A-SDF: Learning Disentangled Signed Distance Functions for Articulated Shape Representation

Jiteng Mu[1], Weichao Qiu[2], Adam Kortylewski[2], Alan Yuille[2],
Nuno Vasconcelos[1], Xiaolong Wang[1]
[1]UC San Diego, [2]Johns Hopkins University

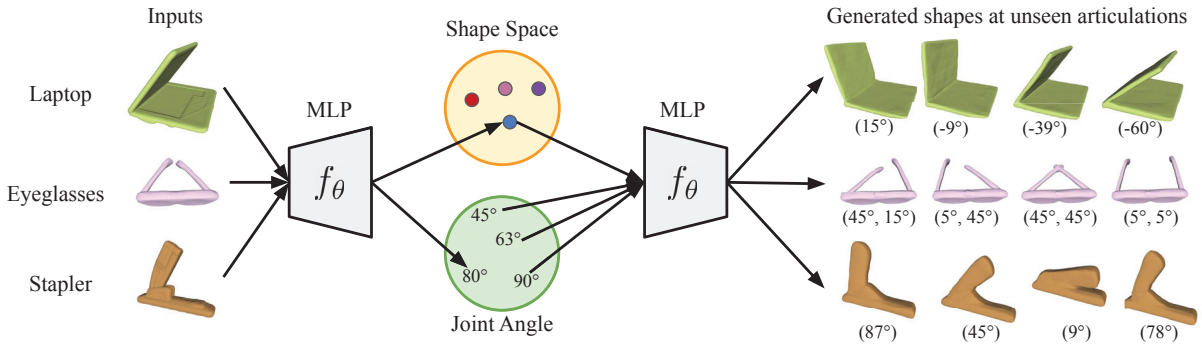

Figure 1: We represent articulated objects with separate codes for encoding shape and articulation. During inference, given an unseen instance, our model first infers the shape and articulation codes via back-propagation. With the inferred shape code, we can generate shapes at unseen angles by only changing the articulation code.

Abstract

Recent work has made significant progress on using implicit functions, as a continuous representation for 3D rigid object shape reconstruction. However, much less effort has been devoted to modeling general articulated objects. Compared to rigid objects, articulated objects have higher degrees of freedom, which makes it hard to generalize to unseen shapes. To deal with the large shape variance, we introduce Articulated Signed Distance Functions (A-SDF) to represent articulated shapes with a disentangled latent space, where we have separate codes for encoding shape and articulation. With this disentangled continuous representation, we demonstrate that we can control the articulation input and animate unseen instances with unseen joint angles. Furthermore, we propose a Test-Time Adaptation inference algorithm to adjust our model during inference. We demonstrate our model generalize well to out-of-distribution and unseen data, e.g., partial point clouds and real-world depth images. Project page with code: https://jitengmu.github.io/A-SDF/.

1. Introduction

Modeling articulated objects has wide applications in multiple fields including virtual and augmented reality, object functional understanding, and robotic manipulation. To understand articulated objects, recent works propose to train deep networks for estimating per-part poses and the joint angle parameters of an object instance in a known category [40, 82]. However, if we want to interact with the articulated object (e.g., open a laptop), estimating its static state is not sufficient. For example, an autonomous agent needs to predict what the articulated object shape will be like after interactions for planning its action.

In this paper, we introduce Articulated Signed Distance Functions (A-SDF), a differentiable category-level articulated object representation, which can reconstruct and predict the object 3D shape under different articulations. A differentiable model is useful in applications requiring back-propagation through the model to adjust inputs, such as rendering in graphics and model-based control in robotics.

We build our articulated object model based on the deep implicit Signed Distance Functions [58]. While implicit functions have recently been widely applied in modeling static object shape with fine details [64, 65, 72], much less effort has been devoted to modeling general articulated objects. We observe that models with a single shape code input, such as DeepSDF [58], cannot encode the articulation variation reliably. It is even harder for the models to generalize to unseen instances with unseen joint angles.

To improve the generalization ability, we propose to

model the joint angles explicitly for articulated objects. Instead of using a single code to encode all the variance, we propose to use one shape code to model the shape of object parts and a separate articulation code for the joint angles. To achieve this, we design two separate networks in our model: (i) a shape embedding network to produce a shape embedding given a shape code input; (ii) an articulation network which takes input both the shape embedding and an articulation code to deform the object shape. During training, we use the ground-truth joint angles as inputs and learn the shape code jointly with both model parameters. To encourage the disentanglement, we enforce the same instance with different joint angles to share the same shape code.

During inference, given an unseen instance with unknown articulation, we first infer the shape code and articulation code via back-propagation. Given the inferred shape code, we can simply adjust the articulation code to generate the instance at different articulations. We visualize the generation process and results for a few objects in Figure 1. Note the part geometry remains the same as we fix the inferred shape code during generation.

To generalize our model to out-of-distribution and unseen data, e.g., partial point clouds and real-world depth images, we further propose a Test-Time Adaptation (TTA) approach to adjust our model during inference. Note that our unique model architecture with separate shape embedding and articulation network provides the opportunity to do so: the separation of shape embedding and articulation network ensures the disentanglement is maintained when the shape embedding network is adapted. We adapt the shape embedding network to the current test instance by updating its parameters, while fixing the parameters of the articulation network. This procedure allows A-SDF to reconstruct and generate better shapes aligning with the inputs.

To our knowledge, our work is the first paper tackling the problem of generic articulated object synthesis in the implicit representation context. We summarize the contributions of our paper as follows. First, we propose Articulated Signed Distance Functions (A-SDF) and a Test-Time Adaptation inference algorithm to model daily articulated objects. Second, the disentangled continuous representation allows us to control the articulation code and generate corresponding shapes as output on unseen instances with unseen joint angles. Third, the proposed representation shows significant improvement on interpolation, extrapolation, and shape synthesis. More interestingly, our model can generalize to real-world depth images from the RBO dataset [45] and we quantitatively demonstrate superior performance over the baselines.

