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

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.

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 [19, 45]. 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 which require 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 [30]. While implicit functions have recently been widely applied in modeling static object shape with fine details [34, 35, 38], much less effort has been devoted to modeling general articulated objects. We observe that models with a single shape code input, such as DeepSDF [30], 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 encoder 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 enable 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. 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.

2. Related Work

Neural Shape Representation. A large body of work [43, 10, 4, 6, 20, 33] has focused on investigating efficient and accurate 3D object representations. Recent advances suggest that representing 3D objects as continuous and differentiable implicit functions [9, 30, 23, 5, 16] can model various topologies in a memory-efficient way. Most of these work is limited to modeling static objects and scenes [9, 12, 46, 38, 26, 34, 24, 37, 41]. 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.

Articulated Humans. One line of work leverages parametric mesh models [21, 18, 51, 2] to estimate shape and articulation for faces [39, 32, 36], hands[8], humans bodies [31, 1, 48, 15, 13, 28, 47, 42], and animals [50, 14, 17, 49] by directly inferring shape and articulation parameters. However, such parametric models requires substantial efforts from experts to construct and thus is hard to generalize to large-scale object categories. To address the challenge, another line of work [29, 25, 40, 7, 3, 27] employs neural networks to learn shapes from data. In comparison, our method is category-level on general articulated objects and we assume no part label.

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.

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 1, an Articulated Signed Distance Function f_θ is finally defined with the auto-decoder architecture, which is composed of a shape encoder f_s and an articulation network f_a,

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

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.

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 1. The shape code is first concatenated with a sampled point $\boldsymbol{x}$ to form vector of dimension $C + 3$ and input to the shape encoder. Then the articulation network takes the shape embedding and articulation code to predict the SDF value for the input 3D point. 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,

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

where $\boldsymbol{x}_k \in \mathcal{X}$ is a point of instance $\mathcal{X}$, s_k the corresponding ground-truth SDF value, and $k \in \{1, \dots, K\}$. When the object part labels are available, we include a complementary auxiliary part classification loss using cross-entropy.

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

$$\mathcal{L}(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \mathcal{L}^s(\mathcal{X}, \boldsymbol{\phi}, \boldsymbol{\psi}) + \lambda_{\boldsymbol{\phi}} ||\boldsymbol{\phi}||_2^2. \quad (3)$$

Following [30], we include a zero-mean multivariate-Gaussian prior per shape latent code $\boldsymbol{\phi}$ to facilitate learning a continuous shape manifold.

At training time, the shape codes are randomly initialized with a Gaussian distribution at the very beginning of training. 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,

$$\arg\min_{\theta, \boldsymbol{\phi}_n} \sum_{n=1}^{N} \sum_{m=1}^{M} \mathcal{L}(\mathcal{X}_{n,m}, \boldsymbol{\phi}_n, \boldsymbol{\psi}_m), \quad (4)$$

where θ is the network parameters.

