

# **CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism**

Jiahui Lei Kostas Daniilidis  
University of Pennsylvania

<https://www.cis.upenn.edu/~leijh/projects/cadex>



Figure 1. We model the deformable surface through a learned **canonical shape** (the middle shape) and factorize the deformation (orange arrow) by learnable continuous bijective **canonical maps** (green bidirectional arrows) that provide the cycle consistency and topology preservation. Visual results are from the test set on dynamic animals, human bodies and articulated objects.

## Abstract

While neural representations for static 3D shapes are widely studied, representations for deformable surfaces are limited to be template-dependent or to lack efficiency. We introduce Canonical Deformation Coordinate Space (CaDeX), a unified representation of both shape and non-rigid motion. Our key insight is the factorization of the deformation between frames by continuous bijective canonical maps (homeomorphisms) and their inverses that go through a learned canonical shape. Our novel deformation representation and its implementation are simple, efficient, and guarantee cycle consistency, topology preservation, and, if needed, volume conservation. Our modelling of the learned canonical shapes provides a flexible and stable space for shape prior learning. We demonstrate state-of-the-art performance in modelling a wide range of deformable geometries: human bodies, animal bodies, and articulated objects.

## 1. Introduction

Humans perceive, interact, and learn in a continuously changing real world. One of our key perceptual capabilities is the modeling of a dynamic 3D world. Such geometric intelligence requires sufficiently general neural representations that can model different **dynamic** geometries in 4D sequences to facilitate solving robotics [55], computer vision [29], and graphics [44] tasks. Unlike the widely studied 3D neural representations, a dynamic representation has to be able to associate (for example, finding corre-

spondence) and aggregate (for example, reconstruction and texturing) information across the deformation states of the world. Directly extending a successful static 3D representation (for example, [33]) to each deformed frame leads to low efficiency [36], and the inability to model the information flow across frames, which is critical when solving ill-posed problems as in [44]. Our desired dynamic representation needs to simultaneously represent a global surface (**canonical/reference shape**) across all frames and the **consistent deformation** (correspondence/flow/motion) between any frame pair (Fig. 1), so that we can recover the dynamic geometry by reconstructing only one reference surface and generating the rest of the deformed surfaces by using the consistent deformation representation as well as associate and aggregate information across frames (Fig. 2A).

The majority of dynamic representations that satisfy the above desired properties are model-based and rely on parametric models for specific categories like human bodies [1, 31] (Fig. 2B), faces [4, 30], or hands [47]. On the contrary, recent model-free methods like the implicit flow [36, 50] (Fig. 2C) apply one universal 4D representation but model the canonical shape in an ad hoc chosen frame [36, 50] that complicates the shape prior. Alternatively, the choice of an approximate mean/neutral shape [58] as the canonical shape can limit the shape expressibility. Modeling of the deformation is done by either MLPs [50, 58] that ignore the real world deformation properties, or by ODEs [36] that are inefficient for space deformation, or by an optimized embedded graph [6] or Atlas [2] that are sequence specific.

In this work, we introduce a novel and general archi-

ture and representation that enable a competitive reconstruction of every frame and the recovery of consistent correspondence across frames. Our approach is rooted in the factorization of deformation (Sec. 3.1). If we assume that the topology does not change during deformation, all deformed surfaces of one instance can be regarded as equivalent through continuous bijective mappings (homeomorphisms). This allows us to factorize the deformation between two frames by the composition of two continuous invertible functions such that one maps the source frame into a common 3D Canonical Deformation Coordinate Space (**CaDeX**) while another maps it back to the destination frame. Such a factorization and its implementation (Sec. 3.2) is novel, simple, and efficient (compared to ODEs [36]) while it guarantees cycle consistency, topology preservation, and, if necessary, volume conservation (Sec.3.3). The **canonical shape** embedded in the CaDeX can be regarded as the representative element, while the associated invertible mappings that transform between deformed frames and the CaDeX are the **canonical maps**. Therefore, we model the reference surface directly in the CaDeX via an implicit field [33] (Sec. 3.4), which can be optimized together with the canonical maps during training.

In summary, our main **contributions** are: (1) A novel general representation and architecture for dynamic surfaces that jointly solve the canonical shape and consistent deformation problems. (2) Learnable continuous bijective canonical maps and canonical shapes that jointly factorize the shape deformation, and are novel, simple, efficient, and guarantee cycle consistency and topology preservation. (3) A novel solution to the dynamic surface reconstruction and correspondence tasks given sparse point clouds or depth views based on the proposed representation. (4) We demonstrate state-of-the-art performance on modelling different deformable categories: Human bodies [5], Animals [57] and Articulated Objects [53].

## 2. Related Work

Proposed neural representations for static 3D geometry [9, 11, 17, 18, 20, 26, 33, 34, 37, 40, 43, 56] are promising, but most of them do not involve modeling of deformations. A few recent approaches represent or process 3D shapes via deformation [13, 22, 23, 25, 59], but they focus on static 3D shape collections that do not meet the requirements (e.g, efficiency) for processing 4D data. We will focus our related work on dynamic representations of deformable geometry.