2. Related Work

Neural Shape Representation. A large body of work [80, 19, 9, 11, 41, 63, 17, 78, 77, 89, 13, 1, 15] has focused on investigating efficient and accurate 3D object representations. Recent advances suggest that representing 3D objects as continuous and differentiable implicit functions [18, 58, 46, 10, 35, 26, 84, 72, 54, 64, 48, 71, 76] can model various topologies in a memory-efficient way. The basic idea is to exploit neural networks to parameterize a shape as a decision boundary in 3D. Most of these work is limited to modeling static objects and scenes [18, 26, 84, 72, 54, 64, 48, 71, 76]. Different from previous works, our method models articulated objects in a category-level by learning a disentangled implicit representation and we test our model on real depth images. Comparisons to implicit neural networks on deformable shapes will be discussed in depth in the following.

Articulated Humans. One line of work leverages parametric mesh models [43, 38, 92, 5] to estimate shape and articulation for faces [74, 62, 66], hands[16], humans bodies [60, 4, 86, 34, 28, 56, 85, 79], and animals [91, 29, 36, 90] by directly inferring shape and articulation parameters. However, such parametric models require substantial efforts from experts to construct and is hard to generalize to large-scale object categories. To address the challenges, another line of work [57, 53, 75, 12, 8, 55] employs neural networks to learn shapes from data. For example, Niemeyer et al. [53] learned an implicit vector field and deformed shapes in a spatial-temporal space. However, the design does not allow for controlling each part separately. Recently, Hoang et al. [75] defined shapes using patches and shapes can be deformed by manipulating the extrinsic parameters of each patch. Nevertheless, the learned patches do not correspond to parts and the method fails at large deformation. Deng et al. [12] modeled each human body part by a separate implicit function. However, the method is limited to instance-level and requires skinning weights for the learning process. In comparison, our method is category-level on general articulated objects and we assume no part label. Therefore, these previous approaches are not directly comparable to ours. Besides, we model articulated object poses with joint angles, which allows us to articulate each joint separately.

Articulated General Objects. Though reconstructing articulated humans has attracted lots of attention in the community, modeling 3D shapes of daily articulated objects is an under-explored field in terms of both data and approaches. Unlike modeling humans where lots of human priors are injected, modeling articulated daily objects poses additional challenges by assuming no prior part geometry and labels, articulation status, joint type, joint axis and joint location about the category. One recent popular paradigm of research [20, 31, 44, 14, 47, 82] focuses on estimating 6D poses of articulated objects. For example, Desingh et al. [14] proposed a instance-level factored approach to estimate the poses of articulated objects with articulation constraints. However, 6D pose information may not be suffi-

cient for tasks that require detailed shape information, such as robotic manipulation [32, 83, 50, 49]. In this work, we show that implicit functions are suitable for the daily articulated objects modeling. We demonstrate that once an implicit function is learned, shapes at unseen articulations can be generated by manipulating the articulation code.

Disentangled Representation. Disentangled representations focus on modeling complex variations in a low dimensional space, where individual factors control different types of variation. Previous work [37, 22, 7, 24, 33, 30, 23, 52, 67, 6, 88, 87, 68, 61, 42, 2, 59] has shown that disentangled representations are essential to learn meaningful latent space for 2D image synthesis. For example, Zhou et al. [87] proposed an auto-encoder architecture to disentangle human pose and shape. Different from previous work, we focus on modeling general articulated objects in 3D with interpretable pose codes.

Adaptation on Test Instance. Learning on test instance has been recently applied for adapting a trained model to out-of-distribution in multiple applications, including image recognition [25, 51, 73], super-resolution and synthesis [70, 69, 3], and mesh reconstruction and generation [39, 27]. For example, Li et al. [39] exploits training in test-time with self-supervision for consistent mesh reconstruction in a single video. Inspired by previous works, we develop a Test-Time Adaptation inference algorithm in implicit functions for better shape reconstruction and generation, where our disentanglement-based model architecture is the key to allow for Test-Time Adaptation.

3. Method

We propose Articulated Signed Distance Functions (A-SDF), a differentiable category-level articulated object representation to reconstruct and predict the object 3D shape under different articulations. Our model takes sampled 3D point locations, shape codes, and articulation codes as inputs, and outputs SDF values (signed distance) that measure the distance of a point to the closest surface point. The key insight is that all shape codes of the same instance should be identical, independent of its articulation. We argue that, even though the shapes look quite different for different articulations of the same instance, a good representation should capture this variability in a low dimensional space, since the part geometry remains unchanged. An overview of our method is presented in Figure 2. As different categories vary a lot in terms of articulation, we train separate networks for each category.

Our model is based on DeepSDF [58]. DeepSDF is an effective and widely acknowledged baseline and its simple network design allows us to focus on the effectiveness of our key idea rather than designing complex architectures. Note that our approach is generic and it can be applied to other advanced architectures as well. For a fair comparison, our model is designed with similar model size as DeepSDF. Furthermore, compared to feed forward designs [46, 64], the optimization-based shape modeling is naturally compatible with Test-Time Adaptation to address the out-of-distribution data as described in Section 3.3.

In the following, we describe how to learn a model encouraging the disentanglement of shape and articulation. We also introduce a Test-Time Adaptation inference technique allowing for generating unseen shapes at unseen articulations with high quality.

3.1. Formulation

Consider a training set of N instance models for one object category. Each instance is articulated into M poses, leading to a training set of $N \times M$ shapes of the category. Let $\mathcal{X}_{n,m}$ denote the shape articulated from instance n with articulation m, where $n \in \{1, \dots, N\}, m \in \{1, \dots, M\}$.

