Deformation Recovery: Localized Learning for Detail-Preserving Deformations

RAMANA SUNDARARAMAN, LIX, Ecole Polytechnique, France
NICOLAS DONATI, Ansys, France
SIMONE MELZI, University of Milano-Bicocca, Italy
ETIENNE CORMAN, Université de Lorraine, CNRS, Inria, France
MAKS OVSJANIKOV, LIX, Ecole Polytechnique, France

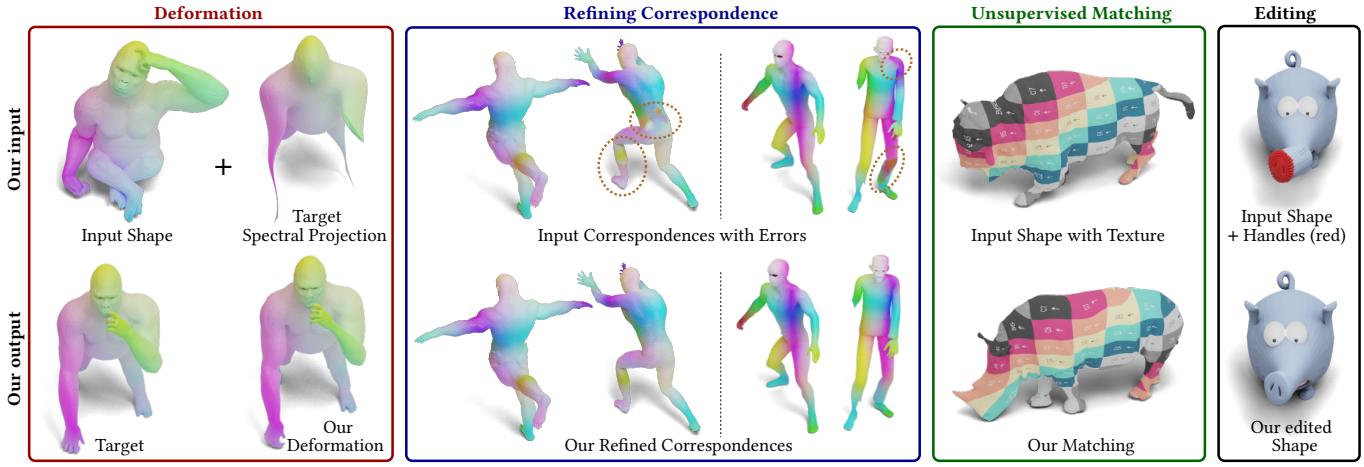


Fig. 1. We introduce a novel data-driven approach to produce high-quality shape deformations. Conditioned on a coarse representation of the deformation as the input signal (left top), we learn a detail-preserving deformation (right-bottom). Leveraging this novel deformation framework, we perform three downstream tasks: refining an approximate shape correspondence, unsupervised shape matching, and interactive editing, as shown in subsequent columns.

We introduce a novel data-driven approach aimed at designing high-quality shape deformations based on a coarse localized input signal. Unlike previous data-driven methods that require a global shape encoding, we observe that detail-preserving deformations can be estimated reliably without any global context in certain scenarios. Building on this intuition, we leverage Jacobians defined in a one-ring neighborhood as a coarse representation of the deformation. Using this as the input to our neural network, we apply a series of MLPs combined with feature smoothing to learn the Jacobian corresponding to the detail-preserving deformation, from which the embedding is recovered by the standard Poisson solve. Crucially, by removing the dependence on a global encoding, every *point* becomes a training example, making the supervision particularly lightweight. Moreover, when trained on a class of shapes, our approach demonstrates remarkable generalization across different object categories. Equipped with this novel network, we explore three main tasks: refining an approximate shape correspondence, unsupervised deformation and mapping, and shape editing.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; *Neural networks*; *Shape modeling*; Shape representations.

Additional Key Words and Phrases: Shape deformation, shape correspondence, Spectral geometry processing

ACM Reference Format:
Ramana Sundararaman, Nicolas Donati, Simone Melzi, Etienne Corman, and Maks Ovsjanikov. 2024. Deformation Recovery: Localized Learning for Detail-Preserving Deformations. In *Proceedings of SIGGRAPH Asia '24.* ACM, New York, NY, USA, Article 219, 15 pages. https://doi.org/10.1145/3687968

1 INTRODUCTION

Estimating meaningful deformations of surfaces is a classical problem in computer graphics [Bechmann 1994; Terzopoulos et al. 1987], with applications in several downstream tasks such as surface mapping and registration, character reposing, and handle-based editing to name a few [Aigerman and Lipman 2013; Amberg et al. 2007; Bouaziz et al. 2013; Innmann et al. 2016; Li et al. 2008; Trappolini et al. 2021]. Due to its broad applicability, numerous techniques have been developed to address this task [Botsch and Sorkine 2008].

While early approaches relied on numerical techniques [Botsch and Sorkine 2008], recent methods, following the ubiquitous trend,

Authors' Contact Information: Ramana Sundararaman, sundararaman@lix.polytechnique.fr, LIX, Ecole Polytechnique, Paris, , France; Nicolas Donati, nicolas.donati@polytechnique.edu, Ansys, Paris, , France; Simone Melzi, melzi@di.uniroma1.it, University of Milano-Bicocca, Milan, , Italy; Etienne Corman, etienne.corman@inria.fr, Université de Lorraine, CNRS, Inria, LORAINE, , France; Maks Ovsjanikov, maks@polytechnique.fr, LIX, Ecole Polytechnique, Paris, , France.

SIGGRAPH Asia '24, December 03–06 2024, Tokyo, JPN

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/3687968

leverage structured data-driven priors to estimate plausible deformations [Aigerman et al. 2022; Dodik et al. 2023; Groueix et al. 2018a; Sundararaman et al. 2022]. Despite significant ongoing efforts, this problem remains challenging. A key difficulty with most existing data-driven methods lies in constructing a feasible *latent space* [Maesumi et al. 2023]. This typically requires amassing a collection comprising all plausible deformations [Aigerman et al. 2022; Groueix et al. 2018b], thus necessitating a significant amount of training data. Furthermore, even when such data is available, representing shapes using a *global encoding* typically ties existing approaches to specific *shape categories*, and thus requires extensive re-training for cross-category generalization.

To that end, instead of relying on a global shape encoding, we condition our predictions on a *coarse*, local deformation input signal. This choice is motivated by the observation that, in specific scenarios, as we will demonstrate in this paper, constructing an appropriate local input signal is sufficient to learn category-agnostic high-quality deformations. Building on the recent success of Jacobian fields as learnable deformation representation [Aigerman et al. 2022], we design a network that predicts Jacobian matrices per simplex. The input to our network is thus a field of Jacobian matrices, averaged to vertices around a one-ring neighborhood, representing a coarse deformation. Our network comprises a series of MLPs coupled with a smoothing operator, which takes coarse Jacobians as input and produces detailed Jacobians. Unlike the globally-conditioned approach in [Aigerman et al. 2022], our network has fully shared weights and is applied *independently* at each simplex.

Our network prediction is trained by supervising Jacobians corresponding to the detailed mesh, mesh vertex positions, and an integrability loss on Jacobians. Due to the purely local and fully shared nature of our networks, they can be trained using a handful of shapes, as each *simplex* becomes a training instance. In summary, our deformation framework learns to *produce* the input deformation at each vertex, represented as Jacobians, without utilizing any global information about the shape. For this reason, we refer to our network as the *Local Jacobian Network* (LJN).

Given this general framework, the key question lies in the choice of the appropriate coarse input signal. In this work, we consider two general scenarios: shape correspondence and shape editing. For map refinement and shape correspondence, given a fixed source shape and a target shape representing a *coarse* deformation of the source, we first express the coordinate function of the target shape using the low-frequency eigenfunctions of its Laplace-Beltrami Operator (LBO) to obtain a smooth approximation of its geometry [Lévy and Zhang 2010], referred to as the *spectrally projected* shape. This is a natural representation supported by shape correspondence methods that use the functional map framework [Ovsjanikov et al. 2012]. We then define the Jacobians from the source shape to the spectrally projected target shape as the coarse deformation signal, which forms the input to our network. For shape editing, we use the rotation matrices at each face, averaged to the incident vertex, as the input signal to produce Jacobians corresponding to a valid shape. The network, training data, and supervision remain consistent across all input signals.

Since LJN relies on localized input signals, unlike a globally shape-aware network, it is inherently agnostic to shape categories [Guerrero et al. 2018]. As a result, there is no need to amass a large training dataset or perform category-specific training, since, as mentioned above, each point (i.e., local region in the shape) becomes a training sample. Therefore, we limit our dataset to only 60 shape pairs in all our supervised training experiments. Trained on these 60 pairs of human shapes, LJN demonstrates remarkable generalization across object categories in recovering minimal distortion embeddings.

We explore three main *tasks* using our framework: map refinement, unsupervised non-isometric shape correspondence, and interactive shape editing as shown in Figure 1. In the first task, given an approximate vertex-based correspondence as input, we refine it using our deformation method. Specifically, we construct a coarse input signal by projecting the target geometry onto the spectral basis, from which we first recover the detail-preserving deformation using LJN. Subsequently, we perform an NNSearch in $\mathbb{R}^3$ to obtain an improved point-wise map (i.e., registration). Evaluated across standard near-isometry and non-isometry benchmarks, LJN achieves significant improvements in accuracy, coverage, map smoothness, and reduction of map inversions compared to existing map-refinement techniques, all while being non-iterative and fully differentiable.

In our second task, we do not assume an input approximate correspondence but rather simultaneously estimate the map and the shape deformation. For this, we train LJN alongside a Deep Functional Map (DFM) [Cao et al. 2023; Donati et al. 2022a; Sun et al. 2023] network on a collection of animal shapes with distinct mesh connectivities from the SMAL [Li et al. 2022] and SHREC-20 [Dyke et al. 2020] datasets. Our joint network improves map coverage and map smoothness compared to current state-of-the-art unsupervised correspondence techniques.

Finally, as our third task, to demonstrate that LJN can generalize to different local input signals, we focus on interactive shape editing. Specifically, a user deforms a shape by displacing selected vertices, and our goal is to solve for an embedding with minimal distortion. In this setup, we utilize the closest rotation matrix to the prescribed deformation as the input signal and learn to produce Jacobians corresponding to a valid shape. At the inference time, this task is akin to ARAP [Sorkine 2006], however, instead of iteratively updating the Jacobians, we learn a Jacobian that corresponds to detail-preserving deformation in a single feed-forward pass. We demonstrate that LJN generalizes to arbitrary object categories, producing more plausible and minimal distortion embeddings in comparison to ARAP.

2 RELATED WORKS

Shape deformation is a thoroughly researched yet continually evolving field. Given the extensive literature, we refer the readers to existing surveys [Botsch and Sorkine 2008], and we focus our discussion on aspects directly relevant to our work.

2.1 Neural Deformation Techniques

In recent years, data-driven deformation techniques have shown promising results in shape registration and correspondence tasks [Deprelle et al. 2019; Groueix et al. 2018b; Sundararaman et al. 2022;

Trappolini et al. 2021]. Earlier models primarily learned displacement fields over a fixed template [Groueix et al. 2018b; Kanazawa et al. 2015; Litany et al. 2017a], relying on a global latent code derived from point-based [Qi et al. 2016], graph-based [Wang et al. 2019], or mesh-based representations [Hanocka et al. 2019], as well as implicit surfaces [Bhatnagar et al. 2020]. The proliferation of extensive training datasets [Osman et al. 2020; Varol et al. 2017] has facilitated the development of more sophisticated, template-free methods [Aigerman et al. 2022; Trappolini et al. 2021]. Although neural deformation techniques predominantly focus on learning *displacement fields*, several alternatives have emerged, including cage-based [Dodik et al. 2023; Yifan et al. 2020], control-point-based [Kurenkov et al. 2017; Sundararaman et al. 2022], vector-field-based [Jiang et al. 2020], and differential-based methods [Aigerman et al. 2022]. Notably, the Neural Jacobian Field approach [Aigerman et al. 2022] stands out due to efficiency and capacity to simulate realistic deformations. Consequently, this representation has been applied to various downstream applications, such as text and image-guided deformation [Gao et al. 2023; Yoo et al. 2024]. However, to the best of our knowledge, nearly all data-driven deformation techniques utilize a *global encoder* for learning across collections. While these methods are discretization-agnostic [Aigerman et al. 2022; Groueix et al. 2018b], they often fail to generalize to unseen object categories or necessitate category-specific training. Additionally, they require a significant number of training shapes that share a common 1:1 correspondence. To overcome these limitations, our work employs a localized input signal for supervised and unsupervised training, thereby rendering our approach more data-efficient and category-agnostic.