**Model-Based Dynamic Representation:** Many successful 3D parametric models for specific shape categories have been introduced, for example, the morphable model [4] and FLAME [30] for faces, SCAPE [1] and SMPL [31] for human bodies, and MANO [47] for hands . These model-based representations (Fig. 2B) suffer from limited expressivity, which can be mitigated by neural networks. Net-

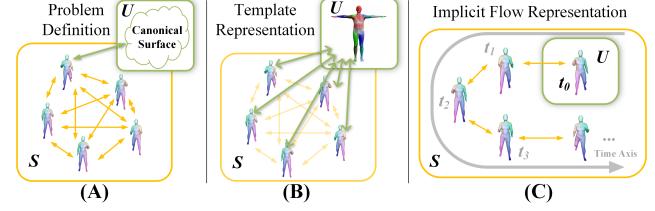


Figure 2. (A) **Problem definition:** A list, or a set of deformed surfaces  $\mathcal{S} = \{S_i\}$  of one instance should be represented by 1.) one canonical 3D surface  $U$  (in the green box) and 2.) the consistent deformation between surfaces (yellow arrows); (B) **Model-based methods:**  $S_i$  is obtained through the skinning function (green arrows) from the template mesh [31]; (C) **Implicit-flow methods:** the first frame serves as the reference shape and the deformation is modeled by Neural-ODEs [36] or MLPs [50].

works can express detailed template shapes based on template meshes [32, 38, 48, 52] or skeletons [12, 27, 51], and can learn more detailed skinning functions [8, 48, 52] or forward deformations [38]. However, they rely on the strong assumption of canonicalization through the pose, skeleton, and template mesh, which makes them limited to specific categories, and insufficient for modeling the rich dynamic 3D world. Our method does not rely on any hard-wired template mesh or skeleton, and the same architecture is universal for all shape categories.

**Model-Free Dynamic Representation:** Recent works [6, 24, 36, 50] extend the success of static 3D representations [9, 33, 40] to 4D by modeling the deformation between frames. Fig. 2C illustrates how the two closest works to ours, O-Flow [36] and LPDC [50], are related to our problem formulation. First, our method differs from O-Flow [36] and LPDC [50] in the representation of the space deformation. We represent the deformation through a novel canonical map factorization that is efficient and guarantees real world properties based on conditional neural homeomorphisms [14, 15], while O-Flow [36] uses a Neural-ODE [7] that also guarantees the production of a well-behaved deformation (see [22] for details) but with higher computational complexity than ours. LPDC [50] replaces the Neural-ODE [7] by a Multilayer Perceptron (MLP) to learn correspondences in parallel. However, the MLP deformation [20, 41, 44, 50] has difficulty to model a homeomorphism or express real world deformation properties. Note that both O-Flow [36] and LPDC [50] compute the reference surface in the first frame, which turns out to be a random choice, since the shape can be in an arbitrary deformation state in the first frame. Our reference shape is modeled in the learned canonical space induced by the canonical map, which is more stable and can be optimized (Fig. 1).

I3DMM [58] learns a near neutral/mean canonical template from human head scans, which limits its expressive ability. CASPR [46] and Garment Nets [10] learn a canonicalization of deformable objects, but rely on the ground

truth canonical coordinate supervision, which is often inaccessible. Other neural dynamic representations include the learned embedded graph [6] first proposed in [49] and parametric atlases [2]. Beyond 4D data, A-SDF [35] models the general articulated objects with a specially designed disentanglement network, but it cannot model correspondence. Instead, our method achieves stronger disentanglement by explicitly modeling the deformation.

**Invertible Networks for 3D representation:** Many works [3, 7, 14, 15, 19, 28, 39] have been proposed to construct invertible networks for generative models. In 3D deep learning, Neural-ODE [7] is widely used as a good model of deformation [22, 23, 25, 36] or transformation of point cloud [56]. ShapeFlow [25] learns a “Hub-and-spoke” surface deformation for 3D shape collection via ODEs, but is inefficient when applied to the 4D data since every frame needs to be lifted to the “hub” through integration. Besides ODEs, I-ResNet [3] is used in [21] to build invertible deformation for shape editing. Our method is inspired by Neural-Parts [42] where Real-NVP [15] is used to model the deformation from a sphere primitive to a local part. While we also use Real-NVP [15] for its simplicity and efficiency, we have two distinct differences compared to [42]: Our canonical shape is a learned implicit surface instead of a fixed sphere that can only capture local parts; we use the inverse of the Real-NVP to close the factorization cycle, while [42] uses the inverse in a complementary training path.

### 3. Method

As in Fig. 2A, given a sequence<sup>1</sup> of point cloud observations of one deforming instance, our goal is to reconstruct a sequence of surfaces. Instead of directly solving a correspondence map between two frames during reconstruction, we propose an architecture (Fig. 3) where the interframe correspondence is computed via a pivot **canonical shape**. We will call the map between a surface in any deformed frame and the canonical shape a **canonical map**.

#### 3.1. CaDeX and Canonical Map

Let us denote  $[x^i, y^i, z^i] \in \mathbb{R}^3$  as the 3D coordinates of the input 3D space<sup>2</sup>, in which a deformed surface  $S_i$  at time  $t_i$  is embedded. Consider a continuous bijective mapping (homeomorphism)  $\mathcal{H}_i : \mathbb{R}^3 \mapsto \mathbb{R}^3$  at time  $t_i$  that maps each deformed coordinate to its global (shared over different time frames) 3D coordinate  $[u, v, w] = \mathcal{H}_i([x^i, y^i, z^i])$ . Note that  $[u, v, w]$  has no index of time and can be seen as a globally consistent indicator of each correspondence trajectory across time in the input 3D space. Hence, we name  $[u, v, w]$  the **canonical deformation coordinates** of the position  $[x^i, y^i, z^i]$  at time  $t_i$  and call the  $uvw$  3D space

<sup>1</sup>Or a set, but for conciseness, we will only refer to the sequence.

<sup>2</sup>The superscript  $i$  refers to the time index.

the **Canonical Deformation Coordinate Space** (CaDeX) of the sequence  $\mathcal{S} = \{S_i\}$ . The homeomorphisms  $\mathcal{H}_t$  that transform  $[x^t, y^t, z^t]$  to  $[u, v, w]$  are called **canonical maps**. Since CaDeX is globally shared across time, we model the **canonical shape** (surface)  $U$  directly in CaDeX instead of selecting an input frame as is the case in [36, 50]. Taking advantage of neural fields [54], we model  $U$  as a level set of an occupancy field [33]:

$$U = \{[u, v, w] \mid \text{OccField}([u, v, w]) = l\}, \quad (1)$$

where  $l$  is the surface level. Using the inverse of each canonical map  $\mathcal{H}_i$  at time  $t_i$ , we can directly obtain each deformed surface at time  $t_i$  in the input 3D space as:

$$S_i = \{\mathcal{H}_i^{-1}([u, v, w]) \mid \forall [u, v, w] \in U\}. \quad (2)$$

