Supplementary Material for NPC: Neural Point Characters from Video

Figure 1: **Unseen pose rendering from MonoPerfCap [23] test split**. Similar to the unseen pose synthesis results on Human3.6M [4, 5], NPC produces sharper and more plausible details, such as shoes, knuckles, and ears.

In this document, we provide additional qualitative comparisons on MonoPerfCap [23] (Section 1) and Human3.6M [4, 5] test splits (Section 2), and show extra examples on the motion retargeting task (Section 3). We further present the initial canonical point clouds from DANBO for all subjects (Section 4). We then describe implementation details, including network architectures and auxiliary features used in NPC (Section 5), followed by a comprehensive ablation study to quantify the impact of each network feature and our tabulated K-NN search. We also include an experiment demonstrating that using SMPL surface points works just as well as DANBO initialization (Section 6). Finally, we provide a visual example for failure cases (Section 7). Our supplemental video shows animated results of our NPC characters[1].

[1]All data sourcing, modeling codes, and experiments were developed at University of British Columbia. Meta did not obtain the data/codes or conduct any experiments in this work.

1. Qualitative Comparisons on MonoPerfCap

We present unseen pose synthesis rendering on MonoPerfCap [23] test split in Figure 1, with 3D poses estimated by SPIN [6] and refined by A-NeRF [21]. Compared to the state-of-the-art method DANBO [20], NPC synthesizes sharper details like cloth wrinkles, ears, and knuckles. NPC also preserves clearer contour and textures on shoes and trousers. Notably, despite one of the feet missing in the initial point clouds (see Section 4 and Figure 4), NPC can recover the shoe to some extent, thanks to the iterative refinement of the initial point locations and the combination of the point-based representation with the continuous neural field.

2. Additional Qualitative Comparisons on Human3.6M

We provide additional unseen pose synthesis results on Human3.6M Anim-NeRF [4, 5, 16] test split in Figure 2. We show, on various subjects, that NPC captures high-frequency appearance more accurately than Neural-

Figure 2: **Unseen pose synthesis on Anim-NeRF Human3.6M [4, 5, 16] test split**. On various subjects, NPC shows better details in shoes, cloth wrinkles and body landmarks like motion capture trackers.

Body [17] and ARAH [22], without relying on template meshes and 3D scan priors.

3. Additional Motion Retargeting Examples

NPC shows improved texture and contour consistency over DANBO [20] when rendering completely out-of-distribution poses on Human3.6M and AIST++ [9] in Figure 3.

4. Canonical Point Clouds Initialization

DANBO provides good point cloud initialization despite being trained only for around half an hour, as shown in Figure 4. Note that the initialization is still noisy, and includes twisted or missing body parts (Figure 4, last row).

To extract the point clouds, we adopt the official DANBO implementation[2] and the training configuration, except for the following changes: we half the samples along each ray to 48 uniform and 24 importance samples, and train DANBO only for 10k iterations. We use matching cube [11] with grid resolution $512 \times 512 \times 512$ to extract the T-pose point clouds. NPC uses 3800 surface points to represent each subject from both Human3.6M and MonoPerfCap. Our surface points are sparser than the commonly used SMPL mesh [2, 10], which contains 6890 vertices.

