This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PaNDaS: Learnable Shape Interpolation Modeling with Localized Control

Thomas Besnier[1,5], Emery Pierson[2], Sylvain Arguillère[3], Maks Ovsjanikov[2], Mohamed Daoudi[1,4]
[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, Lille, F-59000, France
[2]LIX, École Polytechnique, IP Paris
[3]Univ. Lille, CNRS, UMR 8524 - Laboratoire Paul Painlevé, F-59000 Lille, France
[4]IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France
[5]Department of Computer Science, University of Copenhagen
thomas.besnier@di.ku.dk, emery.pierson@courrier.dev,
sylvain.arguillere@univ-lille.fr, maks@lix.polytechnique.fr, mohamed.daoudi@imt-nord-europe.fr

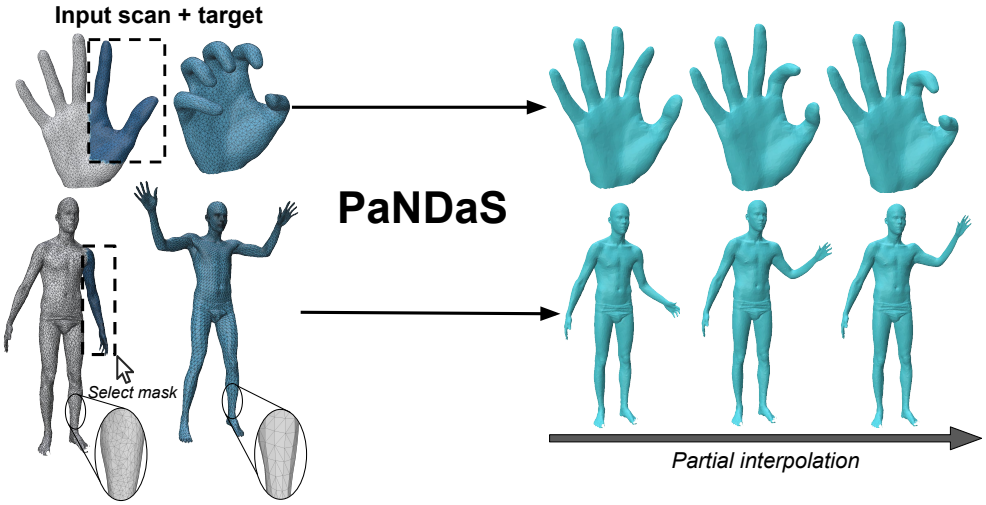


Figure 1. We present **PaNDaS**: a robust learning-based approach for non-rigid deformations of triangular meshes applied to human body surfaces. By combining per-face learned features on a neutral pose mesh (on the left) with a global encoding of a deformed mesh (in blue), the model enables **localized control** of deformations (in cyan) with user-chosen regions (masked in blue on the left). Notably, this approach permits several applications such as mixing poses, transfer of partial poses, interpolations and localized shape statistics.

Abstract

We present PaNDaS, a novel deep learning framework for Partial Non-Rigid Deformations and interpolations of Surfaces (PaNDaS). PaNDaS learns a per-face feature field on the source mesh and fuses it with a global encoding of the target. A deformation generator predicts a Jacobian field and recovers a smooth displacement, enabling precise regional control, pose mixing, and transferable local edits. Unlike previous approaches, our method can restrict the deformations to specific parts of the shape in a versatile way. Across various human body part datasets, PaNDaS achieves state-of-the-art interpolation accuracy and stronger locality than methods based on global shape codes or handles, while remaining robust to remeshing. We demonstrate several localized shape manipulation tasks and show that our method can generate new shapes by combining different input deformations. The project page is available at https://daidedou.sorpi.fr/publication/pandas.

1. Introduction

Predicting deformations of 3D shapes is a cornerstone in human-centered computer vision and graphics, facilitating

a broad range of applications like animation, motion modeling, and character generation. The interpolation problem can be generally defined as follows: how to compute a sequence of non-rigid deformations of a source shape so it ends up matching a target shape? Something left unsaid is that such interpolations should correspond to natural motions to avoid breaking physical plausibility. Over the past few years, the problem of computing interpolation between shapes has been addressed extensively. Proposed solutions use geometric regularization [10, 47], model directly the shape space [2, 4, 20, 22, 26] or, more recently, learn shape interpolation by regularizing neural networks [12, 14, 15]. Most of the approaches rely on a fixed template mesh and do not generalize to unregistered meshes such as shown in Fig. 2. Additionally, this work aims to address a broader

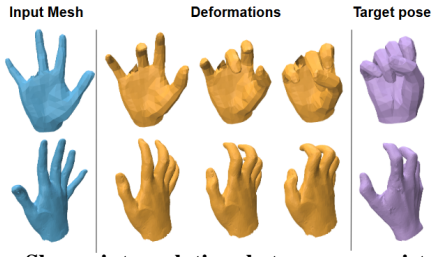


Figure 2. **Shape interpolation between unregistered hand meshes.** The top row shows a predicted interpolation sequence from a hand with missing ring finger. The bottom row shows an interpolation between two raw scans.

version of this deformation problem by also tackling partial deformations as shown in Fig. 1. More precisely, we focus on computing non-rigid deformations of **selected parts** of a source shape so it matches the corresponding parts of a target shape while preserving physical plausibility and without explicitly computing any correspondence map. Most of the cited approaches are not designed to address the challenge of partial deformations and interpolations without a complex configuration. Indeed, maintaining geometric consistency and ensuring smooth transitions between deformed and anchored regions usually require manual corrections or intermediate key points [46]. Moreover, in most of the proposed models, the deformation is extracted from a global latent vector, which makes local control of the deformation difficult. Alternatively, rich and complex shape deformation models have also been constructed through parametric modeling [3, 29, 32, 36, 37, 40], with SMPL [32] being now a standard for many applications and such models do enable localized control. However, they typically rely on human input and modeling for the rig and skinning map, making them difficult to extend to arbitrary shape collections. Our present work aims to improve the flexibility of the deformation and interpolation processes in a purely data-driven way, to establish a deformation model with localized control on any shape category. We propose a novel neural method for Partial Non-rigid Deformations of Surfaces (PaNDaS) to compute local and global near-isometric deformations of triangular meshes. Instead of representing shapes as a single latent vector, we propose to learn both a global shape latent vector and a local point-wise latent representation. Those vectors are fed into a deformation generator based on Neural Jacobian Fields [1] (NJF) to obtain the desired mesh. Localized learning has been proposed in the specific context of deformation recovery from shape-matching algorithms [49], but in this work, we use it to enable local control of the deformation. Indeed, by acting on point-wise features, our model can interpolate between two shapes on selected parts by masking those one wants to deform. As shown in Fig. 1, PaNDaS handles large deformations such as whole human bodies and fingers on a hand. Additionally, one can apply a mix-and-match strategy to create new poses and compute partial shape statistics, as shown in Fig. 8. Thanks to the localized control allowed by the local feature representation, PaNDaS sets a new state-of-the-art regarding partial (and complete) deformations and interpolations of surface meshes, with no additional prior.

