

---

# Grassmann Manifold Flows for Stable Shape Generation

---

Ryoma Yataka<sup>1,2</sup>, Kazuki Hirashima<sup>1</sup>, Masashi Shiraishi<sup>1</sup>

<sup>1</sup>Information Technology R&D Center (ITC), Mitsubishi Electric Corporation, Kamakura, Kanagawa, 247-8501, Japan

<sup>2</sup>Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA  
{Yataka.Ryoma@dw.MitsubishiElectric.co.jp, yataka@merl.com}

## Abstract

Recently, studies on machine learning have focused on methods that use symmetry implicit in a specific manifold as an inductive bias. Grassmann manifolds provide the ability to handle fundamental shapes represented as shape spaces, enabling stable shape analysis. In this paper, we present a novel approach in which we establish the theoretical foundations for learning distributions on the Grassmann manifold via continuous normalization flows, with the explicit goal of generating stable shapes. Our approach facilitates more robust generation by effectively eliminating the influence of extraneous transformations, such as rotations and inversions, through learning and generating within a Grassmann manifold designed to accommodate the essential shape information of the object. The experimental results indicated that the proposed method could generate high-quality samples by capturing the data structure. Furthermore, the proposed method significantly outperformed state-of-the-art methods in terms of the log-likelihood or evidence lower bound. The results obtained are expected to stimulate further research in this field, leading to advances for stable shape generation and analysis.

## 1 Introduction

Many machine learning algorithms are designed to automatically learn and extract latent factors that explain a specific dataset. Symmetry is known to be an inductive bias (that is, prior knowledge other than training data that can contribute significantly to learning results) for learning latent factors (Cohen & Welling (2016, 2017); Weiler & Cesa (2019); Satorras et al. (2021); Bronstein et al. (2021); Puny et al. (2022)). Moreover, they exist in many phenomena in the natural sciences. If a target  $M$  is invariant<sup>1</sup> when the operation  $g \in S$  designated by  $S$  applied to  $M$ ,  $M$  has symmetry  $S$ . For example, a sphere remains a sphere even if a rotation  $R$  is applied; a symmetric shape remains symmetric even if a left-right reversal is applied.

Recent studies have focused on methods incorporating symmetry into models based on equivariance and invariance, when the data space forms a non-Euclidean space (a sphere  $S^n$  or a special unitary group  $SU(n)$ ) (Cohen et al. (2018, 2019b); Graham et al. (2020); Haan et al. (2021); Boyda et al. (2021a)). Among them, discriminative and generative models for subspace data (e.g., shape matrices of point cloud data such as three-dimensional molecules and general object shapes) have been proposed as viable approaches using Grassmann manifolds, which are quotient manifolds obtained by introducing a function that induces invariance with respect to orthogonal transformations into

---

<sup>1</sup>Invariance implies the transformation  $\pi$  satisfies  $\pi(x) = \pi(x') = \pi(g(x))$  for  $x' = g(x)$  obtained by applying the operation  $g$  to the input  $x$ . Transformation  $\pi$  is considered invariant with respect to operation  $g$ . A transformation  $\pi$  is considered equivariant with respect to an operation  $p$  if  $\pi(g(x)) = \pi(g(x))$ . Invariance is a special case of equivariance where  $\pi(g)$  is an identity transformation.

a set of orthonormal basis matrices namely Stiefel Manifold. Numerous studies have confirmed their effectiveness (Liu et al. (2003); Hamm & Lee (2008); Turaga et al. (2008); Harandi et al. (2011); Fan et al. (2011); Lui (2012); Huang et al. (2015, 2018); Souza et al. (2020); Haitman et al. (2021); Doronina et al. (2022); Souza et al. (2022)). Shape theory studies the equivalent class of all configurations that can be obtained by a specific class of transformation (e.g. linear, affine, projective) on a single basis shape (Patrangenaru & Mardia (2003); Sepiashvili et al. (2003); Begelfor & Werman (2006)), and it can be shown that affine and linear shape spaces for specific configurations can be identified by points on the Grassmann manifold. By applying this model to inverse molecular design for stable shape generation to discover new molecular structures, they can contribute to the development of drug discovery, computational anatomy, and materials science (Sanchez-Lengeling & Aspuru-Guzik (2018); Bilodeau et al. (2022)).

Continuous normalizing flows (CNFs; Chen et al. (2018); Grathwohl et al. (2019); Lou et al. (2020); Kim et al. (2020); Mathieu & Nickel (2020)) are generative models that have attracted attention in recent years along with the variational auto-encoders (VAE; Kingma & Welling (2014)), generative adversarial networks (GAN; Goodfellow et al. (2014)), autoregressive models (Germain et al. (2015); Oord et al. (2016)), and diffusion models (Sohl-Dickstein et al. (2015); Ho et al. (2020); Song et al. (2021); Huang et al. (2022); De Bortoli et al. (2022)). The CNF is a method with theoretically superior properties that allow rigorous inference and evaluation of the log-likelihood. It can be trained by maximum likelihood using the change of variables formula, which allows to specify a complex normalized distribution implicitly by warping a normalized base distribution through an integral of continuous-time dynamics. In this paper, we present a novel approach with the explicit goal of generating stable shapes. We employ the CNF to achieve stable shape generation on a Grassmann manifold; however, previous studies have lacked the theoretical foundation to handle the CNF on the Grassmann manifold. To construct this, we focused on the quotient structure of a Grassmann manifold and translated the problem of flow learning on a Grassmann manifold into that of preserving equivariance for orthogonal groups on a Stiefel manifold, which is its total space. The contributions of this study are as follows.

- A theory and general framework for learning flows on a Grassmann manifold are proposed. In our setting, we can train flows on a Grassmann manifold of arbitrary dimension. To the best of our knowledge, this is the first study in which a CNF was constructed on a Grassmann manifold via a unified approach, focusing on the quotient structure.
- The validity of the proposed approach was demonstrated by learning densities on a Grassmann manifold using multiple artificial datasets with complex data distributions. In particular, orthogonally transformed data were proven to be correctly learned without data augmentations by showing that untrained transformed (i.e., rotated or mirrored) data can be generated from the trained model. The model was evaluated on multiple patterns of training data and performed well with a small amount of training data.
- The effectiveness of our approach was confirmed by its state-of-the-art performance in a molecular positional generation task.

## 2 Related Works

**Symmetry-based Learning** The concept of equivariance has been studied in recent years to leverage the symmetries inherent in data (Cohen & Welling (2017); Kondor & Trivedi (2018); Cohen et al. (2018, 2019b,a); Finzi et al. (2020); Haan et al. (2021)). Early work by Cohen & Welling (2017) indicated that when data are equivariant, they can be processed with a lower computational cost and fewer parameters. In the context of CNFs (Rezende & Mohamed (2015); Chen et al. (2018); Grathwohl et al. (2019)), which are generative models, Rezende et al. (2019), Köhler et al. (2020) and Garcia Satorras et al. (2021) proposed equivariant normalizing flows to learn symmetric densities on Euclidean spaces. Furthermore, the symmetries that appear in learning densities on a manifold were introduced by Boyda et al. (2021b) and Katsman et al. (2021) as a conjugate equivariant flow on  $SU(n)$ , which is a quotient manifold, for use in lattice gauge theory. However, a normalizing flow on a Grassmann manifold capable of handling subspace data has not yet been established.

**Shape Data Analysis and Other Applications with Subspace on the Grassmann Manifold**  $k$ -dimensional shape data (Begelfor & Werman (2006); Yoshinuma et al. (2016); Haitman et al. (2021);

Doronina et al. (2022)) such as point clouds are essentially represented as subspace (shape space (Sepiashvili et al. (2003); Srivastava et al. (2005); Begelfor & Werman (2006); Yataka & Fukui (2017)) data on a Grassmann manifold. Furthermore, subspace data can be obtained on many types of data and can provide advantages such as practicability and noise robustness. For example, multiview image sets or video data (Fan et al. (2011); Turaga et al. (2011); Lui (2012); Alashkar et al. (2016)), signal data (Srivastava & Klassen (2004); Gatto et al. (2017); Souza et al. (2019); Yataka et al. (2019)), and text data (Shimomoto et al. (2021)) are often provided as a set of feature vectors. Such raw data in matrix form is not very useful owing to its considerable size and noise. The analysis of its eigenspace, or column subspace, is important because alternatively, the raw data matrix can be well approximated by a low-dimensional subspace with basis vectors corresponding to the maximum eigenvalues of the matrix. Therefore, they inevitably give rise to the necessity of analyzing them on a Grassmann manifold.

### 3 Mathematical Preliminaries

In this section, the basic mathematical concepts covered in this paper are described. Further details on the fundamentals of a Grassmann manifold are summarized in the Appendix D.

#### 3.1 Grassmann Manifold Defined as Quotient Manifold

**Definition of a Grassmann Manifold** A Grassmann manifold  $\text{Gr}(k, D)$  is a set of  $k$ -dimensional subspaces  $\text{span}(\mathbf{Y})$  ( $\mathbf{Y}$  is a matrix of  $k$  basis vectors and the function  $\text{span}(\cdot)$  is onto  $\text{Gr}(k, D)$ ) in the  $D$ -dimensional Euclidean space  $\mathbb{R}^D$ , and is defined as  $\text{Gr}(k, D) = \{\text{span}(\mathbf{Y}) \subset \mathbb{R}^D \mid \dim(\text{span}(\mathbf{Y})) = k\}$  (Absil et al. (2008)). The  $\text{span}(\mathbf{Y})$  is the same subspace regardless of a  $k$ -dimensional rotation or  $k$ -dimensional reflection applied to  $\mathbf{Y}$  on which it is spanned. With respect to the compact Stiefel manifold  $\text{St}(k, D) := \{\mathbf{Y} \in \mathbb{R}^{D \times k} \mid \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k\}$  defined as the set of  $D \times k$ -orthonormal basis matrices  $\mathbf{Y}$ , the equivalence class of  $\mathbf{Y} \in \text{St}(k, D)$  determined from the equivalence relation  $\sim$ <sup>2</sup> is defined by  $[\mathbf{Y}] := \pi(\mathbf{Y}) = \{\mathbf{Y}\mathbf{Q} \in \text{St}(k, D) \mid \mathbf{Q} \in \mathcal{O}(k)\}$ , where  $\pi(\mathbf{Y})$  is a continuous surjection referred to as a quotient map. The equivalence class corresponds one-to-one with the  $k$ -dimensional subspace:

$$[\mathbf{Y}] = [\mathbf{X}] \iff \text{span}(\mathbf{Y}) = \text{span}(\mathbf{X}), \quad (1)$$

where  $\mathbf{Y} \in \text{St}(k, D)$  is the representative of  $[\mathbf{Y}]$ .  $\text{span}(\mathbf{Y})$  is termed invariant under  $\sim$ . The quotient set composed of such  $[\mathbf{Y}]$  as a whole can introduce the structure of a manifold (Sato & Iwai (2014)).

**Definition 1.** A Grassmann manifold as a quotient manifold is defined as follows:

$$\text{Gr}(k, D) := \text{St}(k, D) / \mathcal{O}(k) = \{[\mathbf{Y}] = \pi(\mathbf{Y}) \mid \mathbf{Y} \in \text{St}(k, D)\}, \quad (2)$$

where  $\text{St}(k, D) / \mathcal{O}(k)$  is the quotient manifold by the  $k$ -dimensional orthogonal group  $\mathcal{O}(k)$  with the total space  $\text{St}(k, D)$ , and  $\pi(\mathbf{Y})$  is the quotient map  $\pi : \text{St}(k, D) \rightarrow \text{St}(k, D) / \mathcal{O}(k)$ .

**Tangent Space and Vector Field** Let  $T_{[\mathbf{Y}]} \text{Gr}(k, D)$  be the tangent space of  $[\mathbf{Y}] \in \text{Gr}(k, D)$ . As the point  $[\mathbf{Y}]$  is not a matrix,  $T_{[\mathbf{Y}]} \text{Gr}(k, D)$  cannot be represented by a matrix. Therefore, treating these directly in numerical calculations is challenging. To solve this problem, we can use the representative  $\mathbf{Y} \in \text{St}(k, D)$  for  $[\mathbf{Y}]$  and the tangent vector  $\bar{\xi}_{\mathbf{Y}}^h \in T_{\mathbf{Y}}^h \text{St}(k, D)$ , which is referred to as the horizontal lift of  $\xi_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$ , for  $\xi_{[\mathbf{Y}]}$ . These facilitate computation with matrices (Absil et al. (2008)).  $T_{\mathbf{Y}}^h \text{St}(k, D)$  is a subspace of the tangent space  $T_{\mathbf{Y}} \text{St}(k, D)$  at  $\mathbf{Y}$ , which is referred to as a horizontal space, and  $\bar{\xi}_{\mathbf{Y}}^h$  is referred to as a horizontal vector. The tangent bundle  $T\text{Gr}(k, D) = \bigcup_{[\mathbf{Y}] \in \text{Gr}(k, D)} T_{[\mathbf{Y}]} \text{Gr}(k, D)$  that sums up the tangent spaces  $T_{[\mathbf{Y}]} \text{Gr}(k, D)$  form vector fields

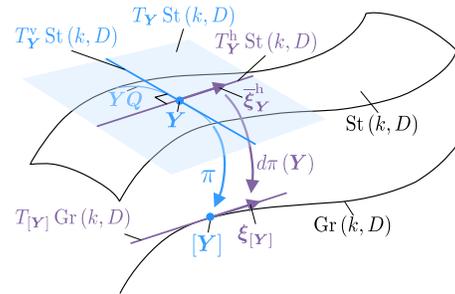


Figure 1: Conceptual diagram of spaces with horizontal lift.

<sup>2</sup>When there exists some  $\mathbf{Q} \in \mathcal{O}(k)$  such that  $\mathbf{X} = \mathbf{Y}\mathbf{Q}$ , then  $\mathbf{X}$  and  $\mathbf{Y}$  are defined to be equivalences  $\mathbf{X} \sim \mathbf{Y}$ . Appendix D.2 provides further details.

$X : \text{Gr}(k, D) \rightarrow T\text{Gr}(k, D)$ . A conceptual diagram of the various spaces is shown in Figure 1. Further details on these concepts can be found in Appendix D.4.

### 3.2 Manifold Normalizing Flow

Let  $(\mathcal{M}, h)$  be a Riemannian manifold. We consider the time evolution of a base point  $\mathbf{x} = \gamma(0)$ ,  $\gamma : [0, \infty) \rightarrow \mathcal{M}$ , whose velocity is expressed by a vector field  $X(t, \gamma(t))$ . Intuitively,  $X(t, \gamma(t))$  represents the direction and speed at which  $\mathbf{x}$  moves on the curve  $\gamma(t)$ . Let  $T_{\mathbf{x}}\mathcal{M}$  be the tangent space at  $\mathbf{x}$  and  $T\mathcal{M} = \bigcup_{\mathbf{x} \in \mathcal{M}} T_{\mathbf{x}}\mathcal{M}$  be the tangent bundle. The time evolution of a point according to a vector field  $X : \mathcal{M} \times \mathbb{R} \rightarrow T\mathcal{M}$  is expressed by the differential equation  $\frac{d\gamma(t)}{dt} = X(t, \gamma(t))$ ,  $\gamma(0) = \mathbf{x}$ . Let  $F_{X,T} : \mathcal{M} \rightarrow \mathcal{M}$  be defined as the map from  $\forall \mathbf{x} \in \mathcal{M}$  to the evaluated value at time  $T$  on the curve  $\gamma(t)$  starting at  $\mathbf{x}$ . This map  $F_{X,T}$  is known as the flow of  $X$  (Lee (2003)). Recently, Mathieu & Nickel (2020) introduced the Riemann CNF (RCNF), wherein the random variable  $\mathbf{z}(t) \in \mathcal{M}$  is assumed to be time-dependent and the change in its log-likelihood follows the instantaneous change formula for the variable. This is an extension of the CNF (Chen et al. (2018); Grathwohl et al. (2019)) to a Riemannian manifold. Specifically, when  $p_{\theta}$  is the density parameterized by  $\theta$ , the derivative of the log-likelihood is expressed as  $\frac{d \log p_{\theta}(\mathbf{z}(t))}{dt} = -\text{div}(X_{\theta}(t, \mathbf{z}(t)))$ , where  $X_{\theta}$  is the vector field parameterized by  $\theta$  and  $\text{div}(\cdot)$  is the divergence. By integrating this over time, the sum of the changes in the log-likelihood with flow  $F_{X,t_1\theta}$  can be computed:

$$\log p_{\theta}(\mathbf{z}(t_1)) = \log p\left(F_{X,t_1\theta}^{-1}(\mathbf{z}(t_1))\right) - \int_{t_0}^{t_1} \text{div}(X_{\theta}(t, \mathbf{z}(t))) dt. \quad (3)$$

## 4 Invariant Densities from Grassmann Manifold Flow

This section provides a tractable and efficient method for learning densities on a Grassmann manifold  $\text{Gr}(k, D)$ . The method for preserving the flow on  $\text{Gr}(k, D)$  is non-trivial. Therefore, we derive the following implications.

1. **Vector field on  $\text{Gr}(k, D) \Leftrightarrow$  Flow on  $\text{Gr}(k, D)$  (Proposition 1).**
2. **Flow on  $\text{Gr}(k, D) \Leftrightarrow$  Probability density on  $\text{Gr}(k, D)$  (Proposition 2).**
3. **Construction of a prior probability density for an efficient sampling (Proposition 3).**

The essence of the proofs is to show that Corollary 2 (Homogeneity Property (4) with regard to  $\forall Q \in \mathcal{O}(k)$ , which is equivariance in the horizontal space) is satisfied.

$$\bar{\xi}_{YQ}^h = \bar{\xi}_Y^h Q, \quad (4)$$

where  $\bar{\xi}_Y^h$  is a horizontal lift at representative  $Y$  relative to  $\xi_{[Y]} \in T_{[Y]}\text{Gr}(k, D)$ . We defer the proofs of all propositions to Appendix B. These series of propositions show that by using a prior distribution on  $\text{Gr}(k, D)$  that can be easily sampled, a flow that generates a complex probability density distribution on  $\text{Gr}(k, D)$  can be obtained.

### 4.1 Construction of Flow from Vector Field

To construct flows on  $\text{Gr}(k, D)$ , we use tools in the theory of manifold differential equations. In particular, there is a natural correspondence between the vector fields on  $\text{Gr}(k, D)$  and the flows on  $\text{Gr}(k, D)$ . This is formalized in the following proposition.

**Proposition 1.** *Let  $\text{Gr}(k, D)$  be a Grassmann manifold,  $X$  be any time-dependent vector field on  $\text{Gr}(k, D)$ , and  $F_{X,T}$  be a flow on a  $X$ . Let  $\bar{X}$  be any time-dependent horizontal lift and  $\bar{F}_{\bar{X},T}$  be a flow of  $\bar{X}$ .  $\bar{X}$  is a vector field on  $\text{St}(k, D)$  if and only if  $\bar{F}_{\bar{X},T}$  is a flow on  $\text{St}(k, D)$  and satisfies invariance condition  $\bar{X} \sim \bar{X}'$  for all  $\bar{F}_{\bar{X},T} \sim \bar{F}_{\bar{X}',T}$ . Therefore,  $X$  is a vector field on  $\text{Gr}(k, D)$  if and only if  $F_{X,T} := [\bar{F}_{\bar{X},T}]$  is a flow on  $\text{Gr}(k, D)$ , and vice versa.*

---

**Algorithm 1** Random Sampling from  $p_{\text{Gr}(k,D)}([\mathbf{X}]; [\mathbf{M}], \mathbf{U}, \mathbf{V})$ 


---

**Require:** A mean matrix  $\mathbf{M} \in \text{St}(k, D)$ , a rows covariance matrix  $\mathbf{U} \in \mathbb{R}^{D \times D}$  and a columns covariance matrix  $\mathbf{V} \in \mathbb{R}^{k \times k}$ .

