LIMP: Learning Latent Shape Representations with Metric Preservation Priors

Luca Cosmo[1,2], Antonio Norelli[1], Oshri Halimi[3], Ron Kimmel[3], Emanuele Rodolà[1]

[1] Sapienza University of Rome, Italy
[2] University of Lugano, Switzerland
[3] Technion - Israel Institute of Technology, Israel

Abstract. In this paper, we advocate the adoption of metric preservation as a powerful prior for learning latent representations of deformable 3D shapes. Key to our construction is the introduction of a geometric distortion criterion, defined directly on the decoded shapes, translating the preservation of the metric on the decoding to the formation of linear paths in the underlying latent space. Our rationale lies in the observation that training samples alone are often insufficient to endow generative models with high fidelity, motivating the need for large training datasets. In contrast, metric preservation provides a rigorous way to control the amount of geometric distortion incurring in the construction of the latent space, leading in turn to synthetic samples of higher quality. We further demonstrate, for the first time, the adoption of differentiable intrinsic distances in the backpropagation of a geodesic loss. Our geometric priors are particularly relevant in the presence of scarce training data, where learning any meaningful latent structure can be especially challenging. The effectiveness and potential of our generative model is showcased in applications of style transfer, content generation, and shape completion.

Keywords: learning shapes,generative model,metric distortion

1 Introduction

Constructing high-fidelity generative models for 3D shapes is a challenging problem that has met with increasing interest in recent years. Generative models are applicable in many practical domains, ranging from content creation to shape exploration, as well as in 3D reconstruction. As a new generation of methods, they come to face a number of difficulties.

Most existing approaches address the case of static or *rigid* geometry, for example, man-made objects like chairs and airplanes, with potentially high intra-class variability; see the ShapeNet [7] repository for such examples. In this setting, the main focus has been on the abstraction capabilities of the encoder and the generator, describing complex 3D models in terms of their core geometric features via parsimonious part-based representations. Shapes generated with these techniques are usually designed to have valid part semantics that are easy

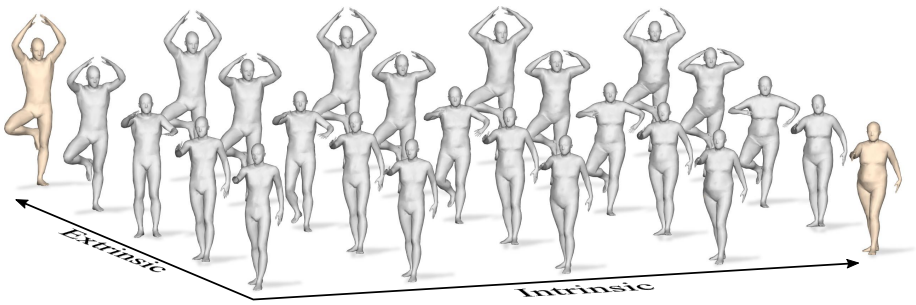


Fig. 1. Disentangled interpolation of FAUST shapes, obtained with our generative model trained under metric preservation priors. The yellow shapes at the two corners are given as input; the remaining shapes are generated by bilinearly interpolating the latent codes of the input, and decoding the resulting codes. Our model allows to disentangle pose from identity, illustrated here as different dimensions.

to parse. Concurrently, several recent efforts have concentrated on the definition of convenient representations for the 3D *output*; these methods find broader application in multiple tasks, where they enable more efficient and high-quality synthesis, and can be often plugged into existing generative models.

To date, relatively fewer approaches have targeted the *deformable* setting, where the generated shapes are related by continuous, non-rigid deformations. These model a range of natural phenomena, such as changes in pose and facial expressions of human subjects, articulations, garment folding, and molecular flexibility to name but a few. The extra difficulties brought by such non-rigid deformations can be tackled, in some cases, by designing mathematical or parametric models for the deformation at hand; however, these models are often violated in practice, and can be very hard to devise for general deformations – hence the need for learning from examples.

The framework we propose is motivated by the observation that existing data-driven approaches for learning deformable 3D shapes, and autoencoders (AE) in particular, do not make use of any *geometric prior* to drive the construction of the latent space, whereas they rely almost completely on the expressivity of the training dataset. This imposes a heavy burden on the learning process, and further requires large annotated datasets that can be costly or even impossible to acquire. In the absence of additional regularization, limited training data leads to limited generalization capability, which is manifested in the generated 3D shapes exhibiting unnatural distortions. Variational autoencoders (VAE) provide a partial remedy by modeling a distributional prior on the data via a parametrized density on the latent space. This induces additional regularization, but is still insufficient to guarantee the preservation of geometric properties in the output 3D models.

In this paper, we introduce *Latent Interpolation with Metric Priors* (LIMP). We propose to explicitly model the local *metric* properties of the latent space by enforcing metric constraints on the decoded output. We do this by phrasing

a metric distortion penalty that has the effect to promote naturally looking deformations, and in turn to significantly reduce the need for large datasets at training time. In particular, we show that by coupling the Euclidean distances among latent codes (hence, along linear paths in the latent space) to the metric distortion among decoded shapes, we obtain a strong regularizing effect in the construction of the latent space. Another novel ingredient of the proposed approach is the backpropagation of intrinsic (namely, geodesic) distances during training, which is made possible by a recent geodesic computation technique. Using geodesics makes our approach more flexible, and enables the successful application of our generative model to style and pose transfer applications. See Figure 1 for an example of novel samples synthesized with our generative model.

2 Related Work

Our method falls within the class of AE-based generative models for 3D shapes. In this Section we cover methods from this family that are more closely related to ours, and refer to the recent survey [8] for a broader coverage.