The correspondence/deformation  $\mathcal{F}_{ij}$  that associates any coordinate (for both surface and non-surface points) from the 3D space at time  $t_i$  to the 3D space at time  $t_j$  can be factorized by the canonical maps as:

$$[x^j, y^j, z^j] = \mathcal{F}_{ij}([x^i, y^i, z^i]) = \mathcal{H}_j^{-1} \circ \mathcal{H}_i([x^i, y^i, z^i]). \quad (3)$$

Note that  $\mathcal{H}_t$  must be invertible; otherwise, the above deformation function cannot be defined. By now, any surface that is topologically isomorphic to the deformable instance satisfies the above definitions, leading to infinitely many valid canonical shapes and maps. In the following, we will optimize the canonical shapes and maps predicted by the architecture in Fig. 3 subject to the priors from the dataset.

#### 3.2. Canonical Map Implementation

**Neural Homeomorphism** One key technique of our implementation is an efficient way to parameterize and learn the homeomorphism between coordinate spaces. Unfortunately, the widely used Neural-ODEs [7] do not meet our efficiency requirements since a full integration would have to be applied to every frame. Inspired by [42], we utilize the Conditional Real-NVP [15] (Real-valued Non-Volume Preserving) or the NICE [14] (Nonlinear Independent Component Estimation) normalizing flow implementations to learn the homeomorphism. Taking the more general NVP [15] as an example (Fig. 3-C), with the network being a stack of Coupling Blocks [15], we apply NVP to 3D coordinates. During initialization, each block is randomly assigned an input split pattern; for example, a block always splits  $[x, y, z]$  to  $[x, y]$  and  $[z]$ . Given a condition latent code  $c$ , each block takes in 3D coordinates  $[x, y, z]$  and outputs the transformed coordinates  $[x', y', z']$  by changing one part based on the other part in the input coordinate split:

$$[x', y', z'] = [x, y, z \exp(s_\theta(x, y|c)) + t_\theta(x, y|c)] \quad (4)$$

where  $s_\theta(\cdot|c) : \mathbb{R}^2 \mapsto \mathbb{R}$  and  $t_\theta(\cdot|c) : \mathbb{R}^2 \mapsto \mathbb{R}$  are scale and translation predicted by any network conditioned on  $c$ .

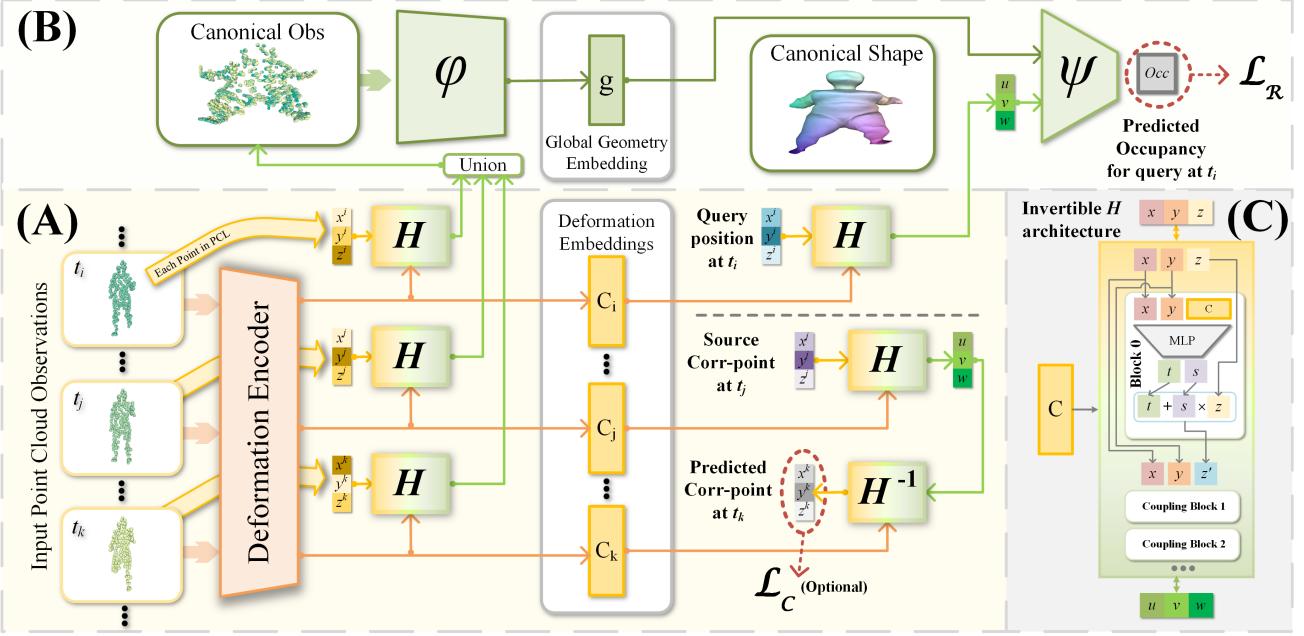


Figure 3. (A) **Canonical Map** (Sec. 3.2): A sequence (or a set) of input point clouds is first sent to the Deformation Encoder generating deformation embeddings  $c_i$  for each frame. Then the canonical map  $H$  can transform any coordinate (e.g. a yellow point in the point cloud, a blue query position for implicit field or a purple source point for correspondence) from any deformed frame to the canonical coordinate via  $H$  conditioning on the corresponding deformation embedding. The correspondence prediction (right bottom) can be obtained by directly mapping back the canonical coordinate through  $H^{-1}$ . (B) **Canonical Shape Encoder-Decoder** (Sec. 3.4): All input multi-frame point clouds are first transformed to the canonical space via  $H$  and are directly unioned to aggregate a canonical observation. The global geometry embedding  $g$  (unique across frames) is encoded via a PointNet [45]  $\phi$ , and the occupancy value for the canonical coordinate of a query position at  $t_i$  (blue point) is predicted through a standard OccNet [33]  $\psi$ . During training, the occupancy is supervised by  $\mathcal{L}_R$ , and the correspondence can be optionally supervised by  $\mathcal{L}_C$  (Sec. 3.5). (C) **The Real-NVP [15] invertible architecture of  $H$**  (Sec. 3.2).