Each shape $\mathcal{X}_{n,m}$ is assigned with a shape code $\boldsymbol{\phi}_n \in \mathbb{R}^C$, where C denotes the latent dimension, and an articulation code $\boldsymbol{\psi}_m \in \mathbb{R}^D$ with D denoting the number of DoFs. The shape code $\boldsymbol{\phi}_n$ is shared across the same object instance n across different articulations. During training, we maintain and update one shape code for each instance. We use joint angles to represent the articulation code. For example, the articulation code of a 2-DoF object (e.g., eyeglasses) with both joints articulated to $45°$ is $\boldsymbol{\psi}_m = (45°, 45°)$. The joint angle is defined as a relative angle to the canonical pose of the object.

Let $\boldsymbol{x} \in \mathbb{R}^3$ be a sampled point from a shape. For notational simplicity, we omit the subscripts and denote $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as the corresponding shape and articulation code of the shape. As shown in Figure 2, an Articulated Signed Distance Function f_θ is finally defined with the auto-decoder architecture, which is composed of a shape embedding network f_s and an articulation network f_a,

$$f_\theta(\boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\psi}) = f_a[f_s(\boldsymbol{x}, \boldsymbol{\phi}), \boldsymbol{x}, \boldsymbol{\psi}] = s, \tag{1}$$

where $s \in \mathbb{R}$ is a scalar SDF value (the signed distance to the 3D surface). The sign of the SDF value indicates whether the point is inside (negative) or outside (positive) the watertight surface. The 3D shape is implicitly represented by the zero level-set $f_\theta(\cdot) = 0$.

3.2. Training

During training, given the ground-truth articulation code $\boldsymbol{\psi}$, sampled points and their corresponding SDF values, the model is trained to optimize the shape code $\boldsymbol{\phi}$ and the model parameters θ.

The training process is illustrated in Figure 2. The shape code is first concatenated with a sampled point $\boldsymbol{x}$ to form vector of dimension $C+3$ and input to the shape embedding network. Similarly, the articulation code $\boldsymbol{\psi}$ (joint angles) is

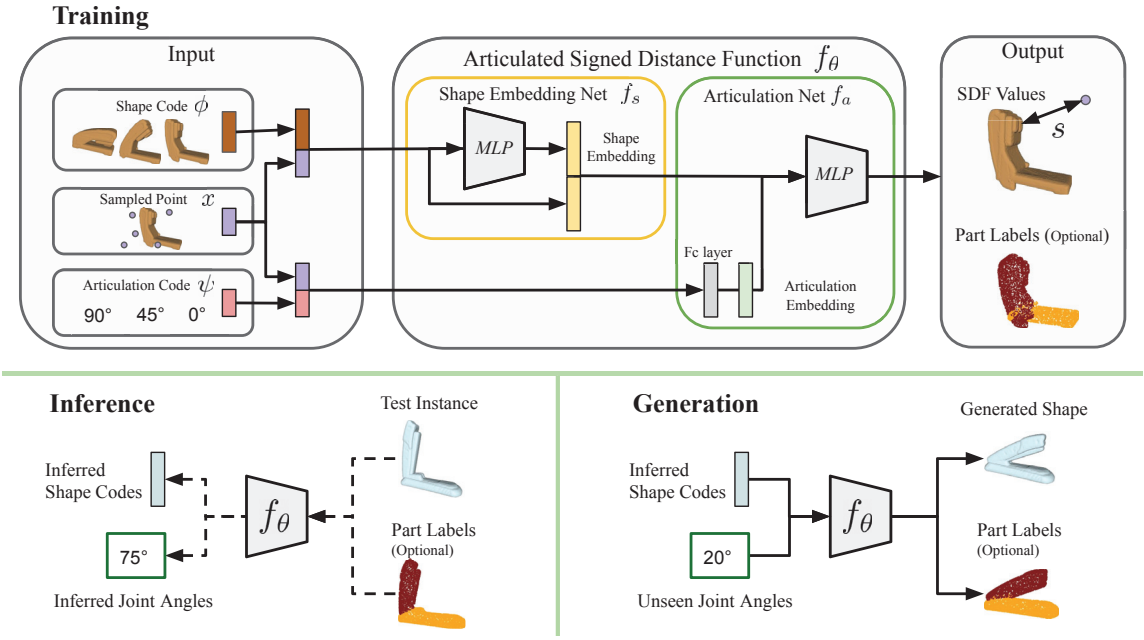

Figure 2: Overview of the proposed method. At training time, the articulation code and randomly sampled shape codes are first concatenated with a sampled point separately. The produced embeddings are then input to an articulated signed distance function f_θ to regress SDF values (signed distance) and predict part labels (optional). Note that the same instance is associated with one shape code regardless of its articulation state, as illustrated with brown staplers. During inference, back-propagation is used to jointly infer the shape code and articulation code for an unseen instance. With the inferred shape code, the model can faithfully generate new shape at unseen articulations.

first concatenated with a sampled point $\boldsymbol{x}$ to form a $D+3$ dimensional vector.

Then the articulation network takes the concatenated shape embedding and articulation code to predict the SDF value for the input 3D point. It's worth noting that, at the beginning of the articulation network, a fully connected layer is employed to embed the articulation code into a space of dimension $C+3$. Besides the SDF value, part supervision is optionally provided for training, using different labels to index different object parts. The number of object parts is fixed per category. When part supervision is available, a linear classifier is added to the last hidden layer of the articulation network to simultaneously output the part label.

The training loss functions are defined as following. Let K be the number of sampled points per shape. The function f_θ is trained with the per-point L_1 loss function to regress SDF values,

$$\mathcal{L}^s(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{1}{K}\sum_{k=1}^{K}\Big|\Big|f_\theta(\boldsymbol{x}_k, \boldsymbol{\phi}, \boldsymbol{\psi}) - s_k\Big|\Big|_1, \quad (2)$$

where $\boldsymbol{x}_k \in \mathcal{X}$ is a point of instance $\mathcal{X}$, s_k the corresponding ground-truth SDF value, and $k \in \{1, \ldots, K\}$.