In the 3D computer vision and graphics realms, generative models for part-compositional 3D objects play the lion's share. Such approaches directly exploit the hierarchical, structural nature of 3D man-made objects to drive the construction of encoder and generator [32,22,29,31]. These methods leverage on the insight that objects can be understood through their components [26], making an interpretable representation close to human parsing possible. In this setting, a continuous exploration of the generated latent spaces is not always meaningful; the mechanism underlying typical operations like sampling and interpolation happen instead in *discrete* steps in order to generate plausible intermediate shape configurations (e.g., for transitioning from a 4-legged chair to a 3-legged stool). For this reason, with rigid geometry one usually deals with "structural blending" rather than continuous deformations. Structural blending has been realized, for instance, by learning abstractions of symmetry hierarchies via spatial arrangements of oriented bounding boxes [22], or by explicitly modeling part-to-part relationships [29]; generative-adversarial modeling has been applied on volumetric object representations [46]; structural hierarchies have been applied for the generation of composite 3D scenes [23] and building typologies [28] as well. Contributing to their success, is the fact that all these methods train on ShapeNet-scale annotated datasets with $> 50K$ unique 3D models, and the recent publication of dedicated benchmarks like PartNet [30] testify to the increasing interest of data-driven models for structure-aware geometry processing. In this paper, we address a different setting; we do not assume part-compositionality of the 3D models since we deal with deformable shapes, where continuous deformations are well-defined, and where annotated datasets are not as prominent.

A second thread of research revolves around the definition of a meaningful representation for the generated 3D output. While many approaches mostly use polygonal meshes with predefined topology or directly synthesize point clouds [1,42], the focus has been recently shifting towards more effective representations

in terms of overall quality, fidelity, and flexibility. These include approaches that predict implicit shape representations at the output, requiring an ex-post isosurface extraction step to generate a mesh at the desired resolution [27,33,16]; isosurfacing has been replaced by binary space partitioning in [9]; while in [18], shapes are represented by a set of parametric surface elements. In this work, we focus on learning a better *latent* representation for deformable shapes, rather than on constructing a better representation for the output.

More closely related to ours are some recent methods from the area of geometric deep learning. A graph-convolutional VAE with dynamic filtering convolutional layers [45] was introduced in [24] for the task of deformable shape completion of human shapes. The method is trained on ∼7000 shapes from the DFAUST dataset of real human scans [4]; due to the lack of any geometric prior, the learned generator introduces large distortions around points in the latent space that are not well represented in the training set. Geometric regularization was injected in [17] in the form of a template that parametrizes the surface. The method shows excellent performance in shape matching, however, it crucially relies on a large and representative dataset of $230,000$ shapes, and performance drops significantly with smaller training sets or bad initialization. More recently, a geometric disentanglement model for deformable point clouds was introduced in [2]. The proposed method uses Laplacian eigenvalues as a weak geometric prior to promote the separation of intrinsic and extrinsic shape information, together with several other de-correlation penalties, and a training set of $> 40K$ shapes. In the absence of enough training examples, the approach tends to produce a "morphing" effect between point clouds that does not correspond to a natural motion; a similar phenomenon was observed in [1]. Finally, in [44], a time-dependent physical prior was used to regularize interpolations in the latent space with the goal of obtaining a convincing simulation of moving tissues.

In particular, our approach bears some analogies with the theory of shape spaces [20], in that we seek to synthesize geometry that minimizes a deformation energy. For example, in [14] it was shown how to *axiomatically* modify a noisy shape such that its intrinsic measures would fit a given prior in a different pose. Differentiating the geodesic distances was done by fixing the update order in the fast marching scheme [21]. Our energy is not minimized over a fixed shape space, but rather, it drives the construction of a novel shape space in a data-driven fashion.

In this paper, we leverage classical ideas from shape analysis and metric geometry to ensure that shapes on the learned latent space correspond to plausible (i.e., low-distortion) deformations of the shapes seen at training time, even when only few training samples are available. We do this by modeling a geometric prior that promotes deformations with bounded distortion, and show that this model provides a powerful regularization for shapes *within* as well as *across* different classes, e.g., when transitioning between different human subjects.

3 Learning with metric priors

Our goal is to learn a latent representation for deformable 3D shapes. We do this by training a VAE on a training set $\mathcal{S} = \{\mathbf{X}_i\}$ of $|\mathcal{S}|$ shapes, under a purely geometric loss:

$$\ell(\mathcal{S}) = \ell_{\text{recon}}(\mathcal{S}) + \ell_{\text{interp}}(\mathcal{S}) + \ell_{\text{disent}}(\mathcal{S})\,. \tag{1}$$

The loss is composed of three terms. The first is a geometric reconstruction loss on the individual training shapes, as in classical AE's; the second one is a pairwise interpolation term for points in the latent space; the third one is a disentanglement term to separate intrinsic from extrinsic information.

The main novelty lies in (1) the interpolation loss, and (2) the disentanglement loss *not* relying upon corresponding poses in the training set. The interpolation term provides control over the encoding of each shape *in relation to the others*. This induces a notion of proximity between latent codes that is explicitly linked, in the definition of the loss, to a notion of metric distortion between the decoded shapes. As we show in the following, this induces a strong regularization on the latent space and rules out highly distorted reconstructions.

The disentanglement loss promotes the factorization of the latent space into two orthogonal components: One that spans the space of isometries (e.g., change in pose), and another that spans the space of non-isometric deformations (e.g., change in identity). As in the interpolation loss, for the disentanglement we also exploit the metric properties of the decoded shapes.

3.1 Losses

We define $\mathbf{z} := \text{enc}(\mathbf{X})$ to be the latent code for shape $\mathbf{X}$, and $\mathbf{X}' := \text{dec}(\mathbf{z})$ to be the corresponding decoding. During training, the decoder (dec) and encoder (enc) are updated so as to minimize the overall loss of Eq. (1); see Section 3.3 for the implementation details.

