GeoLatent: A Geometric Approach to Latent Space Design for Deformable Shape Generators

HAITAO YANG, The University of Texas at Austin, USA
BO SUN, The University of Texas at Austin, USA
LIYAN CHEN, The University of Texas at Austin, USA
AMY PAVEL, The University of Texas at Austin, USA
QIXING HUANG, The University of Texas at Austin, USA

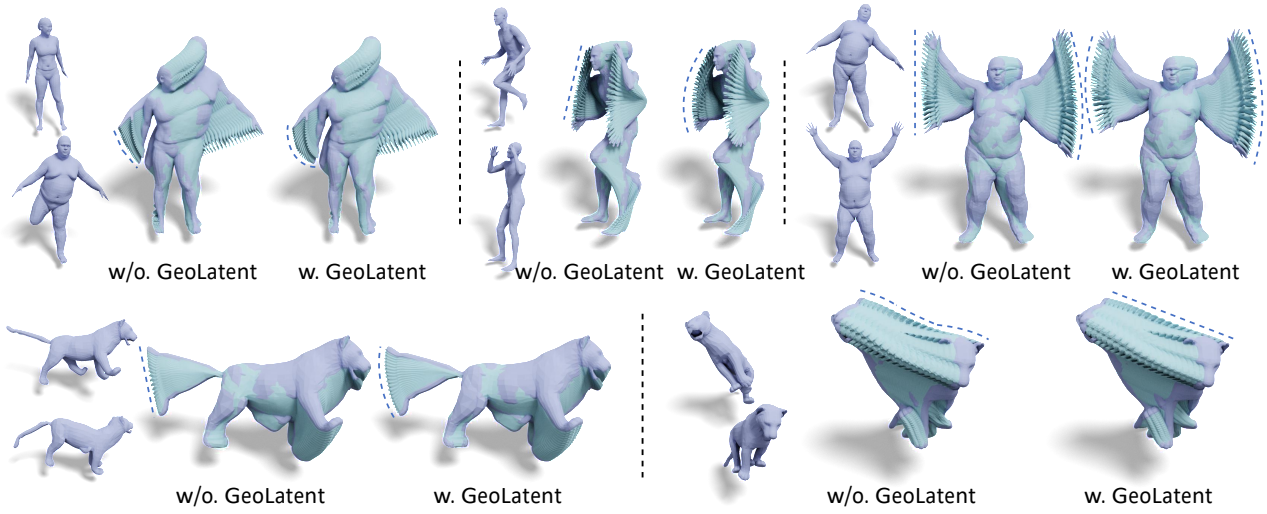


Fig. 1. For each group of shapes, we show (left) the source and target shape in shape interpolation, (middle) linear interpolation without latent space design, and (right) linear interpolation with latent space design. 20 interpolated shapes are shown. Linear interpolation with latent shape design preserves rigid part structures better.

We study how to optimize the latent space of neural shape generators that map latent codes to 3D deformable shapes. The key focus is to look at a deformable shape generator from a differential geometry perspective. We define a Riemannian metric based on as-rigid-as-possible and as-conformal-as-possible deformation energies. Under this metric, we study two desired properties of the latent space: 1) straight-line interpolations in latent codes follow geodesic curves; 2) latent codes disentangle pose and shape variations at different scales. Strictly enforcing the geometric interpolation property, however, only applies if the metric matrix is a constant. We show how to achieve this property approximately by enforcing that geodesic interpolations are axis-aligned, i.e., interpolations along coordinate axis follow geodesic curves. In addition, we introduce a novel approach that decouples pose and shape variations via generalized eigendecomposition. We also study eficient regularization terms for learning deformable shape generators, e.g., that promote smooth interpolations. Experimental results on benchmark datasets show that our approach leads to interpretable latent codes, improves the generalizability of synthetic shapes, and enhances performance in geodesic interpolation and geodesic shooting.

CCS Concepts: • Computing methodologies → Regularization; Shape analysis; • Mathematics of computing → Mesh generation.

Additional Key Words and Phrases: generative model, Riemannian geometry

ACM Reference Format:
Haitao Yang, Bo Sun, Liyan Chen, Amy Pavel, and Qixing Huang. 2023. GeoLatent: A Geometric Approach to Latent Space Design for Deformable Shape Generators. ACM Trans. Graph. 42, 6, Article 240 (December 2023), 20 pages. https://doi.org/10.1145/3618371

Authors' addresses: Haitao Yang, The University of Texas at Austin, 2317 Speedway, Austin, Texas, 78712, USA; Bo Sun, The University of Texas at Austin, 2317 Speedway, Austin, Texas, 78712, USA; Liyan Chen, The University of Texas at Austin, 2317 Speedway, Austin, Texas, 78712, USA; Amy Pavel, The University of Texas at Austin, 2317 Speedway, Austin, Texas, 78712, USA; Qixing Huang, The University of Texas at Austin, 2317 Speedway, Austin, Texas, 78712, USA.

0730-0301/2023/12-ART240 $15.00
https://doi.org/10.1145/3618371

1 INTRODUCTION

Thanks to advances in neural 3D geometric representations, there is growing interest in learning parametric neural shape generators that map latent codes to 3D shapes in some ambient space [Atzmon

et al. 2021; Huang et al. 2021; Litany et al. 2018; Muralikrishnan et al. 2022; Rakotosaona and Ovsjanikov 2020; Tan et al. 2018; Tretschk et al. 2020; Verma et al. 2018]. Most existing approaches generalize from their 2D counterparts, where the primary goal is to align the distribution of training shapes with that of synthetic shapes under suitable 3D shape representations. One fundamental difference between 2D images and 3D shapes is that a 3D shape space possesses rich geometric structures. Take deformable shapes, for example (e.g., humans and animals); the underlying deformations exhibit salient articulation structures. Merely performing distribution alignments typically incurs implicit regularizations (e.g., the magnitudes of network weights are minimized) that do not preserve such geometric structures, c.f. [Huang et al. 2021]. In this paper, we study deformable shape generators from a differential geometry perspective, i.e., a shape generator is a sub-manifold parameterized by the latent code. This view opens the door to developing novel techniques to make the latent space interpretable, improving the generalizability of synthetic shapes and enforcing geometric properties when performing shape interpolations and shape extrapolations.

Our approach, named GeoLatent, builds on a Riemannian metric of deformable shape generators derived from as-rigid-as-possible (ARAP) and as-conformal-as-possible (ACAP) deformation energies. Under this metric, we study two desired properties of the latent space: 1) straight-line interpolations in the latent space follow geodesic curves; 2) latent codes disentangle pose and shape variations. Strictly enforcing the geodesic interpolation property results in a very stiff constraint, i.e., the metric matrix is a constant. In light of the pose-shape disentanglement property, GeoLatent enforces geometric interpolation approximately so that it is satisfied along the axes of the latent space. GeoLatent then employs a novel generalized eigendecomposition formulation of the metric matrix to achieve pose-shape disentanglement. Compared to prior work [Aumentado-Armstrong et al. 2019; Cosmo et al. 2020; Mu et al. 2021; Zhou et al. 2020a] that performs hard pose-shape disentanglement, this new formulation is more flexible, e.g., it achieves progressive pose-shape disentanglement. We introduce how to enforce these two properties by developing suitable regularization terms.

Using this Riemannian metric, GeoLatent also incorporates shape regularization terms to promote that interpolated shapes are smooth and preserve local rigidity and conformality. Compared to merely minimizing the metric matrix [Huang et al. 2021], GeoLatent minimizes both the metric matrix and its gradients. This approach prevents large metric gradients from concentrating on the training instances. It leads to noticeable improvements in synthetic shapes in the presence of large and complex deformations.

GeoLatent can be combined with most generative models such as generative adversarial network (GAN) [Arjovsky et al. 2017; Goodfellow et al. 2020], variational auto-encoder (VAE) [Bouritsas et al. 2019; Litany et al. 2018; Ranjan et al. 2018; Tan et al. 2018], and auto decoder (AD) [Huang et al. 2021; Park et al. 2019; Zadeh et al. 2019], as well as any distance metrics for measuring the distances between training shapes and synthetic shapes. For simplicity, we study GeoLatent using AD.

We have evaluated the performance of GeoLatent on various deformable shape collections, such as humans and animals. Experimental results show that the latent space of GeoLatent is superior to existing approaches in visualizing shape variabilities, shape interpolations, and shape extrapolations. The resulting shape generator also exhibits better generalizability than baseline approaches. The effectiveness of GeoLatent is demonstrated in two applications, i.e., geodesic shape interpolation and geodesic shooting, in which the latent space of GeoLatent offers simpler solutions than alternative approaches.

In summary, GeoLatent presents the following contributions:

- A differential geometry view of deformable shape generators using a metric based on as-rigid-as-possible and as-conformal-as-possible deformation priors.
- Latent space design that combines axis-aligned geodesic interpolation and multi-scale pose-shape disentanglement based on the Riemannian metric.
- Shape regularization terms enhance the generalizability of synthetic shapes by minimizing the metric and its gradients under suitable norms.
- Applications in geodesic shape interpolation and geodesic shooting.

2 RELATED WORKS

We discuss relevant works in five categories, i.e., deformable shape generators (Section 2.1), regularizations for shape generators (Section 2.2), pose-shape disentanglement in deformable shape generators (Section 2.3), deformation models (Section 2.4), and differential geometry in generative modeling (Section 2.5).

2.1 Deformable Shape Generators

Most existing works on deformable shape generators focus on architecture design under different 3D representations. Several early approaches [Kolotouros et al. 2019; Litany et al. 2018; Rakotosaona and Ovsjanikov 2020; Tan et al. 2018; Tretschk et al. 2020; Verma et al. 2018] define variational auto-encoders on meshes. A common strategy is to treat triangular meshes as graphs and define convolution and deconvolution operations to synthesize triangular meshes (c.f. [Atzmon et al. 2022; Bouritsas et al. 2019; Chen and Kim 2021; Litany et al. 2018; Tretschk et al. 2020; Verma et al. 2018; Yuan et al. 2020]). In [Atzmon et al. 2022], the authors utilized the Frame Averaging framework for equivariant shape space learning. However, the method requires skinning weights to achieve good performance, which are usually unavailable for general shape collections. Several other approaches studied (deformable) shape generation under the point cloud representation, including PointFlow [Yang et al. 2019], LGF [Cai et al. 2020], and DPM [Luo and Hu 2021].

In contrast to these approaches, GeoLatent focuses on principles for designing the latent space. The goals include making the latent space interpretable and reducing the size of the training data by enforcing geometric regularizations. GeoLatent can be applied to deformable shape generators under any shape representation.

2.2 Regularizations for Shape Generators

Regularization losses have been explored in prior work for generative models. In [Peebles et al. 2020], Peebles et al. studied a Hessian regularization term for learning generative image models. A spectral regularization loss is introduced in [Aumentado-Armstrong et al.

2019] for 3D generative modeling. Several papers [Balashova et al. 2018; Kanazawa et al. 2018; Wang et al. 2018; Wu et al. 2018] studied geometric regularizations for image-based reconstruction.

Several recent papers studied enforcing geometric regularization to improve 3D shape generation quality. In [Cosmo et al. 2020], the authors studied a distance matrix-based representation under a VAE generative model and proposed to regularize the shape generator so that linear interpolations in the latent space between shape pairs are consistent with linear interpolations between corresponding distance matrices. This approach, however, does not model potentially large variations in a shape collection. Another limitation is the distance matrix representation, which is not ideal for modeling articulated deformations. Both issues are addressed in ARAPReg [Huang et al. 2021], which is based on regularizing the tangent space of the shape manifold defined by the mesh generator under an as-rigid-as-possible deformation model. In GenCorres [Yang et al. 2023], the authors studied geometric regularizations of deformable shape generators under an implicit representation for inter-shape correspondence computation. GeoLatent innovates in optimizing the latent space of the shape generator. Moreover, both ARAPReg and GenCorres define regularization losses on the metric matrix. In contrast, we study high-order regularizations based on metric gradients.

2.3 Pose-Shape Disentanglement

Motivated from style-content separation in latent spaces of image-based generative models [Karras et al. 2019; Nguyen-Phuoc et al. 2019; Zhang et al. 2018], many 3D generation approaches have considered similar latent space separations. For man-made shapes, several recent papers [Li et al. 2017; Mo et al. 2019; Yang et al. 2022] studied how to divide latent codes into part latent codes and geometry latent codes. However, these approaches typically require part-labels as inputs. For deformable generative models, several papers [Aumentado-Armstrong et al. 2021, 2019; Cosmo et al. 2020; Foti et al. 2023; Levinson et al. 2019; Mu et al. 2021; Tatro et al. 2020; Zhou et al. 2020a] examined pose and shape disentanglement. In particular, [Zhou et al. 2020a] introduced an unsupervised approach to decouple pose and shape variables. The idea is to separate intrinsics and extrinsics using corresponding meshes known only to have the same shape but different pose and apply an ARAP geometric prior to perform disentanglement. [Foti et al. 2023] introduced a spectral technique to decouple latent codes that reflect local shape attributes. Finally, [Aumentado-Armstrong et al. 2021] learned a generative representation that factorizes the latent deformation space into shape and pose, without supervision.

GeoLatent takes a different approach that uses generalized eigenvectors to achieve pose-shape disentanglement. The generalized eigenvalue problem is driven by the ARAP and ACAP deformation energies, where pose variations have small values and shape variations have large values. In this way, GeoLatent achieves multi-scale disentanglement, which is more general than pose-shape disentanglement. GeoLatent also does not require pose/identity labels.

2.4 Deformation Models

Isometric deformation models, which preserve the edge lengths of a mesh, are among the early deformation models for non-rigid geometry [Bronstein et al. 2009]. One issue of these models is that they are not suitable for large non-rigid deformations and do not model deformations that exhibit piece-wise structures well, which are popular in organic shapes. To handle piece-wise structures, more recent deformation models enforce that the transformations within local surface patches can be approximated well by afine transformations [Sumner et al. 2007], rigid transformations [Sorkine and Alexa 2007; Wand et al. 2007; Xu et al. 2007], and similarity trans-formations [Yoshiyasu et al. 2014], which are termed as-afine-as possible (AAAP), as-rigid-as-possible (ARAP), and as-conformal-as-possible (ACAP) deformation models, respectively. These defor-mation models also root in rich literature in computer graphics on mesh parameterization [Bouaziz et al. 2012; Crane et al. 2011; Liu and Jacobson 2021; Vaxman et al. 2015].

Prior work has explored all these models to regularize neural shape spaces. Many approaches [Habermann et al. 2020; Li et al. 2020; Zhou et al. 2020a] enforce ARAP losses between a base shape and synthetic shapes. They are not ideal for shape collections that possess significant inter-shape variations. GeoLatent is relevant to ARAPReg [Huang et al. 2021] and GenCorres [Yang et al. 2023], which enforce ARAP and ACAP losses among neighboring synthetic shapes. However, GeoLatent focuses on using the Riemannian metrics of ARAP and ACAP losses to design the latent space of a deformable shape generator. In principle, GeoLatent can incorporate any deformation model among neighboring synthetic shapes.

2.5 Differential Geometry of Generative Models

Several papers modeled image generative models as sub-manifolds [Arvanitidis et al. 2018; Chen et al. 2018; Kalatzis et al. 2020; Kühnel et al. 2018; Wang and Ponce 2021]. A central topic is geodesic curve computation. [Chen et al. 2018; Shao et al. 2018] studied optimizing a curve in the latent space so that the induced curve in the ambient space follows a geodesic curve based on a metric associated with the ambient space. [Shao et al. 2018] also studied other operations, such as parallel translation and geodesic shooting. [Arvanitidis et al. 2018; Yang et al. 2018] extended geodesic interpolations from deterministic generative models to stochastic generative models. [Stolberg-Larsen and Sommer 2022] studied geodesic computation under the setting of multi-chart generative models, emphasizing the smoothness of geodesic paths across chart boundaries. However, these approaches build on pulling back the Euclidean metric of the ambient space. Several methods used other means to define the Riemannian metric. For example, [Roy and Hauberg 2022] pulled back the Wasserstein metric tensor on the decoder distributions to the latent space. Exact geodesic computation involves computing the Hessian of the generator, which is computationally expensive. [Chen et al. 2019] studied an approximate approach based on sampling the latent space. Moreover, several approaches [Arvanitidis et al. 2021; Petzka et al. 2022] studied defining a metric of the ambient space to induce the metric of the latent space. However, their primary goal is to drive geodesic interpolations to follow interesting regions of the ambient space.

