

Rotation-Equivariant Neural Jacobian Fields for Generalizable 3D Mesh Deformations

LUKAS UZOLAS, Delft University of Technology, Netherlands

RUBEN WIERSMA, ETH Zurich, Switzerland and Adobe Research, France

OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

PETR KELLNHOFER, Delft University of Technology, Netherlands

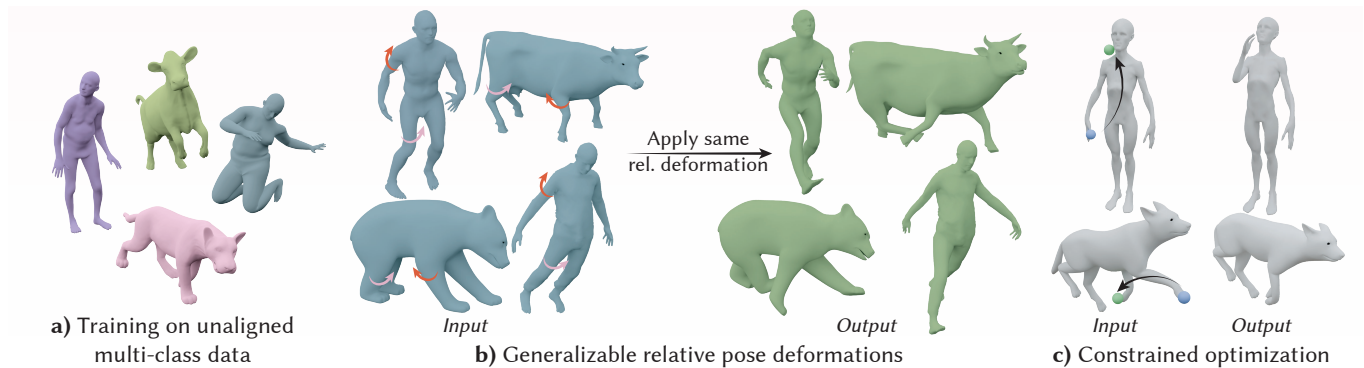


Fig. 1. We present Re-NJF, a rotation-equivariant formulation of Neural Jacobian Fields. a) Our method can be trained on datasets without extrinsic alignment and generalizes across classes. b) Thanks to these generalization capabilities, we can apply relative pose deformations to meshes with various shapes. c) Furthermore, when trained on articulated pose data, Re-NJF can be used for inverse kinematics without rigging the input meshes.

Neural Jacobian Fields are state-of-the-art neural representations for deformations on 3D meshes. A fundamental limitation of both the original formulation and follow-up works is that the learned Jacobians are not equivariant to ambient-space rotations of the source mesh. This hinders generalization to shapes that are misaligned or greatly different in shape. Moreover, our analysis shows that this rotation dependence stems from the formulation itself and cannot be remedied with rotation-invariant features alone. Instead, we propose a rotation-equivariant formulation of Neural Jacobian Fields, Re-NJF, representing Jacobians in local frames of triangle faces aligned with their normals. The frames are learned end-to-end in the gradient space of semantic surface features distilled from 2D image foundation models. Robust to rotations and large geometry changes, these features support learning tangent frames and Jacobians that generalize across transformations in $SO(3)$. Moreover, our localized tangent frame definition also facilitates generalization to non-rigid transformations of the input and substantial shape changes (e.g., dog to human). We evaluate our method on human and animal pose deformation datasets to simultaneously demonstrate competitive performance in the well-explored aligned data regime, as well as superior performance in tasks requiring generalization across ambient space rotations and input shape variations. Our code is available at <https://graphics.tudelft.nl/Re-NJF>.

Authors' Contact Information: Lukas Uzolas, Delft University of Technology, Netherlands, luzolas@tudelft.nl; Ruben Wiersma, ETH Zurich, Switzerland and Adobe Research, France, rubenwiersma@gmail.com; Olga Sorkine-Hornung, ETH Zurich, Switzerland, sorkine@inf.ethz.ch; Petr Kellnhöfer, Delft University of Technology, Netherlands, p.kellnhofer@tudelft.nl.

SA Conference Papers '26, Kuala Lumpur, Malaysia

© 2026 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26)*, December 01–04, 2026, Kuala Lumpur, Malaysia, <https://doi.org/10.1145/3829340.3842184>.

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**; **Mesh models**; **Neural networks**; **Animation**.

Additional Key Words and Phrases: Neural Fields, Semantic Features, Pose Transfer

ACM Reference Format:

Lukas Uzolas, Ruben Wiersma, Olga Sorkine-Hornung, and Petr Kellnhöfer. 2026. Rotation-Equivariant Neural Jacobian Fields for Generalizable 3D Mesh Deformations. In *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26)*, December 01–04, 2026, Kuala Lumpur, Malaysia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3829340.3842184>

1 Introduction

Shape deformation is integral to turning static meshes into animated, moving characters. Deformations are typically applied through a lower-dimensional parameterization embodied by a control object, such as a cage [Ju et al. 2005], a skeleton rig [Lewis et al. 2000] or a set of point or area handles [Sorkine and Botsch 2009]. Point handles, for example, let users control complex shape deformations with only a few manual interventions. The construction of the hand-crafted deformation control structures and their manipulation require considerable domain expertise, posing a substantial barrier to producing realistic 3D deformations. Data-driven methods, on the other hand, can learn such a mapping between user-facing control and the shape deformation directly from examples. At inference time, a learned model can be applied to new meshes, enabling an easier interactive deformation scheme, such as through text [Tevet et al. 2022] or a given target mesh [Aigerman et al. 2022].

Neural Jacobian Fields (NJFs) [Aigerman et al. 2022] are a popular building block for learning shape deformations [Maesumi et al. 2025;

Muralikrishnan et al. 2024; Yoo et al. 2024]. NJFs represent deformations in the gradient domain using a neural network conditioned on input features (e.g., positions, normals, wave kernel signatures) and achieve state-of-the-art performance across different shapes, in part thanks to their triangulation agnosticism.

In our work, we push this generalizability further towards robustness to significant changes in *shape*, so that one can learn deformations on one set of shapes and generalize to unseen shapes, even of different classes. This requires two core properties: equivariance to input transformations (rotation, translation, scale) and input features robust to changes in geometry. Jacobians are equivariant to translation and scaling. Therefore, translation or scaling of NJF’s input will result in a corresponding transformation of the output. However, NJFs are not *rotation*-equivariant, even when using either rotation-invariant input features or features learned by rotation-invariant networks, such as PoissonNet [Maesumi et al. 2025]. Data augmentation with random rotations at training time also does not solve this problem (Fig. 9). Fundamentally, the lack of rotation-equivariance in NJF is caused by how Jacobians are represented. Jacobians map tangent vectors (2D, intrinsic) to the deformed vector in ambient space (3D, extrinsic). Therefore, the input shape of an NJF must be registered to a canonical coordinate system and cannot diverge too much from the training shape, which severely restricts its application.

To illustrate the issue, we overfit an NJF on a single deformation of a single mesh (Fig. 2). For the rotation used in training (0° , top row), NJF matches the target shape. In contrast, when the input is rotated (90° , bottom row), the output fails to match the target. In general, any non-zero rotation results in error (vertex error plot, right). Note that the input feature has no influence on this result: the per-vertex noise is constant for each rotation.

Our approach, Rotation-equivariant NJF (Re-NJF), solves this problem by decoupling deformation from extrinsic coordinates. Instead, we express Jacobians in local tangent frames, which ensures that deformations naturally follow extrinsic rotations. Our local, per-triangle frames also make our method robust to localized, non-rigid transformations, e.g., rotating an arm around the shoulder joint. Rather than defining tangent frames based on extrinsic geometry (e.g., principal curvatures), we propose to learn tangent frames for deformation from the gradients of an intrinsic input feature, inspired by approaches for intrinsic geometric deep learning [Maesumi et al. 2025; Sharp et al. 2022; Wiersma et al. 2022]. The resulting tangent frames are robust to deformation, smooth and consistent across diverse geometries.

Furthermore, instead of geometric descriptors, we leverage 2D image features from foundation models, inspired by recent work [Dutt et al. 2024]. Because these features encode semantic information, they are robust to variations in shape geometry and orientation. When used as a conditioning signal for NJF, they significantly improve deformation transfer between strongly differing shapes [Uzolas et al. 2025], addressing the limitation of poor cross-shape generalization and forming a robust signal to learn tangent frames.

To summarize, we make the following contributions toward more generalizable NJFs:

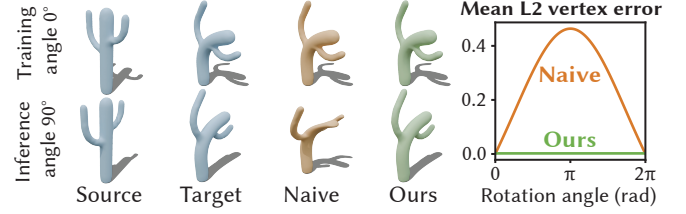


Fig. 2. We train a ‘naive’ NJF and our Re-NJF to deform the source mesh to the target mesh [Botsch and Sorkine 2008] with only per-vertex random noise as input, then apply the learned deformation on a 90° -rotated input. NJF produces large errors for rotated inputs, whereas Re-NJF remains correct.

- We describe a rotation-equivariant formulation of Neural Jacobian Fields, Re-NJF.
- We propose an intrinsic approach for end-to-end tangent-frame learning.
- We evaluate a full pipeline for rotation-equivariant NJFs implemented using semantic features with PoissonNet processing for robustness.

We empirically demonstrate that Re-NJF achieves generalization under rotation and across diverse shape classes, including humanoids and animals. By design, Re-NJF can be used as a drop-in replacement for existing NJF pipelines based on general rotation-invariant features, such as heat- or wave kernel signatures [Aubry et al. 2011; Sun et al. 2009], making it a practical tool for future applications.

2 Related works

We build upon learning mesh deformations on surfaces conditioned on semantic features obtained from 2D foundation models. In this section, we discuss related works on surface deformations, semantic features and rotation-equivariant processing.