We summarize the main contributions of this paper as: *(i)* An end-to-end framework for localized, non-rigid mesh deformations without local supervision, described in Sec. 4; *(ii)* The first learning-based method to allow vertex-level partial interpolations without requiring any texture information, through the manipulation of deep features as shown by extensive experiments in Sec. 5.3; *(iii)* We propose several applications such as new pose generation, partial pose deformation transfer and a robust statistical model of shapes handling partial descriptions detailed in Sec. 5.4.

2. Related work

2.1. Geometric methods for shape deformation

Generating a partial deformation on a mesh has been tackled for a long time, as it is crucial for animation. A family of methods [30, 43, 48] deform the mesh based on user inputs, called handles, which allows the user to drag a point and deform the mesh accordingly. One of the most popular approaches is As-Rigid-As-Possible (ARAP) [47] that solves efficiently the problem by formulating an optimization [2] as a few linear solves. Other approaches use cages [25, 53] defined over the whole mesh to allow the user to edit the mesh locally. However, those approaches require the user to manually select handles to deform the mesh, but do not allow vertex-level deformations. Moreover, only the target deformation is computed (see Sec. 3.1), but not the intermediate motion. In contrast, our model generates partial motions with vertex-level details. Alternatively, some methods optimize the motions by minimizing a path energy, defined by deformation energies such as PriMo [10] that encourages local rigidity. Some methods define the energy by modeling shape as points in a Rieman-

nian shape space [4, 6], where the deformation corresponds to a geodesic. Several Riemannian metrics have been proposed, such as the As-Isometric-As-Possible [26], Square Root Normal Fields [24, 28], or a parameterized family of elastic metrics [5, 27] with terms penalizing specific deformations such as bending or stretching. Only a few approaches have tackled the problem of deforming unregistered shapes by combining the deformation energy with a data attachment term [20–22]. In most cases, the induced optimization problem is highly non-convex and can fail dramatically when the initialization is far from the desired solution. Instead, our approach is data-driven, avoiding hyperparameter tuning of such energy terms.

2.2. Learning approaches for shape deformation

More recently, deep learning methods have been used to learn shape deformations. A popular approach consists of building a latent space using an auto-encoder formulation. Graph convolutional auto-encoders are often used [11, 17, 31, 50, 51, 60], but they tend to overfit information from mesh connectivity. Other approaches, such as 3D-CODED [18], learn auto-encoders for shape matching [19, 38] but are not optimized for shape deformations and tend to fail when the target deformation is non-linear. Neural Jacobian Fields (NJF) [1] improves the generated deformations by predicting the derivatives of the deformation but does not work well for partial deformations. To address this problem, other methods propose to regularize the latent space. The authors of LIMP [14] propose to penalize the deviation of geodesic distances in the latent space. ARAPReg [23] penalizes directions in the latent space that affect the ARAP energy, a strategy that has been improved to provide a geometric latent space [57] or to learn correspondences [58]. Other methods [12, 15] directly learn to produce interpolations between pairs of shapes, but they can struggle with details and may fail to generalize to new data. All these methods focus on learning full shape deformations, which limits applications such as partial shape interpolations.

To address this, a large body of work represents humans with low-dimensional pose and shape parameters with a skinning map [3, 32, 56] with, in particular, SMPL [32] for bodies, MANO [40] for hands, FLAME [29] for faces, and SMPL-X [37] for a unified body-face-hands model. However, pipelines for registering raw scans to a parametric template project geometry onto a coarse, low-dimensional space and fixed topology which can be inaccurate around hands, face or clothing, and high-frequency details are often smoothed away unless extra per-vertex offsets are added on top of the template [7, 41]. These constraints motivate surface-domain approaches that offer vertex-level locality without committing to a rig or a template projection.

In this paper, we therefore propose a method to overcome this limitation by working on the surface level while allowing interpolations of parts of meshes, extending the range of possible deformation sequences by a large margin.

2.3. Neural methods for Partial Deformations

While optimization-based algorithms can be adapted to partial deformation, they usually require a custom configuration for each data category. A few learning-based methods [33, 60] enable partial deformations by structuring the latent space according to the initial shape structure. This allows the user to modify latent points that affect specific parts of the shape. However, as the relationship between latent points and mesh vertices is predefined before training, the level of detail of partial deformations remains limited. Recently, deep methods have started addressing localized control for images [45, 55]. While extending these to surface data is not straightforward because of the non-Euclidean nature of shapes, several works leverage this idea for mesh deformations [52, 54, 59], by combining image diffusion priors with suitable deformation models such as Neural Jacobian fields [1] or by defining key points during movements [35]. However, those methods rely on additional information such as texture maps, user prompts [16] for the diffusion model, or skeletal joints. They also require costly optimization steps, making them impractical for applications in which these priors are unavailable.

To overcome this limitation, our method leverages a learned and controllable feature field paired with a deformation generator, which predicts a constrained Jacobian field used to generate partial, natural deformations of the mesh.

3. Motivations

Now, we motivate our problem, explain why traditional handle-based methods cannot fit our scenario, and present the motivation of our method. If needed, notations can be found in the first section of the supplementary material.