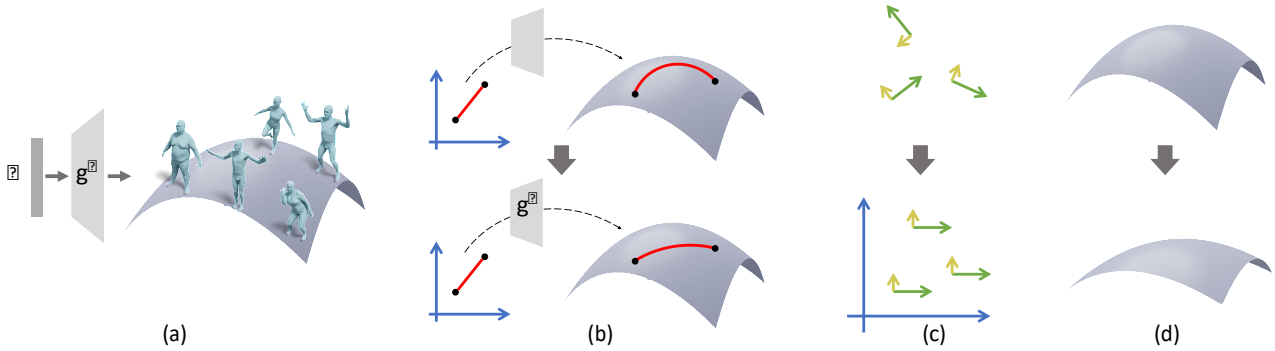


Fig. 2. (a) GeoLatent learns a generative model to fit a shape collection using (b-c) latent space regularizations and (d) shape space regularizations. (b) Linear interpolation in the latent space corresponds to geodesic interpolation in the shape space. (c) Enforcing eigenvectors to be axis-aligned converts local pose-shape disentanglement into global pose-shape disentanglement. Yellow and green arrow indicate pose and shape direction, respectively. (d) Minimization of the elements and the norm of the gradients of the Riemannian metric improves generalizability of the generator.

In contrast, the metric of GeoLatent is derived from 3D deformable models. Moreover, GeoLatent focuses on achieving desired geodesic interpolation and pose-shape disentanglement properties.

3 PROBLEM STATEMENT AND APPROACH OVERVIEW

We begin with a problem statement in Section 3.1. We then present an overview of the proposed GeoLatent in Section 3.2.

3.1 Problem Statement

The input to GeoLatent is a collection of 3D shapes $S = \{\square_1, \cdots, \square_\square\}$ that are samples from an underlying deformable shape space. Our goal is to train a mesh generator $g^\square : Z \to R^{3\square}$ that maps a latent code $z \in Z = R^\square$ to a triangular mesh of $\square$ vertices with pre-defined topology. Here $\square$ denotes the network parameters.

As our primary goal is to examine latent space design, we focus on the correspondence-based setting [Atzmon et al. 2022; Bouritsas et al. 2019; Huang et al. 2021; Muralikrishnan et al. 2022; Ranjan et al. 2018; Zhou et al. 2020b], where we assume $\square_\square$ are consistently meshed with $\square$ vertices. Our goal is to optimize a latent code $z_\square$ of each shape $\square_\square$ and the network parameters $\square$ to achieve five goals:

(1) The geometric shape of $g^\square(z_\square)$ aligns with that of $\square_\square$;
(2) The empirical distribution of the latent codes $\{z_\square\}$ aligns with a prior latent distribution, e.g., the Gaussian $N_\square$.
(3) Linear interpolations in the latent space follow approximate geodesic curves under a suitable deformation metric.
(4) Axes of the latent space disentangle deformations at different scales, ranging from pose variations to shape variations.
(5) Interpolations in the latent space are smooth.

3.2 Overview of GeoLatent

As illustrated in Figure 2, GeoLatent optimizes the generator $g^\square$ by minimizing a total loss that combines a data loss and two regularization losses:

$$\min_{\square} \ \square_{AD}(g^\square, S) + \square_{latent}\square_{latent}(\square) + \square_{shape}\square_{shape}(\square) \qquad (1)$$

The data loss is an auto-decoder (AD) loss that optimizes the decoder $g^\square$ to fit the training shapes S. The details are explained in Section 7. $\square_{latent}(\square)$ is a regularization term prioritizing that the latent space satisfies the desired properties of approximate geodesic interpolation and pose-shape disentanglement. $\square_{shape}(\square)$ regularizes the generated shapes of $g^\square$.

We organize the remainder of this paper as follows. GeoLatent builds on a Riemannian metric on $g^\square$, which will be discussed in Section 4. In Section 5, we study desired properties of the latent space under this metric, which are our key contributions. We then introduce how to model $\square_{latent}(\square)$ and $\square_{shape}(\square)$ in Section 6. Section 7 discusses the training procedure. Section 8 and Section 9 present the experimental results and applications, respectively. Finally, we conclude this paper in Section 10.

4 A RIEMANNIAN METRIC

We aim at defining a Riemannian metric for a deformable mesh generator $g^\square(z) : Z := R^\square \to R^{3\square}$ that maps a latent code z to vertex positions of a triangular mesh $M = (V, E)$ with $\square$ vertices. A well-known Riemannian framework for deformable shape analysis is Sobolev metric developed by Bauer and co-authors [Bauer et al. 2021, 2011; Hartman et al. 2023; Su et al. 2020], which roots in early work [Mennucci et al. 2008; Michor and Mumford 2005, 2007]. While Sobolev metrics are well grounded in theory, they are intrinsic metrics which are not ideal to encode articulated deformations. Such deformations are extrinsic in nature and extremely popular in real deforming objects. In contrast, our formulation largely follows ARAPReg [Huang et al. 2021] and GenCorres [Yang et al. 2023], which are extrinsic deformation models. In particular, we make the definition of the Riemannian metric explicit, which is the foundation of GeoLatent.

Specifically, we define the Riemannian metric based on measuring the distortion between a synthetic shape $g^\square(z)$ and its perturbation $g^\square(z + \square v)$, where $v \in R^\square$ is the unit direction of perturbation and $\square$ is an infinitesimal value. The distortion then induces a metric in the tangent space at z. To this end, we consider two deformation models.

The first deformation model, termed the as-rigid-as possible (ARAP) model [Huang et al. 2021], measures the cumulative deviations between the displacements $g_i(z + \epsilon v) - g(z)$, $i \in N_i$ in the 1-ring neighbor N_i of each vertex g_i from a rotation. In the infinitesimal regime, we parameterize the rotation as $I_3 + c_i \times$, $c_i \in R^3$, leading to

$$\mathcal{E}_{arap}(z, v) = \sum_{i=1}^{n} \min_{c_i} \sum_{j \in N_i} \| (I_3 + c_i \times)(g_i(z) - g_j(z)) - (g_i(z + \epsilon v) - g_j(z + \epsilon v)) \|^2 \quad (2)$$

The following proposition, due to [Huang et al. 2021, 2009], specifies that $\mathcal{E}_{arap}(z, v)$ is a quadratic form in $g(z + \epsilon v) - g(z)$.

Proposition 1. ([Huang et al. 2021, 2009]) We have

$$\mathcal{E}_{arap}(z, v) = (g(z + \epsilon v) - g(z))^T L_{arap}(z) (g(z + \epsilon v) - g(z))$$

where

$$L_{arap}(z) = L \otimes I_3 - A_{arap}(z) D_{arap}(z)^{-1} A_{arap}(z)^T$$

where $L \in R^{n \times n}$ is the unnormalized graph Laplacian matrix of $g(z)$; $A_{arap}(z) \in R^{3n \times 3n}$ and $D_{arap}(z) \in R^{3n \times 3n}$ are $n \times n$ sparse block and diagonal block matrices whose blocks are given by

$$A_{arap,ij}(z) = \begin{cases} e_{ij}'(z) \times & j \in N_i \\ e_{ij}(z) \times & j \in N \\ 0 & \text{otherwise} \end{cases}$$

$$D_{arap,ii}(z) = \sum_{j \in N_i} e_{ij}(z) e_{ij}(z)^T I_3 - e_{ij}'(z) e_{ij}'(z)^T$$

where $e_{ij}(z) = g_i(z) - g_j(z)$.

This ARAP model is effective when the absolute scales of the input shapes remain similar. However, typical deformable shape collections present great scale variations, e.g., tall people vs short people and fat people vs thin people. To address this issue, we introduce the second deformation model, termed the as-conformal-as possible (ACAP) [Yang et al. 2023] model. Specifically, ACAP measures the cumulative deviations between the displacements $g_i(z + \epsilon v) - g_i(z)$, $i \in N_i$ in the 1-ring neighbor N of each vertex g_i from a similarity transformation. Similar to (2), we parameterize the similarity transformation by $(1 + s_i) I_3 + c_i \times$, leading to

$$\mathcal{E}_{acap}(z, v) = \sum_{i=1}^{n} \min_{c_i, s_i} \sum_{j \in N} \| (I_3 + s_i I_3 + c_i \times)(g_i(z) - g_j(z)) - (g_i(z + \epsilon v) - g_j(z + \epsilon v)) \|^2 \quad (3)$$

Similar to $\mathcal{E}_{arap}(z, v)$, $\mathcal{E}_{acap}(z, v)$ is also a quadratic form in $g(z + \epsilon v) - g(z)$.

Proposition 2. ([Yang et al. 2023]) We have

$$\mathcal{E}_{acap}(z, v) = (g(z + \epsilon v) - g(z))^T L_{acap}(z) (g(z + \epsilon v) - g(z))$$

where

$$L_{acap}(z) = L \otimes I_3 - A_{acap}(z) D_{acap}(z)^{-1} A_{acap}(z)^T$$

where $A_{acap}(z) \in R^{3n \times 4n}$ and $D_{acap}(z) \in R^{4n \times 4n}$ are $n \times n$ sparse block and diagonal block matrices whose blocks are given by

$$A_{acap,ij}(z) = \begin{cases} e_{ij}'(z) \times, -e_{ij}(z) & j \in N_i \\ e_{ij}(z) \times, -e_{ij}(z) & j \in N \\ 0 & \text{otherwise} \end{cases}$$

$$D_{acap,ii}(z) = \sum_{j \in N_i} \begin{bmatrix} e_{ij}(z) e_{ij}(z)^T I_4 - \begin{bmatrix} e_{ij}'(z) e_{ij}'(z)^T & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix}$$

When $\epsilon \to 0$, we have $g(z + \epsilon v) - g(z) = \epsilon \frac{\partial g}{\partial z}(z) v$. In this case,

$$\mathcal{E}_{arap}(z, v) = \epsilon^2 v^T \frac{\partial g}{\partial z}(z)^T L_{arap}(z) \frac{\partial g}{\partial z}(z) v \quad (4)$$

$$\mathcal{E}_{acap}(z, v) = \epsilon^2 v^T \frac{\partial g}{\partial z}(z)^T L_{acap}(z) \frac{\partial g}{\partial z}(z) v \quad (5)$$

Combing (4) and (5), we arrive at the following definition.

Definition 1. Denote $L_\alpha(z) = (1 - \alpha) L_{arap}(z) + \alpha L_{acap}(z)$, where $0 \le \alpha \le 1$ is a tradeoff parameter between ARAP and ACAP deformations. We define the Riemannian metric of a deformation shape generator $g(z)$ as

$$M(z) := \frac{\partial g}{\partial z}(z)^T L_\alpha(z) \frac{\partial g}{\partial z}(z) \quad (6)$$

Numerically, we compute $\frac{\partial g}{\partial z}(z)$ using finite difference, i.e.,

$$\frac{\partial g}{\partial z}(z) = \frac{1}{\epsilon} [g(z + \epsilon e_1) - g(z), \cdots, g(z + \epsilon e_n) - g(z)] \quad (7)$$

where e_i is the canonical basis of R^n. Note that according to (7), the metric matrix $M(z)$ is a function on the vertex coordinates of the generator $g(z)$, allowing for eficient computation.

Note that unlike most Riemannian metrics of generative models which are pull-back metrics of the Euclidean metric of the ambient space [Arvanitidis et al. 2018; Chen et al. 2018; Kalatzis et al. 2020; Wang and Ponce 2021], $M(z)$ can be understood as the pull-back metric of a metric that measures deformations in the ambient space.

5 LATENT SPACE DESIGN

This section studies desired properties of the latent space guided by the metric $M(z)$. The desired properties fall into two categories, i.e., geodesic interpolation and pose-shape disentanglement, which we examine in Section 5.1 and Section 5.2, respectively.

5.1 Geodesic Interpolation

We begin with introducing the Christoffel symbols induced from $M(z)$, which are used to characterize geodesic curves.

Definition 2. The induced Christoffel symbols from $M(z)$ are given by $\Gamma_{ij,k}(z) := \{1, \cdots, n\}$,

$$\Gamma_{ij}^k(z) = \frac{1}{2} \sum_{l=1}^{n} (M(z)^{-1})_{kl} \left(\frac{\partial M_{jl}}{\partial z_i}(z) + \frac{\partial M_{il}}{\partial z_j}(z) - \frac{\partial M_{ij}}{\partial z_l}(z) \right). \quad (8)$$

where $(M(z)^{-1})_{kl}$ is the klth entry of the inverse of $M(z)$.

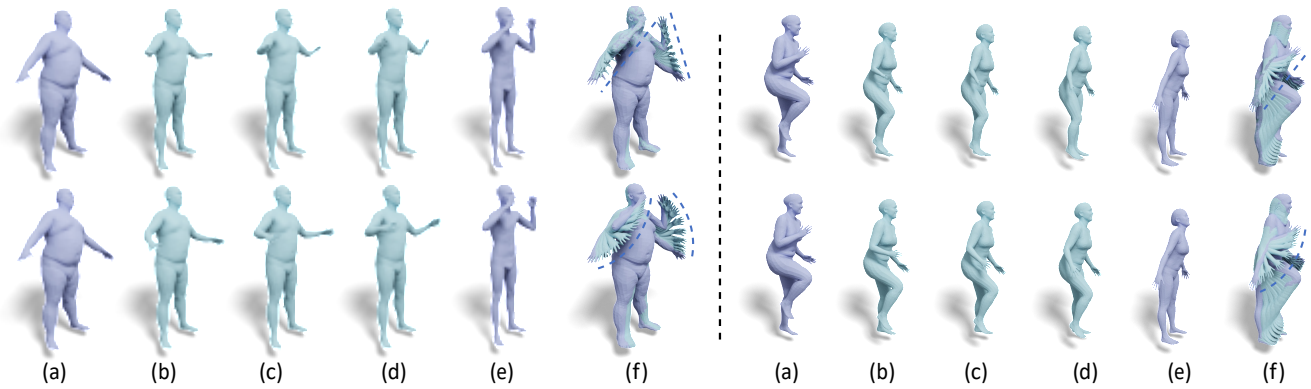


Fig. 3. Comparisons between linear interpolations and geodesic interpolations using the pre-trained ARAPReg model [Huang et al. 2021]. Under large deformations, (bottom) geodesic interpolations are more meaningful than (top) linear interpolations, where geometric features such as fingers are highly distorted in the interpolated shapes. (a, e) Source and target shape. (b-d) Sampled interpolated shapes. (f) 10 interpolated shapes are shown.

It is easy to check that $\Gamma^{\alpha,\beta}_{\gamma\gamma}(z) = \Gamma^{\beta,\alpha}_{\gamma\gamma}(z)$.

In the context of a shape space, a popular scheme for shape interpolation is that the interpolated shapes follow a geodesic curve, c.f., [Kilian et al. 2007]. For neural generative models, linear interpolations between latent codes are the most widely used. Therefore, a desirable property of the latent space is that the straight line interpolation $g^{\alpha}(z_1 + t(z_2 - z_1)), t \in [0, 1]$ between any two latent codes z_1 and z_2 follows a geodesic curve with respect to $g^{\alpha}(z)$ (See Figure 3). The following theorem states that the geodesic interpolation property is satisfied iff the metric matrix is a constant.

Theorem 1. The straight line interpolation $g_{\alpha}(z_1 + t(z_2 - z_1)), t \in [0, 1]$ between any two latent codes z_1 and z_2 follows a geodesic curve, i.e., there exists a reparameterization $t(t')$ so that the interpolation is a geodesic in t', iff the metric matrix $g^{\alpha}(z)$ is a constant.

Proof. The proof, which is non-trivial, is deferred to Section A. □

Therefore, we relax the desired geodesic interpolation property. Theorem 1 indicates that if we ensure that geodesic interpolations are satisfied within each 2D sub-spaces formed by e_α, e_β, then $g^{\alpha}(z)$ is still a constant. Thus, we propose to constrain that the geodesic interpolation property is satisfied along each axis $g^{\alpha}(z + te_\alpha)$ at each latent code z, relaxing 2D sub-spaces to 1D sub-spaces.

As we will discuss later, the motivation of enforcing axis-aligned geodesic interpolation is two-fold. First, geodesic interpolation is still satisfied approximately. Second, it enables the formulation of pose-shape disentanglement locally, which induces global pose-shape-disentanglement.