[2]Github repository: https://github.com/LemonATsu/DANBO-pytorch

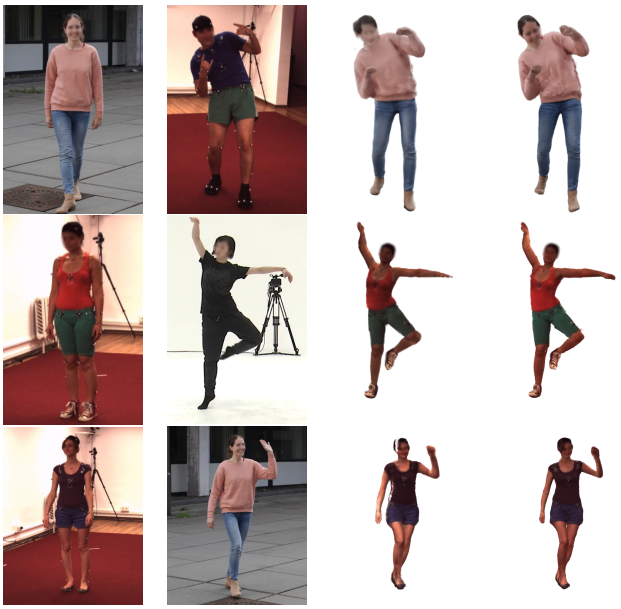


Figure 3: **Motion retargeting from out-of-distribution poses on various subjects**. NPC retains better appearance consistency and texture details.

5. Implementation details

NPC is supervised using only the frames from the training videos, along with 4 regularization terms that helps improve the model quality:

$$L = L_p + \lambda_{eik} L_{eik} + \lambda_{\Delta p} L_{\Delta p} + \lambda_N L_N + \lambda_S L_S, \quad (1)$$

with $\lambda_{eik} = 0.01$, $\lambda_{\Delta p} = 1.0$, $\lambda_N = 10.0$, and $\lambda_S = 0.1$. For both L_{eik} and L_s, we construct $\tilde{\mathbf{p}}$ by randomly selecting 100 points from each body part for the loss computation.

5.1. Training configurations

NPC hyperparameters. For all datasets, we train our model for 150k iterations with a learning rate 0.0005 that decays to 0.0001 in 500k iterations. We form each training batch by sampling 4096 images from 16 images, with 64 uniform and 32 importance samples along each ray. We use $K = 8$ for Human3.6M [4, 5] and MonoPerfcap [23]. On ZJU-Mocap [3, 17], we set $K = 16$ to handle the long-range deformation dependency, and adopt per-timestep encodings following ARAH to capture temporal deformations beyond body poses. We train NPC for 150k iterations. The per-point learnable influence scale β_j in the main paper Eq (4) is initialized to 0.0005.

5.2. Auxiliary features

Below, we describe auxiliary features that we adopt in our design.

Per-frame features. Like prior work [12, 16, 17, 20, 21], we use per-frame codes $f(t)$ to model varying illumination that cannot be learned without illumination information available. Note that we also add $f(t)$ to our TAVA baseline [8], which improves its results on sequences with illumination changes. We notice that their proposed ambient occlusion method cannot handle Human3.6M [4, 5] dataset properly as multiple light sources are presented in the scene.

Geometric features. We supply F_ψ with two geometric features. We encode the first feature f_i^v, the view direction, as the dot product to the bone-to-surface-point vector $\mathbf{b}_{B \to \mathbf{p}_i^o}$. The second geometric feature is the projection of $\mathbf{q}^o$ onto the bone-to-surface-point vector, $r_i = \mathbf{q}^o \cdot \mathbf{b}_{B \to p_i^o}$. We concatenate the two features and weighted sum the features via

$$g(\mathbf{q}) = \left(\sum_{i=1}^{K} \frac{a_i}{\sum_{j=1}^{K} a_j} \cdot [r_i, f_i^v] \right). \quad (2)$$

We conduct ablation studies on both r_i and f_i^v, and report the result in Section 6.

5.3. Network Architecture

Linear blend skinning MLP. Following [16], we employ a small (3-layers with 32 hidden units per layer) coordinate MLP to predict the final LBS weights. We introduce one $\text{MLP}_{\mathbf{w},j}$ for each part j,

$$\mathbf{w}_{i,j} = \text{MLP}_{\mathbf{w},j}(\mathbf{p}_{i,j}) + \mathbf{w}_{i,j}^0, \quad (3)$$

with $\mathbf{p}_{i,j}$ the i-th surface point belonging to part j, and $\mathbf{w}_{i,j}^0$ the initial LBS weight. Figure 5(a) depicts the network structure of MLP_w. The use of coordinate MLP implicitly regularizes the LBS weights spatially.

