburner and K Friston. Morphometry. 2004. 1

[5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Convolutional sequence modeling revisited. 2018. 2

[6] Peter J Basser, James Mattiello, and Denis LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, 1994. 1

[7] Alessandro Bissacco, Alessandro Chiuso, Yi Ma, and Stefano Soatto. Recognition of human gaits. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001. 2

[8] Dorje C Brody and Lane P Hughston. Statistical geometry in quantum mechanics. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 2445–2475. The Royal Society, 1998. 7

[9] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1

[10] Rudrasis Chakraborty, Monami Banerjee, and Baba C Vemuri. H-cnns: Convolutional neural networks for riemannian homogeneous spaces. *arXiv preprint arXiv:1805.05487*, 2018. 1

[11] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep network framework for manifold-valued data. *arXiv preprint arXiv:1809.06211*, 2018. 1, 2, 3, 4

[12] Rudrasis Chakraborty, Chun-Hao Yang, Xingjian Zhen, Monami Banerjee, Derek Archer, David Vaillancourt, Vikas Singh, and Baba C Vemuri. Statistical recurrent models on manifold valued data. *arXiv preprint arXiv:1805.11204*, 2018. 2, 5, 6

[13] Guang Cheng, Hesamoddin Salehian, and Baba C Vemuri. Efficient recursive algorithms for computing the mean diffusion tensor and applications to DTI segmentation. In *European Conference on Computer Vision*, pages 390–401. Springer, 2012. 1

[14] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 1

[15] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016. 1

[16] Francesca Dominici, Aidan McDermott, Scott L Zeger, and Jonathan M Samet. On the use of generalized additive models in time-series studies of air pollution and health. *American journal of epidemiology*, 156(3):193–203, 2002. 1

[17] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 2

[18] Eleftherios Garyfallidis, Matthew Brett, Bagrat Amirbekian, Ariel Rokem, Stefan Van Der Walt, Maxime Descoteaux, and Ian Nimmo-Smith. Dipy, a library for the analysis of diffusion mri data. *Frontiers in neuroinformatics*, 8:8, 2014. 6

[19] David Groisser. Newton's method, zeroes of vector fields, and the riemannian center of mass. *Advances in Applied Mathematics*, 33(1):95–135, 2004. 3

[20] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575. IEEE, 2012. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[22] Geoffrey E Hinton. Connectionist learning procedures. In *Machine Learning, Volume III*, pages 555–610. Elsevier, 1990. 2

[23] Jeffrey Ho, Guang Cheng, Hesamoddin Salehian, and Baba C Vemuri. Recursive karcher expectation estimators and geometric law of large numbers. In *AISTATS*, pages 325–332, 2013. 5

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 5

[25] Xue Hua, Alex D Leow, Neelroop Parikshak, Suh Lee, Ming-Chang Chiang, Arthur W Toga, Clifford R Jack Jr, Michael W Weiner, Paul M Thompson, Alzheimer's Disease Neuroimaging Initiative, et al. Tensor-based morphometry as a neuroimaging biomarker for alzheimer's disease: an MRI study of 676 AD, MCI, and normal subjects. *NeuroImage*, 43(3):458–469, 2008. 1

[26] Zhiwu Huang and Luc J Van Gool. A riemannian network for spd matrix learning. In *AAAI*, volume 1, page 3, 2017. 1

[27] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. *arXiv preprint arXiv:1611.05742*, 2016. 1

[28] Milos D Ikonomovic, William E Klunk, Eric E Abrahamson, Chester A Mathis, Julie C Price, Nicholas D Tsopelas, Brian J Lopresti, Scott Ziolko, Wenzhu Bi, William R Paljug, et al. Post-mortem correlates of in vivo pib-pet amyloid imaging in a typical case of alzheimer's disease. *Brain*, 131(6):1630–1645, 2008. 6

[29] Clifford R Jack, David A Bennett, Kaj Blennow, Maria C Carrillo, Billy Dunn, Samantha Budd Haeberlein, David M Holtzman, William Jagust, Frank Jessen, Jason Karlawish, et al. Nia-aa research framework: Toward a biological definition of alzheimer's disease. *Alzheimer's & Dementia*, 14(4):535–562, 2018. 8

[30] Clifford R Jack Jr, Kaj Blennow, C Maria, Billy Dunn, Cerise Elliott, Samantha Budd Haeberlein, David Holtzman, William

Jagust, Frank Jessen, Jason Karlawish, et al. 2018 nia-aa research framework to investigate the alzheimer's disease continuum. 6

[31] Bing Jian, Baba C Vemuri, Evren Özarslan, Paul R Carney, and Thomas H Mareci. A novel tensor distribution model for the diffusion-weighted MR signal. *NeuroImage*, 37(1):164–176, 2007. 1