Geometric reconstruction. The reconstruction loss is defined as follows:

$$\ell_{\text{recon}}(\mathcal{S}) = \sum_{i=1}^{|\mathcal{S}|} \|\mathbf{D}_{\mathbb{R}^3}(\mathbf{X}'_i) - \mathbf{D}_{\mathbb{R}^3}(\mathbf{X}_i)\|_F^2\,, \tag{2}$$

where $\mathbf{D}_{\mathbb{R}^3}(\mathbf{X})$ is the matrix of pairwise Euclidean distances between all points in $\mathbf{X}$, and $\|\cdot\|_F$ denotes the Frobenius norm. Eq. (2) measures the cumulative reconstruction error (up to a global rotation) over the training shapes.

Metric interpolation. This loss is defined over all possible pairs of shapes $(\mathbf{X}_i, \mathbf{X}_j)$:

$$\ell_{\text{interp}}(\mathcal{S}) = \sum_{i \neq j}^{|\mathcal{S}|} \|\mathbf{D}(\text{dec}(\underbrace{(1-\alpha)\mathbf{z}_i + \alpha\mathbf{z}_j}_{\substack{\text{interpolation of} \\ \text{latent codes}}})) - \underbrace{((1-\alpha)\mathbf{D}(\mathbf{X}'_i) + \alpha\mathbf{D}(\mathbf{X}'_j))}_{\substack{\text{interpolation of} \\ \text{geodesic or local distances}}}\|_F^2\,, \tag{3}$$

where $\alpha \sim \mathcal{U}(0,1)$ is a uniformly sampled scalar in $(0,1)$, different for each pair of shapes. In the equation above, the matrix $\mathbf{D}(\mathbf{X})$ encodes the pairwise distances between points in $\mathbf{X}$. We use two different definitions of distance, giving rise to two different losses which we sum up together. In one loss, $\mathbf{D}$ contains *geodesic* distances between *all* pairs of points. In the second loss, we consider *local Euclidean* distances from each point to points within a small neighborhood (set to 10% of the shape diameter); the rationale is that local Euclidean distances capture local detail and tend to be resilient to non-rigid deformations, as observed for instance in [40]. All distances are computed on the fly, on the decoded shapes, at each forward step.

Since the error criterion in Eq. (3) encodes the discrepancy between pairwise distance matrices, we refer to it as a *metric preservation prior*. We refer to Section 3.2 for a more in-depth discussion from a continuous perspective.

Disentanglement. We split the latent codes into an intrinsic and an extrinsic part, $\mathbf{z} := (\mathbf{z}^{\text{int}}|\mathbf{z}^{\text{ext}})$. The former is used to encode "style", i.e., the space of non-isometric deformations; the latter is responsible for changes in pose, and is therefore constrained to model the space of possible isometries.

The loss is composed of two terms:

$$\ell_{\text{disent}}(\mathcal{S}) = \ell_{\text{int}}(\mathcal{S}) + \ell_{\text{ext}}(\mathcal{S}), \quad \text{with} \tag{4}$$

$$\ell_{\text{int}}(\mathcal{S}) = \sum_{\substack{i \neq j \\ \text{iso}}}^{|\mathcal{S}|} \|\mathbf{D}_{\mathbb{R}^3}(\text{dec}(\underbrace{(1-\alpha)\mathbf{z}_i^{\text{int}} + \alpha \mathbf{z}_j^{\text{int}}}_{\substack{\text{interpolation of} \\ \text{style}}} | \mathbf{z}_i^{\text{ext}})) - \mathbf{D}_{\mathbb{R}^3}(\mathbf{X}_i)\|_F^2 \tag{5}$$

$$\ell_{\text{ext}}(\mathcal{S}) = \sum_{\substack{i \neq j \\ \text{non-iso}}}^{|\mathcal{S}|} \|\mathbf{D}_g(\text{dec}(\mathbf{z}_i^{\text{int}} | \underbrace{(1-\alpha)\mathbf{z}_i^{\text{ext}} + \alpha \mathbf{z}_j^{\text{ext}}}_{\substack{\text{interpolation of} \\ \text{pose}}})) - \mathbf{D}_g(\mathbf{X}_i)\|_F^2 \tag{6}$$

The ℓ_{int} term is evaluated only on isometric pairs (i.e., just a change in pose), for which we expect $\mathbf{z}_i^{\text{int}} = \mathbf{z}_j^{\text{int}}$. For a pair $(\mathbf{X}_i, \mathbf{X}_j)$, it requires that $\mathbf{X}_i$ can be reconstructed exactly even when its intrinsic part $\mathbf{z}_i^{\text{int}}$ is interpolated with that of $\mathbf{X}_j$. This enforces $\mathbf{z}_i^{\text{int}} = \mathbf{z}_j^{\text{int}}$, thus all the pose-related information is forced to move to $\mathbf{z}^{\text{ext}}$.

The ℓ_{ext} term is instead evaluated on *non*-isometric pairs. Here we require that the *geodesic* distances of $\mathbf{X}_i$ are left untouched when we interpolate its pose with that of $\mathbf{X}_j$. This way, we force all the style-related information to be moved to $\mathbf{z}^{\text{int}}$. We see that by having direct access to the metric on the decoded shapes, we can phrase the disentanglement easily in terms of distances.

The assumption that the metric is nearly preserved under pose changes is widely used in many shape analysis applications such as shape retrieval [38], matching [39,12,11,19] and reconstruction [5,10]

Relative error. In practice, we always measure the error on the Euclidean distances (appearing in Eqs. (2),(3),(5)) in a *relative* sense. Let $\mathbf{A}$ be the "ground truth"

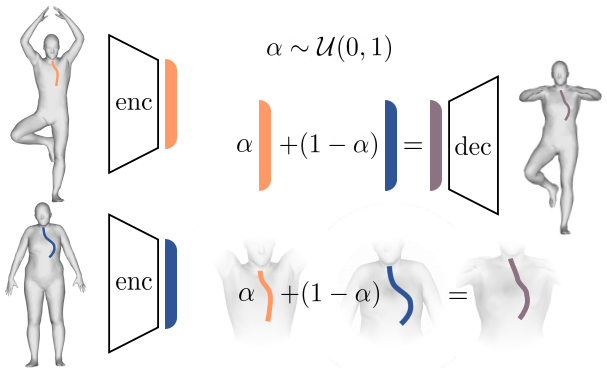


Fig. 2. Our architecture is a standard VAE, with PointNet as the encoder and a fully connected decoder. Our loss asks that the geodesic distances on the decoded convex combination of latent codes (middle row) are equal to the the convex combination of the input distances.