2.2 Map-Refinement

Map-refinement techniques are typically iterative and can be broadly classified into two categories: spatial and spectral refinement techniques. The former considers the embedding of shapes in Euclidean space, while the latter focuses on the spectral domain, spanned by the first k-eigenfunctions of the Laplace-Beltrami Operator [Dubrovina and Kimmel 2010]. A classical approach in the spatial domain includes Iterative Closest Point (ICP) [Besl and McKay 1992; Chen and Medioni 1992] and its specific adaptations [Amberg et al. 2007; Bouaziz et al. 2013; Gelfand et al. 2005]. While the aforementioned approaches treat individual points separately, probabilistic approaches such as Gaussian Mixture Models (GMMs) [Hirose 2023; Myronenko and Song 2010] and Optimal Transport [Mandad et al. 2017; Solomon et al. 2016] have been well-explored. However, these methods struggle to converge when the shapes in the input pair are geometrically distant. Spectral approaches [Ovsjanikov et al. 2012] are less affected by geometric proximity but often fail to recover high-frequency details due to the truncation of basis. To counteract this, upsampling techniques [Li et al. 2022; Melzi et al. 2019b; Ren et al. 2021, 2018] augment the spectral frequency during each iteration. Yet, their dependence on the alignment of intrinsic quantities limits generalization to significant non-isometries. While spatial coupling has been explored to address non-isometries [Ezuz and Ben-Chen 2017; Ezuz et al. 2019a,b; Magnet et al. 2022], these methods are costly and challenging to integrate into differentiable frameworks. In contrast, our approach achieves a more accurate point-wise map through a single feed-forward pass and a back-substitution being orders of magnitude faster.

2.3 Functional Map and Deformation

Corman et al. [Corman et al. 2017] were first to define distortion between pairs of meshes with differing connectivity using the functional map framework. Similarly to our approach, the authors proposed to pull back the intrinsic metric of the target shape onto the source. However, in that work, the authors used the shape difference operator [Rustamov et al. 2013] and constructed the embedding via Poisson solve [Panozzo et al. 2014]. Since the Functional Map operator lacks extrinsic awareness, they employed an offset surface to fully recover the embedding. In contrast, our work involves pulling back the *coordinate functions* in the truncated basis and *learning* the Jacobian corresponding to the coordinate functions in the full basis. More recently, the basic deep functional map approach [Donati et al. 2020; Litany et al. 2017b] has been enhanced with spatial awareness, often defined via properness [Ren et al. 2021] to enforce cycle consistency [Cao et al. 2023; Sun et al. 2023]. Leveraging this observation, recent [Jiang et al. 2023] and concurrent work [Cao et al. 2024] have explored various deformation models to improve point-wise map extraction. Our work aligns with these developments, as our unsupervised deformation-mapping experiments demonstrated. Our deformation model is *learned* alongside map estimation without requiring any correspondence information.

3 PRELIMINARIES

Notations: We represent shapes as compact 2-dimensional Riemannian manifolds $\mathcal{M}$ possibly with boundary $\partial\mathcal{M}$. The space of square integrable functions on $\mathcal{M}$ is noted $\mathcal{L}^2(\mathcal{M})$ and equiped with the scalar product $\langle f, g\rangle = \int_{\mathcal{M}} fg \, \mathrm{d}\mu$. We denote the tangent plane at p as $T_p\mathcal{M}$. We discretize $\mathcal{M}$ as a triangle mesh $\mathcal{S} := \{\mathcal{V}, \mathcal{F}\}$ with $\mathcal{V}$ vertices and $\mathcal{F}$ faces. We refer to the list of the coordinates of the vertices $\mathcal{V} \in \mathbb{R}^3$ as the embedding of the shape. On the manifold $\mathcal{M}$, we discretize the Laplace-Beltrami Operator (LBO), denoted Δ, using the standard cotangent-based discretization [Pinkall and Polthier 1993]. We adopt the face-based discretization of the tangent plane [Azencot et al. 2013]. Since Δ is a symmetric operator, by solving the generalized eigenvalue problem $\Delta\Psi = \lambda M\Psi$, we obtain the LBO eigen-basis $\Psi = [\psi_1 \dots \psi_k]$, where, M is the diagonal-lumped vertex-mass matrix [Meyer et al. 2003]. These eigen-basis are an orthonormal basis (w.r.t. M) for the truncated subspace of $\mathcal{L}^2(\mathcal{M})$ of smooth, low-frequency functions. Given an embedding $\mathcal{V}$, we consider its spectral projection $\bar{\mathcal{V}} = \Psi\Psi^\dagger\mathcal{V}$, where $\Psi^\dagger$ is the Moore-Penrose pseudo-inverse of Ψ. Analogously, we use $\bar{\mathcal{S}}$ to denote the shape whose vertices are $\bar{\mathcal{V}}$.

For each face, we define a possibly non-orthonormal frame $\mathcal{E} := [\vec{e}_1, \vec{e}_2, \vec{N}]^T$, which is a 3×3 matrix. The frame comprises the first two edge vectors of the face and the face normal. We denote the frames for all faces as $\mathbf{E} := [\mathcal{E}_1 \dots \mathcal{E}_{|\mathcal{F}|}]$. For any given point $x \in \mathcal{M}$, a local deformation can be defined using a Jacobian matrix $J_x \in \mathbb{R}^{3\times3}$, the matrix of all the first-order partial derivatives of the deformation. In the discrete setting, given our face-based discretization of the tangent plane, Jacobians are piece-wise constant per-face. We denote

$\mathbf{J} := [J_1 \dots J_{|\mathcal{F}|}]$ to be the Jacobian across all faces represented in matrix form. If $x \in F_i$, we have $J_x = J_i$, where F_i is the i^{th} face.

Given two discretized shapes $\mathcal{S}_1$ and $\mathcal{S}_2$ (often called source and target shapes) with respectively n_1 and n_2 vertices, we can write a correspondence $\varphi : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ between those shapes as a binary matrix $\Pi \in \mathbb{R}^{n_1 \times n_2}$ where $\Pi[i, j] = 1$ denotes j^{th} vertex on $\mathcal{S}_2$ being the image of i^{th} vertex on $\mathcal{S}_1$. When the two shapes are in 1:1 correspondence, Π_{12} is an identity matrix.

Functional Map. The functional map pipeline, introduced in [Ovsjanikov et al. 2012], is an efficient and compact representation for maps between shapes. More specifically, let $\varphi : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ be a point-wise map, and Π_{12} its corresponding binary matrix. The pull-back operator associated with this map, expressed in the LBO truncated eigen-basis and denoted as $\mathbf{C}_{21} \in \mathbb{R}^{k \times k}$, is referred to as the Functional Map [Ovsjanikov et al. 2012]. $C_{21} : \mathcal{L}^2(\mathcal{S}_2) \rightarrow \mathcal{L}^2(\mathcal{S}_1)$ acts as a linear operator between the square integrable functions on the respective shapes. This can be derived from the binary matrix $\mathbf{C}_{21} = (\Psi_1)^{\dagger}\Pi_{12}\Psi_2$. Where, $\Psi_1^{\dagger}$ is the Moore-Penrose pseudoinverse of Ψ_1. Given the functional map C_{21}, in the simplest setup, we can compute an associated Π_{12}^* via a nearest-neighbors search [Ovsjanikov et al. 2017]:

$$\Pi_{12}^* = \underset{\Pi_{12}}{\operatorname{argmin}} \|\Psi_2 C_{21} - \Pi_{12}\Psi_1\|_F^2 \tag{1}$$

Jacobian-based deformation. Let $\mathcal{S}_1, \mathcal{S}_2$ be shapes with 1:1 corresponding vertices and same connectivity but two different vertex embeddings $\mathcal{V}_1, \mathcal{V}_2 \in \mathbb{R}^3$. We can consider the deformation between them as a per-vertex coordinate re-assignment. The linear part of this deformation quantifying the change in edge-vectors is referred to as the Jacobian $\mathbf{J}_{12}$ between $\mathcal{S}_1 \rightarrow \mathcal{S}_2$. This change of edge vectors [Botsch et al. 2006; Sumner and Popović 2023]) is explicitly given by:

$$\mathbf{J}_{12} = \mathbf{E}_1^{-1}\mathbf{E}_2 \tag{2}$$

where $\mathbf{E}_\ell = [\mathcal{E}_1 \dots \mathcal{E}_{|\mathcal{F}_\ell|}]$, $\ell \in 1, 2$, are the frames corresponding to all faces, rewritten in matrix form. Note that while quantifying the change in edge vectors is sufficient to define $\mathbf{J}_{12}$, solving Eqn (2) becomes under-determined (see Chap 3.1.2 [Sumner 2005]). While [Sumner and Popović 2023] overcomes this by tetrahedralization of each face, we simply use the *unit-normal* vector, as changes along normal directions are inconsequential for a surface undergoing deformation in $\mathbb{R}^3$. Given a Jacobian $\mathbf{J}_{12}$, we can compute its closest embedding in the least-square sense by solving the following Poisson equation:

$$\Delta_1 \mathcal{V}_2 = \nabla_1^T A_1 \mathbf{J}_{12} \tag{3}$$

$\nabla_1^T A_1$ is the divergence operator where ∇_1 is the gradient operator and A_1 is the area element of each face of $\mathcal{S}_1$. Since Δ_1 is only semi-definite, the solution $V\mathcal{V}_2$ is not unique and is only valid up-to global translation [Botsch et al. 2006].

4 PROPOSED METHOD

We begin by describing the general framework for training and performing inference with LJN in Section 4.1. In Section 4.2, we discuss the supervised training strategy for map refinement, followed by the inference strategy that alleviates the necessity for consistent connectivity in Section 4.3. In Section 4.4, we introduce a novel method to simultaneously learn the deformation and mapping between shape pairs with different triangulations. Finally, in Section 4.5, we discuss the utility of LJN for handle-based shape editing.

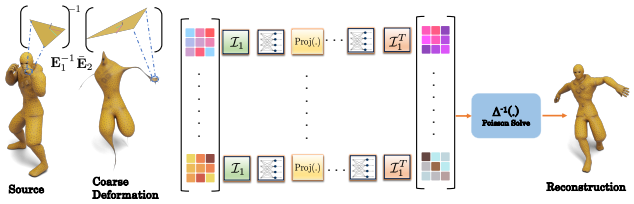
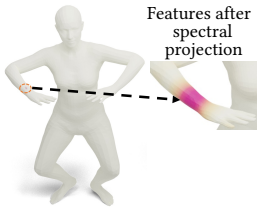


Fig. 2. Illustration of our learning framework with spectrally projected input signal. Given a source shape and the coarse deformation signal, we feed in individual triangle deformations, as Jacobians, averaged to incident vertices. Then, we apply series MLPs coupled with spectral projection in the *feature-space* to recover detail-preserving deformation.

4.1 Detailed Deformation Learning from Coarse Signals

Given a source shape $\mathcal{S}_1$ and a coarse deformation input signal, we aim to learn the detailed deformation of the source shape, expressed as Jacobian $\mathbf{J}_{12}$. Note that our coarse input signal might not correspond to a valid shape. When these Jacobians are integrated to produce an embedding, they must correspond to a plausible shape. We let $\mathcal{S}_2$ be the shape which the target detailed Jacobian $\mathbf{J}_{12}$ corresponds to. We refer to $\mathcal{S}_2$ as the target shape. We let $\Theta_{12} \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the $d-$dimensional coarse input signal defined at vertex. In the case of Jacobians defined on faces, we first average them to vertices with the operator $\mathcal{I}$ mentioned below. Recall that, unlike [Aigerman et al. 2022], we do not rely on a global encoding of the target shape but instead condition the input to our network using Θ_{12} alone.