3.1. Limitations of handle-based deformations

Methods such as As-Rigid-As-Possible (ARAP) [47] as and more recent learning models such Neural Shape Deformation Prior [52] can provide partial deformations of meshes in real time, by relying on a set of handles and anchor points. It allows users to select multiple handles and to move them according to user instructions as shown in Fig. 3. However, it provides only the *target* deformation, but not the intermediate poses between the source mesh and the target deformation. One can linearly interpolate the source and target,

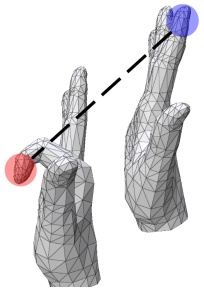


Figure 3. **Principle of handle-based methods.**

but that results in incorrect intermediate poses as shown in Fig. 6. To overcome this, we formulate the interpolation in a learned latent representation.

3.2. Problem formulation

In this work, we aim at predicting full motions of selected, partial regions of a given input mesh. Our inputs are:

- A source mesh $\mathcal{S}$, with $n_{\mathcal{S}}$ vertices $(x_i)_{0\leq i\leq n_{\mathcal{S}}}$, and $m_{\mathcal{S}}$ faces $(t_j)_{0\leq j\leq m_{\mathcal{S}}}$
- A target deformation, represented as a target mesh $\mathcal{T}$, with $n_{\mathcal{T}} \neq n_{\mathcal{S}}$ vertices $(y_i)_{0\leq i\leq n_{\mathcal{T}}}$, and $m_{\mathcal{T}}$ faces $(t_j)_{0\leq j\leq m_{\mathcal{T}}}$
- A selected region of the source $\mathcal{S}$, represented as a binary mask $\mathfrak{M}$ over the source mesh faces

The output is a trajectory

$$\gamma : [0, 1] \to D(\mathcal{S}),$$

where $D(\mathcal{S})$ is the space of all deformations of $\mathcal{S}$, such that: $\gamma(0) = \mathcal{S}, \gamma(1) \odot \mathfrak{M} \simeq \mathcal{T}, \gamma(1) \odot (1 - \mathfrak{M}) \simeq \mathcal{S}$ (partially), and for all $t \in [0, 1]$, $\gamma(t)$ is (near)-isometric to $\mathcal{S}$.

3.3. Motivation

A naive solution requires matching the source mesh to the partial target mesh and then optimizing a path from a boundary value problem with the resulting shapes, using well-known deformation energies (e.g. [21]). However, both the matching, with its refinement, and the optimization of the final path are computation-intensive, and we lose the advantage of real-time approaches such as ARAP.

An appealing solution is given by Neural Jacobian Fields [1], in which a neural network learns per-face Jacobian matrices and recover the global deformation fields via integration (solving a Poisson equation). By learning on individual faces, this representation allows to learn the deformation of unregistered meshes. By masking the predicted Jacobian field, one can generate partial deformations. However, as shown in Figure 10, the resulting deformation introduces noticeable distortions.

In contrast to other approaches, *our method doesn't require any matching at test time*. Also, by learning local features to represent the deformation of input meshes, we supervise our method using *only global deformations*, and use the mask *only at test time*. PaNDaS generalizes to new poses and user-selected masks, and by interpolating features, it computes plausible partial deformations.

4. Method

The main component of our framework is a deformation model that takes as input a shape in neutral pose $\mathcal{S}$, and the target deformation of this shape $\mathcal{T}$, made of three modules illustrated in Fig. 4.

- A **global pose feature extractor** f_{θ_1}, which takes an input $\mathcal{T}$ and outputs a global feature $z = f_{\theta_1}(\mathcal{T}) \in \mathbb{R}^r$
- A **local deformation feature extractor** g_{θ_2}, which takes an input $\mathcal{S}$, and outputs a local deformation embedding $u_j = g_{\theta_2}(\mathcal{S})_j \in \mathbb{R}^l$, on each triangle t_j of $\mathcal{S}$

We then concatenate both information into a single feature vector $\omega_j = (z, u_j)$ (section 4.1). To generate the deformation of $\mathcal{S}$, we use:

- A **deformation generator**, h_{θ_3}, that takes as input the feature field ω_j and outputs a vertex-wise deformation $\mathcal{V} = (v_i) = h_{\theta_3}(\omega)$ of $\mathcal{S}$ (section 4.2), where $\hat{\mathcal{T}} = \mathcal{V} + \mathcal{S} \simeq \mathcal{T}$.

The network weights $\theta = (\theta_1, \theta_2, \theta_3)$ are optimized during training (we drop them from the notation when not necessary), using pairs $(\mathcal{S}, \mathcal{T})$ (section 4.3). We emphasize that *no masking is required during training*. Note that the feature field $(z, 0)$ corresponds to the case for which the pair is $(\mathcal{S}, \mathcal{S})$. At test time, when given a mask as input, we can mask the features z to generate a partial deformation. This simple strategy allows the model to generalize well to new poses and partial deformations, even if the model is only trained on full deformations (section 4.4).

4.1. Feature field of deformation

We describe in more detail the construction of our feature field of deformations.

Local feature extractor. We use a local feature extractor based on DiffusionNet [44] with a simple vertex-to-face aggregation strategy to compute per-face features $(u_j)_j$.

Global feature aggregation. To extract the global feature, we first extract local features $(b_j)_j$, and we propose a novel strategy to aggregate information, as an alternative to the max-pooling [13] or mass-weighted sum strategies.

Let $(b_j)_j$ be a set of face-wise local features on $\mathcal{T}$. Denote e^k the k-th eigenvector of the cotangent Laplacian $\Delta^{\mathcal{T}}$ of $\mathcal{T}$. Note that each e^k_j is a scalar for each face t_j. Now $b = (b_j)_j$ can be decomposed as $b_j = p_1 e^1_j + p_2 e^2_j + \dots$, with each p_k being a r-dimensional vector corresponding to the $k-$th eigenprojection of b, computed through a weighted average so that

$$p_k = \frac{1}{\text{Area}(\mathcal{T})} \sum_{j=1}^{m_{\mathcal{T}}} \text{Area}(t_j) b_j e^k_j. \tag{1}$$

Here, $m_{\mathcal{T}}$ denotes the number of faces in $\mathcal{T}$, and $\text{Area}(t_j)$ is the area of the j-th face of $\mathcal{T}$. In particular, p_1 is just the area-weighted average of b. Next, we concatenate the first s projections and pass the obtained vector through a linear layer to obtain a global encoding $z \in \mathbb{R}^r$. The global encoder f is the combination of local feature b extraction and our novel feature aggregator. We encode $z_{\mathcal{S}} = f(\mathcal{S})$ and $z_{\mathcal{T}} = f(\mathcal{T})$ at the same time, and the final latent vector of deformation is $z = z_{\mathcal{T}} - z_{\mathcal{S}}$. When $\mathcal{S} = \mathcal{T}$, $z = 0$.

