

ManifoldGLOW: Extending Flow-based Generative Models to Manifolds

Anonymous ECCV submission

Paper ID 80

Abstract. Many measurements in computer vision and machine learning manifest as non-Euclidean data samples. While recent work in the community has successfully extended a number of deep neural network architectures to data where the samples live on certain mathematical manifolds, and this has often provided surprising improvements in performance, the literature on generative models for manifold data is extremely sparse in comparison. If available, this could open the door to a broad range of tools such as image synthesis and data augmentation, which typically utilize generative models, for applications that involve acquisition or analysis of manifold data. In this paper, we propose a novel way to extend flow-based invertible generative models for manifold-valued data. On the theoretical side, we introduce three kinds of invertible layers on manifold-valued data, which are not only analogous to their functionality in flow-based generative models such as GLOW but also preserve the computational benefits (determinants of the Jacobian is easy to calculate). We also develop a two-stream version of GLOW that can transfer information from one manifold to another, which may be of independent interest. On the experimental side, we focus on brain image data such as diffusion Magnetic Resonance (MR) images where we show that we can reconstruct images with orientation distribution function (ODF) from diffusion tensor images (which have poorer angular resolution).

1 Introduction

Many measurements in computer vision and machine learning appear in a form that does not satisfy common Euclidean geometry assumptions. Operating on data where the data samples live in structured spaces often leads to situations where even simple operations such as distances, angles and inner products need to be redefined: while occasionally, Euclidean operations may suffice, the error progressively increases depending on the curvature of the space at hand [1]. One encounters such data quite often – shapes [2], surface normal directions [3], graphs and trees [4,5] as well as probability distribution functions [6] are some common examples in vision and computer graphics [7,8]. Symmetric positive definite matrices [9,10], rotation matrices [11], samples from a sphere [12], subspaces/Grassmannians [13,14], and a number of other algebraic objects are key ingredients in the design of efficient algorithms in computer vision and medical image analysis as well as in the development or theoretical analysis

of various machine learning problems. While a mature literature on extending classical models such as principal components analysis [15], Kalman filtering [16,17], regression [18] to such a manifold data regime is available, identifying ways in which deep neural network architectures can be adapted to leverage and utilize the geometry of such data has only become a prominent research topic in the community over the last few years [19,20,21,13,22]. This research direction, in a short period of time, has already provided convolutional neural networks for various types of manifold measurements [23,24] and recent results also demonstrate how sequential models such as LSTM [25]/GRU [26] can be updated to exploit the manifold structure of the data [27,28,29,30].

The results reported so far in the literature on harnessing the power of deep neural networks for better analysis of manifold or structured data are impressive, but a majority of approaches are discriminative in nature. In other words, the goal is to characterize the conditional distribution $P(Y|\phi(X))$ based on the predictor variables or features X , which happen to be manifold-valued and the responses or label Y . The technical thrust is the design of mechanisms to specify $\phi(\cdot)$ in a manner that respects the geometry of the data space. In contrast, work on the generative side is very sparse, and to our knowledge, only a couple of methods for a few specific manifolds have been proposed thus far. What this means is that our ability to approximate the full joint probability distribution when the data are manifold-valued remains limited. As a result, the numerous application settings in vision where generative models have shown tremendous promise, namely, semi-supervised learning, data augmentation [31,32] and synthesis of new image samples by modifying a latent variable as well as numerous others, currently cannot be evaluated for domains with data-types that are not Euclidean vector-valued data.

GANs for Manifold data: what is challenging? There are some reasons why generative models have sparingly been applied to manifold data. A practical consideration is that many application areas where manifold data are common, such as shape analysis and medical imaging, cannot often provide the sample sizes needed to train off-the-shelf generative models such as Generative adversarial networks (GANs) [33] and Variational auto-encoders (VAEs) [34,35]. There are also several issues on the technical side. Consider the case where a data sample corresponds to an image where each pixel is a manifold variable (such as a covariance matrix). This means that each sample lives on a product space of the manifold of covariance matrices. In attempting to leverage state of the art methods for GANs such as Wasserstein GANs [36] will involve, as a first step, defining appropriate generators that take uniformly distributed samples on a product space of manifolds and transforming it into “realistic” samples which are also samples on a product space of manifolds. In principle, this can be attempted via recent developments by extending spherical CNNs or other architectures for manifold data [37,29]. Next, one would not only need to define optimal transport [38] or Wasserstein distances [39] in fairly complicated spaces, but develop new algorithms to approximate such distances (e.g., Sinkhorn iterations) to make the overall procedure computationally feasible. It is easy to see that closing

these gaps will involve non-trivial extensions. VAEs are mathematically more convenient in comparison to such data, and as a result, a few recent works show how they can be used for dealing with manifold-valued data. While these methods inherit VAE's advantages such as ease of synthesis, VAEs are known to suffer from optimization challenges as well as a tendency to generate smoothed samples. It is not clear how the numerical issues, in particular, will be amplified once we move to manifold data where the core operations of calculating geodesics and distances, evaluating derivatives, and so on, must also invoke numerical optimization routines.

Contributions Instead of GANs or VAEs, we make use of recent progress on flow-based generative models [40,41], which enable latent variable inference and log-likelihood evaluation. It turns out, as we will show in our development shortly, that the key components (and layers) involved in flow-based generative models with some mathematical and procedural adjustments, extends very nicely to the manifold setting. The overarching goal of this work is to describe our theoretical developments and show simple representative examples in applications on manifold-valued data.

2 Preliminaries

This subsection is intended for a very brief summarization of some differential geometric terminologies we are going to use in the rest of the paper. For a more concrete treatment, the readers are encouraged to look at [42].

Definition 1 (Riemannian manifold and metric). Let $(\mathcal{M}, g^{\mathcal{M}})$ be a orientable complete Riemannian manifold with a Riemannian metric g , i.e., $\forall x \in \mathcal{M} : g_x : T_x \mathcal{M} \times T_x \mathcal{M} \rightarrow \mathbf{R}$ is a bi-linear symmetric positive definite map, where $T_x \mathcal{M}$ is the tangent space of \mathcal{M} at $x \in \mathcal{M}$. Let $d : \mathcal{M} \times \mathcal{M} \rightarrow [0, \infty)$ be the distance induced from the Riemannian metric g .

Definition 2. Let $p \in \mathcal{M}$, $r > 0$. Define $\mathcal{B}_r(p) = \{q \in \mathcal{M} | d(p, q) < r\}$ to be a open ball at p of radius r .

Definition 3 (Local injectivity radius [43]). The local injectivity radius is defined as $r_{\text{inj}}(p) = \sup \{r | \text{Exp}_p : (\mathcal{B}_r(\mathbf{0}) \subset T_p \mathcal{M}) \rightarrow \mathcal{M}\}$ where Exp_p is defined and is a diffeomorphism onto its image at $p \in \mathcal{M}$. The injectivity radius [44] of \mathcal{M} is defined as $r_{\text{inj}}(\mathcal{M}) = \inf_{p \in \mathcal{M}} \{r_{\text{inj}}(p)\}$.