2.1 3D mesh deformations

Traditional deformation approaches enable free-form 3D mesh deformation through user-defined manipulation handles [Sorkine and Botsch 2009], including cages [Ju et al. 2005], point or area handles [Sorkine and Alexa 2007], bone rotations within skeletal animation pipelines [Lewis et al. 2000], blend shape interpolation weights [Parke 1972, 1974] and physics-based simulation [Baraff and Witkin 1998]. These classical deformation methods achieve rotation equivariance by, e.g., propagating the local frame rotation of handle points or via optimization [Sorkine and Botsch 2009]. However, these methods typically produce mesh-dependent solutions that restrict generalization, require significant manual effort (e.g., character rigging) and can result in unnatural deformations.

Data-driven methods aim to reduce this burden by learning deformations from examples [Sumner et al. 2005]. One line of work focuses on learned template models that decouple pose and shape from training data for specific shape classes, such as human bodies [Loper et al. 2015], hands [Romero et al. 2017] and animals [Zuffi et al. 2017]. Another line of work addresses deformation transfer, where a source mesh is deformed to match a target, either through skeletal deformations [Aberman et al. 2020; Li et al. 2023]

or free-form mesh deformations [Aigerman et al. 2022; Gao et al. 2018; Sumner and Popović 2004]. Among the latter, Neural Jacobian Fields (NJFs) [Aigerman et al. 2022] achieve state-of-the-art, detail-preserving results by modeling deformations in the gradient domain through learned ambient-space Jacobians. NJF has since been adopted for various pose and motion transfer tasks, facilitating the transfer of deformation between broad shape classes [Muralikrishnan et al. 2024, 2025; Yoo et al. 2024]. However, NJF is not rotation-equivariant, which can hinder its generalization, a limitation that we address in this work.

2.2 Semantic features

Image foundation models trained on large-scale image datasets exhibit strong internal semantic representations. Examples include the DINO family [Caron et al. 2021; Oquab et al. 2024; Siméoni et al. 2026], a series of self-supervised vision transformers, and Stable Diffusion [Rombach et al. 2022], a diffusion model trained for image generation. These models produce semantically meaningful dense features that have proven effective for a range of 2D tasks, including image segmentation, classification and correspondence estimation [Amir et al. 2022; Oquab et al. 2024; Siméoni et al. 2026; Zhang et al. 2024].

The 3D domain currently lacks equivalent foundation models that produce domain-native semantic features. Therefore, several works have explored lifting 2D features to 3D. This can be achieved by rendering images of the 3D representations, extracting 2D image features and re-projecting them onto the 3D surface [Dutt et al. 2024; Wimmer et al. 2024]. These aggregated features have been utilized for keypoint correspondences [Dutt et al. 2024; Shtedritski et al. 2024; Uzolas et al. 2025; Weißberg et al. 2026; Wimmer et al. 2024], dense surface correspondences [Dutt et al. 2024; Morreale et al. 2024], 3D appearance transfer [Fischer et al. 2024], embedding language into 3D representations [Kerr et al. 2023], 3D deformation transfer [Uzolas et al. 2025], and for distilling 2D segmentation models for 3D segmentation tasks [Liu et al. 2025, 2023; Yang et al. 2024]. The latter example demonstrates the effectiveness of surface-aware features for conditioning pose transfer based on NJF. However, prior work requires alignment of training and inference shapes, which is a limitation addressed by our contribution.

2.3 Rotation-equivariant learning on surfaces

The past decade has seen many efforts to generalize deep learning from images and text to 3D surfaces and graphs, termed geometric deep learning [Bronstein et al. 2021, 2017]. Rotation equivariance is a core concern here because the geometry of a shape is often represented in an extrinsic coordinate system, which may be rotated arbitrarily. In addition, core building blocks for neural networks, such as convolution, require a local coordinate system that can be rotated arbitrarily about the normal. Because we use semantic features as input, we do not target rotation equivariance to input coordinates. Instead, our focus is on a localized, rotation-equivariant Jacobian operator to encode deformation. Still, we share several characteristics with related work.

PointNet, a pioneering work in geometric deep learning, addresses rotations by aligning the input using a transform-network [Qi et al. 2017], inspired by spatial transformer networks for images [Jaderberg et al. 2015]. We also learn transformation matrices (tangent frames); however, our transformations are localized (per-triangle) and aligned to feature *gradients*. That means our approach is robust to non-rigid deformations of the input shape. We include a comparison with a global transformation in our ablations. Other approaches adjust the convolution kernels in CNNs to ensure global rotation equivariance [Cohen et al. 2019; Thomas et al. 2018] and tangent-space rotation equivariance [De Haan et al. 2021; Mitchel et al. 2021; Poulenard and Ovsjanikov 2018; Wiersma et al. 2020]. We point readers to the survey by Bronstein et al. [2021] for more details. The latter approaches typically apply convolution kernels through a mapping from the tangent plane to the surface. An alternative is to construct coordinate-free convolution-like blocks from geometric operators, such as the (graph) Laplacian and gradient operator [Kipf and Welling 2017; Maesumi et al. 2025; Sharp et al. 2022; Wiersma et al. 2022]. We take inspiration from these approaches by aligning local coordinate systems to feature gradients. In addition, we process semantic features with PoissonNet [Maesumi et al. 2025], which is robust to arbitrary choices of global and local coordinate systems, given rotation-invariant input features.

3 Background and problem analysis

3D Representation. We assume as input a triangular surface mesh $\mathcal{M} := \{\mathbf{V}, \mathbf{F}\}$, where $\mathbf{V} \in \mathbb{R}^{V \times 3}$ represents the vertex locations and $\mathbf{F} \in \mathbb{N}^{F \times 3}$ provides the vertex indices for F triangles. Each face is associated with two tangent vectors $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^3$ orthonormal to each other and to the face normal $\mathbf{n}_i$. Together, these vectors form a right-handed local basis $\mathbf{B}_i := [\mathbf{x}_i, \mathbf{y}_i, \mathbf{n}_i]$. The tangent-vector pair can be arbitrarily rotated in the tangent plane; we discuss how we set them in the method section.

The discrete intrinsic gradient operator $\mathbf{G} \in \mathbb{R}^{2F \times V}$ is computed from $\mathcal{M}$ and outputs the gradient vector in the per-face tangent space spanned by $[\mathbf{x}_i, \mathbf{y}_i]$. We also use an extrinsic gradient operator $\mathbf{G}_{\text{ext}} \in \mathbb{R}^{3F \times V}$, which can be constructed from $\mathbf{G}$ by representing the tangent-space gradient in ambient space coordinates.

Mesh deformation and Jacobians. The Jacobian captures the deformation of a surface as a map from the 2-dimensional tangent space to the 3-dimensional ambient space. For a triangle mesh, the Jacobian is constant on each triangle. Given the vertex positions of a deformed mesh, we can determine the Jacobian $\mathbf{J}_i \in \mathbb{R}^{3 \times 2}$ for triangle i from the transformation of the triangle edges:

$$\mathbf{J}_i [\mathbf{x}_i, \mathbf{y}_i]^\top [\mathbf{v}_k - \mathbf{v}_j, \mathbf{v}_l - \mathbf{v}_j] = [\mathbf{v}'_k - \mathbf{v}'_j, \mathbf{v}'_l - \mathbf{v}'_j], \quad (1)$$

where j, k, l are the vertex indices of triangle i , and $\mathbf{v}_i, \mathbf{v}'_i \in \mathbb{R}^3$ are the (global) vertex coordinates in column format of the input and the deformed mesh, respectively. One can find the vertex positions $\mathbf{V}'$ whose Jacobian is closest to the target Jacobian $\mathbf{J}' \in \mathbb{R}^{3 \times 2F}$ (stacked horizontally for all faces) by solving the Poisson equation

$$\mathbf{L}\mathbf{V}' = \mathbf{G}^\top \mathbf{A} \mathbf{J}'^\top, \quad (2)$$

where $\mathbf{L} \in \mathbb{R}^{V \times V}$ is the cotangent Laplacian and $\mathbf{A} \in \mathbb{R}^{2F \times 2F}$ the triangle area matrix. For more details on linear deformation models, we refer the reader to the survey by Botsch and Sorkine [2008].

The core insight of NJF is to encode the Jacobian over the surface as a smooth, continuous field represented as a neural network. The neural field d_θ takes a feature $\mathbf{c}_i$ (defined over the mesh surface) as input and outputs a 3×2 target Jacobian

$$\mathbf{J}'_i = d_\theta(\mathbf{c}_i). \quad (3)$$

By moving away from a per-face Jacobian to a continuous field, the deformation can generalize to different triangulations and, depending on the input feature $\mathbf{c}$, to other shapes. NJF [Aigerman et al. 2022] chooses a combination of input features per task: concatenating the barycenters and normals of each face or intrinsic wave kernel signatures (WKS) [Aubry et al. 2011]. Maesumi et al. [2025] use PoissonNet to learn more expressive features from similar inputs (positions and normals or WKS). The deformation can be controlled at inference time by further concatenating a global shape encoding, such as bone-rotation angles or a text prompt to $\mathbf{c}$. In our case, $\mathbf{c}_i \in \mathbb{R}^S$ is defined as a semantic feature (see Sec. 4.2), such that $d_\theta : \mathbb{R}^S \rightarrow \mathbb{R}^{3 \times 2}$. Instead of regressing the Jacobians directly, follow-up works [Maesumi et al. 2025; Muralikrishnan et al. 2024] use a *relative* formulation, where only the difference to the gradient of the input shape is learned:

$$\mathbf{J}'_i = \mathbf{P}_i(\mathbf{G}\mathbf{V}) + d_\theta(\mathbf{c}_i), \quad (4)$$

where $\mathbf{P}_i \in \mathbb{R}^{3 \times 2F}$ is the row selection matrix, selecting the pairs of rows belonging to the i 'th face. A benefit of this formulation is that, when $d_\theta(\mathbf{c}_i) = \mathbf{0}$, the original vertex positions $\mathbf{V}$ are recovered when solving the Poisson equation (2).