Notably, to build features invariant to remeshing, one only needs to use a remeshing invariant feature extractor for both f and g. We use DiffusionNet [44].

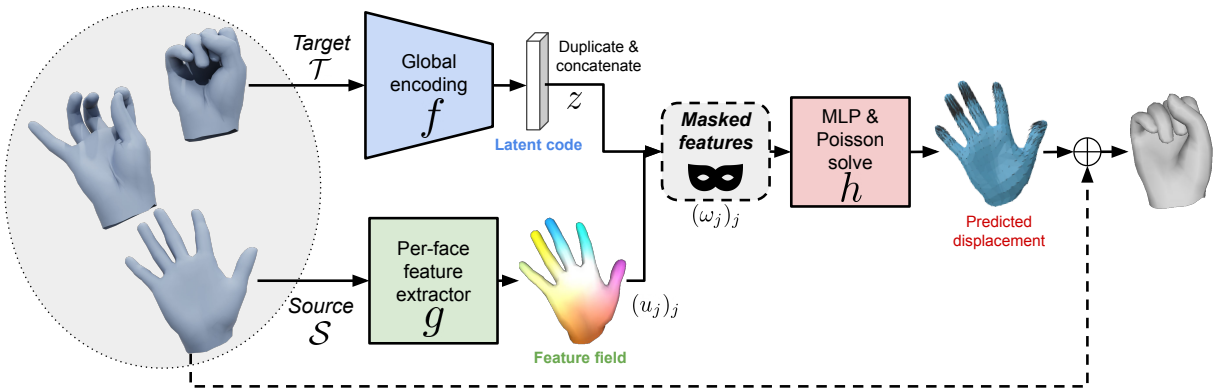


Figure 4. **Overview of the proposed architecture.** A pair of meshes is given as input: a source pose $\mathcal{S}$ and a target pose $\mathcal{T}$. Per-triangle features $(u_j)_j$ are learned on the source mesh and are enriched with a global encoding z of the target mesh to obtain a feature field $(\omega_j)_j$ over the source mesh. Finally, the deformation generator learns a Jacobian field from the feature field over the neutral mesh to predict a first-order, per-triangle displacement field $(v_i)_{1\leq i\leq n_{\mathcal{S}}}$.

Feature concatenation. As explained in the previous section, our feature field of deformation is the concatenation

$$\omega_j = (u_j, z).$$

We tried more complex strategies, but their improvement in reconstruction led to a weak generalization to new poses and partial deformation (see supplementary material).

4.2. Deformation generator

Neural Jacobian Fields. Given a Jacobian field J we search for the closest corresponding deformation $\mathcal{V}$, i.e., $\mathcal{V}$ that minimizes the Poisson equation

$$\min_{\mathcal{V}} ||J_j - (\nabla \mathcal{V})_j||^2,$$

which derives as:

$$\nabla^{\mathcal{S}} v = \nabla^T \mathcal{M} J,$$

with $\mathcal{M}$ the mass matrix of $\mathcal{S}$. This solution can be computed efficiently and in a differentiable way [1]. Because the features used to predict the Jacobians are invariant to remeshing, the resulting deformation will be invariant to remeshing, making our overall method robust to remeshing.

Jacobian generator. We first generate per-face Jacobian matrices $(J_j)_j$ of the final deformation field $\mathcal{V}$ before the Poisson solve. In the original paper, the authors attach the triangle centroids to a global encoding. But, we already have local information in local features $(u_j)_j$, so we instead predict the Jacobian field directly from the features $(\omega_j)_j$. The final deformation generator h is the composition of the Jacobian generator and the Poisson solve. It is differentiable and robust to remeshing by design.

4.3. Training

At training time, our model takes as input a shape in neutral pose $\mathcal{S}$, and a registered target deformation of this shape $\mathcal{T}$. The model predicts a reconstruction $\hat{\mathcal{T}}$ and is optimized by minimizing a discrepancy loss between $\mathcal{T}$ and $\hat{\mathcal{T}}$.

Reconstruction loss. We use the Mean Squared Error (MSE) as the reconstruction loss:

$$\mathcal{L}^{rec}(\mathcal{T}, \hat{\mathcal{T}}) = \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} \|y_i - (x_i + v_i)\|_2^2 \qquad (2)$$

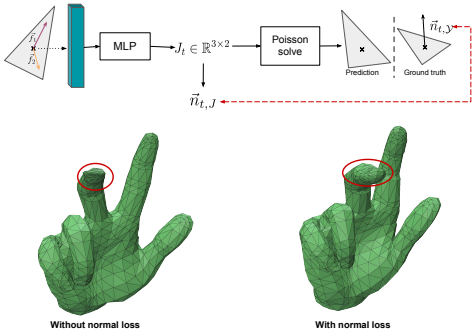


Figure 5. **Qualitative comparisons of the regularization effect when adding a loss term on the normal map.** On the bottom **left** is a mid-interpolation prediction for PaNDaS trained only with the MSE loss, on the **right** is a similar prediction with a model trained with a loss on the normal map.

Jacobian normal regularization. We regularize the normal map to obtain smoother deformations. Given a Jacobian J_j, we can compute the corresponding normal $\vec{n}_{j,J}$ (as a cross product) of the deformed mesh. While this intermediate normal is not necessarily the final normal of the face, we found that minimizing the cosine distance between this normal field and the ground truth target normal:

$$\mathcal{L}^n(\mathcal{T}, \hat{\mathcal{T}}) = \frac{1}{m_{\mathcal{T}}} \sum_{j=1}^{m} [1 - \vec{n}_{j,\mathcal{T}} \cdot \vec{n}_{j,J}], \qquad (3)$$

enhances the quality of deformations of the interpolations as shown in Fig. 5.