Within $\mathcal{B}_r(p)$, where $r \leq r_{\text{inj}}(\mathcal{M})$, the mapping $\text{Exp}_p^{-1} : \mathcal{B}_r(p) \rightarrow \mathcal{U} \subset T_p \mathcal{M} \subset \mathbf{R}^m$, is called the inverse Exponential/Log map, m is the dimension of \mathcal{M} . For each point $p \in \mathcal{M}$, there exists an open ball $\mathcal{U} = \mathcal{B}_r(q)$ for some $q \in \mathcal{M}$ such that $p \in \mathcal{U}$, where $r = r_{\text{inj}}(\mathcal{M})$. Thus, we can cover \mathcal{M} by an indexed (possibly infinite) cover $\{\text{Exp}_q^{-1}\}_{q \in \mathcal{T}}$. An example is shown in Fig.1.

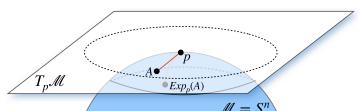


Fig. 1: Schematic description of an exemplar manifold (\mathbf{S}^n) and the corresponding tangent space at a pole.

135 Each of these mapping Exp_q^{-1} is called a *chart map*. For notational simplicity,
 136 we will denote a chart map covering $p \in \mathcal{M}$ by Φ_p . Note that the domain for two
 137 chart maps may not necessarily be disjoint.

138 Given a differentiable function $F : \mathcal{M} \rightarrow \mathcal{M}$ defined as $x \mapsto \Psi^{-1}(\tilde{F}(\Phi(x)))$,
 139 where Φ is the chart map for x and Ψ is the chart map for $F(x)$ and for some
 140 differentiable $\tilde{F} : \mathbf{R}^m \rightarrow \mathbf{R}^m$, the Jacobian of F (denoted by $\frac{dy}{dx}$) is defined as
 141 (note that as the derivative can not be defined on manifold-valued data, $\frac{dy}{dx}$ is
 142 not meaningful. Hence, we use the notation $\tilde{\frac{dy}{dx}}$ to emphasize the difference) :

$$\tilde{\frac{dy}{dx}} := \frac{\partial \Phi \circ \Psi^{-1}}{\partial \psi(x)} \frac{d\tilde{F}}{dx} \Big|_{\Psi(x)} \quad (1)$$

144 **Definition 4 (Group of isometries of \mathcal{M} ($I(\mathcal{M})$)).** A diffeomorphism $\phi :
 145 \mathcal{M} \rightarrow \mathcal{M}$ is an isometry if it preserves distance, i.e., $d(\phi(x), \phi(y)) = d(x, y)$.
 146 The set $I(\mathcal{M})$ forms a group with respect to function composition.

147 Rather than write an isometry as a function ϕ , we will write it as a group
 148 action. Henceforth, let G denote the group $I(\mathcal{M})$, and for $g \in G$, $x \in \mathcal{M}$, let
 149 $g \cdot x$ denote the result of applying the isometry g to point x . Similar to the
 150 terminologies in [29], we will use the term “translation” to denote the group
 151 action ϕ . This is due to the distance preserving property and is inspired by the
 152 analogy from the Euclidean space.

153 3 Flow-based generative models

154 In this section, we will introduce the flow-based generative model for manifold-
 155 valued data. We will first describe the Euclidean formulation and specify which
 156 components need to be generalized to get the manifold-valued formulation.

157 3.1 Flow-based generative models: Euclidean case

158 The flow-based generative models [40,41,45] aim to maximize the log-likelihood of
 159 the input data from an unknown distribution. The main idea lies in mapping the
 160 unknown distribution in the input space to a known distribution in the hidden
 161 space using an invertible function, f .

162 Let $\{\mathbf{x}_i\}$ be i.i.d. samples drawn from an unknown distribution $p^*(\mathbf{x})$. Let
 163 this unknown distribution be parameterized by θ . In the rest of the paper, we
 164 use $p_\theta(\mathbf{x})$ as a proxy for $p^*(\mathbf{x})$. We learn θ over a dataset \mathcal{D} . We maximize
 165 the likelihood of the model θ given the dataset \mathcal{D} by minimizing the equivalent
 166 formulation of negative log-likelihood as given:

$$\ell(\theta|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N -\log p_\theta(\mathbf{x}_i) \quad (2)$$

167 But to minimize the above equation, we need to know p_θ . One way to bypass
 168 this requirement is learning a mapping from a known distribution in the latent
 169 space. Let the latent space be \mathbf{z} . Then the generative step is given by:

$$\mathbf{z} \sim p(\mathbf{z}), \mathbf{x} = g(\mathbf{z})$$

Here $p(\mathbf{z})$ can be Gaussian distribution $\mathcal{N}(\mathbf{z}; \mathbf{0}, I)$. Let f be the inverse of g . For the normalizing flow [40], the f is composed by a sequence of invertible functions $f = f_1 \circ f_2 \circ \dots \circ f_K$. Hence, we have

$$\mathbf{x} \xleftrightarrow{f_1} \mathbf{h}_1 \xleftrightarrow{f_2} \mathbf{h}_2 \dots \xleftrightarrow{f_K} \mathbf{z}$$

Using $\mathbf{h}_0 = \mathbf{x}$ and $\mathbf{h}_K = \mathbf{z}$, the log-likelihood of $p_\theta(\mathbf{x})$ is

$$\log p_\theta(\mathbf{x}) = \log p(\mathbf{z}) + \log |\det(d\mathbf{z}/d\mathbf{x})| = \log p(\mathbf{z}) + \sum_{j=0}^K \log |\det(d\mathbf{h}_j/d\mathbf{h}_{j-1})| \quad (3)$$

In [41], the authors proposed the so-called GLOW model composed of three different layers whose Jacobian $d\mathbf{h}_j/d\mathbf{h}_{j-1}$ is a triangular matrix. Hence the log-determinant can be easily computed:

$$\log |\det(d\mathbf{h}_j/d\mathbf{h}_{j-1})| = \text{sum}(\log |\text{diag}(d\mathbf{h}_j/d\mathbf{h}_{j-1})|)$$

Fig. 2 shows the basic block of the GLOW model. The data is squeezed before getting into the block. After that, the data is split as in [46].

Three layers in the basic GLOW block are invertible functions and are defined as *Actnorm*, *Invertible 1 × 1 convolution*, and *Affine Coupling Layers*. See Table 1 for a summarization.