Definition 3. A metric $g^{\alpha}(z)$ satisfies the axis-aligned geodesic interpolation property iff for each $\alpha \in [n]$ and every z, there exists a parameterization $t(t')$, so that $g^{\alpha}(z + t(t')e_\alpha)$ is geodesic in t'.

The following proposition characterizes a necessary and suficient condition on axis-aligned geodesic interpolation property.

Proposition 3. The axis-aligned geodesic interpolation property is satisfied iff

$$\Gamma^{\alpha,\beta}_{\gamma}(z) = 0, \quad \alpha \neq \beta \neq \gamma \neq \alpha \in [n] \tag{9}$$

$$\frac{\partial g_{\alpha\beta}}{\partial z_\gamma}(z) = 2\Gamma^{\alpha,\beta}_{\gamma}(z) \quad \alpha \neq \beta \in [n] \tag{10}$$

Proof. See Appendix B. □

The following proposition specifies that axis-aligned geodesic interpolation can be achieved locally via re-parameterization.

Proposition 4. Given a shape generator $g^{\alpha}(z)$ and a latent code z_0, there exists a reparameterization $z(z') : R^n \to R^n$ where $z(0) = z_0$ and a radius r, such that axis-aligned geodesic interpolation is satisfied under z' in the ball $B_r(0) = \{z' \mid \|z'\| \leq r\}$.

Proof. See Appendix C. □

Note that the parameterization described in Prop. 4 is different from the geodesic parameterization (via exponential map) (pp 141)[Petersen 2006] on Riemannian manifolds. In fact, the coordinate differences along each axis may not be identical to the length along the corresponding geodesic curve. When $n = 2$ and the $g^{\alpha}(z)$ is the pull-back metric from R^3, the parameterization is the so-called geodesic web [Pottmann et al. 2010; Vekhter et al. 2019].

Even though axis-aligned geodesic interpolation only constrains that geodesics follow the axes, geodesic interpolation between two arbitrary points is satisfied approximately (See Figure 9). This property is essential because linear interpolations, which are easy to compute, are geometrically meaningful because they roughly follow geodesic curves. On the other hand, axis-aligned geodesic interpolation allows some flexibility in the shape generator compared to exact geodesic interpolations.

5.2 Pose-Shape Disentanglement

Besides geodesic interpolation, another desired property is that different axes of the latent space disentangle shape variations and pose variations. As studied in prior works [Aumentado-Armstrong

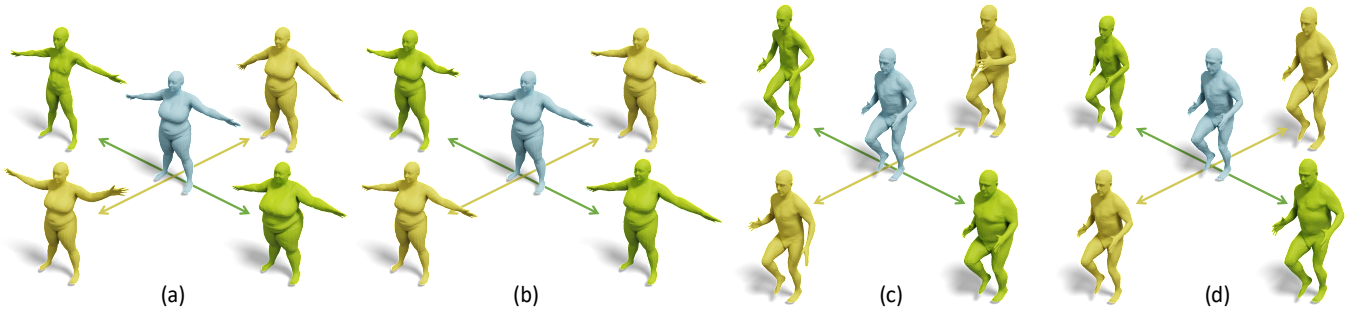


Fig. 4. Extrapolations using ARAPReg [Huang et al. 2021] along the maximum eigenvector (green) and minimum eigenvector (yellow) from the base shape (blue). Extrapolations under generalized eigenvectors of $\mathcal{H}^{\mathcal{E}}(z)$ with respect to $\mathcal{H}^{\mathcal{M}}(z)$ (a, c) provide better pose-shape disentanglement than under normal eigenvectors of $\mathcal{H}^{\mathcal{E}}(z)$ (b, d).

et al. 2019; Mu et al. 2021; Zhou et al. 2020a], a traditional definition of pose-shape disentanglement is that one can decompose the latent space $Z = Z_{shape} \times Z_{pose}$, so that at any latent code z_0, $g^{\mathcal{M}}(z_0 + v)$ varies the pose(shape) of $g^{\mathcal{M}}(z_0)$ if $v \in Z_{pose}(Z_{shape})$. An issue of this approach is that without pose and shape labels (which are hard to obtain), the boundary between pose and shape variations is fuzzy.

GeoLatent achieves pose-shape disentanglement using a simple scoring function of a tangent direction $v \in \mathbb{R}^{\mathcal{D}}$ at z:

$$\mathcal{S}_z(v) := \frac{v^{\mathcal{T}}\mathcal{H}^{\mathcal{E}}(z)v}{v^{\mathcal{T}}\mathcal{H}^{\mathcal{M}}(z)v}, \quad \mathcal{H}^{\mathcal{M}}(z) = \frac{\partial g^{\mathcal{M}}}{\partial z}(z)^{\mathcal{T}}(\mathcal{I}_{3\mathcal{V}} - \mathcal{P}\mathcal{P}^{\mathcal{T}})\frac{\partial g^{\mathcal{M}}}{\partial z}(z), \quad (11)$$

where $\mathcal{P} \in \mathbb{R}^{3\mathcal{V}\times 6}$ collects an orthonormal basis of the rigid body motion space, i.e.,

$$\begin{bmatrix} \mathcal{I}_3 & g_1^{\mathcal{M}}(z)\times \\ \vdots & \vdots \\ \mathcal{I}_3 & g_{\mathcal{V}}^{\mathcal{M}}(z)\times \end{bmatrix} = \mathcal{P}\,\mathcal{R}$$

is the QR decomposition. When the norm of v is small, the numerator of (11) measures the distortion of the deformation specified by v. The denominator of (11) measures the magnitude of the corresponding vertex displacements, in which the rigid component is factored out.

It is easy to check that v corresponds to a pose variation if $\mathcal{S}_z(v)$ is small. In this case, the distortions from rotations and similarity transformations among the 1-ring neighborhoods within each shape part are small, while the absolute displacements are large. In contrast, v corresponds to a shape variation if $\mathcal{S}_z(v)$ is large. In this case, distortions from rotations and similarity transformations among the 1-ring neighborhoods are large in the interior regions of shape parts, while vertex displacements are local. The nice thing of using $\mathcal{S}_z(v)$ is that it offers multi-scale disentanglement rather than the binary disentanglement in prior work.

Figure 4 illustrates generalized eigenvectors $u_{\mathcal{I}}$ that satisfy $\mathcal{H}^{\mathcal{E}}(z)u_{\mathcal{I}} = \lambda_{\mathcal{I}}\mathcal{H}^{\mathcal{M}}(z)u_{\mathcal{I}}$. We can see that these eigenvectors nicely disentangle pose and shape variations. Note that generalized eigenvectors, which consider magnitudes of the shape variations, provide better disentanglement than the eigenvectors of $\mathcal{H}^{\mathcal{E}}(z)$.

Definition 4. We say a shape generator $g^{\mathcal{M}}(z)$ possesses pose-shape disentanglement if the canonical axes $e_{\mathcal{I}}$ are generalized eigenvectors, i.e., there exist $\lambda_{\mathcal{I}}$ so that

$$\mathcal{H}^{\mathcal{E}}(z)e_{\mathcal{I}} = \lambda_{\mathcal{I}}\mathcal{H}^{\mathcal{M}}(z)e_{\mathcal{I}}. \quad (12)$$

While (12) only constrains multi-scale pose-shape disentanglement locally, combing axis-aligned geodesic interpolation generalizes multi-scale pose-shape disentanglement globally (See Figure 5).

Note that unlike axis-aligned geodesic interpolation that may be achieved by reparameterization without changing the generated shapes $g^{\mathcal{M}}(z)$, not all shape generators can achieve both axis-aligned geodesic interpolation and pose-shape disentanglement. Therefore, enforcing both properties also provide regularizations on the underlying shape generator.

6 REGULARIZATION TERMS

This section studies regularization terms on the generator $g^{\mathcal{M}}(z)$. We begin with regularization terms that enforce the desired properties of the latent space in Section 6.1. We then study regularization terms on the shape space defined by $g^{\mathcal{M}}(z)$ in Section 6.2. These shape space regularization terms determine the best shape generator that satisfies the latent space properties.

6.1 Latent Space Regularization Terms

Given a prior distribution of the latent space, e.g., the Gaussian distribution $\mathcal{N}_{\mathcal{I}}$, we define the latent space regularization term as

$$\mathcal{L}_{\text{latent}}(\mathcal{I}) = \mathcal{I}_{z\sim\mathcal{N}_{\mathcal{I}}}\mathcal{L}_{\text{latent}}(z, \mathcal{I}) \quad (13)$$

where $\mathcal{L}_{\text{latent}}(z, \mathcal{I})$ measures deviations of (9), (10), and (12).

One challenge of enforcing (9) and (10) is that each $\Gamma^{\mathcal{I}}_{\mathcal{I}\,\mathcal{I}}$ is the sum of $\mathcal{I}$ terms, making it dificult to optimize. To address this issue, we use a suficient condition on (9) and (10) that only involves the gradients of $\mathcal{H}^{\mathcal{E}}(z)$.

Proposition 5. (9) and (10) are satisfied if

$$2\frac{\partial \lambda_{\mathcal{I}}}{\partial z}(z) = \frac{\partial \mathcal{I}_{\mathcal{I}\mathcal{I}}}{\partial z}(z), \quad \mathcal{I}, \mathcal{I}, \mathcal{I} \in [\mathcal{I}]. \quad (14)$$

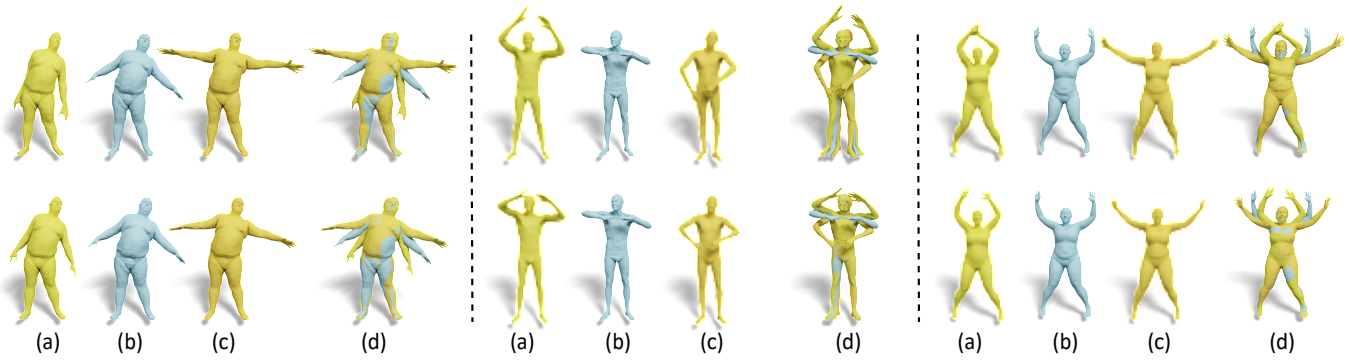


Fig. 5. Comparison of enforcing local disentanglement and global disentanglement. (Top) Merely enforcing local disentanglement described in (12) does not generalize well under large latent code perturbations. (Bottom) Global disentanglement is achieved by enforcing both (12) and axis-aligned geodesic interpolation. The shapes (a) and (c) are generated by perturbing the latent codes of the shape (b) along the direction in the latent space that corresponds to the smallest generalized eigenvalue. Shape (a), (b), and (c) are aligned in the same coordinate system for better comparison (d). Please check the orientation differences from (a) to (b) and from (b) to (c) in the regions of hands.

Proof. Note that

$$\Gamma_{ij}^{k} = \sum_{l=1}^{k} \left(G^{\theta}(z)^{-1}\right)_{kl} \left(2\frac{\partial g_{il}^{\theta}}{\partial z_j} - \frac{\partial g_{ij}^{\theta}}{\partial z_l}\right).$$

So if (14) holds, then $\Gamma_{ij}^{k} = 0, \forall i, j, k \in [k]$. Also $\frac{\partial g_{ij}^{\theta}}{\partial z_k}(z) = 0$ if (14) holds. Therefore, both (9) and (10) hold. □

We proceed to enforce pose-shape disentanglement. The challenge of enforcing (12) is that they involve unknown variables λ_i. We address this issue by enforcing an equivalent set of constraints:

$$G_{sp}^{\theta}(z)G_{ps}^{\theta}(z) = G_{ss}^{\theta}(z)G_{pp}^{\theta}(z), \qquad \forall s, p \in [k] \tag{15}$$

With this setup, we define $r_{\text{latent}}(z, \theta)$ by enforcing (14) and (15) in the least square sense:

$$\begin{aligned} r_{\text{latent}}(z, \theta) = & \sum_{i=1}^{k}\sum_{j=1}^{k}\left\| 2\frac{\partial g_{ij}^{\theta}}{\partial z_j}(z) - \frac{\partial g_{jj}^{\theta}}{\partial z_i}(z)\right\|^2 \\ & + \lambda_{ps}\sum_{s=1}^{k}\left\| G_{sp}^{\theta}(z)G_{ps}^{\theta}(z) - G_{ss}^{\theta}(z)G_{pp}^{\theta}(z)\right\|^2 \end{aligned} \tag{16}$$

where λ_{ps} is the trade-off parameter in front of the pose-shape disentanglement term.

We use finite difference to compute the elements of the metric matrix and its derivatives in (12):

$$\begin{aligned} G_{ij}^{\theta}(z) = & \frac{1}{\epsilon^2}\left(g^{\theta}(z+\epsilon e_i) - g^{\theta}(z)\right)^{T} M^{\theta}(z)\left(g^{\theta}(z+\epsilon e_j) - g^{\theta}(z)\right) \\ \frac{\partial G_{ij}^{\theta}}{\partial z_k}(z) = & \frac{1}{\epsilon^3}\left(g^{\theta}(z+\epsilon e_i) - g^{\theta}(z)\right)^{T}\left(M^{\theta}(z+\epsilon e_k) - M^{\theta}(z)\right) \\ & \left(g^{\theta}(z+\epsilon e_j) - g^{\theta}(z)\right) \end{aligned} \tag{17}$$

where ϵ is a hyperparameter.

Figure 5 shows that the first term in (16) is critical for achieving global multi-scale pose-shape disentanglement.

6.2 Shape Space Regularization Terms

Merely constraining the latent space does not explicitly regularize the shape space defined by the shape generative model $g^{\theta}(z)$. Besides desired properties of the latent space, we also want to regularize synthetic shapes, so that they offer smooth interpolations between input shapes. To this end, we propose to minimize regularization terms that are defined on $g^{\theta}(z)$.

One approach is to minimize the elements of $G^{\theta}(z)$. However, merely doing so may adversely compete against fitting the generator to the training shapes. Another issue is that it will force large gradients of $G^{\theta}(z)$ to concentrate at the training shapes. This could be understood through the effects of merely minimizing the elasticity term in the active contour model [Kass et al. 1988], which leads to piece-wise linear interpolations.

A popular approach to address this issue is to combine the minimization of the Riemannian curvature tensor, c.f. [Kass et al. 1988]. which distributes the norms of the gradients of $G^{\theta}(z)$. However, in high-dimensions, the curvature tensor is of dimension k^4, which is intractable. We address this issue by minimizing the gradients of $G^{\theta}(z)$, which also minimizes the Levi-Civita connection. The difference between the gradients of $G^{\theta}(z)$ and the Riemannian curvature tensor is that the Riemannian curvature tensor is parameterization-free while the gradients of $G^{\theta}(z)$ are not. For example, the gradients of $G^{\theta}(z)$ are scaled by s if we uniformly scale the parameter domain by $\frac{1}{s}$. However, this issue is addressed in AD, where the distribution of latent codes is forced to match N_k.

Another consequence of minimizing the gradients of $G^{\theta}(z)$ is that it prioritizes that $G^{\theta}(z)$ is a constant matrix locally, promoting the geodesic interpolation property.

