

GenSP: Consistent Spherical Parameterization via Learning Shape Generative Models

Sai Karthikey Pentapati^{1*}, Shashank Gupta^{1*}, Rajesh Sureddi¹, Yuezhi Yang¹,
Alan C. Bovik², and Qixing Huang¹

¹ The University of Texas at Austin
{ps.karthik,shashank.gupta}@utexas.edu

² University of Colorado Boulder

Abstract. We introduce GenSP, a data-driven framework that learns consistent spherical parameterizations across a collection of genus-0 shapes. Instead of optimizing the parameterization of each shape independently, our method learns a neural generative model that predicts a continuous mapping from the unit sphere to shapes in a dataset. Under this formulation, spherical parameterizations are obtained through the inverse mappings of the learned generator, which encourages similar shapes to share consistent parameterizations. To make this formulation practical, we address several key challenges in learning such a generative model. First, we introduce a continuous neural deformation model that predicts surface points from sphere coordinates and latent shape codes, avoiding discretization artifacts common in mesh-based formulations. Second, we augment the training space with intermediate shapes that bridge the sphere and input shapes, allowing the model to learn meaningful deformations across a heterogeneous shape collection. Third, we compute reliable initial correspondences by propagating mappings along a spanning tree of training shapes in the latent space. Experiments on the ShapeNet [10] dataset demonstrate that our approach significantly reduces geometric distortion and improves cross-shape consistency compared with state-of-the-art spherical parameterization methods.

Keywords: Spherical Parameterization, Neural Geometry Processing, Generative Models.

1 Introduction

Computing spherical parameterizations of genus-0 surfaces is a fundamental problem in geometry processing and visual computing [27, 44, 52, 62]. Given a genus-0 surface, spherical parameterization computes a bijective mapping between the surface and the unit sphere, providing a canonical domain for applications such as texture transfer, shape correspondence, and geometric analysis. Compared with planar parameterization methods that map surfaces to the UV plane, spherical parameterization does not require cutting the surface. Such cuts introduce discrete optimization decisions that are difficult to solve and often lead

* Equal contribution.

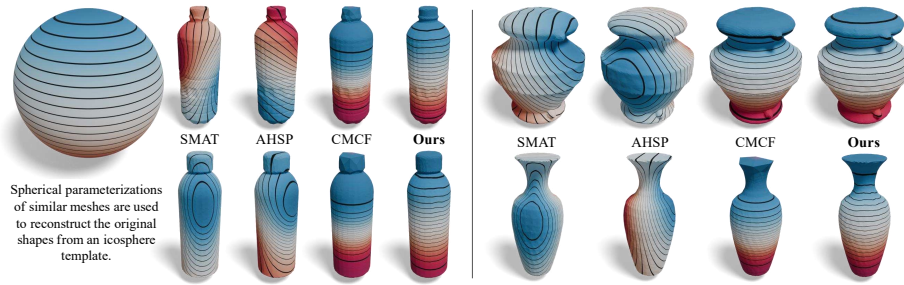


Fig. 1: Comparison of spherical parameterizations with existing approaches. A texture defined on an icosphere is mapped onto each shape through the spherical parameterization to visualize the consistency of the mappings across shapes. Existing approaches either produce inconsistent parameterizations across shapes (SMAT [50], AHSP [28]) or exhibit high geometric distortion in thin or high-curvature regions (CMCF [33]). In contrast, GenSP produces consistent, feature-preserving and accurate parameterizations across the shape collection.

to additional distortion in the parameterization. Most existing approaches [22, 47] compute spherical parameterizations for each surface independently. However, a fundamental challenge is that there exist infinitely many spherical parameterizations for a genus-0 surface. Even when the mapping is restricted to be conformal, the space of valid parameterizations forms a Möbius group [37]. As a result, two geometrically similar shapes may obtain very different parameterizations when using existing methods, as shown in Figure 1.

In this paper, we introduce a data-driven approach for computing consistent spherical parameterizations across a collection of genus-0 shapes. Our approach learns a generative model that continuously deforms the unit sphere to each shape in the dataset. Under this formulation, spherical parameterizations arise naturally as the inverse mappings of the learned generator. To ensure consistent parameterizations across shapes, we formulate common objectives of spherical parameterization as regularization losses on the generative model. These losses encourage the generator to produce smooth deformations, which lead to similar spherical parameterizations among neighboring shapes in the dataset. We further introduce a deformation regularization term that effectively minimizes geometric distortion in the resulting parameterizations.

Although the conceptual idea of this approach is straightforward, there are three fundamental challenges. The first challenge is the representation of the generative model. The straightforward approach of learning a mesh generative model is not ideal due to large variations in conformal factors of the underlying parameterization, leading to unevenly large and tiny triangles during parameterization. To address this issue, we introduce a continuous neural deformation model that maps points on a sphere to corresponding surface points on the target shape.

The second challenge is that, to date, most shape generative models with fixed topology have only been applied to relatively limited shape spaces, e.g.,

humans/animals [39, 72]. In many cases, the underlying sub-space does not contain the sphere, making it difficult to produce smooth latent space trajectories from a sphere to the shapes in a dataset. To address this issue, we propose augmenting the shape space with additional shapes that bridge the sphere and the input shapes. This is achieved by running existing flow-based models [33] to deform a shape into a sphere and adding suitable intermediate shapes as training shapes to learn the generative model. The training dataset, augmented with intermediate shape samples, samples the distribution of a larger genus-0 shape space that contains both the sphere and the input shapes in a more natural manner.

The final challenge is to compute the correspondences between the sphere and each input shape to initialize our neural deformation model, since learning this generator from scratch is prone to getting trapped in local minima. We find that doing so directly is quite difficult, particularly when the deformation between the sphere and an input shape is large. We address this by finding a path of training (input and augmented) shapes that gradually deforms the sphere into each input shape. To this end, we learn a point cloud encoder and an implicit decoder using an auto-encoder loss with additional regularization losses. The first loss controls the distribution of latent codes, wherein the latent code of the sphere is forced to lie at the origin of the latent space, and the latent codes of two shapes are close if and only if they are similar. The second loss regularizes the shape space defined by the shape decoder, so that intermediate shapes between two adjacent input shapes are deformation preserving, prioritizing high-quality inter-shape correspondences. We then construct a spanning tree among the latent codes of the training shapes, with the latent code of the sphere at the root. Then, obtaining correspondences between a complex training shape and the sphere is done by composing correspondences between shapes associated with each edge along the shortest path between the given shape and the sphere. These point-to-point correspondences are consistent between similar training shapes and are used to initialize the neural-deformation based mesh generative model. Finally, we regularize the model to minimize the isometric distortion in the generated mappings.

We tested our approach on the Genus-0 subset of the ShapeNet [10] dataset. The experimental results show that our approach yields parameterizations with significantly improved inter-shape consistency compared to state-of-the-art spherical parameterizations while simultaneously having low geometric distortions.

2 Related Work

2.1 Spherical Mesh Parameterization

Existing spherical parameterizations have evolved through four generations. The first generation reduces the problem to the planar case. [24] first cut out one triangle to serve as a boundary, then parameterize the resulting open mesh over the unit triangle using any planar parameterization method, and finally use the inverse stereographic projection to map the plane to the sphere. [11] is a modern

variant of this approach. The main problem with these methods is the severe distortion of large regions. The key idea of the second generation [16, 28, 47, 50, 51] is to perform mesh simplification into a tetrahedron, map the tetrahedron to the sphere, and revert the procedure to compute the spherical parameterization of each intermediate mesh in the simplification process. Although this method is efficient, it is difficult to optimize the parameterization due to its greedy nature. Additionally, the bottom-to-top approach makes it challenging to preserve the consistency between parameterizations of two similar shapes. The third generation consists of various global optimization methods. The seminal work of [22] connects discrete spherical parameterizations to spectral graph theory and computes spherical parameterization via a generalized Laplacian. However, this formulation is non-convex and has many local minima. Several other papers [2, 36, 61] solve various global optimization relaxations, but all have issues of local minima. The fourth generation models spherical parameterizations as geometric flows [6, 14, 33], which gradually deform a given shape into the sphere. These approaches enjoy theoretical guarantees and improved consistency between parameterizations. However, these methods often yield extremely poor isometry factors (see Figure 1). Our approach can be considered a variant of the flow-based approaches, in which the flow from each shape to the sphere is not determined by a PDE but by data, i.e., a path of intermediate shapes. This approach greatly enhances the stability of spherical parameterization.

2.2 Shape Correspondences

Spherical parameterization can be formulated as a shape matching problem, which computes dense correspondences between vertices of a triangular mesh and the sphere. However, existing shape-matching approaches do not apply to the problem of matching a sphere with an arbitrary genus-0 shape. A breakthrough result in this domain is the functional map representation [25, 41]. Due to their simple algebraic representation, functional maps have been widely adopted in the deep learning era [3, 17, 18, 38, 45, 71]. Several other approaches [20, 21, 23, 59] also introduced deep neural networks for shape matching. Compared to non-deep-learning-based approaches, learning-based methods relax the constraint that the input shapes should be similar to each other. However, they are still not applicable to matching a sphere shape with an arbitrary genus-0 shape due to a lack of identifiable keypoints on the sphere.

One promising approach to extend the range of input shapes is to perform joint shape matching [31, 34, 40], where the map between two shapes is the composition of pairwise maps along a path of similar shape pairs. Joint shape matching typically enforces the cycle-consistency constraint, which is effective for pruning incorrect maps computed between pairs of shapes. Several papers have studied how to enforce this constraint using convex [29], non-convex [69], and spectral techniques [42, 53]. The most relevant to our approach is GenCorres [63], which learns mesh generative models to establish consistent correspondences. However, learning a mesh generative model requires high-quality initial correspondences, which are difficult to obtain. Moreover, a mesh generative model is not applicable

in our setting due to large variations of conformal factors between the underlying shapes. This paper addresses these two issues by studying the problems of learning a shape generative model from a heterogeneous shape collection of diverse genus-0 shapes and encoding the shape generative model as neural fields.