Our final target loss is given by:

$$\mathcal{L} = \mathcal{L}^{\text{rec}} + \lambda^n \mathcal{L}^n \tag{4}$$

4.4. Interpolation, Partial Deformation, Mixing Deformations with masking

Once the model is trained, one can perform basic arithmetic operations in the latent space to generate new deformations and motions. Notably, the generalization of our model enables the generation of partial motions by simple masking in the global latent space.

Interpolation. We interpolate between $\mathcal{S}$ and $\mathcal{T}$ in Fig. 9 by decoding a linear interpolation at each triangle t_j using the global variables z. Let $\omega_{j,\mathcal{S}} = (u_j, 0)$ (the encoded features for $\mathcal{S}$ and no pose deformation) and $\omega_{j,\mathcal{T}} = (u_j, z_{\mathcal{T}})$ (the encoded features for $\mathcal{S}$, and $\mathcal{T}$ pose deformation):

$$\gamma(t) = h((1-t) \times \omega_{j,\mathcal{S}} + t \times \omega_{j,\mathcal{T}}) \tag{5}$$

Partial deformation. We leverage PaNDaS locality of the feature field $\omega_j = (u_j, z)$ to obtain completely new poses. To obtain partial deformations, one can multiply the $z_{\mathcal{T}}$ part of the local features by a triangle-wise binary mask $\mathfrak{M} = (\mathfrak{M}_j)_{1 \leq j \leq m} \in \{0,1\}^m$, to only deform the masked part of the surface by inputting $\omega_j^{\text{partial}} = (u_j, \mathfrak{M}_j \odot z_{\mathcal{T}})$ to the deformation generator, where $\odot$ is the Hadamard product (element-wise) applied to the tensor of duplicated global variable z. With a linear interpolation of the masked features of the neutral pose and the partially deformed shape (see Fig. 1), it predicts the partial motion:

$$\gamma(t) = h((1-t) \times \omega_{\mathcal{S}} + t \times \omega^{\text{partial}}) \tag{6}$$

Pose mixing. More generally, we can take several global features $z^1, \dots, z^k$ encoded from several poses $\mathcal{T}^1, \dots, \mathcal{T}^k$, predict an augmented feature field $\omega^1 = (u, z^1)$ and several global encodings $z^1, \dots, z^k$, assign corresponding masks $\mathfrak{M}^1, \dots, \mathfrak{M}^k$ to each one and decode the feature field composed of u and $z^{\text{new}} = \frac{1}{k}\left(\mathfrak{M}^1 \odot z^1 + \dots + \mathfrak{M}^k \odot z^k\right)_{1 \leq j \leq k}$ to produce new deformations (see supplementary material).

Note that, because of the Poisson solve, there is no guarantee about the deformation being strictly restricted to the masked part. However, we observed that the norm of the deformation decreases rapidly the further away from the mask boundary. We report this observation in Fig. 7 by comparing the norm of the restricted Jacobians to the gradient of the deformation after masking.

5. Numerical Experiments

We showcase the efficiency of our method with three different experiments: the first one **compares full interpolations** obtained with the model inference. A second experiment compares the **partial deformation sequences on registered meshes**. Finally, a third experiment compares **partial deformation sequences on scans**.

Additionally, qualitative visualizations are given along the way and more results, such as animations performed using the model, are in the supplementary material videos.

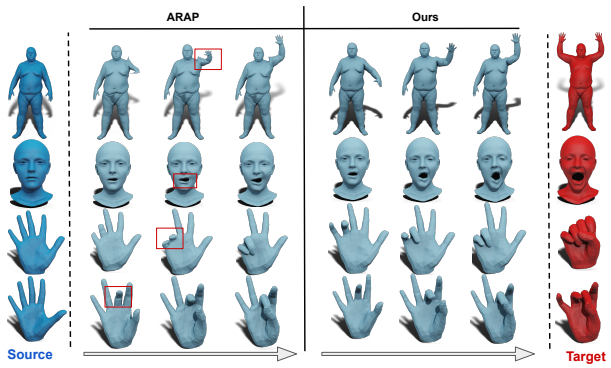


Figure 6. **Qualitative comparisons of partial deformations on DFAUST, COMA and MANO.** The As-Rigid-As-Possible deformations handle efficiently deformations that are close to linear (2nd and last row) but fail to adapt to large deviations.

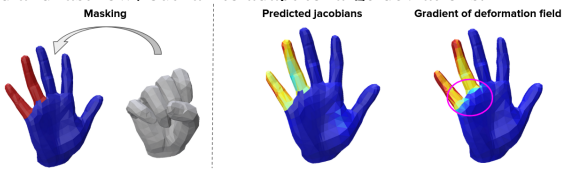


Figure 7. **Restriction of the deformation field.** Example of partial deformation, restricted to the red parts on the left. The corresponding norm of the predicted jacobians and the resulting gradient of the deformation field (after the Poisson solve). The deformation is localized and smoothed around the boundary of the mask (circled in purple).

5.1. Metrics and data

Datasets. We showcase the adaptability of our method with experiments on 3 different datasets.

MANO [40] is a hand mesh dataset comprising 30 hand identities, each in around 20 poses. We perform a prior Procrustes alignment [42] of the meshes.

DFAUST [9] is a 4D dataset of 10 human subjects that perform 13 different movements. It is made of 40,000 meshes, and we manually select frames in order to have elementary deformations for each sequence. The final dataset contains 308 meshes of static poses.

COMA [39] is a face mesh dataset of 12 individuals executing 12 "extreme" expressions. The full dataset contains more than 20,000 meshes and we sampled 500 static expressions. We also removed the eyeballs that were separate connected components of the face mesh.

Metrics. To evaluate mesh interpolations on registered data, we use the MSE and the mean Cosine Distance (Cosd) between the predicted normal field and the ground truth one. For interpolations of scans, we used the unregistered Hausdorff distance (HD) and Chamfer distance (CD) detailed in

the supplementary material.