- 1: Sample a  $\text{Vec}(\mathbf{Z}) \in \mathbb{R}^{Dk}$  from Gaussian Distribution  $\mathcal{N}_{Dk}(\mathbf{0}, \mathbf{V} \otimes \mathbf{U})$  and reshape to  $\mathbf{Z} \in \mathbb{R}^{D \times k}$ .
  - 2: Compute a projected horizontal vector  $\bar{\xi}_M^h = \mathbf{Z} - \mathbf{M}(\mathbf{M}^\top \mathbf{Z}) \in T_M^h \text{St}(k, D)$ .
  - 3: Compute a representative  $\mathbf{Y} = \bar{R}_M(\bar{\xi}_M^h) = \mathbf{M} + \bar{\xi}_M^h - \left(\frac{1}{2}\mathbf{M} + \frac{1}{4}\bar{\xi}_M^h\right) \left(\mathbf{I}_k + \frac{1}{4}\bar{\xi}_M^h \bar{\xi}_M^h\right)^{-1} \bar{\xi}_M^h \bar{\xi}_M^h$  of the equivalence class  $[\mathbf{Y}] \in \text{Gr}(k, D)$ .
- 

## 4.2 Construction of Probability Densities with Flow

We show that the flow on  $\text{Gr}(k, D)$  induces density on  $\text{Gr}(k, D)$ .

**Proposition 2.** *Let  $\text{Gr}(k, D)$  be a Grassmann manifold. Let  $p$  be the probability density on  $\text{Gr}(k, D)$  and  $F$  be the flow on  $\text{Gr}(k, D)$ . Suppose  $\bar{p}$  is a density on  $\text{St}(k, D)$  and  $\bar{F}$  is a flow on  $\text{St}(k, D)$ . Then, the distribution  $\bar{p}_{\bar{F}}$  after transformations by  $\bar{F}$  is also a density on  $\text{St}(k, D)$ . Further, the invariance condition  $\bar{p}_{\bar{F}} \sim \bar{p}_{\bar{F}'}$  is satisfied for all  $\bar{F} \sim \bar{F}'$ . Therefore,  $p_F := [\bar{p}_{\bar{F}}]$  is a distribution on  $\text{Gr}(k, D)$ .*

In the context of the RCNF, Proposition 2 implies that the application of a flow on  $\text{Gr}(k, D)$  to a prior distribution on  $\text{Gr}(k, D)$  results in a probability density on  $\text{Gr}(k, D)$ . Thus, the problem of constructing a probability density on  $\text{Gr}(k, D)$  is reduced to that of constructing a vector field.

## 4.3 Prior Probability Density Function

To construct a flow on  $\text{Gr}(k, D)$ , a prior distribution that is easy to sample as a basis for the transformation is required, although the method for constructing such a distribution is non-trivial. In this study, a distribution based on the matrix-variate Gaussian distribution is introduced as a prior on  $\text{Gr}(k, D)$  that is easy to sample, and a flow on  $\text{Gr}(k, D)$  is constructed.

**Proposition 3.** *The distribution  $p_{\text{Gr}(k,D)}$  on a Grassmann manifold  $\text{Gr}(k, D)$  based on the matrix-variate Gaussian distribution  $\mathcal{MN}$  can be expressed as follows.*

$$p_{\text{Gr}(k,D)}([\mathbf{X}]; [\mathbf{M}], \mathbf{U}, \mathbf{V}) = V_{\text{Gr}(k,D)} \mathcal{MN}(\bar{\xi}_M^h; \mathbf{0}, \mathbf{U}, \mathbf{V}) \left| \det \left( \nabla_{\bar{\xi}_M^h} \bar{R}_M \right) \right|, \quad (5)$$

where  $\mathbf{M}$  is an orthonormal basis matrix denoting the mean of the distribution,  $\mathbf{U}$  is a positive definite matrix denoting the row directional variance,  $\mathbf{V}$  is a positive definite matrix denoting the column directional variance, and  $\bar{\xi}_M^h$  is a random sample from  $\mathcal{MN}$  in an  $(D - k) \times k$ -dimensional horizontal space  $T_M^h \text{St}(k, D)$ .  $V_{\text{Gr}(k,D)}$  denotes the total volume of  $\text{Gr}(k, D)$  defined by (113),  $\bar{R}_M$  denotes the horizontal retraction at  $\mathbf{M}$ , and  $\left| \det \left( \nabla_{\bar{\xi}_M^h} \bar{R}_M \right) \right|$  denotes the Jacobian.

A retraction is a map for communicating data between a manifold and its tangent bundles and is a first-order approximation of an exponential map (Zhu & Sato (2021)). Various methods of retraction have been proposed at (Absil et al. (2008); Fiori et al. (2015); Zhu & Sato (2021)); in particular, the one based on the Cayley transform is differentiable and does not require matrix decomposition. We use (40) as a Cayley transform based horizontal retraction. Further details are provided in Appendix D.6.

## 5 Learning Probability Densities with Flow

### 5.1 Training Paradigms

Using the results of Section 4, a flow model on a Grassmann manifold  $\text{Gr}(k, D)$  is constructed. Herein, the flow model on  $\text{Gr}(k, D)$  is expressed as **GrCNEF**. In the proposed GrCNEF, the probability density function in Section 4.3 is first constructed as a prior distribution. Subsequently, the vector field  $\mathbb{X}_\theta : \text{Gr}(k, D) \times \mathbb{R} \rightarrow T\text{Gr}(k, D)$  that generates the flow model  $F_{\mathbb{X}_\theta, T}$  is constructed using a neural network, wherein stepwise integration on the manifold occurs in accordance with (3). The

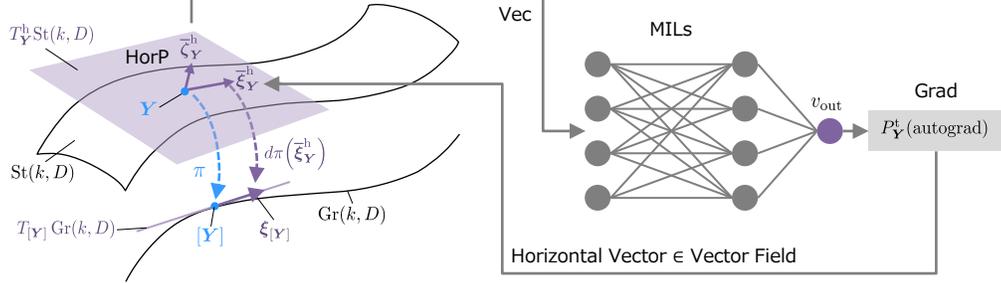


Figure 2: Conceptual diagram of the vector field calculation procedure. HorP denotes the horizontal projection layer, MILs denotes the multiple intermediate layers, and Grad denotes the horizontal lift layer. The sequence of procedure represents that, given an input of  $\mathbf{Y} \in [\mathbf{Y}]$ , a horizontal vector  $\bar{\xi}_{\mathbf{Y}}^h$ , which is equivalent to determining  $\xi_{[\mathbf{Y}]}$  since  $\bar{\xi}_{\mathbf{Y}}^h$  corresponds one-to-one with  $\xi_{[\mathbf{Y}]}$ , is obtained as a result.

RCNF (Mathieu & Nickel (2020)) framework is used for loss function, integration and divergence calculations. In the integration steps, the ordinary differential equation (ODE) is solved using the ODE solver with orthogonal integration. The learnable parameters are updated using backpropagation to maximize the sum of the computed log-likelihood. In order to efficiently compute gradients, we introduce an adjoint method to calculate analytically the derivative of a manifold ODE, based on the results of Lou et al. (2020).

## 5.2 Sampling Algorithm from Prior

We propose a sampling algorithm on  $\text{Gr}(k, D)$  derived from the Cayley transform, according to the results of Section 4.3. The algorithm of sampling from a distribution on  $p_{\text{Gr}(k, D)}$  is described in Algorithm 1. Vec denotes the map of vertically concatenating matrices and converting them into a vector and  $\mathbf{I}_D$  is a  $D \times D$  identity matrix.

## 5.3 Construction of Vector Field

We describe the method for constructing the vector field  $\mathbf{X}_\theta$ . The learnable parameters  $\theta$  are feedforward neural networks, which accept the representative  $\mathbf{Y} \in \text{St}(k, D)$  of  $[\mathbf{Y}] \in \text{Gr}(k, D)$  inputs and output a horizontal lift  $\bar{\xi}_{\mathbf{Y}}^h \in T_{\mathbf{Y}}^h \text{St}(k, D)$  of  $\xi_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$  that satisfies (4). The structure of  $\mathbf{X}_\theta$  is important because it directly determines the ability of the distribution to be represented. To address these geometric properties, the following specific input and intermediate layers are used to obtain a  $\mathcal{O}(k)$ -invariant function value  $v_{\text{out}}$ . Figure 2 shows a conceptual diagram.

**Input Layer** The input layer maps the input representative  $\mathbf{Y}$  to a point  $\bar{\zeta}_{\mathbf{Y}}^h \in T_{\mathbf{Y}}^h \text{St}(k, D)$ . A horizontal projection  $\text{HorP} : \text{St}(k, D) \rightarrow T_{\mathbf{Y}}^h \text{St}(k, D) \cong \mathbb{R}^{(D-k) \times k}$  is constructed as an input layer:

$$\text{HorP}(\mathbf{Y}) = \mathbf{W} - \mathbf{Y}\mathbf{Y}^\top \mathbf{W}, \quad (6)$$

where  $\mathbf{W}$  is the output of NN obtained with  $\mathbf{Y}\mathbf{Y}^\top$  as input. HorP is well-defined because it is uniquely determined regardless of how the representative  $\mathbf{Y} \in [\mathbf{Y}]$  is chosen.

**Intermediate Layer** To model the dynamics over the horizontal space, intermediate layers based on neural networks are constructed. The layers are particularly important for the ability to represent the distribution, and various configurations are possible such as concatsquash (CS) layer (Grathwohl et al. (2019)). The layer can be stacked multiply, and the overall power of expression increases with the number of layers. The input  $\mathbf{x}$  of the first intermediate layer is  $\mathbf{x} = \text{Vec} \circ \text{HorP}$ . The last intermediate layer (one layer before the output layer) must be constructed such that it exhibits a potential function value. This is to obtain the gradient for the potential function in the output layer.

**Output Layer** We construct a gradient  $\text{Grad} : (\mathbb{R}, \text{St}(k, D)) \rightarrow T_{\mathbf{Y}}^h \text{St}(k, D)$  as an output layer. In Grad,  $\bar{\xi}_{\mathbf{Y}}^h \in T_{\mathbf{Y}}^h \text{St}(k, D)$  is obtained by taking the gradient of the potential function  $v_{\text{out}}$ :

$$\text{Grad}(v_{\text{out}}, \mathbf{Y}) = P_{\mathbf{Y}}^t(\text{autograd}(v_{\text{out}}, \mathbf{Y})) \text{ s.t. } P_{\mathbf{Y}}^t(\mathbf{Z}) = \mathbf{Z} - \mathbf{Y} \text{sym}(\mathbf{Y}^\top \mathbf{Z}), \quad (7)$$

where  $\text{autograd}(v_{\text{out}}, \mathbf{Y})$  denotes automatic differentiation (Paszke et al. (2017))  $\frac{\partial v_{\text{out}}}{\partial \mathbf{Y}}$ ,  $P_{\mathbf{Y}}^t(\mathbf{Z})$  denotes the projection of  $\mathbf{Z}$  onto  $T_{\mathbf{Y}} \text{St}(k, D)$  and  $\text{sym}(\mathbf{B}) = (\mathbf{B} + \mathbf{B}^\top) / 2$  is a symmetric part of  $\mathbf{B}$ .  $\text{Grad}$  is well-defined on  $\text{Gr}(k, D)$  by the horizontal lift (Absil et al. (2008)).

#### 5.4 ODE Solver with Orthogonal Integration

We present an ordinary differential equation (ODE) solver on  $\text{Gr}(k, D)$ . Celledoni & Owren (2002) proposed an intrinsic ODE solver that was applicable to  $\text{St}(k, D)$  and did not assume an outer Euclidean space. This solver works on  $\text{St}(k, D)$ , thus it must be reformulated into a solver suitable for  $\text{Gr}(k, D)$ . We introduce a solver on  $\text{Gr}(k, D)$  via ODE operating on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$ , based on results from Section 4. This is an intrinsic approach regardless of whether the manifold has been embedded in a bigger space with a corresponding extension of the vector field. The ODE defined on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  is obtained as:

$$\frac{d\text{Vec}(\epsilon(t))}{dt} = \nabla_{\gamma} \bar{R}_{\mathbf{Y}}^{-1} \text{Vec}(\mathbf{X}_{\theta}(t, \gamma(t))) \text{ s.t. } \gamma(t) = \bar{R}_{\mathbf{Y}}(\epsilon(t)), \quad (8)$$

where  $\epsilon : [0, \infty) \rightarrow T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  is the curve on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$ ,  $\bar{R}_{\mathbf{Y}}$  denotes the horizontal retraction,  $\nabla_{\gamma} \bar{R}_{\mathbf{Y}}^{-1}$  can be calculated using (55) and  $\text{Vec}$  denotes the map of vertically concatenating matrices and converting them into a single vector. Further details on our ODE can be found in Appendix C.1.

#### 5.5 Loss Function

The total change in log-likelihood using GrCNF can be calculated using the following equation.

$$\log p_{\theta}(\mathbf{Y}(t_1)) = \log p_{\text{Gr}(k, D)}\left(F_{\mathbf{X}, t_1 \theta}^{-1}(\mathbf{Y}(t_1))\right) - \int_{t_0}^{t_1} \text{div}(\mathbf{X}_{\theta}(t, \mathbf{Y}(t))) dt, \quad (9)$$

where  $F_{\mathbf{X}, t_1 \theta}$  is a flow. In this study, we defined the loss function  $\text{Loss}$  for maximizing the log-likelihood:  $\text{Loss} = \text{NLL} = -\log p_{\theta}(\mathbf{Y}(t_1))$  where  $\text{NLL}$  denotes negative log-likelihood.

## 6 Experimental Results

In this section, we discuss the results of several experiments for validating GrCNF. Further details regarding each experimental condition, such as the architectures and hyperparameters used in training, can be found in Appendix C.3.

### 6.1 Generation and Density Estimation on Artificial Textures

We first trained GrCNF on five different  $\text{Gr}(1, 3)$  (1-dimensional subspace in  $\mathbb{R}^3$ , that is, a line through the origin) data to visualize the model and the trained dynamics. The five datasets<sup>3</sup> were 2 spirals, a swissroll, 2 circles, 2 sines, and Target. We represented  $\text{Gr}(1, 3)$  with a sphere of radius 1 centered at the origin by mapping a 1-dimensional subspace to two points on the sphere (a point on the sphere and its antipodal point) for visualization.

**Result** The five types of probability densities transformed by GrCNF and the results of data generation are shown in Figure 3. The top, middle, and bottom rows in the figure present the correct data, the generated data with GrCNF when the left-most column is the prior distribution, and the probability density on a  $\text{Gr}(1, 3)$  obtained by training when the leftmost column was the prior distribution, respectively. Regarding the probability density, a brighter region corresponds to a higher probability. As shown in the figure, GrCNF can generate high-quality samples that are sufficiently accurate to the distribution of the correct data. To investigate whether GrCNF learns the distribution on  $\text{Gr}(1, 3)$ , we generated antipodal points in Figure 3 with the trained GrCNF used in Figure 3. As shown in Figure 4, GrCNF generated exactly the same high quality samples as the original for the untrained antipodal points. This experimental result implies that the proposed flow accurately learns the distribution on a  $\text{Gr}(k, D)$  and that all the orthogonal transformed samples can be obtained with equal quality by only training with arbitrary representatives.

<sup>3</sup>The code for the data distributions is presented in Appendix C.3.1

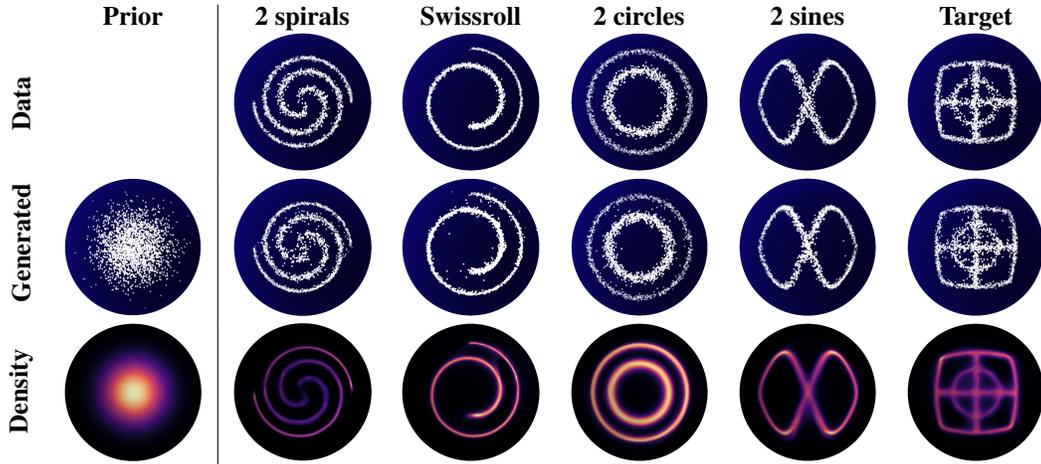


Figure 3: Generated samples and probability densities using the GrCNF trained on each of the five distributions. (top) Ground truth data, (middle) Generated data with the GrCNF when the leftmost column represents the prior distribution, and (bottom) densities obtained via training.

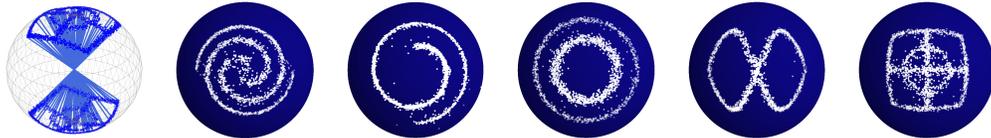


Figure 4: Results of the GrCNF transformation for the antipodal point of the prior in the middle panel of the Figure 3. The antipodal points are those that pass through the origin of the sphere and appear in the southern hemisphere (leftmost). It is evident that the antipodal points are generated with exactly the same quality as in the Figure 3 for the un-trained antipodal points.

## 6.2 Comparison with Conventional Methods for DW4 and LJ13

Comparative experiments were performed using DW4 and LJ13—the two systems presented by Köhler et al. (2020). These datasets were generated synthetically by sampling from their respective energy functions using Markov chain Monte Carlo (MCMC) methods. Both energy functions (DW4 and LJ13) are ideal for analyzing the advantages of various methods when they exist on data that are equivariant for rotation and reflection, respectively. We used the datasets generated by MCMC that were employed in Köhler et al. (2020). The orthonormalized data  $\mathbf{Y}$  were used for this study by applying Gram-Schmidt’s orthogonalization to each column of  $\mathbf{P}$  such that the  $D$ -dimensional data was a matrix with  $k$  orthonormal basis vectors (DW4:  $D = 4, k = 2$ ; LJ13:  $D = 13, k = 3$ ).

To match the experiment conducted in Garcia Satorras et al. (2021), 1,000 validation and testing samples each were used for both datasets. For DW4 and LJ13, different numbers of training samples, i.e.,  $\{10^2, 10^3, 10^4, 10^5\}$  and  $\{10, 10^2, 10^3, 10^4\}$ , respectively, were selected, and their performance for each amount of data was examined. The proposed approach was compared with the state-of-the-art  $E(n)$  equivariant flow (E-NF) presented by Garcia Satorras et al. (2021) and simple dynamics presented by Köhler et al. (2020). In addition, comparisons with graph normalizing flow (GNF), GNF with attention (GNF-att), and GNF with attention and data augmentation (GNF-att-aug) (data augmentation by rotation), which are non-equivariant variants of E-NF, were performed. All the reported values are averages of cross-validations (three runs). Otherwise, the network structures of all these conventional methods were the same as that used in Garcia Satorras et al. (2021).