2.3 Shape Generative Models

GenSP is based on recent advances in generative models of deformable shapes. Early work on this topic employs explicit shape representations, such as distance matrices [13] between all pairs of points and triangular meshes [8, 49, 68, 70]. These approaches require consistently meshed training shapes as input for learning the generative model. The idea of alternating between computing consistent correspondences and learning an explicit generative model requires good initializations, which this paper addresses.

Several other papers studied learning an implicit shape generative model that bypasses the requirement of correspondences among the input data. Motivated by DeepSDF [43], SALD [4] learns an MLP to encode an implicit shape generative model for deformable shapes, e.g., humans and animals. This approach does not apply to training shapes with large pose variations. Several papers introduced regularization losses for implicit generative models. Among them, the killing vector field approach [54], which provides weak regularizations, has been adopted in several papers [5, 12, 48, 60]. GenCorres [63] and GenAnalysis [65] introduce a stronger regularization, which is based on discretizing an implicit surface generator into a mesh generator locally by computing dense correspondences of mesh vertices on adjacent implicit surfaces. This allows us to employ prior work that enforces deformation losses on meshes [30, 64] to regularize the implicit generator. This approach is flexible because it allows us to employ a reduced deformation model that enables stronger and more efficient regularizations, which we will explore in this paper.

2.4 Neural Deformation Models

Our approach uses classical shape deformation models such as as-rigid-as-possible [55] and as-conformal-as-possible [66]. Unlike the standard formulations that do not admit a closed-form expression, we employ the linear approximation [32], which applies to shapes that are close to each other and admits a closed-form expression.

The application of this deformation model to regularize generative models has been studied in the literature, e.g., ARAPReg [30], GeoLatent [64] and GenCorres [63]. These methods introduce regularization losses to minimize the deformation between adjacent synthetic shapes. Although our approach also follows this paradigm, the key difference is that we normalize the deformation using the absolute shape displacements, addressing the issue of irregular distributions of shape latent codes.

Several other papers have studied neural deformation models, e.g., NJField [1], LLDPD [57], and TutteNet [56]. A common scheme among these models is to employ neural networks to predict the deformation field, e.g., displacements of deformation nodes and deformation gradients. The focus is on shapes that undergo

large deformations. However, neural shape deformation models have generalization issues. In contrast, our deformation model is between adjacent shapes, and deformations between different shapes are computed using a generative model that implicitly composes deformations between adjacent shapes.

3 Problem Statement and Approach Overview

3.1 Problem Statement

The input consists of a collection of triangular meshes $\mathcal{S} = \{S_0, \dots, S_N\}$. Each mesh is closed and has genus-0. Here S_0 is a mesh of the unit sphere with n vertices. $S_1, \dots, S_N$ are actual training meshes. As a starting point, we assume that these meshes are roughly normalized in some shared ambient shape. However, we do not assume that they are consistently meshed.

Our goal is to find, for each shape S_i , a mapping $m_i : S_i \rightarrow S_0$ to the unit sphere S_0 . m_i has the following properties: 1) m_i is bijective, i.e., the image $m_i(S_i)$ covers S_0 and does not have self-intersections. 2) m_i has minimal isometric deformations. 3) For similar shapes S_i and S_j , the mappings m_i and m_j are also similar.

3.2 Approach Overview

Our approach learns a neural shape generative model of the inverse mapping from the unit sphere to each shape in the genus-0 shape space. Denote $\mathcal{Z} \cong \mathbb{R}^d$ as the latent space of genus-0 shapes. The generative model is denoted as $\mathbf{g}^\theta(\mathbf{z}, \cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, where $\mathbf{g}^\theta(\mathbf{z}, \mathbf{p})$ is the corresponding surface point of a point on a sphere $\mathbf{p} \in \mathbb{S}^2$. We consider four objectives:

1. The latent code of S_0 is at the origin, i.e., $\mathbf{g}^\theta(\mathbf{0}, \mathbf{p}) \in S_0, \forall \mathbf{p} \in \mathbb{S}^2$.
2. $\mathbf{g}^\theta(\mathbf{z}_i, \mathbf{p}) \in S_i, \forall \mathbf{p} \in \mathbb{S}^2$, where $\mathbf{z}_i$ is the latent code of shape S_i .
3. The Jacobian $\frac{\partial \mathbf{g}^\theta(\mathbf{z}, \mathbf{p})}{\partial \mathbf{z}}, \forall \mathbf{p} \in \mathbb{S}^2$ follows a regularized as-conformal-as possible deformation prior.
4. The geometric distance between two shapes S_i and S_j correlates with the Euclidean distance between $\mathbf{z}_i$ and $\mathbf{z}_j$.

There are two fundamental challenges to learning the generative model described above. First, the sphere S_0 is usually not in the underlying shape space of $\mathcal{S} = \{S_1, \dots, S_N\}$, making it very difficult to learn a shape generative model that fits $\mathcal{S}$ and provides meaningful interpolations between S_0 and each S_i . To address this challenge, our key idea is to augment the input shapes with additional shapes $S_{N+1}, \dots, S_{\bar{N}}$ by deforming each input shape S_i into the sphere S_0 using an off-the-shelf method to obtain the augmented dataset $\bar{\mathcal{S}}$. Second, we need consistent correspondences between the sphere and the training shapes to define a meaningful data term to learn the shape generative model. Without correspondences, minimizing the Chamfer distance or Earth-Mover distance between the synthetic mesh and each training shape can easily result in getting

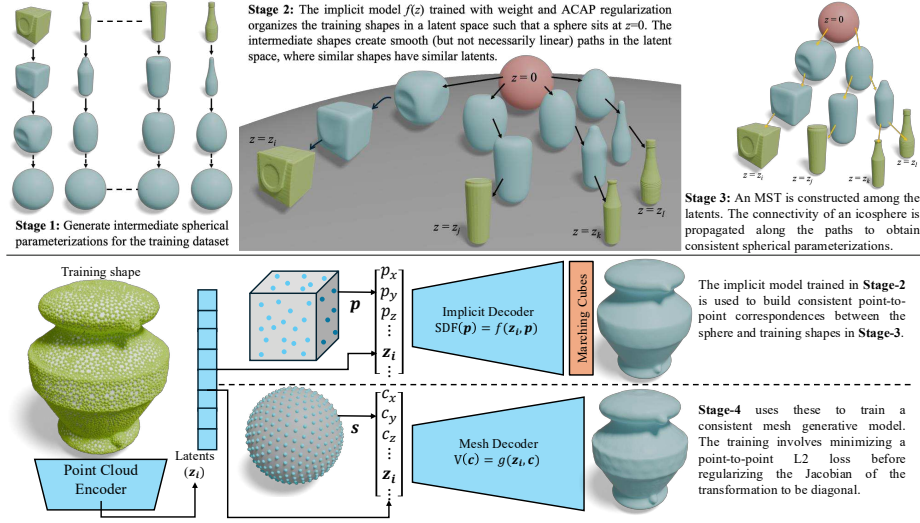


Fig. 2: An overview of the four stages of GenSP.

trapped in local minima. To address this challenge, we propose to introduce an implicit generative model $f^\phi : \mathbb{R}^3 \times \mathbb{R}^d \rightarrow \mathbb{R}$ to facilitate the computation of initial correspondences for learning this shape generative model.

As shown in Figure 2, GenSP proceeds in four stages. The first stage applies an off-the-shelf flow method [33] to deform each shape S_i into the sphere. These intermediate shapes are used to augment the training shapes.

The second stage learns an implicit shape generator $f^\phi : \mathbb{R}^d \times \mathbb{R}^3 \rightarrow \mathbb{R}$ together with a point cloud encoder $e^\gamma : \mathcal{S} \rightarrow \mathcal{Z}$ that maps a shape into the latent space using an auto-encoder loss. We add an additional regularization loss on e^γ to enforce that similar shapes have close-by latent codes. The decoder f^ϕ will be used to generate intermediate shapes between pairs of adjacent training shapes to propagate inter-shape correspondences. Therefore, we introduce a deformation regularization loss in f^ϕ to improve the quality of these intermediate shapes.

Based on the learned implicit shape generator, the third stage constructs a spanning tree among the training shapes where S_0 is the root. Starting from the root, we apply template-based shape registration using intermediate shapes generated by the implicit generator f^ϕ to compute dense correspondences between S_0 and each training shape. The resulting correspondences are used to initialize the shape generator g^θ , which is trained using an auto-encoder loss.

The last stage fine-tunes the shape generator g^θ and the encoder together e^γ together, where we introduce a deformation regularization term on the shape generator g^θ to optimize the final spherical mesh parameterizations.

4 Approach

4.1 Stage I: Data Augmentation

The first stage augments the training data $S_0, \dots, S_N$ with suitable genus-0 shapes $S_{N+1}, \dots, S_{\bar{N}}$ that bridge S_0 and $S_1, \dots, S_N$, as shown in Figure 2. We achieve this goal by using an off-the-shelf flow-based method to deform each input shape S_i into the sphere shape S_0 and sample intermediate shapes. We chose CMCF [33], which has a stable open-source implementation. CMCF has the step-size as a critical hyper-parameter, and no fixed value is optimal among all input shapes. Therefore, we need to define a scoring function for intermediate shapes to decide which intermediate shapes to use for augmentation.

For a given hyper-parameter m and one shape S_i , denote by $S_i^m(t)$, $0 \leq t \leq 1$ the result of m such that $S_i^m(0) = S_i$, and $S_i^m(1)$ converges to the sphere or becomes degenerate. We define the scoring function as

$$s(m, S_i) = d^2(S_0, S_i^m(1)) + \eta \int_0^1 e_{\text{deform}}(S_i^m(t), S_i) dt \quad (1)$$

where the first term measures the squared Hausdorff distance between S_0 and $S_i^m(1)$, and the second term integrates the deformation between $S_i^m(t)$ and S_i . For two meshes $\mathbf{p} \in \mathbb{R}^{3n}$ and $\mathbf{q} \in \mathbb{R}^{3n}$ each with n vertices that share the same topology, their deformation is defined as

$$e_{\text{deform}}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n \min_{R_i \in \text{SO}(3)} \sum_{j \in \mathcal{N}_i} \|R_i(\mathbf{p}_i - \mathbf{p}_j) - (\mathbf{q}_i - \mathbf{q}_j)\|^2 \quad (2)$$

where $\mathcal{N}_i$ is the 1-ring neighborhood of the vertex i .