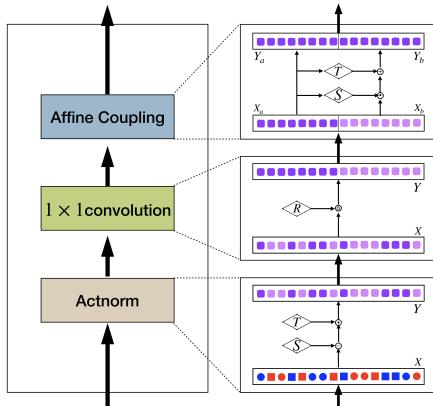


Fig. 2: The basic block of GLOW [41]. The color represents the mean while the shape represents the standard deviation. The target distribution on the latent space is the “Grape” rounded rectangles. *Actnorm* normalizes the data to almost “Grape” rounded rectangles, while the disurbance part is “Lavender”. *1 × 1 convolution* organizes the channels. *Affine Coupling* operates on half of the channels to fit the target distribution.

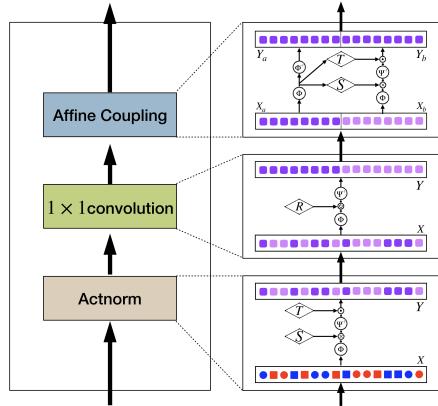


Fig. 3: The basic block of our model. The meanings of colors and shapes are the same as Fig. 2, while all variables lie on the manifold instead of Euclidean space. The major difference between our manifold-valued GLOW and the original GLOW model is we use tangent space transformation before and after every operators.

Actnorm normalizes the input to be zero-mean and identity standard deviation. In Eq. 4, μ, σ are initialized from the data and then trained independently. 1×1 convolution applies the invertible matrix R on the channel dimension. In Eq. 5, $X \in \mathbf{R}^{s_r \times c_r}$ and $R \in \mathbf{R}^{c_r \times c_r}$. The s_r is the resolution of the input variables while c_r is the number of channels.

Affine Coupling uses the idea of split and concatenation. In Eq. 6, the input variable X is split along the channel to X_a, X_b , and then Y_a, Y_b are concatenated to get the final output Y . Here, S and T are real valued matrix of the same dimension as X_b for element-wise scaling and translation respectively.

Actnorm	1×1 convolution	Affine Coupling
$Y = \frac{1}{\sigma} \odot (X - \mu)$ (4)	$Y = R \cdot X$ (5)	$S, T = \text{NN}(X_a)$ $Y_b = S \odot X_b + T$ (6) $Y_a = X_a$

Table 1: Definition of Actnorm, 1×1 convolution and Affine Coupling layers in basic GLOW block. \odot is the elementwise multiplication. The function $\text{NN}()$ is a nonlinear mapping. **caption need modify**

In [41], the author introduced the closed forms of the inverse of these layers. It is also noticeable that the determinant of $\frac{dY}{dX} = \begin{bmatrix} \frac{dY_a}{dX_a} & \frac{dY_a}{dX_b} \\ \frac{dY_b}{dX_a} & \frac{dY_b}{dX_b} \end{bmatrix}$ the Jacobian is simple to compute for all these layers. The only non-trivial determinant of the Jacobian is the Eq. 6. Since $\frac{dY_a}{dX_a} = 0, \frac{dY_a}{dX_b} = I$, the determinant of the Jacobian is $\det(S)$.

With the description above, we can now list the key operational components in Eq. 4-7, which we need to modify for our manifold-valued extension.

Key ingredients: In Eq. 4 and Eq. 6, the operators are elementwise multiplication for σ, S and the addition of bias for μ, T . In Eq. 5, it requires the invertible matrices. Finally, to compute the log-likelihood, we need the calculation of derivative in Eq. 7. Thus we can verify that the key ingredients to define the model in GLOW are **(i)** elementwise multiplication; **(ii)** addition of bias; **(iii)** invertible matrix; **(iv)** derivative calculation. In theory, if we can modify those components from Euclidean space to manifolds, we can have the flow-based generative models on Riemannian manifold. Observe that items **(i)** and **(iii)** are matrix multiplication, which is non-trivial to define on a manifold. In Def. 3, we can use the chart map to map the manifold to a subspace of \mathbf{R}^m where the matrix multiplication exists. This also provides the way to solve item **(iv)** based on the chart map. Details are discussed in Eq. 1. For the item **(ii)**, adding bias can be viewed as the “translation” in the Euclidean space, while in Def. 4 we define the translation on manifold-valued data by the group action. With these in hand, we are ready to present our proposed manifold version of these layers in the following.

3.2 Flow-based generative models: Riemannian manifold case

In this section, we will introduce the manifold counterpart of the key operations. See Table 2 for a summarization of functions.

Determinant of the Jacobian can be computed as in Eq. 8. In general, s_r can be a tuple, i.e., for 3D data, it is a 3 dimensional tuple.

$$\det \left(\frac{dY}{d\tilde{X}} \right) = \prod_{s_r \times c_r} \left(\prod s_i \right) \det (\Phi \circ \Psi^{-1}) \quad (8)$$

1 × 1 convolution: We define 1×1 convolution to have the flexibility of interaction between channels. This layer function is defined as in Eq. 13. We use the same notation as in Eq. 5. Here R is a $c_r \times c_r$ matrix. In general, we can learn any $R \in \text{GL}(c_r)$, i.e., a full rank matrix like Eq. 5 does. But in practice, maintaining full rank is a hard constraint and may become unbounded. To put regularization constraint, we choose R to be a rotation matrix.

Determinant of the Jacobian can be computed as in Eq. 9. Notice that for R to be a rotation matrix, the contribution from $\det(R)$ is 1.

$$\det \left(\frac{dY}{d\tilde{X}} \right) = \prod_{s_r} \det(R) \det (\Phi \circ \Psi^{-1}) \quad (9)$$

Affine Coupling: For manifold-valued data, given $X \in \mathcal{M}^{s_r \times c_r}$ (where s_r and c_r are spatial and channel resolutions), we first split the data along the channel dimension, i.e., partition c_r into two parts. Let the two parts of the data be denoted by $X_a \in \mathcal{M}^{s_r \times c_a}$ and $X_b \in \mathcal{M}^{s_r \times c_b}$, where $c_r = c_a + c_b$. Then, the affine coupling layer is defined as in Eq. 14. Here, $S \in (\mathbf{R}^{m \times m})^{s_r \times c_a}$ and $T \in G^{s_r \times c_b}$. These two operators play the same roles as in Eq. 6, scaling and shifting. We need S to be of full rank. One may put additional constraint like orthogonality or bounded matrix for stability of optimization. After performing the coupling, we combine Y_a and Y_b to get $Y \in \mathcal{M}^{s_r \times c_r}$ as our output.