Given this input signal, we apply MLPs per-vertex in tandem with a *spectral projection* layer to learn the detailed Jacobian corresponding to $\mathcal{S}_2$. The spectral projection layer projects the learned features to the eigenbasis of the LBO operator as given in Equation (6). We observed that this enables effective information sharing in a small local neighborhood around each vertex. For visualization purposes, we consider a single Jacobian as a Dirac signal defined at the i^{th} vertex as shown in the inset figure (left). Then, upon projecting this signal to the eigenbasis of the LBO operator, the features are *smoothly* spread out in a small neighborhood around the vertex of interest as shown in the inset figure (right). Our learning process can then be described as follows:

$$\hat{\mathbf{J}}_{12} = \mathcal{I}_1^T \mathcal{G}_6(\dots(\text{Proj}_1(\mathcal{G}_1(\Theta_{12})))) \tag{4}$$

The above equation sequentially applies an MLP, $\mathcal{G}(.)$, whose specifications are provided in the Supplementary. Here, $\text{Proj}(\cdot) := \Psi_1\Psi_1^T M_1 \cdot$ expresses features in the orthonormal basis defined by Ψ_1. The operator $\mathcal{I}_1$ is a sparse row-stochastic matrix that averages quantities from faces to vertices. The non-zero entries are corresponding to row i of $\mathcal{I}_1$ sum up to 1 and are all equal to $\frac{1}{f_i}$, where

f_i denotes the number of faces incident on vertex i. Its transpose, $\mathcal{I}_1^T$ averages the per-vertex predictions back to faces. Given the ground-truth Jacobian $\mathbf{J}_{12}^* : \mathcal{S}_1 \rightarrow \mathcal{S}_2$, to train our network, we minimize a loss function consisting of three terms as follows:

$$\mathcal{L}_{\text{Tr}} = \alpha_1 \left\|\mathbf{J}_{12}^* - \hat{\mathbf{J}}_{12}\right\|_F^2 + \alpha_2 \left\|\mathcal{V}_2 - \hat{\mathcal{V}}_1\right\|_2^2 + \alpha_3 \left\|\mathbf{J}_{12}^* - \mathbf{E}_1^{-1}\hat{\mathbf{E}}\right\|_F^2. \quad (5)$$

Here $\hat{\mathcal{V}}_1 = \Delta_1^{-1}\nabla_1^T A_1 \hat{\mathbf{J}}_{12}$ is the embedding corresponding to the predicted Jacobian recovered via the Poisson solve (cf. Equation 3). The first two terms are used to supervise the positions and predicted Jacobians w.r.t the ground truth, while the third term penalizes discrepancies between the *integrated* Jacobian and the ground truth Jacobian. More precisely, we compute the Jacobian corresponding to the recovered embedding $\hat{\mathcal{V}}_1$ using the corresponding non-orthonormal frame as $\mathbf{E}_1^{-1}\hat{\mathbf{E}}_1$ and minimize the discrepancy with the ground truth Jacobian.

Our learning pipeline, along with one possible coarse input signal used in this paper, is visualized in Figure 2. In summary, given some input Jacobians, we first average them to vertices, spectrally project them and then train our to reconstruct the detailed Jacobian between the source and target shapes. At inference time, the embedding can be obtained via a simple feedforward pass and a Poisson solve (cf. Eqn 3). Task-specific inference is discussed in subsequent sections.

4.2 Supervised Learning from Spectral Inputs

In this section, we discuss the construction of a coarse deformation input signal geared for shape registration-related tasks. Our choice of representation for the input signal is designed to be easily integrated into existing shape correspondence frameworks [Ovsjanikov et al. 2012]. To construct the input signal, we first express the coordinate function of the target shape using the first k eigenfunctions of the LBO, referred to as the *spectrally projected* shape. Let $\bar{\mathcal{S}}_2$ be the shape obtained by applying spectral projection to $\mathcal{S}_2$, and $\bar{\mathcal{V}}_2$ be its vertex set. Then, the closed-form expression for $\bar{\mathcal{V}}_2$ is given as follows:

$$\text{Proj}(\mathcal{V}_2) = \bar{\mathcal{V}}_2 = \Psi_2 \Psi_2^{\dagger} \mathcal{V}_2 = \Psi_2 \Psi_2^T M_2 \mathcal{V}_2 \quad (6)$$

Where $\Psi_2^{\dagger} = \Psi_2^T M_2$ because of orthonormality with respect to M_2. Denoting the frames per face corresponding to the spectrally projected shape as $\bar{\mathbf{E}}_2$, the input signal to our network *during training* is defined as follows:

$$\Theta_{12} := \mathcal{I}_1 \mathbf{J}_{12} = \mathcal{I}_1 \mathbf{E}_1^{-1} \bar{\mathbf{E}}_2 \quad (7)$$

Discussion. For a learning problem to be well-posed, distinct inputs should produce distinct outputs. We demonstrate this qualitatively on pairs of shapes, as shown in Figure 3. We consider a fixed source shape and a vertex of interest in this source shape as the "seed" vertex, say the i^{th} vertex. We then plot the Frobenius norm of the difference between the input signal at the i^{th} vertex and all the remaining vertices, expressed as $\|\Theta_{i,12} - \Theta_{j,12}\|_F \forall i \neq j$, in the first row. Similarly, in the second row, we plot the difference between the ground truth Jacobian i^{th} vertex and all the remaining vertices, $\|J_{i,12}^* - J_{j,12}^*\|_F \forall i \neq j$. The similarity in the distributions of the norm at each vertex between input and ground truth across different deformations asserts the well-posed nature of our learning problem. Additionally, in Figure 4, we provide an intuition on how this input signal behaves over near and non-isometric deformations.

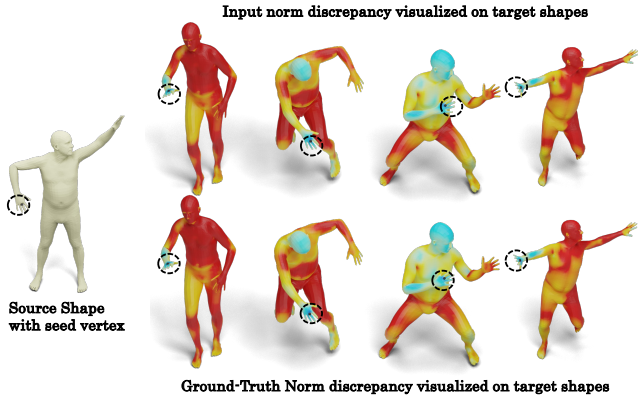


Fig. 3. Illustration of the fidelity of our training framework. We select a *seed* vertex, marked with a dotted circle. The Frobenius norm of the difference between the Jacobian at this seed vertex and all other vertices is plotted. The similarity in the distribution between the input (projected Jacobian) and the ground truth suggests the well-posed nature of our learning problem.

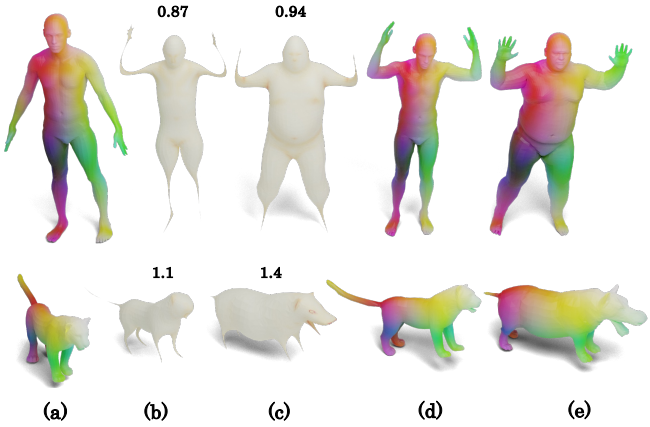


Fig. 4. Visual insights on how our input signal is affected by near-isometric and non-isometric deformations. In the first row, we show a human shape undergoing two deformations into a similar pose but taken by two subjects, one similar and one not. We compare the discrepancy between the input Jacobian and the ground truth Jacobian as the Frobenius norm (plotted in the second and third columns). We repeat this for a pair of animals with a significantly higher degree of non-isometry between them. We observe that our input signal Θ_{12}, remains comparable to the ground-truth across different levels of non-isometry.

4.3 Inference: Meshes with differing connectivity.

The requirement for shapes to be in 1-1 correspondence when constructing the input signal is impractical for real-world applications. To address this, we utilize the Functional Map framework to model deformations between shapes with differing connectivities. Therefore, given a pair of shapes $\mathcal{S}_1, \mathcal{S}_2$ with differing connectivity and

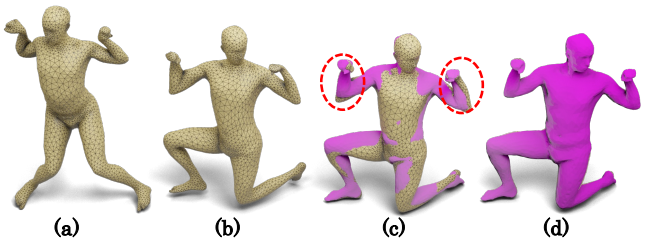


Fig. 5. Given a source (a) and a deformation (Jacobian), we compute the embedding following Eqn 3 and visualize in (b). While the deformation is near-perfect, the surfaces do not '*align*' as shown in (c) where (b) is juxtaposed to the target shape. Computing the embedding via Eqn 10 yields a surface that is geometrically closer to the target as shown in (d).

a functional map C_{21} between them, we aim to extract the deformation $\hat{\mathbf{J}}_{12}$ between them. The Functional Map can be estimated via an existing Deep Functional Map (DFM) framework [Cao et al. 2023; Sun et al. 2023], or using our unsupervised method detailed in Section 4.4.

Different from training time, at inference, we first write the spectral projection of the coordinate function of the target shape and 'pull' back the embedding function to the source shape using the functional map, as follows,

$$\bar{\mathcal{V}}_1 = \Psi_1 C_{21} \Psi_2^{\dagger} \mathcal{V}_2 \tag{8}$$

The above equation expresses the coordinate function of the target shape in the LBO eigenbasis of the source shape, with the functional map acting as the change of basis operator. Since $\mathcal{V}_1$ and $\bar{\mathcal{V}}_1$ share the same connectivity, we can define the input to our network as given in Equation (7) and perform a feedforward pass to obtain $\hat{\mathbf{J}}_{12}$. To retrieve the embedding, instead of simply solving the Poisson equation (3), we also leverage the point-wise map Π_{12} arising from the Functional Map (using c.f Eqn (1)). We do this to address numerical errors in the learned Jacobian which might lead to detailed deformation but "un-aligning" surface as highlighted in Figure 5. To this end, our embedding recovery from Jacobian is solved via the following minimization:

$$\hat{\mathcal{V}}_1 = \min_{\hat{\mathcal{V}}_1} \alpha_4 \left\| \hat{\mathcal{V}}_1 - \Pi_{12} \mathcal{V}_2 \right\|_{M_1}^2 + \alpha_5 \left\| \Delta_1 \hat{\mathcal{V}}_1 - \nabla_1^T A_1 \hat{\mathbf{J}}_{12} \right\|_{M_1}^2 + \left\| \hat{\mathcal{V}}_1 \right\|_{\Delta_1}^2 \tag{9}$$

Here, the first term ensures that the recovered embedding remains geometrically close to the target surface, while the second term promotes the recovered embedding to respect the learned Jacobians. The final term encourages smoothness in the recovered embedding by favoring a minimum norm solution. Setting the gradient of the above expression w.r.t $\mathcal{V}_1$ to be zero yields:

$$(\Delta_1 + \alpha_4 M_1 + \alpha_5 \Delta_1^T M_1 \Delta_1) \hat{\mathcal{V}}_1 = \alpha_4 M_1 \Pi_{12} \mathcal{V}_2 + \alpha_5 \Delta_1^T M_1 \nabla_1^T A_1 \hat{\mathbf{J}}_{12} \tag{10}$$

Since the LHS in the above expression depends only on the Laplacian and the mass matrix, it can be pre-factored. If registration is the final goal, one can obtain a refined pointwise map using the deformed embedding $\hat{\mathcal{V}}_1$. This can be done by the following minimization:

$$\Pi_{12}^* = \underset{\Pi_{12}^*}{\operatorname{argmin}} \| \hat{\mathcal{V}}_1 - \Pi_{12}^* \mathcal{V}_2 \|_2^2 \tag{11}$$

which yields, $\Pi_{12}^* = \text{NNSearch}(\hat{\mathcal{V}}_1, \mathcal{V}_2)$.