Following the discussion trial, we define the shape space regularization term as

$$r_{\text{shape}}(\theta) = E_{z\sim N_k}\left(\sum_{i}\sum_{j=1}^{k}\left(G_{ij}^{\theta}(z)\right)^2 + \lambda_{g}\sum_{i=1}^{k}\left\|\frac{\partial G^{\theta}}{\partial z_i}(z)\right\|^2\right) \tag{18}$$

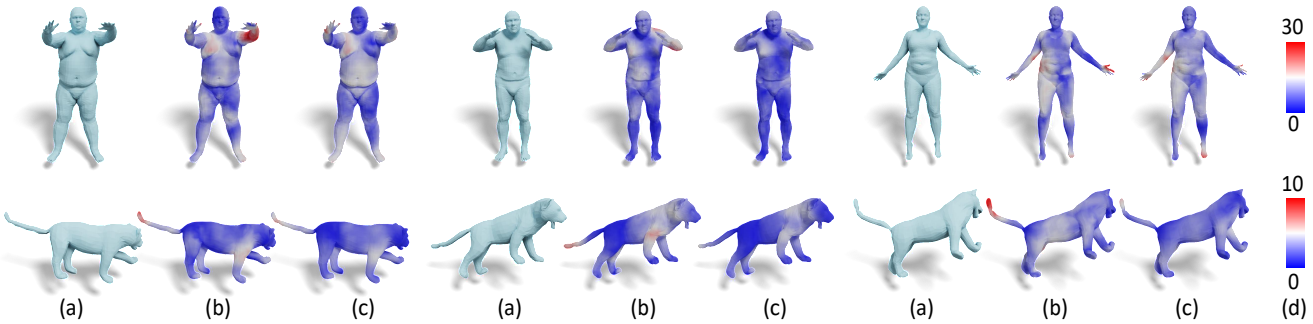


Fig. 6. Qualitative shape generation results of ARAPReg [Huang et al. 2021] and GeoLatent. We visualize the reconstruction errors of (a) sampled testing meshes. (b) ARAPReg. (c) GeoLatent. (d) Error bar for (top) Human and (bottom) Animal.

where $\square$ is a hyperparameter. We again apply the finite-difference approach described in (17) to compute the elements of $\square^{\square}(z)$ and their gradients.

The total regularization term combines $\square_{\text{latent}}$ defined in (13) and $\square_{\text{shape}}$ (18). Regarding the trade-off parameters, we set $\square_{\text{ps}} = 0.01$, $\square_{\square} = 0.01$, $\square_{\text{latent}} = 1$, and $\square_{\text{shape}} = 0.1$ in our experiments.

7 LEARNING GENERATIVE MODELS

This section describes the technical details to learning the generative model $g^{\square}(z)$ from a shape collection S. Following [Huang et al. 2021; Litany et al. 2018; Rakotosaona and Ovsjanikov 2020; Tan et al. 2018; Tretschk et al. 2020; Verma et al. 2018], we assume the input shapes are consistently meshed as the generator model $g^{\square}(z)$. With $\{s_{\square} \mid 1 \le \square \le \square\}$ we denote the mesh vertices of $\square_{\square}$. In this case, we then define the data loss of each training shape $\square_{\square}$ as

$$\square_{\text{AD}}(g^{\square}, S) = \frac{1}{\square} \sum_{\square=1}^{\square} \sum_{\square=1}^{\square} \square g^{\square}_{\square}(z_{\square}) - s_{\square}\,\square + \text{KL}(\{z_{\square}\}, N_{\square}). \quad (19)$$

where $\text{KL}(\{z_{\square}\}, N_{\square})$ measures the KL-divergence between the empirical distribution $\{z_{\square}\}$ of latent codes and $N_{\square}$.

Network training amounts to solve

$$\min_{\square} \square_{\text{AD}}(g^{\square}, S) + \square_{\text{latent}}\square_{\text{latent}}(\square) + \square_{\text{shape}}\square_{\text{shape}}(\square) \quad (20)$$

To solve (20), similar to [Huang et al. 2021], we first drop $\square_{\text{latent}}(\square)$ and $\square_{\text{shape}}(\square)$ in (20) to initialize $\square$. This step becomes the standard procedure of learning an AD for meshes. After that, we activate $\square_{\text{latent}}(\square)$ and $\square_{\text{shape}}(\square)$ and refine $\square$. Network training employs ADAM [Kingma and Ba 2015].

8 EXPERIMENTAL RESULTS

This section presents an experimental evaluation of GeoLatent. We begin with the experimental setup in Section 8.1. We then analyze the shape generation quality in Section 8.2. Section 8.3 performs a user study to evaluate the latent space design. Finally, Section 8.4 presents an ablation study on GeoLatent.

8.1 Experimental Setup

8.1.1 Datasets. We have tested our approach on two datasets, i.e., Human and Animal. The Human category is DFAUST [Bogo et al. 2017]. We use the registered SMPL model from the original DFAUST dataset. Since there is low variety between the adjacent shapes, we subsample 2000 meshes from the original dataset. Animal category has one dataset of 400 shapes [Huang et al. 2021], which is generated from SMAL [Zufi et al. 2017].

8.1.2 Evaluation protocols. We evaluate GeoLatent in two ways. Evaluations combine the reconstruction errors (RE) on testing shapes that do not participate in learning the shape generators and the Frechet inception distance (FID) score [Heusel et al. 2017] that is widely used in evaluating image generative models. The original definition of FID employs the second-to-last layer of a pre-trained image classifier inception v3 [Heusel et al. 2017]. To this end, we pre-train a mesh auto-encoder with graph neural network. The encoder inputs a generated mesh and outputs the meaningful latent representation. The encoder combines a feature extraction tower with four convolution layers and two fully connected linear layers. After training, we then use the output of the feature extraction tower to calculate the FID score.

Second, we evaluate GeoLatent through user studies to assess the effects of axis-aligned geodesic interpolation and multi-scale pose-shape disentanglement in shape exploration.

8.2 Analysis of Shape Generation

For baseline comparisons, we choose five state-of-the-art mesh-based generative models. The first three baseline techniques, i.e., 3DMM [Bouritsas et al. 2019], COMA [Ranjan et al. 2018], and MeshConv [Zhou et al. 2020b], focus on network architecture design. The remaining two baseline approaches, i.e., FrameAVE [Atzmon et al. 2022] and ARAPReg [Huang et al. 2021] focus on novel training approaches. We re-implement FrameAVE without using the skinning weight matrix for fair comparison. Note that the network architecture of GeoLatent follows that of ARAPReg, while the training loss is fundamentally different.

Table 1 and Table 2 present the quantitative results of GeoLatent and the baseline approaches with respect to reconstruction errors and FID scores. Figure 6 presents qualitative comparisons between GeoLatent and ARAPReg.

Quantitatively, GeoLatent outperforms all baseline approaches. In terms of RE, GeoLatent outperforms the top-performing baseline ARAPReg by 14.3% and 10.5% on Human and Animal, respectively. Regarding FID, GeoLatent outperforms ARAPReg by 7.0% and 4.3%

Trial 1. Evaluations of shape generation quality using mean reconstruction errors (mm). No-AIGC: without axis-aligned geodesic interpolation. No-PS-Dis: without pose-shape disentanglement. No-SSR: without shape space regularization.

	3DMM	COMA	MeshConv	FrameAVE	ARAPReg	GeoLatent	No-AIGC	No-PS-Dis.	No-SSR.
Human	26.39	20.06	18.24	15.75	14.13	12.11	13.49	12.38	13.72
Animal	17.78	14.52	8.01	12.45	6.68	5.98	6.42	6.12	6.37

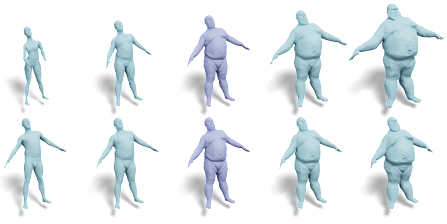
Table 2. Evaluations of shape generation quality using FID (Frechet inception distance).

	COMA	ARAPReg	GeoLatent
Human	8.41	8.02	7.46
Animal	13.59	13.24	12.67

(a)

	GeoL	ARAP	COMA
GeoL	X	82.1%	87.5%
ARAP	17.9%	X	72.3%
COMA	12.5%	27.7%	X

(b)



(c)

	GeoL	ARAP	COMA
GeoL	X	86.1%	91.6%
ARAP	13.9%	X	67.1%
COMA	8.4%	32.9%	X

(d)

Fig. 7. Qualitative evaluation of latent space design. GeoL: GeoLatent. ARAP: ARAPReg [Huang et al. 2021], COMA: [Ranjan et al. 2018]. (a-b) Evaluations on shape interpolation. (c-d) Evaluations on consistency of shape extrapolations for shape exploration. In (a), left is from ARAPReg and right is from GeoLatent. In (c), top is from ARAPReg and bottom is from GeoLatent. In (b) and (d), each element (e.g, row GeoL and column ARAP), shows the percentage that the corresponding row (e.g., GeoL) wins the corresponding column (e.g., ARAP).

on Human and Animal, respectively. This shows the effectiveness of GeoLatent.

Qualitatively, the improvements of GeoLatent from baseline approaches are around places where the underlying deformations are large, such as joint regions between different bones and torso regions in which deformations are highly non-isometric.

We can understand the improvements of GeoLatent from baseline approaches in three aspects. The first aspect is minimizing the metrics and the metric gradients, which reduces the curvature of the manifold. The second is axis-aligned geodesic interpolation and pose-shape disentanglement, which also impose regularizations on the shape generator. The third aspect is the combination of ARAP and ACAP deformation models. This approach is suitable for large inter-shape deformations.

8.3 Analysis of Latent Space Design

In this section, we present a user study on GeoLatent to assess the goals of latent space design. The study consists of two components.

The first component concerns shape interpolations in the latent space for shape morphing. Specifically, for each pair of shapes and each shape generative model, we use linear interpolation (the most straightforward approach) to generate a morphing sequence that transforms one shape into another. We then visualize the interpolations and ask users to compare between two methods visually. We compare GeoLatent with two top-performing baselines: ARAPReg and COMA. In total, we tested 200 randomly sampled shape pairs. For each pair of methods on each shape pair, we asked three users with experience in shape modeling but without knowledge of deep generative models to judge which shape interpolation is more meaningful, e.g., smooth and feature-preserving, which are desired goals of shape morphing. Figure 7(a) shows a sampled shape pair and the interpolation results of two methods.

Figure 7(b) shows the relative ratings among GeoLatent and the baseline techniques. We can see that GeoLatent strongly dominates the other baseline approaches, with 82.1% better than the top-performing baseline. These results show the importance of enforcing that shape interpolations follow geodesic curves (under ARAP and ACAP metrics) and optimizing the latent space so that linear interpolations are approximately geodesic curves.

The second component concerns pose-shape disentanglement. We ask the user to explore the underlying shape space, given a shape as a starting point. Again, we compare GeoLatent with ARAPReg and COMA. For fair comparisons, for each baseline, we solve a generalized eigendecomposition problem for each shape to obtain the corresponding eigenvectors and provide the users with the corresponding directions for exploration. We asked users to compare how consistent the deformed shapes under different magnitudes are and how natural the displacements are (See Figure 7(c) and Figure 8).

Figure 7(d) collects the statistics on 100 shapes. We can see that GeoLatent offers much more consistent extrapolations than baseline approaches, i.e., by 86.1% to the top-performing baseline. This is attributed to the axis-aligned geodesic interpolation property, which prioritizes that extrapolations are consistent. Moreover, GeoLatent is also superior to baseline approaches in terms of the naturalness of the synthetic shapes. We can understand this from the fact that GeoLatent employs shape regularization terms.

8.4 Ablation Study

Finally, we present an ablation study on GeoLatent. Table 1 describes the quantitative results.

8.4.1 Without axis-aligned geodesic interpolation. Without the axis-aligned geodesic interpolation term, the RE values increase by 11.4% and 7.4% on Human and Animal, respectively. These results show

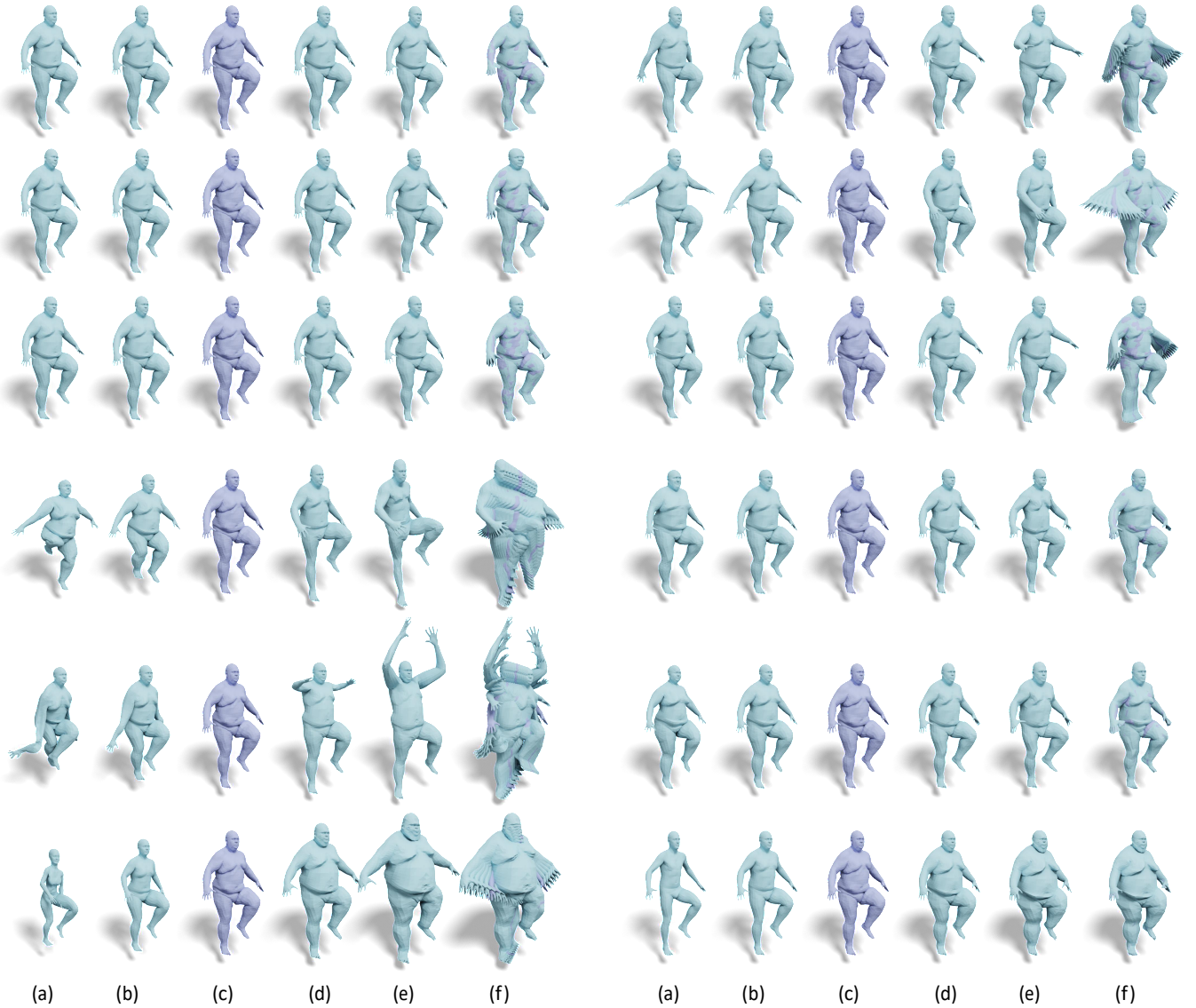


Fig. 8. Evaluations on consistency of shape extrapolations for shape exploration. Each row shows (a, b, d, e) the shape extrapolations along one eigenvector direction from (c) the base shape. The first three rows correspond to the smallest three eigenvalues. The last three rows correspond to the largest three eigenvalues. (f) Ten extrapolated shapes are shown. (Left) ARAPReg [Huang et al. 2021]. (Right) GeoLatent.

that this regularization objective not only optimizes the latent space but also offers implicit regularizations on the shape space.

8.4.2 Without pose-shape disentanglement. Without the pose-shape disentanglement term, the RE values increase by 2.2% and 2.3% on Human and Animal, respectively. These results show that similar to axis-aligned geodesic interpolation, pose-shape disentanglement also has some effects on regularizing the shape space. However, the regularization effects are less effective than the axis-aligned geodesic interpolation since the regularization term mainly aims to enhance the interpretability of the latent space.

8.4.3 Without shape space regularization. Without the shape space regularization term, the RE values increase by 13.3% and 6.5% on Human and Animal, respectively. These results show that minimizing the metrics and the metric gradients can prevent large metric values and promote metric smoothness, which helps to preserve geometric features and is critical for better generalization.

9 APPLICATIONS

We present two applications of GeoLatent in geodesic interpolation (Section 9.1) and geodesic shooting (Section 9.2).

