# Degrees of Freedom Matter: Inferring Dynamics from Point Trajectories

Yan Zhang[†2], Sergey Prokudin[1,3], Marko Mihajlovic[1], Qianli Ma[†4], Siyu Tang[1]
[1]ETH Zürich, Switzerland , [2]Meshcapade
[3]ROCS, University Hospital Balgrist, University of Zürich, [4]Nvidia

## Abstract

*Understanding the dynamics of generic 3D scenes is fundamentally challenging in computer vision, essential in enhancing applications related to scene reconstruction, motion tracking, and avatar creation. In this work, we address the task as the problem of inferring dense, long-range motion of 3D points. By observing a set of point trajectories, we aim to learn an implicit motion field parameterized by a neural network to predict the movement of novel points within the same domain, without relying on any data-driven or scene-specific priors. To achieve this, our approach builds upon the recently introduced dynamic point field model [48] that learns smooth deformation fields between the canonical frame and individual observation frames. However, temporal consistency between consecutive frames is neglected, and the number of required parameters increases linearly with the sequence length due to per-frame modeling. To address these shortcomings, we exploit the intrinsic regularization provided by SIREN [53], and modify the input layer to produce a spatiotemporally smooth motion field. Additionally, we analyze the motion field Jacobian matrix, and discover that the motion degrees of freedom (DOFs) in an infinitesimal area around a point and the network hidden variables have different behaviors to affect the model's representational power. This enables us to improve the model representation capability while retaining the model compactness. Furthermore, to reduce the risk of overfitting, we introduce a regularization term based on the assumption of piece-wise motion smoothness. Our experiments assess the model's performance in predicting unseen point trajectories and its application in temporal mesh alignment with guidance. The results demonstrate its superiority and effectiveness. The code and data for the project are publicly available*[1].

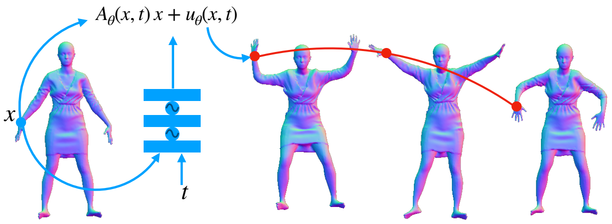

Figure 1. We introduce *DOMA*, a compact implicit motion model designed to capture generic dynamics of 3D scenes. By processing a 3D point $\boldsymbol{x}$ in the canonical frame alongside a 1D time step $t$, DOMA predicts an affine mapping, parameterized by a linear map $\boldsymbol{A}_\theta$ and a translation vector $\boldsymbol{u}_\theta$. By leveraging the inherent regularity of the utilized SIREN framework [54], DOMA ensures the generation of a spatiotemporally smooth motion field. The model's capacity to represent complex dynamics can be controlled by adjusting the degrees of freedom of the output affine mapping.

## 1. Introduction

Motion estimation plays a crucial role in several key areas of computer vision, such as dynamic scene reconstruction, autonomous navigation, and avatar creation. Treated as a distinct task, it emerges in various contexts as non-rigid tracking [11], point set [39, 57] and mesh registration [3, 17], shape matching [42], as well as optical and scene flow estimation [68]. The solutions adopted in these contexts significantly differ based on the ultimate objective and the foundational assumptions about the scene. A substantial amount of research exists concentrating on human-centric [4, 47] and rigid object motion [2, 63], alongside efforts in learning generic 2D motion priors in a data-driven manner [12, 23]. The diversity in applications and approaches underscores the complexity and significance of motion estimation within the field of computer vision [34].

In this work, we aim to develop a motion model capable of reconstructing the dynamics of generic 3D scenes without relying on data-driven motion priors or object-specific models. Specifically, by analyzing observed point trajectories within dynamic 3D scenes, we seek to learn an implicit motion model capable of predicting the movement of

[1] https://yz-cnsdqz.github.io/eigenmotion/DOMA/
† This work was done when YZ and QM were at ETH Zürich.

novel 3D points. This task bears significant relevance to the process of warping 3D points across frames, a procedure commonly encountered in neural rendering [29, 44], point cloud alignment [26], object tracking [59], and avatar creation [43, 50, 61]. Typically, warping methods developed in these domains are intended to supplement the primary objectives such as the quality of novel view synthesis. Consequently, critical features such as the representational capability of the motion model, along with the consistency and plausibility of the motion it recovers, have not been the primary concern in these studies.

In this context, the closest work is the recently proposed dynamic point field (DPF) model [48], which addresses the task of recovering the correct implicit deformation function based on the observed pair of 3D surfaces. It proposes a lightweight deformation field formulated by a SIREN network [53], an MLP with periodic activation functions. Due to the regularity introduced by SIREN, its modeled deformation is spatially smooth, enabling various applications such as robust mesh registration and avatar animation.

Despite its effectiveness, the DPF method is limited to learning deformation fields between just two frames, the canonical and the target frames. To extend this to multiple frames, it proposes creating a set of deformation field models, where each model transforms points from the canonical frame to a distinct target frame. Consequently, the number of models increases with the sequence length, leading to significant memory and computational overhead. Moreover, the frame-wise approach fails to ensure temporal motion consistency, potentially resulting in discontinuities between consecutive frames and jittering artifacts. We conduct a thorough analysis of the representational capabilities of DPF and its underlying SIREN network to address these shortcomings. To achieve temporal smoothness, we follow the wave equation formulation in [53], and modify the input layer by incorporating a 1D time step along with the existing 3D point location input, similar to the strategy regularly implied in the implicit warping fields for neural rendering [49].

Compared to the per-frame modeling scheme in DPF, this operation attempts to compress the entire sequence into a single network, which raises challenges on the model's capacity of motion representation. Rather than increasing the number of hidden variables in the SIREN network, we opt to refine its output layer by introducing more motion degrees of freedom (DOFs). From a mathematical standpoint of continuum mechanics [55], the advantages of additional DOFs are demonstrated in the Jacobian matrix of the motion field: Two points that are infinitely close to each other in space gain greater movement flexibility, provided the same number of network hidden variables. Therefore, the model representation power is improved, and the model compactness is retained.