4.4 Unsupervised Deformation Learning and Mapping

In this section, we propose an unsupervised training methodology on a collection of shapes without consistent triangulation, leveraging the Functional Map framework. Unlike in Section 4.3, we do not assume to have access C_{21} to as input, but rather *learn* it alongside $\mathbf{J}_{12}$. To this end, we employ a recent two-branch Deep Functional Map (DFM) [Cao et al. 2023; Sun et al. 2023] approach and combine it with LJN. We denote the feature extractor within the standard DFM [Sharp et al. 2020] corresponding to source and target shapes as $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. Between the input pair, we estimate the Functional Map in a differentiable manner, similar to [Cao et al. 2023] as follows,

$$\hat{C}_{21} = \Psi_1^{\dagger} \tilde{\Pi}_{12} \Psi_2 \tag{12}$$

Where $\tilde{\Pi}_{12} = \text{Softmax}\left(\mathcal{D}_1 \mathcal{D}_2^T / \tau\right)$, is the soft pointwise map with $\tau = 0.07$ being the temperature parameter. Using the $\hat{C}_{21}$ estimated above, we compute $\bar{\mathcal{V}}_1$ using Eqn 8 and use it to compute Θ_{12} via Eqn (7). Then, we perform a feed-forward pass through LJN to get predicted Jacobian $\hat{\mathbf{J}}_{12}$. We train the two-branch functional map network and our LJN jointly by minimizing the following loss:

$$\mathcal{L}_{\text{un}} = \sum_{i \in \{1,2\}, j \in \{2,1\}} \left\| \mathbf{C}_{i,j}^T \mathbf{C}_{i,j} - \mathbf{I} \right\|_F^2 + \left\| \mathbf{C}_{i,j} - \hat{\mathbf{C}}_{i,j} \right\|_F^2 + \mathcal{L}_{\text{J}}(\mathbf{J}_{j,i}) \tag{13}$$

Where, $C_{i,j}$ is the predicted Functional Map and $\hat{C}_{i,j}$ is the functional map arising from the soft-pointwise map (c.f Eqn (12)). Note that the summation in the above equation optimizes the Functional Map and Jacobian in both directions. For simplicity and consistency, we elaborate on the minimization objective for $\mathbf{J}_{1,2}$ while noting that the same applies analogously to $\mathbf{J}_{2,1}$. Our unsupervised deformation objective $\mathcal{L}_{\text{J}}(\mathbf{J}_{1,2})$ is given as follows:

$$\mathcal{L}_{\text{J}}(\hat{\mathbf{J}}_{12}) = \left\| \mathring{\mathbf{J}}_{12} - \hat{\mathbf{J}}_{12} \right\|_F^2 + \alpha_6 \left\| \hat{\mathbf{J}}_{12} - \mathcal{H}_1 \hat{\mathbf{J}}_{12} \right\|_F^2 + \alpha_7 \left\| \det(\hat{\mathbf{J}}_{12}) - 1. \right\|^2 \tag{14}$$

Here, $\mathring{\mathbf{J}}_{12}$ is the Jacobian corresponding to the deformation prescribed by $\tilde{\Pi}_{12}$. More specifically, we first compute the deformation of the source embedding as $\mathring{\mathcal{V}}_1 = \tilde{\Pi}_{12} \mathcal{V}_2$. Then, denoting the frame corresponding to mesh $\mathring{\mathcal{S}}_1 = \{\mathring{\mathcal{V}}_1, \mathcal{F}_1\}$ as $\mathring{\mathbf{E}}_1$, we define $\mathring{\mathbf{J}}_{12} := \mathbf{E}_1^{-1} \mathring{\mathbf{E}}_1$. The second term in the loss function is a smoothness prior, enforcing minimal norm differences between predicted Jacobians corresponding to adjacent faces with $\mathcal{H}_1$ denoting the unsigned face-face incidence matrix. The third term promotes volume preservation, an important regularization term, which is deemed empirically useful when supervising with $\tilde{\Pi}_{12}$ due to the effect of truncation of basis.

4.5 Interactive Editing

To demonstrate that the LJN deformation framework is applicable to tasks beyond shape correspondence, we consider the task of interactive shape editing. In this scenario, a user prescribes deformation

at selected vertices (handles). Our goal is to produce a minimal distortion embedding consistent with the user-prescribed deformation. Interactive editing can often result in meshes with degenerate elements, making the computation of differential operators and their eigendecomposition ill-defined. Therefore, we use the closest rotation matrix to the ground truth Jacobians as our input signal. This scenario is more challenging since the input deformation signal does not correspond to a valid shape, i.e, there is no guarantee that an embedding exists whose differentials correspond to the input signal. To compute the input signal, we apply polar decomposition to the ground truth Jacobian $\mathbf{J}_{12}^* = \mathcal{Q}_{12}\mathcal{W}_{12}$, where $\mathcal{Q}_{12}$ is the orthonormal matrix and $\mathcal{W}_{12}$ is an SPD matrix. Since our training pairs consist of valid shapes, $\mathbf{J}_{12}^*$ is always full rank and admits a unique polar decomposition. Therefore, to compute Θ_{12} (cf. Eqn 7), we use $\mathcal{Q}_{12}$ instead of $\mathbf{E}_1^{-1}\bar{\mathbf{E}}_2$. The rest of the training pipeline remains identical to our discussions in Section 4.2. In summary, given rotation matrices averaged over the one-ring neighborhood, we learn the Jacobian corresponding to the detail-preserving deformation.

4.6 Implementation details

All our supervised training is performed over the first 50 shapes from SCAPE [Anguelov et al. 2005] dataset and the first 10 shapes from the FAUST dataset [Bogo et al. 2014]. Our choice stems from the compactness of the dataset size and the range of deformation between each pair. We use a 4-layered MLP of 256 feature dimensions with the spectral projection followed by 2 layers without spectral projection with 9-dimension output, which is our Jacobian. We train with ADAM optimizer for a total of 50 epochs, with an initial learning rate of 1e-3 decayed progressively to 1e-5. The coefficients across different objective functions used in this paper are respectively $\alpha_1 = 1$, $\alpha_2 = 10.$, $\alpha_3 = 2$, $\alpha_4 = 20000$, $\alpha_5 = 150000$, $\alpha_6 = 20.$ and $\alpha_7 = 10.$ respectively. We used 128 eigenfunctions for smoothing the learned feature while we used 40 eigenfunctions to project the embedding of the shape. While training our network, as form of data augmentation, we vary the number of eigenfunctions used to construct the spectrally smooth embedding. Our full implementation and data will be released upon publication.

5 EXPERIMENTAL RESULTS

In this section, we demonstrate the utility of our deformation module, LJN, across three tasks - namely, Map Refinement in Section 5.2, Unsupervised Deformation and Mapping in Section 5.3 and finally Interactive Editing in Section 5.4. Before delving into individual tasks, we first elaborate the metrics used to gauge different methods in Subsection 5.1. For supervised experiments, our network is trained on 60 shape pairs from FAUST [Bogo et al. 2014] and SCAPE [Anguelov et al. 2005] datasets.

5.1 Evaluation Metrics

For evaluating shape correspondence, we used the standard mean geodesic error [Kim et al. 2011]. Additionally, we assessed map inversion, the Dirichlet energy [Magnet et al. 2022] of the map, and coverage [Ren et al. 2018]. For completeness, we provide elaboration on these metrics below.

(1) Map Inversion: This metric measures the change in orientation induced by the map. For surface correspondence techniques, matching via nearest neighbors search can result in a 180° twist, commonly referred to as the candy-wrapper effect (see [Abulnaga et al. 2022] Fig. 2). To quantify this, we use map-inversion metric, which penalizes the deviation in the sign of the dot product between the normal on the target shape and the mapped shape. The mapped shape normal is estimated by *permuting* the target vertices via Π_12 and estimating the normals using the orientation prescribed by the *source mesh*. More precisely, let $\mathcal{S}_1 : \{\mathcal{V}_1, \mathcal{F}_1\}$ and $\mathcal{S}_2 : \{\mathcal{V}_2, \mathcal{F}_2\}$ be the source and the target mesh. We let $\vec{n}_1^i$ be the normal corresponding to the mesh $\hat{\mathcal{S}}_1 : \{\Pi_{12}\mathcal{V}_2, \mathcal{F}_1\}$, whose vertices image of $\mathcal{V}_2$ under Π_{12} with winding defined by $\mathcal{F}_1$. Since Π_{12} acts as a map between vertices, let $\vec{n}_2^j$ be the normal at vertex index j on $\mathcal{S}_2$, such that $\Pi_{12}[i, j] = 1$. Note $\vec{n}_2^j$ is defined by the orientation of $\mathcal{S}_2$. Then, map inversion is quantified as

$$\text{Inv} := \frac{\sum_{j=1}^{\|\mathcal{V}_2\|} \mathbf{1}\left(\sum_{i=1}^{m} \vec{n}_1^i \cdot \vec{n}_2^j < 0\right)}{\|\mathcal{V}_2\|} \times 100 \tag{15}$$

(2) Dirichlet Energy: Following the standard convention used in previous works [Ezuz et al. 2019c; Magnet et al. 2022], we define map smoothness as $\|\Pi_{12}\mathcal{V}_2\|_{\Delta_1}$. For uniformity, we scale all shapes to have a unit area before computing this energy.
(3) Coverage: Similar to [Ren et al. 2018], we define coverage as the percentage sum of the unique area of the vertex image given by $\Pi_{12}\mathcal{V}_2$. Note that since we scale shapes to have a unit area, normalization is not required.

5.2 Map-Refinement

5.2.1 Overview. Given an approximate shape correspondence, either as a fuzzy point-wise map or a functional map [Ovsjanikov et al. 2012], our task is to produce a refined point-wise map. To that end, we first construct the spectrally projected input signal (c.f Eqn 8), then, perform a feedforward pass over LJN to obtain the Jacobian, from which we compute the embedding using Eqn 10. Finally, the refined point-wise map is obtained via Eqn 11.

5.2.2 Setup. As our input correspondence, we use two recent Deep Functional Map methods [Cao et al. 2023; Sun et al. 2023] as our baseline maps, for which open-source implementations are available. We evaluated our approach and all baselines on the FAUST, SCAPE [Ren et al. 2018] and the SHREC-19 dataset [Donati et al. 2020] for near-isometric shape correspondence. We used the inter-class category of Deforming-4D [Magnet et al. 2022] and SMAL-remeshed [Li et al. 2022] for non-isometric correspondence. We use geodesic error [Kim et al. 2011] (cm), map coverage [Ren et al. 2018] (%), map inversions (%) and map smoothness as evaluation metrics.