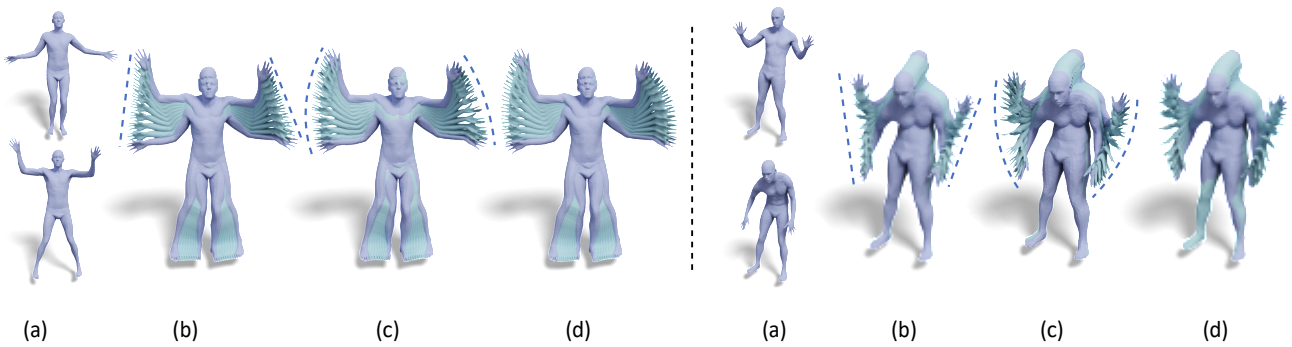


Fig. 9. (d) Geodesic interpolations between (a) two shapes that minimize (21), starting from linear interpolations in the latent space. We can see that (c) GeoLatent offers better solutions than (b) without GeoLatent.

9.1 Geodesic Interpolation

The first application is in geodesic interpolation. To this end, we generalize the formulation in [Shao et al. 2018] to the case of having a Riemannian metric of the latent space. Specifically, given a starting latent code z_0 and a target latent code z_1, our goal is to compute a series of intermediate codes $z_t, 0 \le t \le 1$ where $z_0 = z_0$ and $z_1 = z_1$ to minimize the following energy potential:

$$E(\{z_t\}) := \sum_{t=0}^{1} g^t(z_{t+1}) - g^t(z_t)^T G^t_{t+\frac{1}{2}} g^t(z_{t+1}) - g^t(z_t),$$

$$G^t_{t+\frac{1}{2}} := \frac{G^t(z_t) + G^t(z_{t+1})}{2}. \tag{21}$$

Starting from linear interpolation $z_t = z_0 + \frac{t}{1}(z_1 - z_0), 0 \le t \le 1$, we apply gradient descent to minimize $E(\{z_t\})$. Note that, we do not use the eficient solution in ([Shao et al. 2018]) that leverages the encoder, as our goal here is to compute an accurate discrete geodesic.

Figure 9 compares the final geodesic curve of using the linear interpolation when dropping our proposed regularization terms and the linear interpolation derived from GeoLatent. We can see that GeoLatent leads to interpolations that are very close to the optimized geodesic curve. In contrast, without our proposed regularization terms, the linear interpolations present various geometric distortions. The benefit of GeoLatent comes from avoiding costly geodesic curve optimization during testing time.

9.2 Geodesic Shooting

The second application is in geodesic shooting (GS), where the goal is to perform shape extrapolation in the tangent direction $v \in R^n$ at a given shape $z_{root} = g^n(z_{root})$ for a user specified length l_0. Specifically, GS aims to compute a series of latent codes $z_0 = z_{root}, \cdots, z_1$, so that 1) $\frac{v}{\|v\|} = \frac{z_1 - z_0}{\|z_1 - z_0\|}$, 2) $g^n(z_t), 0 \le t \le 1$ form a discrete geodesic curve, and 3) its length

$$E(\{z_t\}) = \sum_{t=0}^{1} \sqrt{g^n(z_{t+1}) - g^n(z_t)^T G^t_{t+\frac{1}{2}} g^n(z_{t+1}) - g^n(z_t)}$$

is equal to l_0.

We set $z_1 = z_{root} + tv$ and compute $z_t, t \ge 2$ in order until $l(\{z_t\}) \ge l_0$. At each iteration t, we compute z_{t+1} so that

$$z_t = \operatorname*{argmin}_z (z_t - z_{t-1})^T G^t(z_t)(z_t - z_{t-1}) + (z_t - z_{t+1})^T G^t(z_t)(z_t - z_{t+1})$$

This leads to the following quadratic system on z_{t+1}:

$$0 = 2G^t(z_t)(2z_t - z_{t-1} - z_{t+1}) + (z_t - z_{t-1})^T \frac{\partial G^t}{\partial z}(z_t) (z_t - z_{t-1}) + (z_t - z_{t+1})^T \frac{\partial G^t}{\partial z}(z_t)(z_t - z_{t+1}) \tag{22}$$

Starting from $z_{t+1} = 2z_t - z_{t-1}$, we apply Newton's method to solve (22).

We compare our approach with that of ([Shao et al. 2018]). For each approach, we fix z_0 and apply the approach in Section 9.1 to check how close the geodesic shooting result is to the discrete geodesic curve between z_{root} and z_1.

Figure 10 shows the geodesic shooting results. We can see that our results are superior compared to that of ([Shao et al. 2018]).

10 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

In this paper, we have introduced a differential geometry framework to deformable shape generators. The resulting approach called GeoLatent builds on a Riemannian metric on a deformable shape generator. We show how to enforce two desired properties of the latent space, i.e., axis-aligned geodesic interpolation and pose-shape disentanglement, as well as regularization terms that enforce smooth shape interpolations. Experimental results show that GeoLatent improves the interpretability of the latent space and the generalization behavior of the trained shape generator. The effectiveness of GeoLatent is demonstrated in two applications, i.e., geodesic interpolation and geodesic shooting.

One limitation of GeoLatent is that it does not utilize curvature-based regularizations used in the original active contour model. This is because the dimension of the curvature tensor is $O(n^4)$, which is not tractable for large n. One way to address this issue is to make the metric matrix sparse. However, the open question is whether

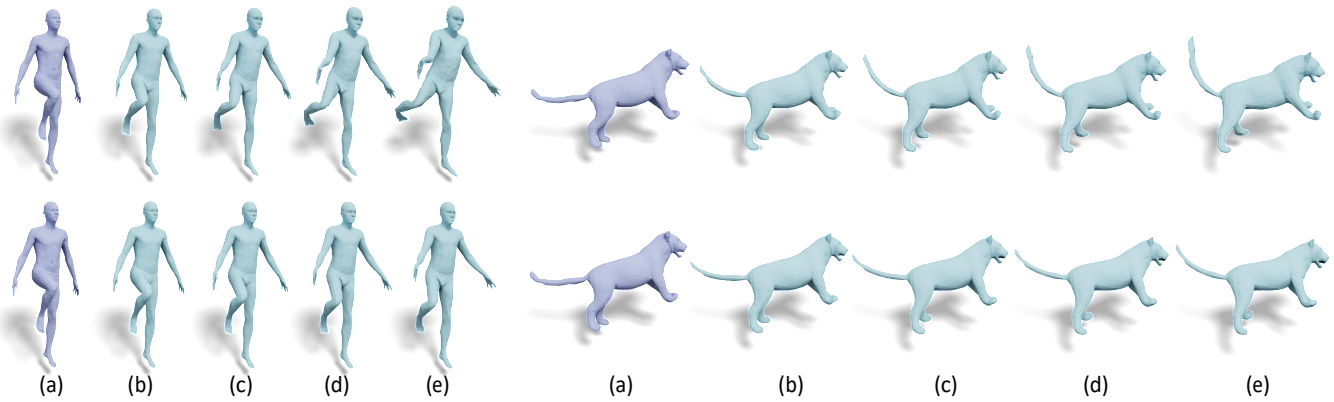


Fig. 10. Geodesic shooting results from shape (a). Compared to (top) [Shao et al. 2018], the extrapolated shapes (b, c, d, e) generated by (bottom) our approach are less distorted and more reasonable.

there exists a shape generator that fits the training data well and has a sparse metric matrix.

Another limitation of GeoLatent is that it requires pre-defined inter-shape correspondences as inputs. One potential approach to address this issue is to employ implicit deformable shape generators, e.g., SALD [Atzmon and Lipman 2021] and GenCorres [Yang et al. 2023]. In particular, GenCorres converts an implicit shape generator into an explicit generator locally, allowing us to define a Riemannian metric and apply the regularization terms of GeoLatent. The major issues are addressing the computational challenge of GenCorres in solving dense linear systems and developing an implicit shape representation that targets deformable shapes. We leave the extension of GeoLatent to implicit deformable shape generators as future work.

There are ample future research opportunities. So far, we have studied latent space design for deformable objects. A natural question is how to extend the idea to handle man-made shapes, e.g., by developing a suitable deformation model for man-made shapes. Another direction is to explore other deformable shape collections, such as faces and bones. We would also like to equip different axes of the latent space with semantic attributes, e.g., by developing a dictionary learning formulation in which a sparse set of axes of the latent space can describe each semantic variation.

ACKNOWLEDGMENTS

Qixing Trial would like to acknowledge NSF IIS-2047677, HDR-1934932, CCF-2019844 and the gift funding from Wormpex AI Research.

REFERENCES

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70), Doina Precup and Yee Whye Teh (Eds.). PMLR, Sydney, NSW, Australia, 214–223. http://proceedings.mlr.press/v70/arjovsky17a.html

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. 2018. Latent Space Oddity: on the Curvature of Deep Generative Models. In ICLR (Poster). OpenReview.net, Online, 15 pages.

Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. 2021. Geometrically Enriched Latent Spaces. In The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 130), Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, Online, 631–639. http://proceedings.mlr.press/v130/arvanitidis21a.html

Matan Atzmon and Yaron Lipman. 2021. SALD: Sign Agnostic Learning with Derivatives. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net. https://openreview.net/forum?id=7EDgLu9reQD

Matan Atzmon, Koki Nagano, Sanja Fidler, Sameh Khamis, and Yaron Lipman. 2022. Frame Averaging for Equivariant Shape Space Learning. In CVPR. IEEE, Washington,DC, USA, 621–631.

Matan Atzmon, David Novotný, Andrea Vedaldi, and Yaron Lipman. 2021. Augmenting Implicit Neural Shape Representations with Explicit Deformation Fields. arXiv:2108.08931 https://arxiv.org/abs/2108.08931

Tristan Aumentado-Armstrong, Stavros Tsogkas, Sven J. Dickinson, and Allan D. Jepson. 2021. Disentangling Geometric Deformation Spaces in Generative Latent Shape Models.

Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan D. Jepson, and Sven J. Dickinson. 2019. Geometric Disentanglement for Generative Latent Shape Models. In ICCV. IEEE, Washington, DC, USA, 8180–8189.

Elena Balashova, Vivek Singh, Jiangping Wang, Brian Teixeira, Terrence Chen, and Thomas A. Funkhouser. 2018. Structure-Aware Shape Synthesis. In 2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018. IEEE Computer Society, Washington, DC, USA, 140–149. https://doi.org/10.1109/3DV.2018.00026

Martin Bauer, Nicolas Charon, Philipp Harms, and Hsi-Wei Hsieh. 2021. A Numerical Framework for Elastic Surface Matching, Comparison, and Interpolation. Int. J. Comput. Vis. 129, 8 (2021), 2425–2444. https://doi.org/10.1007/s11263-021-01476-6

Martin Bauer, Philipp Harms, and Peter W. Michor. 2011. Sobolev metrics on shape space of surfaces. Journal of Geometric Mechanics 3, 4 (2011), 389–438. https://doi.org/10.3934/jgm.2011.3.389

Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2017. Dynamic FAUST: Registering Human Bodies in Motion. In CVPR. IEEE Computer Society, Washington, DC, USA, 5573–5582.

Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-Up: Shaping Discrete Geometry with Projections. Comput. Graph. Forum 31, 5 (2012), 1657–1667. https://doi.org/10.1111/j.1467-8659.2012.03171.x

Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Stefanos Zafeiriou, and Michael M. Bronstein. 2019. Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In ICCV. IEEE, Washington, DC, USA, 7212–7221.

Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. 2009. Numerical Geometry of Non-Rigid Shapes. Springer, New York, NY, USA.

Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. 2020. Learning gradient fields for shape generation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. Springer, Springer, New York, NY, USA, 364–381.

Nutan Chen, Francesco Ferroni, Alexej Klushyn, Alexandros Paraschos, Justin Bayer, and Patrick van der Smagt. 2019. Fast Approximate Geodesics for Deep Generative Models. In Artificial Neural Networks and Machine Learning - ICANN 2019: Deep Learning - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 11728), Igor V. Tetko, Vera Kurková, Pavel Karpov, and Fabian J. Theis (Eds.). Springer, New York, NY, USA, 554–566. https://doi.org/10.1007/978-3-030-30484-

3_45

Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick van der Smagt. 2018. Metrics for Deep Generative Models. In AISTATS (Proceedings of Machine Learning Research, Vol. 84). PMLR, Indio, CA, USA, 1540–1550.

Zhixiang Chen and Tae-Kyun Kim. 2021. Learning Feature Aggregation for Deep 3D Morphable Models. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, Washington, DC, USA, 13164–13173. https://doi.org/10.1109/CVPR46437.2021.01296

Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. 2020. LIMP: Learning Latent Shape Representations with Metric Preservation Priors. In ECCV (3) (Lecture Notes in Computer Science, Vol. 12348). Springer, New York, NY, USA, 19–35.

Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2011. Spin Transformations of Discrete Surfaces. ACM Trans. Graph. 30, 4, Article 104 (jul 2011), 10 pages. https://doi.org/10.1145/2010324.1964999

Simone Foti, Bongjin Koo, Danail Stoyanov, and Matthew J. Clarkson. 2023. 3D Generative Model Latent Disentanglement via Local Eigenprojection. https://doi.org/10.48550/ARXIV.2302.12798

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative Adversarial Networks. Commun. ACM 63, 11 (oct 2020), 139–144. https://doi.org/10.1145/3422622

Marc Habermann, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. 2020. DeepCap: Monocular Human Performance Capture Using Weak Supervision. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. IEEE, Washington, DC, USA, 5051–5062. https://doi.org/10.1109/CVPR42600.2020.00510

Emmanuel Hartman, Yashil Sukurdeep, Eric Klassen, Nicolas Charon, and Martin Bauer. 2023. Elastic Shape Analysis of Surfaces with Second-Order Sobolev Metrics: A Comprehensive Numerical Framework. Int. J. Comput. Vis. 131, 5 (2023), 1183–1209. https://doi.org/10.1007/s11263-022-01743-0

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). Curran Associates Inc., Red Hook, NY, USA, 6626–6637. https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html

Qixing Huang, Xiangru Huang, Bo Sun, Zaiwei Zhang, Junfeng Jiang, and Chandrajit Bajaj. 2021. ARAPReg: An As-Rigid-As Possible Regularization Loss for Learning Deformable Shape Generators. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Computer Vision Foundation / IEEE, Virtual, 5815–5825.

Qixing Huang, Martin Wicke, Bart Adams, and Leonidas J. Guibas. 2009. Shape Decomposition using Modal Analysis. Comput. Graph. Forum 28, 2 (2009), 407–416. http://dblp.uni-trier.de/db/journals/cgf/cgf28.html#HuangWAG09

Dimitrios Kalatzis, David Eklund, Georgios Arvanitidis, and Søren Hauberg. 2020. Variational Autoencoders with Riemannian Brownian Motion Priors. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119). PMLR, Online, 5053–5066. http://proceedings.mlr.press/v119/kalatzis20a.html

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-End Recovery of Human Shape and Pose. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, Washington, DC, USA, 7122–7131. https://doi.org/10.1109/CVPR.2018.00744

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In CVPR. Computer Vision Foundation / IEEE, Washington, DC, USA, 4401–4410.

Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. 1988. Snakes: Active contour models. Int. J. Comput. Vis. 1, 4 (1988), 321–331.

Martin Kilian, Niloy J. Mitra, and Helmut Pottmann. 2007. Geometric Modeling in Shape Space. ACM Trans. Graph. 26, 3 (July 2007), 64–es. https://doi.org/10.1145/1276377.1276457

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Yoshua Bengio and Yann LeCun (Eds.). http://OpenReview.net, Online, 1–11. http://arxiv.org/abs/1412.6980

Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. 2019. Convolutional Mesh Regression for Single-Image Human Shape Reconstruction. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, Washington, DC, USA, 4501–4510. https://doi.org/10.1109/CVPR.2019.00463

Line Kühnel, Tom Fletcher, Sarang C. Joshi, and Stefan Sommer. 2018. Latent Space Non-Linear Statistics. arXiv:1805.07632 http://arxiv.org/abs/1805.07632

Jake Levinson, Avneesh Sud, and Ameesh Makadia. 2019. Latent feature disentanglement for 3D meshes.

Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao (Richard) Zhang, and Leonidas J. Guibas. 2017. GRASS: generative recursive autoencoders for shape structures. ACM Trans. Graph. 36, 4 (2017), 52:1–52:14.

Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. 2020. Online Adaptation for Consistent Mesh Reconstruction in the Wild. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). Curran Associates Inc., Red Hook, NY, USA, 10 pages. https://proceedings.neurips.cc/paper/2020/hash/aba3b6fd5d186d28e06ff97135cade7f-Abstract.html

Or Litany, Alexander M. Bronstein, Michael M. Bronstein, and Ameesh Makadia. 2018. Deformable Shape Completion With Graph Convolutional Autoencoders. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, Salt Lake City, UT, USA, 1886–1895. https://doi.org/10.1109/CVPR.2018.00202

Hsueh-Ti Derek Liu and Alec Jacobson. 2021. Normal-Driven Spherical Shape Analogies. Comput. Graph. Forum 40, 5 (2021), 45–55. https://doi.org/10.1111/cgf.14356

Shitong Luo and Wei Hu. 2021. Diffusion probabilistic models for 3d point cloud generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Washington, DC, USA, 2837–2845.

A. C.G. Mennucci, A. Yezzi, and G. Sundaramoorthi. 2008. Properties of Sobolev-type metrics in the space of curves. Interfaces and Free Boundaries 10, 4 (2008), 423–445. https://doi.org/10.4171/IFB/196

Peter Michor and David Mumford. 2005. Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. Documenta Mathematica 10 (2005), 217–245. http://eudml.org/doc/125727

Peter W. Michor and David Mumford. 2007. An overview of the Riemannian metrics on spaces of curves using the Hamiltonian approach. Applied and Computational Harmonic Analysis 23, 1 (2007), 74–113.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. 2019. StructureNet: hierarchical graph networks for 3D shape generation. ACM Trans. Graph. 38, 6 (2019), 242:1–242:19.

Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan L. Yuille, Nuno Vasconcelos, and Xiaolong Wang. 2021. A-SDF: Learning Disentangled Signed Distance Functions for Articulated Shape Representation. In ICCV. IEEE, Washington, DC, USA, 12981–12991.

Sanjeev Muralikrishnan, Siddhartha Chaudhuri, Noam Aigerman, Vladimir G. Kim, Matthew Fisher, and Niloy J. Mitra. 2022. GLASS: Geometric Latent Augmentation for Shape Spaces. In CVPR. IEEE, Washington, DC, USA, 470–479.

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In ICCV. IEEE, Washington, DC, USA, 7587–7596.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE(CVF), Washington, DC, USA, 165–174.

William S. Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. 2020. The Hessian Penalty: A Weak Prior for Unsupervised Disentanglement. In Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 12351), Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, New York, NY, USA, 581–597. https://doi.org/10.1007/978-3-030-58539-6_35

Peter Petersen. 2006. Riemannian Geometry. Springer Science & Business Media, New York, NY, USA.

Henning Petzka, Ted Kronvall, and Cristian Sminchisescu. 2022. Discriminating Against Unrealistic Interpolations in Generative Adversarial Networks. https://doi.org/10.48550/arXiv.2203.01035 arXiv:2203.01035

Helmut Pottmann, Qixing Huang, Bailin Deng, Alexander Schiftner, Martin Kilian, Leonidas Guibas, and Johannes Wallner. 2010. Geodesic Patterns. ACM Trans. Graph. 29, 4, Article 43 (jul 2010), 10 pages. https://doi.org/10.1145/1778765.1778780

Marie-Julie Rakotosaona and Maks Ovsjanikov. 2020. Intrinsic Point Cloud Interpolation via Dual Latent Space Navigation. In Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12347), Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, Washington, DC, USA, 655–672. https://doi.org/10.1007/978-3-030-58536-5_39

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. 2018. Generating 3D faces using convolutional mesh autoencoders. In Proceedings of the European Conference on Computer Vision (ECCV). Springer, New York, NY, USA, 704–720.

Hrittik Roy and Soren Hauberg. 2022. Optimal Latent Transport. In Symmetry and Geometry in Neural Representations. OpenReview.net, Online, 10 pages.

Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. 2018. The Riemannian Geometry of Deep Generative Models. In 2018 IEEE Conference on Computer Vision and Pattern

Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, Washington, DC, USA, 315–323. https://doi.org/10.1109/CVPRW.2018.00071

Olga Sorkine and Marc Alexa. 2007. As-Rigid-as-Possible Surface Modeling. In Proceedings of the Fifth Eurographics Symposium on Geometry Processing (Barcelona, Spain) (SGP '07). Eurographics Association, Goslar, DEU, 109–116.

Jakob Stolberg-Larsen and Stefan Sommer. 2022. Atlas Generative Models and Geodesic Interpolation. Image Vision Comput. 122, C (jun 2022), 8 pages. https://doi.org/10.1016/j.imavis.2022.104433

Zhe Su, Martin Bauer, Stephen C. Preston, Hamid Laga, and Eric Klassen. 2020. Shape Analysis of Surfaces Using General Elastic Metrics. J. Math. Imaging Vis. 62, 8 (2020), 1087–1106. https://doi.org/10.1007/s10851-020-00959-4

Robert W. Sumner, Johannes Schmid, and Mark Pauly. 2007. Embedded Deformation for Shape Manipulation. ACM Trans. Graph. 26, 3 (jul 2007), 80–es. https://doi.org/10.1145/1276377.1276478

Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. 2018. Variational Autoencoders for Deforming 3D Mesh Models. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, Salt Lake City, UT, USA, 5841–5850. https://doi.org/10.1109/CVPR.2018.00612

N. Joseph Tatro, Stefan C. Schonsheck, and Rongjie Lai. 2020. Unsupervised Geometric Disentanglement for Surfaces via CFAN-VAE.

T

Edgar Tretschk, Ayush ewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. 2020. DEMEA: Deep Mesh Autoencoders for Non-rigidly Deforming Objects. In Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV (Lecture Notes in Computer Science, Vol. 12349), Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, Glasgow,UK, 601–617. https://doi.org/10.1007/978-3-030-58548-8_35

Amir Vaxman, Christian Müller, and Ofir Weber. 2015. Conformal mesh deformations with Möbius transformations. ACM Trans. Graph. 34, 4 (2015), 55:1–55:11. https://doi.org/10.1145/2766915

Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. 2019. Weaving Geodesic Foliations. ACM Trans. Graph. 38, 4, Article 34 (jul 2019), 22 pages. https://doi.org/10.1145/3306346.3323043

Nitika Verma, Edmond Boyer, and Jakob Verbeek. 2018. FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, Salt Lake City, UT, USA, 2598–2606. https://doi.org/10.1109/CVPR.2018.00275

Michael Wand, Philipp Jenke, Qi-Xing Huang, Martin Bokeloh, Leonidas J. Guibas, and Andreas Schilling. 2007. Reconstruction of deforming geometry from time-varying point clouds. In Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, 2007 (ACM International Conference Proceeding Series, Vol. 257), Alexander G. Belyaev and Michael Garland (Eds.). Eurographics Association, Avenue de Frontenex 32, 1207 Geneve, Switzerland, 49–58. https://doi.org/10.2312/SGP/SGP07/049-058

Binxu Wang and Carlos R. Ponce. 2021. A Geometric Analysis of Deep Generative Image Models and Its Applications. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, Online, 11 pages. https://openreview.net/forum?id=GH7QRzUDdXG

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI. Springer, New York, NY, USA, 55–71. https://doi.org/10.1007/978-3-030-01252-6_4

T

Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. enenbaum, Antonio Torralba, and William T. Freeman. 2018. 3D Interpreter Networks for Viewer-Centered Wireframe Modeling. Int. J. Comput. Vis. 126, 9 (2018), 1009–1026. https://doi.org/10.1007/s11263-018-1074-6

Weiwei Xu, Kun Zhou, Yizhou Yu, Qifeng Tan, Qunsheng Peng, and Baining Guo. 2007. Gradient Domain Editing of Deforming Mesh Sequences. In ACM SIGGRAPH 2007 Papers (San Diego, California) (SIGGRAPH '07). Association for Computing Machinery, New York, NY, USA, 84–es. https://doi.org/10.1145/1275808.1276482

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. In Proceedings of the IEEE/CVF international conference on computer vision. IEEE, Washington, DC, USA, 4541–4550.

Haitao Yang, Xiangru Huang, Bo Sun, Chandrajit Bajaj, and Qixing Huang. 2023. GenCorres: Consistent Shape Matching via Coupled Implicit-Explicit Shape Generative Models. arXiv:2304.10523 [cs.CV]

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. 2022. DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation. ACM Trans. Graph. 42, 1, Article 1 (aug 2022), 17 pages. https://doi.org/10.1145/3526212

Tao Yang, Georgios Arvanitidis, Dongmei Fu, Xiaogang Li, and Søren Hauberg. 2018. Geodesic Clustering in Deep Generative Models. arXiv:1809.04747 http://arxiv.org/abs/1809.04747

Yusuke Yoshiyasu, Wan-Chun Ma, Eiichi Yoshida, and Fumio Kanehiro. 2014. As-Conformal-as-Possible Surface Registration. In Proceedings of the Symposium on Geometry Processing (Cardiff, United Kingdom) (SGP '14). Eurographics Association, Goslar, DEU, 257–267. https://doi.org/10.1111/cgf.12451

Yu-Jie Yuan, Yu-Kun Lai, Jie Yang, Qi Duan, Hongbo Fu, and Lin Gao. 2020. Mesh Variational Autoencoders with Edge Contraction Pooling. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020. Computer Vision Foundation / IEEE, Washington, DC, USA, 1105–1112. https://doi.org/10.1109/CVPRW50498.2020.00145

Amir Zadeh, Yao-Chong Lim, Paul Pu Liang, and Louis-Philippe Morency. 2019. Variational Auto-Decoder.

Yexun Zhang, Ya Zhang, and Wenbin Cai. 2018. Separating Style and Content for Generalized Style Transfer. In CVPR. Computer Vision Foundation / IEEE Computer Society, Washington, DC, USA, 8447–8455.

Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. 2020a. Unsupervised Shape and Pose Disentanglement for 3D Meshes. In ECCV (22) (Lecture Notes in Computer Science, Vol. 12367). Springer, New York, NY, USA, 341–357.

Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. 2020b. Fully Convolutional Mesh Autoencoder Using Efficient Spatially Varying Kernels. In Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 776, 12 pages.

Silvia Zufi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 2017. 3D menagerie: Modeling the 3D shape and pose of animals. In Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Washington, DC, USA, 6365–6373.

A PROOF OF THEOREM 1

The proof combines the following two lemmas.

Lemma 2. If linear interpolations follow geodesic curves, then the Christoffel symbols satisfy the following constraints:

$$\Gamma^{i,j}_{k}(z) = 0, \qquad \forall\, i \neq j,\ j \neq k,\ i \neq k,\ i,j,k \in [d] \tag{23}$$

$$2\Gamma^{i,i}_{j}(z) = \Gamma^{j,i}_{i}(z), \qquad \forall\, i \neq j,\ i,j \in [d] \tag{24}$$

where $[d] = \{1, \cdots, d\}$.

Lemma 3. When (23) and (24) are satisfied, then $g^{ij}(z)$ is a constant matrix.

A.1 Proof of Lemma 2

Consider the interpolation between z and z + v as $g^{ij}(z(t))$ where $z(t) = z + t(t)v$. It is under arc-length parameterization if

$$\Big(\frac{d}{dt}z(t)\Big)^2 v^i \partial_i g^{jk}(z(t)) v_j v_k = 1. \tag{25}$$

Under the arc-length parameterization, $g^{ij}(z(t))$ is a geodesic iff

$$\frac{d^2}{dt^2}z_k + \sum_{1\le i,j\le d} \Gamma^{i,j}_{k}(z(t))\frac{d}{dt}z_i \frac{d}{dt}z_j = 0. \tag{26}$$

Computing the derivatives of (25) with respect to t, we have

$$2\Big(\frac{d^2}{dt^2}z(t)\Big)^i v^j \partial_i g^{jk}(z(t))v_k + \Big(\frac{d}{dt}z(t)\Big)^2 v^i v^j \Big\langle \frac{\partial}{\partial z}\partial_i g^{jk}(z(t)), v\Big\rangle v_k = 0 \tag{27}$$

Combing (27) and (25), we have

$$\frac{d^2}{dt^2}z(t) = -\frac{1}{2}\Big(\frac{d}{dt}z(t)\Big)^4 v^i v^j \Big\langle \frac{\partial}{\partial z}\partial_i g^{jk}(z(t)), v\Big\rangle v \tag{28}$$

Substituting (28) into (26), we have

$$\frac{1}{2}\Big(\frac{d}{dt}z(t)\Big)^2 v^i \Big\langle \frac{\partial}{\partial z}\partial_i g^{jk}(z(t)), v\Big\rangle v_j = \sum_{1\le i,j,k\le d} \Gamma^{i,j}_{k}(z(t))v_i v_j \tag{29}$$

Combing (25) and (29), we have

$$v^{\top}\langle \frac{\partial}{\partial z}(z(\theta)), v\rangle v_{\theta} = 2 \sum_{1\le i,j\le d} \Gamma^{i,j}_{\theta}(z(\theta))\theta_i\theta_j \cdot v^{\top}\theta(z(\theta))v \quad (30)$$

We first show that (30) is true when (23) and (24) are satisfied. In this case, we have

$$\sum_{1\le i,j\le d} \Gamma^{i,j}_{\theta}(z(\theta))\theta_i\theta_j = \Gamma^{i,j}_{\theta}(z(\theta))^2\theta_i + 2 \sum_{i\neq j} \Gamma^{i,j}_{\theta}\theta_i\theta_j = (\Gamma^{i,j}_{\theta}\theta_i)^2_{1\le i\le d} \quad (31)$$

Substituting (31) into (30), we need to show that $\forall z, v$,

$$\sum_{1\le i,j,k\le d} \frac{\partial}{\partial z_k}(z)\theta_i\theta_j\theta_k = 2 \sum_{1\le i,j\le d} \Gamma^{i,j}_{\theta}(z)\theta_i\theta_j$$

or equivalently

$$\frac{\partial}{\partial z_k}(z) = 2\Gamma^{k,j}_{\theta}(z)\theta_j(z), \quad \forall k \in [d] \quad (32)$$

$$\frac{\partial}{\partial z_k}(z) + 2\frac{\partial}{\partial z_j}(z) = 2\Gamma^{j,j}_{\theta}(z)\theta_j(z) + 4\Gamma^{k,j}_{\theta}(z)\theta_j(z), \quad \forall k \neq j \in [d] \quad (33)$$

$$\frac{\partial}{\partial z_i}(z) + \frac{\partial}{\partial z_j}(z) + \frac{\partial}{\partial z_k}(z) = 2\Gamma^{i,j}_{\theta}(z)\theta_j(z) + \Gamma^{j,k}_{\theta}(z)\theta_k(z) + \Gamma^{i,k}_{\theta}(z)\theta_k(z), \quad \forall i \neq j, j \neq k \in [d] \quad (34)$$

By the definition of $\Gamma^{i,j}$, we have

$$\sum_{k=1}^{d} \theta_k(z(\theta))\Gamma^{k,i}_{\theta}(z(\theta)) = \frac{1}{2}\left(\frac{\partial}{\partial z_i}(z(\theta)) + \frac{\partial}{\partial z_j}(z(\theta)) - \frac{\partial}{\partial z_k}(z(\theta))\right) \quad (35)$$

Substituting (23) and (24) into (35) and setting $i = j$, we have

$$\theta_k(z(\theta))\Gamma^{k,i}_{\theta}(z(\theta)) = \frac{\partial}{\partial z_i}(z(\theta)) - \frac{1}{2}\frac{\partial}{\partial z_k}(z(\theta)) \quad (36)$$

When setting $i = j$, we have that (32) is satisfied. When setting $i \neq j$, we have

$$\begin{aligned}
&\frac{\partial}{\partial z_i}(z(\theta)) + \frac{\partial}{\partial z_j}(z(\theta)) - \frac{\partial}{\partial z_k}(z(\theta)) \\
&= 2\theta_j(z(\theta))\Gamma^{i,j}_{\theta}(z(\theta)) + 2\theta_i(z(\theta))\Gamma^{j,i}_{\theta}(z(\theta)) \\
&= 2\theta_j(z(\theta))\Gamma^{i,j}_{\theta}(z(\theta)) + 2\theta_i(z(\theta))\Gamma^{i,j}_{\theta}(z(\theta))
\end{aligned} \quad (37)$$