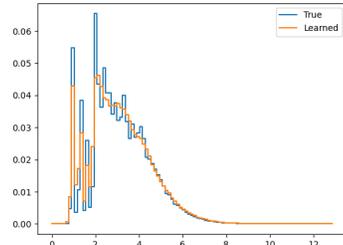
**Result** Table 1 presents the results of the cross-validated experiments (negative log-likelihood; NLL) for the test samples. The proposed GrCNF outperformed both the conventional non-equivariant models (GNF, GNF-att, and GNF-att-aug) and the conventional equivariant models (Simple dynamics, E-NF) in all data domains.

Table 1: Negative log-likelihood comparison on the test partition of different methods for DW4 and LJ13 datasets for different amount of training samples averaged over 3 runs.

# of Samples	DW4				LJ13			
	$10^2$	$10^3$	$10^4$	$10^5$	10	$10^2$	$10^3$	$10^4$
<b>GNF</b>	-2.30	-7.04	-7.19	-7.93	6.77	-0.76	-4.26	-12.43
<b>GNF-att</b>	-2.02	-4.13	-5.25	-6.74	6.91	1.40	-6.81	-12.05
<b>GNF-att-aug</b>	-3.11	-4.04	-6.51	-9.42	2.95	-6.11	-13.94	-15.74
<b>Simple dynamics</b>	-1.22	-1.28	-1.36	-1.39	-1.10	-3.87	-3.72	-3.59
<b>E-NF</b>	-0.54	-9.89	-12.15	-15.29	-12.86	-15.75	-31.51	-32.83
<b>GrCNF</b>	<b>-12.53</b>	<b>-13.74</b>	<b>-14.09</b>	<b>-16.07</b>	<b>-23.64</b>	<b>-44.24</b>	<b>-58.02</b>	<b>-58.71</b>

Table 2: Results for the QM9 Positional dataset. (Left) Negative log-likelihood,  $-\text{ELBO}$  and JSD for the QM9 Positional dataset on the test data, and (Right) normalized histogram of relative distances between atoms for QM9 Positional and generated samples with GrCNF.

	NLL	$-\text{ELBO}$	JSD
<b>Simple dynamics</b>	73.0	-	0.086
<b>Kernel dynamics</b>	38.6	-	0.029
<b>GNF</b>	-00.9	-	0.011
<b>GNF-att</b>	-26.6	-	0.007
<b>GNF-att-aug</b>	-33.5	-	0.006
<b>E-NF</b>	-70.2	-	0.006
<b>GrCNF</b>	NLL $\leq$ <b>-85.3</b>	-85.3	0.005



### 6.3 Comparison with Conventional Methods for QM9 Positional

A comparative experiment was performed using the QM9 Positional—a subset of the QM9 molecular dataset that considers only positional information. The purpose of this experiment was to evaluate the feasibility of generating a practical point cloud. The QM9 Positional comprises only molecules with 19 atoms/nodes, and each node has a 3-dimensional position vector associated with it. However, the likelihood of the molecules must be invariant to translation and rotation in 3-dimensional space; thus, the proposed model is suitable for this type of data. The dataset consisted of 13,831, 2501, and 1813 training, validation, and testing samples, respectively. In this experiment, we used evaluation metrics based on the NLL and Jensen-Shannon divergence (JSD; Lin (1991)), in accordance with the experiments conducted in Garcia Satorras et al. (2021). The JSD calculates the distance between the normalized histograms that are generated from the model and obtained from the training set, by creating a histogram of the relative distances among all the node pairs in each molecule.

GrCNF handles subspace data. Therefore, orthonormalized data  $\mathbf{Y}$  was used only in the proposed GrCNF, as in  $\mathbf{P}\mathbf{P}^\top \simeq \mathbf{Y}\mathbf{\Lambda}\mathbf{Y}^\top$  s.t.  $\mathbf{\Lambda}$  is diagonal (Huang et al. (2015)), such that the  $k$ -dimensional point cloud data  $\mathbf{P}$  of  $N$  points  $N \times k$  matrix is a matrix with  $k$  orthonormal basis vectors ( $D = 19$ ,  $k = 3$ ). However, a complete point cloud generating task, such as QM9 Positional, must also store a scale parameter  $\sqrt{\mathbf{\Lambda}}$  such that  $\mathbf{P} = \mathbf{Y}\sqrt{\mathbf{\Lambda}}$ . Therefore, GrCNF incorporates an additional architecture to estimate  $\sqrt{\mathbf{\Lambda}}$ . In addition, the proposed GrCNF encounters difficulties in computing the NLL in distribution  $p_\psi(\mathbf{P})$  of  $\mathbf{P}$ ; thus, a new loss function is required to address this. In this study, the evidence lower bound (ELBO), i.e., the lower bound of the log-likelihood  $\log p_\psi(\mathbf{P})$ , was maximized using a variational inference framework. This is equal to the minimization of  $-\text{ELBO}(\mathbf{P})$ . Further details regarding ELBO can be found in Appendix C.2.

We compared our GrCNF with the GNF, GNF-att, GNF-att-aug, Simple dynamics, Kernel dynamics, and E-NF methods. The network structure and experimental conditions for all these conventional methods were identical to those used in Garcia Satorras et al. (2021). In all the experiments, the training was performed for 160 epochs. The JSD values were the averages of the last 10 epochs for all the models.

**Result** Table 2 presents the NLL and JSD cross-validated against the test data. Although the proposed GrCNF could not directly compute the NLL in this problem setting, it outperformed all

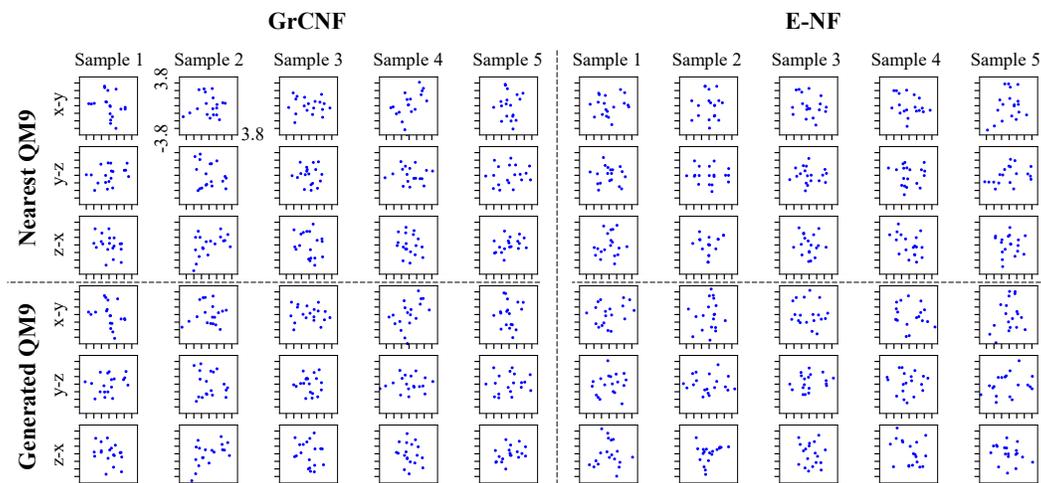


Figure 5: Visualization results of generated samples in Section 6.3. We visualize Generated QM9 by GrCNF and E-NF, as well as the aligned data points from the QM9 dataset that are closest to the Generated QM9 samples. These visualizations illustrate that GrCNF is capable of generating realistic data points that align well with the existing dataset.

the other algorithms because the relation  $NLL \leq -ELBO$  is true in general. With regard to the JSD, GrCNF achieved the best performance. Figure 2 presents the normalized histograms of the QM9 Positional molecular data and the data generated by GrCNF, in relation to the JSD of GrCNF. As shown, the histograms of the molecules generated by GrCNF are close to the histograms of the data set; thus, GrCNF can generate realistic molecules stably.

## 7 Conclusion

We proposed the concept of CNF on a Grassmann manifold (GrCNF) for stable shape generation. The proposed model is a generative model capable of handling subspace data; i.e., it is a CNF that considers the equivariance on the Stiefel manifold. Through suitable experiments, the ability of the proposed GrCNF to generate qualitatively in 1-dimensional subspace datasets in  $\mathbb{R}^3$  was confirmed. Further, GrCNF significantly outperformed existing normalizing flows methods in terms of the log-likelihood or ELBO for DW4, LJ13, and QM9 Positional. It was shown that GrCNF can be used to generate realistic data more stably.

**Societal Impact and Limitations** Our work is concerned with accurately modeling the data topology; thus, we do not expect there to be any negative consequence in our application field. The proposed theory and implementation are valid in a remarkably general setting, although there are limitations that can be addressed in future works: **1.** The proposed method requires calculation of the Kronecker product at each ODE step (Section 5.4), which is computationally expensive. **2.** Our current paper does not include experiments on high-dimensional data. Therefore, future experiments should be conducted using higher dimensional data, such as point clouds that represent general object geometry. However, the results presented in this paper still provide numerous valuable insights in the field of stable shape generation.

## References

- P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3. URL <https://press.princeton.edu/absil>.
- Taleb Alashkar, Boulbaba Ben Amor, Mohamed Daoudi, and Stefano Berretti. A grassmann framework for 4d facial shape analysis. *Pattern Recognition*, 57:21–30, 2016. ISSN 0031-3203. URL <https://doi.org/10.1016/j.patcog.2016.03.013>.
- E. Begelfor and M. Werman. Affine invariance revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 2087–2094. IEEE Computer Society, 2006. URL <https://doi.org/10.1109/CVPR.2006.50>.
- Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F. Jensen. Generative models for molecular discovery: Recent advances and challenges. *WIREs Computational Molecular Science*, pp. e1608, 2022. URL <https://doi.org/10.1002/wcms.1608>.
- Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S. Albergo, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Sampling using  $su(n)$  gauge equivariant flows. *Phys. Rev. D*, 103(7):074504, 2021a. URL <https://doi.org/10.1103/PhysRevD.103.074504>.
- Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S. Albergo, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Sampling using  $su(n)$  gauge equivariant flows. *Physical Review D*, 103(7), 2021b. URL <https://link.aps.org/doi/10.1103/PhysRevD.103.074504>.
- Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021. URL <https://arxiv.org/abs/2104.13478>.
- Elena Celledoni and Brynjulf Owren. A class of intrinsic schemes for orthogonal integration. *SIAM Journal on Numerical Analysis*, 40(6):2069–2084, 2002. URL <http://www.jstor.org/stable/4100984>.
- Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf>.
- Yasuko Chikuse. *Statistics on Special Manifolds. Lecture Notes in Statistics*, volume 174. Springer, 01 2003. URL <https://doi.org/10.1007/978-0-387-21540-2>.
- Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1321–1330. PMLR, 2019a. URL <https://proceedings.mlr.press/v97/cohen19d.html>.
- Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2990–2999. PMLR, 2016. URL <https://proceedings.mlr.press/v48/cohenc16.html>.
- Taco S. Cohen and Max Welling. Steerable cnns. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=rJQKYt511>.
- Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=Hkdb5xZrb>.
- Taco S. Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL <https://proceedings.neurips.cc/paper/2019/file/b9cfe8b6042cf759dc4c0cccb27a6737-Paper.pdf>.

- Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2406–2422. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/105112d52254f86d5854f3da734a52b4-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/105112d52254f86d5854f3da734a52b4-Paper-Conference.pdf).
- R. J. Dormand and J. P. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, pp. 19–26, 1980. URL [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3).
- Olga Doronina, Zachary Grey, and Andrew Glaws. Grassmannian shape representations for aerodynamic applications. In *AAAI 2022 Workshop on AI for Design and Manufacturing (ADAM)*, 2022. URL <https://openreview.net/forum?id=1RRU6ud9YC>.
- Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. URL <https://doi.org/10.1137/S0895479895290954>.
- Lawrence Craig Evans and Garipey Ronald. *Measure Theory and Fine Properties of Functions, Revised Edition (1st ed.)*. CRC Press, Revised Edition, 2015. URL <https://doi.org/10.1201/b18333>.
- Yong Fan, Yong Liu, Hong Wu, Yihui Hao, Haihong Liu, Zhening Liu, and Tianzi Jiang. Discriminant analysis of functional connectivity patterns on grassmann manifold. *NeuroImage*, 56(4):2058–2067, 2011. URL <https://doi.org/10.1117/12.844495>.
- Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3165–3176, 2020. URL <https://dl.acm.org/doi/10.5555/3524938.3525235>.
- Simone Fiori, Tetsuya Kaneko, and Toshihisa Tanaka. Tangent-bundle maps on the grassmann manifold: Application to empirical arithmetic averaging. *IEEE Transactions on Signal Processing*, 63(1):155–168, 2015. URL <https://doi.org/10.1109/TSP.2014.2365764>.
- Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling.  $E(n)$  equivariant normalizing flows. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4181–4192. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/21b5680d80f75a616096f2e791affac6-Paper.pdf>.
- Bernardo B. Gatto, Juan G. Colonna, Eulanda M. dos Santos, and Eduardo F. Nakamura. Mutual singular spectrum analysis for bioacoustics classification. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2017. URL <https://doi.org/10.1109/MLSP.2017.8168113>.
- Evan S. Gawlik and M. Leok. High-order retractions on matrix manifolds using projected polynomials. *SIAM J. Matrix Anal. Appl.*, 39:801–828, 2018a. URL <https://doi.org/10.1137/17M1130459>.
- Evan S. Gawlik and Melvin Leok. High-order retractions on matrix manifolds using projected polynomials. *SIAM Journal on Matrix Analysis and Applications*, 39(2):801–828, 2018b. URL <https://doi.org/10.1137/17M1130459>.
- Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 881–889. PMLR, 2015. URL <http://proceedings.mlr.press/v37/germain15.html>.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.

- Simon Graham, David B. A. Epstein, and Nasir M. Rajpoot. Dense steerable filter cnns for exploiting rotational symmetry in histology images. *IEEE Trans. Medical Imaging*, 39(12):4124–4136, 2020. URL <https://doi.org/10.1109/TMI.2020.3013246>.
- Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=rJxgknCcK7>.
- Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=Jnspzp-oIZE>.
- Ernst Hairer. *Geometric Numerical Integration : Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd ed. edition, 2006. URL <https://doi.org/10.1007/3-540-30666-8>.
- Yuval Haitman, Joseph M. Francos, and Louis L. Scharf. Grassmannian dimensionality reduction for optimized universal manifold embedding representation of 3d point clouds. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 4196–4204, 2021. URL <https://doi.org/10.1109/ICCVW54120.2021.00468>.
- Jihun Hamm and Daniel D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pp. 376–383. Association for Computing Machinery, 2008. URL <https://doi.org/10.1145/1390156.1390204>.
- Mehrtash T. Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C. Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR 2011*, 2011. URL <https://doi.org/10.1109/CVPR.2011.5995564>.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf>.
- Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2750–2761. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/123d3e814e257e0781e5d328232ead9b-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/123d3e814e257e0781e5d328232ead9b-Paper-Conference.pdf).
- Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 140–149, 2015. URL <https://doi.org/10.1109/CVPR.2015.7298609>.
- Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 3279–3286, 2018. URL <https://dl.acm.org/doi/abs/10.5555/3504035.3504436>.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015. URL <https://proceedings.mlr.press/v37/loffef15.html>.
- Arieh Iserles, Hans Z. Munthe-Kaas, Syvert P. Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000. URL <https://doi.org/10.1017/S0962492900002154>.
- Isay Katsman, Aaron Lou, Derek Lim, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Equivariant manifold flows. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10600–10612. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/581b41df0cd50ace849e061ef74827fc-Paper.pdf>.

- Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. Softflow: Probabilistic framework for normalizing flow on manifolds. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16388–16397. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/dbdca288fee7f92f2bfa9f7012727740-Paper.pdf>.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. URL <http://arxiv.org/abs/1312.6114>.
- Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370. PMLR, 2020. URL <http://proceedings.mlr.press/v119/kohler20a.html>.
- Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2747–2755. PMLR, 2018. URL <https://proceedings.mlr.press/v80/kondor18a.html>.
- J.M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2003. ISBN 9780387954486.
- J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. URL <https://doi.org/10.1109/18.61115>.
- Xiuwen Liu, A. Srivastava, and K. Gallivan. Optimal linear representations of images for object recognition. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pp. I–I, 2003. URL <https://doi.org/10.1109/CVPR.2003.1211358>.
- Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. Neural manifold ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17548–17558. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/cbf8710b43df3f2c1553e649403426df-Paper.pdf>.
- Yui Man Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6–7): 380–388, 2012. URL <https://doi.org/10.1016/j.imavis.2011.08.002>.
- J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. Wiley, 2019. URL <https://doi.org/10.1002/9781119541219>.
- Arak M. Mathai, Serge B. Provost, and Hans J. Haubold. The matrix-variate gaussian distribution. In *Multivariate Statistical Analysis in the Real and Complex Domains*, pp. 217–288. Springer International Publishing, 2022. URL [https://doi.org/10.1007/978-3-030-95864-0\\_4](https://doi.org/10.1007/978-3-030-95864-0_4).
- Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/1aa3d9c6ce672447e1e5d0f1b5207e85-Paper.pdf>.
- Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003. URL <https://doi.org/10.1137/S00361445024180>.
- Hans Munthe-Kaas. High order runge-kutta methods on manifolds. *Applied Numerical Mathematics*, 29(1):115–127, 1999. URL [https://doi.org/10.1016/S0168-9274\(98\)00030-0](https://doi.org/10.1016/S0168-9274(98)00030-0).
- Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 1747–1756. PMLR, 2016. URL <http://proceedings.mlr.press/v48/oord16.html>.

- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017. URL <https://openreview.net/forum?id=BJJsrnfCZ>.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>.
- Vic Patrangenaru and Kanti V. Mardia. Affine shape analysis and image analysis. In *22nd Leeds Annual Statistics Research Workshop*, 2003. URL <http://www1.maths.leeds.ac.uk/Statistics/workshop/lasr2003/proceedings/patrangenaru.pdf>.
- Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. URL [https://openreview.net/forum?id=ywxtmG1nU\\_6](https://openreview.net/forum?id=ywxtmG1nU_6).
- Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017. URL <http://arxiv.org/abs/1710.05941>.
- Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538. PMLR, 2015. URL <https://proceedings.mlr.press/v37/rezende15.html>.
- Danilo Jimenez Rezende, Sébastien Racanière, Irina Higgins, and Peter Toth. Equivariant hamiltonian flows, 2019. URL <https://doi.org/10.48550/arXiv.1909.13739>.
- Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. URL <https://www.science.org/doi/abs/10.1126/science.aat2663>.
- Hiroyuki Sato and Toshihiro Iwai. Optimization algorithms on the grassmann manifold with application to matrix eigenvalue problems. *Japan Journal of Industrial and Applied Mathematics*, 31: 355–400, 2014. URL <https://doi.org/10.1007/s13160-014-0141-9>.
- Víctor Garcia Satorras, Emiel Hooeboom, and Max Welling.  $e(n)$  equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332. PMLR, 2021. URL <https://proceedings.mlr.press/v139/satorras21a.html>.
- D. Sepiashvili, J.M.F. Moura, and V.H.S. Ha. Affine-permutation symmetry: invariance and shape space. In *IEEE Workshop on Statistical Signal Processing, 2003*, pp. 307–310, 2003. URL <https://doi.org/10.1109/SSP.2003.1289406>.
- Erica K. Shimomoto, François Portet, and Kazuhiro Fukui. Text classification based on the word subspace representation. *Pattern Analysis and Applications*, 24(3):1075–1093, 2021. URL <https://doi.org/10.1007/s10044-021-00960-6>.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015. URL <https://proceedings.mlr.press/v37/sohl-dickstein15.html>.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=St1giarCHLP>.