5.2.3 Comparisons. We compare our method against axiomatic *iterative* map-refinement methods: ZoomOut (ZO) [Melzi et al. 2019b]

Refinement	Near-Isometry												Non-Isometry							
	FAUST [Ren et al. 2018]				SCAPE [Ren et al. 2018]				SHREC-19 [Melzi et al. 2019a]				SMAL [Li et al. 2022]				DT4D-Inter [Magnet et al. 2022]			
	Geod	Inv	DirE	Cov	Geod	Inv	DirE	Cov	Geod	Inv	DirE	Cov	Geod	Inv	DirE	Cov	Geod	Inv	DirE	Cov
Init	1.7	7.4	3.2	80.8	2.3	8.1	4.2	76.2	5.1	7.9	6.6	72.6	4.7	13.6	13.9	62.0	6.8	9.8	17.8	61.9
Spec Proj	3.1	11.4	3.1	58.5	3.0	9.9	3.2	58.3	5.8	10.5	3.5	53.9	4.9	**10.1**	**3.5**	58.0	7.4	10.5	4.4	47.0
α_5=0	1.7	7.2	3.0	81.2	2.1	5.7	3.4	78.2	5.1	7.9	4.5	72.9	4.7	14.2	7.5	62.9	6.6	9.4	8.1	63.1
ZO	1.7	10.9	3.1	83.3	2.2	8.2	3.6	80.8	5.1	9.7	4.0	74.2	5.6	11.9	24.2	69.9	6.6	11.1	11.6	69.7
DZO	1.8	10.4	3.2	79.8	2.4	10.4	3.9	78.3	5.1	10.7	3.9	71.6	4.9	11.1	9.6	68.5	6.3	9.9	9.7	70.1
SmFM	2.2	14.7	2.8	75.5	3.8	19.7	3.0	69.9	6.1	20.7	**3.1**	65.2	6.1	12.0	4.2	61.1	7.5	21.2	**4.0**	59.7
Ours	**1.5**	**4.1**	**2.7**	**83.7**	**1.9**	**1.8**	**2.8**	**81.3**	**4.8**	**5.9**	3.3	**75.8**	**4.1**	10.2	**3.5**	**71.5**	**6.1**	**6.4**	**4.0**	**71.1**

Table 1. Summary of quantitative comparisons for map refinement with different baselines across various datasets and averaged over two different initializations: CDFM [Sun et al. 2023] and ULRSSM [Cao et al. 2023]. The best scores are highlighted in bold, and the second-best scores are highlighted in blue.

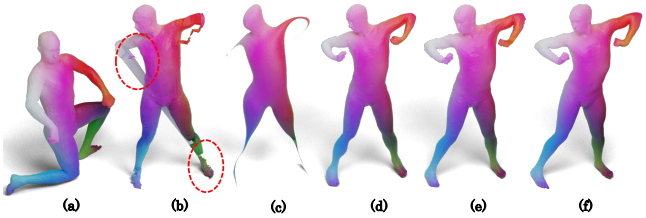


Fig. 6. Illustrating *how* LJN achieves high map-refinement accuracy: Given (a) the source shape and (b) an input map containing high-frequency artifacts, (c) we convert it to the Functional Map and *pull* the coordinates. The pulled coordinates are devoid of high-frequency artifacts. Subsequently, (d) our reconstruction and (e) the obtained map are detail-preserving. (f) denotes the target shape. We color-code correspondence from refined maps.

, Discrete Optimization (DZO) [Ren et al. 2021], and Smooth Functional Maps (SmFM) [Magnet et al. 2022]. More details on the evaluation and implementation of baselines are provided in the Supplementary. Additionally, we consider two more plausible alternatives. First, we set $\alpha_5 = 0$, so that the recovered embedding (viz., the refined map) does not rely on the learned Jacobians. Secondly, we use the coordinate function 'pulled' by Functional Map (cf. Equation 8) to recover the map. i.e, we solve for the map as $\Pi_{12} = \text{NNSearch}(\mathcal{V}_1, \mathcal{V}_2)$ (c.f Eqn 11). We denote this baseline as "Spec Proj".

5.2.4 Discussion. The quantitative evaluation averaged across two initializations, is summarized in Table 1. Our approach consistently outperforms iterative baseline methods ZO, DZO, and SmFM across various benchmarks and evaluation metrics. Notably, despite being trained solely on human shapes, our method demonstrates remarkable generalization in modeling animal deformations from the SMAL dataset. Additionally, maps recovered by our method exhibit less distortion compared to those from SmFM [Magnet et al. 2022], which *explicitly* aims to minimize Dirichlet energy. This improvement is attributed to the truncation of the basis inherent in their approach, whereas our deformation operates in the spatial domain ($\mathbb{R}^3$). We illustrate the effects of map smoothness and overall map refinement accuracy in Figure 7. Furthermore, we show qualitative reconstruction across more object categories with different triangulations in Figure 8. Finally, we provide an intuitive explanation in Figure 6 to

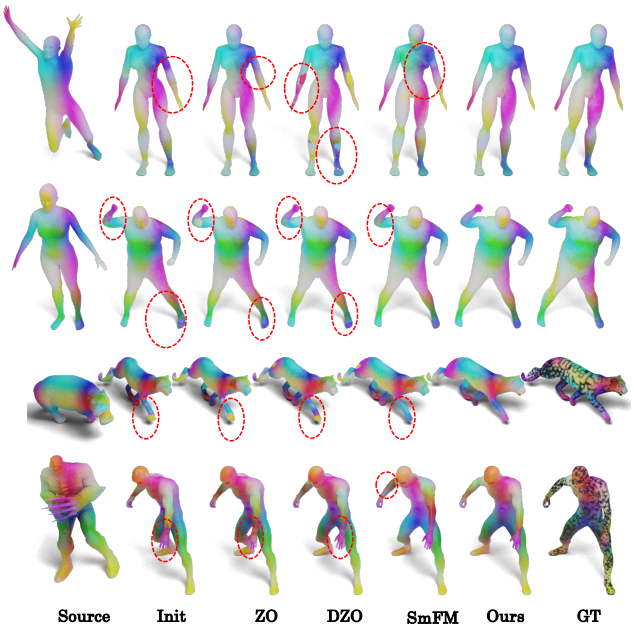


Fig. 7. Qualitative results for map refinement. The first two rows depict near-isometric shape pairs from SHREC'19 [Melzi et al. 2019a] dataset, while the last two rows show non-isometric datasets: SMAL [Donati et al. 2020] and Deforming4D [Magnet et al. 2022]. We note that LJN not only preserves correspondence but also improves the smoothness of the produced map. For initialization, we used CDFM [Sun et al. 2023] for the first two rows and ULRSSM [Cao et al. 2023] for the latter two.

illustrate *how* our approach achieves superior performance. Overall, LJN shows state-of-the-art results, which is remarkable due to the difficulty of the problem and the presence of strong recent baselines.

5.3 Unsupervised Deformation and Mapping

Given a collection of shapes with different triangulations, we aim to produce the deformation and the map as detailed in Section 4.4. We use the SMAL [Li et al. 2022] and SHREC20 [Dyke et al. 2020]

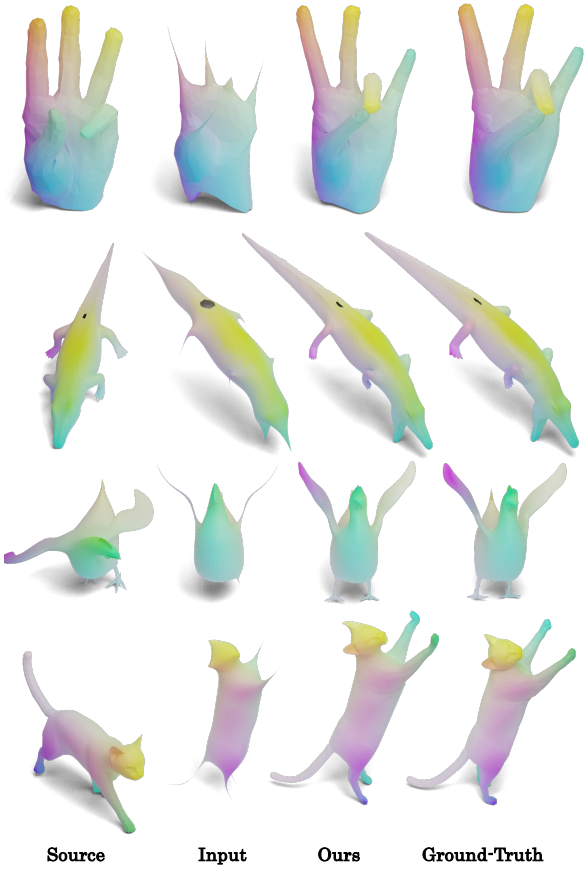


Fig. 8. Generalization of LJN to different object categories. Trained on 60 pairs of human shapes, our LJN shows robust generalization to unseen categories with near-perfect reconstruction. Meshes in the first row are from the MANO dataset [Romero et al. 2017], the subsequent three rows are from the Deforming-4D Dataset [Magnet et al. 2022], and the last row is from the TOSCA dataset [Ren et al. 2018]. Shapes in rows 2-5 do not share identical triangulations. To estimate the deformation, we used a 40×40 FMap. The embedding is then recovered following Section 4.3.

datasets for two separate experiments, which include a variety of largely non-isometric animal shapes. For the former setup, we train on 28 shape pairs in SMAL across 5 categories and evaluate on 20 shape pairs spanning 3 unseen categories. In Figure 9, we provide the qualitative deformations obtained from unsupervised training on SMAL. Despite being supervised with a noisy training signal (first-row), LJN produces detail-preserving deformation. Due to the limited training data in SHREC20, following [Cao et al. 2023], we perform zero-shot learning on each pair. Owing to the absence of dense correspondence in the dataset, we only provide coverage and smoothness as the quantitative measures. We compare with recent DFM methods ULRSSM [Cao et al. 2023], CDFM [Sun et al. 2023], and a variant of our approach where point-wise map is extracted without our deformation module (c.f Eqn 1), denoted as W/o LJN in Figure 10.

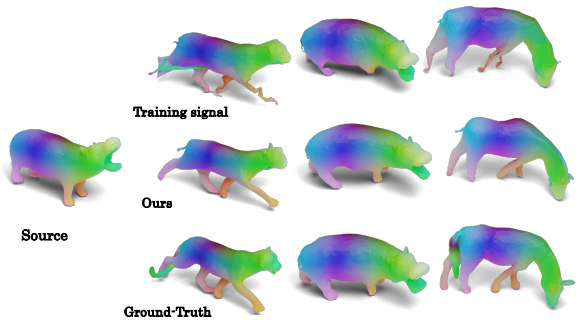


Fig. 9. Qualitative reconstruction results of training LJN without supervision. For a given source, the first row depicts the training signal used in our unsupervised loss (c.f Eqn 14). In spite of training with noisy supervision, LJN recovers cleaner deformation as depicted in the subsequent row. Please refer to Tab.3 in Appendix for quantitative result.

FAUST-Challenge. To demonstrate the generalizability of our approach to real-world scans, we evaluated our *unsupervised* LJN (Sec. 4.4) on the FAUST-Challenge dataset [Bogo et al. 2014]. This dataset contains 100 shape pairs across near-isometric (INTRA) and non-isometric (INTER) registration challenges. The scans include various artifacts such as non-manifold edges, self-intersections, and topological noise. Despite not having specific mechanisms for handling data imperfections and without hyperparameter tuning, LJN achieves comparable performance to unsupervised state-of-the-art methods, with a mean error of **3.89** cm for INTRA and **2.62** cm for INTER challenges respectively. We remark especially on the low error on the more challenging INTER dataset. For the INTER dataset, our average error is dominated by three outlier pairs with significant error due, especially to symmetry mixing, which have an average error of 30 cm. We believe that such errors could be potentially be reduced by using symmetry disambiguation [Donati et al. 2022b,a] and post-processing [Huang et al. 2020; Melzi et al. 2019b; Vestner et al. 2017] techniques.

5.4 Interactive Editing

For this task, we re-trained our network on the same dataset of 60 human shape pairs, but with the closest rotation as the input signal to LJN. We evaluated two setups for handle-based deformation. In the first, we performed handle-based manual deformation, where a user drags a selected portion of the mesh. This was done by selecting regions of the mesh in Blender and applying a sequence of rigid transformations to the selected vertices. We evaluated our method on five different object categories, as shown in Figure 11. We compared our approach with ARAP and observed that our method produces lower distortion, measured by symmetric Dirichlet [Smith and Schaefer 2015]. LJN produces a smoother embedding than ARAP without undesirable artifacts such as changes in pose (row 2) and self-intersection (row 4).