Setting $i = j$, we have

$$\frac{\partial}{\partial z_k}(z(\theta)) = 2\theta_j(z(\theta))\Gamma^{k,j}_{\theta}(z(\theta)) + 2\theta_k(z(\theta))\Gamma^{j,k}_{\theta}(z(\theta)) \quad (38)$$

Setting $i = j$ in (36) and combine (38), we obtain (33).
Switching i, j, k in (37) and add the three resulting equations,

Now suppose (30) is true. Let $v = e_i$. When $i \neq j$, we have

$$\Gamma^{i,j}_{\theta}(z(\theta)) = 0. \quad (39)$$

Now fix $i \neq j$. Let $\theta_k = 0, \forall k \neq i, k \neq j$. Then (30) leads to

$$\begin{aligned}
&\frac{\partial^2}{\partial z_i^2}(z(\theta))\theta_i^3 + \frac{\partial^2}{\partial z_i\partial z_j}(z(\theta))\theta_i^3 + \frac{\partial^2}{\partial z_j^2}(z(\theta))\theta_i^2\theta_j + \frac{\partial^2}{\partial z_i\partial z_j}(z(\theta))\theta_i^2\theta_j + 2\frac{\partial^2}{\partial z_j^2}(z(\theta))\theta_i\theta_j^2 \\
&= 2\Gamma^{i,j}_{\theta}(z(\theta))\theta_i^2 + \Gamma^{j,j}_{\theta}(z(\theta))\theta_i^2 + 2\Gamma^{i,j}_{\theta}(z(\theta))\theta_i\theta_j \\
&\cdot \frac{\partial^2}{\partial z_i^2}(z(\theta))\theta_i^2 + \frac{\partial^2}{\partial z_i\partial z_j}(z(\theta))\theta_i^2 + \frac{\partial^2}{\partial z_j^2}(z(\theta))\theta_i\theta_j
\end{aligned} \quad (40)$$

Setting $i \neq j, j \neq k$ in (40), we have

$$\Gamma^{i,j}_{\theta}(z(\theta))\theta_i^2 + \Gamma^{j,j}_{\theta}(z(\theta))\theta_i^2 + 2\Gamma^{i,j}_{\theta}(z(\theta))\theta_i\theta_j = 0. \quad (41)$$

As (41) is true for any θ_i and θ_j, we have $\Gamma^{i,j}_{\theta} = 0$, which means (23) is satisfied. Setting $i = j$ in (40) and comparing the coeficients in front of θ_i^4 and $\theta_i^3\theta_j$, we have

$$\frac{\partial^2}{\partial z_i^2}(z(\theta)) = 2\Gamma^{i,j}_{\theta}(z(\theta))\theta_j(z(\theta)) \quad (42)$$

$$\begin{aligned}
\frac{\partial^2}{\partial z_i\partial z_j}(z(\theta)) &= 4\Gamma^{i,j}_{\theta}(z(\theta))\theta_j(z(\theta)) + 4\Gamma^{j,j}_{\theta}(z(\theta))\theta_j(z(\theta)) \\
&= 4\Gamma^{i,j}_{\theta}(z(\theta))\theta_j(z(\theta))
\end{aligned} \quad (43)$$

Combing (42) and (43), we have that (24) is satisfied, which ends the proof. □

A.2 Proof of Lemma 3

For simplicity, we denote

$$\theta^{-1}_{\theta}(z(\theta)) = \theta_{\theta}(\theta^{-1}(z(\theta))) = (\theta^{-1}(z(\theta)))^{-1})_{\theta}, \quad \Gamma^{i,j}_{\theta}(z(\theta)) = \Gamma^{i,j}_{\theta}.$$

Note that

$$\sum_{k=1}^{d} \theta_k\Gamma^{k,i}_{\theta} = \frac{1}{2}\left(\frac{\partial}{\partial z_i}\theta_j + \frac{\partial}{\partial z_j}\theta_i - \frac{\partial}{\partial z_k}\theta_k\right). \quad (44)$$

Since $\Gamma^{i,j}_{\theta} = 0$ when $i \neq j \neq k$ and $2\Gamma^{i,j}_{\theta} = \Gamma^{j,j}_{\theta}$ when $i \neq j$. We have $\forall i \neq j$, (44) leads to

$$\theta_i\Gamma^{i,i}_{\theta} + \theta_j\Gamma^{j,i}_{\theta} = \frac{1}{2}\left(\frac{\partial}{\partial z_i}\theta_i + \frac{\partial}{\partial z_j}\theta_j - \frac{\partial}{\partial z_j}\theta_i\right). \quad (45)$$

When $i = j$, (46) leads to

$$\theta_i\Gamma^{i,i}_{\theta} = \frac{1}{2}\frac{\partial}{\partial z_i}\theta_i. \quad (46)$$

Let $\theta = \theta$, we have

$$\Gamma^{i,i}_{\theta} = \frac{1}{2}\theta_i^{-1} \cdot \frac{\partial}{\partial z_i}\theta_i. \quad (47)$$

Substituting (47) to (46), we have when $i \neq j$,

$$\frac{\partial}{\partial z_j}\theta_i = \frac{\partial}{\partial z_i}\theta_j$$

we obtain (34). This ends the proof of (30).

$$\frac{\partial}{\partial} 2 \quad - \quad = \cdot \quad . \tag{4}$$

8) Substituting (47) to (45), we have when $\neq$,

$$\frac{-1}{} + \frac{1}{} = 2 \quad + \quad - \quad . \tag{49}$$

Let $\theta = \phi$ and $\theta = \phi$ in (49), we obtain the following two constraints

$$\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) + \frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) = 2\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\right) \tag{50}$$

$$\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) + \frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) = 2\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\right) \tag{51}$$

Combing (48) and (50), we have

$$\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) - 2\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\right). \tag{52}$$

Similarly, we have the following inequality when combing (48) and (51):

$$\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\frac{\partial \theta}{\partial \theta}\right) - 2\frac{\partial}{\partial \theta}\left(\frac{1}{\sin\theta}\right). \tag{53}$$

The following Lemma, which will be proved in Section A.3, shows that by combing (49), (52) and (53) we have

$$\frac{\partial^2}{\partial \theta^2}\left(\frac{\partial \theta}{\partial \theta}\right) = 0, \quad \forall \theta \in [\theta]. \tag{54}$$

Lemma 4. Suppose three 2D functions $\theta(\theta,\theta), \theta(\theta,\theta), \theta(\theta,\theta)$ satisfy the following partial differential equations:

$$\frac{\partial \theta}{\partial \theta} = \theta\left(2\frac{\partial \theta}{\partial \theta} - \theta\right), \tag{55}$$

$$\frac{\partial \theta}{\partial \theta} = \theta\left(2\frac{\partial \theta}{\partial \theta} - \theta\right), \tag{56}$$

$$\frac{\partial \theta}{\partial \theta} = \theta\left(3\frac{\partial \theta}{\partial \theta} - 2\theta\right), \tag{57}$$

$$\frac{\partial \theta}{\partial \theta} = \theta\left(3\frac{\partial \theta}{\partial \theta} - 2\theta\right), \tag{58}$$

Then θ, θ and θ are constant functions.

Substituting (54) into (48), we have

$$\frac{\partial}{\partial \theta}\left(\frac{\partial \theta}{\partial \theta}\right) + \frac{\partial}{\partial \theta}\left(\frac{\partial \theta}{\partial \theta}\right) - \frac{\partial}{\partial \theta}\left(\frac{\partial \theta}{\partial \theta}\right) = 0, \quad \forall \theta, \theta, \theta \in [\theta]. \tag{59}$$

It follows that

$$\frac{\partial \theta}{\partial \theta} = 0, \quad \forall \theta, \theta, \theta \in [\theta].$$

□

A.3 Proof of Lemma 4

Our goal is to prove that θ, θ, and θ are constant functions within a small arbitrary domain $\Omega \subset [0, 2\pi] \times [0, 2\pi]$ (We can always shift Ω, so that it resides in $[0, 2\pi] \times [0, 2\pi]$). Without losing generality, we can assume that θ, θ, and θ are real-value periodic functions on $[0, 2\pi] \times [0, , 2\pi]$, i.e., $\forall \theta, \theta \in \mathbb{N}$

$$\theta(\theta + 2\pi, \theta + 2\pi) = \theta(\theta, \theta), \quad \forall \theta \in \{\theta, \theta, \theta\}. \tag{60}$$

Otherwise, we can fix the values of θ, θ, and θ in Ω, and extend them smoothly inside $[0, 2\pi] \times [0, 2\pi]$ so that they are periodic.

Under the assumption of (60), 2D Fourier series give rise to the following decomposition

$$\theta(\theta, \theta) = \sum_{\theta=0}^{\infty}\sum_{\theta=0}^{\infty} \cos(\theta\theta)\left(\theta^1_{\theta,\theta}\cos(\theta\theta) + \theta^2_{\theta,\theta}\sin(\theta\theta)\right) + \sin(\theta\theta)\left(\theta^3_{\theta,\theta}\cos(\theta\theta) + \theta^4_{\theta,\theta}\sin(\theta\theta)\right), \quad \forall \theta \in \{\theta, \theta, \theta\} \tag{61}$$

As $\sin(0\theta) = 0$, we assume

$$\theta^2_{\theta,0} = \theta^3_{0,\theta} = \theta^4_{\theta,0} = \theta^4_{0,\theta} = 0, \quad \forall 0 \le \theta, \theta, \ \forall \theta \in \{\theta, \theta, \theta\}. \tag{62}$$

In addition, we assume

$$\theta^1_{0,0} \neq 0, \quad \forall \theta \in \{\theta, \theta, \theta\}. \tag{63}$$

Otherwise, we can always shift θ, θ, and θ by the same constants, so that (63) is true.

The derivatives of (61) are given by

$$\frac{\partial \theta}{\partial \theta}(\theta, \theta) = \sum_{\theta=0}^{\infty}\sum_{\theta=0}^{\infty} -\sin(\theta\theta)\theta\theta^1_{\theta,\theta}\cos(\theta\theta) + \theta^2_{\theta,\theta}\sin(\theta\theta) + \cos(\theta\theta)\theta\theta^3_{\theta,\theta}\cos(\theta\theta) + \theta^4_{\theta,\theta}\sin(\theta\theta), \quad \forall \theta \in \{\theta, \theta, \theta\} \tag{64}$$

$$\frac{\partial \theta(\theta, \theta)}{\partial \theta} = \sum_{\theta=0}^{\infty}\sum_{\theta=0}^{\infty} -\sin(\theta\theta)\theta\theta^1_{\theta,\theta}\cos(\theta\theta) + \theta^2_{\theta,\theta}\sin(\theta\theta) + \cos(\theta\theta)\theta\theta^3_{\theta,\theta}\cos(\theta\theta) + \theta^4_{\theta,\theta}\sin(\theta\theta), \quad \forall \theta \in \{\theta, \theta, \theta\} \tag{65}$$

Our goal is to show that

$$\theta^1_{\theta,\theta} \neq 0, \quad \text{iff} \quad \theta, \theta = 1, \theta = \theta = 0. \tag{66}$$

To this end, we use (61),(64), and (65) to express each constraint from (55) to (58). Consider representing them in the general form $\theta(\theta, \theta) = 0$ where

$$\theta(\theta, \theta) = \sum_{\theta,\theta,\theta,\theta,\ 0\le\theta_1,\theta_2,\theta_3,\theta_4\le 1} \theta^{8\theta_1+4\theta_2+2\theta_3+\theta_4}(1-\theta^1)^{\cos(\theta\theta)}\theta^{\theta,\theta,\theta,\theta}\cdot \theta_1\sin(\theta\theta)^{(1-\theta_2)\cos(\theta'\theta)+\theta_2\sin(\theta'\theta)^{(1-\theta_3)\cos(\theta\theta)}} \cdot \theta_3\sin(\theta\theta)^{(1-\theta_4)\cos(\theta'\theta)+\theta_4\sin(\theta'\theta)}. \tag{67}$$

The following proposition characterizes a necessary and suficient condition on $\theta(\theta, \theta) = 0$.

Proposition 6. Suppose $\theta(\theta, \theta)$ admits the expression of (67). If $\theta(\theta, \theta) = 0$. Then $\forall \theta, \theta, \theta', \theta'$,

$$\theta^{\theta_1,\theta_2,\theta_3,\theta_4}_{\theta,\theta,\theta,\theta}\theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta} + \theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta}\theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta}\theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta}\theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta} = 0 \tag{68}$$

where

$$(\theta_1, \theta_2, \theta_3, \theta_4) \in \{(1,1,1,1), (2,2,3,3), (3,3,2,2), (4,4,4,4), (5,9,5,9), (6,10,7,11), (7,11,6,10), (8,12,8,12), (9,5,9,5), (10,6,11,7), (11,7,10,6), (12,8,12,8), (13,13,13,13), (14,14,15,15), (15,15,14,14), (16,16,16,16)\}$$

Proof. See Section A.3.1. □

Let $\theta^{\theta,\theta,\theta,\theta}_{\theta,\theta,\theta,\theta}$, $1 \le \theta \le 16$ be the coeficients in (67) with respect to

$$\theta^{\theta\theta}_{\theta\theta} = \theta\left(2\frac{\partial \theta}{\partial \theta} - \theta\right).$$

Then the coeficients $\alpha^{1,\cdot}_{\cdot,\cdot,\cdot,\cdot}$, $1 \le \cdot \le 16$ are given by $1 \le \cdot \le 4$

$$\alpha^{1,4}_{3,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} + \beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{69}$$

$$\alpha^{1,4}_{2,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - \beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot} \tag{70}$$

$$\alpha^{1,4}_{1,\cdot,\cdot,\cdot} = -\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot} + \beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{71}$$

$$\alpha^{1,4}_{0,\cdot,\cdot,\cdot} = -\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} - \beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{72}$$

Similarly, let $\alpha^{2,\cdot}_{\cdot,\cdot,\cdot,\cdot}$, $1 \le \cdot \le 16$ be the coeficients in (67) with respect to

$$\frac{\cdot}{\cdot} = (2\frac{\cdot}{\cdot} - \frac{\cdot}{\cdot})\frac{\cdot}{\cdot}$$

). Then we have

$$\alpha^{2,4}_{3,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + \beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{73}$$

$$\alpha^{2,4}_{2,\cdot,\cdot,\cdot} = -\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot} + \beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{74}$$

$$= 2\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - \beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot} \tag{75}$$

$$= -2\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - \beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{76}$$

Let $\alpha^{3,\cdot}_{\cdot,\cdot,\cdot,\cdot}$, $1 \le \cdot \le 16$ be the coeficients in (67) with respect to

$$\frac{\cdot}{\cdot} = 3\frac{\cdot}{\cdot} - 2\frac{\cdot}{\cdot}$$

. Then we have

$$\alpha^{3,4}_{3,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} = 3\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{77}$$

$$\alpha^{3,4}_{2,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} - 3\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{78}$$

$$\alpha^{3,4}_{1,\cdot,\cdot,\cdot} = -\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 3\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{79}$$

$$= 3\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} + 3\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{80}$$

Finally, let $\alpha^{4,\cdot}_{\cdot,\cdot,\cdot,\cdot}$, $1 \le \cdot \le 16$ be the coeficients in (67) with respect to

$$\frac{\cdot}{\cdot} = 3\frac{\cdot}{\cdot} - 2\frac{\cdot}{\cdot}$$

Then we have

$$\alpha^{4,4}_{3,\cdot,\cdot,\cdot} = \beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - 3\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{81}$$

$$= -\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 3\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} \tag{82}$$

$$= -\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{1}_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot} - 3\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot} \tag{83}$$

$$= -\beta'_{\cdot,\cdot}\beta^{4}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} - 2\beta'_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot}\beta^{2}_{\cdot,\cdot} + 3\beta'_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot}\beta^{3}_{\cdot,\cdot} \tag{84}$$

In the following, we describe of the roadmap of proving (66) using (69)-(84).