Euclidean distance matrix computed on the input shape, and let $\mathbf{B}$ be its predicted reconstruction. Instead of taking $\|\mathbf{A}-\mathbf{B}\|_F^2 = \sum_{ij}(\mathbf{A}_{ij}-\mathbf{B}_{ij})^2$, we compute the relative error $\sum_{ij}\frac{(\mathbf{A}_{ij}-\mathbf{B}_{ij})^2}{\mathbf{A}_{ij}^2}$. In our experiments, this resulted in better reconstruction of local details than by using the simple Frobenius norm.

3.2 Continuous interpretation

In the continuous setting, we regard shapes as metric spaces $(\mathcal{X}, d_{\mathcal{X}})$, each equipped with a distance function $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Given two shapes $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$, a map $\phi : \mathcal{X} \to \mathcal{Y}$ is an *isometry* if it is surjective and preserves distances, $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x),\phi(x'))$ for all $x, x' \in \mathcal{X}$. Isometries play a fundamental role in 3D shape analysis, since they provide a mathematical model for natural deformations like changes in pose. In practice, however, isometry is rarely satisfied exactly.

Why interpolation? Our approach is based on the insight that *non*-isometric shapes are related by sequences of near-isometric deformations, which, in turn, have a well defined mathematical model. In our setting, we do not require the training shapes to be near-isometric. Instead, we allow for maps ϕ with *bounded metric distortion*, i.e., for which there exists a constant $K > 0$ such that:

$$|d_{\mathcal{X}}(x,x') - d_{\mathcal{Y}}(\phi(x),\phi(x'))| \leq K \tag{7}$$

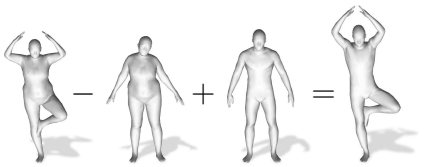
for all $x, x' \in \mathcal{X}$. For $K \to 0$ the map ϕ is a near-isometry, while for general $K > 0$ we get a much wider class of deformations, going well beyond simple changes in pose. We therefore assume that there exists a map with bounded distortion between all shape pairs in the training set.

At training time, we are given a map $\phi : \mathcal{X} \to \mathcal{Y}$ between two training shapes $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$. We then assume there exists an abstract metric space $(\mathcal{L}, d_{\mathcal{L}})$ where each point is a shape; this "shape space" is the latent space that we seek to represent when training our generative model. Over the latent space we construct a parametric sequence of shapes $\mathcal{Z}_\alpha = (\mathcal{X}, d_\alpha)$, parametrized by $\alpha \in (0, 1)$, connecting $(\mathcal{X}, d_{\mathcal{X}})$ to $(\mathcal{Y}, d_{\mathcal{Y}})$. By modeling the intermediate shapes as $(\mathcal{X}, d_\alpha)$, we regard each $\mathcal{Z}_\alpha$ as a continuously deformed version of $\mathcal{X}$, with a different metric defined by the interpolation:

$$d_\alpha(x, x') = (1 - \alpha) d_{\mathcal{X}}(x, x') + \alpha d_{\mathcal{Y}}(\phi(x), \phi(x')) \,, \tag{8}$$

for all $x, x' \in \mathcal{X}$. Each $\mathcal{Z}_\alpha$ in the sequence has the same points as $\mathcal{X}$, but the shape is different since distances are measured differently.

It is easy to see that if the training shapes $\mathcal{X}$ and $\mathcal{Y}$ are isometric, then $d_\alpha(x, x') = d_{\mathcal{X}}(x, x')$ for all $x, x' \in \mathcal{X}$ and the entire sequence is isometric, i.e., we are modeling a change in pose. However, if $\phi : \mathcal{X} \to \mathcal{Y}$ has bounded distortion without being an isometry, each intermediate shape $(\mathcal{X}, d_\alpha)$ also has bounded distortion with respect to $(\mathcal{X}, d_{\mathcal{X}})$, with $K_\alpha < K$ in Eq. (7); in particular, for $\alpha \to 0$ one gets $K_\alpha \to 0$ and therefore a near-isometry. In other words, by using the metric interpolation loss of Eq. (3), as α grows from 0 to 1 we are modeling a general non-isometric deformation as a sequence of approximate isometries.



Flattening of the latent space. Taking a linear convex combination of latent vectors as in Eq. (3) implies that distances between codes should be measured using the Euclidean metric $\| \cdot \|_2$. This enables algebraic manipulation of the codes and the formation of "shape analogies", as shown in the inset (real example based on our trained model). By the connection of Euclidean distances in the latent space with intrinsic distances on the decoder's output, our learning model performs a "flattening" operation, in the sense that it requires the latent space to be as Euclidean as possible, while absorbing any embedding error in the decoder. A similar line of thought was followed, in a different context, in the purely axiomatic model of [41].

3.3 Implementation

We design our deep generative model as a VAE (Figure 2). The input data is a set of triangle meshes; each mesh is encoded as a matrix of vertex positions $\mathbf{X} \in \mathbb{R}^{n \times 3}$, together with connectivity encoded as a $n \times n$ adjacency matrix. We anticipate here that mesh connectivity is never accessed directly by the network.