In the second setup, we used datasets containing meshes corresponding to deformation sequences. Please note that this second experimental setup is more challenging since the deformation is significantly larger. We considered animals from the [Li et al. 2021; Sumner and Popović 2023] datasets and automatically selected key

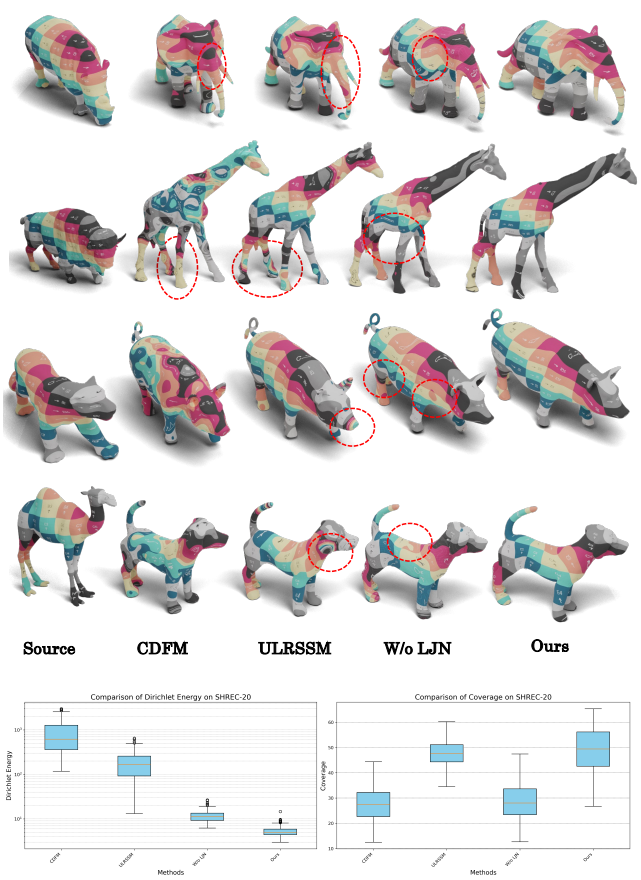


Fig. 10. Qualitative and Quantitative results for unsupervised shape correspondence on the SHREC'20 [Dyke et al. 2020] dataset. We compare our results with those from CDFM [Sun et al. 2023], ULRSSM [Cao et al. 2023], and a variant of our approach (W/o LJN) where the point-wise map is obtained by following the standard procedure (cf. Eqn 1) instead of our proposed method (cf. Eqn 11). In the last column, we show the point-wise map recovered by LJN. ULRSSM produces plausible maps but exhibits severe distortion in various highlighted regions. In contrast, our approach preserves correspondence while minimizing distortion.

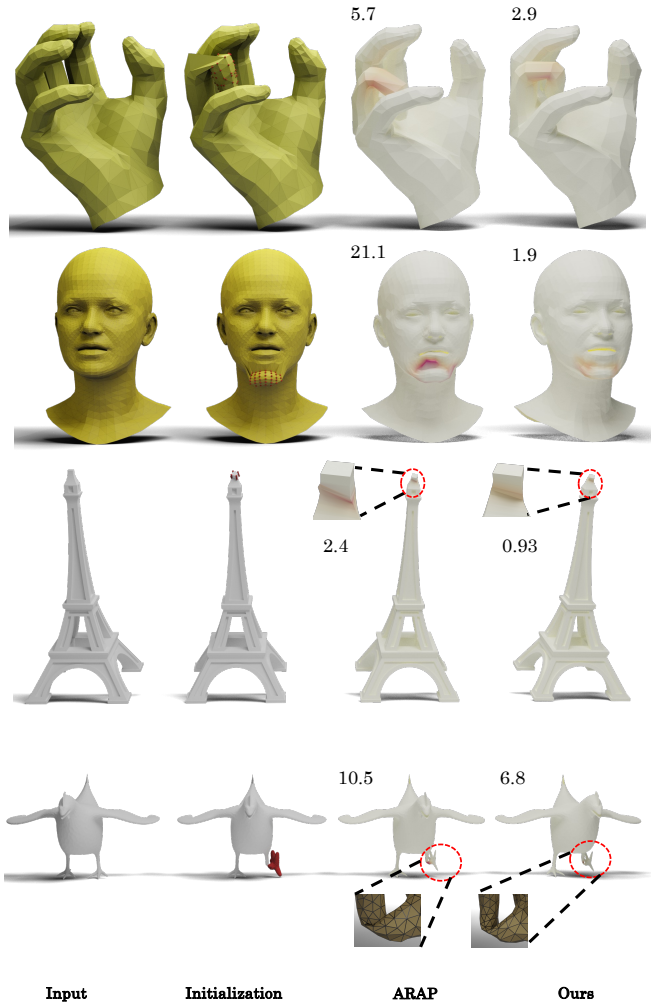


Fig. 11. Qualitative results for handle-based deformation. We selected a set of handles (shown in red) and applied a deformation by displacing those vertices. We then compare the reconstruction between our approach and ARAP. The error plot on the outputs depicts the distribution of symmetric Dirichlet energy, with the average value given in the annotation. LJN produces a more plausible mesh with smooth embedding compared to ARAP [Sorkine 2006]. Please refer to the supplementary material for more quantitative insights.

points, corresponding to the group of vertices that undergo the largest displacement. Given the new positions of the key points, we estimated the Jacobians between the source mesh and the suggested deformation, projected them to the closest rotation, and estimated the smoother deformation predicted by LJN. We evaluated this on three animals—Cat, Buck, and Horse—as shown in Figure 12. LJN produces more plausible deformation, especially at regions pertaining to bending in comparison with ARAP.

6 ADDITIONAL EXPERIMENTAL INSIGHTS

We first perform an ablation study in Section 6.1, where we empirically justify the simple architecture of LJN. Next, we provide a runtime comparison between our approach and different baselines across various datasets evaluated in Section 6.4.

6.1 Ablation Studies

We perform an ablation study to understand the effect of supervising Jacobians, using displacement fields instead of Jacobians, input discretization (face vs. vertex), and impact of feature smoothing (Projection of features to the eigenspace of LBO Operator). To this end, we compare the reconstruction accuracy of the deformation and the correspondence error by measuring the mean geodesic discrepancy. We summarize our results in Table 2. Face-based discretization lacks information sharing within the local neighborhood, resulting in implausible deformations leading to poor reconstruction results. On the other hand, LJN promotes smoother feature space and as a result, achieves more plausible deformation. We visualize the problems

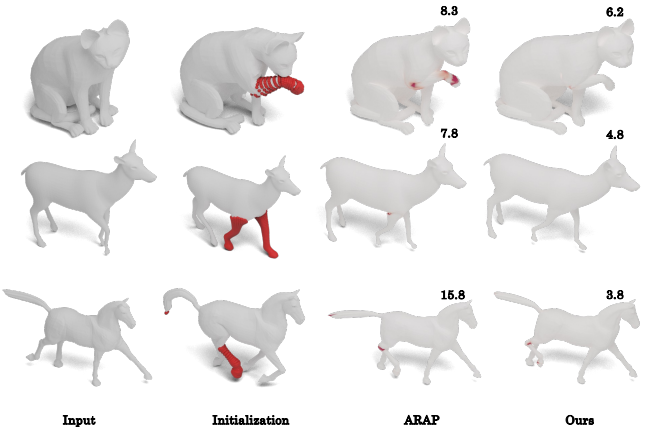


Fig. 12. We perform interactive editing on animation sequence meshes by selecting handles as vertices with the largest motion between the first and second columns. Our approach produces more plausible meshes in bending regions compared to ARAP. The values annotated above are symmetric Dirichlet scaled by 10^{-2}

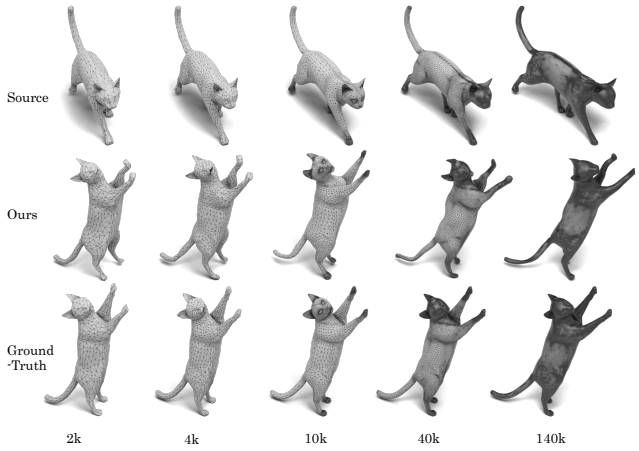


Fig. 13. Comparison of deformation between the same source and target geometry across different tessellations. The face counts are annotated along the rows. Across varying resolutions, our deformation remains faithful to the target geometry.

inherent to alternatives we tried in Figure 14. All baselines produce artifacts in reconstruction corresponding to high-frequency parts of the shape while ours recovers the near-exact geometry. For a fair comparison, all methods were trained on the same training data, following the same hyper-parameters.

6.2 Robustness to changes in tesselation

We analyze the effect of tesselation on LJN. To this end, we take a pair of shapes from the animal dataset in [Sumner and Popović 2023] and re-mesh to generate shapes with varying triangulation. Namely, we generate the same shape with varying triangulation ranging from 2K faces to 200K faces. We apply mesh decimation with

Module	FAUST		SCAPE	
	CD	Geod	CD	Geod
$\alpha_1 = 0$	5.1	1.7	10.0	2.4
Displacements	5.2	1.8	10.4	2.3
Face-based	7.4	2.1	13.1	3.0
W/o Smooth	4.8	1.8	10.7	2.6
Baseline (Ours)	3.9	1.5	7.6	2.1

Table 2. Ablation study on map refinement and deformation reconstruction on the test set of the FAUST and SCAPE datasets [Ren et al. 2018]. CD denotes Chamfer's Distance and Geod is the mean geodesic error [Kim et al. 2011]

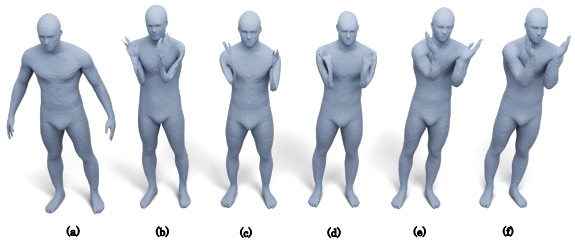


Fig. 14. Ablation study performing a qualitative comparison between possible alternatives. Given the (a) Source shape, (b) is the reconstruction from displacement field, (c) denotes Face-based discretization, and (d) is the reconstruction without using the feature smoothness in the MLP. Our reconstruction is depicted in (e) and finally, (f) denotes the target shape.

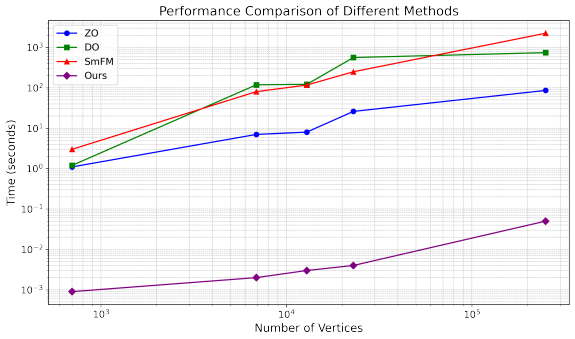


Fig. 15. Performance (run-time) comparison of different methods. Our LJN shows orders of magnitude faster performance than other baselines.

Quadric Edge collapse to obtain the simplified meshes with reduced vertex and face count. To increase the resolution, we apply loop subdivision until a mesh with the desired face count is obtained. This re-meshing is done to the pair of initial meshes. Then, we compute the spectrally projected Jacobians between (c.f. Eqn 6) and estimate the deformation from the network prediction. We compare different mesh resolutions in Figure 13. Across different discretizations, our LJN produces a deformation which is consistently faithful to the target geometry.