- Lincon S. Souza, Bernardo B. Gatto, and Kazuhiro Fukui. Classification of bioacoustic signals with tangent singular spectrum analysis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355, 2019. URL <https://doi.org/10.1109/ICASSP.2019.8682493>.
- Lincon S. Souza, Bernardo B. Gatto, Jing-Hao Xue, and Kazuhiro Fukui. Enhanced grassmann discriminant analysis with randomized time warping for motion recognition. *Pattern Recognition*, 97:107028, 2020. URL <https://doi.org/10.1016/j.patcog.2019.107028>.
- Lincon S. Souza, Naoya Sogi, Bernardo B. Gatto, Takumi Kobayashi, and Kazuhiro Fukui. Grassmannian learning mutual subspace method for image set recognition. *Neurocomputing*, 2022. ISSN 0925-2312. URL <https://doi.org/10.1016/j.neucom.2022.10.040>.
- A. Srivastava, S.H. Joshi, W. Mio, and Xiuwen Liu. Statistical shape analysis: clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):590–602, 2005. URL <https://doi.org/10.1109/TPAMI.2005.86>.
- Anuj Srivastava and Eric Klassen. Bayesian and geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, 2004. URL <http://www.jstor.org/stable/1428352>.
- Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. URL <https://doi.org/10.1109/CVPR.2008.4587733>.
- Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011. URL <https://doi.org/10.1109/TPAMI.2011.52>.
- Maurice Weiler and Gabriele Cesa. General  $e(2)$ -equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf>.
- Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142:397–434, 2013. URL <https://doi.org/10.1007/s10107-012-0584-1>.
- Zhu Xiaojing. A riemannian conjugate gradient method for optimization on the stiefel manifold. *Computational Optimization and Applications*, 67:73–110, 2017. URL <https://doi.org/10.1007/s10589-016-9883-4>.
- Ryoma Yataka and Kazuhiro Fukui. Three-dimensional object recognition via subspace representation on a grassmann manifold. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, pp. 208–216. SciTePress, 2017. URL <https://doi.org/10.5220/0006204702080216>.
- Ryoma Yataka, Kazuki Hirashima, Takafumi Matsuda, Tai Tanaka, Masato Gocho, and Masashi Shiraiishi. Projection metric learning of updated-subspaces for radar target classification. In *2019 16th European Radar Conference (EuRAD)*, pp. 5–8, 2019. URL <https://ieeexplore.ieee.org/abstract/document/8904617>.
- Takao Yoshinuma, Hideitsu Hino, and Kazuhiro Fukui. Personal authentication based on 3d configuration of micro-feature points on facial surface. In *Image and Video Technology*, pp. 433–446. Springer International Publishing, 2016. URL [https://doi.org/10.1007/978-3-319-29451-3\\_35](https://doi.org/10.1007/978-3-319-29451-3_35).
- Xiaojing Zhu and Chun-Yan Duan. On matrix exponentials and their approximations related to optimization on the stiefel manifold. *Optimization Letters*, 13, 07 2019. URL <https://doi.org/10.1007/s11590-018-1341-z>.
- Xiaojing Zhu and Hiroyuki Sato. Cayley-transform-based gradient and conjugate gradient algorithms on grassmann manifolds. *Adv. Comput. Math.*, 47(4), 2021. URL <https://doi.org/10.1007/s10444-021-09880-9>.

## A Appendix Overview

First, we present the details of the various proofs of Section 4 in Appendix B. Next, in Appendix C, we describe the network layers for the building GrCNF, and the detailed model architectures, hyper-parameters, and implementation on the experiments. Finally, in Appendix D, we provide a summary of the fundamentals of a Grassmann manifold, which is the core concept of this study.

## B Proofs

### B.1 Proposition 1

First, we invoked the following two corollaries.

**Corollary 1** (Diffeomorphism Invariance of Flows). *Let  $F : \mathcal{M} \rightarrow \mathcal{N}$  be a diffeomorphism. If  $X$  is a smooth vector field over  $\mathcal{M}$  and  $\theta$  is the flow of  $X$ , then the flow of  $F_*X$ <sup>4</sup> is  $\eta_t = F \circ \theta_t \circ F^{-1}$ , with domain  $N_t = F(M_t)$  for each  $t \in \mathbb{R}$ .*

*Proof.* See Lee (2003, Corollary 9.14). □

**Corollary 2** (Homogeneity Property). *The horizontal lift  $\bar{\xi}_Y^h$  at representative  $Y \in \text{St}(k, D)$  relative to  $\xi_{[Y]} \in T_{[Y]} \text{Gr}(k, D)$  satisfies the following homogeneity (equivariance) property (4) with regard to  ${}^\forall Q \in \mathcal{O}(k)$ .*

$$\bar{\xi}_{YQ}^h = \bar{\xi}_Y^h Q.$$

*Proof.*  $\pi(Y) = \pi(YQ)$  is true for  ${}^\forall Y \in \text{St}(k, D)$ ,  $Q \in \mathcal{O}(k)$ . Therefore,  $\pi(Y) = (\pi \circ q)(Y)$  is true when defined as  $q(Y) = YQ$ . When the derivative  $d\pi(\cdot)[\cdot]$  of both sides is applied to the horizontal lift  $\bar{\xi}_Y^h$  of  $\xi_{[Y]}$ , the following is obtained:

$$d\pi(Y) \left[ \bar{\xi}_Y^h \right] = d(\pi \circ q)(Y) \left[ \bar{\xi}_Y^h \right] = d\pi(q(Y)) \left[ dq(Y) \left[ \bar{\xi}_Y^h \right] \right] = d\pi(YQ) \left[ \bar{\xi}_{YQ}^h \right]. \quad (10)$$

Moreover, from (98) which is definition of horizontal lift, the following equation is true.

$$\xi_{[Y]} = d\pi(Y) \left[ \bar{\xi}_Y^h \right] = d\pi(YQ) \left[ \bar{\xi}_{YQ}^h \right]. \quad (11)$$

Subsequently, we obtain the following equation.

$$\xi_{[Y]} = d\pi(YQ) \left[ \bar{\xi}_{YQ}^h \right] = d\pi(YQ) \left[ \bar{\xi}_Y^h Q \right]. \quad (12)$$

Finally, the uniqueness of the horizontal lift yields  $\bar{\xi}_{YQ}^h = \bar{\xi}_Y^h Q$ . □

**Proposition 1.** *Let  $\text{Gr}(k, D)$  be a Grassmann manifold,  $X$  be any time-dependent vector field on  $\text{Gr}(k, D)$ , and  $F_{X,T}$  be a flow on a  $X$ . Let  $\bar{X}$  be any time-dependent horizontal lift and  $\bar{F}_{\bar{X},T}$  be a flow of  $\bar{X}$ .  $\bar{X}$  is a vector field on  $\text{St}(k, D)$  if and only if  $\bar{F}_{\bar{X},T}$  is a flow on  $\text{St}(k, D)$  and satisfies invariance condition  $\bar{X} \sim \bar{X}'$  for all  $\bar{F}_{\bar{X},T} \sim \bar{F}_{\bar{X}',T}$ . Therefore,  $X$  is a vector field on  $\text{Gr}(k, D)$  if and only if  $F_{X,T} := \left[ \bar{F}_{\bar{X},T} \right]$  is a flow on  $\text{Gr}(k, D)$ , and vice versa.*

*Proof.* **Flow  $F_{X,T}$  on  $\text{Gr}(k, D) \Rightarrow$  Vector Field  $X$  on  $\text{Gr}(k, D)$ .** Let  $\theta : \text{St}(k, D) \times \mathcal{O}(k) \rightarrow \text{St}(k, D)$ ,  $(Y, Q) \mapsto YQ$  be a map representing the right action of the orthogonal group. In addition, let  $F_{X,T}$  be a flow on  $\text{Gr}(k, D)$  and  $\bar{F}_{\bar{X},T}$  be a flow on  $\text{St}(k, D)$ . These satisfy  $\bar{F}_{\bar{X},T} \sim$

<sup>4</sup> $F_*$  denotes the pushforward, that is, another notation for the differential of  $F$ .

$$\bar{F}_{\bar{X},T}, \bar{F}_{\bar{X}Q,T} \in F_{X,T}, \bar{F}_{\bar{X},T} \in F_{X,T}.$$

$$\bar{X} \left( t, \bar{F}_{\bar{X}Q,t}(\mathbf{Y}Q) \right) = \bar{X} \left( t, \bar{F}_{\bar{X},t}(\mathbf{Y})Q \right) \quad (13)$$

$$= \frac{d}{dt} \left\{ \bar{F}_{\bar{X},t}(\mathbf{Y})Q \right\} \quad (14)$$

$$= \frac{d}{dt} \left( \theta \circ \bar{F}_{\bar{X},t} \right) (\mathbf{Y}) \quad (15)$$

$$= d(\theta)_{\mathbf{Y}} \left\{ \frac{d}{dt} \bar{F}_{\bar{X},t}(\mathbf{Y}) \right\} \quad (16)$$

$$= d(\theta)_{\mathbf{Y}} \left\{ \bar{X} \left( t, \bar{F}_{\bar{X},t}(\mathbf{Y}) \right) \right\} \quad (17)$$

$$= \bar{X} \left( t, \bar{F}_{\bar{X},t}(\mathbf{Y}) \right) Q. \quad (18)$$

Thus,  $\bar{X} \sim \bar{X}Q$  is true. Therefore,  $\bar{X}$  is the horizontal lift of the vector field  $X$  on  $\text{Gr}(k, D)$  and is unique for  $X$ .

**Flow  $F_{X,T}$  on  $\text{Gr}(k, D) \Leftarrow$  Vector Field  $X$  on  $\text{Gr}(k, D)$ .** Let  $\theta : \text{St}(k, D) \times \mathcal{O}(k) \rightarrow \text{St}(k, D), (\mathbf{Y}, Q) \mapsto \mathbf{Y}Q$  be a map representing the right action of the orthogonal group. In addition, let  $\bar{X}$  be a vector field over a horizontal bundle  $T^{\text{h}} \text{St}(k, D)$  on  $\text{St}(k, D)$  and  $\bar{F}_{\bar{X},T}$  be its flow. From the Corollary 1,

$$\bar{F}_{\theta_* \circ \bar{X}, T} = \theta \circ \bar{F}_{\bar{X}, T} \circ \theta^{-1} \quad (19)$$

$$\bar{F}_{\theta_* \circ \bar{X}, T} \circ \theta = \theta \circ \bar{F}_{\bar{X}, T} \quad (20)$$

$$\bar{F}_{d(\theta)_{\mathbf{Y}} \bar{X}, T} \circ \theta = \theta \circ \bar{F}_{\bar{X}, T} \quad (21)$$

$$\bar{F}_{\bar{X}Q, T}(\mathbf{Y}Q) = \bar{F}_{\bar{X}, T}(\mathbf{Y})Q. \quad (22)$$

Note that  $d(\theta)_{\mathbf{Y}} \bar{X} = \bar{X}Q$  is derived from the Corollary 2 and (4) in Zhu & Sato (2021). This indicates that  $\bar{F}_{\bar{X}, T} \sim \bar{F}_{\bar{X}', T}$  is true for any  $\bar{X}, \bar{X}' \in T^{\text{h}} \text{St}(k, D)$  that satisfies  $\bar{X} \sim \bar{X}'$ . Thus, a new flow can be defined as  $F_{X,T} := \left[ \bar{F}_{\bar{X}, T} \right]$ . This is a flow on a  $\text{Gr}(k, D)$ . Because  $\bar{X}$  is a vector field in a horizontal bundle  $T^{\text{h}} \text{St}(k, D)$  on  $\text{St}(k, D)$ , it is a horizontal lift of the vector field  $X$  on  $\text{Gr}(k, D)$  and is therefore unique for  $X$ .

Thus, the proof is complete.  $\square$

## B.2 Proposition 2

**Proposition 2.** Let  $\text{Gr}(k, D)$  be a Grassmann manifold. Let  $p$  be the probability density on  $\text{Gr}(k, D)$  and  $F$  be the flow on  $\text{Gr}(k, D)$ . Suppose  $\bar{p}$  is a density on  $\text{St}(k, D)$  and  $\bar{F}$  is a flow on  $\text{St}(k, D)$ . Then, the distribution  $\bar{p}_{\bar{F}}$  after transformations by  $\bar{F}$  is also a density on  $\text{St}(k, D)$ . Further, the invariance condition  $\bar{p}_{\bar{F}} \sim \bar{p}_{\bar{F}'}$  is satisfied for all  $\bar{F} \sim \bar{F}'$ . Therefore,  $p_F := \left[ \bar{p}_{\bar{F}} \right]$  is a distribution on  $\text{Gr}(k, D)$ .

*Proof.* Let  $\theta : \text{St}(k, D) \times \mathcal{O}(k) \rightarrow \text{St}(k, D) : (\mathbf{Y}, \mathbf{Q}) \mapsto \mathbf{Y}\mathbf{Q}$  be a map representing the right action of the orthogonal group.

$$\bar{p}_F(\theta \circ \mathbf{Y}) = \bar{p}_F(\theta \circ \mathbf{Y}) \frac{|\det \{J_\theta(\mathbf{Y})\}|}{|\det \{J_\theta(\mathbf{Y})\}|} = \frac{\bar{p}_{\theta^{-1} \circ F}(\mathbf{Y})}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (23)$$

$$= \bar{p}((F^{-1} \circ \theta)(\mathbf{Y})) \frac{|\det \{J_{F^{-1} \circ \theta}(\mathbf{Y})\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (24)$$

$$= (\bar{p} \circ F^{-1}) \circ \theta(\mathbf{Y}) \frac{|\det \{J_{\theta \circ F^{-1}}(\mathbf{Y})\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (25)$$

$$= \theta \circ (\bar{p} \circ F^{-1})(\mathbf{Y}) \frac{|\det \{J_\theta(F^{-1}(\mathbf{Y})) J_{F^{-1}}(\mathbf{Y})\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (26)$$

$$= \theta \circ (\bar{p} \circ F^{-1})(\mathbf{Y}) \frac{|\det \{J_\theta(F^{-1}(\mathbf{Y}))\}| |\det \{J_{F^{-1}}(\mathbf{Y})\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (27)$$

$$= \theta \circ \bar{p}(F^{-1})(\mathbf{Y}) |\det \{J_{F^{-1}}(\mathbf{Y})\}| \frac{|\det \{J_\theta(F^{-1}(\mathbf{Y}))\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (28)$$

$$= \theta \circ \bar{p}_F(\mathbf{Y}) \frac{|\det \{J_\theta(F^{-1}(\mathbf{Y}))\}|}{|\det \{J_\theta(\mathbf{Y})\}|} \quad (29)$$

$$= \theta \circ \bar{p}_F(\mathbf{Y}), \quad (30)$$

where  $|\det \{J_\theta(\mathbf{X})\}| = 1$  is true because  $\theta$  is the action of the orthogonal group. Therefore, as  $\bar{p}_F(\theta \circ \mathbf{Y}) = \theta \circ \bar{p}_F(\mathbf{Y})$  is true,  $\bar{p}_F \sim \bar{p}_F \mathbf{Q} \in p_F$  is true. Based on this, it can be concluded that the subject is satisfied.  $\square$

### B.3 Proposition 3

**Proposition 3.** *The distribution  $p_{\text{Gr}(k, D)}$  on a Grassmann manifold  $\text{Gr}(k, D)$  based on the matrix-variate Gaussian distribution  $\mathcal{MN}$  can be expressed as follows.*

$$p_{\text{Gr}(k, D)}([\mathbf{X}]; [\mathbf{M}], \mathbf{U}, \mathbf{V}) = V_{\text{Gr}(k, D)} \mathcal{MN}(\bar{\boldsymbol{\xi}}_M^h; \mathbf{0}, \mathbf{U}, \mathbf{V}) \left| \det \left( \nabla_{\bar{\boldsymbol{\xi}}_M^h} \bar{R}_M \right) \right|, \quad (31)$$

where  $\mathbf{M}$  is an orthonormal basis matrix denoting the mean of the distribution,  $\mathbf{U}$  is a positive definite matrix denoting the row directional variance,  $\mathbf{V}$  is a positive definite matrix denoting the column directional variance, and  $\bar{\boldsymbol{\xi}}_M^h$  is a random sample from  $\mathcal{MN}$  in an  $(D - k) \times k$ -dimensional horizontal space  $T_M^h \text{St}(k, D)$ .  $V_{\text{Gr}(k, D)}$  denotes the total volume of  $\text{Gr}(k, D)$  defined by (113),  $\bar{R}_M$  denotes the horizontal retraction at  $\mathbf{M}$ , and  $\left| \det \left( \nabla_{\bar{\boldsymbol{\xi}}_M^h} \bar{R}_M \right) \right|$  denotes the Jacobian.

*Proof.* Let  $p_{\text{Gr}}([\mathbf{X}])$  be a probability density function on a  $\text{Gr}(k, D)$ . From (101), let  $(d\mathbf{X})$  be the invariant measure on  $\text{Gr}(k, D)$  and  $d\bar{\boldsymbol{\xi}}_M^h$  be the Lebesgue measure on  $T_M^h \text{St}(k, D)$ . Subsequently, a change of variables was performed according to the following:

$$p_{\text{Gr}(k, D)}([\mathbf{X}]) (d\mathbf{X}) = p_{\text{Gr}(k, D)} \left( \left[ \bar{R}_M \left( \bar{\boldsymbol{\xi}}_M^h \right) \right] \right) d\bar{\boldsymbol{\xi}}_M^h \quad (32)$$

$$p_{\text{Gr}(k, D)}([\mathbf{X}]) = p_{\text{Gr}(k, D)} \left( \left[ \bar{R}_M \left( \bar{\boldsymbol{\xi}}_M^h \right) \right] \right) \left| \det \left( \frac{d\bar{\boldsymbol{\xi}}_M^h}{d\bar{R}_M} \right) \right| \quad (33)$$

$$p_{\text{Gr}(k, D)}([\mathbf{X}]) = p_{\text{Gr}(k, D)} \left( \left[ \bar{R}_M \left( \bar{\boldsymbol{\xi}}_M^h \right) \right] \right) \left| \det \left( \nabla_{\bar{\boldsymbol{\xi}}_M^h} \bar{R}_M \right) \right|^{-1}. \quad (34)$$

Suppose  $p_{\text{Gr}}([\mathbf{X}])$  is integrable with the probability measure  $[d\mathbf{X}]$  on  $\text{Gr}(k, D)$  defined by (114). Then, we obtain the following relation.

$$\int_{\text{Gr}(k, D)} p_{\text{Gr}(k, D)}([\mathbf{X}]) [d\mathbf{X}] \quad (35)$$

$$= \frac{1}{V_{\text{Gr}(k, D)}} \int_{\text{Gr}(k, D)} p_{\text{Gr}(k, D)}([\mathbf{X}]) (d\mathbf{X}) \quad (36)$$

$$= \frac{1}{V_{\text{Gr}(k, D)}} \int_{T_M^{\text{h}} \text{St}(k, D)} p_{\text{Gr}(k, D)}\left(\left[\bar{R}_M\left(\bar{\xi}_M^{\text{h}}\right)\right]\right) \left|\det\left(\nabla_{\bar{\xi}_M^{\text{h}}}\bar{R}_M\right)\right|^{-1} d\bar{\xi}_M^{\text{h}}. \quad (37)$$

In addition, we obtain the following equation based on  $[d\mathbf{X}]$ .

$$\int_{\text{Gr}(k, D)} p_{\text{Gr}(k, D)}([\mathbf{X}]) [d\mathbf{X}] = \int_{T_M^{\text{h}} \text{St}(k, D)} \mathcal{MN}\left(\bar{\xi}_M^{\text{h}}\right) d\bar{\xi}_M^{\text{h}} = 1, \quad (38)$$

where  $\mathcal{MN}\left(\bar{\xi}_M^{\text{h}}\right)$  denotes the matrix-variate Gaussian distribution (Mathai et al. (2022)). Thus, the probability density function on  $\text{Gr}(k, D)$  can be expressed as follows:

$$p_{\text{Gr}(k, D)}\left(\left[\bar{R}_M\left(\bar{\xi}_M^{\text{h}}\right)\right]\right) = V_{\text{Gr}(k, D)} \mathcal{MN}\left(\bar{\xi}_M^{\text{h}}\right) \left|\det\left(\nabla_{\bar{\xi}_M^{\text{h}}}\bar{R}_M\right)\right|. \quad (39)$$