Problem analysis. The issue that arises in both the direct and relative setting is that the Jacobian maps to coordinates in ambient space. This means that the mapping $d_\theta(\mathbf{c}_i)$ is inherently tied to the rotation of the input and output shape in ambient space, *regardless of the input features* $\mathbf{c}_i$. Formally, we want a mesh deformation pipeline f , parameterized by an NJF, that provides the following rotation-equivariant property: $f(\mathbf{V}\mathbf{R}) = f(\mathbf{V})\mathbf{R}$, where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ is an arbitrary rotation matrix. The conditioning features $\mathbf{c}_i$ must be rotation-invariant for this equivariance to hold. The central question of the current work is: how should we design an NJF to guarantee this property for global rotations and, ideally, per-part rotations of the shape?

4 Method

We first propose a rotation-equivariant formulation of Neural Jacobian Fields and its parameterization. Secondly, we discuss the semantic features used to condition deformation and tangent frame regression. Thirdly, we explain the deformation conditioning scheme. Lastly, we describe the training procedure.

4.1 Rotation-equivariant neural Jacobians

To learn rotation-equivariant Neural Jacobian Fields (Re-NJF), we decouple the Jacobian from ambient space by representing learned Jacobians d_θ (see (3)) in a local tangent frame $\mathbf{B}_i \in \mathbb{R}^{3 \times 3}$ (see Fig. 3).

The ambient-space per-face Jacobian is then given by

$$\mathbf{J}'_i = \mathbf{B}_i d_\theta(\mathbf{c}_i), \quad (5)$$

where $\mathbf{B}_i := [\mathbf{x}_i, \mathbf{y}_i, \mathbf{n}_i]$. Note that the same basis is assumed for the intrinsic gradient operator in (2). The orthonormal tangent vectors $\mathbf{x}_i, \mathbf{y}_i$ are determined up to a rotation about the normal vector. Typically, $\mathbf{x}_i$ is set to an arbitrary direction, such as the first edge of the corresponding triangle. This ties the tangent frame to the exact triangulation, hindering generalizability. Another option is to align $\mathbf{x}_i$ and $\mathbf{y}_i$ to the (smoothed) principal curvature directions of the surface. While this might work for similar source and target shapes, it hurts generalization to target shapes with different geometry, as demonstrated in our experiments (see Tbl. 1).

Inspired by work on intrinsic geometric deep learning [Maesumi et al. 2025; Sharp et al. 2022; Wiersma et al. 2022], we align the tangent frame to a learned function $\sigma_\phi : \mathbb{R}^S \rightarrow \mathbb{R}^3$ of the gradient of semantic features $\mathbf{S} \in \mathbb{R}^{V \times S}$, which outputs a single 3-vector per face:

$$\mathbf{x}_i = [\sigma_\phi(\mathbf{G}_{\text{ext}}\mathbf{S})]_i, \quad \mathbf{y}_i = \mathbf{n}_i \times \mathbf{x}_i. \quad (6)$$

We detail the input features $\mathbf{S}$ in the following section. The simplest choice for the tangent-frame head, σ_ϕ , with gradient vectors $\mathbf{G}_{\text{ext}}\mathbf{S}$, is their linear combination expressed as

$$\sigma_\phi(\mathbf{G}_{\text{ext}}\mathbf{S}) = \mathbf{G}_{\text{ext}}\mathbf{S}\phi, \quad (7)$$

where ϕ is an S -dimensional vector weighing the most informative semantic features, similarly to the first principal component vector in principal component analysis (PCA). In consequence, $\mathbf{B}_i$ in (5) is a learned function, because it is defined by $\mathbf{x}_i$. Furthermore, $\mathbf{x}_i$ is guaranteed to be orthonormal to $\mathbf{n}_i$ because it is computed as a gradient that lies in the local tangent plane and $\mathbf{x}_i$ is normalized.

In our initial experiments, we also explored a multi-layer, non-linear tangent-frame head (using vector non-linearities [Maesumi et al. 2025; Wiersma et al. 2022]); however, since we already process $\mathbf{S}$ using PoissonNet (see the following section), this did not improve our results.

Why local bases? An alternative to a per-face basis is to find a global coordinate basis for the entire shape and align the mesh to this basis before applying NJF; however, this would limit the applicability of NJF to global rigid rotations. In practice, deformations might be applied locally, such as rotating an arm or a leg. Learning a local basis makes our formulation robust to even such local deformations and rotations. See Fig. 7 for an example.

Singularities. The proposed method could run into singularities when regressing the frames. However, since we do not trace any vector fields but integrate Jacobians expressed in the regressed frames using the discrete Poisson equation, we are guaranteed to get a result.

4.2 Robust semantic features

As input to both (6) and the NJF, we use the semantic features $\mathbf{S}$ computed using Surface-Aware Semantic Features [Uzolas et al. 2025], which are robust to geometric variation across shape classes and

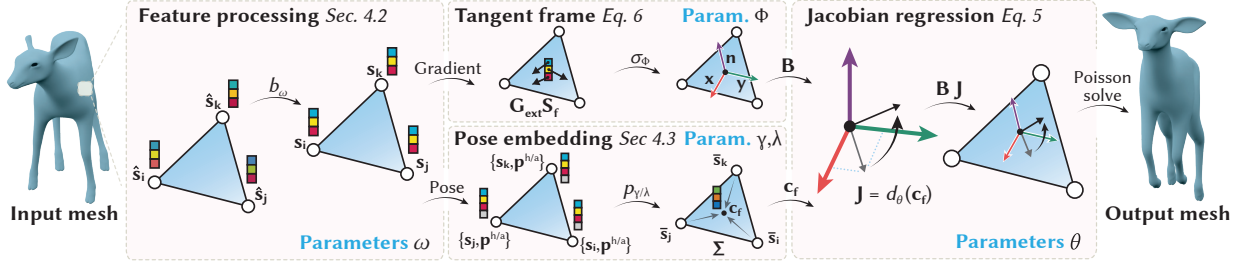


Fig. 3. Overview of our method. As input, we receive a mesh with per-vertex semantic features $\mathbf{s}_i$, which are passed to b_ω for preprocessing. Next, we construct the tangent frame $\mathbf{B}$ by aligning the vector $\mathbf{x}$ with the learned feature gradient (upper middle). We then learn how to deform each face by regressing a Jacobian conditioned on aggregated per-face features $\mathbf{c}_f$ (lower middle). Finally, we align the Jacobians to an ambient space and solve the Poisson equation to obtain the output mesh vertices.

largely rotation-invariant¹. They demonstrate that their features allow for a mapping between similar classes (e.g., horse $\leftrightarrow$ dog) as well as distant classes (e.g., horse $\leftrightarrow$ human), while also separating limbs (right hand vs. left hand) when conditioning an NJF and learning mesh deformations across categories. However, computing per-face features as the averages of associated per-vertex semantic features distilled from image-based feature extractors ‘as-is’ could result in issues. Due to projective occlusions, some surface regions may not receive any features. In addition, a high-frequency variation of extracted features makes the tangent bases sensitive to noise. A known solution is to compute the bases on a smoothed input function or compute a smoothed vector field [Diamanti et al. 2014; Panozzo et al. 2014]. To preclude a manually set smoothing factor and allow additional non-linear filtering, we pass the semantic features through a PoissonNet, b_ω with trained parameters ω , before computing (6) and (4), which also handles the required smoothing. We refer to the raw semantic features as $\hat{\mathbf{S}}$, and the smoothed semantic features as $\mathbf{S}$, such that $b_\omega : \hat{\mathbf{S}} \times \mathcal{M} \rightarrow \mathbf{S}$. Note that, in contrast to a base NJF, the network d_θ is not conditioned on any spatial information and operates purely on the semantic features.

4.3 Learning relative pose deformations

A possible application includes training NJF on a dataset of mesh deformations, each paired with a global pose encoding. At inference, the model can be used to deform unseen meshes for a new pose, without requiring an explicit skeleton or parametrization. Typically, the pose is encoded relative to some ‘canonical’ pose, such as a T-pose. This limits the trained model to inputs in the same canonical pose. Alternatively, one can learn to encode the pose explicitly [Aigerman et al. 2022] or implicitly [Maesumi et al. 2025], and let a network learn how to deform *any* input mesh to an *absolute* target pose. However, training a generalizable pose encoder is challenging on its own and typically requires extrinsic information to encode how body parts are arranged in Euclidean space. Given our localized rotation-equivariant formulation, it is now possible to apply our

method to any starting configuration and target deformation, without learning how to encode the input shape pose. To demonstrate the benefit of this new capability, we condition deformations on *relative* target poses. This makes it possible, for example, to rotate the arm on a mesh in any input pose by a given angle.

In our experiments, we derive relative pose vectors, $\mathbf{p}$, from the kinematic tree of two template models, namely SMPL [Loper et al. 2015] and SMAL [Zuffi et al. 2017]. As these models differ in their number of bones and connectivity, we denote them as $\mathbf{p}^h$ and $\mathbf{p}^a$ for *human* and *animal*, respectively. These vectors encode the relative bone rotations while omitting the global rotation component. Next, they are jointly embedded with the semantic features via an end-to-end learned function. As $\mathbf{p}^h \in \mathbb{R}^{69}$ and $\mathbf{p}^a \in \mathbb{R}^{102}$ differ in size, we parameterize them as two separate, small MLPs:

$$\bar{\mathbf{s}}_j = p_\gamma^h(\mathbf{p}^h, \mathbf{s}_j) \text{ and } \bar{\mathbf{s}}_j = p_\lambda^a(\mathbf{p}^a, \mathbf{s}_j), \quad (8)$$

with $\bar{\mathbf{s}}_j \in \mathbb{R}^S$. These feature vectors are then averaged for each face (Sec. 4.2) and used as inputs to the Jacobian regression network, d_θ , (5). After training, this formulation allows choosing either pose representation to drive the deformation of humanoid or animal meshes alike, as the semantic features implicitly align corresponding parts across classes. Moreover, adopting relative deformations eliminates the need for a canonical pose shared between the two templates and opens a path towards more classes in the future.