When the object part labels are available, we include a complementary auxiliary part classification loss as,

$$\mathcal{L}^p(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{1}{K}\sum_{k=1}^{K}\Big[CE\Big(f_\theta(\boldsymbol{x}_k, \boldsymbol{\phi}, \boldsymbol{\psi}), p_k\Big)\Big], \quad (3)$$

where CE denotes the cross-entropy loss and p_k is the ground-truth label for the part containing $\boldsymbol{x}_k$. Intuitively, the part classification task helps the network disambiguate the object parts.

The full loss $\mathcal{L}(x, \boldsymbol{\phi}, \boldsymbol{\psi})$ is defined as,

$$\begin{aligned}\mathcal{L}(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \mathcal{L}^s(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) \\ + \lambda_p\mathcal{L}^p(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) + \lambda_{\boldsymbol{\phi}}||\boldsymbol{\phi}||_2^2.\end{aligned} \quad (4)$$

Following [58], we include a zero-mean multivariate-Gaussian prior per shape latent code $\boldsymbol{\phi}$ to facilitate learning a continuous shape manifold. Part coefficient λ_p and shape code coefficient $\lambda_{\boldsymbol{\phi}}$ are introduced to balance different losses. We optionally set $\lambda_p = 0$ when the part labels are not available.

At training time, the shape codes are randomly initialized with a Gaussian distribution at the very beginning of training. Each shape code is then optimized through the training steps and it is shared across all the shapes articulated from the same instance. The articulation codes are constants given from the ground-truths. The objective is to

optimize the loss function over all $N \times M$ training shapes, defined as follows,

$$\arg\min_{\theta, \boldsymbol{\phi}_n} \sum_{n=1}^{N} \sum_{m=1}^{M} \mathcal{L}(\mathcal{X}_{n,m}, \boldsymbol{\phi}_n, \boldsymbol{\psi}_m), \tag{5}$$

where θ is the network parameters.

3.3. Inference

Basic Inference. In the inference stage, illustrated in the Inference Section of Figure 2, an instance $\mathcal{X}$ is given and the goal is to recover the corresponding shape code $\boldsymbol{\phi}$ and the articulation code $\boldsymbol{\psi}$. This can be done by back-propagation. The two codes are initialized randomly, the articulation network parameters are fixed, and the codes are inferred jointly by solving the optimization with the following objective,

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\phi}, \boldsymbol{\psi}} \mathcal{L}(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}). \tag{6}$$

However, directly applying Equation 6 is hard, since both the shape and articulation spaces are non-convex. Gradient based optimization can converge to local minima, producing an inaccurate estimation.

To overcome the challenge, we first use Equation 6 to optimize both shape and articulation codes as our initial estimation. In practice, we observe that the articulation code usually converges to a good estimate but the inferred shape codes tends to lead to noisy outputs. So the estimated articulation code $\hat{\boldsymbol{\psi}}$ is then kept and the shape code is discarded. In the second step, the shape code is re-initialized, the articulation code is fixed to $\hat{\boldsymbol{\psi}}$, and the optimization is only solved for the shape code $\hat{\boldsymbol{\phi}}$.

Test-Time Adaptation Inference. Though fixing the parameters of decoder f_θ makes sense if the test distribution aligns well with the training distribution, it can be problematic given out-of-distribution observations. To generalize better to out-of-distribution data, the Test-Time Adaptation (TTA) for shape embedding network f_s is further introduced. It is built on the basic inference procedure with the estimated shape code $\hat{\boldsymbol{\phi}}$ and articulation code $\hat{\boldsymbol{\psi}}$. We fix both estimated codes and finetune the shape embedding network f_s using the following objective,

$$\hat{f}_s = \arg\min_{f_s} \mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}}), \tag{7}$$

where $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\psi}}$ are obtained as described in the basic inference. Note that our proposed model architecture is the key for TTA. The separation of shape embedding network and articulation network ensures the disentanglement is maintained when the shape embedding network is finetuned. After adaptation, the generation ability (Secion 3.4) is still well preserved. Moreover, in practice, one can easily revert to the non-updated model before adapting to individual instance.

To this end, the proposed full Test-Time Adaptation Inference Algorithm is summarized in Algorithm 1. Since back-propagation is very efficient, the iterative optimization only introduces negligible additional computation.

Algorithm 1 Test-Time Adaptation Inference Algorithm

Input: Target shape $\mathcal{X}$.
Output: shape code $\hat{\boldsymbol{\phi}}$; articulation code $\hat{\boldsymbol{\psi}}$, updated shape embedding network $\hat{f}_s$.

1: Init $\boldsymbol{\phi} \sim \mathcal{N}(0, \sigma), \boldsymbol{\psi}$
2: $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\phi}, \boldsymbol{\psi}} \mathcal{L}(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi})$
3: ▷ *Articulation Estimation*
4: Re-init $\hat{\boldsymbol{\phi}} \sim \mathcal{N}(0, \sigma)$
5: $\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \mathcal{L}(\mathcal{X}, \boldsymbol{\phi}, \hat{\boldsymbol{\psi}})$ ▷ *Shape Code Estimation*
6: $\hat{f}_s = \arg\min_{f_s} \mathcal{L}(\mathcal{X}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ ▷ *Test-time Adaptation*

3.4. Articulated Shape Synthesis

A main advantage of the proposed disentangled continuous representation is that, once a shape code is inferred, it can be applied to synthesize shapes of unseen instances with unseen joint angles, by simply varying the articulation code. This is shown in Figure 2, Generation section. In this stage, the shape code and finetuned shape embedding network f_s obtained in the inference stage is fixed and new shapes are generated by simply inputting new joint angles to the network. If trained with part labels, the model can also output part labels for unseen shapes, as shown in Figure 6.