Architecture. The encoder takes vertex positions $\mathbf{X}$ as input, and outputs a d-dimensional code $\mathbf{z} = \mathrm{enc}(\mathbf{X})$. Similarly, the decoder outputs vertex positions $\mathbf{Y} = \mathrm{dec}(\mathbf{z}) \in \mathbb{R}^{n \times 3}$. In order to clarify the role of our priors versus the sophisticacy of the architecture, we keep the latter as simple as possible. In particular, we

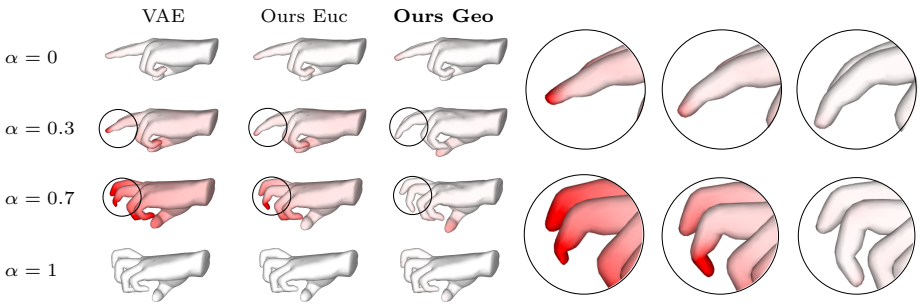


Fig. 3. Interpolation example on a small training set of just 5 shapes, where the deformation evolves from top ($\alpha = 0$) to bottom ($\alpha = 1$). Color encodes the per-point metric distortion, growing from white to red; changes in pose as in this example should have distortion close to zero. We show the results obtained by three different networks: baseline VAE; ours with Euclidean metric regularization only; ours with Euclidean *and* geodesic regularization (i.e., the complete loss).

adopt a similar architecture as in [2]; we use PointNet [34] with spatial transform as the encoder, and a simple MLP as the decoder. We reserve 25% of the latent code for the extrinsic part and the remaining 75% for the intrinsic representation, while the latent space and layer dimensions vary depending on the dataset size. A detailed description of the network is deferred to the Supplementary Material. We implemented our model in PyTorch using Adam as optimizer with learning rate of 1e-4. To avoid local minima and numerical errors in gradient computation, we start the training by optimizing just the reconstruction loss for 10^4 iterations, and add the remaining terms for the remaining epochs.

Geodesic distance computation. A crucial ingredient to our model is the computation of geodesic distances $\mathbf{D}_g(\mathrm{dec}(\mathbf{z}))$ during training, see Eq. (3). We use the heat method of [13] to compute these distances, based on the realization that its pipeline is fully differentiable. It consists, in particular, of two linear solves and one normalization step, and all the quantities involved in the three steps depend smoothly on the vertex positions given by the decoder (we refer to the Supplementary Material for additional details).

To our knowledge, this is the first time that on-the-fly computation of geodesic distances appears in a deep learning pipeline. Previous approaches using geodesic distances, such as [19], do so by taking them as pre-computed input data, and leave them untouched for the entire training procedure.

Supervision. We train on a collection of shapes with known pointwise correspondences; these are needed in Eq. (3), where we assume that the distance matrices have compatible rows and columns. From a continuous perspective, we need maps for the interpolated metric of Eq. (8) to be well defined. Known correspondences are also needed by other approaches dealing with deformable data [25,24,17]. In practice, we only need few such examples (we use < 100 training shapes), since we rely for the most part on the regularization power of our geometric priors.

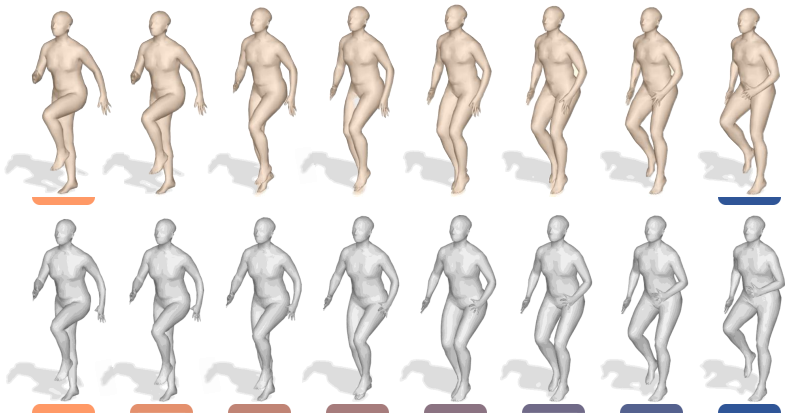


Fig. 4. *Top row*: A 4D sequence from the real-world dataset DFAUST. We train our generative model on the left- and right-most keyframes (indicated by the orange and blue bar respectively), together with keyframes extracted from other sequences and different individuals. *Bottom row*: The 3D shapes generated by our trained model. Visually, both the generated and the real-world sequences look plausible, indicating that geometric priors are well-suited for regularizing toward realistic deformations.

Differently from [24,17] we do *not* assume the training shapes to have the same mesh, since the latter is only used as an auxiliary structure for computing geodesics in the loss; the network only ever accesses vertex positions. Further, we do not require training shapes with similar poses across different subjects.

4 Results

4.1 Data

To validate our method, we performed experiments using 5 different datasets (3 are obtained from real-world scans, 2 are fully synthetic). **FAUST** [3] is composed

Table 1. Ablation study in terms of interpolation and disentanglement error on 4 datasets. Our full pipeline (denoted by 'Ours Geo') achieves the minimum error in all cases, and is **more than one order of magnitude** better than the baseline VAE on the interpolation. We do not report the disentanglement error for HANDS, since the dataset only contains one hand style.