Nevertheless, additional DOFs can increase the risk of overfitting, in particular when the observed point trajectories are excessively sparse. To overcome this issue, we leverage a generic assumption on motion, *i.e.* piece-wise smoothness, and propose a motion smoothness term by penalizing the approximate L1 norm of spatial derivatives of the predicted transformations. Here, rather than employing an auto-differentiation framework, we derive analytical gradients of our employed SIREN network to speed up the computation.

We undertake comprehensive experiments to validate the efficacy of our method. To assess its motion representation capabilities, we extract seven challenging sequences from the DeformingThings4D dataset [27] and generate four synthetic sequences that exhibit basic 3D motions. Our method demonstrates consistently superior performance in predicting the motion of novel points, when compared to both state-of-the-art methods and their variations. Furthermore, we employ our technique in the task of temporal mesh alignment with guidance, and evaluate its performance on complex sequences from the Resynth dataset [31, 32]. Compared to the DPF baseline, our approach achieves comparable alignment accuracy, better temporal regularity, and significantly smaller models, occupying approximately 200KB versus 8MB in the saved checkpoint for a 30-frame sequence.

We refer to our approach as *DOMA*, an acronym for **D**egrees **O**f freedo**M** m**A**tter, contending that additional degrees-of-freedom is essential to improve the expressivity of implicit motion models. Technical contributions are summarized as follows:

- We extend the state-of-the-art implicit model for surface deformation [48] for continuous, multi-frame motion modeling, leading to an implicit, spatiotemporally smooth affinity field;
- We leverage the Jacobian matrix to analyze the motion field complexity, and discover that additional DOFs at the output layer improve the model representation capability while retaining the model compactness;
- To enhance the quality of the motion learned, we introduce a regularization term based on the piece-wise smoothness assumptions of the domain;
- We assess our model, demonstrating the benefits of various modeling decisions through experiments, on challenging long-term scene flow estimation and guided mesh alignment.

## 2. Related Work

**Motion representation with object models.** Given the motion of a set of points, it is a highly unconstrained problem to infer the motion of other points in their proximity. In many applications, such an inverse problem is solved based on an object model that performs as a strong prior of the dynamics. Typical examples are marker-based human motion estimation [30, 33, 70], or 3D pose estimation from imagery data [21, 46, 51, 69], in which human parametric body models such as [46, 65] are leveraged. Here, the bones of a body model serve as an intermediate proxy for all other points' motion: the trajectory of a point on the body surface is generated by the weighted average of the bone transformations, a technique referred to as linear blend skinning (LBS). When extending the skinning weights to a vector field, as in [10, 35, 36, 52, 60, 61, 64], any point in the space can be animated by the bone transformations. The same technology can be applied to animals [74], babies [18], humanoids [67], and other categories.

In cases where the object model is not directly available, it can be jointly optimized together with the motion from observations. This provides flexibility on the object categories to handle more generic dynamics. For example, BANMO [66] proposes a generic deformable model, where a set of 3D Gaussians serve as the motion control proxy, analogous to bones. During optimization, the Gaussian locations and orientations are optimized together with their transformations. Likewise, KeyTr [41] proposes a bone basis to deform a point cloud across frames, in which the basis coefficients play a similar role as the skinning weights.

**Motion representation without object models.** Another line of work models the motion of points without the reliance of intermediate proxies like bones. Instead, they represent the motion of all points in space as a dense field, in which each location stores a transformation matrix. The motion of a point will be determined by the transformations of its infinitesimal neighbourhood. Methods under this paradigm are frequently employed in neural rendering [29, 44, 56], dense tracking [59], surface reconstruction [40, 43] and non-rigid geometry alignment [26, 48]. Niemeyer *et al*. [40] employ neural ODE [9] to model the dynamics, and estimate the implicit occupancy function at the canonical frame and its evolution as time progresses. Prokudin *et al*. [48], Pumarola *et al*. [49], and Palafox *et al*. [43] leverage MLP to model a translation field (or scene flow field), and warp the point from the canonical frame to the target frame via addition. Li *et al*. [25] employ a neural network to parameterize the flow field for regularization. The exploited network is a MLP with ReLU [24] activation functions. Park *et al*. [44] design a SE(3) transformation field, warping the points on the camera ray from the observation frame to the canonical frame, so as to train the neural radiance field [38] reliably. Lombardi *et al*. [29] employs a mixture of scaled SE(3) warping fields for the purpose of neural rendering dynamic scenes. Likewise, Li and Harada [26] apply SE(3) or scaled SE(3) transformations to perform non-rigid point cloud alignment. Compared to the SE(3) transformation, the scaled-SE(3) transformation is capable of representing the dilation or shrinking of an object. Going beyond points' locations, the spatial transformations can also be applied to features in neural networks and potentially improve the performance on *e.g*. image classification [20].

**Relations to object shape and view recovery from images.** Existing works such as [15, 16, 22] study to learn neural models from an image collection, and recover the 3D shape in a canonical frame, the camera pose, and the texture of an object from a single image. Despite addressing different tasks, their solutions of composing the instance-level shape by the mean shape and deformation is relevant to our manner of motion modeling. Furthermore, we are encouraged by these works to reconstruct dynamic scenes from multiview videos as future work.

**DOMA in context.** Existing motion modeling approaches are developed together with individual applications, in which the network architectures, coordinate encodings, and other properties are diverse. The motion representation capabilities of their models are seldom investigated. In contrast, we start with the basic assumption that the motion field has spatiotemporal regularity. Therefore, we leverage the SIREN [54] network, and extend the start-of-the-art work DPF [48] to a multi-frame, smooth affinity field model. We leverage knowledge of continuum mechanics, exploit the Jacobian matrix to describe the motion field complexity, and find that DOFs at the output layer and the network hidden variables affect the model representation power in different manners. Guided by these insights, we propose a solution to increase the model capacity while retaining the model lightweight. Moreover, we introduce a smoothness regularization term to overcome overfitting, which does not assume the underlying motion is *e.g*. rigid like in [44]. The effectiveness of DOMA is demonstrated with experiments in Sec. 4.