Such a block models a bijection since the inverse can be immediately derived as:

$$[x, y, z] = \left[ x', y', \frac{z' - t_\theta(x', y' | c)}{\exp(s_\theta(x', y' | c))} \right]. \quad (5)$$

Therefore, the whole stack of blocks is invertible. If the activation functions in each block are continuous, then the whole network models a homeomorphism. NICE [14] is simply removing the scale freedom from the NVP block, i.e.:  $s_\theta(\cdot | c) \equiv 0$ . Note that the inverse of NVP and NICE is as simple as the forward, which induces our desired efficiency and simplicity, and enables using the definition in Eq. 3.

**H architecture** Note that in Eq. 2, each deformed surface  $S_i$  has a different canonical map  $\mathcal{H}_i$  that associates  $S_i$  with  $U$ . We implement them with the conditional real-NVP or NICE, denoted by  $H$  (noncalligraphic). Given the vector  $c_i$  that encodes the deformation information at time  $t_i$  such that  $\mathcal{H}_i(\cdot) \equiv H(\cdot; c_i)$ , where the network  $H$  is shared across different time frames. The canonical deformation coordinates are predicted as (Fig. 3-A, boxes marked with  $H$ ):

$$[u, v, w] = H([x^i, y^i, z^i]; c_i). \quad (6)$$

Note that on the right side of Eq. 6, the input coordinates and the deformation embedding have the index  $t_i$  since they come from each deformed frame. However, after application of the canonical map, the coordinates on the left side are independent of the index because there is only one global CaDeX for this sequence. Finally, the correspondence/deformation between two deformed frames (Eq. 3) can be implemented as:

$$[\hat{x}^j, \hat{y}^j, \hat{z}^j] = H^{-1}(H([x^i, y^i, z^i]; c_i); c_j) \quad (7)$$

where  $[\hat{x}^j, \hat{y}^j, \hat{z}^j]$  is the mapped position at time  $t_j$  of the original position  $[x^i, y^i, z^i]$  at time  $t_i$ . Regarding the choice of  $H$ , Real-NVP [15] can provide more flexible deformation since it has one more degree of freedom (scale); NICE [14] guarantees volume conservation (Sec. 3.3) that results in a more regularized deformation.

**Deformation Encoder** To obtain the per-frame deformation embedding  $c_i$  that is used as the condition of  $H$ , we demonstrate two kinds of inputs and three encoder types (Fig. 3-A, orange box). One direct approach is to employ a PointNet that summarizes the deformation code separately per frame (*PF*). If the input is a sequence of point clouds,

we can alternatively use the ST-PointNet variant proposed in [50] to get the deformation code (*ST*). The *ST* encoder processes the 4D coordinates and applies the pooling spatially and temporally. If the input is a set without order, we develop a 2-phase PointNet to obtain a global set deformation code (*SET*), and then use a 1-D code query network to output the deformation embedding  $c_i$  taking the query articulation angle and the global deformation code as input. Since these are not our main contributions, we refer the reader to the supplementary for details on these encoders.

### 3.3. Properties of the Canonical Map

The novel factorization and its implementation induce the following desired properties of real world deformation:  
**Cycle consistency:** The deformation/correspondence between deformed frames predicted by our factorization (Eq. 3, 7) is cycle consistent (path-invariant). The reason is that every canonical map maps any deformed frame in the sequence (or set) to the global CaDeX of this sequence (or set), and the canonical maps are invertible:

$$\mathcal{F}_{jk} \circ \mathcal{F}_{ij} = \mathcal{H}_k^{-1} \circ \mathcal{H}_j \circ \mathcal{H}_i^{-1} \circ \mathcal{H}_i = \mathcal{H}_k^{-1} \circ \mathcal{H}_i = \mathcal{F}_{ik}. \quad (8)$$

**Topology preserving deformation:** Since our factorization (Eq. 3, 7) is a composition of two homeomorphisms, the induced deformation function is thus a homeomorphism as well, and therefore never changes the surface topology.

**Volume conservation (NICE):** If  $H$  is implemented by NICE [14], then the predicted deformation preserves the volume of the geometry, which can be proved by the fact that the determinant of the Jacobian of every coupling block in NICE [14] is 1 (see Supp. for more details).

**Continuous deformation if  $c$  is continuous:** Some applications require the sequence  $\mathcal{S} = \{S\}$  to be dense on time axis, for example, modeling continuous deformation across time [36]. In this case, the deformation codes  $c$  become a function of time  $c(t)$ . Since all activation functions we are using in the canonical map are continuous, it is obvious that if  $c(t)$  is continuous, then the predicted deformation in Eq. 7 must be continuous across  $t$ .

### 3.4. Representing Canonical Shape

**Geometry Encoder** We represent the canonical shape in the CaDeX by a standard Occupancy Network [33]. The canonical map brings additional benefits for encoding the global geometry embedding (Fig. 3-B). Denote the observed point cloud at time  $t_i$  as  $X_i = \{[x_j^i, y_j^i, z_j^i] \mid j = 0, 1, \dots, N_i\}$ <sup>3</sup>, where  $[x_j^i, y_j^i, z_j^i]$  is the 3D coordinate of each point in the point cloud. The observations from different  $t_i$ 's are partial, noisy, and not aligned. We overcome such irregularity by using the same canonical map (Sec. 3.2)

<sup>3</sup>Note here the superscripts are still the index of the time, the subscripts are the index of the points in the cloud.

to obtain a canonical aggregated observation. Given the deformation embedding  $c_i$  per-frame, the canonical observations are merged via set union as:

$$\bar{X} = \bigcup_{t_i} \{ H([x_j^i, y_j^i, z_j^i]; c_i) \mid \forall [x_j^i, y_j^i, z_j^i] \in X_i \}. \quad (9)$$

The global geometry embedding  $g$  of the sequence  $\mathcal{S}$  is encoded by a PointNet  $\phi$ :  $g = \phi(\bar{X})$ .