	Interpolation Error			Disentanglement Error		
	VAE	Ours Euc	Ours Geo	VAE	Ours Euc	Ours Geo
FAUST	3.89e-2	5.08e-3	**3.82e-3**	7.16	4.04	**3.48**
DFAUST	9.82e-2	3.43e-3	**2.89e-4**	6.15	4.90	**4.11**
COMA	1.32e-3	1.03e-3	**7.51e-4**	1.55	1.30	**1.22**
HANDS	6.01e-3	8.12e-4	**4.62e-4**	-	-	-

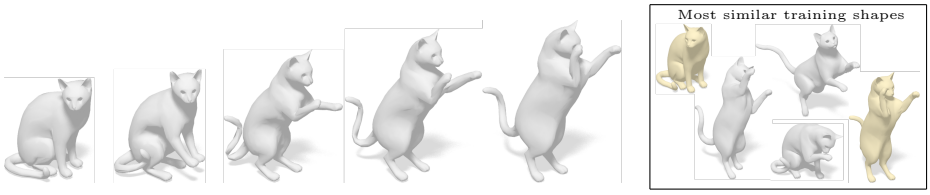


Fig. 5. Interpolation example on the *cat* shapes of TOSCA dataset [6]. On the left, we show an interpolation sequence between two shapes of the training set (yellow shapes on the right). On the right, we manually selected the most similar shapes present in the training set, composed in total by just 11 shapes. You can appreciate how shapes in the middle of the interpolated sequence significantly differ from the training shapes.

of 10 different human subjects, each captured in 10 different poses. We train our network on 8 subjects (thus, 80 meshes in total) and leave out the other 2 subjects for testing. **DFAUST** [4] is a 4D dataset capturing the motion of 10 human subjects performing 14 different activities, spanning *hundreds* of frames each. As training data **we only use 4 representative frames from each subject/sequence pair**. **COMA** [36] is another 4D dataset of human faces; it is composed of 13 subjects, each performing 13 different facial expressions represented as a sequence of 3D meshes. As opposed to the test split proposed in [37], where 90% of the data is used for training, we only select 14 frames for each subject (one representative for each of the 13 expressions, plus one in a neutral pose), thus **training with less than 1% of the dataset**. **TOSCA** [6] is a synthetic dataset containing both animals and human bodies. In our experiments we use only the *cat* class, containing 11 shapes in different poses. The last dataset, which we refer to as **HANDS**, is also completely synthetic and consists of 5 meshes depicting one hand in 5 different poses. For all the datasets, we subsample the meshes to 2500 vertices by iterative edge collapse [15].

4.2 Interpolation

We first perform a classical interpolation experiment. Given two shapes $\mathbf{X}$ and $\mathbf{Y}$, we visualize the decoded interpolation of their latent codes, given by $\mathrm{dec}((1-\alpha)\mathrm{enc}(\mathbf{X})+\alpha\mathrm{enc}(\mathbf{Y}))$ for a few choices of $\alpha \in (0,1)$. We measure the interpolation quality via the *interpolation error*, defined as the average (over all surface points) geodesic distortion of the interpolated shapes.

Two examples of interpolation are shown in Figures 3 and 5. In these examples, the training sets consist of just **5** and **11 shapes** respectively, meaning that the intermediate poses have never been seen before. In this few-shot setting, proper regularization is crucial to get meaningful results. In the experiment in Figure 3, we also conduct an ablation study. We disable all the interpolation terms from our complete loss, resulting in a baseline VAE; then we disable the geodesic regularization only; finally we keep the entire loss intact, showing best results. Quantitative results on 4 different datasets are reported in Table 1 (first 3 columns), showing that best results are obtained when our full loss is used.

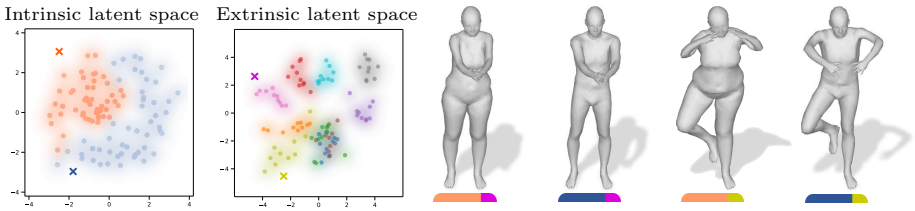


Fig. 6. *Plots on the left*: Planar embedding of the intrinsic and extrinsic parts of the latent codes from FAUST. Colors identify gender (left) and pose (ten different poses; right). We observe cohesive clusters in either case, suggesting that the encoder has generalized the projection onto each factor. The four small crosses are random samples. *Right*: Decoded shapes from the four combinations of the random samples; the specific combinations are illustrated by compatible colors between the crosses and the bars below each shape.

As an additional qualitative experiment, in Figure 4 we show the decoded shapes in-between two keyframes of a 4D sequence from DFAUST. We remark that none of the intermediate shapes were seen at training time, nor was any similar-looking shape present in the training set. We then compare our reconstructed sequence with the original sequence of real-world scans. The purpose of this experiment is to show that our geometric priors are essential for the generation of realistic motion; apart from a perceptual evaluation, any quantitative comparison here would not be meaningful – there is not a unique "true" way to transition between two given poses.

4.3 Disentanglement

Our second set of experiments is aimed at demonstrating the effectiveness of our geometric priors for the disentanglement of intrinsic from extrinsic information. We illustrate this in different ways.

In Figure 6, we show disentanglement for a generator trained on the FAUST dataset. For visualization purposes, for each vector $\mathbf{z} := (\mathbf{z}^{\text{int}}|\mathbf{z}^{\text{ext}})$ in the latent space (here comprising both training and test shapes), we embed the $\mathbf{z}^{\text{int}}$ and $\mathbf{z}^{\text{ext}}$ parts *separately* onto the plane (via multidimensional scaling), and attribute different colors to different gender and poses. We then randomly sample two new $\mathbf{z}^{\text{int}}$ and two new $\mathbf{z}^{\text{ext}}$, and compose them into four latent codes by taking all the combinations. The figure illustrates the four decoded shapes.

In Figure 7 we show the simultaneous action of disentanglement and interpolation. Given a source and a target shape, we show the interpolation of pose while fixing the style, and the interpolation of style while fixing the pose. We do so with different combinations of source and target. In all cases, our generative model is able to synthesize realistic shapes with the correct semantics, suggesting high potential in style and pose transfer applications.