Determinant of the Jacobian can be computed as: $\begin{bmatrix} \frac{dY_a}{dX_a} & \frac{dY_a}{dX_b} \\ \frac{\tilde{d}Y_b}{dX_a} & \frac{\tilde{d}Y_b}{dX_b} \end{bmatrix}$. Similar to Eq. 7,

observe that $\frac{dY_b}{dX_a}$ involves taking the gradient of a neural network! But fortunately, we only require the determinant of the Jacobian matrix, and the independence of Y_a on X_b saves the computation of $\frac{dY_b}{dX_a}$ as $\frac{dY_a}{dX_b} = 0$.

Thus, given $X, Y \in \mathcal{M}^{s_r \times c_r}$, the determinant of the Jacobian is given as in Eq. 10.

$$\det \left(\frac{dY}{d\tilde{X}} \right) = \prod_{s_r \times c_r} \det(S) \det (\Phi \circ \Psi^{-1}) \quad (10)$$

Distribution on the latent space: After the cascaded functional transformations described above, we transform X to the latent space $Z \in \mathcal{M}^{s_h \times c_h}$. We define a Gaussian distribution on Z by inducing multi-variate Gaussian distribution from \mathbf{R}^m . The Gaussian distribution are defined as follows:

$$P(Z; M, \Sigma) = \frac{1}{C(\Sigma)} \exp \left(-\frac{(\Phi(Z) - \Phi(M))^T \Sigma^{-1} (\Phi(Z) - \Phi(M))}{2} \right) \quad (11)$$

where $M \in \mathcal{M}^{s_h \times c_h}$ and $\Sigma \in \text{SPD}(m)^{s_h \times c_h}$.

Generating manifold-valued data Now with the basic tools in our hand, we can generate manifold-valued data. We start by drawing a random sample from $P(Z; M, \Sigma)$ with fixed M and Σ , then pass it to the inverses of the *affine coupling*, 1×1 convolution, and *actnorm* defined in Table 2 to generate the corresponding sample of \mathcal{M} . This gives us a fully trainable invertible flow-based generative model for manifold-valued data. A basic block of our model is shown in Fig. 3.

Actnorm	1×1 convolution	Affine Coupling
$Y = \Psi^{-1}(S\Phi(X)) \cdot T$ (12)	$Y = \Psi^{-1}(R\Phi(X))$ (13)	$S, T = \text{NN}(\Phi(X_a))$ $Y_b = \Psi^{-1}(S\Phi(X_b)) \cdot T$ (14) $Y_a = X_a$
$X = \Phi^{-1}(S^{-1}\Psi(Y \cdot T^{-1}))$ (15)	$X = \Phi^{-1}(R^{-1}\Psi(Y))$ (16)	$S, T = \text{NN}(\Phi(Y_a))$ $X_b = \Phi^{-1}(S^{-1}\Psi(Y_b \cdot T^{-1}))$ (17) $X_a = Y_a$

Table 2: Definition of *Actnorm*, 1×1 convolution and *Affine Coupling* layers in our ManifoldGLOW block, with forward function on the top and reverse function in the bottom. Here Φ and Ψ^{-1} are the Chart Map and its inverse. S is a diagonal matrix, so S^{-1} can be computed elementwise. T^{-1} represents the inverse of the group action. The R is chosen as the rotation matrix. Thus, $R^{-1} = R^T$.
caption need modify

3.3 Flow-based generative models: learning function between two manifolds

In this section, we go one step further and ask the question: *Can we draw manifold-valued data conditioned on other manifold-valued sample?* Due to the nature of the invertibility of our proposed generative model, this seems to be possible as all we need to do extra is to develop a scheme to draw data from Euclidean space conditioned on a vector-valued input.

Recently there is some work using GLOW model in Euclidean setup to generate samples from space \mathcal{X} conditioned on space \mathcal{Y} . In this section, we generalize the parallel GLOW with connection in latent space on a manifold setting to generate the sample from manifold \mathcal{M} given a sample on manifold \mathcal{N} . The underlying assumption behind this setup is that there exists a (smooth) function from \mathcal{N} to \mathcal{M} . Given variables $X \in \mathcal{M}^{s_x \times c_x}$ and $Y \in \mathcal{N}^{s_y \times c_y}$ with the dimension of the manifolds \mathcal{M} and \mathcal{N} to be m and n respectively, we use the parallel GLOW model to get the corresponding latent space. Let it be denoted by Z_m and Z_n respectively.

After getting the respective latent spaces, the next step is to fit a distribution on it. As our intention is to generate samples from \mathcal{M} , the distribution on the respective latent space Z_m must be induced from the variables in Z_n , i.e., the latent space for \mathcal{N} . We do not have any constraint on the distribution parameters for Z_n , hence, as before we use a Gaussian distribution with fixed M and Σ .

on Z_n . The parameters for the Gaussian distribution on Z_m are defined as functions of Z_n . More formally, we define $P(Z_m; M_m, \Sigma_m)$ using Eq. 11, where, $M_m = \Phi^{-1}(F_M(\Phi(Z_n)))$ and $\Sigma_m = F_S(\Phi(Z_n))$. Here the two functions F_M and F_S are modeled using a neural network. The scheme is shown in Fig. 4.

3.4 Specific examples of manifolds

In this section, we give particular examples of manifolds and their respective key operations. In order to implement Eq. 12, 13 and 14 mentioned in the previous sections, basic operations specific to a manifold are (a) the choice of distance, d , (b) the isometry group, G , (c) the chart map Φ and its inverse, Φ^{-1} . We use three types of non-Euclidean Riemannian manifold in the proof-of-concept type experiments presented in this work (including the supplement section), they are (a) hypersphere, S^{n-1} (b) space of positive real numbers, \mathbf{R}_+ (c) space of $n \times n$ symmetric positive definite matrices ($SPD(n)$). We give the explicit formulation for the basic operations in Table 3.

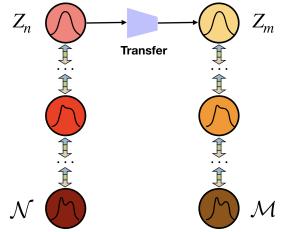


Fig. 4: Transfer from the source manifold N to the target manifold M with the generative model.