4.4 Training

We train d_θ , σ_ϕ , b_ω , p_γ^h , and p_λ^a end-to-end by learning how to deform a random source mesh $\mathcal{M}_s$ to match a target mesh $\mathcal{M}_t$, given the ground-truth relative pose embedding. Parameterizations are discussed in the experiments. Remarkably, we find it sufficient to indirectly supervise σ_ϕ with commonly used task-specific deformation losses to produce smooth frames that generalize across shapes. We predict Jacobians $\mathbf{J}'_i$ (5), solve for the vertex positions (2), and iteratively update the network parameters via gradient descent by minimizing $\mathcal{L} = \mathcal{L}_v + \mathcal{L}_j$ with losses proposed in NJF [Aigerman et al. 2022]. The vertex-vertex loss $\mathcal{L}_v = \sum_i m_i^v \| \mathbf{v}_i^{gt} - \mathbf{v}_i' \|^2$ is weighted by the vertex lumped mass m_i^v and computed after shifting both prediction and ground truth to their center of mass. The Jacobian loss $\mathcal{L}_j = \sum_i a_i \| \mathbf{J}_i^{gt} - \mathbf{J}_i' \|^2$ uses the Frobenius norm and is weighted by face area a_i .

¹We classify the features as “largely rotation-invariant”, because they are distilled from 2D image models that can be sensitive to upside-down inputs: while Dutt et al. [2024, supplement “Robustness to Rotation”] demonstrate the rotation-invariance of such features, Uzolas et al. [2025] observe that performance degrades if meshes flipped upside down. However, we show that our core contributions are compatible with fully rotation-invariant features, such as WKS (see Tbl. 1).

5 Experiments

We first validate our rotation equivariance and compare it to prior work. Then, we demonstrate effective generalization when training on a varied-shape dataset, and also ablate our design decisions. Lastly, we show that our generalized deformation model improves performance in a downstream optimization task. Our code is available at <https://graphics.tudelft.nl/Re-NJF>.

5.1 Rotation equivariance

First, we validate the rotation equivariance property of our method in an isolated setting, where input features and changes in geometry do not play a role. To this end, we overfit an NJF on a single source-target ellipsoidal mesh pair $\mathcal{M}_s - \mathcal{M}_t$ and apply it to $\mathcal{M}_s$ augmented by ambient-space rotations. We expect a *naive* approach matching a vanilla NJF to only perform well with the original $\mathcal{M}_s$. In contrast, our method should yield constant performance across all input rotations.

In this simplified setting, we parameterize σ_ϕ as a linear transformation and d_θ as an MLP. Since no mapping between shapes is needed, we sample a fixed set of random Gaussian features $\mathbf{s}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We follow Sec. 4.4 to train our Re-NJF, but omit b_ω and the pose conditioning, as they are not necessary in this setup. We similarly omit the pose encoder for the vanilla NJF.

At test time, we evaluate both models on $\mathcal{M}_s$ rotated around the Y-axis in constant increments (see Fig. 2) and we confirm that *our* method produces a constant output and error for all rotations. In contrast, the *naive* method faithfully reproduces the target at the training angle of 0° , but fails under ambient space transformations.

5.2 Generalized setting

We demonstrate the effectiveness of our Re-NJF on an end-to-end, generalized task: learning relative pose deformations for arbitrary humanoid and animal meshes. We learn generalizable deformations across classes by leveraging semantic features. Importantly, thanks to rotation equivariance, we do not need to train a pose encoder or align meshes to a canonical pose during training or inference.

Data. We construct a dataset with humanoid and animal shapes (see Fig. 5). For humans, we randomly sample body shapes from the STAR model [Osman et al. 2020] and use poses from the AMASS dataset [Mahmood et al. 2019]. For animals, we randomly sample shapes from the SMAL model [Biggs et al. 2020; Zuffi et al. 2017] and obtain poses from Animal3D [Xu et al. 2023] with additional augmentation. Our training source-target mesh pairs match in shape, but differ in pose. Within each category (humans and animals), all meshes share the same order of vertices and the same connectivity, which makes their per-vertex features interchangeable. Therefore, we pair source meshes with randomly sampled features from a pre-computed set during training to reduce the feature computation time. We follow this to build a training set as well as separate validation and test sets with distinct poses and features for quantitative evaluation. For qualitative evaluation, we further test on unseen animals from DT4D [Li et al. 2021] and TOSCA [Bronstein et al. 2008], as well as humans with challenging high-frequency surface details from CAPE [Ma et al. 2020; Pons-Moll et al. 2017], and other humanoids [Maesumi et al. 2025]. See Supplement B.2 for details.

Baselines and implementation. We benchmark our method as a drop-in extension (“+ Rot.-Eq.”) of mesh deformation heads d_θ in prior works. In PoissonNet [Maesumi et al. 2025], d_θ is parameterized as a modulated Vector MLP (*VectorMLP*). In NJF [Aigerman et al. 2022] and SAF3D [Uzolas et al. 2025], d_θ is an standard *MLP*. We adapt unrestricted Jacobian $\mathbf{J}'_i \in \mathbb{R}^{3 \times 3}$ formulation to solve (2) when “+ Rot.-Eq.” and *MLP* are used. See Supplement A for a discussion.

We further evaluate the performance difference between the deformation heads when using the vertex positions (*XYZ*) as input, compared to the semantic features denoted as *S*. Parameterizations of σ_ϕ are evaluated and discussed in the ablations.

We preprocess the semantic features with b_ω , which is parameterized as three Poisson blocks with a hidden dimension size of 128, each deformation head contains 5 layers of size 128, p_λ^a , and p_λ^b are two-layer MLPs of size 128. See Supplement B.1 for more details. We train all networks end-to-end with a batch size of 16, where each batch consists entirely of either human or animal meshes, making use of the AdamW optimizer [Loshchilov and Hutter 2019]. Due to the pose imbalance, we further sample animal meshes with only a 30% probability. We find that the models stabilize with respect to their validation metrics after around 120k iterations, and therefore choose the model for final evaluation at 150k training iterations.

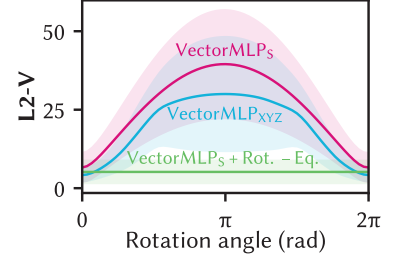
Metrics. We report three common metrics [Aigerman et al. 2022] while comparing predictions for 2000 test pairs to ground truth: **L2-V** as the average L2 error between sphere-normalized vertices scaled by 10^2 ; **L2-J** as the average L2 error between Jacobians scaled by 10; and **L2-N** as the average L2 error between face angles between prediction and target in degrees. See Supplement B.3 for details.

Results. We evaluate all methods with source test meshes undergoing a full range of rotations around a vertical axis. Rotations are restricted to a single axis for visualization clarity only. However, our contribution will lead to the same result for any general rotation (provided the input features are available). We do not recompute the features for each new angle.

Only our extension (“+ Rot.-Eq.”) produces consistent results across source mesh rotations. This leads to a competitive performance for the training poses (see Fig. 4b, “No Rotation”) and a superior mean performance when all poses are considered (“With Rotation”). In contrast, the baselines produce high errors for source mesh orientations different from the training pose (0°), which is qualitatively reflected in unnatural outputs seen in Fig. 5. Additionally, we generally observe a slight degradation in performance when conditioning on semantic features (*S*) rather than world-space coordinates (*XYZ*) (see ears of the cow in the first row and the hands of the human in the third row in Fig. 5). However, we consider this trade-off well compensated by cross-class generalization enabled by *S*. Finally, we find that the *VectorMLP* deformation head generally outperforms *MLP*. Therefore, we use the *VectorMLP_S* + *Rot.Eq.* as our Re-NJF model in all consecutive experiments.

Smooth basis. In Fig. 6, we observe that our learned tangents $\mathbf{x}_i$ (6) form smooth vector fields. A comparison of the animal’s front right legs to the human’s right hand showcases the similarity of directions between paws and hands across classes. This allows our method to learn generalized deformation between diverse meshes.

	With rotation			No rotation		
	L2-V	L2-J	L2-N	L2-V	L2-J	L2-N
MLP _S + Rot.-Eq.	5.36±3.94	4.62±1.89	14.59±6.72	5.36±3.94	4.62±1.89	14.59±6.72
MLP _S	22.53±14.07	9.38±3.73	32.26±15.45	5.74±3.60	4.84±1.91	13.36±6.03
MLP _{XYZ}	19.65±15.04	8.36±3.98	30.45±17.75	<u>4.43±2.39</u>	<u>4.03±1.65</u>	<u>11.28±5.17</u>
VectorMLP _S + Rot.-Eq.	5.18±3.92	4.25±1.77	13.48±6.23	5.18±3.92	4.25±1.77	13.48±6.23
VectorMLP _S	26.17±16.86	9.86±4.66	35.01±17.04	6.60±4.94	4.46±1.96	14.97±7.51
VectorMLP _{XYZ}	20.02±15.41	7.87±4.09	31.49±18.63	4.27±2.39	2.98±1.29	9.87±4.33



(a) ‘With Rotation’ values are averaged across 360° range of source-mesh rotations (60 steps). ‘No Rotation’ only reports the results at the training pose of 0°. Our method (“+ Rot.-Eq.”) achieves consistent results across all angles. We select *VectorMLP_S + Rot.-Eq.* for the following experiments. (b) The baseline’s L2-V errors (red, blue) increase with rotation, while ours (green) remains constant.

Fig. 4. Test of different deformation heads (*VectorMLP* [Maesumi et al. 2025] and *MLP* [Aigerman et al. 2022; Uzolas et al. 2025]) and input features (*XYZ* [Maesumi et al. 2025] and *S* [Uzolas et al. 2025]) either with our method (“+ Rot.-Eq.”), or without our method.