As we did with the case of interpolation, we also provide a notion of *disentanglement error*, defined as follows. Given shapes $\mathbf{X}_i$ and $\mathbf{X}_j$ with latent

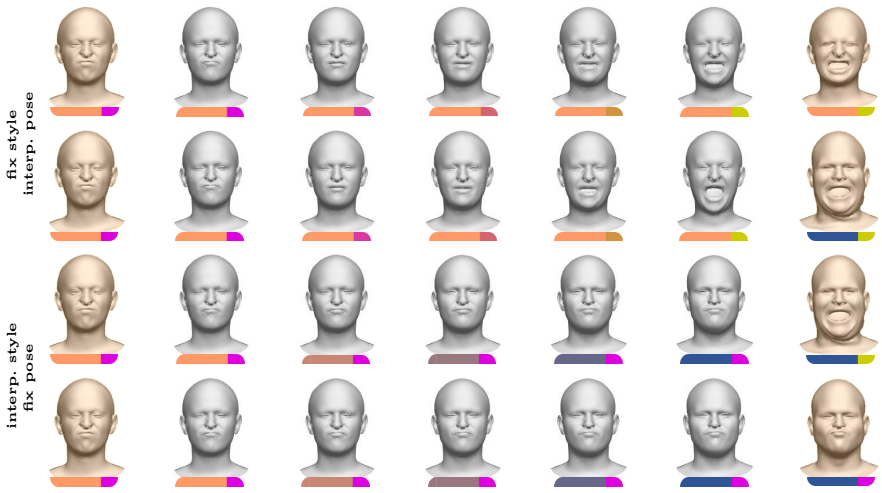


Fig. 7. Disentanglement + interpolation examples on the COMA dataset; the source shape is always the same. Each row presents a different scenario, with interpolation happening left-to-right. Please refer to the color code below each shape as a visual aid; for example, for the first column we have (**style**|**pose**).

codes $(\mathbf{z}_i^{\text{int}}|\mathbf{z}_i^{\text{ext}})$ and $(\mathbf{z}_j^{\text{int}}|\mathbf{z}_j^{\text{ext}})$, we swap $\mathbf{z}_i^{\text{ext}}$ with $\mathbf{z}_j^{\text{ext}}$ and then measure the average point-to-point distance between $\text{dec}(\mathbf{z}_i^{\text{int}}|\mathbf{z}_j^{\text{ext}})$ and the corresponding ground-truth shape from the dataset. In Table 1 (last 3 columns) we report the disentanglement error on all 4 datasets, together with the ablation study.

Finally, in Figure 8 we show a qualitative comparison with the recent state-of-the-art method [2] (using public code provided by the authors), which uses Laplacian eigenvalues as a prior to drive the disentanglement, together with multiple other de-correlation terms. Similarly to other approaches like [24,43], the quality of the interpolation of [2] mostly depends on the smoothness properties of the VAE, on the complexity of the deep net, or on the availability of vast training data. For this comparison, both generative models were trained on the same 80 FAUST shapes.

5 Conclusions

We introduced a new deep generative model for deformable 3D shapes. Our model is based on the intuition that by directly connecting the Euclidean distortion of latent codes to the metric distortion of the decoded shapes, one gets a powerful regularizer that induces a well-behaved structure on the latent space. Our idea finds a theoretical interpretation in modeling deformations with bounded metric distortion as sequences of approximate isometries. Under the manifold hypothesis, our metric preservation priors explicitly promote a flattening of the true data manifold onto a lower-dimensional Euclidean representation. We demonstrated how having access to the metric of the decoded shapes during training enables

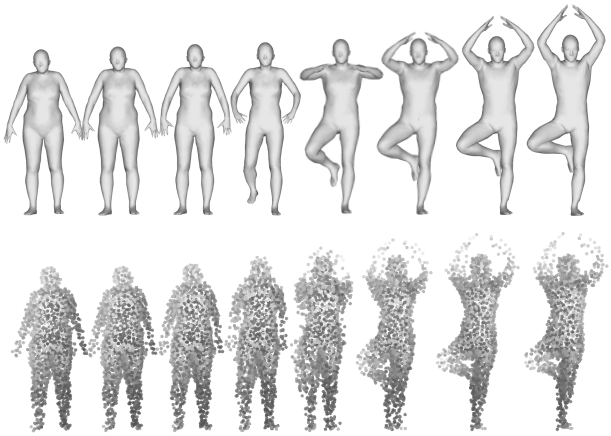


Fig. 8. Comparison of our method (top row) with the state-of-the-art method of [2] (bottom row). Both generative models are trained on the same data. The leftmost and rightmost shapes are from the training set, while the intermediate shapes are decodings of a linear sequence in the latent space. Observe that source and target are *not* isometric; according to our continuous interpretation of Sec. 3.2, our trained model decomposes the non-isometric deformation into a sequence of approximate isometries.

high-quality synthesis of novel samples, with practical implications in tasks of content creation and style transfer.

Perhaps the main **limitation** of our method, which we share with other geometric deep learning approaches, lies in the requirement of labeled pointwise correspondences between the training shapes. These can be hard to obtain in certain settings, for example, when dealing with shapes from the same semantic class but with high intra-class variability. Few interesting directions of future work may consist in a self-supervised variant of our model, where dense correspondences are not needed for the training, but are estimated during the learning process or in the exploitation of spectral properties of the reconstructed shape, that has been shown [10,35] to contain important information of the embedding geometry.

Finally, while in this paper we showed that even a simple prior such as metric distortion can have a significant effect, we foresee that bringing techniques from the areas of shape optimization and analysis closer to deep generative models will enable a fruitful line of stimulating research.