[32] Hyunwoo J Kim, Nagesh Adluru, Heemanshu Suri, Baba C Vemuri, Sterling C Johnson, and Vikas Singh. Riemannian nonlinear mixed effects models: Analyzing longitudinal deformations in neuroimaging. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 1

[33] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018. 1

[34] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *arXiv preprint arXiv:1802.03690*, 2018. 1

[35] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 2

[36] C. Lenglet, M. Rousson, and R. Deriche. DTI segmentation by statistical surface evolution. *IEEE Trans. on Medical Imaging*, 25(6):685–700, 2006. 1

[37] Natasha Lepore, Caroline A Brun, Ming-Chang Chiang, Yi-Yu Chou, Rebecca A Dutton, Kiralee M Hayashi, Oscar L Lopez, Howard J Aizenstein, Arthur W Toga, James T Becker, et al. Multivariate statistics of the jacobian matrices in tensor based morphometry and their application to hiv/aids. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pages 191–198. Springer, 2006. 1

[38] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*, pages 1996–2003. IEEE, 2009. 5

[39] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'I. H. P.*,, 10(4):215–310, 1948. 3

[40] Junier B Oliva, Barnabás Póczos, and Jeff Schneider. The statistical recurrent unit. *arXiv preprint arXiv:1703.00381*, 2017. 5

[41] Beatriz G Perez-Nievas, Thor D Stein, Hwan-Ching Tai, Oriol Dols-Icardo, Thomas C Scotton, Isabel Barroeta-Espar, Leticia Fernandez-Carballo, Estibaliz Lopez De Munain, Jesus Perez, Marta Marquie, et al. Dissecting phenotypic traits linked to human resilience to alzheimer?s pathology. *Brain*, 136(8):2510–2526, 2013. 6

[42] Pradeep Reddy Raamana and Stephen C Strother. graynet: single-subject morphometric networks for neuroscience connectivity applications. *J. Open Source Software*, 3(30):924, 2018. 3

[43] Gerard Robert Ridgway. *Statistical analysis for longitudinal MR imaging of dementia*. PhD thesis, UCL (University College London), 2009. 1

[44] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016. 2

[45] Anuj Srivastava, Ian Jermyn, and Shantanu Joshi. Riemannian analysis of probability density functions with applications in vision. In *Computer Vision and Pattern Recognition, 2007. CVPR. IEEE Conference on*, pages 1–8. IEEE, 2007. 1, 7

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2

[47] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 5

[48] Julian Straub, Jason Chang, Oren Freifeld, and John Fisher III. A dirichlet process mixture model for spherical data. In *Artificial Intelligence and Statistics*, pages 930–938, 2015. 1

[49] Ruey S Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005. 1

[50] Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[51] Jelle Veraart, Dmitry S Novikov, Daan Christiaens, Benjamin Ades-Aron, Jan Sijbers, and Els Fieremans. Denoising of diffusion mri using random matrix theory. *NeuroImage*, 142:394–406, 2016. 6

[52] Setsu Wakana, Arvind Caprihan, Martina M Panzenboeck, James H Fallon, Michele Perry, Randy L Gollub, Kegang Hua, Jiangyang Zhang, Hangyi Jiang, Prachi Dubey, et al. Reproducibility of quantitative tractography methods applied to cerebral white matter. *Neuroimage*, 36(3):630–644, 2007. 6

[53] Zhizhou Wang and Baba C Vemuri. Dti segmentation using an information theoretic tensor dissimilarity measure. *IEEE Transactions on Medical Imaging*, 24(10):1267–1277, 2005. 1

[54] Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-train recurrent neural networks for video classification. *arXiv preprint arXiv:1707.01786*, 2017. 5, 6

[55] Anastasia Yendiki, Kami Koldewyn, Sita Kakunoori, Nancy Kanwisher, and Bruce Fischl. Spurious group differences due to head motion in a diffusion mri study. *Neuroimage*, 88:79–90, 2014. 6

[56] Anastasia Yendiki, Patricia Panneck, Priti Srinivasan, Allison Stevens, Lilla Zöllei, Jean Augustinack, Ruopeng Wang, David Salat, Stefan Ehrlich, Tim Behrens, et al. Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas of the underlying anatomy. *Frontiers in neuroinformatics*, 5:23, 2011. 6

[57] Anastasia Yendiki, Martin Reuter, Paul Wilkens, H Diana Rosas, and Bruce Fischl. Joint reconstruction of white-matter pathways from longitudinal diffusion mri data with anatomical priors. *Neuroimage*, 127:277–286, 2016. 6

[58] Kaicheng Yu and Mathieu Salzmann. Second-order convolutional neural networks. *arXiv preprint arXiv:1703.06817*, 2017. 5

[59] Ernesto Zacur, Matias Bossa, and Salvador Olmos. Multivariate tensor-based morphometry with a right-invariant riemannian distance on GL+ (n). *Journal of mathematical imaging and vision*, 50(1-2):18–31, 2014. 1