4. Experiment

We experiment with 3D reconstruction and demonstrate that the proposed method can successfully interpolate, extrapolate the articulation space, and synthesize new shapes. We also quantitatively show the proposed method generalize well to partial point clouds obtained from both synthetic and real depth images.

4.1. Experiment Setup

For all experiments, the mesh models used are from the Shape2Motion dataset [81]. More details about the network design and training are discussed in the supplementary materials. To evaluate the quality of generated shapes, Chamfer-L1 distance is used as the main evaluation metric, which is defined as the mean of an accuracy and a completeness metric. Following DeepSDF [58], 30,000 points are sampled for each shape for evaluation and Chamfer-L1 distances shown in the paper are multiplied by 1,000. For joint angle estimation, we follow [40] to evaluate the average joint angle error in degrees for each joint.

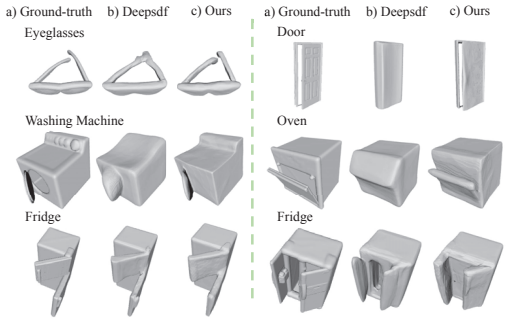

Figure 3: Reconstruction of test instances. The proposed method encodes better details whereas DeepSDF produces over-smoothed surfaces. Note that we model fridges with two different kinds of door types using a single network.

	Laptop	Stapler	Wash	Door	Oven	Eyeglasses	Fridge
DeepSDF [58]	0.35	3.73	4.29	0.61	5.33	1.63	0.80
Ours (w/o TTA)	0.15	3.39	2.60	0.21	3.72	1.25	0.90
Ours	**0.13**	**2.55**	**2.13**	**0.17**	**1.83**	**1.16**	**0.69**

Table 1: Chamfer-L1 distance comparison for reconstruction. The proposed method yields smaller Chamfer-L1 distance.

	Laptop	Stapler	Wash	Door	Oven	Eyeglasses	Fridge
DeepSDF [58]	2.77	8.69	8.04	7.79	11.13	3.33	1.74
Ours (w/o TTA)	**0.26**	**3.75**	**5.72**	**0.51**	**4.10**	**2.63**	**0.99**

Table 2: Chamfer-L1 distance comparison for interpolation.

4.2. Reconstruction

We task our model with 3D reconstruction on the held-out data. For all methods, latent codes are first inferred using back-propagation given the observed shape, and then SDF values of sampled points are predicted by a forward pass. Quantitative evaluation shows the proposed method yields better results compared to DeepSDF for all classes, as shown in Table 1.

Even though the proposed shape representation is more compact and represented with less parameters, quantitative results suggest that it achieves better performance. As visualized in Figure 3, our results show rich details compared to the over-smoothed results produced DeepSDF. We conjecture that, by introducing the articulation code and the shape code sharing, the proposed model can take advantage of multiple shapes articulated from the same instance to learn better shape priors. In addition, applying Test-Time Adaption helps the model fit the observation better. In Figure 3, we show that one single network is capable of learning priors for fridges with different in-class articulation patterns (door types).

4.3. Interpolation and Extrapolation

The ability to interpolate and extrapolate between shapes is important for a good 3D articulated object representation.

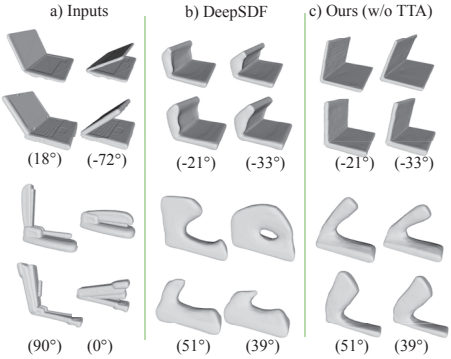

Figure 4: Comparison for interpolation. Each row shows a different object. *Inputs*, *DeepSDF* , and *Ours (w/o TTA)* results are shown from left to right respectively.

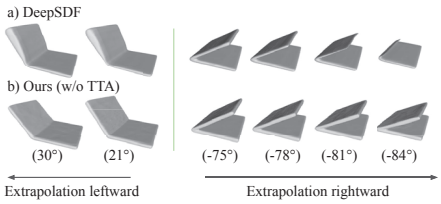

Figure 5: Comparison for extrapolation. Each row is corresponding to the same instance extrapolated in two directions. The proposed method successfully generates shapes with joint angles beyond the range seen during training.

In this section, given two articulated shapes of the same instance, we ask the models to output shapes with articulations in between or beyond.

As shown in Table 2, we quantitatively demonstrate that the proposed model can reliably interpolate shapes. In both cases, the proposed method outperforms DeepSDF by a large margin. As visualized in Figure 4, the baseline generates unrealistic deformed shapes whereas the proposed method produces accurate shape estimations.

In the extrapolation setting on laptops, the proposed method also significantly outperforms DeepSDF. We compute the Chamfer-L1 distance for both methods. The proposed method achieves 0.280 whereas DeepSDF only gets 3.396. As illustrated in Figure 5, the proposed method is capable of extrapolating beyond the angle range seen during training while DeepSDF fails to produce valid shapes.