**Geometry Decoder** Given the global geometry embedding  $g$ , we obtain the canonical shape encoded in  $g$  via an occupancy network [33] that takes  $g$  as well as the query position  $[u, v, w]$  in the CaDeX as input, and predicts the occupancy in the CaDeX:  $\hat{o} = \psi([u, v, w]; g)$ , where the decoder  $\psi$  is an MLP. However, the ground truth  $([u, v, w], o^*)$  supervision pair is unavailable in the CaDeX since the canonical shape is not known in advance and is learned during training. Available types of supervision are the query-occupancy pairs  $([x^i, y^i, z^i], o^{i*})$  in each deformed coordinate space where the deformed surface  $S_i$  is embedded at each time  $t_i$ . Therefore, we predict the occupancy field of any deformed frame through the canonical map via Eq. 6:

$$\hat{o} = \psi(H([x^i, y^i, z^i]; c_i); g) \quad (10)$$

### 3.5. Losses, Training, Inference

Our model is fully differentiable and is trained end-to-end. Following Eq. 10, the main loss function is the reconstruction loss in each deformed frame:

$$\mathcal{L}_R = \frac{1}{T} \sum_{i=1}^T \frac{1}{M_i} \sum_{j=1}^{M_i} BCE [\psi(H(p_j^i; c_i); g), o_j^{i*}] \quad (11)$$

where  $T$  is the total number of frames that have occupancy field supervision and  $M_i$  is the number of queried positions at each frame. We denote by  $p_j^i$  the  $j^{th}$  query position in frame  $t_i$ , and by  $o_j^{i*}$  the corresponding ground truth occupancy state. Optionally, if the ground truth correspondence pairs are given, we can utilize them as a supervision signal via Eq. 7. The additional correspondence loss reads:

$$\mathcal{L}_C = \frac{1}{|\mathcal{Q}|} \sum_{(p_k^i, p_k^j) \in \mathcal{Q}} \| H^{-1}(H(p_k^i; c_i); c_j) - p_k^j \|_l \quad (12)$$

where  $\mathcal{Q}$  is the set of ground truth correspondence pairs:  $p_k^i$  is the source position (Fig. 3 purple coordinate) in frame  $t_i$  and  $p_k^j$  is the ground truth corresponding position in frame  $t_j$ ; and  $k$  is the index of all supervision pairs. We denote by  $l$  the order of the error norm. Note that the cycle consistency guaranteed by our method (Sec. 3.3) does not depend on  $\mathcal{L}_C$ . The overall loss function is  $\mathcal{L} = w_R \mathcal{L}_R + w_C \mathcal{L}_C$ , where  $w_C$  can be zero if no correspondence supervision is provided. Note that there is no loss directly applied to the

predicted canonical deformation coordinates  $[u, v, w]$ . This gives the maximum freedom to the canonical shape to form a pattern that helps the prediction accuracy. All patterns of the canonical shape emerge automatically during training (see the Supplement for an additional discussion).

During training, our model is trained directly from scratch with the mandatory reconstruction loss (Eq. 11). For efficiency, at each training iteration, we randomly select a subset of frames in the input sequence and supervise the occupancy prediction. If the ground-truth correspondence supervision is also provided, we predict the corresponding position of surface points in the first frame for every other frame and minimize the correspondence loss in Eq. 12.

During inference, our model generates all surfaces of a sequence in parallel after a single marching cubes mesh extraction. Directly marching the CaDeX is intractable since it is learned. However, by using Eq. 10 as a query function, we can extract the mesh  $(\mathcal{V}_0, \mathcal{E}_0)$  in the first frame, which is equivalent to marching the CaDeX given the canonical map. The equivalent canonical mesh in the CaDeX is  $(\mathcal{V}_c, \mathcal{E}_c) = (H(\mathcal{V}_0; c_0), \mathcal{E}_0)$ . Then any mesh in other frames can be extracted as:

$$(\mathcal{V}_i, \mathcal{E}_i) = (H^{-1}(\mathcal{V}_c; c_i), \mathcal{E}_c). \quad (13)$$

Note that all meshes above share the same connectivity  $\mathcal{E}_0$ , so the mesh correspondence is produced. Eq. 13 can be implemented in batch to achieve better efficiency.

## 4. Results

To demonstrate CaDeX as a general and expressive representation, we investigate the performance in modeling three distinct categories: human bodies (Sec. 4.1), animals (Sec. 4.2) and articulated objects (Sec. 4.3). Finally, we examine the effectiveness of our design choice in Sec. 4.4.

**Metrics:** To measure our performance for shape and correspondence modeling, we follow the paradigm of [36, 50] and use the same metrics: evaluating the reconstruction accuracy using the IoU and Chamfer Distance, and the motion accuracy by correspondence  $l_2$ -distance error.

**Baselines:** We compare with the closest model-free dynamic representations. The main baselines described in Sec. 2 are: *O-Flow* [36] and *LPDC* [50] for sequence inputs and *A-SDF* [35] for articulated object set inputs.

### 4.1. Modeling Dynamic Human Bodies

We first demonstrate the power of modeling the dynamic human body across time. We use the same experiment setup, dataset, and split as [24, 36, 50]. The data are generated from D-FAUST [5], a real 4D human scan dataset. Following the setting of [36], the input is a randomly sampled sparse point cloud trajectory (300 points) of 17 frames evenly sampled across time. The ground-truth occupancy

Method	Seen Individual			Unseen Individual		
	IoU↑	CD↓	Corr↓	IoU↑	CD↓	Corr↓
PSGN-4D [16]	-	0.108	3.234	-	0.127	3.041
ONet-4D [33]	77.9%	0.084	-	66.6%	0.140	-
O-Flow [36]	79.9%	0.073	0.122	69.6%	0.095	0.149
LCR [24]	81.8%	0.068	-	68.2%	0.100	-
LCR-F [24]	81.5%	0.068	-	69.9%	0.094	-
Ours	<b>85.5%</b>	<b>0.056</b>	<b>0.100</b>	<b>75.4%</b>	<b>0.074</b>	<b>0.126</b>

Table 1. Results on D-FAUST [5] human bodies, trained **without** correspondence supervision.