5.2. Interpolation sequences

We evaluate the quality of latent interpolation sequences using comparisons with short movements in DFAUST and COMA. We report the MSE and mean Cosine distance of the generated sequences in Tab. 1 and Tab. 2 against ground truth movements. We compare the performances with latent interpolation from ARAPReg [23] and Spectral Meets Spatial (SMS) [12]. PaNDaS surpasses other methods on both metrics. To support the quantitative evaluation, we also show qualitative results in Fig. 9.

	MSE ↓			Cosd ↓		
	ARAPReg	SMS	**ours**	ARAPReg	SMS	**ours**
shake	6.4	5.4	**3.2**	0.17	0.12	**0.11**
chicken	6.0	5.6	**5.3**	**0.16**	0.17	0.18
knees	**1.8**	3.2	2.0	0.5	0.03	**0.01**
jumping	10.6	14.5	**5.8**	0.24	0.18	**0.09**
Mean	6.2	7.2	**4.1**	0.17	0.14	**0.10**

Table 1. **Interpolation evaluation on DFAUST sequences.**

	MSE ↓			Cosd (10^{-2}) ↓		
	NJF	ARAPReg	**ours**	NJF	ARAPReg	**ours**
bareteeth	0.22	0.07	**0.05**	2.5	1.3	**1.1**
cheeks in	0.21	0.09	**0.05**	2.6	1.7	**1.5**
mouth up	0.13	**0.06**	0.07	1.7	**1.2**	1.8
high smile	0.27	0.08	**0.05**	3.3	1.5	**1.1**
lips back	0.39	0.10	**0.05**	3.4	1.2	**0.8**
lips up	0.21	0.07	**0.04**	2.1	1.6	**1.2**
mouth side	0.57	0.10	**0.07**	3.5	**1.2**	1.3
mouth ext.	0.43	0.08	**0.07**	2.8	1.5	**1.2**
Mean	0.27	0.08	**0.06**	2.5	1.4	**1.3**

Table 2. **Interpolation evaluation on COMA sequences.**

5.3. Partial deformations and local manipulations

For partial deformations, predicted interpolations maintain a global smooth coherent mesh. In particular, it enables the prediction of new poses, unseen by the deformation generator for both registered and unregistered meshes.

5.3.1. Registered partial interpolations

We compare partial deformations with a localized As-Rigid-As-Possible (ARAP) vertex displacement. The implementation is borrowed from [34]. Providing accurate quantitative evaluation results for partial deformations is challenging as no ground truth is available. As a simple workaround, we propose to compute the same metrics as for full interpolations but restricted to vertices that belong to the parts we want to deform. For DFAUST, we predict partial deformations of the left half of the body and compare the performance against other methods: VCMC [60] based on latent points using a graph convolutional neural network (limited to a fixed topology) and localized ARAP optimization. Results are reported Tab. 3 and qualitative comparison with ARAP are presented in Fig. 6. Our method yields competitive results on both metrics and a clear improvement can be observed in the supplementary videos.

	MSE ↓			Cosd↓		
	VCMC	ARAP	**ours**	VCMC	ARAP	**ours**
shake	4.6	4.1	**2.3**	0.15	**0.09**	0.12
chicken	5.4	**3.2**	3.9	0.15	**0.09**	0.13
knees	2.3	1.5	**1.4**	0.06	0.02	**0.01**
jumping	11.9	8.8	**6.1**	0.26	0.12	**0.10**
Mean	6.0	4.4	**3.4**	0.15	**0.08**	0.09

Table 3. **Partial interpolation evaluation on DFAUST.**

We emphasize the versatility of this method: from a limited collection of different poses, our model enables a wide range of possible partial deformations. Moreover, this deep feature manipulation allows for precise animation of articulated meshes. Notably, it does not require additional skeleton information to produce meaningful movements.

5.3.2. Partial interpolations of scans

Our framework extends its application to unregistered meshes at test time thanks to the robustness of its different components. To support this, we propose a similar evaluation protocol as in the previous section but using the unregistered metrics (namely CD and HD). We compare against the original implementation of Neural Jacobian Fields (NJF) and ARAP. Results are gathered in Tab. 4 and show a clear superiority of PaNDaS for this partial interpolation task. In particular, NJF concatenates a global deformation code to vertex positions where our approach exclusively relies on the feature space, giving a better representation of the local geometry.

	CD (10^{-3})	HD
NJF	6.67	0.79
ARAP	6.03	0.71
Ours	**4.80**	**0.65**

Table 4. **Partial interpolation evaluation of DFAUST scans.**

5.4. Applications: mixing and shape statistics

Our approach also offers additional properties that open the way to various applications, as shown in Fig. 8.

Deformation mixing: As described in Sec. 4.4, PaNDaS can mix partial deformations in different parts of the shape. This is achieved using chosen masks on the deep features, which determine how each part is deformed. We can then generate new poses, as illustrated in Fig. 8.

Statistical Model of Deformations: PaNDaS can be used to estimate the mean shape from a collection of given shapes as well as partial means and principal components, as shown in Fig. 8. Interestingly, these components can be used to build a basis for controllable and localized movements and, because PaNDaS relies on modules that are ro-