Proposition 7. First of all, $\forall \cdot \ge 1$,

$$\alpha^{\cdot}_{\cdot,0} = \alpha^{\cdot}_{0,\cdot} = 0, \quad \forall 1 \le \cdot \le 4,\ 1 \le \cdot,\cdot,\ \cdot \ne \{\cdot,\cdot,\cdot\}. \tag{85}$$

Proof. See Section A.3.2 □

Proposition 8. Second, $\forall \cdot \ge 1$,

$$\alpha^{\cdot}_{0,\cdot,\cdot} = 0, \quad \forall 1 \le \cdot' \le 4,\ 1 \le \cdot,\cdot. \tag{86}$$

Proof. See Section A.3.3 □

Proposition 9. Finally, $\forall \cdot \ge 1$,

$$\alpha^{\cdot}_{\cdot,\cdot} = 0, \quad \forall 1 \le \cdot \le 4,\ 1 \le \cdot,\cdot,\ \cdot \ne \{\cdot,\cdot,\cdot\}. \tag{87}$$

Proof. See Section A.3.4 □

A.3.1 Proof of Prop. 6. We first prove that 16 Fourier basis are orthogonal. Then if $\cdot(\cdot,\cdot)$ is zero, it means the coeficient before each basis is zero. We denote the basis functions as $\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)$, $\cdot = 1, ..., 16$. Here $\beta^{1}_{\cdot,\cdot,\cdot,\cdot} = \cdot(\cdot)\cdot(\cdot)\cdot(\cdot)\cdot(\cdot)$, $\beta^{2}_{\cdot,\cdot,\cdot,\cdot} = \cdot(\cdot)\cdot(\cdot)\cdot(\cdot')\cdot(\cdot')$, ... $\cdot(\cdot,\cdot)$ can be written as

$$\cdot(\cdot,\cdot) = \sum_{\cdot,\cdot,\cdot,\cdot=0}^{\cdot} \alpha^{\cdot}_{\cdot,\cdot,\cdot,\cdot}\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot) \tag{88}$$

Multiply a basis function on both side and take the integral over the period:

$$\int_{-\cdot}^{\cdot}\int_{-\cdot}^{\cdot} \cdot(\cdot,\cdot)\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\,\cdot\cdot = \tag{89}$$

$$\int_{-\cdot}^{\cdot}\int_{-\cdot}^{\cdot} \sum_{\cdot,\cdot,\cdot,\cdot=0}^{\cdot} \alpha^{\cdot}_{\cdot,\cdot,\cdot,\cdot}\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\,\cdot\cdot \tag{90}$$

We want to show $\alpha^{\cdot}_{\cdot,\cdot,\cdot,\cdot}$ is zero. It's easy to show that if $\cdot \ne \cdot$, or $\cdot \ne \cdot$, or $\cdot \ne \cdot$, or $\cdot' \ne \cdot'$, or $\cdot' \ne \cdot'$

$$\int_{-\cdot}^{\cdot}\int_{-\cdot}^{\cdot} \sum_{\cdot,\cdot,\cdot,\cdot=0}^{\cdot} \alpha^{\cdot}_{\cdot,\cdot,\cdot,\cdot}\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\,\cdot\cdot = 0 \tag{91}$$

So

$$\int_{-\cdot}^{\cdot}\int_{-\cdot}^{\cdot} \cdot(\cdot,\cdot)\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)\,\cdot\cdot \tag{92}$$

$$= \int_{-\cdot}^{\cdot}\int_{-\cdot}^{\cdot} \beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot))^{2}\,\cdot\cdot \tag{93}$$

$$= \frac{1}{\cdot}\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot} \tag{94}$$

We have $\cdot(\cdot,\cdot) = 0$, so $\alpha^{\cdot}_{\cdot,\cdot,\cdot,\cdot} = 0$.

Next we gather all different coeficients before $\beta^{\cdot}_{\cdot,\cdot,\cdot,\cdot}(\cdot,\cdot)$. This difference is caused by the different arrangement of $(\cdot,\cdot,\cdot',\cdot')$. By observation, we obtain (76).

□

A.3.2 Proof of Prop. 7. We only prove $\alpha^{\cdot}_{\cdot,0} = 0$ as the proof of $\alpha^{\cdot}_{0,\cdot} = 0$ is essentially similar. Note that $\alpha^{\cdot,\cdot'}_{\cdot,0,0,0} = 0$. Therefore, according to (68), we have

$$\alpha^{0,\cdot}_{\cdot,\cdot,\cdot,0} = 0, \quad 1 \le \cdot, 1 \le \cdot \le 4. \tag{95}$$

Substituting (95) into (69), (73), (77), and (81), we have

$$\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0} = 2\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0}$$

$$\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0} = 0$$

$$\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0} = 3\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0}$$

$$\alpha^{1}_{0,0}\alpha^{3}_{\cdot,0} = 0$$

As $\Gamma^1_{0,0} \neq 0, \forall i \in \{x, y, z\}$, we have $\Gamma^3_{i,0} = \Gamma^3_{i,0} = \Gamma^3_{i,0} = 0$. Similarly, we have

$$\Gamma^{j,k}_{i,j,0} = 0, \quad 1 \leq i, 1 \leq j \leq k \leq 4. \tag{96}$$

Substituting (96) into (69), (73), (77), and (81), we have

$$\begin{aligned}
&\Gamma^1_{0,0}\Gamma^2_{1,0} = \\
&2\Gamma^1_{0,0}\Gamma^3_{1,0} - \Gamma^2_{0,0}\Gamma_{1,0} = \\
&0 - \Gamma^1_{0,0}\Gamma^2_{1,0} = \\
&3\Gamma^1_{0,0}\Gamma^3_{1,0} - \Gamma^2_{0,0}\Gamma_{1,0} = \\
&0
\end{aligned}$$

As $\Gamma^1_{0,0} \neq 0, \forall i \in \{x, y, z\}$, we have $\Gamma^2_{i,0} = \Gamma^2_{i,0} = \Gamma^2_{i,0} = 0$. Moreover, (68) leads to

$$\Gamma^{j,k}_{i,j,0} + \Gamma^{j,k}_{i,j,0} = 0, \quad 1 \leq i, 1 \leq j \leq k \leq 4. \tag{97}$$

Substituting (97) into (69), (73), (77), and (81), we have

$$\begin{aligned}
&\Gamma^1_{0,0}(\Gamma^4_{i,0} - \Gamma^1_{i,0}) = 2\Gamma^2_{0,0}(\Gamma^3_{1,0} \\
&- \Gamma^3_{i,0}) - 2\Gamma^2_{0,0}(\Gamma^2_{i,0} - \Gamma^2_{i,0}) = 0 \\
&\Gamma^1_{0,0}(\Gamma^4_{i,0} - \Gamma^1_{i,0}) = 3\Gamma^2_{0,0}(\Gamma^4_{i,0} - \Gamma^1_{i,0}) \\
&2\Gamma^1_{0,0}(\Gamma^4_{i,0} - \Gamma^1_{i,0}) = 0
\end{aligned}$$

As $\Gamma^4_{i,0} = 0, \Gamma^1_{0,0} \neq 0, \forall i \in \{x, y, z\}$, we have $\Gamma^2_{i,0} = \Gamma^3_{i,0} = 0, \forall i \in \{x, y, z\}$.

□

A.3.3 Proof of Prop. 8. When $\Gamma^i_{j,0} = \Gamma^j_{0,i} = 0, 1 \leq i \leq 4, \forall i \in \{x, y, z\}$, applying (69)-(84) we have

$$\Gamma^{j,k}_{i,j,k,0} = \Gamma^{0,k}_{i,j,k,0} = \Gamma^{0,0}_{i,j,k,0} = 0, 1 \leq i \leq 4, \ 1 \leq j \leq k \leq 16, \ 1 \leq i, j, k. \tag{98}$$

Using (68), we have

$$\Gamma^{j,k}_{0,i,j,k} = 0, 1 \leq i \leq 4, \ 1 \leq j \leq k \leq 16. \tag{99}$$

□

A.3.4 Proof of Prop. 9. We only show $\Gamma^2_{i,j} = \Gamma^3_{i,j} = 0, \forall i, j \in \{x, y, z\}$ as the proof of $\Gamma^1_{i,j} = \Gamma^4_{i,j} = 0$ is very similar. Note that $\Gamma^0_{0,i,j} = \Gamma^1_{i,j}$. Applying (69), (73), (77), and (81), we arrive at

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + \Gamma^1_{0,0}\Gamma^2_{i,j} = 2\Gamma^1_{0,0}\Gamma^3_{i,j} \tag{100}$$

$$\Gamma^1_{0,0}\Gamma^2_{i,j} + \Gamma^3_{0,0}\Gamma^3_{i,j} = 2\Gamma^1_{0,0}\Gamma^3_{i,j} \tag{101}$$

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + 2\Gamma^1_{0,0}\Gamma^2_{i,j} = 3\Gamma^2_{0,0}\Gamma^3_{i,j} \tag{102}$$

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + 2\Gamma^1_{0,0}\Gamma^3_{i,j} = 3\Gamma^3_{0,0}\Gamma^3_{i,j} \tag{103}$$

From (100) and (101), we have

$$\Gamma^3_{i,j} = \frac{\Gamma^1_{0,0}}{2\Gamma^1_{0,0}}\Gamma^3_{i,j} + \frac{\Gamma^2_{i,j}}{2\Gamma^1_{0,0}}\Gamma^2_{i,j} \tag{104}$$

$$\Gamma^2_{i,j} = \frac{\Gamma^1_{0,0}}{2\Gamma^1_{0,0} - \Gamma^0_{0,0}}\Gamma^2_{i,j} + \frac{\Gamma^2_{i,j}}{2\Gamma^1_{0,0} - \Gamma^0_{0,0}}\Gamma^3_{i,j} \tag{105}$$

Substituting (105) and (104) into (102) and (101), respectively, we obtain

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + \frac{\Gamma^1_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Gamma^2_{i,j} = 2\Gamma^2_{0,0}\Gamma^3_{i,j} \tag{106}$$

$$\Gamma^1_{0,0}\Gamma^2_{i,j} + \frac{\Gamma^1_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Gamma^3_{i,j} = 2\Gamma^3_{0,0}\Gamma^2_{i,j} \tag{107}$$

which lead to

$$\Gamma^3_{i,j} = \frac{2\Gamma^1_{0,0}}{\Gamma^1_{0,0}}\Gamma^3_{i,j} - \frac{\Gamma^1_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}}{\Gamma^1_{0,0}\Gamma^1_{0,0}}\Gamma^2_{i,j} \tag{108}$$

$$\Gamma^2_{i,j} = \frac{\Gamma^1_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}}{2\Gamma^1_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}}\Gamma^2_{i,j} + \frac{\Gamma^1_{0,0}\Gamma^2_{0,0} - \Gamma^0_{0,0}\Gamma^3_{0,0}}{\Gamma^1_{0,0}\Gamma^1_{0,0}}\Gamma^3_{i,j} \tag{109}$$

Moreover, substituting $\Gamma^4_{i,j,k} = 0$ into (72), (76), (80), and (84), we arrive at

$$\Gamma^1_{0,0}\Gamma^2_{i,j} + \Gamma^1_{0,0}\Gamma^3_{i,j} = 2\Gamma^2_{0,0}\Gamma^3_{i,j} \tag{110}$$

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + \Gamma^2_{0,0}\Gamma^2_{i,j} = 2\Gamma^1_{0,0}\Gamma^2_{i,j} \tag{111}$$

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + 2\Gamma^1_{0,0}\Gamma^2_{i,j} = 3\Gamma^2_{0,0}\Gamma^3_{i,j} \tag{112}$$

$$\Gamma^2_{0,0}\Gamma^2_{i,j} + 2\Gamma^2_{0,0}\Gamma^3_{i,j} = 3\Gamma^3_{0,0}\Gamma^3_{i,j} \tag{113}$$

Substituting (104) and (105) into (110)-(113), we arrive at

$$\Gamma^1_{0,0}\Gamma^2_{i,j} + \Gamma^1_{0,0}\Gamma^3_{i,j} = \Gamma^1_{0,0}\frac{\Gamma^1_{0,0}}{\Gamma^1_{0,0}}\Gamma^2_{i,j} + \frac{\Gamma^2_{i,j}}{\Gamma^1_{0,0}}\Gamma^3_{i,j} \tag{114}$$

$$\Gamma^1_{0,0}\Gamma^3_{i,j} + \Gamma^2_{0,0}\Gamma^2_{i,j} = \Gamma^2_{0,0}\frac{\Gamma^1_{0,0}}{\Gamma^1_{0,0}}\Gamma^2_{i,j} + \Gamma^2_{0,0}\Gamma^3_{i,j} \tag{115}$$

$$3\Gamma^2_{0,0}\Gamma^2_{i,j} = \Gamma^1_{0,0}\Gamma^2_{i,j} + \frac{\Gamma^1_{0,0}\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Gamma^2_{i,j} + \Gamma^1_{0,0}\Gamma^3_{i,j} \tag{116}$$

$$3\Gamma^3_{0,0}\Gamma^3_{i,j} = \Gamma^1_{0,0}\Gamma^3_{i,j} + \frac{\Gamma^1_{0,0}\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Gamma^3_{i,j} + \Gamma^1_{0,0}\Gamma^2_{i,j} \tag{117}$$

Substituting (108) and (109) into (114)-(117), we arrive at

$$\Gamma^1_{0,0}\Gamma^2_{i,j} + \frac{\Gamma^2_{0,0}\Gamma^1_{0,0}}{\Gamma^1_{0,0}\Gamma^1_{0,0}} - \frac{\Delta_{0,0}}{2\Gamma^1_{0,0}}\Gamma^2_{i,j} = 2\Gamma^1_{0,0} - \frac{\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Delta_{0,0}\Gamma^3_{i,j} \tag{118}$$

$$2\Gamma^1_{0,0}\Gamma^3_{i,j} = \Gamma^1_{0,0} + \frac{\Gamma^2_{0,0}(3\Gamma^0_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^0_{0,0}\Gamma^1_{0,0})}{2\Gamma^1_{0,0}\Gamma^1_{0,0}}\Gamma^2_{i,j} \tag{119}$$

$$\frac{\Gamma^2_{0,0}\Gamma^0_{0,0}(3\Gamma^0_{0,0}\Gamma^1_{0,0} - \Gamma^0_{0,0}\Gamma^0_{0,0}\Gamma^1_{0,0})}{2\Gamma^1_{0,0}\Gamma^1_{0,0}} + \frac{\Gamma^1_{0,0}\Gamma^1_{0,0} + \Gamma^0_{0,0}\Gamma^2_{0,0}}{2\Gamma^1_{0,0}}\Gamma^2_{i,j}$$

$$= \Gamma^1_{0,0}\frac{\Gamma^1_{0,0}}{\Gamma^1_{0,0}} - \Gamma^2_{i,j}\Gamma^3_{i,j} \tag{120}$$

$$5\Gamma^1_{0,0} - \frac{\Gamma^2_{0,0}}{\Gamma^1_{0,0}}\Gamma^3_{i,j} = \frac{3\Gamma^2_{0,0}\Gamma^1_{0,0}}{\Gamma^1_{0,0}}\Gamma^2_{i,j} \tag{121}$$

where $\Delta_{0,0} = \Gamma^1_{0,0}\Gamma^0_{0,0} - \Gamma^0_{0,0}\Gamma^2_{0,0}$. It is clear that (118)-(121) are true iff $\Gamma^2_{i,j} = \Gamma^3_{i,j} = 0$, which means $\Gamma^2_{i,j} = \Gamma^2_{i,j} = 0$ according to (108) and (109), and $\Gamma^3_{i,j} = \Gamma^3_{i,j} = 0$ according to (104) and (105).

B PROOF OF PROP. 3

Axis-aligned geodesic interpolation property is satisfied iff (30) is satisfied for $v = e_{\square}$. When $\square \neq \square$, $\square_{\square} = 0$. In this case, (30) is equivalent to (9). When $\square = \square$, $\square_{\square} = 1$. In this case, (30) is equivalent to (10). □

C PROOF OF PROP. 4

Given initial tangent directions $v_{\square}, \square \in [\square]$ at z_0 that form a basis of $R^{\square}$, we construct a mapping $z' \rightarrow z$ in a local neighborhood of 0, so that 1) $0 \rightarrow z_0$, 2) $\frac{\partial z}{\partial \square} = v_{\square}$, and 3) axes of z' follow geodesic curve. To this end, we define $z(z')$ recursively. To begin with, we define $z(\square'_1, 0, \cdots, 0)$, so that it follows the geodesic curve passing thorough z_0 with tangent direction v_1 and $\square'_1$ specifies the displacement along this geodesic curve. Suppose we have defined $z(\square'_1, \cdots, \square'_{\square}, 0, \cdots, 0)$. We first parallel translate $v_{\square+1}$ along geodesic segments $z(\square'_1, \cdots, 0, , \cdots, 0)$ to $z(\square'_1, \cdots, \square'_{\square}, 0, \cdots, 0)$ where $1 \leq \square \leq \square$. Let $v'_{\square+1}$ be the result of this parallel transport. We then define $z(\square'_1, \cdots, \square'_{\square}, \square'_{\square+1}, \cdots, 0)$ by shooting a geodesic curve at $z(\square'_1, \cdots, \square'_{\square}, 0, \cdots, 0)$ in the direction of $\hat{v}_{\square+1}$ and traverse this curve by $\square'_{\square+1}$. It is clear that, by definition, different axes of z' follow geodesic curves. In addition, such a mapping is unique in a local neighborhood of z_0 because of properties of parallel transport and the fact that geodesic curves between two points are unique with a small local neighborhood of a point on a Riemannian manifold.

□