For each S_i , we pick the hyper-parameter m_i^* that has the smallest $s(m_i^*, S_i)$. When $s(m_i^*, S_i) < 2\delta$, where δ is the median of $s(m_i^*, S_i)$ among all input shapes, we add $S_i^{m_i^*}(t)$ to the training shapes, where t is randomly sampled in $(0, 1)$. The total number of training shapes $\bar{N} = 24N$, i.e., we place 23 samples in t .

4.2 Stage II: Autoencoder for Learning the Shape Space

The second stage learns a point cloud encoder e^γ and an implicit generator f^ϕ by combining an auto-encoder data loss and two regularization losses on e^γ and f^ϕ , respectively. The data loss fits f^ϕ to the training shapes:

$$l_{\text{data}}(\phi, \gamma) := \sum_{i=0}^{\bar{N}} \sum_{(\mathbf{p}, d) \in \mathcal{D}(S_i)} (f^\phi(\mathbf{p}, e^\gamma(S_i)) - d)^2 \quad (3)$$

where $\mathcal{D}(S_i)$ collects signed-distance function (SDF) samples (cf. [4]) on-and-off the surface of S_i to learn the implicit decoder.

The first regularization loss is enforced on e^γ to ensure that similar training shapes remain close to each other in the latent space. This is important when

using latent codes to construct a spanning tree to initialize the correspondences between S_0 and each training shape S_i . Here, we directly regularize the magnitude of γ :

$$l_{\text{reg,I}}(\gamma) = \rho(\|\gamma\|) \quad (4)$$

where ρ is chosen as the L^1 norm.

The second regularization loss is enforced on the implicit shape generator so that the intermediate shapes that interpolate adjacent synthetic shapes are deformation-preserving. As we will linearly interpolate latent codes of adjacent synthetic shapes to generate intermediate shapes to propagate correspondences, we collect these interpolated latent codes into an empirical distribution p_{sync} and define the regularization loss as

$$l_{\text{reg,II}}(\phi) = \mathbb{E}_{\mathbf{z} \sim p_{\text{sync}}} r(f^\phi(\cdot, \mathbf{z})). \quad (5)$$

We adopt the approach in GenCorres [63] to define $r(f^\phi(\cdot, \mathbf{z}))$ but make two significant modifications. First, we define $r(f^\phi(\cdot, \mathbf{z}))$ using a reduced deformation model, which greatly increases the efficiency of training while making the effectiveness of regularization more prominent. Specifically, consider a mesh discretization of $f^\phi(\cdot, \mathbf{z})$ with m vertices $\mathbf{p}^\phi(\mathbf{z}) \in \mathbb{R}^{3m}$. Consider $l \ll m$ deformation nodes obtained by farthest point sampling ($m = 2000$ and $l = 200$ in our experiments). With $\mathcal{N}_i$ we collect adjacent vertices of node i , where adjacent $\mathcal{N}_i$ overlap and $\{\mathcal{N}_i\}$ cover all mesh vertices. With this setup, we parameterize $\mathbf{p}^\phi(\mathbf{z} + \epsilon \mathbf{v}) \approx \mathbf{p}^\phi(\mathbf{z}) + \epsilon \mathbf{d}(\mathbf{z}, \mathbf{v})$ in the neighborhood of $\mathbf{z}$ where $\mathbf{d}(\mathbf{z}, \mathbf{v}) :=$

$$\begin{aligned} \underset{\mathbf{d}}{\text{argmin}} \min_{s_i, \mathbf{c}_i} \sum_{i=1}^l \left(\sum_{j \in \mathcal{N}_i} \|A_i(\mathbf{p}_i^\phi(\mathbf{z}) - \mathbf{p}_j^\phi(\mathbf{z})) - (\mathbf{d}_i - \mathbf{d}_j)\|^2 + \eta s_i^2 \right) \\ \text{s.t.} \quad \frac{\partial f^\phi}{\partial \mathbf{x}}(\mathbf{p}_j^\phi(\mathbf{z}), \mathbf{z}) \mathbf{d}_j + \frac{\partial f^\phi}{\partial \mathbf{z}}(\mathbf{p}_j^\phi(\mathbf{z}), \mathbf{z}) \mathbf{v} = 0, \quad 1 \leq j \leq m \end{aligned} \quad (6)$$

where

$$A_i = s_i I_3 + \mathbf{c}_i \times I_3.$$

Through some derivations (see the supp. material), we can rewrite Eq. (6) as

$$\mathbf{d}^v := \underset{\mathbf{d}}{\text{argmin}} \quad \mathbf{d}^T L^\phi(\mathbf{z}) \mathbf{d} \quad \text{s.t.} \quad C^\phi(\mathbf{z}) \mathbf{d} + F^\phi(\mathbf{z}) \mathbf{v} = 0. \quad (7)$$

This leads to an explicit expression of

$$\mathbf{d}^v = G^\phi(\mathbf{z}) \mathbf{v}, \quad G^\phi(\mathbf{z}) = -(I, 0) \begin{pmatrix} L^\phi(\mathbf{z}) & C^\phi(\mathbf{z})^T \\ C^\phi(\mathbf{z}) & 0 \end{pmatrix}^\dagger \begin{pmatrix} 0 \\ F^\phi(\mathbf{z}) \end{pmatrix} \mathbf{v}.$$

Second, unlike merely minimizing $\mathbf{d}^{vT} L^\phi(\mathbf{z}) \mathbf{d}^v$, we define the regularization loss at $\mathbf{z}$ by integrating the ratio $\frac{\mathbf{d}^{vT} L^\phi(\mathbf{z}) \mathbf{d}^v}{\mathbf{d}^{vT} \mathbf{d}^v}$, which measures the relative deformation with respect to the absolute displacement. This becomes important as the

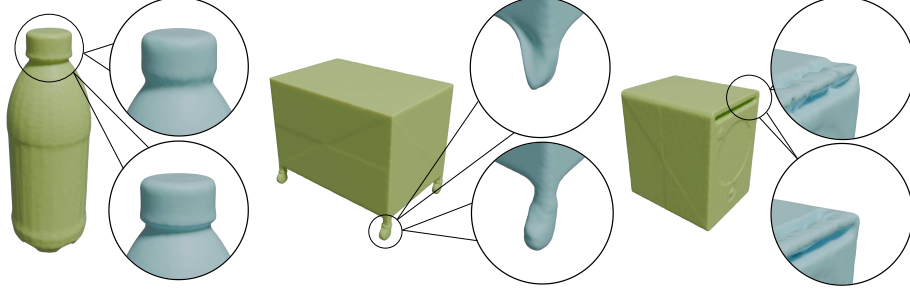


Fig. 3: (Top) If we deform S_0 to align S_i through intermediate shapes defined by $f^\phi(\cdot, \mathbf{z})$ the alignment is poor, as some intermediate shapes have large distortions. (Bottom) When we deform S_0 to align S_i through intermediate shapes between adjacent training shapes on a spanning tree rooted at S_0 , the alignment is more accurate as intermediate shapes are much more meaningful.

distribution of shapes in the AE latent space becomes irregular. Specifically, we have

$$r(f^\phi(\cdot, \mathbf{z})) = \int_{\mathbf{v} \in \mathcal{S}^d} \frac{\mathbf{d}^{\mathbf{v}^T} L^\phi(\mathbf{z}) \mathbf{d}^{\mathbf{v}}}{\|\mathbf{d}^{\mathbf{v}}\|^2} d\mathbf{v}. \quad (8)$$

Combining Eqs.(3), (4), and (5), the total training loss for stage II is

$$\min_{\gamma, \phi} l_{\text{data}}(\phi, \gamma) + \lambda l_{\text{reg}, I}(\gamma) + \mu l_{\text{reg}, II}(\phi). \quad (9)$$

To ensure that the latent code of S_0 is at the origin of the latent space, we sample S_0 once every batch and penalize the L^2 -norm of its predicted latent.

4.3 Stage III: Shape Generator Initialization

The goal of the third stage is to compute the initial dense correspondences between S_0 and each training shape S_i , which will be used to initialize the shape generator $\mathbf{g}^\theta$. A naive approach is to use $f^\phi(\cdot, \mathbf{z})$ to generate intermediate shapes between S_0 and S_i and to align S_0 with these intermediate shapes sequentially to obtain the correspondences with S_i . However, this approach does not work when S_i and S_0 are very different from each other, as many intermediate shapes present large deformations (see Figure 3(Top)).

Our idea is to introduce a more meaningful interpolation path from S_0 to S_i using the training data. Specifically, we build a minimum spanning tree among $\mathcal{S}$ where the root is S_0 . The weight of each edge (S_i, S_j) is given by the distance $\|e^\gamma(S_i) - e^\gamma(S_j)\|$ between S_i and S_j in the latent space. We also use $f^\phi(\cdot, \mathbf{z})$ to generate intermediate shapes between S_i and S_j to propagate the correspondences between S_0 and S_i to S_j . Correspondence propagation employs the standard non-rigid registration approach with an as-rigid-as possible deformation prior [58]. As shown in Figure 3(Bottom), this approach yields accurate correspondences for shapes S_i that present large variations from S_0 .

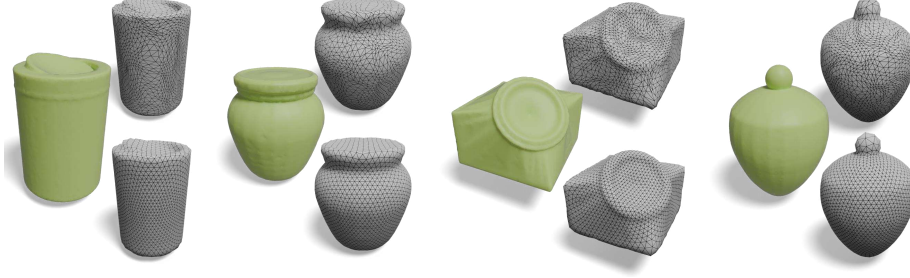


Fig. 4: Comparisons between (Top) initial correspondences between S_0 and each training shape and (Bottom) the correspondences obtained by optimized shape generator. We can see that the optimized correspondences are smoother and align the training data (green) better than the initial correspondences. This is attributed to the deformation loss on the mesh generator, which optimizes all correspondences together.