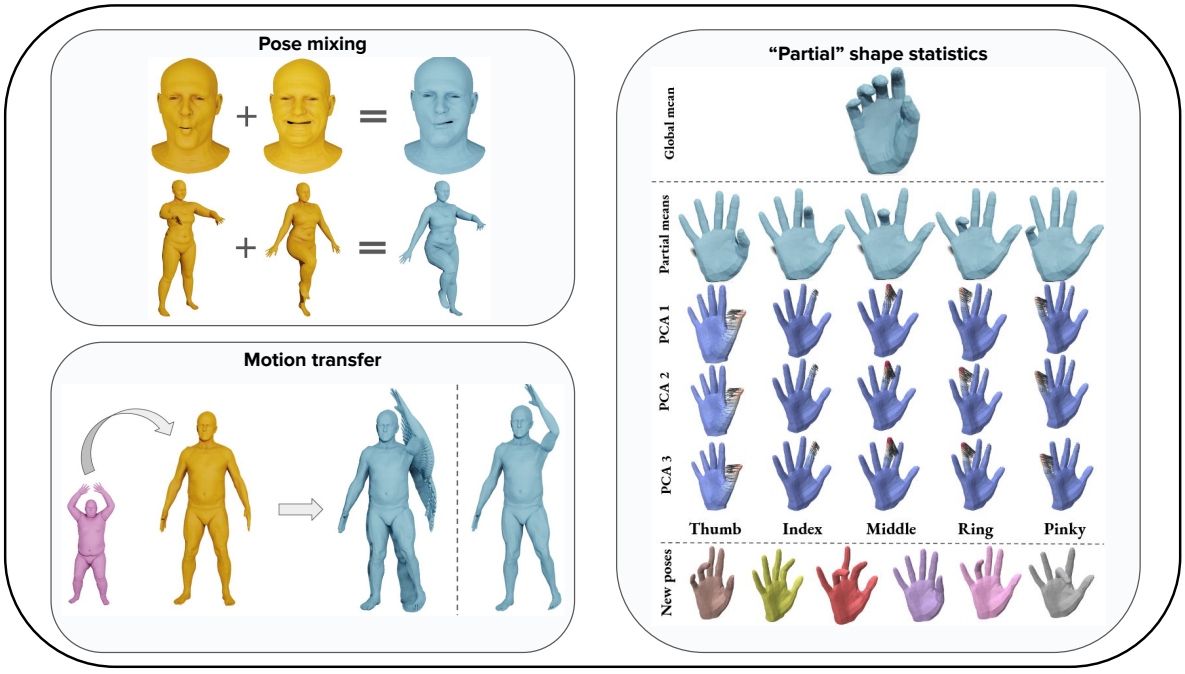


Figure 8. **Applications of PaNDaS–Pose mixing**: By combining the feature fields of different poses, we can combine deformations to generate new poses. **Transfer**: The deformation encoder is trained to encode the pose which allows partial motion transfer. **Statistics**: The learned feature field is a tensor with Euclidean structure on which statistical insights can be mapped back to the space of deformations.

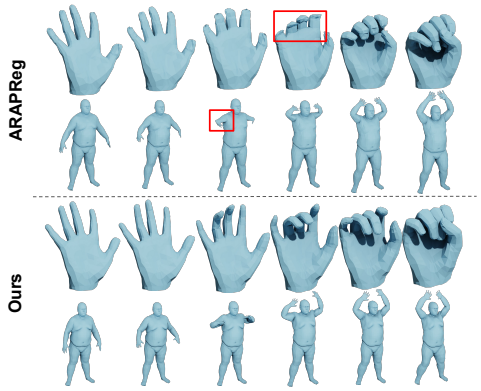


Figure 9. **Pose interpolation examples.** We compare latent linear interpolation between ARAPReg (top row) and PaNDaS (bottom row). The latter handles better large, non linear deformations.

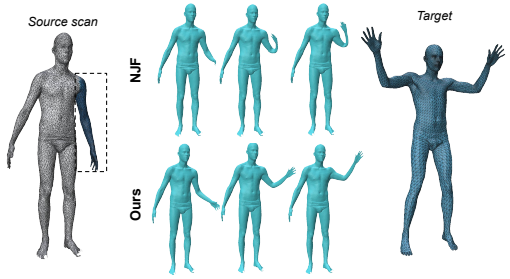


Figure 10. **Partial interpolation of a scan from FAUST [8].**

bust against remeshing, full and partial statistical predictions are possible on unregistered meshes such as scans.

Motion transfer. An interesting property arising from our framework is the ability to transfer motions on selected parts while maintaining the integrity of the whole mesh. Due to a training process focused on learning pose deformations, the deformation encoder naturally disentangles the identity and pose of the target mesh.

6. Limitations and conclusion

Limitations. Although PaNDaS is invariant to mesh topology, like other state-of-the-art methods, it is trained with registered meshes. An exciting possibility of future work is to explore recent strategies for unregistered training [19, 58], to train PaNDaS on unregistered data. Then, while the quality of partial deformations is satisfying, our simple masking strategy of latent vectors can create local artifacts on the boundary of the selected parts. Improving this strategy, by using weighted masking or learning the mix of latent vectors, is left for future work.

Conclusion. In this paper, we presented PaNDaS, a neural method for partial deformations of meshes. With combined global and local latent vectors, it enables the computation of partial deformations of surface meshes. Moreover, our learning strategy allows to learn non-rigid deformations of shapes, and demonstrates state-of-the-art generalization capabilities. We highlighted the efficiency of our method for complete and partial shape interpolations. Finally, PaNDaS can predict new poses by combining masked features and compute local shape statistics.

Acknowledgments

This work was partially supported by the project 4DSHAPE ANR-24-CE23-5907 of the French National Research Agency (ANR). This work was also supported by the ERC Consolidator Grant 101087347 (VEGA).

References

[1] Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Trans. Graph.*, 41(4), 2022. 2, 3, 4, 5

[2] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, page 157–164, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 2

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005. 2, 3

[4] Martin Bauer, Philipp Harms, and Peter W. Michor. Sobolev metrics on shape space of surfaces. *Journal of Geometric Mechanics*, 3(4):389–438, 2011. 2, 3

[5] Martin Bauer, Nicolas Charon, Philipp Harms, and Hsi-Wei Hsieh. A numerical framework for elastic surface matching, comparison, and interpolation. *International Journal of Computer Vision*, 129:2425 – 2444, 2020. 3

[6] Mirza Faisal Beg, Michael Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61:139–157, 2005. 3

[7] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 3

[8] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, 2014. IEEE. 8

[9] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[10] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, page 11–20, Goslar, DEU, 2006. Eurographics Association. 2

[11] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[12] D. Cao, M. Eisenberger, N. El Amrani, D. Cremers, and F. Bernard. Spectral meets spatial: Harmonising 3d shape matching and interpolation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3658–3668, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 2, 3, 7

[13] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 77–85. IEEE, 2017. 4

[14] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. Limp: Learning latent shape representations with metric preservation priors. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, page 19–35, Berlin, Heidelberg, 2020. Springer-Verlag. 2, 3

[15] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021. 2, 3

[16] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *ACM Transactions on Graphics (SIGGRAPH)*, 2023. 3