Acknowledgments. LC, AN and ER are supported by the ERC Starting Grant No. 802554 (SPECGEO) and the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University. OH and RK are supported by the Israel Ministry of Science and Technology grant number 3-14719, the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel Cyber Directorate.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning. pp. 40–49 (2018)
2. Aumentado-Armstrong, T., Tsogkas, S., Jepson, A., Dickinson, S.: Geometric disentanglement for generative latent shape models. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8181–8190 (2019)
3. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway, NJ, USA (Jun 2014)
4. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)
5. Boscaini, D., Eynard, D., Kourounis, D., Bronstein, M.M.: Shape-from-operator: Recovering shapes from intrinsic operators. In: Computer Graphics Forum. vol. 34, pp. 265–274. Wiley Online Library (2015)
6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer Science & Business Media (2008)
7. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
8. Chaudhuri, S., Ritchie, D., Xu, K., Zhang, H.R.: Learning Generative Models of 3D Structures. In: Jakob, W., Puppo, E. (eds.) Eurographics 2019 - Tutorials. The Eurographics Association (2019). https://doi.org/10.2312/egt.20191038
9. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. arXiv preprint arXiv:1911.06971 (2019)
10. Cosmo, L., Panine, M., Rampini, A., Ovsjanikov, M., Bronstein, M.M., Rodolà, E.: Isospectralization, or how to hear shape, style, and correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7529–7538 (2019)
11. Cosmo, L., Rodola, E., Albarelli, A., Mémoli, F., Cremers, D.: Consistent partial matching of shape collections via sparse modeling. In: Computer Graphics Forum. vol. 36, pp. 209–221. Wiley Online Library (2017)
12. Cosmo, L., Rodola, E., Masci, J., Torsello, A., Bronstein, M.M.: Matching deformable objects in clutter. In: 2016 Fourth International Conference on 3D Vision (3DV). pp. 1–10. IEEE (2016)
13. Crane, K., Weischedel, C., Wardetzky, M.: Geodesics in heat: A new approach to computing distance based on heat flow. ACM Trans. Graph. **32**(5) (Oct 2013)
14. Devir, Y.S., Rosman, G., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: On reconstruction of non-rigid shapes with intrinsic regularization. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. pp. 272–279. IEEE (2009)
15. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 209–216 (1997)
16. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099 (2020)

17. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 230–246 (2018)
18. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018)
19. Halimi, O., Litany, O., Rodolà, E., Bronstein, A.M., Kimmel, R.: Unsupervised learning of dense shape correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4370–4379 (2019)
20. Heeren, B., Rumpf, M., Schröder, P., Wardetzky, M., Wirth, B.: Exploring the geometry of the space of shells. Computer Graphics Forum **33**(5), 247–256 (2014)
21. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proceedings of the national academy of Sciences **95**(15), 8431–8435 (1998)
22. Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L.: Grass: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017)
23. Li, M., Patil, A.G., Xu, K., Chaudhuri, S., Khan, O., Shamir, A., Tu, C., Chen, B., Cohen-Or, D., Zhang, H.: Grains: Generative recursive autoencoders for indoor scenes. ACM Transactions on Graphics (TOG) **38**(2), 1–16 (2019)
24. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1886–1895 (2018)
25. Litany, O., Remez, T., Rodolà, E., Bronstein, A., Bronstein, M.: Deep functional maps: Structured prediction for dense shape correspondence. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5659–5667 (2017)
26. Liu, Z., Freeman, W.T., Tenenbaum, J.B., Wu, J.: Physical primitive decomposition. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
27. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proc. CVPR. pp. 4460–4470 (2019)
28. de Miguel, J., Villafane, M.E., Piskorec, L., Sancho-Caparrini, F.: Deep form finding – using variational autoencoders for deep form finding of structural typologies. In: Architecture in the Age of the 4th Industrial Revolution - Proceedings of the 37th eCAADe and 23rd SIGraDi Conference. pp. 71–80 (2019)
29. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.: Structurenet: hierarchical graph networks for 3d shape generation. ACM Transactions on Graphics **38**(6) (2019)
30. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 909–918 (2019)
31. Nash, C., Ganin, Y., Eslami, S., Battaglia, P.W.: Polygen: An autoregressive generative model of 3d meshes. arXiv preprint arXiv:2002.10880 (2020)
32. Nash, C., Williams, C.K.: The shape variational autoencoder: A deep generative model of part-segmented 3d objects. Computer Graphics Forum **36**(5), 1–12 (2017)
33. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)

34. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
35. Rampini, A., Tallini, I., Ovsjanikov, M., Bronstein, A.M., Rodolà, E.: Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In: 2019 International Conference on 3D Vision (3DV). pp. 37–46. IEEE (2019)
36. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 704–720 (2018)
37. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 704–720 (2018)
38. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. Computer-Aided Design **38**(4), 342–366 (2006)
39. Rodola, E., Bronstein, A.M., Albarelli, A., Bergamasco, F., Torsello, A.: A game-theoretic approach to deformable shape matching. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 182–189. IEEE (2012)
40. Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D.: Partial functional correspondence. Computer Graphics Forum **36**(1), 222–236 (2017)
41. Shamai, G., Kimmel, R.: Geodesic distance descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6410–6418 (2017)
42. Shu, D.W., Park, S.W., Kwon, J.: 3d point cloud generative adversarial network based on tree structured graph convolutions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3859–3868 (2019)
43. Tan, Q., Gao, L., Lai, Y.K., Xia, S.: Variational autoencoders for deforming 3d mesh models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5841–5850 (2018)
44. Tan, Q., Pan, Z., Gao, L., Manocha, D.: Realtime simulation of thin-shell deformable materials using cnn-based mesh embedding. IEEE Robotics and Automation Letters **5**(2), 2325–2332 (2020)
45. Verma, N., Boyer, E., Verbeek, J.: Feastnet: Feature-steered graph convolutions for 3d shape analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2598–2606 (2018)
46. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in neural information processing systems. pp. 82–90 (2016)