Let $\mathbf{p}^i \in \mathbb{R}^{3n}$ be the correspondences of the mesh vertices $\mathbf{c} \in \mathbb{R}^{3n}$ of S_0 on S_i . We initialize the mesh generator $\mathbf{g}^\theta(\mathbf{z}, \cdot)$ using the standard L^2 loss:

$$\min_{\theta} \sum_{i=1}^N \sum_{j=1}^n \|\mathbf{g}^\theta(e^\gamma(S_i), \mathbf{c}_j) - \mathbf{p}_j^i\|^2. \quad (10)$$

4.4 Stage IV: Shape Generator Fine-tuning

The fourth stage jointly refines the shape generator $\mathbf{g}^\theta$ and the encoder e^γ . The training loss consists of a modified data loss and a deformation regularization loss on $\mathbf{g}^\theta$. Its effect on improving the quality is shown in Figure 4.

The modified data loss is given by

$$l_{\text{data}}(\gamma, \theta) := \sum_{i=0}^N \int_{\mathbf{p} \in S^2} d^2(\mathbf{g}^\theta(e^\gamma(S_i), \mathbf{p}), S_i) \quad (11)$$

where $d^2(\mathbf{q}, S_i)$ is the squared distance between $\mathbf{q}$ and the triangular mesh S_i .

The deformation regularization loss models deformation priors on $\mathbf{g}^\theta(\mathbf{z}, \mathbf{p})$. As $\mathbf{g}^\theta(\mathbf{z}, \mathbf{p})$ is continuous in $\mathbf{p}$, we model the deformation regularization loss based on the Jacobian $J_{\mathbf{p}}^\theta(\mathbf{z}, \mathbf{p}) = \frac{\partial \mathbf{g}^\theta(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \in \mathbb{R}^{3 \times 2}$ which is computed in a local tangent frame at each sphere point $\mathbf{p}$, formed by two orthonormal vectors obtained by Gram-Schmidt against the radial direction at $\mathbf{p}$. The Jacobian itself is computed via PyTorch `autograd`. It is easy to see that the desired properties for $\mathbf{p}^\theta(\mathbf{z}, \mathbf{p})$ to be an isometric and conformal mapping are that $J_{\mathbf{p}}^\theta(\mathbf{z}, \mathbf{p})^T J_{\mathbf{p}}^\theta(\mathbf{z}, \mathbf{p})$ is the identity matrix I_2 and the scaled identity matrix, respectively. Therefore, we define the following regularization loss

$$l_{\text{def}}(\theta) = \mathbb{E}_{\mathbf{z} \sim p_{\text{sync}}, \mathbf{p} \in S^2} r(J_{\mathbf{p}}^\theta(\mathbf{z}, \mathbf{p})^T J_{\mathbf{p}}^\theta(\mathbf{z}, \mathbf{p})) \quad (12)$$

where

$$r(A) = \beta \|A - I_2\|^2 + \|A - \frac{\langle A, I_2 \rangle}{2} I_2\|^2.$$

We set $\beta = 0.01$ in all of our experiments.

The training loss at this stage is then given by

$$\min_{\theta} l_{\text{data}}(\gamma, \theta) + \bar{\mu} l_{\text{def}}(\theta).$$

5 Experimental Results

5.1 Experimental Setup

We collected a dataset of genus-0 shapes from the ShapeNet [10] dataset, totaling 1K shapes. These shapes cover diverse objects. We use 900 shapes for training and 100 shapes for testing. For baseline comparisons, we choose five state-of-the-art single-shape spherical parameterization approaches, i.e., CMCF [33] and SMAT [50], AHSP [28], ARAP [61], VC25 [11]. Here, CMCF is flow-based, while the other four baselines are optimization-based.

We consider four metrics for experimental evaluation. Similar to VC25, the first three metrics evaluate the quality of each spherical parameterization via its geometric distortion. Three of these metrics consider the distortions of each triangle t and each method. Specifically, we compute the Jacobian J_t between the image of t on the sphere and itself. We report two distortions based on J_t :

$$\sigma_t = \max\left(\sigma_{\max}(J_t), \frac{1}{\sigma_{\min}(J_t)}\right), \quad \sigma_t^{\text{con}} = \max\left(\frac{\sigma_{\max}(J_t)}{\sigma_{\text{mean}}(J_t)}, \frac{\sigma_{\text{mean}}(J_t)}{\sigma_{\min}(J_t)}\right).$$

where $\sigma_{\max}(J_t)$, $\sigma_{\min}(J_t)$, and $\sigma_{\text{mean}}(J_t)$ are the maximum, minimum, and mean singular values of J_t . We report the mean and maximum of σ_t and the maximum of σ_t^{con} among all triangles. The fourth metric evaluates the consistency of the spherical parameterizations. First we compute a graph $\mathcal{G} = (\{S_1, \dots, S_N\}, \mathcal{E})$ that connects similar input shapes. $\mathcal{E}$ connects each shape S_i to its 6 nearest neighbors according to the Hausdorff distance. Let $m_{ij} : S_i \rightarrow S_j$ be the ground-truth mapping between two similar shapes $(S_i, S_j) \in \mathcal{E}$. We use ground truth mappings m_{ij} provided by KeypointNet [67]. We compute the differences $\|m_{ij} \circ m_i^{-1}(S_0), m_j^{-1}(S_0)\|$ between images of the same sample points of S_0 on S_i and S_j . We report the average of $c(m_i, m_j) = d(m_{ij} \circ m_i^{-1}(S_0), m_j^{-1}(S_0)) / d_{\text{Haus}}(S_i, S_j)$. The lower the value of $c(m_i, m_j)$ the more consistent m_i and m_j . Note that for all baseline approaches, we employ Möbius registration to factor out the underlying Möbius group [6] before calculating $c(m_i, m_j)$.

5.2 Sphere Parameterization Quality

Table 1 presents quantitative comparisons between GenSP and the baseline approaches. We can see that GenSP outperforms all baseline approaches considerably across all metrics. Compared to CMCF, GenSP reduces mean(σ_t)

	$\text{mean}(\sigma_t)$	$\text{max}(\sigma_t)$	$\text{mean}(\sigma_t^{\text{con}})$	$\text{mean}(c(m_i, m_j))$	run-time
CMCF	24.22	1.13×10^3	1.029	1.626	~ 5 min
ARAP	177.19	1.219×10^6	1.029	1.608	~ 6 min
AHSP	1.578	2.349	1.232	6.006	~ 32 sec
SMAT	1.503	3.392	1.049	5.122	~ 2 min
VC25	12.520	216.70	6.892	6.573	~ 5 min
GenSP	1.681	7.719	1.212	1.425	< 1 sec
No-DA	2.326	614.14	1.443	3.164	-
No-GP	1.866	23.228	1.214	1.613	-
No-GR	2.296	271.26	1.627	1.740	-

Table 1: (Top) Baseline comparisons on the held-out test set between GenSP and five state-of-the-art spherical parameterization approaches, CMCF [33], SMAT [50], AHSP [28], ARAP [61], and VC25 [11]. We report mean and max of isometric distortion — $\text{mean}(\sigma_t)$ and $\text{max}(\sigma_t)$, max of conformal distortion — $\text{max}(\sigma_t^{\text{con}})$, and the consistency of spherical parameterizations — $\text{mean}(c(m_i, m_j))$. (Bottom) Ablation study results. (No-DA) No data augmentation. (No-GP). Matching the sphere to each shape directly using intermediate shapes from the implicit generator, i.e., without using a path of training shapes. (No-GR) Without using regularization losses when training the generative models.

and $\text{max}(\sigma_t)$ by $14.3\times$ and $146\times$ respectively. Even for $\text{mean}(\sigma_t^{\text{con}})$ which favors CMCF, GenSP only increases by $1.17\times$. GenSP’s $\text{mean}(\sigma_t)$, $\text{max}(\sigma_t)$, and $\text{mean}(\sigma_t^{\text{con}})$ also remain on par with per-shape optimization based methods.

With respect to the consistency metric, $\text{mean}(c(m_i, m_j))$, GenSP leads to significant performance gains compared to per-shape optimization methods, i.e., $3.6\times$ better than the top-performing optimization-based baseline SMAT. It also outperforms flow based approaches to a lesser extent. This is expected as GenSP explicitly models the consistency of spherical parameterizations as an optimization objective.

Figure 5 shows qualitative comparisons between GenSP and baseline approaches. Qualitative results are consistent with quantitative results. The individual spherical parameterizations of GenSP exhibit fewer distortions, which is expected. Moreover, GenSP is the only approach that offers consistent parameterizations across the shape collection, in which semantic feature keypoints are in correspondence.

Bijectivity Guarantees: GenSP does not provide formal bijectivity guarantees, but it is empirically near-bijective. On our 100-shape ShapeNet test-set, 91 shapes produce fully bijective parameterizations and the mean per-shape rate of self-intersecting or degenerate faces is 0.099% as shown in Table 2. This is comparable to optimization-based methods that explicitly target bijectivity per shape and is substantially better than other flow-based or stereographic methods.

5.3 Ablation Study

No data augmentation The first ablation study removes the data augmentation step and uses only input shapes to learn the generative models. As shown

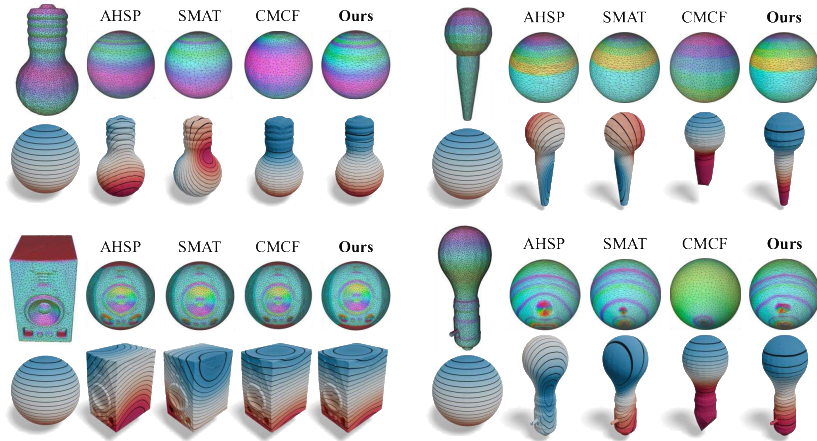


Fig. 5: Additional qualitative spherical parameterization results. The top row in each block visualizes the mapping of each method from a shape to the sphere. (The spheres are manually rotated for ease of visualization.) The bottom row shows the mapping of each method from the sphere to the shape. We can see that our method offers more consistent and high-quality parameterizations.