# 3. Method

## 3.1. Preliminaries

### 3.1.1 SIREN [53]

SIREN proposes an implicit neural representation, which is a multilayer perceptron (MLP) with periodic activation functions. Specifically, the MLP with $n+1$ layers is given by

$$\boldsymbol{y} = \boldsymbol{W}_n \left(\phi_{n-1} \circ \phi_{n-2} \circ \cdots \circ \phi_0\right)(\boldsymbol{x}) + \boldsymbol{b}_n, \qquad (1)$$

with $\phi_i = \sin(\boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{b}_i)$ and $i = \{0, 1, ..., n-1\}$. The gradient of the model w.r.t. the input is another phase-shifted SIREN network, and hence is infinitely differentiable. As reported in [53], the sinusoidal activation functions boost the model performance on the convergence speed, reconstruction quality, and smoothness, letting the

MLP outperform baselines consistently and considerably. However, this network requires special initialization to be trainable. Given $\boldsymbol{x} \in \mathbb{R}^d$, it is suggested to have the weights $w_i$ in the uniform distribution $\mathcal{U}(-\sqrt{6/d}, \sqrt{6/d})$, so that the model output will converge to a normal distribution.

### 3.1.2 Dynamic Point Field (DPF) [48]

DPF proposes an implicit deformation field to model point dynamics, achieving state-of-the-art performance on surface reconstruction, geometry deformation, and avatar animation with challenging clothing. Given a point $\boldsymbol{x} \in \mathbb{R}^3$ in the canonical frame, it learns a field $\boldsymbol{u} : \mathbb{R}^3 \to \mathbb{R}^3$ formulated by a SIREN network [54], and then transforms the point to a new location $\boldsymbol{y}$, *i.e.*

$$\boldsymbol{y} = g(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}). \tag{2}$$

To handle complex and rapid motion, an *as-isometric-as-possible* (AIAP) loss term [48, Eq.13] is proposed to minimize changes of pair-wise distances between neighbor points during deformation. Furthermore, it proposes guided geometry deformation via corresponding keypoints, which can avoid sub-optimal matching caused by the Chamfer distance [14]. Provided a set of keypoint pairs $\{(\boldsymbol{v}_i^c, \boldsymbol{v}_i^t)\}_{i=1}^N$, and a pair of non-corresponding geometries (*e.g.* meshes and point clouds) to align $(\boldsymbol{M}_c, \boldsymbol{M}_t)$, the guided geometry deformation can be performed by minimizing

$$\alpha_1 \mathcal{L}_{CD}\left(\boldsymbol{M}_t, \boldsymbol{g}(\boldsymbol{x})\right) + \alpha_2 \mathcal{L}_V(\boldsymbol{v}^c, \boldsymbol{v}^t) + \alpha_3 \mathcal{L}_{AIAP}\left(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{x}\right), \tag{3}$$

in which $\boldsymbol{x} \in \boldsymbol{M}_c$, $\alpha$s denote the loss weights, and $\mathcal{L}_{CD}$, $\mathcal{L}_V$, and $\mathcal{L}_{AIAP}$ denote the Chamfer loss, the L1 loss, and the AIAP loss on the corresponding keypoints, respectively. To align a sequence of geometries, DPF [48] suggests learning a set of deformation fields that warp points in the canonical frame to individual target frames.

## 3.2. DOMA: Spatiotemporal Affinity Motion Fields

DOMA is an implicit motion field formulated by

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}, t)\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t), \tag{4}$$

in which $\boldsymbol{x} \in \mathbb{R}^3$ is a point in the canonical frame, $t \in \mathbb{R}$ is the time step, $\boldsymbol{A} : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^{3\times3}$ and $\boldsymbol{u} : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^3$ are formulated by a shared SIREN network. Following Sitzmann et al. [53, Sec.5.4 of supp. mat.] that how the wave equation is formulated and solved, we incorporate the 1D time into SIREN as input, letting $\frac{\partial \boldsymbol{y}}{\partial t}$ be another phase-shifted SIREN and get regularized.

### 3.2.1 On The Representation Power

Different from the per-frame modeling mechanism of DPF [48], incorporating the 1D time domain into the input layer compresses the entire sequence into a single network, raising challenges on the model representation power.

Referring to [55], the DPF formula Eq. (2) is generic to model object deformation in continuum mechanics. However, it has limitations in empirical studies, motivating us to investigate the reasons. Rather than studying the entire domain, we look into an infinitesimal region around an arbitrary 3D point $\boldsymbol{x}$ in the canonical frame, and derive its Jacobian matrix as

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \boldsymbol{I} + \nabla \boldsymbol{u}(\boldsymbol{x}), \tag{5}$$

which is the optimal linear approximation of the motion around $\boldsymbol{x}$ and $\nabla$ denotes the spatial gradient.

This Jacobian matrix is able to reflect the motion complexity. Intuitively, $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ indicates the difference of movements between two points that are infinitely close to each other. Without any constraints on $\nabla \boldsymbol{u}$, the model is capable of representing highly complex motion. However, $\boldsymbol{u}$ is formulated by SIREN [53], letting $\nabla \boldsymbol{u}$ become to

$$\nabla \boldsymbol{u} = \boldsymbol{W}_n \left( \prod_{i=0}^{n-1} \boldsymbol{W}_i \circ \varphi_i(\boldsymbol{x}) \right), \tag{6}$$

with $\varphi_i = \cos(\boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{b}_i)$. Due to $|\varphi_i| \leq 1$, we can derive (see Sec. 6 in supp. mat.)

$$\|\nabla \boldsymbol{u}\|_2 \leq d^n \cdot \prod_{i=0}^{n} \|\boldsymbol{W}_i\|_2, \tag{7}$$

in which $\|\cdot\|_2$ is the L2 norm, *i.e.* **the largest singular value**, of the matrix. Consequently, the movement difference between two neighboring points in the domain is constrained by Eq. (5) and Eq. (7). To increase the representation power, one can straightforwardly increase the number of hidden layers, or the dimension of hidden variables, because both can increase the upper-bound of $\|\nabla \boldsymbol{u}\|_2$.