The Jacobian can be represented as  $\left|\det\left(\nabla_{\bar{\xi}_M^{\text{h}}}\bar{R}_M\right)\right| = \left|\det\left\{\left(\partial\bar{R}_M\right)^\top\left(\partial\bar{R}_M\right)\right\}\right|^{\frac{1}{2}}$  from Evans & Ronald (2015), where  $\partial\bar{R}_M = \frac{\partial\bar{R}_M}{\partial\bar{\xi}_M^{\text{h}}}$ . Further,  $\partial\bar{R}_M$  can be computed as follows. First, we define the horizontal retraction  $\bar{R}_Y : T_Y^{\text{h}} \text{St}(k, D) \rightarrow \text{St}(k, D)$  based on the Cayley transform from Zhu & Sato (2021).

$$\mathbf{X} = \bar{R}_M\left(\bar{\xi}_M^{\text{h}}\right) = \mathbf{M} + \bar{\xi}_M^{\text{h}} - \left(\frac{1}{2}\mathbf{M} + \frac{1}{4}\bar{\xi}_M^{\text{h}}\right) \left(\mathbf{I}_k + \frac{1}{4}\bar{\xi}_M^{\text{h}\top}\bar{\xi}_M^{\text{h}}\right)^{-1} \bar{\xi}_M^{\text{h}\top}\bar{\xi}_M^{\text{h}}. \quad (40)$$

This is a fixed time ( $t = 1$ ) version of (122). Next, for improved visibility in subsequent calculations, let  $\mathbf{E} = \bar{\xi}_M^{\text{h}}$ ,  $\mathbf{F} = \frac{1}{2}\mathbf{M} + \frac{1}{4}\mathbf{E}$ ,  $\mathbf{G} = \left(\mathbf{I}_k + \frac{1}{4}\mathbf{H}\right)^{-1}$ ,  $\mathbf{H} = \mathbf{E}^\top\mathbf{E}$ . In addition, let  $\mathbf{D}$  be defined as the operator for the derivative of a matrix by a matrix. Then, the derivative  $\nabla_{\bar{\xi}_M^{\text{h}}}\bar{R}_M = \mathbf{D}\mathbf{X}$  by  $\mathbf{E}$  is as follows:

$$\nabla_{\bar{\xi}_M^{\text{h}}}\bar{R}_M = \mathbf{D}\mathbf{X} = \mathbf{D}\mathbf{M} + \mathbf{D}\mathbf{E} - \mathbf{D}(\mathbf{F}\mathbf{G}\mathbf{H}). \quad (41)$$

Finally, each derivative can be calculated as follows:

$$\mathbf{D}\mathbf{M} = \mathbf{0}, \quad (42)$$

$$\mathbf{D}\mathbf{E} = \mathbf{I}_{Dk}, \quad (43)$$

$$\mathbf{D}(\mathbf{F}\mathbf{G}\mathbf{H}) = \mathbf{D}(\mathbf{F}(\mathbf{G}\mathbf{H})) \quad (44)$$

$$= \left\{(\mathbf{G}\mathbf{H})^\top \otimes \mathbf{I}_D\right\} \mathbf{D}\mathbf{F} + (\mathbf{I}_k \otimes \mathbf{F}) \mathbf{D}(\mathbf{G}\mathbf{H}) \quad (45)$$

$$= \left\{(\mathbf{G}\mathbf{H})^\top \otimes \mathbf{I}_D\right\} \mathbf{D}\mathbf{F} + (\mathbf{I}_k \otimes \mathbf{F}) \left\{(\mathbf{H}^\top \otimes \mathbf{I}_k) \mathbf{D}\mathbf{G} + (\mathbf{I}_k \otimes \mathbf{G}) \mathbf{D}\mathbf{H}\right\} \quad (46)$$

$$= \left\{(\mathbf{G}\mathbf{H})^\top \otimes \mathbf{I}_D\right\} \mathbf{D}\mathbf{F} + (\mathbf{I}_k \otimes \mathbf{F}) (\mathbf{H}^\top \otimes \mathbf{I}_k) \mathbf{D}\mathbf{G} + (\mathbf{I}_k \otimes \mathbf{F}) (\mathbf{I}_k \otimes \mathbf{G}) \mathbf{D}\mathbf{H} \quad (47)$$

$$= (\mathbf{G}^\top \mathbf{H}^\top \otimes \mathbf{I}_D) \mathbf{D}\mathbf{F} + (\mathbf{H}^\top \otimes \mathbf{F}) \mathbf{D}\mathbf{G} + (\mathbf{I}_k \otimes \mathbf{F}\mathbf{G}) \mathbf{D}\mathbf{H}, \quad (48)$$

$$\mathbf{D}\mathbf{F} = \mathbf{D}\left(\frac{1}{2}\mathbf{M}\right) + \mathbf{D}\left(\frac{1}{4}\mathbf{E}\right) = \frac{1}{4}\mathbf{D}\mathbf{E}, \quad (49)$$

$$\mathbf{D}\mathbf{G} = -(\mathbf{G}^\top \otimes \mathbf{G}) \mathbf{D}\mathbf{H}, \quad (50)$$

$$\mathbf{D}\mathbf{H} = (\mathbf{I}_{k^2} + \mathbf{K}_{k,k}) (\mathbf{I}_k \otimes \mathbf{E}^\top) \mathbf{D}\mathbf{E}, \quad (51)$$

where  $\otimes$  denotes the Kronecker product.  $\mathbf{K}_{D,k}$  is a  $Dk \times Dk$  matrix  $\mathbf{K}_{D,k} = \sum_{i=1}^m \sum_{j=1}^n (\mathbf{L}_{i,j} \otimes \mathbf{L}_{i,j}^\top)$  referred to as the commutation matrix, which denotes the transposition operation of  $D \times k$ . Further,  $\mathbf{L}_{i,j}$  is a  $D \times k$  matrix whose  $(i, j)$  component is 1 whereas all other components are 0.  $\square$

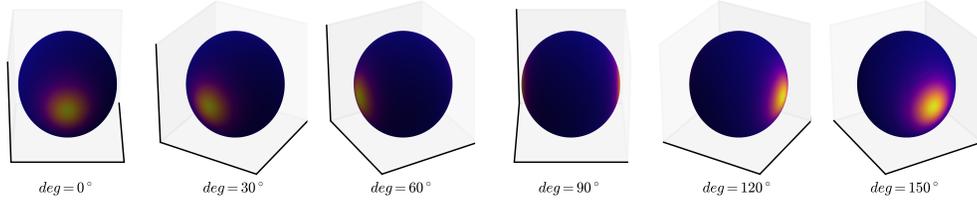


Figure 6:  $p_{\text{Gr}(1,3)}([\mathbf{X}])$  with  $\mathbf{M} = (1.0, 0.0, 0.0)^\top$ ,  $\mathbf{U} = \sigma^2 \mathbf{I}_3$ ,  $\mathbf{V} = \mathbf{I}_1$ ,  $\sigma^2 = 0.5$ . Each sphere in the figure indicates  $\text{Gr}(1, 3)$ , with brighter spheres representing higher densities.

For details on the formulae for matrix derivatives used in this proof, please refer to Magnus & Neudecker (2019).

$p_{\text{Gr}(k,D)}([\mathbf{X}]) = p_{\text{Gr}(k,D)}([\mathbf{X}]; [\mathbf{M}], \mathbf{U}, \mathbf{V})$  is a probability distribution following mean  $[\mathbf{M}]$  and matrix variance  $\mathbf{U}, \mathbf{V}$ . Using  $\text{Gr}(1, 3)$  as an example, we qualitatively confirmed through visualization that  $p_{\text{Gr}(1,3)}([\mathbf{X}])$  is a density on  $\text{Gr}(1, 3)$ .  $\text{Gr}(1, 3)$  is a 1-dimensional subspace in a 3-dimensional space; that is, a space whose elements are lines passing through the origin in 3-dimensional space. For the visualization, we expressed  $\text{Gr}(1, 3)$  by mapping a 1-dimensional subspace to two points on a sphere (one point on the sphere and its antipodal point) of radius 1 centered at the origin.

Figure 6 shows the density of  $p_{\text{Gr}(1,3)}([\mathbf{X}])$  with  $\mathbf{M} = (1.0, 0.0, 0.0)^\top$ ,  $\mathbf{U} = \sigma^2 \mathbf{I}_3$ ,  $\mathbf{V} = \mathbf{I}_1$ ,  $\sigma^2 = 0.5$ . Each sphere in the figure indicates  $\text{Gr}(1, 3)$ , with brighter spheres representing higher densities. The leftmost figure shows  $\mathbf{M}$  as viewed from the front diagonally above, and the other figures present the views when the viewpoint is rotated clockwise around the  $z$ -axis by  $30^\circ$  to  $150^\circ$  with movement to the right. In the leftmost figure, the density is highly spread around  $\mathbf{M}$ . In the other figures (particularly the rightmost one), the antipodal point  $(-\mathbf{M} = (-1.0, 0.0, 0.0)^\top)$  is densely spread out. This implies that when only one  $\mathbf{M}$  is specified as the representative of the equivalence class  $[\mathbf{M}]$ , the density around the other elements in the equivalence class  $[\mathbf{M}]$  is as high as that around the representative. Thus, we can confirm that  $p_{\text{Gr}(k,D)}([\mathbf{X}])$  has a density of  $\text{Gr}(1, 3)$ .

## C Experimental Details for Learning GrCNF

### C.1 Details on ODE Solver with Orthogonal Integration

We explain in detail an ordinary differential equation (ODE) solver on  $\text{Gr}(k, D)$ . Several studies on ODE solvers employed on a manifold  $\mathcal{M}$  have been reported (Munthe-Kaas (1999); Iserles et al. (2000); Hairer (2006)). Hairer (2006) proposed a simple projection method that projected onto the manifold at each step and the symmetric projection method suitable for long-time integration. However, this method requires  $\mathcal{M}$  to be a submanifold in Euclidean space, and thus cannot be applied on  $\text{Gr}(k, D)$ . In contrast, Celledoni & Owren (2002) proposed an intrinsic ODE solver that was applicable to a Stiefel manifold and did not assume an outer Euclidean space. This solver works on a Stiefel manifold, thus it must be reformulated into a solver suitable for  $\text{Gr}(k, D)$ , which is our problem setting. We introduce below a solver on  $\text{Gr}(k, D)$  via ODE operating on the horizontal space  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$ , based on results from Celledoni & Owren (2002) and Section 4. This is an intrinsic approach regardless of whether  $\mathcal{M}$  has been embedded in a larger space with a corresponding extension of the vector field.

First, consider an ODE expressed using a vector field  $\mathbf{X}_{\theta}$  on a curve  $\gamma(t) : [0, \infty) \rightarrow \text{St}(k, D)$  such that for each time  $t$ .

$$\frac{d\gamma(t)}{dt} = \mathbf{X}_{\theta}(t, \gamma(t)), \quad \gamma(0) = \mathbf{Y}, \quad (52)$$

where  $\mathbf{X}_{\theta}$  is constructed by a neural network with parameter  $\theta$ , as described in Section 5.3. Horizontal retraction  $\bar{R}_{\mathbf{Y}}$  which is defined as (122) and is described in Appendix D.6, defines local coordinates of  $\text{Gr}(k, D)$  in a neighborhood of the point  $[\mathbf{Y}]$ . Thus, the solution of ODE can be expressed as:

$$\gamma(t) = \bar{R}_{\mathbf{Y}}(\epsilon(t)), \quad (53)$$

where  $\epsilon : [0, \infty) \rightarrow T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  is the curve on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$ . By differentiating (53) with  $t$ , the following equation is obtained:

$$\frac{d\gamma(t)}{dt} = \frac{d}{dt} \bar{R}_{\mathbf{Y}}(\epsilon(t)) = \mathbf{X}_{\theta}(t, \gamma(t)). \quad (54)$$

Therefore, the ODE defined on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  is obtained as (8):

$$\begin{aligned} \frac{d\text{Vec}(\epsilon(t))}{dt} &= (\nabla_{\epsilon} \bar{R}_{\mathbf{Y}})^{-1} \text{Vec}(\mathbf{X}_{\theta}(t, \bar{R}_{\mathbf{Y}}(\epsilon(t)))) \\ &= \nabla_{\gamma} \bar{R}_{\mathbf{Y}}^{-1} \text{Vec}(\mathbf{X}_{\theta}(t, \bar{R}_{\mathbf{Y}}(\epsilon(t)))), \end{aligned}$$

where  $\text{Vec}$  denotes the map of vertically concatenating matrices and converting them into a single vector.  $\nabla_{\gamma} \bar{R}_{\mathbf{Y}}^{-1}$  can be calculated using the derivative of (123):

$$\nabla_{\gamma} \bar{R}_{\mathbf{Y}}^{-1} = 2(\mathbf{N}^{-\top} \otimes \mathbf{I}_D) \nabla M + 2(\mathbf{I}_k \otimes M) \nabla \mathbf{N}^{-1}, \quad (55)$$

where  $\mathbf{X} = \gamma(t)$ ,  $M = \mathbf{Y} - \mathbf{X}\mathbf{X}^{\top}\mathbf{Y}$ ,  $\mathbf{N} = \mathbf{I}_k + \mathbf{X}^{\top}\mathbf{Y}$ ,  $\nabla M = \mathbf{I}_{Dk} - (\mathbf{I}_k \otimes \mathbf{X}\mathbf{X}^{\top})$  and  $\nabla \mathbf{N}^{-1} = -(\mathbf{N}^{-\top} \otimes \mathbf{N}^{-1})(\mathbf{I}_k \otimes \mathbf{X}^{\top})$ . Because (8) is an ODE on  $T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D) \cong \mathbb{R}^{(D-k) \times k}$  from Absil et al. (2008), it can be solved using an ODE solver such as Runge-Kutta methods that operate on Euclidean space. This study used Algorithm 5.1 presented in Celledoni & Owren (2002). In each step, first, (8) was solved using the Runge-Kutta method of order 5 as in Dormand & Prince (1980). Subsequently, the solution of ODE (52) was obtained by applying the solution  $\epsilon$  to (53).

### C.2 Loss Function based on Variational Inference

In the setting in Section 6.3, we used orthonormalized data  $\mathbf{Y}$  as in  $\mathbf{P}\mathbf{P}^{\top} \simeq \mathbf{Y}\mathbf{\Lambda}\mathbf{Y}^{\top}$ , where  $\mathbf{\Lambda}$  is diagonal (Huang et al. (2015)), such that the  $k$ -dimensional point cloud data  $\mathbf{P}$  of  $N$  points  $N \times k$  matrix is a matrix with  $k$  orthonormal basis vectors. Thus, generating a complete point cloud requires the estimation of the scale parameters  $\sqrt{\mathbf{\Lambda}}$  to be  $\mathbf{P} = \mathbf{Y}\sqrt{\mathbf{\Lambda}}$  and a loss function that incorporates this. In this study, we approximated by maximizing the evidence lower bound (ELBO), which is the lower bound of the overall log-likelihood  $\log p_{\psi}(\mathbf{P})$  of  $p_{\psi}(\mathbf{P})$ , using a variational inference framework. The loss function is the variational energy  $-\text{ELBO}(\mathbf{P})$  with negative ELBO.

$$\text{NLL} = -\log p_{\psi}(\mathbf{P}) \leq -\text{ELBO}(\mathbf{P}) = \text{Loss}. \quad (56)$$

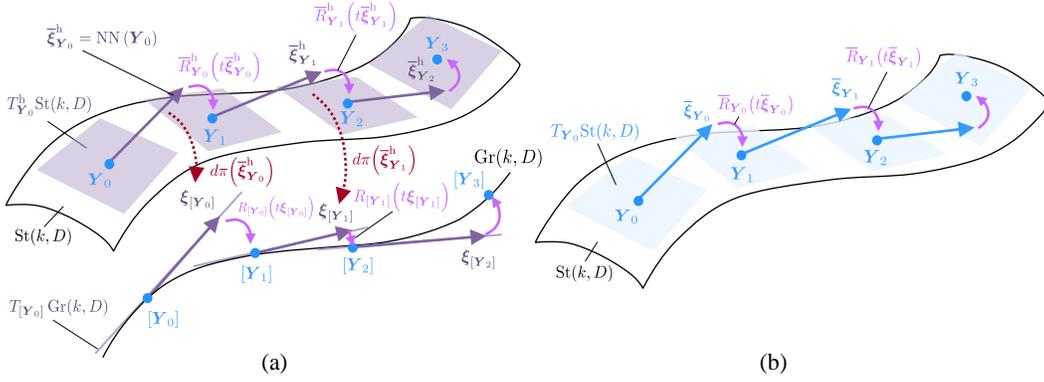


Figure 7: The proposed ODE solver on the Grassmann manifold and the ODE solver on the Stiefel manifold (Celledoni & Owren (2002)). (a) The proposed ODE solver working on a Grassmann manifold. The  $\bar{\xi}_{\mathbf{Y}}^h$  obtained by the proposed neural architecture NN is the horizontal lift (98), that is,  $d\pi(\mathbf{Y}) \left[ \bar{\xi}_{\mathbf{Y}}^h \right] = \xi_{[\mathbf{Y}]}$ , of the tangent vector  $\xi_{[\mathbf{Y}]}$  at the point  $[\mathbf{Y}]$  on the Grassmann manifold. The proposed ODE solver maps and updates the tangent vector  $\xi_{[\mathbf{Y}]}$  at the point  $[\mathbf{Y}]$  onto the Grassmann manifold at each step using horizontal retraction (121). On the other hand, (b) the solver of Celledoni & Owren (2002), working on Stiefel manifolds, updates in each step by mapping the tangent vector  $\bar{\xi}_{\mathbf{Y}}$  at the point  $\mathbf{Y}$  onto the Stiefel manifold. In other words, the difference between the proposed ODE solver and the ODE solver on Stiefel manifolds is that the ODE solver on Stiefel manifolds works only on Stiefel manifolds, while the proposed ODE solver always updates in each step with the Stiefel manifold and Grassmann manifolds linked together.

ELBO ( $\mathbf{P}$ ) can be decomposed as follows.

$$\text{ELBO}(\mathbf{P}) = \log p_{\psi}(\mathbf{P}) - D_{KL}(q_{\phi}(\mathbf{Y} | \mathbf{P}) || p_{\psi}(\mathbf{Y} | \mathbf{P})) \quad (57)$$

$$= \mathbb{E}_{q_{\phi}(\mathbf{Y} | \mathbf{P})} [\log p_{\psi}(\mathbf{P} | \mathbf{Y})] - D_{KL}(q_{\phi}(\mathbf{Y} | \mathbf{P}) || p_{\theta}(\mathbf{Y})), \quad (58)$$

where  $q_{\phi}(\mathbf{Y} | \mathbf{P})$  is the inference model with parameter  $\phi$ ,  $p_{\psi}(\mathbf{P} | \mathbf{Y})$  is the decoder model with parameter  $\psi$ ,  $p_{\psi}(\mathbf{Y} | \mathbf{P})$  is the posterior distribution with parameter  $\psi$ , and  $p_{\theta}(\mathbf{Y})$  is the prior distribution with parameter  $\theta$ . Further,  $D_{KL}(q_{\phi}(\mathbf{Y} | \mathbf{P}) || p_{\theta}(\mathbf{Y}))$  can be formulated using differential entropy as follows.

$$D_{KL}(q_{\phi}(\mathbf{Y} | \mathbf{P}) || p_{\theta}(\mathbf{Y})) = -\mathbb{E}_{q_{\phi}(\mathbf{Y} | \mathbf{P})} [p_{\theta}(\mathbf{Y})] - H[q_{\phi}(\mathbf{Y} | \mathbf{P})]. \quad (59)$$

Thus, the final loss function is as follows.

$$\text{Loss} = -\text{ELBO}(\mathbf{P}) \quad (60)$$

$$= -\mathbb{E}_{q_{\phi}(\mathbf{Y} | \mathbf{P})} [\log p_{\psi}(\mathbf{P} | \mathbf{Y})] - \mathbb{E}_{q_{\phi}(\mathbf{Y} | \mathbf{P})} [p_{\theta}(\mathbf{Y})] - H[q_{\phi}(\mathbf{Y} | \mathbf{P})]. \quad (61)$$

Each term of the loss function can be calculated as follows.