Source	GenSP	Diff3F	Diffumatch	ULRSSM
				
				
Time	~ 0.3s	~ 9 min	~ 80s	~ 50s

Fig. 6: Comparison against modern correspondence methods.

Method	Bad face %	# Affected meshes
SMAT	0.004	4
AHSP	0.002	3
CMCF	0.241	15
ARAP	0.434	31
VC25	1.898	91
GenSP	0.099	9

Table 2: Percentage of self-intersecting and degenerate faces in spherical parameterizations.

in Table 1, all metrics of this variant drop significantly. This is because extrinsic generative models fail to recover the underlying complex deformations between the sphere and these shapes without training on shapes that bridge them.

No intermediate training shapes The second ablation study replaces correspondence propagation along a path of adjacent training shapes in stage III with intermediate shapes defined by the implicit shape generator via straight-line interpolation in the latent space. Table 1 shows that this variant leads to notable performance drops in all metrics, except the consistency metric, which is similar to CMCF. This is because the mesh generator depends on the quality of the initial correspondences and cannot recover from severe errors in initial correspondences, which leads to poor isometry errors. On the other hand, as the

latent space is still organized according to the intermediate CMCF shapes, the consistency metric remains similar to that of CMCF.

No geometric regularizations The third ablation study drops the geometric regularizations used in training the implicit and mesh generators. Table 1 shows that this variant leads to noticeable performance drops. This is expected as geometric regularizations are critical for determining how to interpolate shapes between sparse shape samples, e.g., between complex shapes. Figure 4 depicts the improvement in the quality of parameterizations when this regularization is used, which also contributes to the improvement in metrics.

5.4 Applications

In Fig. 6, we compare GenSP against three recent correspondence methods (Diff3F [19], Diffumatch [46], and ULRSSM [9]). GenSP produces more coherent inter-class and intra-class surface correspondences while running at least $100\times$ faster. ULRSSM fails on axially symmetric shapes due to rotationally non-unique features. We show a further application in shape morphing and comparison with classical methods in Sec. C.

6 Conclusions and Future Work

In this paper, we have presented a data-driven approach that computes a consistent spherical parameterization among a collection of genus-0 shapes. We have shown how to formulate this problem as learning a neural deformation-based mesh generative model. We present a technique to compute consistent correspondences between training shapes and the sphere to initialize the mesh generative model. The experimental results reveal that GenSP retains the high-quality isometric factors yielded by per-shape optimization based spherical parameterization approaches while simultaneously providing consistent parameterizations like flow based methods.

There are ample opportunities for future research. First is how to extend this framework to higher genera shapes. CMCF could be replaced with a curvature flow admitting non-constant final mean curvature (e.g., Willmore flow variants [15]) and the sphere with high-genera templates. Second, the shape generative models are extrinsic, while spherical parameterization is an intrinsic problem. Developing an intrinsic shape generative model for spherical parameterization is another interesting problem.

7 Acknowledgments

This project was supported by NSF-2047677, 2413161, 2504906, and 2515626; GIFTs from Adobe, Google, and Ericsson; and computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at UT Austin.

References

1. Aigerman, N., Gupta, K., Kim, V.G., Chaudhuri, S., Saito, J., Groueix, T.: Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Trans. Graph.* **41**(4) (Jul 2022). <https://doi.org/10.1145/3528223.3530141>, <https://doi.org/10.1145/3528223.3530141>
2. Alexa, M.: Merging polyhedral shapes with scattered features. *The Visual Computer* **16**(1), 26–37 (2000). <https://doi.org/10.1007/s003710050003>, [doi.org](https://doi.org/10.1007/s003710050003)
3. Attaiki, S., Pai, G., Ovsjanikov, M.: DPFM: deep partial functional maps. In: *International Conference on 3D Vision, 3DV 2021, London, United Kingdom, December 1-3, 2021*. pp. 175–185. IEEE, London, United Kingdom (2021). <https://doi.org/10.1109/3DV53792.2021.00040>, <https://doi.org/10.1109/3DV53792.2021.00040>
4. Atzmon, M., Lipman, Y.: SALD: sign agnostic learning with derivatives. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. pp. –. OpenReview.net, Virtual Event, Austria (2021), <https://openreview.net/forum?id=7EDgLu9reQD>
5. Atzmon, M., Novotný, D., Vedaldi, A., Lipman, Y.: Augmenting implicit neural shape representations with explicit deformation fields (2021), <https://arxiv.org/abs/2108.08931>
6. Baden, A., Crane, K., Kazhdan, M.: Möbius registration. *Comput. Graph. Forum* **37**(5), 211–220 (2018). <https://doi.org/10.1111/CGF.13503>, <https://doi.org/10.1111/cgf.13503>
7. Baden, A., Crane, K., Kazhdan, M.: Möbius registration. In: *Computer Graphics Forum*. vol. 37, pp. 211–220. Wiley Online Library (2018)
8. Bouritsas, G., Bokhnyak, S., Ploumpis, S., Zafeiriou, S., Bronstein, M.M.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. pp. 7212–7221. IEEE, New York, NY, USA (2019), <https://doi.org/10.1109/ICCV.2019.00731>
9. Cao, D., Roetzer, P., Bernard, F.: Unsupervised learning of robust spectral shape matching. *ACM TOG* (2023)
10. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015)
11. Chen, W., Xiao, D., Chen, R.: Bijective spherical parameterization via stereographic projection. *Vis. Comput.* **41**(9), 6413–6424 (2025). <https://doi.org/10.1007/S00371-025-03942-5>, <https://doi.org/10.1007/s00371-025-03942-5>
12. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. pp. 6968–6979. Computer Vision Foundation / IEEE, Washington, DC, USA (2020). <https://doi.org/10.1109/CVPR42600.2020.00700>, https://openaccess.thecvf.com/content_CVPR_2020/html/Chibane_Implicit_Functions_in_Feature_Space_for_3D_Shape_Reconstruction_and_CVPR_2020_paper.html
13. Cosmo, L., Norelli, A., Halimi, O., Kimmel, R., Rodolà, E.: LIMP: learning latent shape representations with metric preservation priors. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) *Computer Vision - ECCV 2020 - 16th European*

- Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12348, pp. 19–35. Springer, New York, NY, USA (2020). https://doi.org/10.1007/978-3-030-58580-8_2, https://doi.org/10.1007/978-3-030-58580-8_2
14. Crane, K., Pinkall, U., Schröder, P.: Robust fairing via conformal curvature flow. *ACM Trans. Graph.* **32**(4) (Jul 2013). <https://doi.org/10.1145/2461912.2461986>, <https://doi.org/10.1145/2461912.2461986>
 15. Crane, K., Pinkall, U., Schröder, P.: Robust fairing via conformal curvature flow. *ACM Trans. Graph.* **32** (2013)
 16. Das, G., Goodrich, M.T.: On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Comput. Geom. Theory Appl.* **8**(3), 123–137 (1997). [https://doi.org/10.1016/S0925-7721\(97\)00006-0](https://doi.org/10.1016/S0925-7721(97)00006-0), [https://doi.org/10.1016/S0925-7721\(97\)00006-0](https://doi.org/10.1016/S0925-7721(97)00006-0)
 17. Donati, N., Corman, E., Ovsjanikov, M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 732–741. IEEE, Washington, DC, USA (2022). <https://doi.org/10.1109/CVPR52688.2022.00082>, <https://doi.org/10.1109/CVPR52688.2022.00082>
 18. Donati, N., Sharma, A., Ovsjanikov, M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 8589–8598. Computer Vision Foundation / IEEE, Washington, DC, USA (2020). <https://doi.org/10.1109/CVPR42600.2020.00862>, https://openaccess.thecvf.com/content_CVPR_2020/html/Donati_Deep_Geometric_Functional_Maps_Robust_Feature_Learning_for_Shape_Correspondence_CVPR_2020_paper.html
 19. Dutt, N.S., Muralikrishnan, S., Mitra, N.J.: Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In: *IEEE CVPR* (2024)
 20. Eisenberger, M., Löhner, Z., Cremers, D.: Divergence-free shape correspondence by deformation. *Comput. Graph. Forum* **38**(5), 1–12 (2019). <https://doi.org/10.1111/CGF.13785>, <https://doi.org/10.1111/cgf.13785>
 21. Eisenberger, M., Novotný, D., Kerchenbaum, G., Labatut, P., Neverova, N., Cremers, D., Vedaldi, A.: Neuromorph: Unsupervised shape interpolation and correspondence in one go. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 7473–7483. Computer Vision Foundation / IEEE, Washington, DC, USA (2021). <https://doi.org/10.1109/CVPR46437.2021.00739>, https://openaccess.thecvf.com/content/CVPR2021/html/Eisenberger_NeuroMorph_Unsupervised_Shape_Interpolation_and_Correspondence_in_One_Go_CVPR_2021_paper.html
 22. Gotsman, C., Gu, X., Sheffer, A.: Fundamentals of spherical parameterization for 3d meshes. In: *ACM SIGGRAPH 2003 Papers*, p. 358–363. SIGGRAPH '03, Association for Computing Machinery, New York, NY, USA (2003). <https://doi.org/10.1145/1201775.882276>, <https://doi.org/10.1145/1201775.882276>
 23. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: *Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11206*, pp. 235–251. Springer, New York, NY, USA (2018).