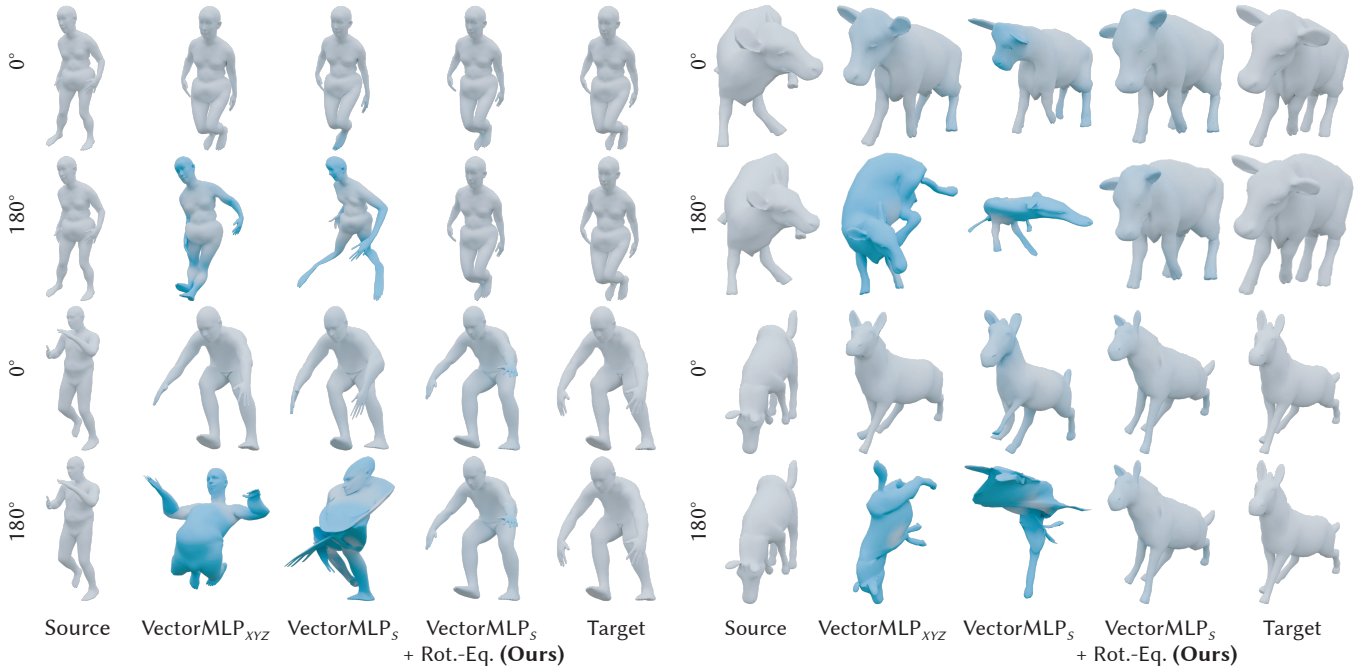


Fig. 5. Qualitative comparison of *our* extension (“+ Rot.-Eq.”) with the baselines when deforming a test *Source* mesh to a *Target* mesh. While all methods produce plausible results at a training orientation (0°), only our method produces consistent results when the *Source* mesh is rotated 180° around its vertical axis before the deformation is applied. We rotate the mesh back after deformation for visualization purposes. Blue color denotes per-vertex L2-V errors.

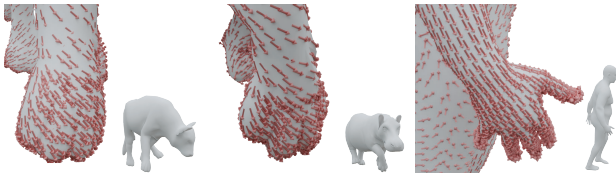


Fig. 6. Visualization of learned tangents $\mathbf{x}_i$ for three shapes. The cross-class consistency supports generalized deformation transfer.

Generalized deformation results. In Fig. 7, we demonstrate successful cross-dataset generalization by deforming shapes from datasets

unused in training. By manually defining pose vectors $\mathbf{p}^h$ and $\mathbf{p}^a$, we can repose even unrigged meshes. This is enabled by our relative pose formulation, as it removes the need for absolute alignment of the source mesh poses during training and inference. Moreover, thanks to the semantic features, our method produces realistic deformations not only for pose codes matching the shape class (*within-class*, top row), but also for mismatched pose and shape classes (*in-between-class*, bottom row). This allows us to apply SMLP motion sequences from the test set of the AMASS human capture dataset [Mahmood et al. 2019] to both animal and humanoid shapes

without their prior alignment (see Fig. 8). The produced motion sequences are smooth and natural, even for unseen shapes.

Training orientation. Beyond inference generalization, we demonstrate the benefits of Re-NJF for training. For this purpose, we sample random global rotations across all axes and consistently augment source and training shapes, $\mathcal{M}_s$ and $\mathcal{M}_t$, during training. In Fig. 9, we show that only our method exhibits convergence behavior identical to training without random rotations, whereas the other baseline methods converge considerably slower or fail to converge at all. We find that the baselines struggle to converge under this augmentation as they are unable to learn the augmented source shape space. The nearly constant learning curve of *VectorMLP_S* underpins that semantic features alone cannot disambiguate the source orientation. In contrast, the performance of our method combining these features with our equivariant formulation is not affected by the augmentation.

5.3 Ablations

We ablate other important design choices included in our method quantitatively in a *within-class* setup (Tbl. 1) and qualitatively in a *in-between-class* setup (Fig. 10). First, we find that replacing the semantic features S [Uzolas et al. 2025] with Wave Kernel Signatures (WKS) [Aubry et al. 2011] retains rotation equivariance and good performance *within-class*. However, unlike Re-NJF with S , WKS struggle to generalize *in-between-class*. This demonstrates the semantic features’ value for generalized deformation learning across widely varying shapes.

Second, we evaluate three alternatives for our local tangent frames $\mathbf{B}_i$: a global frame defined as the orthonormal basis obtained by the three principal components originating from PCA, accounting for consistent orientation and handedness (*Global*); local frames calculated by setting $\mathbf{x}_i$ (6) to the first principal curvature direction (*PC*); and local frames calculated by replacing $\mathbf{x}_i$ consistently with one of the triangle’s edges (*Face*). We find that our full Re-NJF outperforms all three alternatives both *within* and *in-between-class*. While *Global* frames also reproduce plausible generalized targets, they feature noticeable local artifacts (see Fig. 10).

Lastly, we show that removing the feature preprocessor b_ω (*no b_ω*), as well as using an absolute rather than our relative pose formulation (*absolute pose*), both substantially increase the errors. A visual example of the effect of the feature processing is shown in Supplement C.

5.4 Application: Inverse kinematics without rigging

We demonstrate an application of our trained Re-NJF for inverse kinematics. Here, in contrast to direct pose control, we optimize for a pose yielding a deformation satisfying a target constraint. For example, we articulate a human for a given hand reach. This typically requires a mesh rigged with a skeleton. However, since our Re-NJF represents a space of expected deformations learned from skeletal data, we can use it in place of common kinematic techniques, such as Linear Blend Skinning (LBS), to uniquely optimize poses even for unrigged meshes without assigned skeletons.

Table 1. Quantitative ablation results in a *within-class* setup where shape and pose classes match. Results are averaged across all angles, equivalent to “With Rotation” in Tbl. 4a. First, we compare different input features for σ_ϕ , as well as using an absolute pose as input. Second, we evaluate alternative parameterizations of the local tangent frames. Lastly, we assess the impact of omitting b_ω .

Variant	L2-V	L2-J	L2-N
Ours	5.18±3.92	4.25±1.77	13.48±6.23
INPUT FEATURES AND POSE			
+ WKS	5.70±3.68	4.48±2.17	14.41±7.74
+ absolute pose	6.70±4.38	4.87±1.94	16.29±6.73
COORDINATE FRAME			
+ Global	11.86±8.63	7.31±2.65	26.03±11.21
+ PC	9.29±5.48	6.63±2.51	20.92±9.43
+ Face	8.18±4.95	6.66±2.41	20.68±8.81
FEATURE PROCESSING			
+ No b_ω	8.69±4.70	6.15±2.16	21.54±8.58

Optimization. Let $\Psi(\mathbf{v}; \mathbf{p}) \rightarrow \mathbb{R}^3$ be a function that deforms mesh vertex $\mathbf{v}$ based on a pose vector $\mathbf{p}$ by solving the Poisson equation (2) for Jacobians produced by our Re-NJF. We solve for $\mathbf{p}$ that minimizes $E_{ik} + E_{ARAP}$. Here, E_{ARAP} is an as-rigid-as-possible regularizer [Sorkine and Alexa 2007] implemented by penalizing the deviation of the singular values of $\mathbf{J}'_i$ from 1 [Rabinovich et al. 2017], and $E_{ik}(\mathbf{p}) = \sum_i m_i^v \|\mathbf{v}_i^{tgt} - \Psi(\mathbf{v}_i; \mathbf{p})\|^2$ enforces the deformation for a specified target position $\mathbf{v}_i^{tgt}$ of a vertex $\mathbf{v}_i$ with a lumped mass m_i^v . The vertex pairs are user-defined by selecting all vertices contained within two specified spheres (see Fig. 11). We use a differentiable least-squares Poisson solver [Aigerman et al. 2022; Maesumi et al. 2025] and optimize the pose vectors end-to-end for 500 iterations with the AdamW optimizer and a learning rate of 0.005.

Baselines. We compare our Re-NJF against a variant of Ψ with *VectorMLP_{XYZ}* to demonstrate the benefit of rotation equivariance in this setup. Note that the input source meshes are purposefully rotated to highlight the shortcomings of *VectorMLP_{XYZ}*. We further compare against a baseline (*ARAP-only*), where we directly optimize the relative per-face Jacobians with an objective $E_{ik} + 1000 E_{ARAP}$. The higher E_{ARAP} weight compensates for the lack of model constraint.

Results. While all methods shown in Fig. 11 succeed in placing the vertices at the correct targets, our Re-NJF produces the best visual results, such as a realistic bend at the elbow to enforce the hand’s position (top row), or a natural bend of the bear’s leg and torso (bottom row). In contrast, *VectorMLP_{XYZ}* produces spurious deformations of other limbs, and *ARAP-only* is not able to preserve the volume of the limbs (top row). This demonstrates the importance of rotation equivariance also for inverse problems and suggests that Re-NJF could help with restricting deformations in other optimization tasks. This comes at a cost of only a modest computational time increase from 48.6 ms (*VectorMLP_{XYZ}*) to 56.1 ms (ours) per optimization step on an RTX 3090.