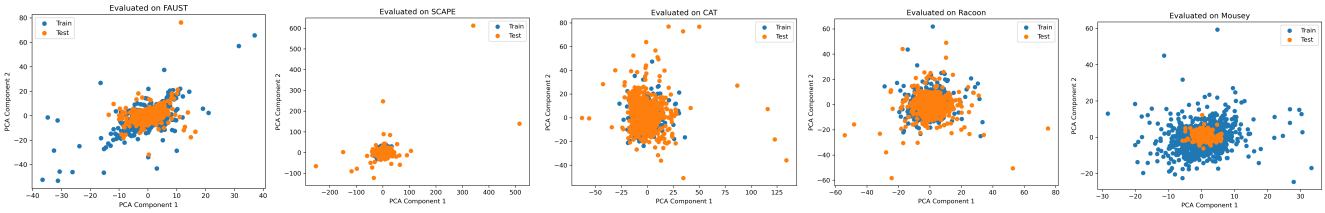


Fig. 16. We compare the input signal between different evaluation datasets and our *standard* training set, which was identical across two supervised experiments detailed in the main paper using Principal Component Analysis (PCA). The evaluation datasets we compare with respectively are FAUST [Bogo et al. 2014], SCAPE [Anguelov et al. 2005], sequences of Cats from [Sumner and Popović 2023]. Racoon and Mousey are two object categories from the Deforming4D Dataset [Li et al. 2021]

6.3 Baseline Details

Across all experiments, we used the open-source code released by the authors of the respective papers for baseline comparisons. Starting with axiomatic refinement methods, we ran ZoomOut beginning with a 20×20 Functional Map, upsampling with a step-size of 5 to reach the resolution of a 120×120 sized functional map. For SmFM [Magnet et al. 2022] and Discrete Optimization [Ren et al. 2021], we used the Python implementation available in the PyFM library. Specifically, for Discrete Optimization, we initialized the Functional Map at 20×20 and upsampled it to 100×100, whereas for SmFM, we initialized a 10×10 Functional Map and upsampled it to 70×70. For Consistent FM [Sun et al. 2023] and ULRSSM [Cao et al. 2023], we used the author-provided code and pre-trained models. For Consistent-FM, we used the point-wise map corresponding to 80×80 Functional Map. Finally, for handle-based deformation, we used the libigl implementation of ARAP [Jacobson et al. 2018] as our baseline.

6.4 Run-Time Comparison

We perform a run-time comparison between our approach and different baselines for the task of map refinement. We show this comparison in Figure 15, evaluated over shape pairs consisting of different number of vertices. We do not count the pre-processing time since both LJN and all baselines require the Eigen decomposition of the LBO operator. To recall, our LJN performs map refinement in a single feedforward pass followed by back substitution, while the baselines are iterative refinement techniques. As a result, LJN is orders of magnitude faster than baselines. These experiments were performed on a machine with AMD 7302 and Nvidia A100 GPU.

6.5 Discussion on Generalization

We provide an empirical overview to explain why our approach generalizes across different categories of objects. To demonstrate this, we compare the input signal from our training set with the input signals across different object categories used in our qualitative and quantitative evaluations. Recall that our input signal is the Jacobian corresponding to the spectrally projected target shape, averaged over vertices. We compare this input signal, defined at vertices, across different datasets using Principal Component Analysis (PCA), as depicted in Figure 16. The strong correlation between the input signals in our training and test datasets highlights the remarkable ability of LJN to generalize.

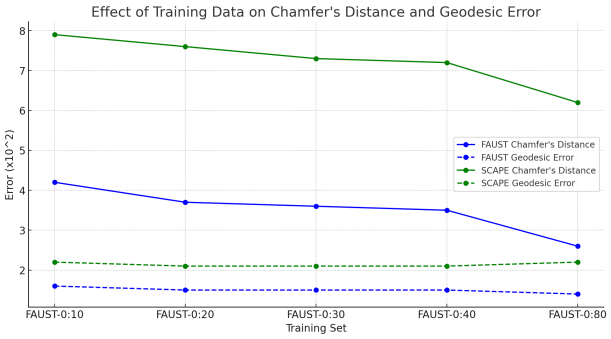


Fig. 17. Understanding the effect of training data required to achieve optimal performance. Trained on 10 shapes, our approach demonstrates comparable performance for shape correspondence as a model that has been trained on 80 shapes.

6.6 Effect of Training Data

LJN is highly data-efficient, requiring only 60 pairs of human shapes to demonstrate significant generalization across different shape categories. In this section, we empirically investigate the minimum data required to achieve optimal performance. We begin with a dataset of 10 shapes and incrementally increase it to 80 shape pairs, all from the FAUST dataset. We evaluate the performance of LJN over different training set using reconstruction and shape correspondence metrics on both the FAUST and SCAPE datasets, following the map-refinement setup detailed in Section 5.2. The results, visualized in Figure 17, indicate that LJN achieves accurate results with as few as 10 shape pairs. Moreover, evaluating SCAPE [Anguelov et al. 2005] also yields accurate results despite the pairs being near-isometric and having different connectivity than the training (FAUST) shapes.

6.7 Comparison with Global Encoding

We also compare our LJN approach with the original Neural Jacobian Fields (NJF) [Aigerman et al. 2022] method which relies on a *global* latent code to define a deformation of a shape. As mentioned above, the global nature of the latent code along with the supervised

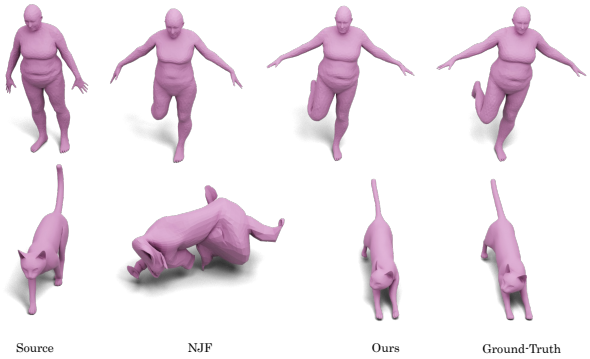


Fig. 18. Comparison of reconstructions produced by a global encoding (NJF) and our LJN which uses local signals to learn deformation. NJF and our approach produce near-perfect deformations when evaluated on a pair of shapes from the same training category (first row). However, NJF fails to produce cross-category generalization, where, LJN shows consistent reconstruction (second row).

nature of NJF imply that that approach requires extensive data and *per-category training* to produce plausible deformations. Thus, the basic version of NJF cannot be directly compared with the various experimental setups in this paper, as its use of global encoding fundamentally limits its generalization. We illustrate this difference in Figure 18. In the first row, we visualize a deformation between a pair of human shapes, a category on which NJF was trained. In the second row, we evaluate the same model on a pair of cat shapes. For fairness, our LJN method *was also trained on human shape pairs*, yet unlike NJF, LJN produces near-perfect deformation for shapes outside the trained category.

7 CONCLUSION, LIMITATIONS AND FUTURE WORK

This paper presented LJN, a data-driven framework that efficiently estimates deformations by computing high-quality Jacobian fields from coarse inputs. Unlike the typical global encoding for representing deformations, we showed that constructing appropriate local signals is generally sufficient to learn robust deformations. These input signals constrain the neural network to learn deformation within a small neighborhood while leveraging the Poisson system for global coherence. This approach makes LJN data-friendly and robust in cross-category generalization, with both supervised and unsupervised pipelines applicable to various deformation-based tasks. While tailored for detailed deformations, LJN can still produce implausible results such as shrinkage of volume or failure to produce sharp bending. Future work could include integrating physics-based energies into the data-driven realm for more realistic deformations. Another avenue is to explore better approximations to deformation spaces, such as constructing input signals from eigenfunctions of the Discrete Shell-Operator [Tamstorf and Grinspun 2013].

8 ACKNOWLEDGEMENT

Parts of this work were supported by the ERC Starting Grant 758800 (EXPROTEA), ERC Consolidator Grant 101087347 (VEGA), ANR AI Chair AIGRETTE, as well as gifts from Ansys and Adobe Research. We also gratefully acknowledge the support of NVIDIA Corporation to the University of Milano-Bicocca, PRIN 2022 project 'GEOPRIDE' and MUR under the grant "Dipartimenti di Eccellenza 2023-2027". This work was performed using HPC resources from GENCI–IDRIS (Grant 2023-AD011013104). We thank Guillaume Coiffier for insightful discussions.

REFERENCES

S Mazdak Abulnaga, Oded Stein, Polina Golland, and Justin Solomon. 2022. Symmetric Volume Maps. *arXiv preprint arXiv:2202.02568* (2022).

Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes. *SIGGRAPH* (2022).

Noam Aigerman and Yaron Lipman. 2013. Injective and bounded distortion mappings in 3D. *ACM Trans. Graph.* 32, 4, Article 106 (jul 2013), 14 pages. https://doi.org/10.1145/2461912.2461931

Brian Amberg, Sami Romdhani, and Thomas Vetter. 2007. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 1–8. https://doi.org/10.1109/CVPR.2007.383165

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: shape completion and animation of people. *ACM SIGGRAPH 2005 Papers* (2005). https://api.semanticscholar.org/CorpusID:3423879

Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. 2013. An Operator Approach to Tangent Vector Field Processing. *Computer Graphics Forum* 32, 5 (Aug. 2013), 73–82. https://doi.org/10.1111/cgf.12174

Dominique Bechmann. 1994. Space deformation models survey. *Computers & Graphics* 18, 4 (1994), 571–586.

P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256. https://doi.org/10.1109/34.121791

Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. 2020. LoopReg: Self-supervised Learning of Implicit Surface Correspondences, Pose and Shape for 3D Human Mesh Registration. In *Neural Information Processing Systems (NeurIPS)*.

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

Mario Botsch and Olga Sorkine. 2008. On Linear Variational Surface Deformation Methods. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 213–230. https://doi.org/10.1109/TVCG.2007.1054

Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. 2006. Deformation Transfer for Detail-Preserving Surface Editing. *Vision, Modeling and Visualization* (2006), 357–364. http://infoscience.epfl.ch/record/307221

Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse iterative closest point. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing* (Genova, Italy) *(SGP '13)*. Eurographics Association, Goslar, DEU, 113–123. https://doi.org/10.1111/cgf.12178

Dongliang Cao, Marvin Eisenberger, Nafie El Amrani, Daniel Cremers, and Florian Bernard. 2024. Spectral Meets Spatial: Harmonising 3D Shape Matching and Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dongliang Cao, Paul Roetzer, and Florian Bernard. 2023. Unsupervised Learning of Robust Spectral Shape Matching. *ACM Trans. Graph.* 42, 4, Article 132 (jul 2023), 15 pages. https://doi.org/10.1145/3592107

Yang Chen and Gérard Medioni. 1992. Object modelling by registration of multiple range images. *Image and Vision Computing* 10, 3 (April 1992), 145–155. https://doi.org/10.1016/0262-8856(92)90066-c

Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. 2017. Functional Characterization of Intrinsic and Extrinsic Geometry. *ACM Transactions on Graphics* 36, 2 (March 2017), 1–17. https://doi.org/10.1145/2999535

Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2019. Learning elementary structures for 3D shape generation and matching. In *NeurIPS*.

Ana Dodik, Oded Stein, Vincent Sitzmann, and Justin Solomon. 2023. Variational Barycentric Coordinates. *ACM Transactions on Graphics* 42, 6 (Dec. 2023), 1–16. https://doi.org/10.1145/3618403

Nicolas Donati, Etienne Corman, Simone Melzi, and Maks Ovsjanikov. 2022b. Complex functional maps: A conformal link between tangent bundles. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 317–334.

Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. 2022a. Deep orientation-aware functional maps: Tackling symmetry issues in Shape Matching. *CVPR* (2022).

Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. 2020. Deep Geometric Maps: Robust Feature Learning for Shape Correspondence. *CVPR* (2020).