- https://doi.org/10.1007/978-3-030-01216-8_15, https://doi.org/10.1007/978-3-030-01216-8_15
24. Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M.: Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* **6**(2), 181–189 (2000). <https://doi.org/10.1109/2945.856994>
 25. Hartwig, F., Sassen, J., Azencot, O., Rumpf, M., Ben-Chen, M.: An elastic basis for spectral shape correspondence. In: *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3588432.3591518>, <https://doi.org/10.1145/3588432.3591518>
 26. Hartwig, F., Sassen, J., Azencot, O., Rumpf, M., Ben-Chen, M.: An elastic basis for spectral shape correspondence. In: *ACM SIGGRAPH 2023 conference proceedings*. pp. 1–11 (2023)
 27. Hormann, K., Lévy, B., Sheffer, A.: Mesh parameterization: theory and practice video files associated with this course are available from the citation page. In: McMains, S., Sloan, P. (eds.) *International Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 2007, San Diego, California, USA, August 5–9, 2007, Courses. p. 1. ACM, San Diego, California, USA (2007). <https://doi.org/10.1145/1281500.1281510>, <https://doi.org/10.1145/1281500.1281510>
 28. Hu, X., Fu, X.M., Liu, L.: Advanced hierarchical spherical parameterizations. *IEEE Transactions on Visualization and Computer Graphics* **24**(6), 1930–1941 (2017)
 29. Huang, Q., Guibas, L.J.: Consistent shape maps via semidefinite programming. *Comput. Graph. Forum* **32**(5), 177–186 (2013). <https://doi.org/10.1111/CGF.12184>, <https://doi.org/10.1111/cgf.12184>
 30. Huang, Q., Huang, X., Sun, B., Zhang, Z., Jiang, J., Bajaj, C.: Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10–17, 2021*. pp. 5795–5805. IEEE, New York, NY, USA (2021). <https://doi.org/10.1109/ICCV48922.2021.00576>, <https://doi.org/10.1109/ICCV48922.2021.00576>
 31. Huang, Q., Wang, F., Guibas, L.J.: Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)* **33**(4), 36:1–36:11 (2014). <https://doi.org/10.1145/2601097.2601111>, <https://doi.org/10.1145/2601097.2601111>
 32. Huang, Q., Wicke, M., Adams, B., Guibas, L.: Shape decomposition using modal analysis. *Computer Graphics Forum* **28**(2), 407–416 (2009). <https://doi.org/10.1111/j.1467-8659.2009.01380.x>
 33. Kazhdan, M., Solomon, J., Ben-Chen, M.: Can mean-curvature flow be modified to be non-singular? *Comput. Graph. Forum* **31**(5), 1745–1754 (2012). <https://doi.org/10.1111/J.1467-8659.2012.03179.X>, <https://doi.org/10.1111/j.1467-8659.2012.03179.x>
 34. Kim, V.G., Li, W., Mitra, N.J., DiVerdi, S., Funkhouser, T.: Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. Graph.* **31**(4), 54:1–54:11 (Jul 2012), [10.1145/2185520.2185550](https://doi.org/10.1145/2185520.2185550)
 35. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. *ACM transactions on graphics (TOG)* **30**(4), 1–12 (2011)
 36. Kobbelt, L.P., Vorsatz, J., Labsik, U., Seidel, H.P.: A shrink-wrapping approach to remeshing polygonal surfaces. *Computer Graphics Forum* **18**(3), 119–130 (1999). <https://doi.org/10.1111/1467-8659.00319>, dl.acm.org

37. Lipman, Y., Funkhouser, T.: Möbius voting for surface correspondence. *ACM Trans. Graph.* **28**(3) (Jul 2009). <https://doi.org/10.1145/1531326.1531378>, <https://doi.org/10.1145/1531326.1531378>
38. Litany, O., Remez, T., Rodolà, E., Bronstein, A.M., Bronstein, M.M.: Deep functional maps: Structured prediction for dense shape correspondence. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017*. pp. 5660–5668. IEEE Computer Society, Washington, DC, USA (2017). <https://doi.org/10.1109/ICCV.2017.603>, <https://doi.org/10.1109/ICCV.2017.603>
39. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* **34**(6), 248:1–248:16 (Oct 2015)
40. Nguyen, A., Ben-Chen, M., Welnicka, K., Ye, Y., Guibas, L.J.: An optimization approach to improving collections of shape maps. *Comput. Graph. Forum* **30**(5), 1481–1491 (2011). <https://doi.org/10.1111/J.1467-8659.2011.02022.X>, <https://doi.org/10.1111/j.1467-8659.2011.02022.x>
41. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.* **31**(4) (Jul 2012). <https://doi.org/10.1145/2185520.2185526>, <https://doi.org/10.1145/2185520.2185526>
42. Pachauri, D., Kondor, R., Singh, V.: Solving the multi-way matching problem by permutation synchronization. In: *NIPS*. pp. 1860–1868 (2013)
43. Park, J.J., Florence, P.R., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019*. pp. 165–174. Computer Vision Foundation / IEEE, New York, NY, USA (2019). <https://doi.org/10.1109/CVPR.2019.00025>
44. Pentapati, S.K., Phillips, G., Bovik, A.C.: Hierarchical neural surfaces for 3d mesh compression. In: *2025 Picture Coding Symposium (PCS)*. pp. 1–5 (2025). <https://doi.org/10.1109/PCS65673.2025.11417660>
45. Pierson, E., Li, L., Dai, A., Ovsjanikov, M.: Diffumatch: Category-agnostic spectral diffusion priors for robust non-rigid shape matching. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 5745–5756. IEEE, Washington, DC, USA (October 2025)
46. Pierson, E., Li, L., Dai, A., Ovsjanikov, M.: Diffumatch: Category-agnostic spectral diffusion priors for robust non-rigid shape matching. *arXiv:2507.23715* (2025)
47. Praun, E., Hoppe, H.: Spherical parametrization and remeshing. *ACM Trans. Graph.* **22**(3), 340–349 (Jul 2003). <https://doi.org/10.1145/882262.882274>, <https://doi.org/10.1145/882262.882274>
48. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19–25, 2021*. pp. 10318–10327. Computer Vision Foundation / IEEE, Washington, DC, USA (2021). <https://doi.org/10.1109/CVPR46437.2021.01018>, https://openaccess.thecvf.com/content/CVPR2021/html/Pumarola_D-NeRF_Neural_Radiance_Fields_for_Dynamic_Scenes_CVPR_2021_paper.html
49. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision - ECCV 2018 - 15th European Conference, Munich*,

- Germany, September 8-14, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11207, pp. 725–741. Springer, New York, NY, USA (2018), https://doi.org/10.1007/978-3-030-01219-9_43
50. Schmidt, P., Pieper, D., Kobbelt, L.: Surface maps via adaptive triangulations. *Comput. Graph. Forum* **42**(2), 103–117 (2023). <https://doi.org/10.1111/CGF.14747>, <https://doi.org/10.1111/cgf.14747>
 51. Shapiro, A., Tal, A.: Polygon realization for shape transformation. *The Visual Computer* **14**(8-9), 429–444 (1998). <https://doi.org/10.1007/s003710050143>, link.springer.com
 52. Sheffer, A., Praun, E., Rose, K.: Mesh parameterization methods and their applications. *Found. Trends Comput. Graph. Vis.* **2**(2), xx–yy (2006). <https://doi.org/10.1561/06000000011>, <https://doi.org/10.1561/06000000011>
 53. Shen, Y., Huang, Q., Srebro, N., Sanghavi, S.: Normalized spectral map synchronization. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, December 5–10, 2016, Barcelona, Spain. pp. 4925–4933 (2016), <https://proceedings.neurips.cc/paper/2016/hash/bb03e43ffe34eeb242a2ee4a4f125e56-Abstract.html>
 54. Slavcheva, M., Baust, M., Cremers, D., Ilic, S.: Killingfusion: Non-rigid 3d reconstruction without correspondences. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, July 21–26, 2017. pp. 5474–5483. IEEE Computer Society, Washington, DC, USA (2017). <https://doi.org/10.1109/CVPR.2017.581>, <https://doi.org/10.1109/CVPR.2017.581>
 55. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. pp. 109–116. SGP ’07, Eurographics Association, Goslar, DEU (2007)
 56. Sun, B., Groueix, T., Song, C., Huang, Q., Aigerman, N.: Tuttenet: Injective 3d deformations by composition of 2d mesh deformations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 21378–21389. IEEE Computer Society, Washington, DC, USA (June 2024)
 57. Sundararaman, R., Donati, N., Melzi, S., Corman, E., Ovsjanikov, M.: Deformation recovery: Localized learning for detail-preserving deformations. *ACM Trans. Graph.* **43**(6) (Nov 2024). <https://doi.org/10.1145/3687968>, <https://doi.org/10.1145/3687968>
 58. Tam, G.K.L., Cheng, Z., Lai, Y., Langbein, F.C., Liu, Y., Marshall, A.D., Martin, R.R., Sun, X., Rosin, P.L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* **19**(7), 1199–1217 (2013). <https://doi.org/10.1109/TVCG.2012.310>, <https://doi.org/10.1109/TVCG.2012.310>
 59. Trappolini, G., Cosmo, L., Moschella, L., Marin, R., Melzi, S., Rodolà, E.: Shape registration in the time of transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, December 6–14, 2021, virtual. pp. 5731–5744. Curran Associates, Inc., Red Hook, NY, USA (2021), <https://proceedings.neurips.cc/paper/2021/hash/2d3d9d5373f378108cdbc30a3c52bd3e-Abstract.html>
 60. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, Montreal, QC, Canada, October 10–17, 2021. pp.