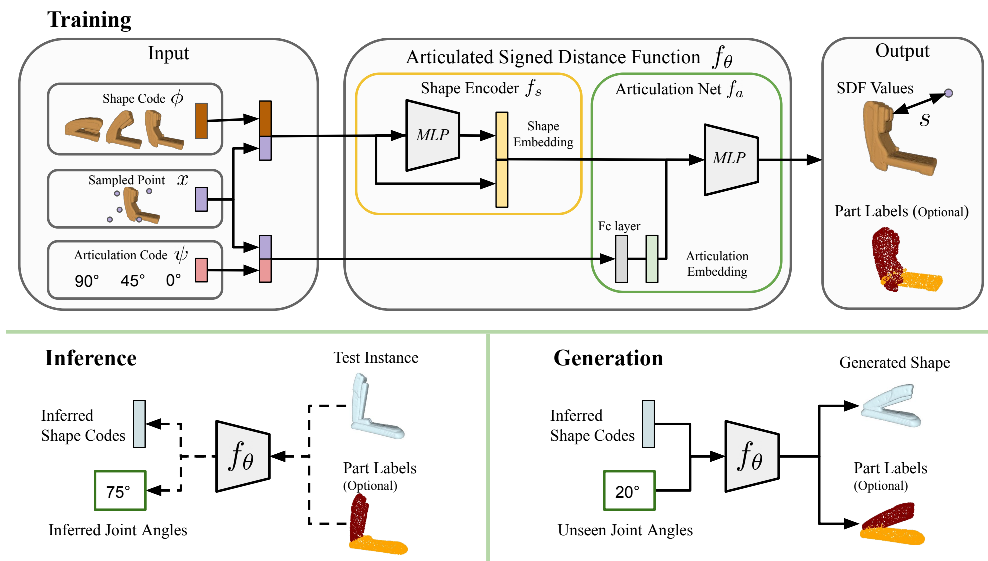

Figure 1: Overview of the proposed method. At training time, the articulation code and randomly sampled shape codes are first concatenated with a sampled point separately. The produced embeddings are then input to an articulated signed distance function f_θ to regress SDF values (signed distance) and predict part labels (optional). Note that the same instance is associated with one shape code regardless of its articulation state. 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.

3.3. Inference

Basic Inference. In the inference stage, illustrated in the Inference Section of Figure 1, an instance $\mathcal{X}$ is given and the goal is to recover the corresponding shape code ϕ and the articulation code ψ. 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,

$$\hat{\phi}, \hat{\psi} = \arg\min_{\phi, \psi} \mathcal{L}(\mathcal{X}, \phi, \psi). \tag{5}$$

We first use Equation 5 to optimize both shape and articulation codes as our initial estimation. So the estimated articulation code $\hat{\psi}$ 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 $\hat{\psi}$, and the optimization is only solved for the shape code $\hat{\phi}$.

Test-Time Adaptation Inference. To generalize better to out-of-distribution data, the Test-Time Adaptation (TTA) for shape encoder f_s is further introduced. It is built on the basic inference procedure with the estimated shape code $\hat{\phi}$ and articulation code $\hat{\psi}$. We fix both estimated codes and finetune the shape encoder f_s using the following objective,

$$\hat{f}_s = \arg\min_{f_s} \mathcal{L}(\mathcal{X}, \hat{\phi}, \hat{\psi}), \tag{6}$$

where $\hat{\phi}$ and $\hat{\psi}$ are obtained as described in the basic inference. Note that our proposed model architecture is the key for TTA. The separation of shape encoder and articulation network ensures the disentanglement is maintained when the shape encoder is finetuned.

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 1, Generation section. In this stage, the shape code and finetuned shape encoder f_s obtained in the inference stage is fixed and new shapes are generated by simply inputting new joint angles to the network.

4. Experiment

4.1. Datasets

For all experiments, the mesh models used are from the Shape2Motion dataset [44]. Shape2Motion is a large scale 3D articulated object dataset containing 2,440 instances. We select seven categories with sufficient number of instances per category, which are laptop, stapler, washing machine, door, oven, eyeglasses, and fridges.

4.2. Shape Synthesis and Part Prediction

One main advantage of our learned disentangled representation is its generation ability. We can easily control the articulation input to generate corresponding shapes of unseen instances with unseen joint angles. In this section, we

	Laptop	Stapler	Washing	Door	Oven	Eyeglasses	Fridge
DeepSDF [30] *(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 1: Chamfer-L1 distance comparison for shape synthesis. Joint angle estimation errors of the proposed method in brackets (·).