**Expectation of log-likelihood**  $\mathbb{E}_{q_{\phi}(\mathbf{Y} | \mathbf{P})} [\log p_{\psi}(\mathbf{P} | \mathbf{Y})]$  is the reconstruction log-likelihood of  $\mathbf{P}$ . The expectation is estimated by Monte Carlo sampling.

**Differential entropy** In the decomposition of a point cloud  $\mathbf{P}$ , there exists arbitrariness in the choice of  $\mathbf{Y}$  and  $\mathbf{\Lambda}$ , as in  $\mathbf{P}\mathbf{P}^{\top} \simeq \mathbf{Y}\mathbf{\Lambda}\mathbf{Y}^{\top}$ . In this study, we assumed that the diagonal components of  $\mathbf{\Lambda}$  are in descending order, and we restricted the decomposition arbitrariness to be an action  $\mathbf{Q} \in \mathcal{O}(k)$ . If we suppose that the action  $\mathbf{Q}$  follows a uniform distribution when  $\mathbf{Y} = \mathbf{X}\mathbf{Q}$  holds, then  $\mathbf{Y}$  also follows a uniform distribution in the  $k$ -dimensional subspace  $\text{span}(\mathbf{Y})$ . Although this is a uniform distribution on  $\text{St}(k, k)$ , we can consider a uniform distribution on  $\mathcal{O}(k)$  because  $\text{St}(k, k) = \mathcal{O}(k)$ . The probability density function of  $\mathbf{Q}$  is represented by  $U_{\mathcal{O}(k)}(\mathbf{Q})$  in (111).

Therefore, the differential entropy of the decoder model can be calculated as follows.

$$H [q_\phi(\mathbf{Y} | \mathbf{P})] = \mathbb{E} [-\log q_\phi(\mathbf{Y} | \mathbf{P})] \quad (62)$$

$$= - \int_{\mathcal{O}(k)} U_{\mathcal{O}(k)}(\mathbf{Q}) \log U_{\mathcal{O}(k)}(\mathbf{Q}) [d\mathbf{Q}] \quad (63)$$

$$= - \int_{\mathcal{O}(k)} \frac{1}{V_{\mathcal{O}(k)}} \log \frac{1}{V_{\mathcal{O}(k)}} [d\mathbf{Q}] \quad (64)$$

$$= \frac{\log V_{\mathcal{O}(k)}}{V_{\mathcal{O}(k)}} \int_{\mathcal{O}(k)} [d\mathbf{Q}] \quad (65)$$

$$= \frac{\log V_{\mathcal{O}(k)}}{V_{\mathcal{O}(k)}}. \quad (66)$$

**Expectation of prior distribution** We used (9) for the prior distribution  $p_\theta(\mathbf{Y})$ . Further, re-parameterization was used to enable differentiable Monte Carlo estimation of expectations.

$$\mathbb{E}_{q_\phi(\mathbf{Y} | \mathbf{P})} [p_\theta(\mathbf{Y})] = \frac{1}{L} \sum_{l=1}^L p_\theta(\mathbf{X}\mathbf{Q}_l) \quad \text{s.t.} \quad \mathbf{X} \sim \mathbf{Y}, \mathbf{Q}_l \in \mathcal{O}(k), \quad (67)$$

where it was assumed that  $\mathbf{Q}$  is sampled from a uniform distribution, which follows Haar measure on  $\mathcal{O}(k)$ .  $L$  is set  $L = 1$ .

### C.3 Implementation Details and Experimental setting

The following sections present more details about the network architectures, training hyperparameters, and experimental conditions for each of the experiments in Section 6.

#### C.3.1 Artificial Textures

**Network Architecture** The vector field was constructed with the specific input, intermediate, and output layers described in Section 5.3. The GrCNF architecture is shown on top in Table 3. Layers are denoted as Layer in the table, and were processed from top to bottom. Norm. and Act. denote the normalization and activation functions to be applied immediately after the Layer, and the Norm. and Act. were applied in that order. Further, Out Size denotes the output size after Act. Vec denotes the map of vertically concatenating matrices and converting them into a single vector. Moreover, only row Input denotes the size of the input data, not the input layer (HorP). (9) was used for the loss function.

**Hyper-parameters** The mean  $\mathbf{M}$  and covariances  $\mathbf{U}$  and  $\mathbf{V}$  in the prior distribution on the Grassmann manifold were set to  $\mathbf{M} = (1.0, 0.0, 0.0)^\top$ ,  $\mathbf{U} = \sigma^2 \mathbf{I}_3$ , and  $\mathbf{V} = \mathbf{I}_1$ ,  $\sigma = 0.3$ , respectively. Other hyperparameters used during the training of GrCNF are shown in Table 4.

**Implementation** We used PyTorch (Paszke et al. (2019)) to implement the model and run the experiments. The CNF is based on the implementation<sup>5</sup> in Chen et al. (2018) and the framework of the RCNF (Mathieu & Nickel (2020)). Thus, the ODE was solved using the explicit and adaptive Runge–Kutta method (Dormand & Prince (1980)) of order 5, and worked by projecting each step onto a manifold (Hairer (2006)). The autograd in (7) was calculated with torch.autograd.grad (Paszke et al. (2017)) in PyTorch. The experimental hardware was built with an Intel Core i7-9700 CPU and a single NVIDIA GTX 1060 GPU with 6 GB of RAM.

The code used in the experiment to generate the data distributions on  $\text{Gr}(1, 3)$  is shown in Listing 1. This implementation of the data distributions is based on the codes in Kim et al. (2020) and Grathwohl et al. (2019)<sup>6</sup>.

<sup>5</sup>We used the authors' implementation: <https://github.com/rtqichen/torchdiffeq.git>.

<sup>6</sup>We used the authors' implementations: <https://github.com/ANLGB0Y/SoftFlow.git> and <https://github.com/rtqichen/ffjord.git>.

Listing 1: Code for the data distributions.

```

import numpy as np

def get_data_batch(batch_size, dist):
    rng = np.random.RandomState()

    if dist == "2spirals":
        n = np.sqrt(np.random.rand(batch_size // 2, 1)) * 540 * (2 * np.pi) / 360
        dix = -np.cos(n) * n + np.random.rand(batch_size // 2, 1) * 0.1
        diy = np.sin(n) * n + np.random.rand(batch_size // 2, 1) * 0.1
        x = np.vstack((np.hstack((dix, diy)), np.hstack((-dix, -diy)))) / 3
        sample_2d = x + np.random.randn(*x.shape) * 0.1

    elif dist == "swissroll":
        data = sklearn.datasets.make_swiss_roll(n_samples=batch_size, noise=.3)[0]
        data = data.astype("float32")[:, [0, 2]]
        sample_2d = data / 5

    elif dist == "2circles":
        data = sklearn.datasets.make_circles(n_samples=batch_size, \
            factor=.5, noise=0.05)[0]
        data = data.astype("float32")
        sample_2d = data * 3

    elif dist == "2sines":
        x = (rng.rand(batch_size) - 0.5) * 2 * np.pi
        u = (rng.binomial(1, 0.5, batch_size) - 0.5) * 2
        y = u * np.sin(x) * 2.5
        x += np.random.randn(*x.shape) * 0.1
        y += np.random.randn(*y.shape) * 0.1
        sample_2d = np.stack((x, y), 1)

    elif dist == "target":
        shapes = np.random.randint(7, size=batch_size)
        mask = []
        for i in range(7):
            mask.append((shapes == i) * 1.)

        theta = np.linspace(0, 2 * np.pi, batch_size, endpoint=False)
        x = (mask[0] + mask[1] + mask[2]) * (rng.rand(batch_size) - 0.5) * 4 + \
            (-mask[3] + mask[4] * 0.0 + mask[5]) * 2 * np.ones(batch_size) + \
            mask[6] * np.cos(theta)
        y = (mask[3] + mask[4] + mask[5]) * (rng.rand(batch_size) - 0.5) * 4 + \
            (-mask[0] + mask[1] * 0.0 + mask[2]) * 2 * np.ones(batch_size) + \
            mask[6] * np.sin(theta)
        x += np.random.randn(*x.shape) * 0.1
        y += np.random.randn(*y.shape) * 0.1
        sample_2d = np.stack((x, y), 1)

    norm = sample_2d / np.max(np.linalg.norm(sample_2d, axis=1))
    sample_3d = np.concatenate((np.ones((batch_size, 1)), norm), axis=1)
    return sample_3d / np.linalg.norm(sample_3d, axis=1)[:, np.newaxis]

```

### C.3.2 DW4 and LJ13

**Network Architecture** As in Appendix C.3.1, the vector field was constructed with the specific input, intermediate, and output layers described in Section 5.3. The GrCNF architecture is shown on the bottom left and right in Table 3. The bottom left and right were used for experiments on the DW4 and LJ13 datasets, respectively. The views presented in the table is the same as in Appendix C.3.1. (9) was used for the loss function. In addition, for architectures in methods other than GrCNF, please refer to Garcia Satorras et al. (2021).

**Hyperparameters** The mean  $M$  and covariances  $U$  and  $V$  in the prior distribution on the Grassmann manifold were set to  $M = I_{4 \times 2}$ ,  $U = \sigma^2 I_4$ , and  $V = \sigma^2 I_2$ ,  $\sigma = 0.3$  for DW4 and  $M = I_{13 \times 3}$ ,  $U = \sigma^2 I_{13}$ , and  $V = \sigma^2 I_3$ ,  $\sigma = 0.3$  for LJ13, respectively. Other hyperparameters used during the training of GrCNF are shown in Table 4. In addition, for the hyperparameters in methods other than GrCNF, please refer to Garcia Satorras et al. (2021).

**Implementation** The experimental hardware was built using a single NVIDIA Quadro RTX 8000 GPU with 48 GB of GDDR6 RAM. The other environments were the same as in Appendix C.3.1.

Table 3: Network architectures for each experiment; (left) for the Simple Texture dataset, (middle) for the DW4 dataset, (right) for the LJ13 dataset.

<b>GrCNF for Textures</b>		
Layer	Out Size	Norm./Act.
Input	3×1	-
HorP	3×1	-/Tanh
Vec	3	-/-
CS	64	-/Tanh
CS	64	-/Tanh
CS	1	-/Tanh
Grad	3×1	-

<b>GrCNF for DW4</b>			<b>GrCNF for LJ13</b>		
Layer	Out Size	Norm./Act.	Layer	Out Size	Norm./Act.
Input	4×2	-	Input	13×3	-
HorP	4×2	-/Tanh	HorP	13×3	-/Tanh
Vec	8	-/-	Vec	39	-/-
CS	64	-/Tanh	CS	32	-/Tanh
CS	64	-/Tanh	CS	32	-/Tanh
CS	64	-/Tanh	CS	32	-/Tanh
CS	1	-/Tanh	CS	1	-/Tanh
Grad	4×2	-	Grad	13×3	-

**Full results** In Tables 5 and 6, the same DW4 and LJ13 averaged results from Section 6.2 were reported; however, they included the standard deviations over the three runs.

### C.3.3 QM9 Positional

**Network Architecture** On the QM9 Positional, we addressed the task of generating the molecular  $P$  by estimating the scale parameter  $\sqrt{\Lambda} = \text{diag}(\{\sqrt{\lambda_i}\}_{i=1}^3)$ , in addition to the generation of the orthonormal basis matrix  $Y$  with GrCNF. Because the molecular generation task requires a specialized loss function based on variational inference, we used (56), as explained in Appendix C.2. We designed two networks to achieve this. The first is the same GrCNF architecture as in previous experiments, and the second is a scale estimator. Table 7 shows the architectures. The left side of the table shows the GrCNF architecture and the right side shows the scale estimator. The scale estimator estimated one scale parameter from each of the three orthonormal basis vectors  $Y = \{\mathbf{y}_i \in \mathbb{R}^{19}\}_{i=1}^3$ , for a total of three parameters  $\{\sqrt{\lambda_i} \in \mathbb{R}\}_{i=1}^3$ . With the orthonormal orthogonal basis matrix  $Y$  and the estimated scale parameter  $\sqrt{\Lambda}$ , we generated a point cloud  $P = Y\sqrt{\Lambda}$ . In this study, the overall architecture that generates  $P$  is also named GrCNF.

**Hyperparameters** The mean  $M$  and covariances  $U$  and  $V$  in the prior distribution on the Grassmann manifold were set to  $M = I_{19 \times 3}$ ,  $U = \sigma^2 I_{19}$ , and  $V = \sigma^2 I_3$ ,  $\sigma = 0.3$ , respectively. In addition, for the hyperparameters in methods other than GrCNF, please refer to Garcia Satorras et al. (2021).

**Implementation** The experimental hardware was built using a single NVIDIA A100 GPU with 80GB PCIe of GDDR6 RAM.

Table 4: List of hyperparameters used in various experiments. A “-” indicates that the hyperparameter is unused.

		Textures	DW4	LJ13	QM9
# of Data	Train	$\infty$	$10^2/10^3/10^4/10^5$	$10/10^2/10^3/10^4$	13, 831
	Validation	500	1000	1000	2,501
	Test	500	1000	1000	1,813
Optimizer	Name	Adam	Adam	Adam	Adam
	beta1	0.9	0.9	0.9	0.9
	beta2	0.999	0.999	0.999	0.999
	Weight Decay	-	1.0e-12	1.0e-12	1.0e-12
	Learning Rate	1.0e-3	1.0e-4	1.0e-4	5.0e-4
Schedule	Epoch	72000	1000/300/50/6	500/1000/300/50	160
	LR Step with 0.1	20000	-	-	-
	Batch Size	500	100	10/100/100/100	128
NeuralODE	Integration Time	Training	Training	Training	0.1
	atol	1.0e-5	1.0e-5	1.0e-5	1.0e-5
	rtol	1.0e-5	1.0e-5	1.0e-5	1.0e-5
	Adjoint	$\times$	$\checkmark$	$\checkmark$	$\checkmark$

Table 5: Negative log-likelihood comparison on the test partition of DW4 dataset for different amounts of training samples; averaged over 3 runs and including standard deviations.

# of Samples	DW4			
	$10^2$	$10^3$	$10^4$	$10^5$
<b>GNF</b>	$-2.30 \pm 1.59$	$-7.04 \pm 0.64$	$-7.19 \pm 0.99$	$-7.93 \pm 1.10$
<b>GNF-att</b>	$-2.02 \pm 1.34$	$-4.13 \pm 1.20$	$-5.25 \pm 0.89$	$-6.74 \pm 0.89$
<b>GNF-att-aug</b>	$-3.11 \pm 2.15$	$-4.04 \pm 3.40$	$-6.51 \pm 0.49$	$-9.42 \pm 1.15$
<b>Simple dynamics</b>	$-1.22 \pm 0.05$	$-1.28 \pm 0.01$	$-1.36 \pm 0.02$	$-1.39 \pm 0.04$
<b>E-NF</b>	$-0.54 \pm 0.45$	$-9.89 \pm 2.30$	$-12.15 \pm 1.16$	$-15.29 \pm 0.53$
<b>GrCNF</b>	<b><math>-12.53 \pm 0.92</math></b>	<b><math>-13.74 \pm 0.30</math></b>	<b><math>-14.09 \pm 0.44</math></b>	<b><math>-16.07 \pm 0.46</math></b>

Table 6: Negative log-likelihood comparison on the test partition of LJ13 dataset for different amounts of training samples; averaged over 3 runs and including standard deviations.

# Samples	LJ13			
	10	$10^2$	$10^3$	$10^4$
<b>GNF</b>	$6.77 \pm 0.39$	$-0.76 \pm 1.12$	$-4.26 \pm 2.76$	$-12.43 \pm 1.21$
<b>GNF-att</b>	$6.91 \pm 0.17$	$1.40 \pm 0.79$	$-6.81 \pm 2.09$	$-12.05 \pm 2.28$
<b>GNF-att-aug</b>	$2.95 \pm 0.55$	$-6.11 \pm 1.12$	$-13.94 \pm 0.95$	$-15.74 \pm 0.58$
<b>Simple dynamics</b>	$-1.10 \pm 2.55$	$-3.87 \pm 0.25$	$-3.72 \pm 0.08$	$-3.59 \pm 0.52$
<b>E-NF</b>	$-12.86 \pm 3.67$	$-15.75 \pm 5.02$	$-31.51 \pm 1.19$	$-32.83 \pm 1.98$
<b>GrCNF</b>	<b><math>-23.64 \pm 2.23</math></b>	<b><math>-44.24 \pm 4.26</math></b>	<b><math>-58.02 \pm 5.43</math></b>	<b><math>-58.71 \pm 4.71</math></b>

Table 7: Network architectures for QM9 Positional. (Left) GrCNF architecture, (Right) scale estimation architecture. The scale estimator estimates one scale parameter from each of the three orthonormal basis vectors  $\mathbf{Y} = \{\mathbf{y}_i \in \mathbb{R}^{19}\}_{i=1}^3$ , for  $\sqrt{\Lambda} = \text{diag}(\{\sqrt{\lambda_i}\}_{i=1}^3)$  with three parameters  $\{\sqrt{\lambda_i} \in \mathbb{R}\}_{i=1}^3$ . Using the orthonormal orthogonal basis matrix  $\mathbf{Y}$  and the estimated scale parameter  $\sqrt{\Lambda}$ , we generate a point cloud  $\mathbf{P} = \mathbf{Y}\sqrt{\Lambda}$ . SiLU is an activation function proposed in (Ramachandran et al. (2017)) and BatchNorm. is a batch normalization layer in (Ioffe & Szegedy (2015)).

<b>GrCNF for QM9 Positional</b>			<b>Scale Estimator for QM9 Positional</b>		
Layer	Out Size	Norm./Act.	Layer	Out Size	Norm./Act.
Input	19×3	-	Input	19×3	-
HorP	19×3	-/Tanh	FC	128	BatchNorm./SiLU
Vec	57	-/-	FC	256	BatchNorm./SiLU
CS	32	-/Tanh	FC	256	BatchNorm./SiLU
CS	32	-/Tanh	FC	128	BatchNorm./SiLU
CS	32	-/Tanh	FC	3	-/ReLU
CS	1	-/Tanh			
Grad	19×3	-			

## D Fundamentals of Concepts Associated with Grassmann Manifold

### D.1 Definition of Stiefel Manifold

**Definition 2.** An (orthogonal or compact) Stiefel manifold  $\text{St}(k, D)$  is defined as the set of orthonormal bases of  $k$ -dimensional subspaces in the Euclidean space  $\mathbb{R}^D$  as in (68).

$$\text{St}(k, D) := \{ \mathbf{Y} \in \mathbb{R}^{D \times k} \mid \mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_k \}. \quad (68)$$

For  $\mathbf{Y} \in \text{St}(k, D)$ , the space  $\text{span}(\mathbf{Y})$  spanned by its column vectors is the element of  $\text{Gr}(k, D)$ .

$$f : \text{St}(k, D) \rightarrow \text{Gr}(k, D) : \mathbf{Y} \mapsto \text{span}(\mathbf{Y}). \quad (69)$$

$\text{St}(k, D)$  is a  $Dk - \frac{k(k+1)}{2}$ -dimensional compact manifold (Absil et al. (2008)).

### D.2 Equivalence Relation

To define the equivalence relation  $\sim^7$  on a Stiefel manifold  $\text{St}(k, D)$ , we introduce the following two lemmas.