- 12939–12950. IEEE, Washington, DC, USA (2021). <https://doi.org/10.1109/ICCV48922.2021.01272>, <https://doi.org/10.1109/ICCV48922.2021.01272>
61. Wang, C., Liu, Z., Liu, L.: As-rigid-as-possible spherical parametrization. *Graph. Models* **76**(5), 457–467 (Sep 2014). <https://doi.org/10.1016/j.gmod.2014.03.016>, <https://doi.org/10.1016/j.gmod.2014.03.016>
 62. Williamson, R., Mitra, N.J.: Neural geometry processing via spherical neural surfaces. *Comput. Graph. Forum* **44**(2), xx–yy (2025). <https://doi.org/10.1111/CGF.70021>, <https://doi.org/10.1111/cgf.70021>
 63. Yang, H., Huang, X., Sun, B., Bajaj, C.L., Huang, Q.: Gencorres: Consistent shape matching via coupled implicit-explicit shape generative models. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7–11, 2024. pp. xx–yy. OpenReview.net, Vienna, Austria (2024), <https://openreview.net/forum?id=dGH4kHFKFj>
 64. Yang, H., Sun, B., Chen, L., Pavel, A., Huang, Q.: Geolent: A geometric approach to latent space design for deformable shape generators. *ACM Trans. Graph.* **42**(6) (Dec 2023). <https://doi.org/10.1145/3618371>, <https://doi.org/10.1145/3618371>
 65. Yang, Y., Yang, H., Nakayama, K., Huang, X., Guibas, L.J., Huang, Q.: Genanalysis: Joint shape analysis by learning man-made shape generators with deformation regularizations. *ACM Trans. Graph.* **44**(4), 61:1–61:19 (2025). <https://doi.org/10.1145/3731164>, <https://doi.org/10.1145/3731164>
 66. Yoshiyasu, Y., Ma, L., Yoshida, E., Gu, X.: As-conformal-as-possible surface registration. *Computer Graphics Forum* **33**(5), 257–267 (2014). <https://doi.org/10.1111/cgf.12451>
 67. You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., Wang, W.: Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13647–13656 (2020)
 68. Zhou, K., Bhatnagar, B.L., Pons-Moll, G.: Unsupervised shape and pose disentanglement for 3d meshes. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) *Computer Vision - ECCV 2020 - 16th European Conference*, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII. Lecture Notes in Computer Science, vol. 12367, pp. 341–357. Springer, New York, NY, USA (2020), https://doi.org/10.1007/978-3-030-58542-6_21
 69. Zhou, X., Zhu, M., Daniilidis, K.: Multi-image matching via fast alternating minimization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4032–4040 (2015)
 70. Zhou, Y., Wu, C., Li, Z., Cao, C., Ye, Y., Saragih, J.M., Li, H., Sheikh, Y.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, December 6–12, 2020, virtual. p. 9251–9262. Curran Associates, Red Hook, NY, USA (2020), <https://proceedings.neurips.cc/paper/2020/hash/68dd09b9ff11f0df5624a690fe0f6729-Abstract.html>
 71. Zhuravlev, A., Löhner, Z., Golyanik, V.: Denoising functional maps: Diffusion models for shape correspondence. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11–15, 2025. pp. 26899–26909. Computer Vision Foundation / IEEE, Washington, DC, USA (2025). <https://doi.org/10.1109/CVPR52734.2025.02505>, <https://openaccess.>

thecvf.com/content/CVPR2025/html/Zhuravlev_Denoising_Functional_Maps_Diffusion_Models_for_Shape_Correspondence_CVPR_2025_paper.html

72. Zuffi, S., Kanazawa, A., Jacobs, D.W., Black, M.J.: 3d menagerie: Modeling the 3d shape and pose of animals. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 5524–5532. IEEE Computer Society, New York, NY, USA (2017). <https://doi.org/10.1109/CVPR.2017.586>, <https://doi.org/10.1109/CVPR.2017.586>

A Derivation of Eq. 6

This section presents a matrix reformulation of Eq. 6, which is

$$\mathbf{d}^T L^\phi(\mathbf{z}) \mathbf{d} = \min_{\mathbf{s}_i, \mathbf{c}_i} \sum_{i=1}^l \left(\sum_{j \in \mathcal{N}_i} \|A_i(\mathbf{p}_i^\phi(\mathbf{z}) - \mathbf{p}_j^\phi(\mathbf{z})) - (\mathbf{d}_i - \mathbf{d}_j)\|^2 + \eta s_i^2 \right) \quad (13)$$

where

$$A_i = s_i I_3 + \mathbf{c}_i \times I_3.$$

Denote $\mathbf{x} \in \mathbb{R}^{4l}$ where $\mathbf{x}_i = (s_i; \mathbf{c}_i)$. Introduce $L \in \mathbb{R}^{m \times m}$, $B^\phi(\mathbf{z}) \in \mathbb{R}^{3m \times 4l}$, and $H^\phi(\mathbf{z}) \in \mathbb{R}^{4l \times 4l}$ where

$$\begin{aligned} L_{ii} &= \begin{cases} |\mathcal{N}_i| & i = j, \quad 1 \leq i \leq l \\ |\{i' | i \in \mathcal{N}_{i'}\}| & i = j, \quad l+1 \leq i \leq m \\ -1 & j \in \mathcal{N}_i, \quad \text{or } i \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases} \\ B_{ji}^\phi(\mathbf{z}) &= \begin{cases} (\mathbf{q}_{ij}^\phi(\mathbf{z}), -\mathbf{q}_{ij}^\phi(\mathbf{z}) \times) & i = i \\ -(\mathbf{q}_{ij}^\phi(\mathbf{z}), -\mathbf{q}_{ij}^\phi(\mathbf{z}) \times) & j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \\ H_{ij}^\phi(\mathbf{z}) &= \begin{cases} \sum_{j' \in \mathcal{N}_i} \begin{pmatrix} \|\mathbf{q}_{ij'}^\phi(\mathbf{z})\|^2 + \eta & 0 \\ 0 & \|\mathbf{q}_{ij'}^\phi(\mathbf{z})\|^2 I_3 - \mathbf{q}_{ij'}^\phi(\mathbf{z}) \mathbf{q}_{ij'}^\phi(\mathbf{z})^T \end{pmatrix} & i = j \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where

$$\mathbf{q}_{ij}^\phi(\mathbf{z}) = \mathbf{p}_i^\phi(\mathbf{z}) - \mathbf{p}_j^\phi(\mathbf{z}).$$

Then

$$\begin{aligned} & \mathbf{d}^T L^\phi(\mathbf{z}) \mathbf{d} \\ &= \min_{\mathbf{x}} \left(\mathbf{d}^T (L \otimes I_3) \mathbf{d} - 2 \mathbf{d}^T B^\phi(\mathbf{z}) \mathbf{x} + \mathbf{x}^T H^\phi(\mathbf{z}) \mathbf{x} \right) \\ &= \mathbf{d}^T \left(L \otimes I_3 - B^\phi(\mathbf{z}) H^\phi(\mathbf{z}) B^\phi(\mathbf{z})^{-1} \right) \mathbf{d}, \quad \left(\mathbf{x} = H^\phi(\mathbf{z})^{-1} (B^\phi(\mathbf{z})^T \mathbf{d}) \right) \end{aligned} \quad (14)$$

meaning

$$L^\phi(\mathbf{z}) = L \otimes I_3 - B^\phi(\mathbf{z}) H^\phi(\mathbf{z}) B^\phi(\mathbf{z})^{-1}.$$

B Implementation Details

B.1 Latent Space Regularization

A desirable property of the latent space is that similar shapes are reconstructed similarly and therefore assigned nearby latent codes. We find that the L1 penalty together with reconstruction loss encourages this desired behavior. We verified that latent distances strongly correlate with 3D Hausdorff distances (Pearson's $r = 0.723$).

B.2 Registration

As mentioned in Sect. 4.3, we construct paths from a template shape S_0 (the sphere) to the target shape S_i via various intermediate shapes. Our goal is to build correspondences between the end points, and to do this we build correspondences between adjacent nodes in the path. Given two meshes in the path, S_i and S_j , we first take their latents and interpolate them to get n interpolated latents. We then decode them to get n intermediate meshes between the two nodes in the path, say $(S_{i \rightarrow j})_1, \dots, (S_{i \rightarrow j})_n$. In our implementation, $n = 5$. We then iteratively register $(S_{i \rightarrow j})_{n-1}$ to $(S_{i \rightarrow j})_n$ via standard non-rigid registration with an as-rigid-as possible deformation prior. This gives a remeshed version of S_j with the same connectivity as S_i .

B.3 Stitching

Our next task is to “stitch” these independently registered segments together such that we can project the connectivity of a target shape S onto the surface of the sphere. Consider two registered segments $S_{i \rightarrow j}$ which has shape S_j remeshed to connectivity of S_i , and $S_{j \rightarrow k}$ which has S_k remeshed to connectivity of S_j . Then essentially, as we have two differently remeshed versions of S_j , we can create a bijective map between them using barycentric interpolation of one vertex layout onto the surface of the other. This map can be extended to S_i , which now has the same connectivity to S_k . Iterating this process through all segments till the sphere, we can project the connectivity of S to the sphere, creating the initializations that are used for training the mesh generative model.

B.4 Dealing with Faulty Correspondences

While our two-step registration and stitching process ensures that correspondences are propagated from the sphere to training shapes without severe self intersections, there are often $\sim 5\%$ triangles that show self intersections on the sphere. To ensure these faulty correspondences do not adversely affect the training of the mesh decoder, we simply omit samples from such regions from our mesh initialization dataset. Our mesh initialization training is robust enough to learn reasonable interpolations in these missing regions. The final joint-training stage relies on the ground truth shapes for minimizing the squared distance between the predicted shape and the original. This final stage fixes the reconstruction in the under-sampled regions.

B.5 Network Architecture

Encoder: We use the PointNet++ architecture for the shape encoder module, which produces latents with 256 dimensions for any given shape.

Decoder: Both our point decoder and mesh decoder follow the architecture of DeepSDF, and consist of 10 fully connected layers with 1024 hidden units in each of them. We have residual connections from the input to the third and sixth

layer. In the case of the mesh decoder, we output a single SDF value, whereas for the mesh decoder produces a 3D location. We use

We train the implicit model for 600 epochs, the first 500 of which are without geometric regularization and the next 100 are including it. We train the mesh decoder for 2500 epochs, out of which the first 2000 are done without geometric regularization and the subsequent 500 include it. For Stage II, $\lambda = 0.1$ and $\mu = 0.01$. For Stage IV, $\bar{\mu} = 0.1$ and $\beta = 0.01$.

B.6 Training Details

We train the implicit model on $8 \times \text{GH200}$ GPUS for 8 hours with a cumulative batch size of 96. Meanwhile, the mesh decoder training was done on a single H100 GPU with a batch size of 8. This takes a total of 14 hours wall-clock time. Learning rate for the implicit model training and mesh initialization is set at $1\text{e-}3$, while the joint training stage uses a learning rate of $1\text{e-}4$. All stages use cosine annealing scheduler and the Adam optimizer.