	$d(X, Y)$	G	$\Phi(X)$	$\Phi^{-1}(\mathbf{v})$
S^{n-1}	$\arccos(X^T Y)$	$SO(n-1)$	$\frac{\theta}{\sin(\theta)}(X - P \cos(\theta))$	$P \cos(\ \mathbf{v}\) + \frac{\mathbf{v}}{\ \mathbf{v}\ } \sin(\ \mathbf{v}\)$
\mathbf{R}_+	$ \log(X/Y) $	$\mathbf{R} \setminus \{0\}$	$\log(X)$	$\exp(\mathbf{v})$
$SPD(n)$	$\ \logm X^{-1} Y\ $	$SO(m)$	$Chol(X)$	$\mathbf{v} \mathbf{v}^T$

Table 3: The explicit formulation for the basic operations. Here P is an anchor point for chart map, which can be one of the poles. $\theta = d(P, X)$, and $SO(n-1)$ is the group of $n \times n$ special orthogonal matrices. $\mathbf{v} \in \mathbf{R}^{n-1}$. $Chol$ is the Cholesky decomposition. caption need modify

3.5 Applications

In this section, we present some applications for our proposed manifold-valued generation scheme. Our major focused application is on diffusion MR scans, go between the two popular representations, namely generate orientation distribution function (ODF) from diffusion tensor imaging (DTI). We present some additional experiments including texture generation and generating DTI from ODF in the supplement. In the rest of this section, we give brief description of the application settings.

Transformation between ODF and DTI In the diffusion-weighted MR images (dMRI) field, it usually uses the manifold value to represent the structure of the brain by measuring the diffusion ability of the water molecule. For DTI [48, 49], the value of each voxel is the covariance matrix of the Gaussian distribution of the water molecule centered at that voxel. Thus, each value lies on the symmetric positive definite (SPD) manifold. The ODF [50] is another method to represent the

405 dMRI, where each voxel is an orientation distribution function. By symmetrically
 406 equally sampling 362 points on the continuous distribution function [51], each
 407 data is a 362 dimensional vector with each element greater than 0 and sum to 1.
 408 By square root parameterization [52,6], the data in each voxel lies on the positive
 409 part of \mathbf{S}^{361} manifold.

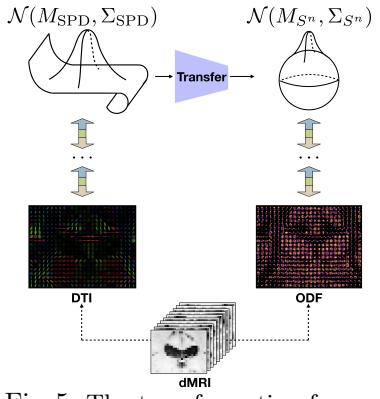
410 It is believed that ODF contains more information than DTI, since ODF can modify
 411 various direction while DTI only contains three eigen vectors. But there is no closed form that
 412 can generate DTI from ODF.
 413

414 Here, we will introduce the transformation
 415 between ODF and DTI, i.e., going from DTI
 416 to ODF. The naïve idea is to train the neural
 417 network directly applying on DTI to get the
 418 ODF. However, it will be quite hard since the
 419 distributions for ODF and DTI are complicated
 420 and very much different. On the contrary, the
 421 latent variable for the flow-based model is a
 422 simple distribution, e.g., Gaussian. So in the
 423 latent space, it will be easier to learn the map-
 424 ping with the assumption that the structure
 425 information is contained in the model. We first
 426 build two flow-based models for DTI and ODF
 427 separately. And in the latent space, we train a
 428 transformation operating between the Gaussian
 429 distribution variable on \mathbf{S}^{361} and Gaussian distribution variable on
 430 $\text{SPD}(3)$. The hybrid model of two flow-based models and the transformer can be jointly trained
 431 as shown in Fig. 5. When going from DTI to ODF,
 432 the manifold \mathcal{N} is $\text{SPD}(3)$ and \mathcal{M} is \mathbf{S}^{361} .

4 Experiments

433 In this section, we focus on the structural brain image generation. As mentioned
 434 before, we generate ODF representation of a 3D brain image from the correspond-
 435 ing DTI representation. We explain the **(a)** rationale behind this application
 436 **(b)** dataset description **(c)** model setup *qualitative and quantitative evaluation*
 437 in the rest of this section. *Note that in the experiment, as we draw samples from*
 438 *the distribution on the latent space to get the target representation, we call it*
 439 *generation rather than reconstruction.*

440 **Why generating ODF from DTI is of significance?** For diffusion MRI,
 441 there are a number of different image types that can be obtained. Usually, the
 442 higher spatial resolution images involve a longer acquisition time and specialized
 443 pulse sequences. Such images are, in principle, of higher quality but longer
 444 acquisition often leads to participant motion which results in motion artifacts.
 445 This means that there is a trade-off that must be considered. If we want to shorten



446 Fig. 5: The transformation from
 447 DTI to ODF. Both are generated
 448 from dMRI. But there might not
 449 be dMRI available in some situations.
 450 Thus, we want to train the
 451 network to transfer DTI to ODF.
 452 The latent space is the Gaussian
 453 distribution variable.

the time of measurement without a corresponding compromise in the quality of the images, we require mechanisms to transform data acquired from a shorter acquisition sequence (DTI) to a higher spatial resolution image such as a field of ODFs. The potential amount of saving in acquisition time is significant and hence makes this problem interesting in the medical community. This motivates us to tackle this problem in this work. The difficulty of this task comes from (a) the per voxel degrees of freedom for ODF representation is 361 (lies on S^{361}) while for DTI is 6 (lies on $SPD(3)$). Hence, it is an ill-posed problem, due to the fact that per-voxel dimensionality of ODF representation is 361 compared to the given DTI representation which is of dimension 6. Thus the generation problem essentially try to find a non-linear mapping from 6 dimensional $SPD(3)$ manifold to 361 dimensional hypersphere, i.e., learning the map to a higher *angular* resolution. (b) requires sophisticated tool to “transform” from one manifold (DTI representation) to another (ODF representation) while preserving structural information . Now, we will describe the detail of the dataset used before describing models and presenting the results.

Dataset The dataset for our method is a subset of the Human Connectome Project (HCP) [60].

We randomly sampled 752 subjects from the whole dataset which have the pre-processed 3T diffusion-weighted MR images (dMRI), with 450 of them as the training set and 302 of them as the testing set. The detail of the demographics is shown in Table 4. Please refer to [60] for more detailed description of this dataset. All the raw dMRI images are preprocessed with the HCP diffusion pipeline with FSL’s ‘eddy’ [61]. After the correction, the orientation distribution functions (ODF) and diffusion tensor imaging (DTI) pairs are generated using the Diffusion Imaging in Python (DIPY) toolbox [51].

Due to computational limitation, generation of ODF of the entire brain is infeasible, hence we restrict ourselves to a given region of interest (ROI).We choose a crop of shape $20 \times 30 \times 26$ (as shown in the red box in Fig. 6), which contains two ROIs (enclosed in blue box) of the white matter containing fiber tracts, and pad it to $32 \times 32 \times 32$. The choice of a region instead of the entire brain is not radical and it is very common in the analysis of diffusion tensor images.