**Lemma 1.** The necessary and sufficient conditions for  $\text{span}(\mathbf{Y}_1) = \text{span}(\mathbf{Y}_2)$  to hold for  $\mathbf{Y}_1, \mathbf{Y}_2 \in \text{St}(k, D)$  are as follows.

$$\exists \mathbf{Q} \in \mathcal{O}(k) \quad \text{s.t.} \quad \mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}. \quad (70)$$

*Proof.*  $\text{span}(\mathbf{Y}_1) = \text{span}(\mathbf{Y}_2) \Leftrightarrow \mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}$ . From the definition,  $\mathbf{Y}_1^\top \mathbf{Y}_1 = \mathbf{Y}_2^\top \mathbf{Y}_2 = \mathbf{I}_k$ , the following is obtained:

$$\mathbf{I}_k = \mathbf{Y}_1^\top \mathbf{Y}_1 = \mathbf{Q}^\top \mathbf{Y}_2^\top \mathbf{Y}_2 \mathbf{Q} = \mathbf{Q}^\top \mathbf{Q}. \quad (71)$$

Thus, there exists a  $k$ -dimensional orthogonal matrix  $\mathbf{Q} \in \mathcal{O}(k)$ . As the subspace  $\text{span}(\mathbf{Y})$  is invariant to coordinate transformations by orthogonal matrices,  $\text{span}(\mathbf{Y}_1) = \text{span}(\mathbf{Y}_2)$  is true.

$\text{span}(\mathbf{Y}_1) = \text{span}(\mathbf{Y}_2) \Rightarrow \mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}$ . From the assumption, we immediately concluded that  $\mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}$  for  $\mathbf{Q} \in \mathcal{O}(k)$ .  $\square$

**Lemma 2.** We define the equivalence relation  $\sim$  on  $\text{St}(k, D)$  to be  $\mathbf{Y}_1 \sim \mathbf{Y}_2$  whenever (70) is satisfied with respect to  $\mathbf{Y}_1, \mathbf{Y}_2 \in \text{St}(k, D)$ . The fact that a binary relation is an equivalence relation  $\sim$  implies that the following three statements hold for  $\forall \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3 \in \text{St}(k, D)$ .

**Reflexivity**  $\mathbf{Y}_1 \sim \mathbf{Y}_1$ .

**Symmetry**  $\mathbf{Y}_1 \sim \mathbf{Y}_2 \Rightarrow \mathbf{Y}_2 \sim \mathbf{Y}_1$ .

**Transitivity**  $\mathbf{Y}_1 \sim \mathbf{Y}_2 \wedge \mathbf{Y}_2 \sim \mathbf{Y}_3 \Rightarrow \mathbf{Y}_1 \sim \mathbf{Y}_3$ .

*Proof.* With Lemma 1, we can confirm that it is valid as follows:

**Reflexivity** From  $\mathbf{Y}_1 = \mathbf{Y}_1 \mathbf{I}$ ,  $\mathbf{I} \in \mathcal{O}$ , we obtain  $\mathbf{Y}_1 \sim \mathbf{Y}_1$ .

**Symmetry** As  $\mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}$  is obtained from  $\mathbf{Y}_1 \sim \mathbf{Y}_2$ , and  $\mathbf{Y}_2 \mathbf{Q}^\top = \mathbf{Y}_1$ ,  $\mathbf{Q}^\top \in \mathcal{O}$  is true, then  $\mathbf{Y}_2 \sim \mathbf{Y}_1$  is obtained.

**Transitivity**  $\mathbf{Y}_3 = \mathbf{Y}_1 \mathbf{Q}_1 \mathbf{Q}_2$  with  $\mathbf{Y}_2 = \mathbf{Y}_1 \mathbf{Q}_1$  and  $\mathbf{Y}_3 = \mathbf{Y}_2 \mathbf{Q}_2$ .  $\mathbf{Q}_1 \mathbf{Q}_2$  is  $(\mathbf{Q}_1 \mathbf{Q}_2)^\top (\mathbf{Q}_1 \mathbf{Q}_2) = \mathbf{Q}_2^\top \mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{Q}_2 = \mathbf{Q}_2^\top \mathbf{Q}_2 = \mathbf{I}$ . Moreover, as  $(\mathbf{Q}_1 \mathbf{Q}_2) (\mathbf{Q}_1 \mathbf{Q}_2)^\top = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_2^\top \mathbf{Q}_1^\top = \mathbf{I}$  holds,  $\mathbf{Q}_1 \mathbf{Q}_2 \in \mathcal{O}$  is true. Therefore, we concluded  $\mathbf{Y}_1 \sim \mathbf{Y}_3$ .  $\square$

The equivalence class of  $\mathbf{Y} \in \text{St}(k, D)$  is denoted by  $[\mathbf{Y}]$ . In other words,  $[\mathbf{Y}]$  is the set of all elements of  $\text{St}(k, D)$  that are equivalent to  $\mathbf{Y}$ , and the equivalence relation on  $\text{St}(k, D)$  divides  $\text{St}(k, D)$  into equivalence classes with no intersection. Thus,  $\mathbf{Y}$  is then referred to as the representative of the equivalence class  $[\mathbf{Y}]$ . The set of equivalence classes is denoted  $\text{St}(k, D)/\sim$  and is referred to as the

<sup>7</sup>Reflexive, symmetric and transitive binary relations. As a consequence of these properties, in a given set, one equivalence relation divides (classifies) the set into equivalence classes. Note that  $R$  is a binary relation in the set  $X$  if for any  $x, y \in X$ , only either  $x$  is related to  $y$  by the relation  $R$ , or  $x$  is not related to  $y$  based on the relation that  $R$  occurs. We write  $x$  is related to  $y$  by relation  $R$  as  $xRy$ .

quotient of  $\text{St}(k, D)$  by the equivalence relation  $\sim$ . In addition,  $\pi : \text{St}(k, D) \rightarrow \text{St}(k, D)/\sim$  is the natural projection that maps  $\mathbf{Y} \in \text{St}(k, D)$  onto its equivalence class  $[\mathbf{Y}]$ . This projection  $\pi$  is surjection.

### D.3 Quotient Manifold

**Definition 3.** Let  $\overline{\mathcal{M}}$  be a manifold with equivalence relation  $\sim$ . The quotient space  $\overline{\mathcal{M}}/\sim$  with  $\sim$  of  $\overline{\mathcal{M}}$  is the set of all equivalence classes. Thus,  $\overline{\mathcal{M}}/\sim := \{\pi(\overline{\mathbf{x}}) \mid \overline{\mathbf{x}} \in \overline{\mathcal{M}}\}$ , where  $\pi : \overline{\mathcal{M}} \rightarrow \overline{\mathcal{M}}/\sim$  is the natural projection and  $\pi(\overline{\mathbf{x}}) := \{\overline{\mathbf{y}} \in \overline{\mathcal{M}} \mid \overline{\mathbf{y}} \sim \overline{\mathbf{x}}\}$ . Then,  $\overline{\mathcal{M}}$  is referred to as the total space or the total manifold. Moreover,  $\overline{\mathcal{M}}/\sim$  is referred to as a quotient manifold of  $\overline{\mathcal{M}}$  if  $\overline{\mathcal{M}}/\sim$  admits a differentiable structure.

Let  $\mathcal{M} = \overline{\mathcal{M}}/\sim$  be a quotient manifold. Further, suppose that  $\overline{\mathcal{M}}$  is endowed with a Riemann metric  $\overline{g}$ , and let  $\mathbf{x} = \pi(\overline{\mathbf{x}})$ . The horizontal space  $T_{\overline{\mathbf{x}}}^h \overline{\mathcal{M}}$  is the orthogonal complement of the vertical space  $T_{\overline{\mathbf{x}}}^v \overline{\mathcal{M}} := T_{\overline{\mathbf{x}}} \pi^{-1}(\mathbf{x})$  in the tangent space  $T_{\overline{\mathbf{x}}} \overline{\mathcal{M}}$  and is defined as the follows:

$$T_{\overline{\mathbf{x}}}^h \overline{\mathcal{M}} := (T_{\overline{\mathbf{x}}}^v \overline{\mathcal{M}})^\perp = \{\overline{\boldsymbol{\eta}}_{\overline{\mathbf{x}}} \in T_{\overline{\mathbf{x}}} \overline{\mathcal{M}} \mid \overline{g}_{\overline{\mathbf{x}}}(\overline{\boldsymbol{\xi}}_{\overline{\mathbf{x}}}, \overline{\boldsymbol{\eta}}_{\overline{\mathbf{x}}}) = 0, \forall \overline{\boldsymbol{\xi}}_{\overline{\mathbf{x}}} \in T_{\overline{\mathbf{x}}}^v \overline{\mathcal{M}}\}. \quad (72)$$

The horizontal lift  $\overline{\boldsymbol{\xi}}_{\overline{\mathbf{x}}}^h \in T_{\overline{\mathbf{x}}}^h \overline{\mathcal{M}}$  of the tangent vector  $\boldsymbol{\xi}_{\mathbf{x}} \in T_{\mathbf{x}} \mathcal{M}$  at point  $\overline{\mathbf{x}} \in \pi^{-1}(\mathbf{x})$  is a tangent vector that is uniquely determined as  $d\pi_{\overline{\mathbf{x}}}(\overline{\boldsymbol{\xi}}_{\overline{\mathbf{x}}}^h) = \boldsymbol{\xi}_{\mathbf{x}}$  (Absil et al. (2008)).

### D.4 Grassmann Manifold Exploiting the Quotient Structure

#### D.4.1 Tangent Space on a Stiefel Manifold

We describe the relationship between tangent space  $T_{[\mathbf{Y}]} \text{Gr}(k, D)$  on  $\text{Gr}(k, D)$  and tangent space  $T_{\mathbf{Y}} \text{St}(k, D)$  on  $\text{St}(k, D)$  to relate the tangent vectors of a Grassmann manifold  $\text{Gr}(k, D)$  to the tangent vectors of a Stiefel manifold  $\text{St}(k, D)$  in a matrix representation. We take the derivative on both sides of  $\mathbf{Y}(t)^\top \mathbf{Y}(t) = \mathbf{I}_p$  in (68) by  $t$  and solve for  $t = 0$ .

$$\frac{d}{dt} \left\{ \mathbf{Y}(t)^\top \mathbf{Y}(t) \right\} = \frac{d}{dt} \mathbf{I}_p \quad (73)$$

$$\frac{d}{dt} \mathbf{Y}(t)^\top \mathbf{Y}(t) + \mathbf{Y}(t)^\top \frac{d}{dt} \mathbf{Y}(t) = 0 \quad (74)$$

$$\frac{d}{dt} \mathbf{Y}(0)^\top \mathbf{Y}(0) + \mathbf{Y}(0)^\top \frac{d}{dt} \mathbf{Y}(0) = 0 \quad (75)$$

$$\overline{\boldsymbol{\xi}}_{\mathbf{Y}}^\top \mathbf{Y} + \mathbf{Y}^\top \overline{\boldsymbol{\xi}}_{\mathbf{Y}} = 0, \quad (76)$$

where  $\overline{\boldsymbol{\xi}}_{\mathbf{Y}} = \frac{d}{dt} \mathbf{Y}(0)$  is the tangent vector at  $\mathbf{Y}$ <sup>8</sup>.

**Definition 4.** Define the tangent space  $T_{\mathbf{Y}} \text{St}(k, D)$  at  $\mathbf{Y}$  on the Stiefel manifold as follows:

$$T_{\mathbf{Y}} \text{St}(k, D) = \left\{ \overline{\boldsymbol{\xi}}_{\mathbf{Y}} \in \mathbb{R}^{D \times k} \mid \overline{\boldsymbol{\xi}}_{\mathbf{Y}}^\top \mathbf{Y} + \mathbf{Y}^\top \overline{\boldsymbol{\xi}}_{\mathbf{Y}} = \mathbf{0}_k \right\}. \quad (77)$$

Let matrix  $\mathbf{Y}_\perp \in \mathbb{R}^{D \times (D-k)}$  be a matrix satisfying the following:

$$\mathbf{Y}_\perp^\top \mathbf{Y}_\perp = \mathbf{I}_{D-k}, \quad \mathbf{Y}^\top \mathbf{Y}_\perp = 0, \quad \mathbf{Y} \mathbf{Y}^\top + \mathbf{Y}_\perp \mathbf{Y}_\perp^\top = \mathbf{I}_D. \quad (78)$$

As  $[\mathbf{Y} \quad \mathbf{Y}_\perp]$  is an orthogonal matrix<sup>9</sup>, the column vectors of  $\mathbf{Y}$  and  $\mathbf{Y}_\perp$  form an orthonormal basis in  $\mathbb{R}^D$ . Thus, any  $D \times k$  matrix can be written in terms of the  $\mathbf{C} \in \mathbb{R}^{k \times k}$  and  $\mathbf{B} \in \mathbb{R}^{(D-k) \times k}$  coefficient matrices as follows:

$$\mathbf{Y} \mathbf{C} + \mathbf{Y}_\perp \mathbf{B}, \quad (79)$$

<sup>8</sup>The tangent space is defined independently for each point of the manifold; hence, the subscript  $\mathbf{Y}$ , as in  $\overline{\boldsymbol{\xi}}_{\mathbf{Y}}$ , is clearly stated to emphasize that it is a tangent vector at  $\mathbf{Y}$ .

<sup>9</sup> $[\mathbf{Y} \quad \mathbf{Y}_\perp]^{-1} [\mathbf{Y} \quad \mathbf{Y}_\perp] = [\mathbf{Y} \quad \mathbf{Y}_\perp]^\top [\mathbf{Y} \quad \mathbf{Y}_\perp] = [\mathbf{Y} \quad \mathbf{Y}_\perp] [\mathbf{Y} \quad \mathbf{Y}_\perp]^\top = \mathbf{I}_D$ .

where  $\bar{\xi}_Y = YC + Y_\perp B$  is inserted. The following equation is obtained.

$$\bar{\xi}_Y^\top Y + Y^\top \bar{\xi}_Y = (YC + Y_\perp B)^\top Y + Y^\top (YC + Y_\perp B) \quad (80)$$

$$= B^\top Y_\perp^\top Y + C^\top Y^\top Y + Y^\top YC + Y^\top Y_\perp B \quad (81)$$

$$= B^\top Y^\top Y_\perp + C^\top + C \quad (82)$$

$$= C^\top + C \quad (83)$$

$$= \mathbf{0}_k. \quad (84)$$

Thus, the following equation is derived.

$$C^\top + C = \mathbf{0}_k. \quad (85)$$

Thus,  $C$  is a  $k \times k$  skew-symmetric matrix  $\text{Skew}(k)$ . Therefore, we obtain the following as another representation of the tangent space on  $\text{St}(k, D)$ .

$$T_Y \text{St}(k, D) = \left\{ YC + Y_\perp B \mid C \in \text{Skew}(k), B \in \mathbb{R}^{(D-k) \times k} \right\}. \quad (86)$$

#### D.4.2 Riemannian Metric on a Stiefel Manifold

In the tangent space  $T_x \mathcal{M}$  defined at each point  $x \in \mathcal{M}$  on the manifold  $\mathcal{M}$ , the inner product  $h$  is endowed as a bilinear map. This  $h$  is referred to as a Riemannian metric on a manifold, and the manifold  $\mathcal{M}$  on which the Riemannian metric  $h$  is endowed is referred to as a Riemannian manifold  $(\mathcal{M}, h)$ . We define the Riemannian metric  $\bar{g}$  on  $\text{St}(k, D)$  as follows.

$$\bar{g}_Y(\bar{\xi}_Y, \bar{\eta}_Y) := \text{tr}(\bar{\xi}_Y^\top \bar{\eta}_Y) \quad \text{s.t.} \quad \bar{\xi}_Y, \bar{\eta}_Y \in T_Y \text{St}(k, D), Y \in \text{St}(k, D). \quad (87)$$

This is the standard inner product of  $\mathbb{R}^{D \times k}$  induced by  $T_Y \text{St}(k, D)$ , with  $T_Y \text{St}(k, D) \subset \mathbb{R}^{D \times k}$ <sup>1011</sup>.

#### D.4.3 Tangent Space on a Grassmann Manifold

We describe the relation between tangent spaces  $T_{[Y]} \text{Gr}(k, D)$  and  $T_Y \text{St}(k, D)$  to relate tangent vectors in tangent spaces  $T_{[Y]} \text{Gr}(k, D)$  on  $\text{Gr}(k, D)$  to tangent vectors  $\bar{\xi}_Y \in T_Y \text{St}(k, D)$ .

First, we define the vertical space  $T_Y^v \text{St}(k, D)$  as a subspace of  $T_Y \text{St}(k, D)$  as follows.

$$T_Y^v \text{St}(k, D) := T_Y \pi^{-1}([Y]), \quad (88)$$

where  $\pi : \text{St}(k, D) \rightarrow \text{Gr}(k, D)$  is the natural projection defined by  $\pi(Y) = [Y]$ <sup>12</sup>. Thus,  $\pi$  converges all  $Y' \in \text{St}(k, D)$  such that  $Y \sim Y'$  to a point  $[Y]$  on  $\text{Gr}(k, D)$ . Therefore, using (1), (88) can be transformed as follows.

$$T_Y^v \text{St}(k, D) = T_Y \{YQ \mid Q \in \mathcal{O}(k)\}. \quad (89)$$

However,  $\bar{\xi}_Y^v \in T_Y^v \text{St}(k, D)$  can be written as  $\bar{\xi}_Y^v = YS$  with  $S \in T_{I_k} \mathcal{O}(k)$ .

$$T_{I_k} \mathcal{O}(k) = T_{I_k} \text{St}(k, k) \quad (90)$$

$$= \left\{ I_k C + (I_{k^\perp} B = \mathbf{0}_k) \mid C \in \text{Skew}(k), B \in \mathbb{R}^{(k-k=0) \times k} \right\} \quad (91)$$

$$= \text{Skew}(k). \quad (92)$$

<sup>10</sup>The inner product  $A \cdot C = A^\top C$  of a vector is typically referred to as the standard inner product. Further, matrices are similarly defined with a standard inner product, defined as  $A \cdot C = \text{tr}(A^\top C)$ . A space  $\mathbb{R}^{D \times k}$  such that the  $D \times k$  matrix  $A$  is an element is referred to as a matrix space. The standard basis of the matrix space can be constructed by a matrix wherein only one element in the matrix is 1 and the remaining are 0. The matrix space is a linear space because it satisfies the linearity that is similar to that in case of a linear vector space.

<sup>11</sup>When  $\mathcal{N}$  is a submanifold of a Riemannian manifold  $(\mathcal{M}, g)$ , we define the Riemannian metric  $\bar{g}$  of  $\mathcal{N}$  to be:

$$\bar{g}_x(\xi, \eta) := g_x(\xi, \eta), \quad x \in \mathcal{N} \subset \mathcal{M}, \xi, \eta \in T_x \mathcal{N} \subset T_x \mathcal{M}.$$

$\bar{g}$  is an induced metric and  $(\mathcal{N}, \bar{g})$  is a Riemannian submanifold of  $(\mathcal{M}, g)$ . As  $\text{St}(k, D)$  is a submanifold of  $\mathbb{R}^{D \times k}$ , we can define the standard inner product  $A \cdot C = \text{tr}(A^\top C)$  of  $\mathbb{R}^{D \times k}$  as the induced metric  $\bar{g}$ . Thus,  $\text{St}(k, D)$  is a Riemannian submanifold of  $\mathbb{R}^{D \times k}$ .

<sup>12</sup>Suppose a set is given a suitable equivalence relation. A natural projection is a map that sends each element of a set to the equivalence class to which it belongs.