Our DOMA model can improve the model capacity without modifying the hidden layers. Referring to Eq. (4), its Jacobian matrix is given by

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \boldsymbol{A} + \langle \nabla \boldsymbol{A}, \boldsymbol{x} \rangle + \nabla \boldsymbol{u}(\boldsymbol{x}), \tag{8}$$

which replaces the identity matrix in Eq. (5) with two complex terms. Since the identity matrix in Eq. (5) does not contribute to the motion complexity, our method intrinsically increases the complexity, while keeping the model hidden layers unchanged. Consequently, more complex linear transformations with additional DOFs, such as scaling and shearing, are introduced to every infinitesimal area in the entire domain, thereby increasing the motion complexity globally.

### 3.2.2 The Variants of DOMA

The motion complexity can be controlled by applying different constraints on $\boldsymbol{A}$, leading to different versions according to the DOFs, inspired by existing works *e.g.* [26, 29, 44]. We denote the model Eq. (4) as **DOMA-Affinity**. In our implementation, the SIREN network outputs a 12-dimensional variable. The first 9 variables are reshaped to $\boldsymbol{A}$ and the rests are regarded as $\boldsymbol{u}$.

**DOMA-Trans.** When $\boldsymbol{A}$ is an identity matrix, Eq. (4) degenerates to a translation field, which formulated by

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t). \tag{9}$$

This model can be regarded as a straightforward extension of DPF [48] to the spatiotemporal domain.

**DOMA-SE(3).** With $\boldsymbol{A} = \boldsymbol{Q} \in SO(3)$, *i.e.* a rotation matrix in the 3D space, Eq. (4) becomes to

$$\boldsymbol{y} = \boldsymbol{Q}(\boldsymbol{x}, t)\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t). \tag{10}$$

Besides producing the translation, we let the SIREN network output the 6D continuous rotation representation [72], and perform orthogonalization to get the rotation matrix.

**DOMA-Scaled SE(3).** By introducing an additional DOF for scaling, we can modify Eq. (4) to

$$\boldsymbol{y} = s(\boldsymbol{x}, t)\boldsymbol{Q}(\boldsymbol{x}, t)\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t), \tag{11}$$

with $s(\boldsymbol{x}, t)$ being the spatiotemporal scalar field. In our implementation, we let the SIREN network to produce an additional 1D variable, and apply the softplus activation function [13, 71] to product $s(\boldsymbol{x}, t)$.

### 3.2.3 Model Complexity Analysis

By changing DOFs at the SIREN output layer, the hidden layers remain unchanged, and the model size is not linearly growing with the sequence length. Provided the motion sequence has $T$ frames, DOMA-Affinity has $16d + nd^2$ parameters, with $n$ and $d$ denoting the number of hidden layers and the hidden dimensions, respectively. In contrast, the per-frame modeling of DPF [48] requires $T - 1$ SIRENs and has $(6d + nd^2)(T - 1)$ parameters in total, leading to $\mathcal{O}(T)$ model size. Please see Tab. A1 in supp. mat. for more details.

## 3.3. Motion Smoothness Regularization

Although introducing extra DOFs can improve the model representation power, it increases the risk of overfitting. Without loss of generality, we assume the motion field is piece-wise smooth in the domain. Referring to variational methods in optical flow *e.g.* [5, 73], we introduce the following smoothness regularization loss, *i.e.*

$$\mathcal{L}_H = \mathbb{E}_{t,\boldsymbol{x}} \left[ \Psi\left( \|\nabla \boldsymbol{A}\|_F^2 + \|\nabla \boldsymbol{u}\|_F^2 \right) \right], \tag{12}$$

in which $\Psi(s^2) = \sqrt{1 + s^2} - 1$ is the convex Charbonnier function [8] to approximate the L1 norm, $\nabla(\cdot)$ denotes computing the spatial gradients, and $\| \cdot \|_F$ denotes the Frobenius norm. Intuitively, the local motion is parameterized by $\boldsymbol{A}$ and $\boldsymbol{u}$, and hence the zero value of the above loss term suggests all points conduct the identical affine transformation. The Charbonnier function plays the role of a robustifier, with which the difference of motions between two neighboring points is less penalized compared to L2 norm. With prior knowledge on the scene dynamics, $\Psi(s^2)$ can be changed to other terms, *e.g.* $\Psi(s^2) = s^2$ to encourage homogeneous motion.

Instead of employing auto-differentiation tools in *e.g.* PyTorch [45], we implement analytical gradients of the SIREN network, *i.e.* Eq. (5), to speed up computation. See Tab. A5 in supp. mat. for an empirical study.

# 4. Experiment

Without explicit mentioning, we set the first frame in the sequence as the canonical frame, and normalize the time steps to $[-1, 1]$. Please see supp. mat. for more details and additional experiments.

## 4.1. Novel Point Motion Prediction

Based on a sparse set of observed point trajectories, we aim to predict the motion of unseen points during training, in order to verify the quality of learned dynamics.

**Datasets.** We select 7 sequences with various object categories, shapes, and motions from the DeformingThings4D [27] dataset. For each sequence, we use 100 consecutive frames. We randomly select 25% mesh vertices for training the motion field, and use the remaining ones for testing. In addition, we create four synthetic sequences of elemental motions, *i.e.* translation, rotation, scaling, and shearing, respectively, in order to investigate the model representation power in detail. Each synthetic sequence has 20 frames and contains 3000 points uniformly sampled from $[-1, 1]^3$. Likewise, we randomly select 25% of points for training, and leave the remaining ones for testing.