GNN-FiLM. Our GNN-FiLM consists of two parts, as illustrated in Figure 5(b). The first part is a 4-layer GNN with 128 hidden units that takes as input the body pose θ, to output the FiLM conditional vectors [18] for each body part j,

$$\alpha_j, \gamma_j = \text{GNN}(\theta). \quad (4)$$

The second part is a 4-layer per-part coordinate MLP that takes as input $\mathbf{p}_{i,j}$ to predict f^θ and Δp. α_j and γ_j scale and shift the first layer output feature z. The modulated features are subsequently forwarded to the rest of the network to produce pose-conditioned outputs f^θ and Δp.

NeRF F_ψ. Our NeRF network F_ψ follows the standard design of NeRF [14], with reduced numbers of linear layers. Precisely, we use only 3 layers, instead of 8 in the original NeRF and DANBO, to extract the feature shared between the radiance and geometry branch. We show the architecture in Figure 5(c).

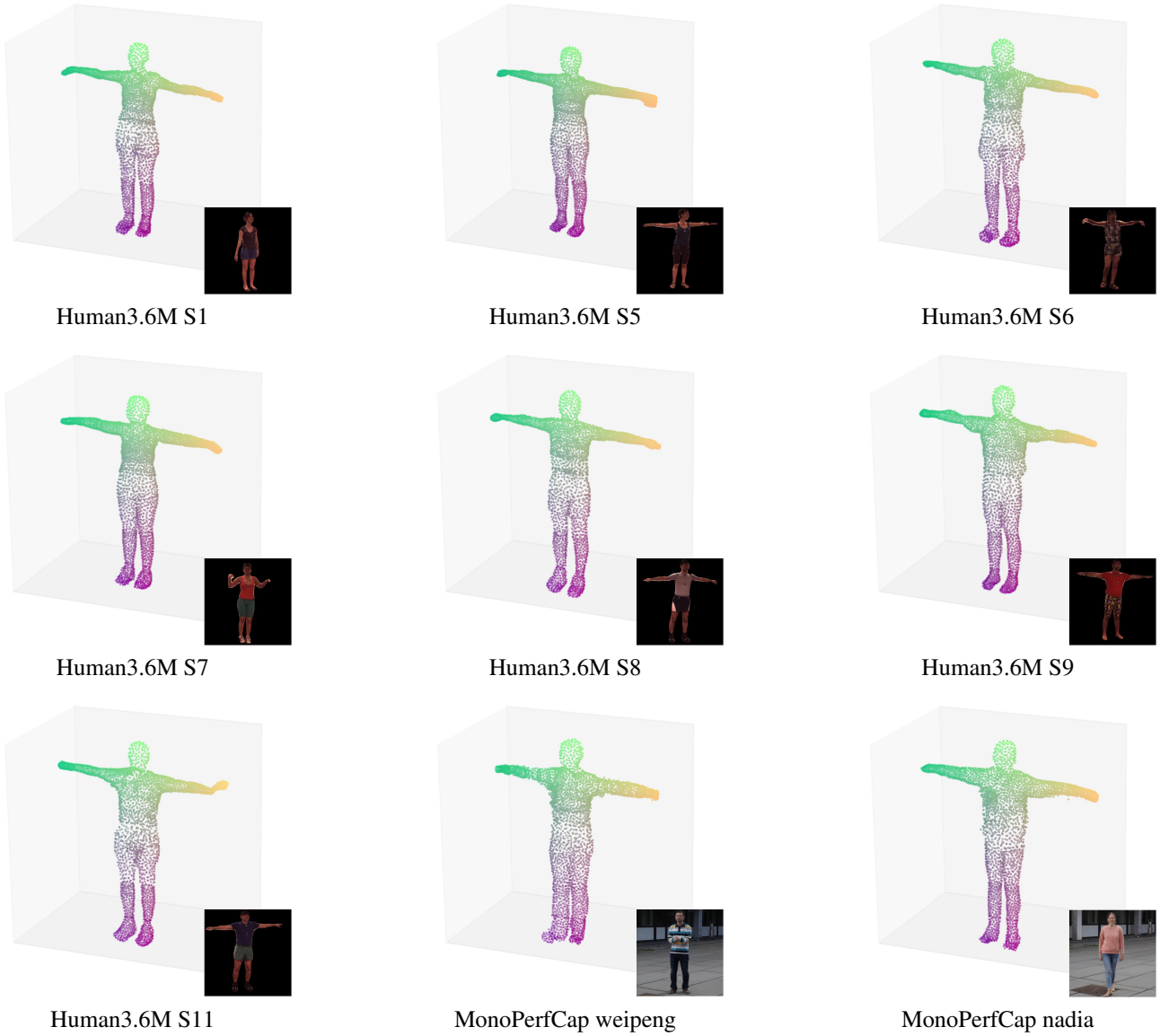


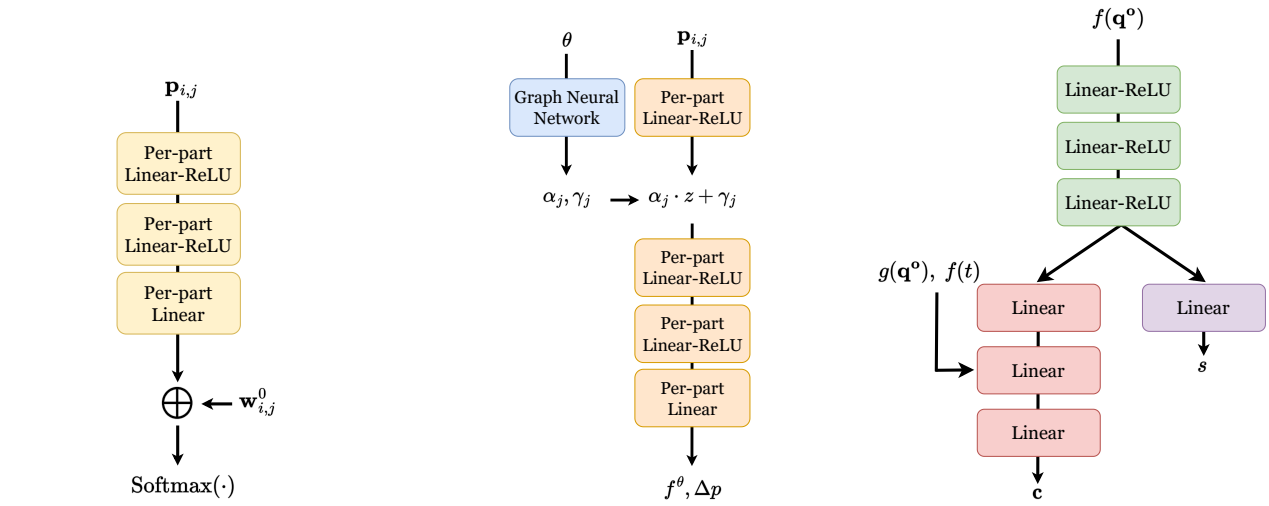
Figure 4: **Initial canonical point clouds extracted using DANBO [20].** Overall, DANBO can capture the rough geometry of the target character in about half an hour. These points serve as a good starting point for NPC to build details appearance on top.

Positional Encoding. We apply positional encoding to help the network learn high-frequency details, similar to all other neural field-based methods [8, 13, 16, 17, 20, 21, 22]. Specifically, we apply positional encoding with 5 frequency bands to the input pose θ for GNN-FiLM, similar to DANBO [20]. Additionally, we use relative spatial positional encoding proposed in KeypointNeRF [13] to encode f^v and (f^p, f^s) with 2 and 6 frequency bands respectively.

Our final network has a total of 2.7M learnable parameters, which is only 0.2M more parameters than one of the state-of-the-art methods DANBO. NPC also requires less training iterations than DANBO (150k v.s. 200k iterations).