Anastasia Dubrovina and Ron Kimmel. 2010. Matching shapes by eigendecomposition of the Laplace-Beltrami operator. https://api.semanticscholar.org/CorpusID:403058

Roberto M. Dyke, Feng Zhou, Yu-Kun Lai, Paul L. Rosin, Daoliang Guo, Kun Li, Riccardo Marin, and Jingyu Yang. 2020. SHREC 2020 Track: Non-rigid Shape Correspondence of Physically-Based Deformations. In *Eurographics Workshop on 3D Object Retrieval*, Tobias Schreck, Theoharis Theoharis, Ioannis Pratikakis, Michela Spagnuolo, and Remco C. Veltkamp (Eds.). The Eurographics Association. https://doi.org/10.2312/3dor.20201161

Danielle Ezuz and Mirela Ben-Chen. 2017. Deblurring and Denoising of Maps between Shapes. *Computer Graphics Forum* 36, 5 (Aug. 2017), 165–174. https://doi.org/10.1111/cgf.13254

D. Ezuz, B. Heeren, O. Azencot, M. Rumpf, and M. Ben-Chen. 2019a. Elastic Correspondence between Triangle Meshes. *Computer Graphics Forum* 38, 2 (May 2019), 121–134. https://doi.org/10.1111/cgf.13624

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019b. Reversible Harmonic Maps between Discrete Surfaces. *ACM Transactions on Graphics* 38, 2 (March 2019), 1–12. https://doi.org/10.1145/3202660

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019c. Reversible Harmonic Maps between Discrete Surfaces. 38, 2, Article 15 (mar 2019), 12 pages. https://doi.org/10.1145/3202660

William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. 2023. TextDeformer: Geometry Manipulation using Text Guidance. In *ACM Transactions on Graphics (SIGGRAPH)*.

Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. 2005. Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (<conf-loc>, <city>Vienna</city>, <country>Austria</country>, </conf-loc>) *(SGP '05)*. Eurographics Association, Goslar, DEU, 197–es.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 2018a. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018b. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 230–246.

Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. 2018. PCPNet: Learning Local Shape Properties from Raw Point Clouds. *Computer Graphics Forum* 37, 2 (2018), 75–85. https://doi.org/10.1111/cgf.13343

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 90:1–90:12.

Osamu Hirose. 2023. Geodesic-Based Bayesian Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2023), 5816–5832. https://doi.org/10.1109/TPAMI.2022.3214191

Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. 2020. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 265–278.

Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. 2016. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*. Springer, 362–379.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas Guibas. 2020. ShapeFlow: Learnable Deformations Among 3D Shapes. In *Advances in Neural Information Processing Systems*.

Puhua Jiang, Mingze Sun, and Ruqi Huang. 2023. Non-Rigid Shape Registration via Deep Functional Maps Prior. *arXiv preprint arXiv:2311.04494* (2023).

Angjoo Kanazawa, Shahar Ziv Kovalsky, Ronen Basri, and David W. Jacobs. 2015. Learning 3D Deformation of Animals from 2D Images. *Computer Graphics Forum* 35 (2015). https://api.semanticscholar.org/CorpusID:1012261

Vladimir Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended Intrinsic Maps. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30, 4 (July 2011).

Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Bongsoo Choy, and Silvio Savarese. 2017. DeformNet: Free-Form Deformation Network for 3D Shape Reconstruction from a Single Image. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2017), 858–866. https://api.semanticscholar.org/CorpusID:9585413

Bruno Lévy and Hao (Richard) Zhang. 2010. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 8, 312 pages. https://doi.org/10.1145/1837101.1837109

Hao Li, Robert W. Sumner, and Mark Pauly. 2008. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum* 27, 5 (July 2008), 1421–1430. https://doi.org/10.1111/j.1467-8659.2008.01282.x

Lei Li, Nicolas Donati, and Maks Ovsjanikov. 2022. Learning Multi-resolution Functional Maps with Spectral Attention for Robust Shape Matching. In *Advances in Neural Information Processing Systems*.

Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Niessner. 2021. 4DComplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Or Litany, Alexander M. Bronstein, Michael M. Bronstein, and Ameesh Makadia. 2017a. Deformable Shape Completion with Graph Convolutional Autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 1886–1895. https://api.semanticscholar.org/CorpusID:4587604

Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. 2017b. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*. 5659–5667.

Arman Maesumi, Paul Guerrero, Vladimir G. Kim, Matthew Fisher, Siddhartha Chaudhuri, Noam Aigerman, and Daniel Ritchie. 2023. Explorable Mesh Deformation Subspaces from Unstructured Generative Models. (2023). https://doi.org/10.1145/3610548.3618192

Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. 2022. Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization. In *International Conference on 3D Vision (3DV)*.

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-Minimizing Transport Plans for Inter-surface Mapping. *ACM Transactions on Graphics* 36, 4, Article 39 (2017).

S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. 2019a. Matching Humans with Different Connectivity. *Eurographics Workshop on 3D Object Retrieval* (2019). https://doi.org/10.2312/3DOR.20191070

Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. 2019b. ZoomOut: spectral upsampling for efficient shape correspondence. 38, 6, Article 155 (nov 2019), 14 pages. https://doi.org/10.1145/3355089.3356524

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. 2003. *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Springer Berlin Heidelberg, 35–57. https://doi.org/10.1007/978-3-662-05105-4_2

Andriy Myronenko and Xubo Song. 2010. Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (2010), 2262–2275. https://doi.org/10.1109/TPAMI.2010.46

Ahmed A A Osman, Timo Bolkart, and Michael J. Black. 2020. STAR: A Sparse Trained Articulated Human Body Regressor. In *European Conference on Computer Vision (ECCV)*. 598–613. https://star.is.tue.mpg.de

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–11.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. 2017. Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 5, 62 pages. https://doi.org/10.1145/3084873.3084877

Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame Fields: Anisotropic and Non-Orthogonal Cross Fields. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 33, 4 (2014), 134:1–134:11.

Ulrich Pinkall and Konrad Polthier. 1993. Computing Discrete Minimal Surfaces and Their Conjugates. *Exp. Math.* 2 (1993), 15–36. https://api.semanticscholar.org/CorpusID:10068306

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint arXiv:1612.00593* (2016).

Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. 2021. Discrete Optimization for Shape Matching. *Computer Graphics Forum* 40, 5 (Aug. 2021), 81–96. https://doi.org/10.1111/cgf.14359

Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 1–16. https://doi.org/10.1145/3272127.3275040

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

Raif M. Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. 2013. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics* 32, 4 (July 2013), 1–12. https://doi.org/10.1145/2461912.2461959

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. 2020. Diffusion is All You Need for Learning on Surfaces. *CoRR* abs/2012.00888 (2020). arXiv:2012.00888 https://arxiv.org/abs/2012.00888

Jason Smith and Scott Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Transactions on Graphics* 34, 4 (July 2015), 1–9. https://doi.org/10.1145/2766947

Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics* 35, 4 (July 2016), 1–13. https://doi.org/10.1145/2897824.2925903

Olga Sorkine. 2006. Differential Representations for Mesh Processing. *Computer Graphics Forum* 25, 4 (Dec. 2006), 789–807. https://doi.org/10.1111/j.1467-8659.2006.

00999.x

Robert W. Sumner. 2005. Mesh modification using deformation gradients. https://api.semanticscholar.org/CorpusID:1741173

Robert W. Sumner and Jovan Popović. 2023. *Deformation Transfer for Triangle Meshes* (1 ed.). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3596711.3596736

Mingze Sun, Shiwei Mao, Puhua Jiang, Maks Ovsjanikov, and Ruqi Huang. 2023. Spatially and spectrally consistent deep functional maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14497–14507.

Ramana Sundararaman, Riccardo Marin, Emanuele Rodola, and Maks Ovsjanikov. 2022. Reduced Representation of Deformation Fields for Effective Non-rigid Shape Matching. *Advances in Neural Information Processing Systems* 35 (2022).

Rasmus Tamstorf and Eitan Grinspun. 2013. Discrete bending forces and their Jacobians. *Graphical Models* 75, 6 (Nov. 2013), 362–370. https://doi.org/10.1016/j.gmod.2013.07.001

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. *ACM SIGGRAPH Computer Graphics* 21, 4 (Aug. 1987), 205–214. https://doi.org/10.1145/37402.37427

Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. 2021. Shape Registration in the Time of Transformers. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=ui4xChWcA4R

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2017. Learning from Synthetic Humans. In *CVPR*.

Matthias Vestner, Roee Litman, Emanuele Rodola, Alex Bronstein, and Daniel Cremers. 2017. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3327–3336.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics* 38, 5 (Oct. 2019), 1–12. https://doi.org/10.1145/3326362

Wang Yifan, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. 2020. Neural Cages for Detail-Preserving 3D Deformations. In *CVPR*.

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

Supplementary Material - Deformation Recovery: Localized Learning for Detail-Preserving Deformations

Object	Symmetric Dirichlet		#V, F	#Handle
	ARAP	Ours		
Pig	0.26	0.05	4K,8K	330
Hand	5.7	2.9	700, 1.3k	72
Eiffel	2.4	0.93	3k, 6.1k	21
Chicken	10.5	6.8	8k, 16k	400
Head	1.9	21.1	5k, 8k	63
Cat	7.8	4.8	7k, 14k	800
Buck	6.2	8.3	5k, 8k	2300
Horse	15.6	3.8	5k, 8k	600

Table 1. Quantitative comparison for handle-based deformation. We provide the quantitative symmetric Dirichlet scores denoting the smoothness in the deformation alongside details of the mesh used.

ACM Reference Format:
. 2024. Supplementary Material - Deformation Recovery: Localized Learning for Detail-Preserving Deformations. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (SIGGRAPH Asia '24).* ACM, New York, NY, USA, Article 219, 1 page. https://doi.org/10.1145/3687968

1 OVERVIEW

In this document, we provide additional details pertaining to our work. We provide an overview of all symbols used throughout the paper in Table 2 for expositional clarity. In Table 1, we provide the details of the dataset used in our handle-based deformation. Finally in Table 3, we provide quantitative results of unsupervised deformation and mapping on the SMAL dataset.

Received 18 May 2024

Author's Contact Information:

SIGGRAPH Asia '24, December 03–06, 2024, Tokyo, JPN

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/3687968

Symbols	Description
$\mathcal{S}_1$	Source Shape
$\mathcal{S}_2$	Target Shape
$\mathcal{V}_i$	Vertex corresponding to $\mathcal{S}_i$
$\mathcal{F}_i$	Face corresponding to $\mathcal{S}_i$
Δ	Laplace Beltrami Operator
∇	Discrete gradient operator $\mathbb{R}^{\lvert 3F\rvert \times \lvert V\rvert}$
A	Area of each face, written as diagonal matrix $\mathbb{R}^{\lvert 3F\rvert \times \lvert 3F\rvert}$
M	Voronoi-area of each vertex, written as diagonal matrix $\mathbb{R}^{\lvert V\rvert \times \lvert V\rvert}$
Ψ	Eigenbasis of Laplace operator
$\mathcal{I}$	$\mathbb{R}^{\lvert V\rvert \times \lvert F\rvert}$ vertex-face incidence matrix
$\mathcal{H}$	$\mathbb{R}^{\lvert F\rvert \times \lvert F\rvert}$ face-face incidence matrix
$\bar{\mathcal{V}}$	Projection of $\mathcal{V}_i$ to the LBO eigenbasis
$\mathcal{E}_k$	Non-orthonormal frame of face k
$\mathbf{E}_k$	Rewriting $\mathcal{E}_k$ across all faces in matrix form
φ_{12}	Continuous map between $\mathcal{S}_1$ and $\mathcal{S}_2$
Π_{12}	Vertex-Vertex map between $\mathcal{S}_1$ and $\mathcal{S}_2$
Π^*_{12}	Refined vertex-vertex map
$\tilde{\Pi}_{12}$	Soft pointwise map
C_{21}	Functional map corresponding to Π_{12}
$\mathbf{J}_{12}$	Jacobian between $\mathcal{S}_1$ and $\mathcal{S}_2$
Θ_{12}	Input signal to our network
e^i_1	First edge vector of i^{th} face
Q_{12}	Closest rotation matrix to $\mathbf{J}_{12}$

Table 2. Collecting all symbols used in our main paper

Method	**Geod**	**Inv**	**DirE**	**Cov**
W/o $\mathcal{L}_J$	5.5	12.5	17.1	59.6
$\alpha_6, \alpha_7 = 0$	5.3	11.9	15.4	59.6
$\alpha_7 = 0$	9.7	11.7	25.2	53.2
Ours	5.0	10.6	9.8	60.4

Table 3. Quantitative result for unsupervised deformation on the SMAL dataset. We ablate the different terms used in our unsupervised loss function $\mathcal{L}_{un}$(c.f Eqn 13) over the SMAL dataset.