**Evaluation metrics.** For evaluation, we employ the learned motion field to transform testing points in the canonical frame to individual target frames. We compute the scene flow end point error (EPE), *i.e.* $\mathbb{E}_{t \in \{1..., T\}}[\|\boldsymbol{v}_t - \boldsymbol{v}_t^{gt}\|_1]$ and $\boldsymbol{v}_t = \boldsymbol{y}_t - \boldsymbol{x}$, in which $\boldsymbol{y}_t$ denotes the estimated corresponding point of $\boldsymbol{x}$ at time $t$. For DeformingThings4D, we additionally use the learned motion field to warp the canonical object mesh to each individual frame, sample $10^6$ points on both the warped mesh and the ground truth mesh, and compute the Chamfer distance [14] $\mathcal{L}_{CD}$ and the Chamfer normal distances $\mathcal{L}_n$, as in [48, Table 2].

**Baselines and ours.** This task is highly related to learning the warping fields in various scenarios, such as deformable

| *Methods* | *EPE*↓ | $\mathcal{L}_{CD}\downarrow$ | $\mathcal{L}_n\downarrow$ |
|---|---|---|---|
| MLP-ReLU [38] | 222.4 | 3.410 | 0.411 |
| MLP-ReLU PE.6 [38] | 237.7 | 3.747 | 0.431 |
| DCT-NeRF [58] | 215.0 | 3.766 | 0.347 |
| BANMO [66] | 488.9 | 13.275 | 0.451 |
| BoneCloud [41, 66] | 136.1 | 1.993 | 0.261 |
| Ours-Trans | 78.5 | 1.401 | **0.215** |
| Ours-SE(3) | 76.7 | 3.706 | 0.225 |
| Ours-Scaled SE(3) | **76.2** | 2.074 | 0.220 |
| Ours-Affinity | 78.1 | **1.266** | 0.218 |

Table 1. Results on DeformingThings4D sequences. EPE and $\mathcal{L}_{CD}$ are in $\times 10^{-4}$. Best results are in boldface.

object modeling [43], scene flow estimation [25], and neural rendering [44, 49]. The warping field is commonly parameterized with a neural network with ReLU activation functions [24] and positional encodings [38]. Therefore, we leverage such kinds of neural networks as baselines. Specifically, we denote *MLP-ReLU* as an MLP with ReLU activation functions and 6 hidden layers of 128 dimensions. It takes the concatenation of the 3D location and the 1D time step as input and outputs motion vector. Additionally, we introduce positional encoding [38], or output DCT coefficients [58], to create *MLP-ReLU PE.6* and *DCT-NeRF* as baselines, respectively. Moreover, we adapt the *BANMO* [66] deformation module into our setting, and implement a modified version named *BoneCloud*, following the idea of [41]. More details of these baseline methods are demonstrated in Sec. 8.1 of supp. mat.

We denote the DOMA models with their suffixes, *i.e.* *-Trans*, *-SE(3)*, *-Scaled SE(3)*, and *-Affinity*, respectively. All their SIREN networks have 128 hidden dimensions and 2 hidden layers. Moreover, we implement the elastic regularization proposed in [44] to encourage rigid motion, which is denoted by *-E*. Likewise, *-H* denotes our motion smoothness regularization *without* the Charbonnier function, encouraging the motion is homogeneous.

**Results.** The results on the DeformingThings4D sequences are presented in Tab. 1. We can see that the SIREN-based methods achieve comparable performance, and outperform other baseline methods consistently.

The results on the synthetic sequences are shown in Tab. A3 and Fig. 2. First, we can see that the incorporated DOFs lead to different motion representation behaviors. Methods with SE(3) transformations are more effective in representing rotation and translation. By comparing '-Scaled SE(3)' and '-SE(3)', we can see the additional DOF benefits modeling scaling. The affine transformation is effective in representing all cases. Second, we can see appropriate regularization is important. The elasticity regularization proposed in [44] is effective to encourage rigid motions, but performs inferior if the motion is non-rigid, *i.e.* scaling and shearing. On the other hand, our smoothness regularization significantly boosts the performance, if the sequence fits the modelled DOFs.

| *Methods* | Rotation | Scaling | Shearing | Translation |
|---|---|---|---|---|
| -Trans | 2725.4 | 1817.8 | 1619.5 | 1042.4 |
| -SE(3) | 730.6 | 1991.4 | 1138.3 | 899.4 |
| -Scaled SE(3) | 801.1 | 685.8 | 1524.7 | 1096.2 |
| -Affinity | 1486.0 | 915.4 | 622.1 | 822.4 |
| -Trans-E | 38.0 | 1669.6 | 753.6 | 38.8 |
| -SE(3)-E | 20.0 | 1761.3 | 832.7 | 26.4 |
| -scaled SE(3)-E | 21.2 | 1161.8 | 961.1 | 24.0 |
| -Affinity-E | 19.2 | 155.7 | 864.0 | 15.7 |
| -Trans-H | 4919.9 | 2056.4 | 2446.8 | 37.8 |
| -SE(3)-H | 52.4 | 2012.4 | 1665.0 | 36.9 |
| -scaled SE(3)-H | 29.3 | 22.1 | 688.0 | 30.3 |
| -Affinity-H | **5.4** | **26.3** | **8.5** | **28.8** |

Table 2. Results on Synthetic sequences w.r.t. EPE (in $\times 10^{-4}$). '-E' denotes the elasticity loss proposed in Nerfies [44], and '-H' denotes our smoothness loss.