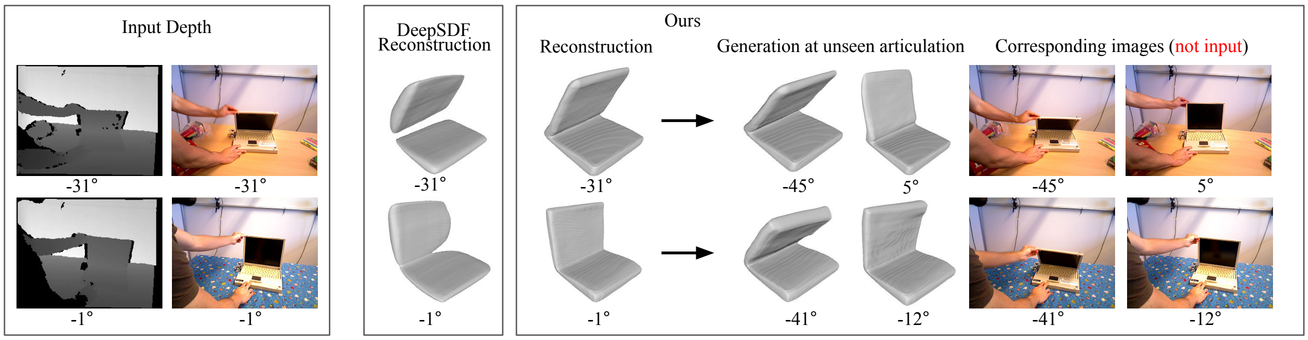

Figure 2: Test on real-world depth images. From left to right: *Input depth*, *DeepSDF reconstruction*, *Ours reconstruction and generation*. Note that the *Ours generation* are generated by only changing the articulation code. The shape code is inferred from the input depth of a laptop at a different articulation. RGB images and joint angles shown are only for visualization purposes and are not input to the model.

study the quality of generated shapes using the proposed generation method in Section 3.4.

Since DeepSDF does not have the ability to generate shapes, to provide comparisons, we employ the DeepSDF Interpolation results as baseline. Given two shapes, the target shape code is simply computed as a linear combination of the two inferred latent codes. 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 1. 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.

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 1. 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*.

4.3. Test on Real-world Depth Images

We quantitatively show the proposed method generalizes better on real-world depth images, as shown in Table 2. The RBO dataset [22] 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 [11] on the corresponding rgb images. We generate corresponding point clouds from real depth images, and then exploit the ground-truth pose to align the point clouds to the canonical space defined by Shape2motion dataset [44].

	Reconstruction	Generation
DeepSDF [30]	4.65	-
Ours (w/o TTA)	2.53	5.09
Ours	**0.76**	**3.22**

Table 2: 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.

In Table 2, 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 2, the proposed model reliably synthesize shapes at unseen articulation whereas DeepSDF does not have the ability to generate shapes. Table 2 results suggest that applying Test-Time Adaption reduces the error further on both reconstruction and generation.

References

[1] 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

[2] 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

[3] 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

[4] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *CVPR*, pages 42–51, 2020. 1

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

[6] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey E. Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *CVPR*, pages 31–41, 2020. 1

[7] 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

[8] 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

[9] 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

[10] Georgia Gkioxari, Justin Johnson, and Jitendra Malik. Mesh R-CNN. In *ICCV*, pages 9784–9794, 2019. 1

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 4

[12] 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

[13] 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

[14] 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

[15] 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

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

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

[18] 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

[19] 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

[20] Yiyi Liao, Simon Donné, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *CVPR*, pages 2916–2925, 2018. 1

[21] 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

[22] Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The RBO dataset of articulated objects and interactions. *Int. J. Robotics Res.*, 38(9), 2019. 4

[23] 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

[24] 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

[25] 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

[26] 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

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

[28] 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

[29] 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

[30] 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, 4

[31] 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

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

[33] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, pages 6620–6629, 2017. 1

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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

[39] 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

[40] 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

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

[42] 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

[43] 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. 1

[44] 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. 3, 4

[45] 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

[46] 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

[47] 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

[48] 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

[49] 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

[50] 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

[51] 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