5.4. Inference Time

NPC takes around 7 seconds to render a 1000×1000 image using a single NVIDIA V100 GPU, roughly matching the rendering speed of DANBO. On the other hand, TAVA [8] requires over 4 minutes in the same setting due to the computationally heavy root-finding algorithm: it takes 0.26 seconds for the root for 10k 3D queries. This corresponds to about 60-70 rays/pixels in the batch.



(a) Linear blend skinning network $\text{MLP}_{\mathbf{w},j}$ (b) GNN-FiLM (c) NeRF F_ψ

Figure 5: **Network architectures.** (a) We use a small MLP for each body part to predict the residual of the initial LBS weights $\mathbf{w}^0_{i,j}$, producing the final weight. (b) Our GNN predicts conditional vectors local to each body part for modulating the pose dependent outputs. (c) Our NeRF F_ψ is modified from the original architecture [14], with less shared layer (green). Our character-specific features $f(\mathbf{q}^o), g(\mathbf{q}^o)$, and $f(t)$ are introduced at two different layers to maintain view direction invariance for predicting s.

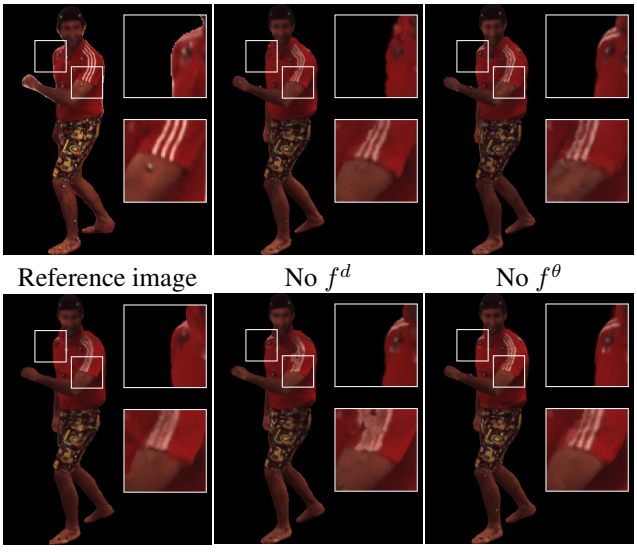


Figure 6: **Ablation study on unseen poses.** Our proposed features f^d, f^θ and f^p enables NPC to achieve better overall perceptual quality, with less noises and improved consistency on the texture.

Table 1: **Ablation study on each of our proposed designs.** Most of our features contribute to the final perceptual quality.

	KIDx100 ↓	LPIPS (VGG) ↓	LPIPS (Alex) ↓
No f^d	5.54	0.122	0.132
No f^p	4.51	0.122	0.131
No f^θ	4.50	0.120	0.132
No Δp	4.61	0.120	0.131
No f^v	4.48	0.115	**0.124**
No r_i	**3.87**	0.116	0.125
Ours full	4.43	**0.115**	**0.124**

Table 2: **Ablation study on running with 4 different canonical point clouds initializations**. We report the standard deviations in the parenthesis. The indicates that our point initialization strategy is reliable, and NPC behaves consistently across different training runs.

S9			WP		
KIDx100 ↓	LPIPS (VGG) ↓	LPIPS (Alex) ↓	KIDx100 ↓	LPIPS (VGG) ↓	LPIPS (Alex) ↓
4.34 (± 0.12)	0.116 (± 0.000)	0.124 (± 0.000)	3.75 (± 0.29)	0.207 (± 0.001)	0.127 (± 0.001)

6. Ablation study

Different feature. We report the full resluts in Table 1. In addition to $(f^d, f^p, f^\theta, \Delta p)$ discussed in the main paper, we observe that the view direction encoding helps moderately in KID [1]. We also notice that the geometric feature r_i does

Table 3: **Tabulated K-NN overlaps with the naive K-NN with high probability**. We report the neighbor overlap rate between ours and the naive K-NN for $k = 8$, as well as the coverage rate of the k-th naive nearest neighbors.