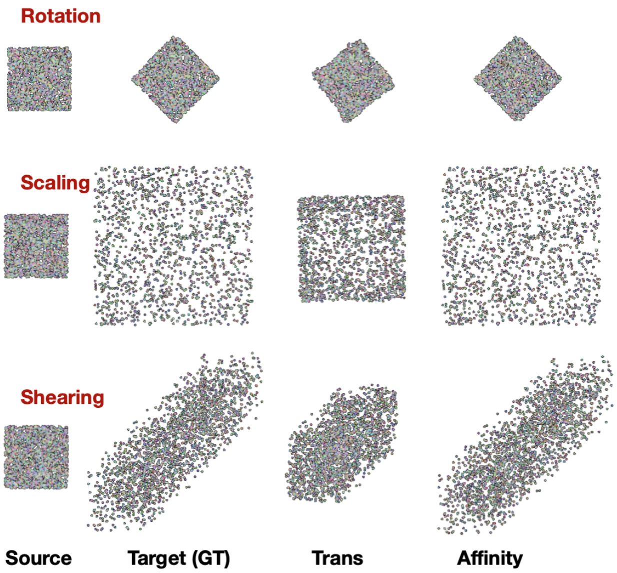

Figure 2. Qualitative results on the Synthetic sequences. The smoothness regularization is applied. Rows show types of motions, and columns show the testing points in the canonical frame, a target frame, and estimated results, respectively. See more descriptions in Sec. 4.1 and Sec. 8.1.3 in supp. mat.

Moreover, we find the structure of the linear matrix $\boldsymbol{A}$ and the network hidden dimension have different behaviors to affect the model representation power. Increasing the hidden dimension cannot simply introduce new DOFs that the model can represent. see Sec. 8.1.3 in supp. mat. for details.

## 4.2. Guided Mesh Alignment

Aligning a mesh template to a sequence of scans is important in various applications, such as graphics [31] and healthcare [18]. We follow the guided geometry deformation method of DPF [48]. Referring to Eq. 3, we minimize

$$\sum_t \{\alpha_1 \mathcal{L}_{CD}\left(\boldsymbol{M}_t, \boldsymbol{y}(\boldsymbol{x})\right) + \alpha_2 \mathcal{L}_V(\boldsymbol{v}^c, \boldsymbol{v}^t) + \alpha_3 \mathcal{L}_{AIAP}\left(\boldsymbol{y}(\boldsymbol{x}), \boldsymbol{x}\right) + \alpha_4 \mathcal{L}_H\}. \tag{13}$$

In this experiment, we set $(\alpha_1, \alpha_2) = (10^3, 1)$. The regularization loss weights are $(\alpha_3, \alpha_4) = (1, 0.001)$ if they are enabled. Training terminates after 2000 iterations for all cases.

**Datasets.** We employ the Resynth dataset [31, 32], in which the human bodies perform articulate motions, while the clothing, in particular the long skirt, moves accordingly in a highly complex manner. We use 4 sequences from 4 individual subjects with different genders, body shapes, and clothing types. Each sequence is downsampled by every 2 frames, and afterwards the first 30 frames are selected, leading to 16 sequences with 480 frames in total. We use the SMPL-X [46] mesh vertices (10,475 points per frame) as the guidance points, and the low-resolution scans (40,000 points per frame) as the targets to fit. Furthermore, we perform Poisson surface reconstruction on the low-res scan at the canonical frame, and obtain a mesh template with about 60K vertices and 130K faces. During training, we learn the motion field based on the guidance points and the low-res scans, and minimize Eq. (13). During testing, we warp the mesh template vertices to individual frames based on the learned motion field, and re-compute the vertex normals. Note these mesh template are unseen during training.

**Evaluation metrics.** For evaluation, we compute the Chamfer distance of the vertex locations and normals between the warped meshes with the target low-res scans, *i.e.* $\mathcal{L}_{CD}$ and $\mathcal{L}_n$, as in Tab. 1 as well as DPF [48, Table 2]. To verify the temporal smoothness, we additionally compute two metrics: 1) The standard deviation (std) of edge lengths along the temporal dimension, and report its maximal value. This metric is able to reflect whether the mesh is significantly stretched or not. 2) The averaged std of the velocity l2-norm of the mesh vertices, which measures the temporal smoothness. These two metrics are denoted as *STD(E)* and *STD(V)*, respectively. Their values are the lower the better, but should not vanish because of the conducted deformation and motion.

**Baselines and ours.** We compare the frame-wise DPF scheme that is suggested by [48]. In addition, we investigate the effectiveness of the regularization loss terms AIAP [48, Eq.13] and our proposed motion smoothness term. The method notations are the same with Sec. 4.1. All SIREN networks have 3 hidden layers and 128 hidden dimensions.

| | $\mathcal{L}_{CD}\downarrow$ | $\mathcal{L}_n\downarrow$ | *STD(E)*↓ | *STD(V)*↓ |
|---|---|---|---|---|
| DPF [48] | 1.149 | **0.122** | **11.6** | 24.6 |
| -Trans | 1.230 | 0.128 | 12.8 | 22.9 |
| -Affinity | **1.142** | 0.125 | 11.9 | **22.8** |
| DPF-A [48] | 1.166 | **0.119** | **10.3** | 24.2 |
| -Trans-A | 1.195 | 0.123 | 10.4 | **23.0** |
| -Affinity-A | **1.151** | 0.122 | 10.6 | **23.0** |
| DPF-H [48] | 1.142 | **0.123** | 10.3 | 24.2 |
| -Trans-H | 1.207 | 0.128 | 10.8 | **22.9** |
| -Affinity-H | **1.127** | 0.127 | **10.1** | **22.9** |
| DPF-AH [48] | 1.189 | **0.120** | 9.3 | 24.3 |
| -Trans-AH | 1.240 | 0.124 | 9.3 | **23.0** |
| -Affinity-AH | **1.187** | 0.124 | **8.9** | **23.0** |

Table 3. Results of guided mesh alignment on our selected Resynth sequences. $\mathcal{L}_{CD}$ is in $\times 10^{-4}$. *STD(E)* and *STD(V)* are given in millimeters. '-A' and '-H' denote the AIAP regularization [48] and our smoothness regularization, respectively. '-AH' denotes both regularization terms are applied. Best results are in boldface. Please see Tab. A7 for the performance of all models.

| | #params. | checkpoint size (KB) |
|---|---|---|
| DPF [48] | 1497600 | 7800 |
| -Trans | 50048 | 139.6 |
| -SE(3) | 50816 | 209.2 |
| -Scaled SE(3) | 50944 | 209.8 |
| -Affinity | 51200 | 210.8 |

Table 4. Evaluations on the model size on the Resynth sequence. Since lightweight models are preferred, the numbers here are the lower the better.