Model Setup The samples are presented to the model in paired form, i.e., a DTI image and a corresponding image comprised of a field of ODFs, one at each voxel location. We use 6 blocks of GLOW, and after every 2 blocks reduce the resolution by half. This setup is same for both for generating DTI and ODF. We use 3 residual network blocks to map the latent space from DTI to ODF. In this setup, no fine-tuning of hyper-parameters was performed

Age	Gender		
	Female	Male	
22-25	26-30	31-35	36+
171	322	251	8
	383(50.9%)	369(49.1%)	

Table 4: The demographics used in the study.

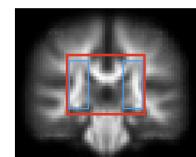
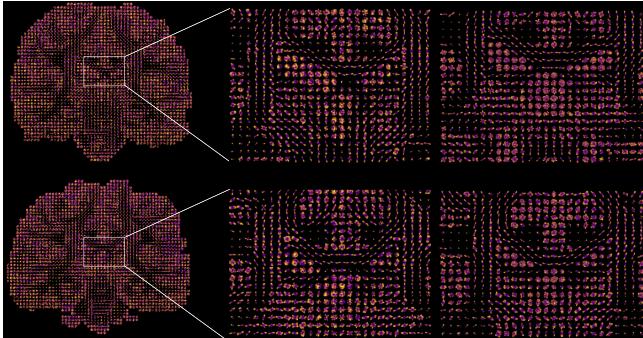


Fig. 6: ROIs are highlighted in bounding box.

495 because training takes several days (similar/more than
 496 the GLOW model) and is very resource intensive.

497 **Baseline:** Since there are no deep generative models for manifold-valued data
 498 with code available, we perform a baseline experiment on GLOW by treating the
 499 inputs as vectors. However, due to the large channel size for ODF, the memory
 500 consumption is $14000 \times$ more than the GLOW paper, which is infeasible even on
 501 massive clusters. Even using one block of two affine coupling layers, the training
 502 does not converge. This gives an empirical justification behind the necessity of
 503 manifold valued generative model, such as our proposed manifold GLOW.

504 **Choice of metrics:** As a quantitative measure, we will use *reconstruction*
 505 *error* using the distance mentioned in Table 3. Although the task here is generation,
 506 the purpose of measuring reconstruction error is to assess how “similar” the
 507 original ODF is to the generated ODF generated only from the corresponding
 508 DTI representation. Additionally, we present an analysis of group difference.
 509 Since the dataset of interest is HCP, which contains only healthy samples, the
 510 major group difference we can test is on the major genders, i.e., *male vs. female*.
 511 As a qualitative analysis, we show both generated ODF with the ground-truth
 512 ODF as well the generated ODF with the given DTI representation (additional
 513 figures are in the supplement). Finally, we show the group differences between
 514 male versus female in heatmaps to highlight the statistically significant region in
 515 the brain scans.



516 Fig. 7: (*Left:*) entire brain image, (*Middle:*) ODF from the ROI and (*Right:*) generated
 517 ODF given the respective DTI representation. The two examples have reconstruction
 518 error (in ODF space) to be 0.018 and 0.019 respectively.

519 **Generation results** Here, we present the quantitative and qualitative results
 520 for generation of ODF given the corresponding DTI representation. In order
 521 to demonstrate the effectiveness of our proposed manifold glow, we start of by
 522 showing the visual quality of the generated ODF (few slices). In Fig. 7, we show
 523 the generated ODF from the corresponding DTI for two representative slices.
 524 In Fig. 8, we have shown some few example slices with the corresponding given
 525 DTI. Visually, we can see the structures well are preserved in the generated ODF.
 526 Here, we observe the reconstruction error to be 0.018 and 0.019 respectively. This
 527 low reconstruction error makes us wonder to dig deeper and do the following
 528 quantitative analysis; 1. a histogram of the reconstruction error over all 302 test
 529

subjects across three different slices (shown in Fig. 9) 2. an error matrix showing how similar the generated ODF is with the other samples of the population, the purpose of this is show that the generated ODF is distinctive across different samples (shown in Fig. 11). From the histogram presented in Fig. 9, we can see that across different slices, the reconstruction error is consistently low over the entire test population. Now, we generate Fig. 11 as follows. For each subject in the test population, we randomly select other 29 samples from the population and plot the reconstruction error with the generated ODF. This gives us a 30×30 matrix (similar to confusion matrix). We then repeat this process 10 times and show the average in Fig. 11 with the darker the shade the larger the reconstruction error. So essentially we are looking for a white diagonal in the perfect case, which is approximately the case as observed in the figure. This summarizes that over the entire test population, the generation is meaningful (respect to preserving structures) and distinctive (regards to maintain variability across subjects) not only visually but also quantitatively.

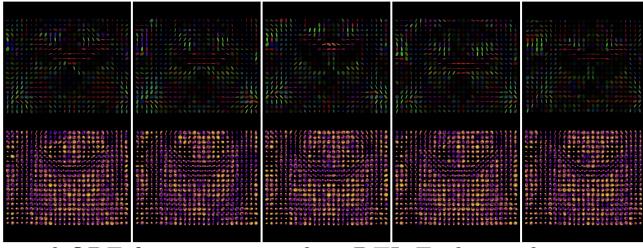


Fig. 8: Generated ODF from corresponding DTI. Each pair here contains the input DTI (top) and the generated ODF (bottom).

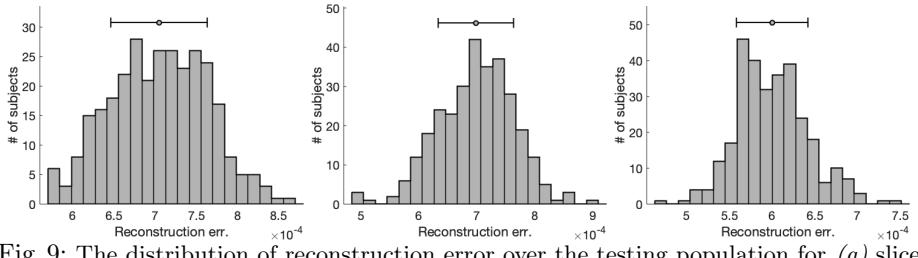


Fig. 9: The distribution of reconstruction error over the testing population for (a) slice 9, (b) slice 12 and (c) slice 15.