1st	2nd	3rd	4th	5th	Overlap rate (all eight)
100%	86.4%	81.5%	75.9%	69.9%	72.6%

Table 4: NPC can also use SMPL [10] surface points for initialization.

	PSNR ↑	LPIPS ↓
DANBO init.	24.86 (+0.00%)	0.115 (+0.00%)
SMPL init.	24.90 (+0.02%)	0.116 (+0.90%)

not contribute meaningfully, and could even be detrimental to the final performance in KID. We include this feature in our design as it was helpful in our early development stage. We hypothesize that r^i brings no benefits since f^d provides bone-relative information with better accuracy. In Figure 6, we visualize how each component can affect the rendering qualitatively. We omit No f^v and No r^i variants from the figure, as they are perceptually indistinguishable from the full model. Note that in the main paper, we report only LPIPS on VGG feature [19], as the VGG and AlexNet [7] scores are highly correlated.

Tabluated K-NN. Our tabulated K-NN returns nearest neighbors that overlap substantially with the naive K-NN, as reported in Table 3. Notably, the tabulated K-NN yields a 9x speed up in forward time compared to the naive counterpart as measured by PyTorch benchmark tool [15].

SMPL initialization. We initialize NPC using SMPL surface points without the LBS weights. As reported in Table 4, both initialization strategies lead to similar PSNR and LPIPS. The results demonstrate the robustness of NPC, offering the option to use SMPL instead of A-NeRF and DANBO if preferred and available.

7. Limitations

In Figure 7, we show an example of the limitations discussed in the main paper.

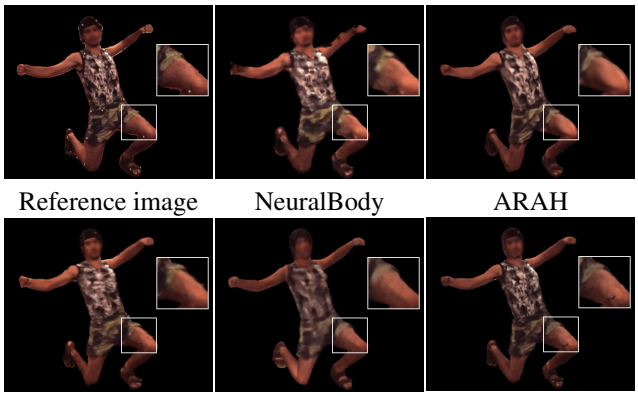


Figure 7: **Limitation.** While NPC retains sharp details on the cloth patterns, the sparse points cannot cover the stretched thigh sufficiently, resulting in ball-shaped artefacts.

References

[1] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 5

[2] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 2

[3] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *CVPR*, 2021. 3

[4] C. Ionescu, F. Li, and C. Sminchisescu. Latent Structured Models for Human Pose Estimation. In *ICCV*, 2011. 1, 2, 3

[5] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *TPAMI*, 2014. 1, 2, 3

[6] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1

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

[8] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *ECCV*, 2022. 3, 4

[9] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021. 2

[10] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG (Proc. SIGGRAPH)*, 34(6):1–16, 2015. 2, 6

[11] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM TOG (Proc. SIGGRAPH)*, 1987. 2

[12] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 3

[13] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. KeypointNeRF: Generalizing

image-based volumetric avatars using relative spatial encoding of keypoints. In *ECCV*, 2022. 4

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 5

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 6

[16] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 1, 2, 3, 4

[17] 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. In *CVPR*, 2021. 2, 3, 4

[18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 3

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 6

[20] Shih-Yang Su, Timur Bagautdinov, and Helge Rhodin. Danbo: Disentangled articulated neural body representations via graph neural networks. In *ECCV*, 2022. 1, 2, 3, 4

[21] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In *NeurIPS*, 2021. 1, 3, 4

[22] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European Conference on Computer Vision*, 2022. 2, 4

[23] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. Monoperfcap: Human Performance Capture from Monocular Video. *ACM TOG (Proc. SIGGRAPH)*, 2018. 1, 3