The insight is that, even in cases where the two given shapes (the same instance) are with large articulation difference, the inferred shape codes of the proposed method are still close in the high dimensional space. The difference between these shapes is mainly captured by the articulation codes. Therefore, the linear interpolation in the articulation space still produces valid shapes. However, the baseline method learns a non-structured shape space and the interpolation result could be a random point in a high dimensional

	Laptop	Stapler	Washing	Door	Oven	Eyeglasses	Fridge
DeepSDF [58] *(Interpolation)*	2.77	8.69	8.04	7.79	11.13	3.33	1.74
Ours (w/o TTA)	0.39 (1.39)	3.77 (3.30)	**2.86 (7.10)**	0.73 (1.09)	3.77 (7.08)	2.48 (2.58)	0.97 (3.47)
Ours	**0.32 (1.59)**	**3.25 (3.53)**	3.01 (8.44)	**0.53 (0.95)**	**2.58 (6.79)**	**2.42 (2.84)**	**0.86 (4.19)**
Ours (w/o TTA) + part label	0.32 (1.45)	3.08 (3.66)	2.16 (2.66)	0.38 (1.04)	5.19 (3.20)	**2.03 (2.12)**	0.85 (3.69)
Ours + part label	**0.29 (1.48)**	**2.48 (3.34)**	**1.96 (2.03)**	**0.33 (1.67)**	**3.10 (2.98)**	2.16 (2.18)	**0.64 (2.98)**

Table 3: Chamfer-L1 distance comparison for shape synthesis. Joint angle estimation errors of the proposed method in brackets (·).

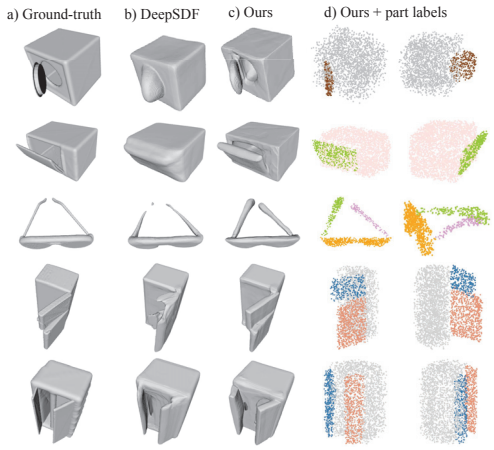

Figure 6: Shape synthesis and part prediction. From left to right: the ground-truth, *DeepSDF interpolation*, *Ours generation*, and the *Ours + part labels* part prediction results.

space without corresponding to any meaningful shape.

For the proposed method, given two shapes, two set of codes are first inferred using the procedure shown in Section 3.3 but only articulation codes are interpolated to generate the new shape. For the baseline, the target shape code is simply computed as a linear combination of the two obtained codes.

4.4. Shape Synthesis and Part Prediction

One main advantage of our learned disentangled representation is its generation ability as described in Section 3.4. One can easily control the articulation input to generate shapes of unseen instances with unseen joint angles.

To provide comparisons, we employ the DeepSDF Interpolation results described in Section 4.3 as baseline. Note that this is not a fair comparison as our method requires only one shape instead of two as for the baseline. Though relying on less information, the proposed method still yields much better results as shown in Table 3. We demonstrate that applying Test-Time Adaption reduces the error further, indicating that Test-Time Adaption helps with inferring better shape while maintaining a disentangled representation. As visualized in Figure 6, note that one single network is capable of synthesize fridges with different articulation patterns.

One additional advantage of the proposed method is that joint angles can be estimated simultaneously. We quantitatively evaluate joint angle prediction errors in degrees, as shown in brackets in Table 3. Results suggest that the proposed model can predicts joint angles accurately during the inference stage.

We also demonstrate that, if provided, part labels can further boost the performance. Models trained with part labels are denoted as *Ours + part labels*. Part label prediction results are visualized in Figure 6.

		Laptop		Door		Washing	
		Recon	Gen	Recon	Gen	Recon	Gen
2-view	DeepSDF [58]	2.40	4.00	3.34	16.63	14.14	11.38
	Ours (w/o TTA)	0.75	1.16	0.62	0.83	7.63	10.03
	Ours	**0.61**	**1.07**	**0.61**	**0.78**	**5.31**	**9.39**
1-view	DeepSDF [58]	3.31	6.75	5.76	20.24	14.08	13.05
	Ours (w/o TTA)	**2.19**	**1.25**	**0.80**	**0.89**	9.58	11.31
	Ours	2.25	1.55	0.86	1.38	**6.29**	**9.52**

Table 4: Chamfer-L1 distance comparison on partial point clouds. 1/2-view distinguishes the setting whether the partial point clouds are generated from one or two depth images.

4.5. Test on Partial Point Clouds

We task the models with shape completion and generation on the partial point clouds from depth observations, which aims to inferring the shape code that best matches the partial point clouds and then generating to unseen articulations. We quantitatively demonstrate the proposed method outperforms the baseline significantly in Table 4. Note that models are not trained on partial point clouds.

As 3D meshes are not provided in this scenario, we sample two points for each depth observation following DeepSDF [58]. We approximate the SDF values to be η and $-\eta$ by perturbing each of them η distance away from the observed depth point along the computed surface normal direction. η is set to be 0.025 in our experiments.

As discussed in Section 4.4, we employ the interpolation results of DeepSDF as a baseline for shape generation. We test the model on the point clouds generated from single and multi-view depth images. As the number of views increases, we observe that applying Test-Time Adaption yields larger performance gain. As visualized in Fig 7, the proposed method encodes stronger shape prior and recovers shapes reliably given the partial point clouds input.