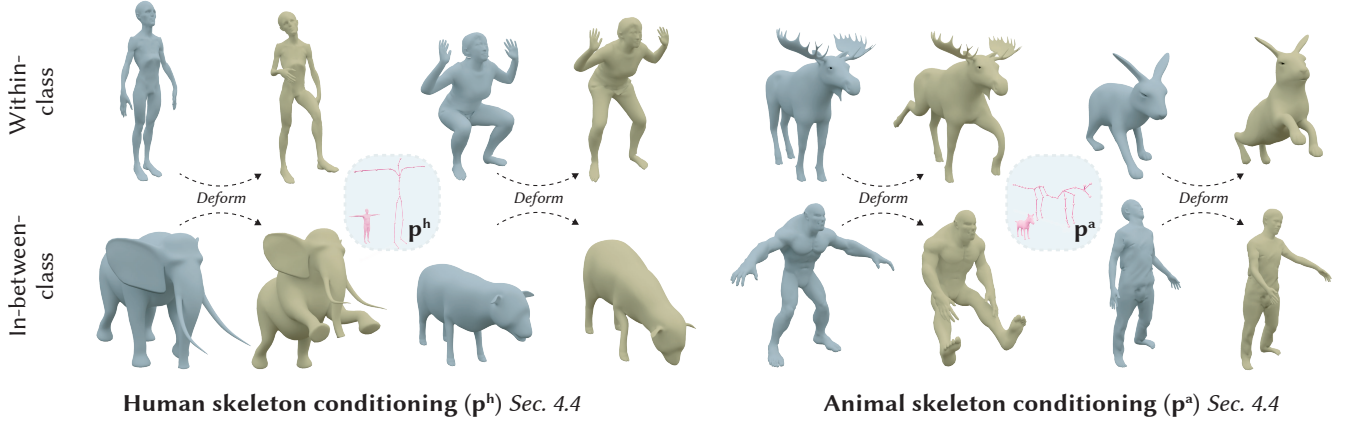


Fig. 7. Results of our Re-NJF when deforming humanoid [Ma et al. 2020; Maesumi et al. 2025] and animal [Li et al. 2021] shapes from datasets unused in training. We manually define pose conditioning vectors from either the human skeletal model (p^h) or the animal skeletal model (p^a) (the middle insets) and mix them across humanoid and animal shape classes. In the top row, the skeleton and shape classes match (“*Within-class*”), while in the bottom row, they are swapped (“*In-between-class*”). Our method achieves a realistic reposing of animal meshes, even with the human conditioning signal, and vice versa.

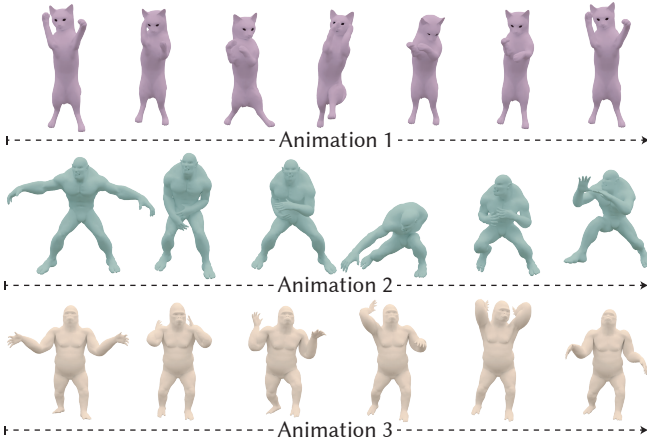


Fig. 8. Three different animation sequences from the AMASS test set [Mahmood et al. 2019] parameterized by SMPL bone rotations applied to meshes not seen during training.

6 Conclusion

We presented Re-NJF, a rotation-equivariant formulation of Neural Jacobian Fields. We achieve rotation equivariance by expressing the Jacobians in local tangent frames aligned with the semantic feature gradients. Our formulation is independent of the input and output ambient space orientations, as well as initial mesh poses, allowing us to apply deformations at inference time without aligning meshes to a predefined training pose, e.g., a T-pose. Because our tangent frames are aligned to semantic feature gradients, our approach generalizes across widely varying shapes and enables learning deformations across shape classes such as humans and animals.

Limitations and future work. Our pipeline uses semantic features distilled from 2D projections. Meshes with strong occlusions and intersections, such as those in the MOYO dataset [Tripathi et al.

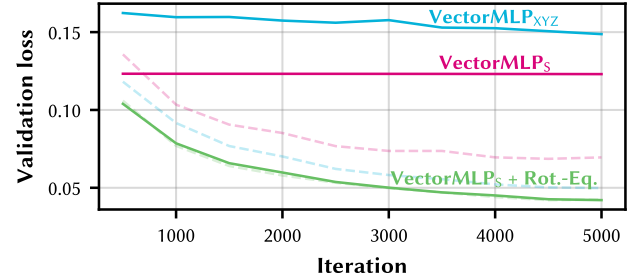


Fig. 9. Learning curves when training with the original training set (dashed) and with rotationally augmented training set (solid) based on our training loss for corresponding validation sets with a matching augmentation condition. Only our method (green) demonstrates the same convergence behavior under both conditions (curves overlap).

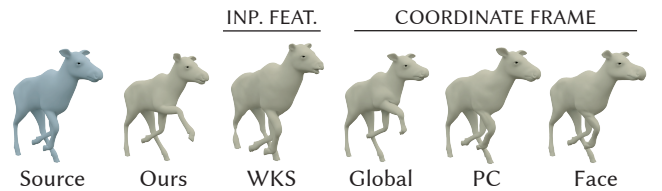


Fig. 10. Qualitative ablation of our design choices when applying a relative human pose vector p^h to an animal shape in an *in-between-class* setup. p^h is manually defined as a 90° human collar bone rotation. WKS represents alternative input features, while Global, PC and Face are alternative tangent frame formulations.

2023] used in PoissonNet [Maesumi et al. 2025], yield poor semantic features. Despite the smoothing, near-zero gradients from poor semantic features can leave the tangent vectors prone to sign flips. Furthermore, computation time is a bottleneck, as extracting the

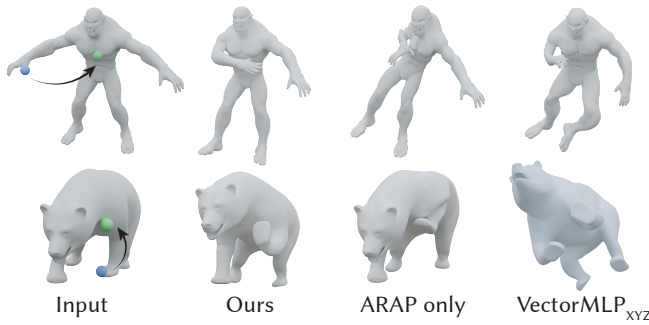


Fig. 11. Results of our inverse kinematics pipeline. The blue and green spheres mark the initial and target surface points. End-to-end optimization of the pose embedding in Re-NJF (Ours) produces articulated deformations despite the meshes not being rigged.

semantic features takes several minutes per mesh. However, as our method is compatible with other intrinsic features, it would, therefore, benefit directly from improvements in the quality and efficiency of semantic features. We have focused on relative reposing, given relative pose encodings, because it is difficult to encode absolute poses with only intrinsic information. Therefore, our method could benefit from an improved absolute pose encoder. Lastly, our method inherits some limitations of NJF and PoissonNet, for example, a restriction to single-component meshes. The rotation equivariance contribution itself is not limited to single-component meshes.

Acknowledgments

Thanks to Justin Solomon for an early discussion on the topic.

References

- Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Trans. Graph.* 39, 4 (2020), 62–1. doi:10.1145/3386569.3392462
- Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Trans. Graph.* 41, 4 (2022), 1–17. doi:10.1145/3528223.3530141
- Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. 2022. Deep ViT Features as Dense Visual Descriptors. (2022). arXiv:2112.05814 [cs.CV]
- Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 1626–1633. doi:10.1109/ICCVW.2011.6130444
- David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. 43–54. doi:10.1145/280814.280821
- Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. 2020. Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop. In *European Conference on Computer Vision*. Springer, 195–211. doi:10.1007/978-3-030-58621-8_12
- Mario Botsch and Olga Sorkine. 2008. On Linear Variational Surface Deformation Methods. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (Jan. 2008), 213–230. doi:10.1109/TVCG.2007.1054
- Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. 2008. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media. doi:10.1007/978-0-387-73301-2
- Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velicković. 2021. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv:2104.13478 [cs.LG]
- Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42. doi:10.1109/MSP.2017.2693418

- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9650–9660. doi:10.1109/ICCV48922.2021.00951
- Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. 2019. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 1321–1330. <http://proceedings.mlr.press/v97/cohen19d.html>
- Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. 2021. Gauge Equivariant Mesh CNNs: Anisotropic convolutions on geometric graphs. In *9th International Conference on Learning Representations, ICLR 2021*. <https://openreview.net/forum?id=Jnspzp-oIzE>
- Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-PolyVector Fields with Complex Polynomials. *Computer Graphics Forum* 33, 5 (2014), 1–11. doi:10.1111/CGF.12426
- Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. 2024. Diffusion 3d features (Diff3F): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4494–4504. doi:10.1109/CVPR52733.2024.00430
- Michael Fischer, Zhengqin Li, Thu Nguyen-Phuoc, Aljaz Bozic, Zhao Dong, Carl Marshall, and Tobias Ritschel. 2024. NeRF Analogies: Example-Based Visual Attribute Transfer for NeRFs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4640–4650. doi:10.1109/CVPR52733.2024.00444
- Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L Rosin, Weiwei Xu, and Shihong Xia. 2018. Automatic unpaired shape deformation transfer. *ACM Trans. Graph.* 37, 6 (2018), 1–15. doi:10.1145/3272127.3275028
- Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, Vol. 28.
- Tao Ju, Scott Schaefer, and Joe Warren. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3 (July 2005), 561–566. doi:10.1145/1073204.1073229
- Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 19729–19739. doi:10.1109/ICCV51070.2023.01807
- Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net. <https://openreview.net/forum?id=SJU4ayYgl>
- John P Lewis, Matt Cordner, and Nickson Fong. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 165–172. doi:10.1145/344779.344862
- Tianyu Li, Jungdam Won, Alexander Legg, Jeonhwan Kim, Akshara Rai, and Sehoon Ha. 2023. Ace: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In *SIGGRAPH Asia 2023 Conference Papers*. doi:10.1145/3610548.3618255
- 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/CVF International Conference on Computer Vision (ICCV)*. 12706–12716. doi:10.1109/ICCV48922.2021.01247
- Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. 2025. PartField: Learning 3D Feature Fields for Part Segmentation and Beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9704–9715. doi:10.1109/ICCV51701.2025.00905
- Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. 2023. PartSLIP: Low-Shot Part Segmentation for 3D Point Clouds via Pretrained Image-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21736–21746. doi:10.1109/CVPR52729.2023.02082
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics* 34, 6 (2015), 248:1–248:16. doi:10.1145/2816795.2818013
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019*. <https://openreview.net/forum?id=Bkg6RiCqY7>
- Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. 2020. Learning to Dress 3D People in Generative Clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/CVPR42600.2020.00650
- Arman Maesumi, Tanish Makadia, Thibault Groueix, Vladimir Kim, Daniel Ritchie, and Noam Aigerman. 2025. PoissonNet: A Local-Global Approach for Learning on Surfaces. *ACM Trans. Graph.* 44, 6 (2025), 1–16. doi:10.1145/3763298
- Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 5442–5451.