Thus, we obtain the following formula.

$$T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D) = \{\mathbf{Y}\mathbf{C} \mid \mathbf{C} \in \text{Skew}(k)\}. \quad (93)$$

Next, we define the horizontal space  $T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  as the orthogonal complement of  $T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D)$  in  $T_{\mathbf{Y}} \text{St}(k, D)$  endowed with the inner product (87).

$$\begin{aligned} T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D) &:= (T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D))^{\perp} \\ &= \left\{ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \in T_{\mathbf{Y}} \text{St}(k, D) \mid \text{tr} \left( \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}\top} \bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{v}} \right) = 0, \bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{v}} \in T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D) \right\}. \end{aligned} \quad (94)$$

Based on the fact that  $T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D)$  is a subspace of  $T_{\mathbf{Y}} \text{St}(k, D)$  and  $T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  is defined as its orthogonal complement, the direct sum decomposition is as follows.

$$T_{\mathbf{Y}} \text{St}(k, D) = T_{\mathbf{Y}}^{\mathbf{v}} \text{St}(k, D) \oplus T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D), \quad (96)$$

where  $\oplus$  denotes direct sum. Moreover, the tangent space is a linear space (Absil et al. (2008)). From (86), element  $\mathbf{Y}\mathbf{C}$  of  $T_{\mathbf{Y}} \text{St}(k, D)$  corresponds to the first term of (93); thus, (96) is formulated as follows:

$$T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D) = \left\{ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} = \mathbf{Y}_{\perp} \mathbf{B} \mid \mathbf{B} \in \mathbb{R}^{(D-k) \times k} \right\}. \quad (97)$$

Note that the horizontal vector  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}}$  is not necessarily an orthogonal matrix.

Finally, define the element  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \in T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  of the horizontal space at  $\mathbf{Y} \in \text{St}(k, D)$  for the tangent vector  $\boldsymbol{\xi}_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$  at  $[\mathbf{Y}] \in \text{Gr}(k, D)$  as satisfying the following formula.

$$d\pi(\mathbf{Y}) \left[ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \right] = \boldsymbol{\xi}_{[\mathbf{Y}]}, \quad (98)$$

where  $d\pi(\mathbf{Y}) : T_{\mathbf{Y}} \text{St}(k, D) \rightarrow T_{[\mathbf{Y}]} \text{Gr}(k, D)$  is the derivative  $\frac{d\pi(\mathbf{Y})}{d\mathbf{Y}}$  of  $\pi : \text{St}(k, D) \rightarrow \text{Gr}(k, D)$  at  $\mathbf{Y} \in \text{St}(k, D)$ . The  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \in T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  is referred to as the horizontal lift at  $\mathbf{Y} \in \text{St}(k, D)$  of  $[\mathbf{Y}] \in \text{Gr}(k, D)$ .

We describe the tangent space of  $\text{Gr}(k, D)$  with the concept of horizontal lift.

**Definition 5.** Let  $T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  be a horizontal space on  $\text{St}(k, D)$ . Then, we define the tangent space  $T_{[\mathbf{Y}]} \text{Gr}(k, D)$  of the  $\text{Gr}(k, D)$  as follows.

$$T_{[\mathbf{Y}]} \text{Gr}(k, D) = \left\{ \boldsymbol{\xi}_{[\mathbf{Y}]} \mid d\pi(\mathbf{Y}) \left[ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \right] = \boldsymbol{\xi}_{[\mathbf{Y}]}, \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \in T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D) \right\}. \quad (99)$$

From the above,  $\boldsymbol{\xi}_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$  is obtained from the map  $d\pi(\mathbf{Y}) \left[ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \right]$  when  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}}$  is obtained. The  $\boldsymbol{\xi}_{[\mathbf{Y}]}$  is defined by an equivalence class and cannot be treated numerically in matrix form; however, it is sufficient to obtain the  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}}$  for actual numerical calculations. For  $\boldsymbol{\xi}_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$ , there exists a  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \in T_{\mathbf{Y}}^{\mathbf{h}} \text{St}(k, D)$  that uniquely satisfies (98). In other words, we can handle it in matrix form by using elements of the horizontal space of Stiefel manifolds through the concept of horizontal lifting. Figure 1 is a conceptual diagram of the tangent space representation of a Grassmann manifold by horizontal lift.

#### D.4.4 Riemannian Metric on a Grassmann Manifold

We define the Riemannian metric  $g$  of  $\text{Gr}(k, D)$  through the concept of horizontal lift.

**Definition 6.** Let  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}}$  and  $\bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{h}}$  be the horizontal lifts that become  $d\pi(\mathbf{Y}) \left[ \bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} \right] = \boldsymbol{\xi}_{[\mathbf{Y}]}$  and  $d\pi(\mathbf{Y}) \left[ \bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{h}} \right] = \boldsymbol{\eta}_{[\mathbf{Y}]}$ , respectively. Then, we define the Riemannian metric on  $\text{Gr}(k, D)$  as follows:

$$g_{[\mathbf{Y}]}(\boldsymbol{\xi}_{[\mathbf{Y}]}, \boldsymbol{\eta}_{[\mathbf{Y}]}) := \bar{g}_{\mathbf{Y}}(\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}}, \bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{h}}) = \text{tr}(\mathbf{B}^{\top} \mathbf{D}), \quad (100)$$

where  $\mathbf{B}$  and  $\mathbf{D}$  are matrices that are  $\bar{\boldsymbol{\xi}}_{\mathbf{Y}}^{\mathbf{h}} = \mathbf{Y}_{\perp} \mathbf{B}$  and  $\bar{\boldsymbol{\eta}}_{\mathbf{Y}}^{\mathbf{h}} = \mathbf{Y}_{\perp} \mathbf{D}$ , respectively.

## D.5 Invariant Measures

Let the column vectors of matrix  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_k\} \in \mathbb{R}^{D \times k}$  be the orthonormal basis that span the subspace  $\text{span}(\mathbf{Y}) \in \text{Gr}(k, D)$  in  $\mathbb{R}^D$ , and the column vectors of  $\mathbf{Y}_\perp = \{\mathbf{y}_{k+1}, \dots, \mathbf{y}_D\} \in \mathbb{R}^{D \times D-k}$  be the orthogonal complementary space  $\text{span}(\mathbf{Y}_\perp)$  of  $\text{span}(\mathbf{Y})$ , respectively. Then, the following differential form can be defined.

$$(d\mathbf{Y}) = \bigwedge_{j=1}^{D-k} \bigwedge_{i=1}^k \mathbf{y}_{k+j}^\top d\mathbf{y}_i \quad (101)$$

$$= (\mathbf{y}_{k+1}^\top d\mathbf{y}_1 \wedge \dots \wedge \mathbf{y}_{k+1}^\top d\mathbf{y}_k) \wedge \dots \wedge (\mathbf{y}_D^\top d\mathbf{y}_1 \wedge \dots \wedge \mathbf{y}_D^\top d\mathbf{y}_k), \quad (102)$$

where  $\wedge$  is the wedge product and the relation satisfies  $\omega_i \wedge \omega_i = \omega_j \wedge \omega_j = 0$  and  $\omega_i \wedge \omega_j = -\omega_j \wedge \omega_i$ . The above equation is in  $k(D-k)$ -order differential form, which is an invariant measure on  $\text{Gr}(k, D)$  (Chikuse (2003)).

If we define the matrix  $\mathbf{X}_\perp$  to be  $[\mathbf{X} \quad \mathbf{X}_\perp]$  for any point  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\} \in \text{St}(k, D)$ , the differential form for an invariant measure on  $\text{St}(k, D)$  is defined as follows.

$$(d\mathbf{X}) = \bigwedge_{j=1}^{D-k} \bigwedge_{i=1}^k \mathbf{x}_{k+j}^\top d\mathbf{x}_i \bigwedge_{i < j}^k \mathbf{x}_j^\top d\mathbf{x}_i = (d\mathbf{Y})(d\mathbf{Q}), \quad (103)$$

where  $(d\mathbf{Q})$  is the invariant measure of  $\mathcal{O}(k)$ . The integral of (103), that is, the volume of  $\text{St}(k, D)$ , can be evaluated as follows:

$$V_{\text{St}(k, D)} = \int_{\text{St}(k, D)} (d\mathbf{X}). \quad (104)$$

(104) can be computed as follows. First, the surface  $S_D$  of the  $D$ -dimensional unit sphere can be defined as follows:

$$S_D = \left. \frac{d}{dr} \right|_{r=1} V_D = DV_D = \frac{D\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2} + 1)} = \frac{2\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2})}, \quad (105)$$

where  $V_D$  is the volume of a  $D$ -dimensional sphere  $\frac{\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2} + 1)} r^D$  and  $\Gamma(\frac{D}{2})$  is the gamma function. Then, the following equation is obtained.

$$\int_{\text{St}(k, D)} (d\mathbf{X}) = S_D \int_{\text{St}(k-1, D-1)} (d\mathbf{X}_1), \quad (106)$$

where  $(d\mathbf{X}_1)$  is the differential form of  $\text{St}(k-1, D-1)$ . Thus, (104) can be transformed as follows.

$$V_{\text{St}(k, D)} = \int_{\text{St}(k, D)} (d\mathbf{X}) = \prod_{i=1}^k S_D = \frac{2^k \pi^{\frac{Dk}{2}}}{\Gamma_k(\frac{D}{2})}, \quad (107)$$

where  $\Gamma_k(\frac{D}{2})$  is the multidimensional gamma function. The invariant measure  $(d\mathbf{X})$  is an unnormalized measure. A measure normalized to be a probability measure can be formulated as follows:

$$[d\mathbf{X}] = \frac{1}{V_{\text{St}(k, D)}} (d\mathbf{X}). \quad (108)$$

This is a uniform distribution on  $\text{St}(k, D)$ . As  $\text{St}(k, k) = \mathcal{O}(k)$ , the volume  $V_{\mathcal{O}(k)}$  of  $\mathcal{O}(k)$  can be represented using  $(d\mathbf{Q})$  as follows.

$$V_{\mathcal{O}(k)} = V_{\text{St}(k, k)} = \frac{2^k \pi^{\frac{k^2}{2}}}{\Gamma_k(\frac{k}{2})}. \quad (109)$$

Furthermore, a measure normalized to be a probability measure can be represented by the following:

$$[d\mathbf{Q}] = \frac{1}{V_{\mathcal{O}(k)}} (d\mathbf{Q}). \quad (110)$$

From the above, the probability density function  $U_{\mathcal{O}(k)}(\mathbf{Q})$  of the uniform distribution on  $\mathcal{O}(k)$  is as follows:

$$U_{\mathcal{O}(k)}(\mathbf{Q}) = \frac{1}{V_{\mathcal{O}(k)}} \quad \text{s.t. } \mathbf{Q} \in \mathcal{O}(k), \quad (111)$$

$$\int_{\mathcal{O}(k)} U_{\mathcal{O}(k)}(\mathbf{Q}) (d\mathbf{Q}) = \int_{\mathcal{O}(k)} \frac{1}{V_{\mathcal{O}(k)}} (d\mathbf{Q}) = \int_{\mathcal{O}(k)} [d\mathbf{Q}] = 1. \quad (112)$$

As  $\text{Gr}(k, D)$  is defined as a quotient manifold  $\text{St}(k, D) / \mathcal{O}(k)$  as in (2), the volume  $V_{\text{Gr}(k, D)}$  of  $\text{Gr}(k, D)$  can be defined as follows.

$$V_{\text{Gr}(k, D)} = \int_{\text{Gr}(k, D)} (d\mathbf{Y}) = \frac{V_{\text{St}(k, D)}}{V_{\mathcal{O}(k)}} = \frac{V_{\text{St}(k, D)}}{V_{\text{St}(k, k)}} = \frac{\pi^{\frac{k(D-k)}{2}} \Gamma_k\left(\frac{k}{2}\right)}{\Gamma_k\left(\frac{D}{2}\right)}. \quad (113)$$

The measure normalized to be a probability measure is expressed as:

$$[d\mathbf{Y}] = \frac{1}{V_{\text{Gr}(k, D)}} (d\mathbf{Y}). \quad (114)$$

## D.6 Retraction

In general, the points except the origin ( $\mathbf{p}(0) = \mathbf{x}$ ) of the tangent space  $T_{\mathbf{x}}\mathcal{M}$  at  $\mathbf{x}$  on the manifold  $\mathcal{M}$  are not elements on  $\mathcal{M}$  ( $\mathbf{p}(t) \in T_{\mathbf{x}}\mathcal{M}, t \neq 0$ ). Therefore, if the result of the operation on the tangent space is to be used at another point on the manifold  $\mathcal{M}$ , it is necessary to map  $\mathbf{p}(t)$  to the manifold  $\mathcal{M}$ . The map from a tangent space to a manifold is referred to as an exponential map. However, because the exponential map is computationally expensive, retraction based on numerical linear algebra is often used as an alternative (Zhu & Sato (2021)). Retraction is a method for approximating an exponential map to first order while maintaining global convergence in optimization algorithms on Riemannian manifolds. The most commonly used retractions on  $\text{Gr}(k, D)$  are methods based on QR decomposition or singular-value decomposition (SVD) (Absil et al. (2008); Zhu & Sato (2021)). In addition, a retraction based on the Cayley transform is introduced in Zhu & Sato (2021). This retraction is closely related to the Cayley transform on  $\text{St}(k, D)$  (Wen & Yin (2013); Xiaojing (2017); Zhu & Duan (2019)) and the Projected polynomial retraction (Gawlik & Leok (2018a)).

### D.6.1 Exponential Map and Retraction

Geodesics on  $\text{Gr}(k, D)$  can be expressed as the equivalence class  $[\exp_{\mathbf{Y}}^{\text{Gr}}(t\bar{\xi}_{\mathbf{Y}})]$ , where

$$\exp_{\mathbf{Y}}^{\text{Gr}}(t\bar{\xi}_{\mathbf{Y}}) = [\mathbf{Y} \quad \mathbf{Y}_{\perp}] \exp(t\mathfrak{B}) \mathbf{I}_{D \times k}. \quad (115)$$

Here,  $\exp$  on the right-hand side is the matrix exponential, and  $\mathfrak{B} = \begin{bmatrix} \mathbf{0}_k & -\mathbf{B}^{\top} \\ \mathbf{B} & \mathbf{0}_{D-k} \end{bmatrix} \in \text{skew}(D)$ ,

where  $\mathbf{B}$  satisfies  $\bar{\xi}_{\mathbf{Y}} = \mathbf{Y}_{\perp} \mathbf{B}$ . We can use a following exponential map that is mathematically equivalent to (115) (Edelman et al. (1998)):

$$\exp_{\mathbf{Y}}^{\text{Gr}}(t\bar{\xi}_{\mathbf{Y}}) := \{\mathbf{Y}\mathbf{V} \cos(\Sigma t) + \mathbf{U} \sin(\Sigma t)\} \mathbf{V}^{\top}, \quad (116)$$

where  $\mathbf{U}, \Sigma, \mathbf{V}^{\top} = \text{SVD}(\bar{\xi}_{\mathbf{Y}})$ .

Further, we can use the Padé approximation to approximate geodesics on Grassmann manifolds as follows:

$$\mathbf{Y}(t) = [\mathbf{Y} \quad \mathbf{Y}_{\perp}] r_m(t\mathfrak{B}) \mathbf{I}_{D \times k} \approx \exp_{\mathbf{Y}}^{\text{Gr}}(t\bar{\xi}_{\mathbf{Y}}), \quad (117)$$

where  $r_m(\mathbf{X})$  is the  $m$ th-order diagonal Padé approximation to the matrix exponential  $\exp(\mathbf{X})$ . See the expression of  $r_m(\mathbf{X})$  in Moler & Loan (2003). The simplest member of this class is surely the first-order Padé approximation

$$\bar{R}_{\mathbf{Y}}(t\bar{\xi}_{\mathbf{Y}}) := [\mathbf{Y} \quad \mathbf{Y}_{\perp}] r_1(t\mathfrak{B}) \mathbf{I}_{D \times k} \quad (118)$$

$$= [\mathbf{Y} \quad \mathbf{Y}_{\perp}] \left( \mathbf{I}_n - \frac{t}{2} \mathfrak{B} \right)^{-1} \left( \mathbf{I}_D + \frac{t}{2} \mathfrak{B} \right) \mathbf{I}_{D \times k}, \quad (119)$$

which is also known as the Cayley transform. From the error expression  $\exp(\mathbf{Y}) = r_m(\mathbf{Y}) + O(\|\mathbf{Y}\|^{2m+1})$  of the Padé approximation, we have

$$\bar{R}_{\mathbf{Y}}(t\bar{\xi}_{\mathbf{Y}}) = \exp_{\mathbf{Y}}^{\text{Gr}}(t\bar{\xi}_{\mathbf{Y}}) + O(t^{2m+1} \|\bar{\xi}_{\mathbf{Y}}\|^{2m+1}), \quad (120)$$

which is also given by Theorem 3 in Gawlik & Leok (2018b).

## D.6.2 Horizontal Retraction

From Definition 3 in Zhu & Sato (2021), (119) is a horizontal retraction, and

$$R_{[\mathbf{Y}]}(t\xi_{[\mathbf{Y}]}) := \left[ \bar{R}_{\mathbf{Y}} \left( t\bar{\xi}_{\mathbf{Y}}^{\text{h}} \right) \right] \quad (121)$$

is a retraction on  $\text{Gr}(k, D)$  as a quotient manifold defined by (2). This is because  $\bar{R}$  satisfies the invariance condition that  $\bar{R}_{\mathbf{Y}} \left( t\bar{\xi}_{\mathbf{Y}}^{\text{h}} \right) \sim \bar{R}_{\mathbf{Y}'} \left( t\bar{\xi}_{\mathbf{Y}'}^{\text{h}} \right)$  for all  $\mathbf{Y} \in \text{St}(k, D)$ ,  $\mathbf{Y}' \in \text{St}(k, D)$ ,  $\bar{\xi}_{\mathbf{Y}} \in T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  and  $\bar{\xi}_{\mathbf{Y}'} \in T_{\mathbf{Y}'}^{\text{h}} \text{St}(k, D)$  such that  $\mathbf{Y} \sim \mathbf{Y}'$  and  $\bar{\xi}_{\mathbf{Y}}$  and  $\bar{\xi}_{\mathbf{Y}'}$  are horizontal lifts of  $\xi_{[\mathbf{Y}]} \in T_{[\mathbf{Y}]} \text{Gr}(k, D)$  at  $\mathbf{Y}$  and  $\mathbf{Y}'$ , respectively.

In low-rank cases, we can obtain an economical version of (119) as follows (Zhu & Sato (2021)).

$$\bar{R}_{\mathbf{Y}} \left( t\bar{\xi}_{\mathbf{Y}}^{\text{h}} \right) = \mathbf{Y} + t\bar{\xi}_{\mathbf{Y}}^{\text{h}} - \left( \frac{t^2}{2} \mathbf{Y} + \frac{t^3}{4} \bar{\xi}_{\mathbf{Y}}^{\text{h}} \right) \left( \mathbf{I}_k + \frac{t^2}{4} \bar{\xi}_{\mathbf{Y}}^{\text{h}\top} \bar{\xi}_{\mathbf{Y}}^{\text{h}} \right)^{-1} \bar{\xi}_{\mathbf{Y}}^{\text{h}\top} \bar{\xi}_{\mathbf{Y}}^{\text{h}}. \quad (122)$$

The inverse retraction  $\left( \bar{R}_{[\mathbf{Y}]}^{-1} \right)_{\mathbf{Y}}^{\text{h}} : \text{St}(k, D) \rightarrow T_{\mathbf{Y}}^{\text{h}} \text{St}(k, D)$  of  $\bar{R}_{\mathbf{Y}} \left( \bar{\xi}_{\mathbf{Y}}^{\text{h}} \right)$  is the following:

$$\left( \bar{R}_{[\mathbf{Y}]}^{-1}([\mathbf{X}]) \right)_{\mathbf{Y}}^{\text{h}} = \bar{R}_{\mathbf{Y}}^{-1}(\mathbf{X}) \quad (123)$$

$$= 2\mathbf{Y}_{\perp} \mathbf{Y}_{\perp}^{\top} \mathbf{X} (\mathbf{I}_k + \mathbf{Y}^{\top} \mathbf{X})^{-1} \quad (124)$$

$$= 2(\mathbf{X} - \mathbf{Y} \mathbf{Y}^{\top} \mathbf{X}) (\mathbf{I}_k + \mathbf{Y}^{\top} \mathbf{X})^{-1}. \quad (125)$$