4.6. Test on Real-world Depth Images

We quantitatively show the proposed method generalizes better on real-world depth images, as shown in Table 5. The

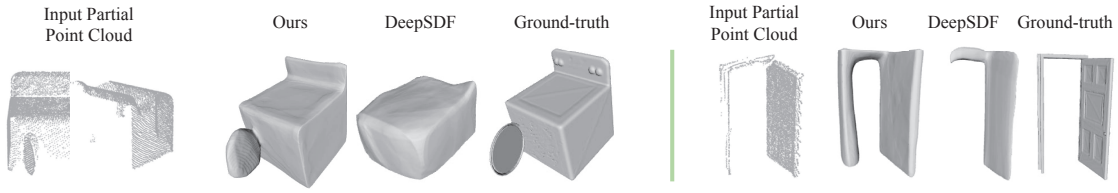

Figure 7: Partial point clouds completion one-view setting. For a given depth image visualized as point clouds, we show a comparison of shape completion from our method against DeepSDF and Ground-truth.

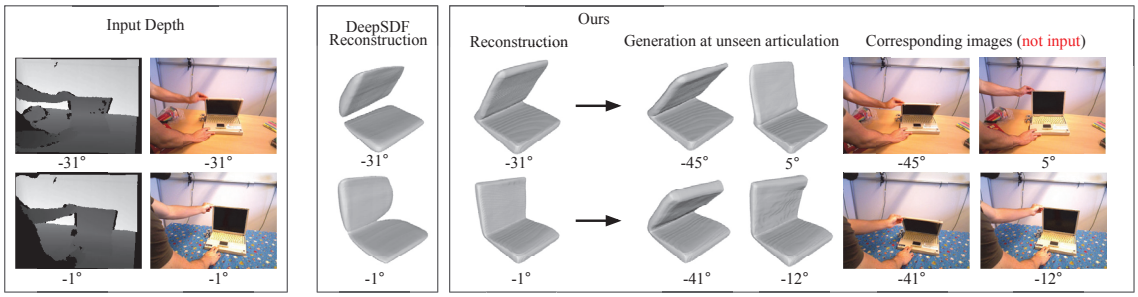

Figure 8: Test on real-world depth images. From left to right: *Input depth*, *DeepSDF reconstruction*, *Ours reconstruction and generation*. Note that the *Ours generation* are generated by only changing the articulation code. The shape code is inferred from the input depth of a laptop at a different articulation. RGB images and joint angles shown are only for visualization purposes and are not input to the model.

	Reconstruction	Generation
DeepSDF [58]	4.65	-
Ours (w/o TTA)	2.53	5.09
Ours	**0.76**	**3.22**

Table 5: Chamfer-L1 distance comparison on real-world depth images. The Chamfer-L1 distance here is from ground-truth depth to reconstructed shape. DeepSDF is not able to generate new shapes.

RBO dataset [45] is a collection of 358 RGB-D video sequences of humans manipulating articulated objects, with the ground-truth poses of the rigid parts annotated by a motion capture system. We take laptop depth images from different sequences in the dataset and crop laptops from depth images by applying Mask R-CNN [21] on the corresponding rgb images. We generate corresponding point clouds from real depth images following Section 4.5, and then exploit the ground-truth pose to align the point clouds to the canonical space defined by Shape2motion dataset [81].

In Table 5, we show both reconstruction and generation results. Note both models are not trained on real-world depth images. Given a real-world depth image, we obtain its corresponding point clouds, input it to the model trained on synthetic data to reconstruct its 3D shape, and evaluate the reconstruction performance as the one-way Chamfer-L1 distance from ground-truth depth to reconstructed shape. Next, we take the shape code from the previous reconstructed shape and change the articulation code to output shapes at multiple unseen articulation. We take the real depth images at these new articulation and use the generated corresponding point clouds as the ground-truth to evaluate the generation performance. As visualized in Fig 8, the proposed model reliably synthesize shapes at unseen articulation whereas DeepSDF does not have the ability to generate shapes. Table 5 results suggest that applying Test-Time Adaption reduces the error further on both reconstruction and generation.

5. Conclusions

We propose Articulated Signed Distance Functions (A-SDF) to model articulated objects with a structured latent space. A Test-Time Adaptation inference algorithm is introduced to infer shape and articulation simultaneously. We experiment on seven articulated object categories from the shape2motion dataset [81], and demonstrated improved shape reconstruction, interpolation, and extrapolation performance. Moreover, the method allows for controlling the articulation code to generate shapes for unseen instances with unseen joint angles. We also go beyond synthetic data and demonstrate the proposed method can reliably generate 3D shapes from real-world depth images in the rbo dataset [45].

Acknowledgements. This work was supported, in part, by grants from DARPA LwLL, iARPA (DIVA) D17PC00342, NSF IIS-1924937, NSF 1730158 CI-New: Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI), NSF ACI-1541349 CC*DNI Pacific Research Platform, and gifts from Qualcomm, TuSimple and Picsart.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *ICLR*, 2018. 2

[2] Ivan Anokhin, Pavel Solovev, Denis Korzhenkov, Alexey Kharlamov, Taras Khakhulin, Aleksei Silvestrov, Sergey Nikolenko, Victor Lempitsky, and Gleb Sterkin. High-resolution daytime translation without domain labels. In *CVPR*, pages 7488–7497, 2020. 3

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 38(4), 2019. 3

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *ICCV*, pages 5419–5429, 2019. 2

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016. 2

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. In *ICCV*, pages 5932–5941, 2019. 3

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2172–2180, 2016. 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, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *CVPR*, pages 42–51, 2020. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. 2

[11] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey E. Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *CVPR*, pages 31–41, 2020. 2

[12] Boyang Deng, John P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey E. Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA neural articulated shape approximation. In *ECCV*, pages 612–628, 2020. 2

[13] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *NeurIPS*, pages 7433–7443, 2019. 2

[14] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *ICRA*, pages 7221–7227, 2019. 2

[15] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 2463–2471, 2017. 2

[16] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single RGB image. In *CVPR*, pages 10833–10842, 2019. 2

[17] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A. Funkhouser. Deep structured implicit functions. *CoRR*, abs/1912.06126, 2019. 2

[18] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A. Funkhouser. Local deep implicit functions for 3d shape. In *CVPR*, pages 4856–4865, 2020. 2