**Results.** The results are shown in Tab. 3. Compared to frame-wise DPF, we can see DOMA-Trans leads to consistently worse alignment accuracy, but better temporal smoothness. This indicates the temporal regularity is obtained by compressing the entire motion into a single SIREN-based network, whereas the model's representational power is not sufficient. When replacing the translation field by an affinity field, *i.e.* DOMA-Affinity, the alignment accuracy is consistently improved to a similar level with frame-wise DPF, and the temporal smoothness is retained. This indicates that introduced extra DOFs can effectively improve the model representation power. In addition, the AIAP loss and the smoothness regularization can individually improve the performances, but their combination does not lead to obvious advantages, except for the edge length variations. Fig. 3 [2] illustrates some pairs of consecutive frames. We can see that the frame-wise DPF scheme causes visible discontinuities and artifacts in some regions,

[2] Quantitatively, *i.e.* ($\mathcal{L}_{CD}$, $\mathcal{L}_n$, STD-E, STD-V) as in Tab. 3, DPF gives $(3.962, 0.229, 16.0, 35.7)$ and $(2.557, 0.175, 13.8, 18.5)$ for the top and bottom rows, respectively, whereas DOMA-Affinity gives $(3.208, 0.216, 19.8, 29.3)$ and $(2.532, 0.169, 10.5, 15.6)$.

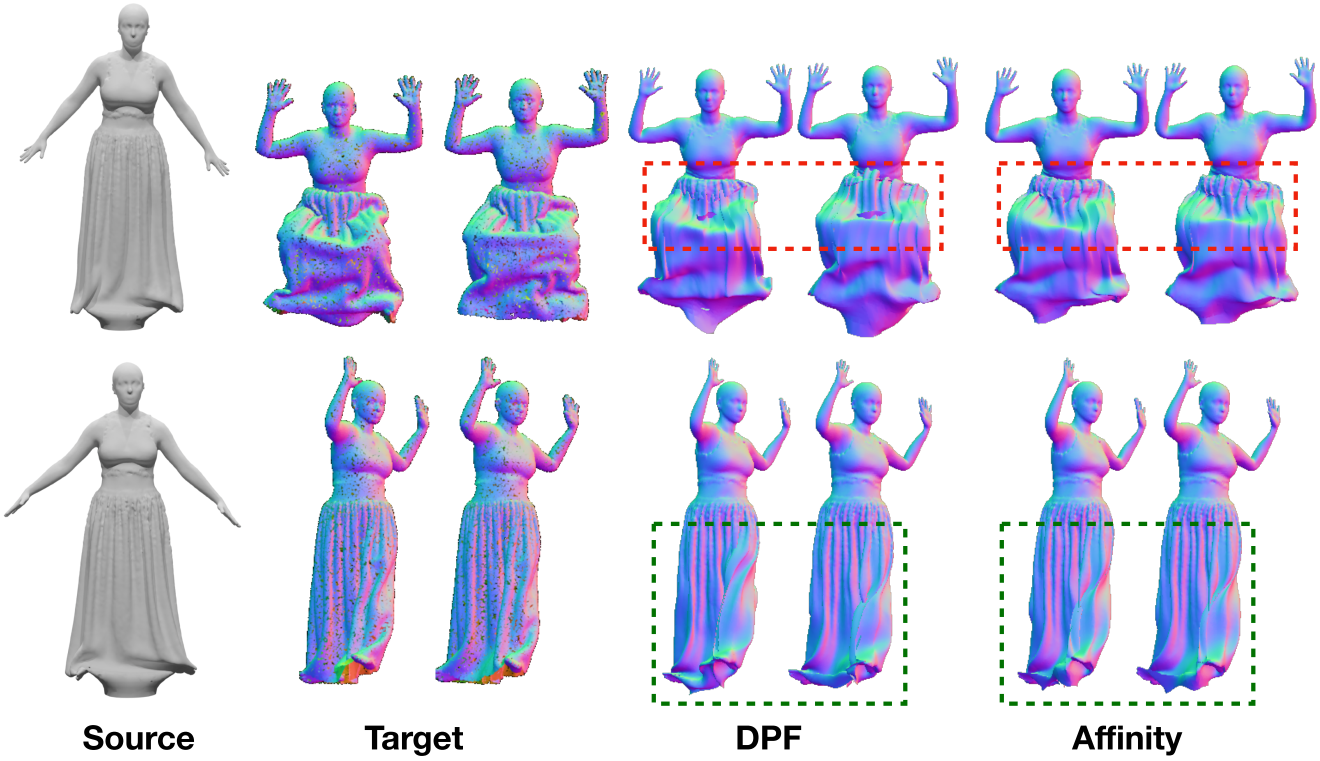

Figure 3. Illustration of results on two Resynth sequences. From left to right: The source mesh in the canonical frame, two consecutive frames of the target scans, the results from frame-wise DPF and DOMA-Affinity, respectively. Both AIAP [48] and smoothness regularization are applied. The bounding boxes highlight significant changes.

whereas the results of the affinity field have better temporal regularity.

Furthermore, the advantage of DOMA can be reflected by the model compactness. As shown in Tab. 4, DOMA models are significantly more lightweight. Adding additional DOFs at the network output layer only increases the number of parameters marginally.

## 5. Conclusion

In this work, we have advanced the DPF framework [48] into a continuous, multi-frame affinity field model, which inherently exhibits spatiotemporal regularity and improves representational capabilities without compromising compactness. Incorporating the 1D time domain to the network input layer ensures temporal regularity, and the DOFs at the output layer can manipulate the model representation power without modifying the network hidden variables. The experimental results on novel point motion prediction and guided mesh alignment show its effectiveness and superiority to baselines.