- doi:10.1109/ICCV.2019.00554
- Thomas W Mitchell, Vladimir G Kim, and Michael Kazhdan. 2021. Field Convolutions for Surface CNNs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 10001–10011. doi:10.1109/ICCV48922.2021.00985
- Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. 2024. Neural Semantic Surface Maps. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e15005. doi:10.1111/CGF.15005
- Sanjeev Muralikrishnan, Niladri Dutt, Siddhartha Chaudhuri, Noam Aigerman, Vladimir Kim, Matthew Fisher, and Niloy J Mitra. 2024. Temporal Residual Jacobians for Rig-Free Motion Transfer. In *European Conference on Computer Vision*. Springer, 93–109. doi:10.1007/978-3-031-73636-0_6
- Sanjeev Muralikrishnan, Niladri Shekhar Dutt, and Niloy J Mitra. 2025. SMF: Template-free and Rig-free Animation Transfer using Kinetic Codes. *ACM Trans. Graph.* 44, 6 (2025), 1–11. doi:10.1145/3763309
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research* (2024). <https://openreview.net/forum?id=a68SUt6zFt>
- Ahmed AA Osman, Timo Bolkart, and Michael J Black. 2020. STAR: Sparse Trained Articulated Human Body Regressor. In *European Conference on Computer Vision*. Springer, 598–613. doi:10.1007/978-3-030-58539-6_36
- Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame fields: anisotropic and non-orthogonal cross fields. *ACM Trans. Graph.* 33, 4 (2014), 134:1–134:11. doi:10.1145/2601097.2601179
- Frederick I. Parke. 1972. Computer generated animation of faces. In *Proceedings of the ACM Annual Conference - Volume 1*. 451–457. doi:10.1145/800193.569955
- Frederic Ira Parke. 1974. *A parametric model for human faces*. The University of Utah.
- Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Trans. Graph.* 36, 4 (2017). doi:10.1145/3072959.3073711
- Adrien Poulenard and Maks Ovsjanikov. 2018. Multi-directional geodesic neural networks via equivariant convolution. *ACM Trans. Graph.* 37, 6 (2018), 1–14. doi:10.1145/3272127.3275102
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 652–660. doi:10.1109/CVPR.2017.16
- Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 4 (2017), 1. doi:10.1145/2983621
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695. doi:10.1109/CVPR52688.2022.01042
- Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Trans. Graph.* 36, 6 (2017). doi:10.1145/3130800.3130883
- Nicholas Sharp, Souhaib Attaki, Keenan Crane, and Maks Ovsjanikov. 2022. Diffusion-Net: Discretization Agnostic Learning on Surfaces. *ACM Trans. Graph.* 41, 3 (2022), 27:1–27:16. doi:10.1145/3507905
- Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2024. SHIC: Shape-Image Correspondences with No Keypoint Supervision. In *European Conference on Computer Vision*. Springer, 129–145. doi:10.1007/978-3-031-72913-3_8
- Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. 2026. DINOv3. *Transactions on Machine Learning Research* (2026). <https://openreview.net/forum?id=2NlGyqNjns>
- Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Geometry Processing*. The Eurographics Association. doi:10.2312/SGP/SGP07/109-116
- Olga Sorkine and Mario Botsch. 2009. Interactive Shape Modeling and Deformation. In *Eurographics 2009 - Tutorials*. The Eurographics Association. doi:10.2312/egt.20091068
- Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 3 (2004), 399–405. doi:10.1145/1015706.1015736
- Robert W. Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. 2005. Mesh-based inverse kinematics. *ACM Trans. Graph.* 24, 3 (2005), 488–495. doi:10.1145/1073204.1073218
- Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. 2009. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In *Computer Graphics Forum*, Vol. 28. 1383–1392. doi:10.1111/J.1467-8659.2009.01515.X
- Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. 2022. MotionCLIP: Exposing Human Motion Generation to CLIP Space. In *European Conference on Computer Vision*. Springer, 358–374. doi:10.1007/978-3-031-20047-2_21
- Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. 2018. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. (2018). arXiv:1802.08219 [cs.LG]
- Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 2023. 3D Human Pose Estimation via Intuitive Physics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4713–4725. doi:10.1109/CVPR52729.2023.00457
- Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. 2025. Surface-Aware Distilled 3D Semantic Features. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–12. doi:10.1145/3757377.3763974
- Tobias Weißberg, Weikang Wang, Paul Roetzer, Nafie El Amrani, and Florian Bernard. 2026. Symmetry Informative and Agnostic Feature Disentanglement for 3D Shapes. In *2026 International Conference on 3D Vision (3DV)*. doi:10.1109/3DV69130.2026.00025
- Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. 2020. CNNs on surfaces using rotation-equivariant features. *ACM Trans. Graph.* 39, 4, Article 92 (Aug. 2020), 12 pages. doi:10.1145/3386569.3392437
- Ruben Wiersma, Ahmad Nasikun, Elmar Eisemann, and Klaus Hildebrandt. 2022. Delta-Conv: anisotropic operators for geometric deep learning on point clouds. *ACM Trans. Graph.* 41, 4 (2022), 105:1–105:10. doi:10.1145/3528223.3530166
- Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. 2024. Back to 3D: Few-Shot 3D Keypoint Detection with Back-Projected 2D Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4154–4164. doi:10.1109/CVPR52733.2024.00398
- Jiacong Xu, Yi Zhang, Jiawei Peng, Wufei Ma, Artur Jesslen, Pengliang Ji, Qixin Hu, Jiehua Zhang, Qihao Liu, Jiahao Wang, et al. 2023. Animal3D: A Comprehensive Dataset of 3D Animal Pose and Shape. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9099–9109. doi:10.1109/ICCV51070.2023.00835
- Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y. Lam, Yan-Pei Cao, and Xihui Liu. 2024. SAMPart3D: Segment Any Part in 3D Objects. arXiv:2411.07184 [cs.CV]
- Seungwoo Yoo, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. 2024. Neural Pose Representation Learning for Generating and Transferring Non-Rigid Object Poses. *Advances in Neural Information Processing Systems* 37 (2024). doi:10.52202/079017-1083
- Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. 2024. Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3076–3085. doi:10.1109/CVPR52733.2024.00297
- Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 2017. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5524–5532. doi:10.1109/CVPR.2017.586

Supplements: Rotation-Equivariant Neural Jacobian Fields for Generalizable 3D Mesh Deformations

LUKAS UZOLAS, Delft University of Technology, Netherlands

RUBEN WIERSMA, ETH Zurich, Switzerland and Adobe Research, France

OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

PETR KELLNHOFER, Delft University of Technology, Netherlands

A Poisson equation with extrinsic gradient operator

PoissonNet regresses restricted Jacobians $J'_i \in \mathbb{R}^{2 \times 3}$, while NJF uses unrestricted Jacobians in their implementation, i.e., $J'_i \in \mathbb{R}^{3 \times 3}$. When extending PoissonNet to use our proposed method and when using NJF, we solve the linear system (see (2)) with unrestricted Jacobians.

In our implementation, we use an extrinsic gradient operator. Here, we provide the derivation of the corresponding Poisson equation. For convenience, we set up a coordinate transform matrix that composes the local coordinate systems in a $3F \times 3F$ block-diagonal matrix.

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_N \end{bmatrix}, \quad (1)$$

where $\mathbf{B}$ is orthonormal by construction, i.e., $\mathbf{B}^\top \mathbf{B} = \mathbf{B} \mathbf{B}^\top = \mathbf{I}$.

We seek to rewrite the following Poisson equation in the ambient coordinate system

$$\mathbf{L}\mathbf{V}' = \mathbf{G}^\top \mathbf{A} \mathbf{J}'^\top, \quad (2)$$

where $\mathbf{G}$ is a $2F \times V$ matrix, $\mathbf{J}'$ a $3 \times 2F$ matrix and $\mathbf{A}$ is a $2F \times 2F$ diagonal matrix with twice-repeated triangle areas.

We can write the extrinsic gradient using a coordinate transform. The intrinsic gradient outputs tangent vectors, resulting in the $2F$ row dimension. For ease of notation, we add a row/column of zeros for the normal direction to both the gradient and Jacobian matrices, such that $\mathbf{G}$ is of size $3F \times V$, $\mathbf{J}$ is of size $3 \times 3F$ and $\mathbf{A}$ is of size $3F \times 3F$. Then, we can write the following identities

$$\mathbf{G}_{\text{ext}} = \mathbf{B}\mathbf{G}, \quad (3)$$

$$\mathbf{G} = \mathbf{B}^\top \mathbf{G}_{\text{ext}}. \quad (4)$$

These identities let us rewrite the Poisson equation (2) as

$$\mathbf{L}\mathbf{V}' = \mathbf{G}_{\text{ext}}^\top \mathbf{B} \mathbf{A} \mathbf{J}'^\top. \quad (5)$$

Note that $\mathbf{J}'$ is a composition of per-face Jacobians computed as $J'_i = \mathbf{B}_i d_\theta(c_i)$. We factor out $\mathbf{B}$ by setting $\tilde{J}'_i = d_\theta(c_i)$ and $\mathbf{J}'_i = \mathbf{B}_i \tilde{J}'_i$.