[19] Georgia Gkioxari, Justin Johnson, and Jitendra Malik. Mesh R-CNN. In *ICCV*, pages 9784–9794, 2019. 2

[20] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S. Sukhatme. Active articulation model estimation through interactive perception. In *ICRA*, pages 3305–3312, 2015. 2

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 8

[22] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2016. 3

[23] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 3

[24] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 3

[25] Vidit Jain and Erik G. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, pages 577–584, 2011. 3

[26] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. Local implicit grid representations for 3d scenes. In *CVPR*, pages 6000–6009, 2020. 2

[27] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. *arXiv preprint arXiv:2104.03304*, 2021. 3

[28] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 2

[29] Angjoo Kanazawa, Shahar Kovalsky, Ronen Basri, and David Jacobs. Learning 3d deformation of animals from 2d images. In *Computer Graphics Forum*, volume 35, pages 365–374. Wiley Online Library, 2016. 2

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 3

[31] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *ICRA*, pages 272–277, 2008. 2

[32] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277, 2008. 3

[33] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018. 3

[34] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: video inference for human body pose and shape estimation. In *CVPR*, pages 5252–5262, 2020. 2

[35] Amit P. S. Kohli, Vincent Sitzmann, and Gordon Wetzstein. Inferring semantic information with 3d neural scene representations. *CoRR*, abs/2003.12673, 2020. 2

[36] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 452–461, 2020. 2

[37] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *NeurIPS*, pages 2539–2547, 2015. 3

[38] 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):194:1–194:17, 2017. 2

[39] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *NeurIPS*, 2020. 3

[40] Xiaolong Li, He Wang, Li Yi, Leonidas J. Guibas, A. Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, pages 3703–3712, 2020. 1, 5

[41] Yiyi Liao, Simon Donné, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *CVPR*, pages 2916–2925, 2018. 2

[42] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros, and Noah Snavely. Learning to factorize and relight a city. In *ECCV*, 2020. 3

[43] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 2

[44] Roberto Martin Martin, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *ICRA*, pages 5091–5097, 2016. 2

[45] Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The RBO dataset of articulated objects and interactions. *Int. J. Robotics Res.*, 38(9), 2019. 2, 8

[46] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2, 3

[47] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, pages 181.1–181.11. BMVA Press, 2015. 2

[48] 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*, pages 405–421, 2020. 2

[49] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. *arXiv preprint arXiv:2103.10534*, 2021. 3

[50] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. *arXiv preprint arXiv:2101.02692*, 2021. 3

[51] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. *ICCV*, Oct 2019. 3

[52] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, pages 7587–7596, 2019. 3

[53] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, pages 5378–5388, 2019. 2

[54] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3501–3512, 2020. 2

[55] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. *arXiv preprint arXiv: 2104.03110*, 2021. 2

[56] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, pages 484–494, 2018. 2

[57] Pablo R. Palafox, Aljaz Bozic, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. *arXiv preprint arXiv:2104.00702*, 2021. 2

[58] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[59] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *NeurIPS*, 33, 2020. 3

[60] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *arXiv preprint arXiv: 2012.15838*, 2021. 2

[61] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *CVPR*, pages 14104–14113, 2020. 3

[62] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *ECCV*, pages 725–741, 2018. 2

[63] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, pages 6620–6629, 2017. 2

[64] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pages 2304–2314, 2019. 1, 2, 3

[65] Shunsuke Saito, Tomas Simon, Jason M. Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 81–90, 2020. 1

[66] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J. Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *CVPR*, pages 7763–7772, 2019. 2

[67] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. *CoRR*, abs/2007.02442, 2020. 3

[68] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, pages 9243–9252, 2020. 3

[69] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and remapping the "dna" of a natural image, 2018. 3

[70] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, pages 3118–3126. IEEE Computer Society, 2018. 3

[71] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *CoRR*, abs/2006.09661, 2020. 2

[72] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS 2019*, pages 1119–1130, 2019. 1, 2

[73] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pages 9229–9248, 2020. 3

[74] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *CVPR*, pages 1493–1502, 2017. 2

[75] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In *ECCV*, pages 293–309, 2020. 2

[76] Alex Trevithick and Bo Yang. GRF: learning a general radiance field for 3d scene representation and rendering. *CoRR*, abs/2010.04595, 2020. 2

[77] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, pages 1466–1474, 2017. 2

[78] Gül Varol, Duygu Ceylan, Bryan C. Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *ECCV*, pages 20–38, 2018. 2

[79] Jiashun Wang, Chao Wen, Yanwei Fu, Haitao Lin, Tianyun Zou, Xiangyang Xue, and Yinda Zhang. Neural pose transfer by spatially adaptive instance normalization. In *CVPR*, pages 5830–5838, 2020. 2

[80] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. In *ECCV*, pages 55–71, 2018. 2

[81] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *CVPR*, pages 8876–8884, 2019. 5, 8

[82] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. *arXiv preprint arXiv:2104.03437*, 2021. 1, 2

[83] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. Sapien: A simulated part-based interactive environment. In *CVPR*, June 2020. 3

[84] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. DISN: deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, pages 490–500, 2019. 2

[85] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, volume 12357 of *Lecture Notes in Computer Science*, pages 34–51, 2020. 2

[86] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *ICCV*, pages 7738–7748, 2019. 2

[87] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3d meshes. In *ECCV*, volume 12367 of *Lecture Notes in Computer Science*, pages 341–357, 2020. 3

[88] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In *NeurIPS*, pages 118–129, 2018. 3

[89] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *ICCV*, pages 900–909, 2017. 2

[90] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *ICCV*, pages 5359–5368, 2019. 2

[91] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *CVPR*, pages 3955–3963, 2018. 2

[92] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *CVPR*, pages 5524–5532, 2017. 2