Method	Seen Individual			Unseen Individual		
	IoU↑	CD↓	Corr↓	IoU↑	CD↓	Corr↓
PSGN-4D [16]	-	0.101	0.102	-	0.119	0.131
O-Flow [36]	81.5%	0.065	0.094	72.3%	0.084	0.117
LPDC [50]	84.9%	0.055	0.080	76.2%	0.071	0.098
Ours(NICE)	85.4%	0.051	0.082	75.6%	0.070	0.104
Ours(ST)	86.7%	0.046	0.077	78.1%	0.063	0.095
Ours(PF)	<b>89.1%</b>	<b>0.039</b>	<b>0.070</b>	<b>80.7%</b>	<b>0.055</b>	<b>0.087</b>

Table 2. Results on D-FAUST [5] human bodies, trained with correspondence supervision.

field as well as the optional surface point correspondence are provided. Our default model is configured by using the *ST*-encoder (Sec. 3.2) and an NVP homeomorphism. The following tables and sections assume such a configuration if not otherwise specified. We test the performance of the *PF*-encoder and the NICE homeomorphism variants as well. The experiments are divided into training without correspondence (Tab. 1) and training with correspondence (Tab. 2) tracks for fair comparison between methods. The testing set has two difficulty levels: unseen motion and unseen individuals [36].

Quantitative comparisons in Tab. 1, 2 indicate that our method outperforms state-of-the-art methods by a significant margin. The qualitative comparison in Fig. 4 shows the advantage of our method in capturing fast moving parts and shape details (marked with red). We attribute such improvements to two main reasons: First, our factorization of the deformation and its implementation provide a strong regularization that other approaches like O-flow [36] can only achieve with an ODE integration. Additionally, supervising a per-frame implicit reconstruction in our model is equivalent to the dense cross-frame reconstruction supervision in [50]. Second, our shape prior is stored in the learned canonical space (marked green in Fig. 4), which is relatively stable across different sequences as shown in the figure. When training without correspondence (Tab. 1), our method can learn the correspondence implicitly and reach a similar reconstruction performance as [50] in Tab. 2, which is trained with dense parallel correspondence supervision. Comparing the different configurations of our method in Tab. 2, the NICE [14] version has a performance drop since the deformation is strongly regularized to conserve the volume, but

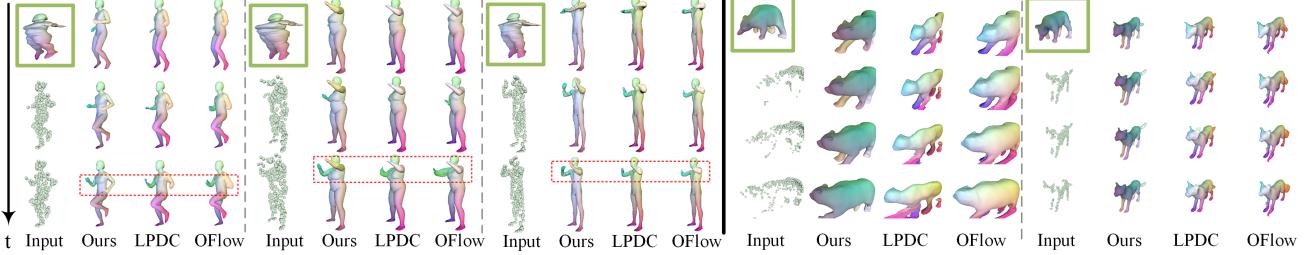


Figure 4. **Left:** Human body modeling (Sec. 4.1); **Right:** Animal body modeling (Sec. 4.2). The left top figure marked in the green box is our canonical shape, the first input is not displayed. The colors of the meshes encode the correspondence. More results are in the Supp.

Input	Mehtod	Seen individual			Unseen individual		
		IoU↑	CD↓	Corr↓	IoU↑	CD↓	Corr↓
PCL	O-Flow [36]	70.6%	0.104	0.204	57.3%	0.175	0.285
	LPDC [50]	72.4%	0.085	0.162	59.4%	0.149	0.262
	Ours	<b>80.3%</b>	<b>0.061</b>	<b>0.133</b>	<b>64.7%</b>	<b>0.127</b>	<b>0.239</b>
Dep	O-Flow [36]	63.0%	0.131	0.250	49.0%	0.228	0.374
	LPDC [50]	58.4%	0.160	0.249	45.8%	0.261	0.388
	Ours	<b>71.1%</b>	<b>0.094</b>	<b>0.186</b>	<b>55.7%</b>	<b>0.175</b>	<b>0.301</b>

Table 3. Results on DeformingThings4D [57] animal bodies, PCL and Dep correspond to the input types.

this is achieved by freezing half of the capacity (scale freedom). We note that the naive per-frame encoder (*PF*) works better compared to the spatial-temporal encoder [50] (*ST*). A potential reason is that the per-frame encoder provides a higher canonicalization level since when deciding the deformation code  $c_i$ , no information from other frames can be considered, so the *PF* encoder might avoid overfitting.