**Limitations and future works.** First, we have 4 loss terms to minimize in the task of guided mesh alignment, and inappropriate loss weights can degrade the performance considerably. How to balance their weights is still not transparent, which is worthy exploring in the future. Second, our method can be employed to model warping fields for dynamic scene reconstruction and rendering, which is not covered in this paper and will be studied as future work. Note that our method requires corresponding points between frames. Additional DOFs are effective to represent fine-grained movements, but might behave as a disadvantage to extract corresponding points due to less constraints. Third, our advanced model representation power is potential to model highly complex dynamics, *e.g.* fluid fields, which can benefit specific applications of medicine, aerodynamics, physics, *etc*. Furthermore, our model is deterministic and does not consider the motion uncertainty. Thus, a future direction is to develop a generative model on dynamics, which synthesizes diverse dynamics based on the same set of point trajectories.

## Acknowledgement

We sincerely thank Shaofei Wang for fruitful suggestions, discussions, and other helps. This project is partially supported by the SNSF grant 200021 204840.

## References

[1] Coding adventure: Simulating fluids. `https://github.com/SebLague/Fluid-Sim`. 18, 20

[2] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *CVPR*, 2019. 1

[3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *CVPR*, 2014. 1

[4] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, 2017. 1

[5] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 5

[6] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *ECCV*, 2020. 14

[7] Djalil Chafaı, Djalil Chafä, Olivier Guédon, Guillaume Lecue, and Alain Pajor. Singular values of random matrices. *Lecture Notes*, 2009. 13

[8] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994. 5

[9] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 2018. 3

[10] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, 2021. 3

[11] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, 2022. 1

[12] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1

[13] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *NeurIPS*, 2000. 5

[14] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 4, 5

[15] Zahra Gharaee, Felix Järemo Lawin, and Per-Erik Forssén. Self-supervised learning of object pose estimation using keypoint prediction. In *ICLR*, 2023. 3, 14

[16] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *ECCV*, 2020. 3, 14

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3D correspondences by deep deformation. In *ECCV*, 2018. 1

[18] Nikolas Hesse, Sergi Pujades, Michael J Black, Michael Arens, Ulrich G Hofmann, and A Sebastian Schroeder. Learning and tracking the 3d body shape of freely moving infants from rgb-d sequences. *PAMI*, 2019. 3, 7

[19] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. 2002. 12

[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *NeurIPS*, 2015. 3

[21] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 3

[22] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 3, 14

[23] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv:2307.07635*, 2023. 1

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 3, 6, 14

[25] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. In *NeurIPS*, 2021. 3, 6

[26] Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. *NeurIPS*, 2022. 2, 3, 5

[27] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *ICCV*, 2021. 2, 5, 14, 18

[28] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023. 14

[29] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019. 2, 3, 5

[30] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Trans. Gr.*, 2014. 3

[31] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J Black. Scale: Modeling clothed humans with a surface codec of articulated local elements. In *CVPR*, 2021. 2, 7, 16

[32] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J Black. The power of points for modeling humans in clothing. In *ICCV*, 2021. 2, 7, 16

[33] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 3

[34] Alexander Mathis, Steffen Schneider, Jessy Lauer, and Mackenzie Weygandt Mathis. A primer on motion capture with deep learning: principles, pitfalls, and perspectives. *Neuron*, 2020. 1

[35] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *CVPR*, 2021. 3

[36] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. COAP: Compositional articulated occupancy of people. In *CVPR*, 2022. 3

[37] Marko Mihajlovic, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Resfields: Residual neural fields for spatiotemporal signals. In *ICLR*, 2024. 14

[38] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 6, 14

[39] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *PAMI*, 2010. 1

[40] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019. 3

[41] David Novotny, Ignacio Rocco, Samarth Sinha, Alexandre Carlier, Gael Kerchenbaum, Roman Shapovalov, Nikita Smetanin, Natalia Neverova, Benjamin Graham, and Andrea Vedaldi. Keytr: keypoint transporter for 3d reconstruction of deformable objects in videos. In *CVPR*, 2022. 3, 6, 14

[42] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ToG*, 2012. 1

[43] Pablo Palafox, Aljaz Bozic, Justus Thies, Matthias Nießner, and Angela Dai. Neural parametric models for 3d deformable shapes. In *ICCV*, 2021. 2, 3, 6

[44] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 2, 3, 5, 6, 14, 15

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5, 15, 16

[46] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 3, 7, 16

[47] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer vision and image understanding*, 2007. 1

[48] Sergey Prokudin, Qianli Ma, Maxime Raafat, Julien Valentin, and Siyu Tang. Dynamic point fields. In *ICCV*, 2023. 1, 2, 3, 4, 5, 7, 8, 12, 14, 17, 18, 19

[49] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2, 3, 6

[50] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *CVPR*, 2024. 2

[51] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, 2021. 3

[52] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *CVPR*, 2021. 3

[53] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 1, 2, 3, 4, 12, 13

[54] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1, 3, 4, 13

[55] Anthony James Merrill Spencer. *Continuum mechanics*. 2004. 2, 4

[56] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, 2022. 3

[57] Yanghai Tsin and Takeo Kanade. A correlation-based approach to robust point set registration. In *ECCV*, 2004. 1

[58] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 6, 14

[59] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *ICCV*, 2023. 2, 3

[60] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. In *NeurIPS*, 2021. 3

[61] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *ECCV*, 2022. 2, 3

[62] Eric W. Weisstein. ”matrix norm.” from mathworld–a wolfram web resource. https://mathworld.wolfram.com/MatrixNorm.html. 12

[63] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Muller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *CVPR*, 2023. 1

[64] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022. 3

[65] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *CVPR*, 2020. 3

[66] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 3, 6, 14

[67] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *CVPR*, 2021. 3

[68] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical flow and scene flow estimation: A survey. *Pattern Recognition*, 2021. 1

[69] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *ICCV*, 2021. 3

[70] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *CVPR*, 2021. 3

[71] Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *IJCNN*, 2015. 5

[72] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 5, 13

[73] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Optic flow in harmony. *IJCV*, 2011. 5

[74] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017. 3