Group difference analysis So far, we observe meaningful ODF generation which is good but we can only argue about preserving the structures based on the visual inspection of Figs. 8, 7. But it would have been interesting to quantitatively measure how much structure is present. In order to do that, we perform a group testing over the test population with regards to male versus female. We perform the group testing over the following representation 1. original ODF 2. generated

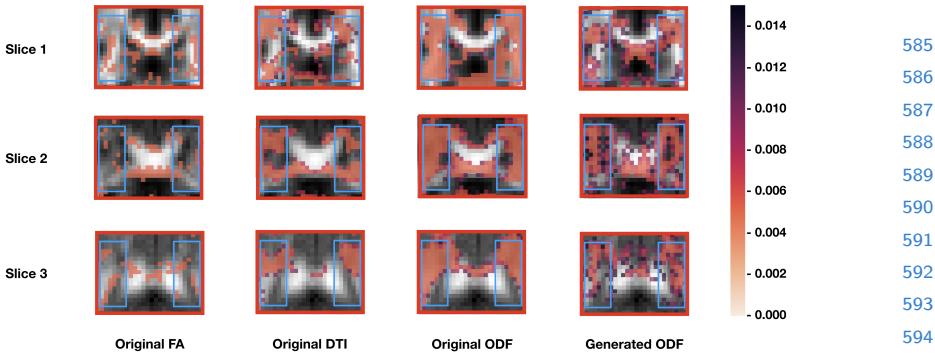


Fig. 10: The statistical significance between male and female in the testing set. The blue boxes are the chosen ROIs as shown in Fig. 6.

ODF 3. DTI 4. functional anisotropy (FA) representation . Both DTI and FA are known to be useful for showing statistically significant differences across genders [29,30]. But as ODF contains more structural information than either of the FA or DTI, our generated ODF should be able to pick up more statistically significant regions over DTI or FA. In order to show this, we not only plot the p-values across a chosen slice (slice 12), but also evaluate the intersection of significant regions with the original ODF (as the original ODF is the one having the most information). As we can see that the generated ODF has larger intersection in the statistically significant regions of the original ODF. This gives us the quantitative proof that the generated ODF preserves the significant structures.

5 Conclusions

Non-Euclidean data and manifold-valued data have so far gained some attention in the research community. While most of these developments are based on application to analysis using CNNs or RNNs, in this work, we introduce the flow-based generative model on the Riemannian manifold. We show three types of layers, Actnorm, Invertible 1×1 convolution, and Affine Coupling layers, can be applied on manifold-valued data which preserve invertibility. We also prove that with the transformation in the latent space between the two manifolds, we can generate manifold-valued data based on the information from another manifold. We demonstrate good generation results in the representation of ODF given DTI on Human Connectome dataset. We like to point out that we can generate manifold-valued representation without any explicit reconstruction error loss.

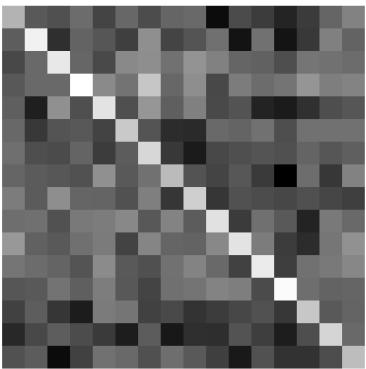


Fig. 11: Reconstruction error over test population

630 References

- 632 1. Feragen, A., Lauze, F., Hauberg, S.: Geodesic exponential kernels: When curvature
633 and linearity conflict. In: Proceedings of the IEEE Conference on Computer Vision
634 and Pattern Recognition. (2015) 3032–3042 [1](#)
- 635 2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese,
636 S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model
637 repository. arXiv preprint arXiv:1512.03012 (2015) [1](#)
- 638 3. Straub, J., Chang, J., Freifeld, O., Fisher III, J.: A dirichlet process mixture model
639 for spherical data. In: Artificial Intelligence and Statistics. (2015) 930–938 [1](#)
- 640 4. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph
641 neural network model. IEEE Transactions on Neural Networks **20**(1) (2008) 61–80
[1](#)
- 642 5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional
643 networks. arXiv preprint arXiv:1609.02907 (2016) [1](#)
- 644 6. Srivastava, A., Jermyn, I., Joshi, S.: Riemannian analysis of probability density
645 functions with applications in vision. In: 2007 IEEE Conference on Computer
646 Vision and Pattern Recognition, IEEE (2007) 1–8 [1, 10](#)
- 647 7. Bruno, R., Conti, M., Gregori, E.: Mesh networks: commodity multihop ad hoc
648 networks. IEEE communications magazine **43**(3) (2005) 123–131 [1](#)
- 649 8. Huang, R., Rakotosaona, M.J., Achlioptas, P., Guibas, L., Ovsjanikov, M.: Operator-
650 net: Recovering 3d shapes from difference operators. arXiv preprint arXiv:1904.10754
651 (2019) [1](#)
- 652 9. Moakher, M.: A differential geometric approach to the geometric mean of symmetric
653 positive-definite matrices. SIAM Journal on Matrix Analysis and Applications
654 **26**(3) (2005) 735–747 [1](#)
- 655 10. Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M.: Kernel methods on
656 the riemannian manifold of symmetric positive definite matrices. In: Proceedings of
657 the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 73–80 [1](#)
- 658 11. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with
659 deep learning. In: Proceedings of the IEEE Conference on Computer Vision and
660 Pattern Recognition. (2017) 5974–5983 [1](#)
- 661 12. Koppers, S., Merhof, D.: Direct estimation of fiber orientations using deep learning
662 in diffusion imaging. In: International Workshop on Machine Learning in Medical
663 Imaging, Springer (2016) 53–60 [1](#)
- 664 13. Huang, Z., Wu, J., Van Gool, L.: Building deep networks on grassmann manifolds.
665 In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018) [1, 2](#)
- 666 14. Chakraborty, R., Hauberg, S., Vemuri, B.C.: Intrinsic grassmann averages for online
667 linear and robust subspace learning. In: Proceedings of the IEEE Conference on
668 Computer Vision and Pattern Recognition. (2017) 6196–6204 [1](#)
- 669 15. Duntzman, G.H.: Principal components analysis. Number 69. Sage (1989) [2](#)
- 670 16. Haykin, S.: Kalman filtering and neural networks. Volume 47. John Wiley & Sons
671 (2004) [2](#)
- 672 17. Grewal, M.S.: Kalman filtering. Springer (2011) [2](#)
- 673 18. Fletcher, P.T.: Geodesic regression and the theory of least squares on riemannian
674 manifolds. International journal of computer vision **105**(2) (2013) 171–185 [2](#)
19. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric
deep learning: going beyond euclidean data. IEEE Signal Processing Magazine
34(4) (2017) 18–42 [2](#)