## 4.2. Modeling Dynamic Animals

We experiment with a more challenging setting: modeling different categories of animals with one model. We generate the same supervision types as Sec. 4.1 based on the DeformingThings4D-Animals [57] dataset (DT4D-A). We use 17 animal categories and generate 2 types of input observations: Sparse point cloud input as Sec. 4.1 as well as the monocular depth video input from a randomly posed static camera. We assume that the camera view point estimation problem is solved so all partial observations live in one global world frame. All models are trained across all animal categories. We refer the reader to the supplementary material for more details. Such a setting is more challenging because animals have both large shape and motion variance across categories. Additionally, the models are required to aggregate information across time and hallucinate the missing parts in depth observation inputs. Quantitative results of both the sparse point cloud input and the depth input in Tab. 3 as well as the qualitative results in Fig. 4 indicate that our method outperforms state-of-the-art methods in these challenging settings. In addition to the reasons mentioned in Sec. 4.1, the improvement when predicting from depth

Input	Method	IoU↑	CD↓	Corr↓	t(s)	$\theta$ (deg)
PCL	A-SDF [35]	55.2%	0.127	-	3.44	3.38
	LPDC [50]	49.2%	0.171	0.230	0.53	3.00
	Ours	<b>58.9%</b>	<b>0.118</b>	<b>0.160</b>	1.12	2.75
Dep	A-SDF [35]	53.9%	0.127	-	3.65	5.06
	LPDC [50]	46.4%	0.195	0.269	0.54	4.85
	Ours	<b>56.4%</b>	<b>0.116</b>	<b>0.161</b>	1.26	4.34

Table 4. Results on Shape2Motion [35, 53] articulated objects, PCL and Dep correspond to the input types. The average performance across 7 categories is reported, we refer the readers to our supplementary for the full table.  $t$  is the surface generation average time and  $\theta$  is the average angle prediction error.

observation can be attributed to our design of the canonical observation encoder (Sec. 3.4) that explicitly aggregates observations in the CaDeX.

## 4.3. Modeling Articulated Objects

We extend CaDeX from modeling the 4D nonrigid surface sequence to representing semi-nonrigid articulated object sets. We generate the dataset and inputs as in Sec. 4.2 from [35] based on Shape2Motion [53], which contains 7 categories of articulated objects with 1 or 2 deformable angles. We configure the model with the *SET* encoder (Sec. 3.2) that produces the global dynamic code and then use the articulation angle to query the deformation code for each frame (for details, see the Supplement). During training, we input the sparse point cloud of 4 randomly sampled deformed frames of one object and then use the ground-truth angle to query per-frame deformation codes; finally, the model predicts the occupancy field for 4 seen (input) frames and 4 unseen frames. We supervise both  $\mathcal{L}_R$  and  $\mathcal{L}_C$ . For completeness, we also predict the articulation angles of the input frames by a small head in the encoder and supervise them. Each category is trained separately for all methods. Note that A-SDF [35] demonstrates the auto-decoder setup, but it only solves half of our problem without correspondence. Simultaneously solving the shape and the correspondence leads to difficulties when applying an auto-decoder with optimization during testing, so we leave this as a future direction. For fair comparison, we adapt A-SDF

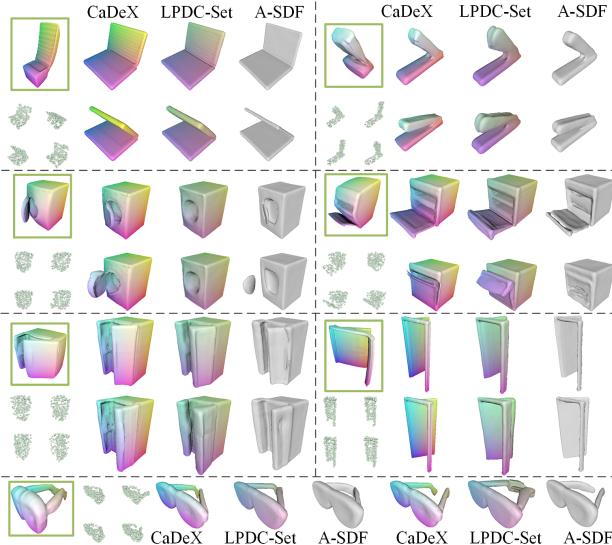


Figure 5. Articulated objects modeling (Sec. 4.3) with 7 distinct categories. The left-top figure marked in the green box is our canonical shape, the four small figures next to it are the inputs. The first row is the reconstruction of an observed deformation angle and the second is for an unobserved angle. Note that A-SDF has no correspondence so is not colored.

with a similar encoder as our model and adapt the decoder to predict the occupancy. We also compare with LPDC [50] which is also adapted to use a similar encoder as ours.

Tab. 4 summarizes the average performance across 7 object categories while Fig. 5 presents the qualitative comparison. Both of them show our state-of-the-art performance on modeling general articulated objects. We produce an accurate reconstruction, while providing the correspondence prediction that A-SDF [35] can not predict. Thus, the marching cube is needed for each frame in [35] and results in a longer inference time as shown in Tab. 4. Note that our method preserves the topology when deforming the objects (Fig. 5 oven) while [35] does not have such guarantees. This is the main reason that our method has a performance drop on eyeglasses category since the dataset contains many unrealistic deformations where the legs of the eyeglasses get crossed. Additionally, our method models more details in the moving parts (e.g., the inner side of the refrigerator door in Fig. 5) due to the learned canonical space, which provides a stable container for the shape prior.

#### 4.4. Ablation Study

We show the effectiveness of our design as the following: First, we replace the invertible canonical map with a one-way MLP that maps the deformed coordinates to the canonical space (such setting is similar to [13, 58, 59]). Since the mapping is one-way, we supervise the correspondence by enforcing the consistency in the canonical space. Every frame needs a separate application of marching cubes

	IoU $\uparrow$	CD $\downarrow$	Corr $\downarrow$	t(s)
Full	66.5%	0.128	0.223	1.8
MLP	61.9%	0.161	0.303	20.5
No G-Enc	63.4%	0.141	0.216	1.7

Table 5. Ablation study,  $t$  is the average surface generation time.

to extract meshes in this version. Second, we remove the geometry encoder in the canonical space and obtain the global geometry embedding via a latent fusion using the ST-encoder [50]. We demonstrate the performance of the deer subcategory from DT4D-A [57] with point cloud inputs (Sec. 4.2). Tab. 5 shows the performance difference, where we observe a significant performance decrease as well as longer inference times when using MLP instead of homeomorphisms. Additionally, we observe the drop in reconstruction accuracy when removing the geometry encoder in the canonical space (Sec. 3.4). We present more details in the supplementary material.