Authors' Contact Information: Lukas Uzolas, Delft University of Technology, Netherlands, luzolas@tudelft.nl; Ruben Wiersma, ETH Zurich, Switzerland and Adobe Research, France, rubenwiersma@gmail.com; Olga Sorkine-Hornung, ETH Zurich, Switzerland, sorkine@inf.ethz.ch; Petr Kellnhöfer, Delft University of Technology, Netherlands, p.kellnhöfer@tudelft.nl.

SA Conference Papers '26, Kuala Lumpur, Malaysia

© 2026 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26)*, December 01–04, 2026, Kuala Lumpur, Malaysia, <https://doi.org/10.1145/3829340.3842184>.

Equation 5 can then be written as

$$\mathbf{L}\mathbf{V}' = \mathbf{G}_{\text{ext}}^\top \mathbf{A} \mathbf{B} \tilde{J}'^\top \mathbf{B}^\top, \quad (6)$$

which we use in our implementation. $\tilde{J}'$ and the underlying neural network d_θ operate entirely in the local coordinate frame, as they are wedged between $\mathbf{B}$ and $\mathbf{B}^\top$, resulting in the desired rotation-equivariance property.

B Implementation details

B.1 Models

For our generalized models, we parametrize $b_\omega: \hat{\mathbf{S}} \times \mathcal{M} \rightarrow \mathbf{S}$ as a PoissonNet consisting of three Poisson blocks, each containing three modulated VectorMLPs with a hidden dimensionality of 128. Since the semantic features have a size of 256, we embed them into a 128-dimensional space using a learned linear layer. $\sigma_\phi: \mathbb{R}^S \rightarrow \mathbb{R}^3$ is parametrized as a simple linear layer without an activation function and normalization layer. We utilize two different deformation heads for $d_\theta: \mathbb{R}^S \rightarrow \mathbb{R}^{3 \times 2}$, either a 5-layer VectorMLP or a standard MLP. In the case of the MLP, we parametrize d_θ as 5 linear layers with LayerNorm normalization [Ba et al. 2016], and GeLU activation function [Hendrycks and Gimpel 2016], except for the last layer. The relative pose embedding networks, $p_\lambda^a: \mathbb{R}^{69} \times \mathbb{R}^S \rightarrow \mathbb{R}^S$ and $p_\lambda^h: \mathbb{R}^{102} \times \mathbb{R}^S \rightarrow \mathbb{R}^S$, are parametrized as two linear layers with LayerNorm and GeLU activation after the first layer. In Sec 5.1, d_θ is parameterized as a 5-layer MLP

B.2 Data

Each dataset entry consists of a pair of source and target meshes, corresponding to the same *shape* parameters, β , but differing in their *pose* parameters, $\mathbf{p}$ (see Sec 4.3). A mesh is then obtained via $\mathcal{M}^a = \text{SMAL}(\mathbf{p}^a, \beta^a)$, and likewise for the humans $\mathcal{M}^h = \text{STAR}(\mathbf{p}^h, \beta^h)$, where the superscript denotes either animal or human.

Poses. AMASS [Mahmood et al. 2019] contains 2,848,368 *absolute* training, 253,972 validation, and 20,848 test poses. During training, we randomly sample poses from the training set with uniform probability for each source-target pair. Animal3D [Xu et al. 2023], on the other hand, only contains 3,065 training and 320 test poses. As the animal pose dataset is several orders of magnitude smaller than the human pose dataset, we devise a different sampling strategy for the training animal poses. We fit a Gaussian distribution to the bone angles in the dataset and then randomly sample angles from it instead. However, we maintain the original poses for validation and the test set. For the validation set, we use 250 human and 250 animal validation pairs. Note that we split 100 validation samples from the train set for animals, as no separate validation set exists.

Our test set contains 2,000 pairs, with equal numbers of animals and humans.

We calculate *relative* poses, $\mathbf{p}^h$ and $\mathbf{p}^a$, by first converting each bone b 's rotation from Rodrigues notation to a rotation matrix $\mathbf{R}_b$. Each relative bone rotation is then given by $\mathbf{R}_b^{rel} = \mathbf{R}_{b,src}^\top \mathbf{R}_{b,tgt}$, where $\mathbf{R}_{b,src}$ is the absolute rotation of bone b in the source pose, and $\mathbf{R}_{b,tgt}$ is the corresponding rotation in the target pose. The relative rotations for each bone are transformed back to Rodrigues notation and concatenated to form $\mathbf{p}^h$ and $\mathbf{p}^a$.

Shapes. We sample human shapes parameters β^h of the human parametric SMPL model [Loper et al. 2015] from a Gaussian distribution: $\beta^h \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with $\beta^h \in \mathbb{R}^{100}$. In addition, we randomly sample the gender with uniform probability. For animal shapes from SMAL [Zuffi et al. 2017], we sample from a multimodal Gaussian distribution: $\beta^a \sim \mathcal{N}(\mu_c, \frac{1}{2}\mathbf{I})$, where μ_c denotes the mean shape of each animal class and is sampled from the set $\mathcal{A} := \{\mu_{hippo}, \mu_{cat}, \mu_{dog}, \mu_{cow}, \mu_{horse}\}$ with equal probability for each class. We only use the top 20 principle shape components and set $\beta^a \in \mathbb{R}^{20}$, which is smaller than the human shape space, as we observe stronger artifacts with SMAL when using higher-dimensional shape parameters.

Features. To mitigate the high computational cost of distilled semantic features during training [Dutt et al. 2024; Uzolas et al. 2025; Wimmer et al. 2024], we precompute representative subsets of unique features and stochastically apply them across our datasets. We utilize the pre-trained 'All-shape' model [Uzolas et al. 2025] for extracting the semantic features. Specifically, we extract semantic features for 100 human- and 100 animal training meshes, forming two feature sets $\mathcal{S}^a = \{\mathbf{S}_i^a\}_{i=1}^{100}$ and $\mathcal{S}^h = \{\mathbf{S}_i^h\}_{i=1}^{100}$. Each element in either feature set has the same size because the mesh topology is shared within the shape class. In addition, we construct separate validation and test sets for the features and poses used for quantitative evaluation. Since the train, validation and test sets are constructed separately, there is no overlap of pre-computed features between them. As such, our experiment demonstrates that even while training on a smaller set of reused features compared to the shape sample count, our method achieves generalization both between the training and test sets, as well as between shape classes. Moreover, due to their smaller size, we compute unique sample-specific features for each instance during experiments in datasets other than AMASS and Animal3D. This further demonstrates that our learning strategy does not restrict the capability of our model in unseen data setups.

B.3 Metrics

We report metrics as defined in NJF [Aigerman et al. 2022]. Metrics are computed for each mesh after unit-sphere normalization. The vertex error is defined as:

$$\mathbf{L2-V} = \frac{1}{V} \sum_{i=1}^V \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|_2 \times 10^2, \quad (7)$$

where V is the number of vertices, $\hat{\mathbf{v}}_i$ is the i -th vertex obtained after solving the Poisson equation, and $\mathbf{v}_i$ is the i -th ground-truth

vertex. The Jacobian error is calculated as:

$$\mathbf{L2-J} = \frac{1}{F} \sum_{f=1}^F \|\hat{\mathbf{J}}_f - \mathbf{J}_f\|_2 \times 10, \quad (8)$$

where F is the number of faces, $\hat{\mathbf{J}}_f$ is the predicted and Jacobian for the f -th face, and $\mathbf{J}_f$ is the ground-truth Jacobian for the f -th face. The normal error, reported in degrees, is given by:

$$\mathbf{L2-N} = \frac{1}{F} \sum_{f=1}^F \frac{180}{\pi} \arccos(\hat{\mathbf{n}}_f \cdot \mathbf{n}_f), \quad (9)$$

where $\hat{\mathbf{n}}_f$ and $\mathbf{n}_f$ are the predicted and ground-truth unit normals for the f -th face, respectively, and F is the number of faces. where $\hat{\mathbf{n}}_f$ is the normal of face f after the Poisson solve, and $\mathbf{n}_f$ the ground-truth normal.

C Visual effect of feature processing

Fig. 1 visualizes our tangents and demonstrates the impact of smoothing the features $\mathbf{x}_i$ using PoissonNet. The tangents computed from processed features form a smoother vector field across the surface.

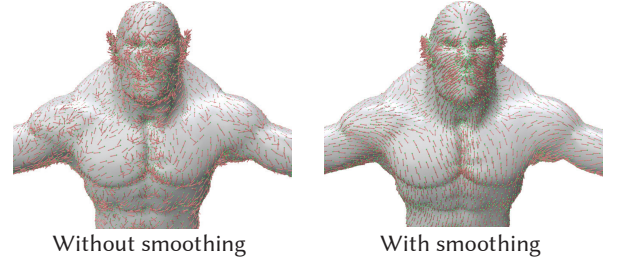


Fig. 1. A tangent field computed from semantic features without (left) and with (right) smoothing using a trained PoissonNet.

References

- Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Trans. Graph.* 41, 4 (2022), 1–17. doi:10.1145/3528223.3530141
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
- Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. 2024. Diffusion 3d features (Diff3F): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4494–4504. doi:10.1109/CVPR52733.2024.00430
- Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics* 34, 6 (2015), 248:1–248:16. doi:10.1145/2816795.2818013
- Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 5442–5451. doi:10.1109/ICCV.2019.00554
- Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. 2025. Surface-Aware Distilled 3D Semantic Features. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–12. doi:10.1145/3757377.3763974
- Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. 2024. Back to 3D: Few-Shot 3D Keypoint Detection with Back-Projected 2D Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4154–4164. doi:10.1109/CVPR52733.2024.00398
- Jiacong Xu, Yi Zhang, Jiawei Peng, Wufei Ma, Artur Jesslen, Pengliang Ji, Qixin Hu, Jiehua Zhang, Qihao Liu, Jiahao Wang, et al. 2023. Animal3D: A Comprehensive Dataset of 3D Animal Pose and Shape. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9099–9109. doi:10.1109/ICCV51070.2023.00835
- Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 2017. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5524–5532. doi:10.1109/CVPR.2017.586