[17] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *ECCV*, 2018. 3

[19] Sara Hahner, Souhaib Attaiki, Jochen Garcke, and Maks Ovsjanikov. Unsupervised representation learning for diverse deformable shape collections. In *International Conference on 3D Vision 2024*, 2024. 3, 8

[20] Emmanuel Hartman, Emery Pierson, Martin Bauer, Nicolas Charon, and Mohamed Daoudi. BaRe-ESA: A riemannian framework for unregistered human body shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14181–14191, 2023. 2, 3

[21] Emmanuel Hartman, Yashil Sukurdeep, Eric Klassen, Nicolas Charon, and Martin Bauer. Elastic shape analysis of surfaces with second-order sobolev metrics: A comprehensive numerical framework. *International Journal of Computer Vision*, 2023. 4

[22] Emmanuel Hartman, Emery Pierson, Martin Bauer, Mohamed Daoudi, and Nicolas Charon. Basis restricted elastic shape analysis on the space of unregistered surfaces. *International Journal of Computer Vision*, 2024. 2, 3

[23] Q. Huang, X. Huang, B. Sun, Z. Zhang, J. Jiang, and C. Bajaj. Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages

5795–5805, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 3, 7

[24] Ian H Jermyn, Sebastian Kurtek, Eric Klassen, and Anuj Srivastava. Elastic shape matching of parameterized surfaces using square root normal fields. In *European conference on computer vision*, pages 804–817. Springer, 2012. 3

[25] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM transactions on graphics (TOG)*, 26(3):71–es, 2007. 2

[26] Martin Kilian, Niloy J. Mitra, and Helmut Pottmann. Geometric modeling in shape space. *ACM Trans. Graph.*, 26(3): 64–es, 2007. 2, 3

[27] Sebastian Kurtek, Eric Klassen, John C Gore, Zhaohua Ding, and Anuj Srivastava. Elastic geodesic paths in shape space of parameterized surfaces. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1717–1730, 2011. 3

[28] Hamid Laga, Qian Xie, Ian H Jermyn, and Anuj Srivastava. Numerical inversion of srnf maps for elastic shape analysis of genus-zero surfaces. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2451–2464, 2017. 3

[29] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 3

[30] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rossi, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. In *Proceedings Shape Modeling Applications, 2004.*, pages 181–190. IEEE, 2004. 2

[31] Or Litany, Alexander M. Bronstein, Michael M. Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1886–1895, 2017. 3

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2, 3

[33] Zhaoyang Lyu, Jinyi Wang, Yuwei An, Ya Zhang, Dahua Lin, and Bo Dai. Controllable mesh generation through sparse latent point diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 271–280, 2023. 3

[34] Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. Smooth non-rigid shape matching via effective dirichlet energy optimization. In *2022 International Conference on 3D Vision (3DV)*, pages 495–504, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 7

[35] Sanjeev Muralikrishnan, Niladri Shekhar Dutt, Siddhartha Chaudhuri, Noam Aigerman, Vladimir Kim, Matthew Fisher, and Niloy J. Mitra. Temporal residual jacobians for rig-free motion transfer, 2024. 3

[36] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, pages 598–613, 2020. 2

[37] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 2, 3

[38] Dafei Qin, Jun Saito, Noam Aigerman, Groueix Thibault, and Taku Komura. Neural face rigging for animating and retargeting facial meshes in the wild. In *SIGGRAPH 2023 Conference Papers*, 2023. 3

[39] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741, 2018. 6

[40] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2, 3, 6

[41] Ann-Malin Schmidt, Ingrid Estefany Peraza, and Yordan Kyosev. Evaluation of automated skeleton fitting to 4d human body scan data using open-source smpl- and osso models. In *3DBODY.TECH Journal - International Journal of 3D Body Technologies*, 2024. 3

[42] Peter Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 6

[43] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986. 2

[44] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 01(1), 2022. 4

[45] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023. 3

[46] Alon Shtern, Matan Sela, and Ron Kimmel. Fast blended transformations for partial shape registration. *Journal of Mathematical Imaging and Vision*, 60:913 – 928, 2016. 2

[47] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, page 109–116, Goslar, DEU, 2007. Eurographics Association. 2, 3

[48] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 2

[49] Ramana Sundararaman, Nicolas Donati, Simone Melzi, Etienne Corman, and Maks Ovsjanikov. Deformation recovery: Localized learning for detail-preserving deformations. *ACM Transactions on Graphics (TOG)*, 43(6):1–16, 2024. 2

[50] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3d mesh models. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5841–5850, 2018. 3

[51] Qingyang Tan, Ling-Xiao Zhang, Jie Yang, Yu-Kun Lai, and Lin Gao. Variational autoencoders for localized mesh deformation component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6297–6310, 2022. 3

[52] Jiapeng Tang, Lev Markhasin, Bi Wang, Justus Thies, and Matthias Nießner. Neural shape deformation priors. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 3

[53] Ju Tao, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM transactions on graphics (TOG)*, 24(3):561–566, 2005. 2

[54] Tianhao Xie, Eugene Belilovsky, Sudhir Mudur, and Tiberiu Popa. Dragd3d: Realistic mesh editing with rigidity control driven by 2d diffusion priors, 2024. 3

[55] Pan Xingang, Tewari Ayush, Leimkuhler Thomas, Li Lingjie, Meka Abhimitra, and Theobalt Christian. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 3

[56] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (Oral)*, pages 6184–6193, 2020. 3

[57] Haitao Yang, Bo Sun, Liyan Chen, Amy Pavel, and Qixing Huang. Geolatent: A geometric approach to latent space design for deformable shape generators. *ACM Trans. Graph.*, 42(6), 2023. 3

[58] Haitao Yang, Xiangru Huang, Bo Sun, Chandrajit L. Bajaj, and Qixing Huang. Gencorres: Consistent shape matching via coupled implicit-explicit shape generative models. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 8

[59] Seungwoo Yoo, Kunho Kim, Vladimir G. Kim, and Minhyuk Sung. As-Plausible-As-Possible: Plausibility-Aware Mesh Deformation Using 2D Diffusion Priors. In *CVPR*, 2024. 3

[60] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 3, 7