#### 5. Limitations

Our method guarantees several desirable properties and achieves state-of-the-art performance on a wide range of shapes, but still has limitations that need future exploration. Although we can produce continuous deformation across time if  $c(t)$  is continuous, the continuity of  $c$  is not guaranteed in the ST-encoder [50] that we use. Therefore, when the input undergoes a large discontinuity, we do observe a trembling in the output of both LPDC [50] and our method. Another issue is that although our method preserves the topology, sometimes the real world deformation also results in topology changes. Future work can explore how to selectively preserve or alter the topology. Finally, it is currently nontrivial to adapt our method in an auto-decoder framework [35, 40] since it requires simultaneously optimizing the canonical map (deformation) and the canonical shape during testing, which future work can explore.

#### 6. Conclusion

We introduced a novel and general representation for dynamic surface reconstruction and correspondence. Our key insight is the factorization of the deformation by continuous bijective canonical maps through a learned canonical shape. We prove that our representation guarantees cycle consistency and topology preservation, as well as (if desired) volume conservation. Extensive experiments on reconstructing humans, animals, and articulated objects demonstrate the effectiveness and versatility of our approach. We believe that CaDeX enables more possibilities for future research on modeling and learning from our dynamic real world.

**Acknowledgement:** The authors appreciate the support of the following grants: ARL MURI W911NF-20-1-0080, NSF TRIPODS 1934960, NSF CPS 2038873, ARL DCIST CRA W911NF-17-2-0181, and ONR N00014-17-1-2093.

## References

- [1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. [1](#), [2](#)
- [2] Jan Bednarik, Vladimir G Kim, Siddhartha Chaudhuri, Shaifali Parashar, Mathieu Salzmann, Pascal Fua, and Noam Aigerman. Temporally-coherent surface reconstruction via metric-consistent atlases. *arXiv preprint arXiv:2104.06950*, 2021. [1](#), [3](#)
- [3] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019. [3](#)
- [4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. [1](#), [2](#)
- [5] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [2](#), [6](#)
- [6] Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2021. [1](#), [2](#), [3](#)
- [7] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018. [2](#), [3](#)
- [8] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. *arXiv preprint arXiv:2104.03953*, 2021. [2](#)
- [9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. [2](#)
- [10] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. *arXiv preprint arXiv:2104.05177*, 2021. [2](#)
- [11] Julian Chibane, Thimo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. [2](#)
- [12] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 612–628. Springer, 2020. [2](#)
- [13] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10296, 2021. [2](#), [8](#)
- [14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. [2](#), [3](#), [4](#), [5](#), [6](#)
- [15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. [2](#), [3](#), [4](#)
- [16] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. [6](#)
- [17] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. [2](#)
- [18] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A Funkhouser. Deep structured implicit functions. 2019. [2](#)
- [19] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889. PMLR, 2015. [3](#)
- [20] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. [2](#)
- [21] Bharath Hariharan Guandao Yang, Serge Belongie and Vladlen Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 33, 2021. [3](#)
- [22] Kunal Gupta. *Neural Mesh Flow: 3D Manifold Mesh Generation via Diffeomorphic Flows*. University of California, San Diego, 2020. [2](#), [3](#)
- [23] Jingwei Huang, Chiyu Max Jiang, Baiqiang Leng, Bin Wang, and Leonidas Guibas. Meshode: A robust and scalable framework for mesh deformation. *arXiv preprint arXiv:2005.11617*, 2020. [2](#), [3](#)
- [24] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5340–5350, 2021. [2](#), [6](#)
- [25] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, Leonidas Guibas, et al. Shapeflow: Learnable deformations among 3d shapes. *arXiv preprint arXiv:2006.07982*, 2020. [2](#), [3](#)
- [26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. [2](#)
- [27] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. *arXiv preprint arXiv:2109.11399*, 2021. [2](#)

- [28] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 3
- [29] Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9730–9740, 2021. 1
- [30] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 1, 2
- [31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 2
- [32] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 2
- [33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 3, 4, 5, 6
- [34] 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 *European conference on computer vision*, pages 405–421. Springer, 2020. 2
- [35] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021. 3, 6, 7, 8
- [36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5379–5389, 2019. 1, 2, 3, 5, 6, 7
- [37] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021. 2
- [38] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npm3: Neural parametric models for 3d deformable shapes. *arXiv preprint arXiv:2104.00702*, 2021. 2
- [39] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017. 3
- [40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2, 8
- [41] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020. 2
- [42] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021. 3
- [43] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2
- [44] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1, 2
- [45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 4
- [46] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. *arXiv preprint arXiv:2008.02792*, 2020. 2
- [47] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 1, 2
- [48] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Sanimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021. 2
- [49] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*, pages 80–es. 2007. 3
- [50] Jiapeng Tang, Dan Xu, Kui Jia, and Lei Zhang. Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6022–6031, 2021. 1, 2, 3, 5, 6, 7, 8
- [51] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-gif: Neural generalized implicit functions for animating people in clothing. *arXiv preprint arXiv:2108.08807*, 2021. 2
- [52] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. *arXiv preprint arXiv:2106.11944*, 2021. 2
- [53] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinpeng Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 2, 7

- [54] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *arXiv preprint arXiv:2111.11426*, 2021. 3
- [55] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3d dynamic scene representations for robot manipulation. *arXiv preprint arXiv:2011.01968*, 2020. 1
- [56] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 2, 3
- [57] Takafumi Taketomi Yang Li, Hikari Takehara, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. *arXiv preprint arXiv:2105.01905*, 2021. 2, 7, 8
- [58] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12803–12813, 2021. 1, 2, 8
- [59] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1429–1439, 2021. 2, 8