- 675 20. Chakraborty, R., Bouza, J., Manton, J., Vemuri, B.C.: Manifoldnet: A deep network
676 framework for manifold-valued data. arXiv preprint arXiv:1809.06211 (2018) 2 675
- 677 21. Kondor, R., Trivedi, S.: On the generalization of equivariance and convolution in
678 neural networks to the action of compact groups. arXiv preprint arXiv:1802.03690
679 (2018) 2 677
- 680 22. Huang, Z., Van Gool, L.: A riemannian network for spd matrix learning. In: Thirty-First
681 AAAI Conference on Artificial Intelligence. (2017) 2 680
- 682 23. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional
683 neural networks on riemannian manifolds. In: Proceedings of the IEEE international
684 conference on computer vision workshops. (2015) 37–45 2 683
- 685 24. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Shapenet: Convolutional
686 neural networks on non-euclidean manifolds. Technical report (2015) 2 685
- 687 25. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation
688 9(8) (1997) 1735–1780 2 687
- 689 26. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural
690 machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259
691 (2014) 2 690
- 692 27. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on
693 spatio-temporal graphs. In: Proceedings of the IEEE Conference on Computer
694 Vision and Pattern Recognition. (2016) 5308–5317 2 693
- 695 28. Pratiher, S., Chatteraj, S., Agarwal, S., Bhattacharya, S.: Grading tumor malig-
696 nancy via deep bidirectional lstm on graph manifold encoded histopathological image.
697 In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW),
698 IEEE (2018) 674–681 2 696
- 699 29. Chakraborty, R., Yang, C.H., Zhen, X., Banerjee, M., Archer, D., Vaillancourt, D.,
700 Singh, V., Vemuri, B.: A statistical recurrent model on the manifold of symmetric
701 positive definite matrices. In: Advances in Neural Information Processing Systems.
702 (2018) 8883–8894 2, 4, 14 699
- 703 30. Zhen, X., Chakraborty, R., Vogt, N., Bendlin, B.B., Singh, V.: Dilated convolutional
704 neural networks for sequential manifold-valued data. In: Proceedings of the IEEE
705 International Conference on Computer Vision. (2019) 10621–10631 2, 14 703
- 706 31. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial
707 networks. arXiv preprint arXiv:1711.04340 (2017) 2 704
- 708 32. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep
709 convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434
710 (2015) 2 705
- 711 33. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,
712 Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural
713 information processing systems. (2014) 2672–2680 2 709
- 714 34. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint
715 arXiv:1312.6114 (2013) 2 711
- 716 35. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908
717 (2016) 2 712
- 718 36. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks.
719 In: International conference on machine learning. (2017) 214–223 2 714
- 720 37. Chakraborty, R., Banerjee, M., Vemuri, B.C.: A cnn for homogeneous riemannian
721 manifolds with applications to neuroimaging. arXiv preprint arXiv:1805.05487
722 (2018) 2 715
- 723 38. Fathi, A., Figalli, A.: Optimal transportation on non-compact manifolds. Israel
724 Journal of Mathematics 175(1) (2010) 1–59 2 716

- 720 39. Huang, Z., Wu, J., Van Gool, L.: Manifold-valued image generation with wasserstein
721 generative adversarial nets. In: Proceedings of the AAAI Conference on Artificial
722 Intelligence. Volume 33. (2019) 3886–3893 2
- 723 40. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. arXiv
724 preprint arXiv:1505.05770 (2015) 3, 4, 5
- 725 41. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions.
726 In: Advances in Neural Information Processing Systems. (2018) 10215–10224 3, 4,
727 5, 6
- 728 42. Boothby, W.M.: An introduction to differentiable manifolds and Riemannian
729 geometry. Volume 120. Academic press (1986) 3
- 730 43. Giroisser, D.: Newton’s method, zeroes of vector fields, and the riemannian center
731 of mass. Advances in Applied Mathematics **33**(1) (2004) 95–135 3
- 732 44. Manton, J.H.: A globally convergent numerical algorithm for computing the centre
733 of mass on compact lie groups. In: ICARCV 2004 8th Control, Automation, Robotics
734 and Vision Conference, 2004. Volume 3., IEEE (2004) 2211–2216 3
- 735 45. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow:
736 3d point cloud generation with continuous normalizing flows. In: Proceedings of
737 the IEEE International Conference on Computer Vision. (2019) 4541–4550 4
- 738 46. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv
739 preprint arXiv:1605.08803 (2016) 5
- 740 47. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by
741 reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
- 742 48. Basser, P.J., Mattiello, J., LeBihan, D.: Mr diffusion tensor spectroscopy and
743 imaging. Biophysical journal **66**(1) (1994) 259–267 9
- 744 49. Alexander, A.L., Lee, J.E., Lazar, M., Field, A.S.: Diffusion tensor imaging of the
745 brain. Neurotherapeutics **4**(3) (2007) 316–329 9
- 746 50. Hess, C.P., Mukherjee, P., Han, E.T., Xu, D., Vigneron, D.B.: Q-ball reconstruction
747 of multimodal fiber orientations using the spherical harmonic basis. Magnetic
748 Resonance in Medicine: An Official Journal of the International Society for Magnetic
749 Resonance in Medicine **56**(1) (2006) 104–117 9
- 750 51. Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Van Der Walt, S., De-
751 scoteaux, M., Nimmo-Smith, I.: Dipy, a library for the analysis of diffusion mri
752 data. Frontiers in neuroinformatics **8** (2014) 8 10, 11
- 753 52. Brody, D.C., Hughston, L.P.: Statistical geometry in quantum mechanics. Pro-
754 ceedings of the Royal Society of London. Series A: Mathematical, Physical and
755 Engineering Sciences **454**(1977) (1998) 2445–2475 10
- 756 53. Bergmann, U., Jetchev, N., Vollgraf, R.: Learning texture manifolds with the
757 periodic spatial gan. In: Proceedings of the 34th International Conference on
758 Machine Learning-Volume 70, JMLR. org (2017) 469–477
- 759 54. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In:
760 Proceedings of the 28th annual conference on Computer graphics and interactive
761 techniques, ACM (2001) 341–346
- 762 55. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural
763 networks. In: Advances in neural information processing systems. (2015) 262–270
- 764 56. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures
765 in the wild. In: Proceedings of the IEEE Conference on Computer Vision and
766 Pattern Recognition. (2014) 3606–3613
- 767 57. Yu, N., Barnes, C., Shechtman, E., Amirghodsi, S., Lukac, M.: Texture mixer: A
768 network for controllable synthesis and interpolation of texture. In: Proceedings
769 of the IEEE Conference on Computer Vision and Pattern Recognition. (2019)
770 12164–12173

- 765 58. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection
766 and classification. In: European conference on computer vision, Springer (2006)
767 589–600
- 768 59. Wu, C.M., Chen, Y.C.: Statistical feature matrix for texture analysis. CVGIP:
769 Graphical Models and Image Processing **54**(5) (1992) 407–419
- 770 60. Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E., Yacoub, E., Ugurbil, K.,
771 Consortium, W.M.H., et al.: The wu-minn human connectome project: an overview.
772 Neuroimage **80** (2013) 62–79 11
- 773 61. Andersson, J.L., Sotiroopoulos, S.N.: An integrated approach to correction for
774 off-resonance effects and subject movement in diffusion mr imaging. Neuroimage
775 **125** (2016) 1063–1078 11

776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809