C More Results and Ablations

We also trained and evaluated our method on the challenging human dataset D-FAUST (trained on 1180 shapes and tested on 120 held-out shapes). As shown in Table 3, GenSP achieves the best consistency and mean conformal distortion by a high margin. Mean isometric distortion is on par with the optimization-based baselines, consistent with the ShapeNet results.

Metric	CMCF	ARAP	AHSP	SMAT	GenSP
$\bar{\sigma}_t$	4.63e4	8.96e2	4.41	4.81	5.12
$\max(\sigma_t)$	4.17e6	4.16e5	10.90	201.56	310.01
$\bar{\sigma}_t^{\text{con}}$	1.71e5	8.95e4	3.95	2.88	2.33
$c(m_i, m_j)$	3.47	3.43	7.95	7.97	1.68

Table 3: Baseline comparisons on the held-out test set of D-FAUST (120 shapes) between GenSP and four state-of-the-art spherical parameterization approaches, CMCF [33], SMAT [50], AHSP [28], and ARAP [61]. VC25 [11] fails to converge on D-FAUST.

In Fig. 1, Fig. 5, and Fig. 8, the geometries of shapes are reconstructed from their spherical parameterizations obtained from different methods, in addition to mapping the texture defined on the sphere.

For the baseline methods which give a spherical parameterization of a shape that have the same connectivity as it, we first find the barycentric coordinates of all 40,962 vertices of the icosphere (level = 6) on the triangles of the given

shape’s spherical parameterization. Then using these coordinates, we interpolate the vertex coordinates of the original shape while maintaining the same connectivity as that of the icosphere.

For GenSP, we first encode the ground truth shape using the shape encoder. We then use the encoded latent to deform the vertices of an icosphere with 40,962 vertices to obtain a reconstruction of a target shape. While reconstructions yielded by GenSP’s mesh decoder are very accurate, the generated points do not lie perfectly on the original shape. As the ground truth shape is available during the process of creating spherical parameterizations, we can simply project vertices generated by GenSP to the ground truth mesh. Finally this yields a bijective map between the ico-sphere and the original shape.

We choose this reconstruction based approach for visualizing various methods of spherical parameterizations for two reasons. Firstly, this remains consistent with the inverse spherical parameterization formulation proposed by GenSP, and secondly, it also helps to visualize the isometry and conformality of the parameterizations through the accuracy of the reconstructions. Due to this manner of reconstruction, we see severely under-reconstructed sections of shapes when using CMCF due to its extremely poor isometric factors.

In Fig. 9, we show qualitative results for removing different components of our method, as discussed in Sec. 5.3. In the No-DA case, we observe bad geometry and major self-intersections. No-GP is perhaps the best, but struggles with thin structures and major concavities. No-GR especially struggles from bad triangulations, evident near the ridges.

We also demonstrate the applicability of GenSP for shape morphing and shape matching in Fig. 10 and Fig. 7 respectively. In Fig. 7 we compare against more classical methods like Möbius Registration [7], BIM [35], and EFM [26].

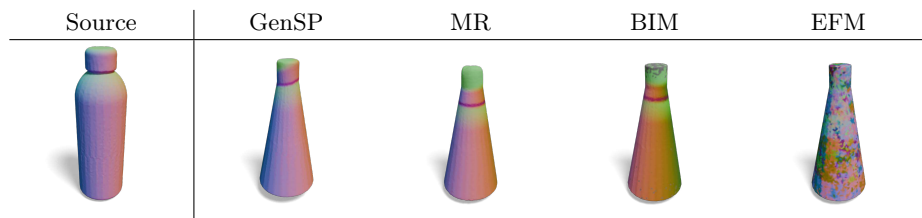


Fig. 7: Application in inter-shape correspondences. Baselines are Möbius registration (MR), blended intrinsic maps (BIM), and elastic functional maps (EFM). Note that EFM needs an initial set of correspondences for initialization, which is why it fails in a zero-shot setting.

D Limitations

All methods of spherical parameterization, including ours and the baselines shown in this paper, fail for shapes with thin structures like the one shown

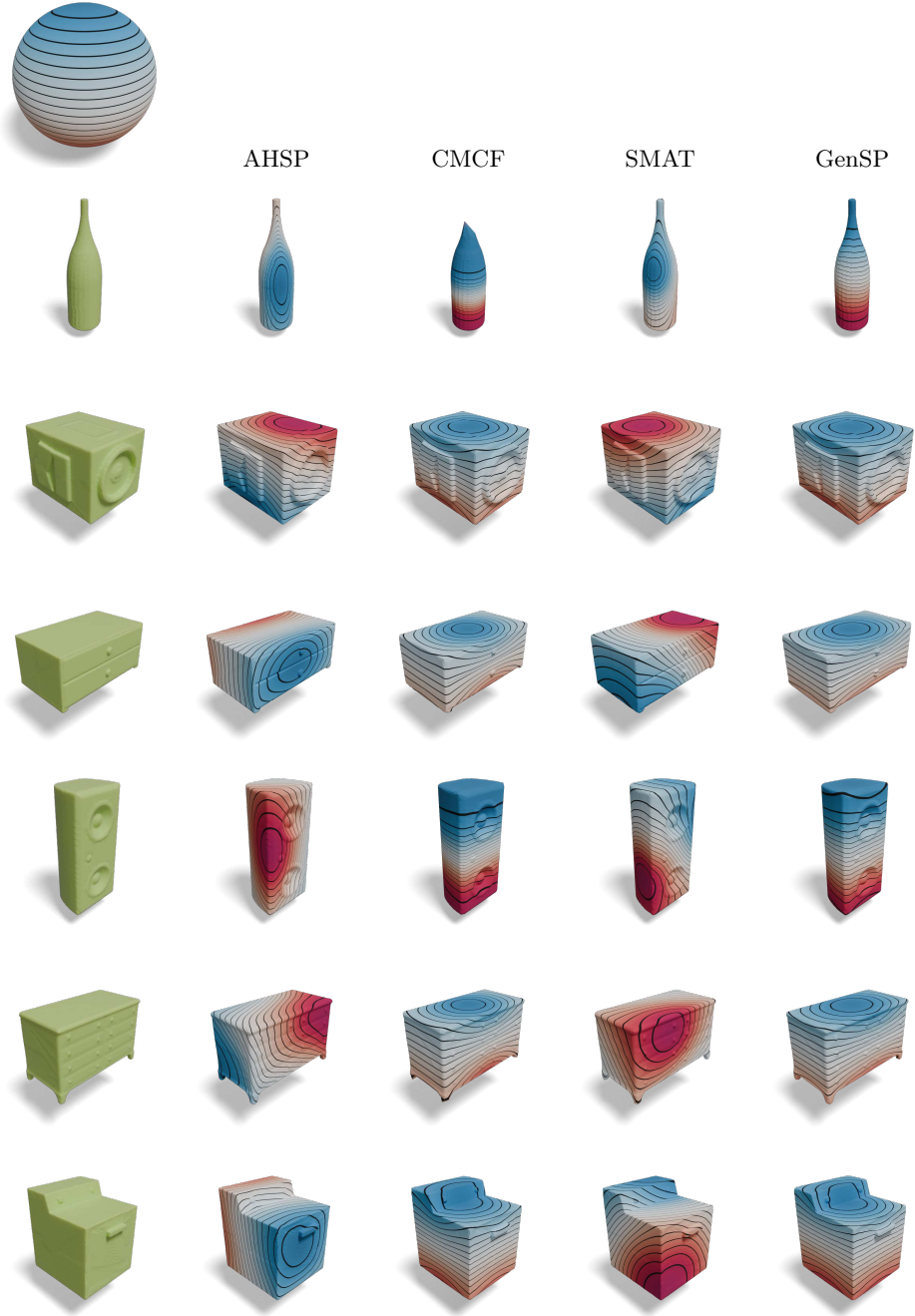


Fig. 8: More qualitative comparisons between GenSP and CMCF [33], SMAT [50], and AHSP [28]. We show images of the same sphere texture under different spherical parameterizations.

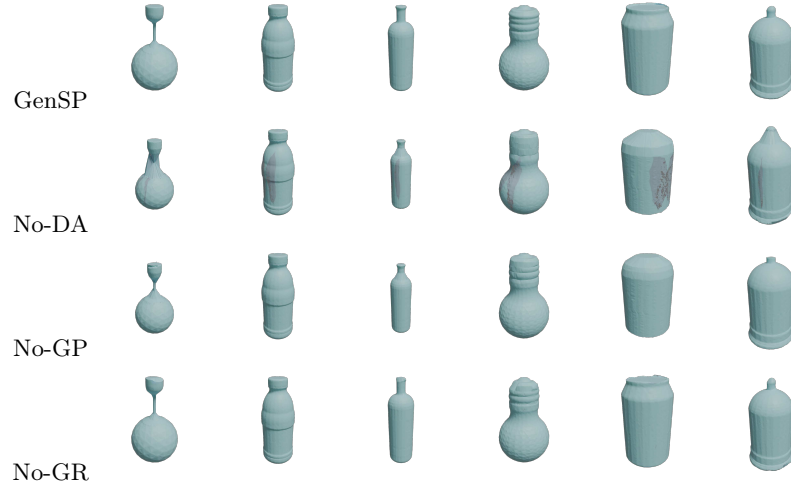


Fig. 9: Qualitative ablation study results.



Fig. 10: Application in shape morphing. Applying GenSP's decoder on interpolated latent vectors produces smoothly morphed shapes.

in Fig. 11. Developing algorithms that address this limitation is an interesting direction for future research.

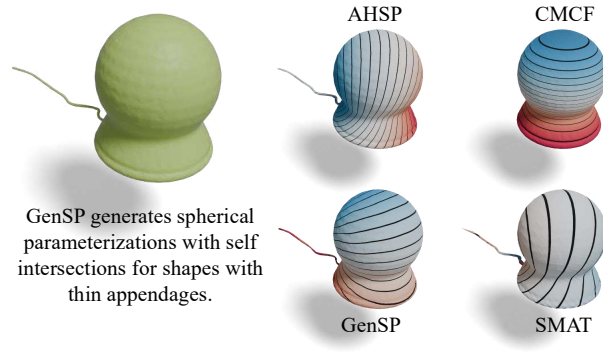


Fig. 11: An example of GenSP's failure mode.