Supplemental Materials for A-NeRF: Articulated Neural Radiance Fields for Learning Human Shape, Appearance, and Pose

Shih-Yang Su[1] **Frank Yu**[1] **Michael Zollhöfer**[2] **Helge Rhodin**[1]

[1]University of British Columbia [2]Facebook Reality Labs

In this document, we present visualizations of the learned A-NeRF body geometry (Section A), and show additional qualitative results on novel view synthesis (Section B) and pose refinement (Section C). We show comprehensive ablation studies on A-NeRF (Section D), and then make quantitative comparisons between A-NeRF and the most related work [13] on a dataset with perfect training poses (Section E). We justify our choice of view encoding in the context of modeling the view-dependent effects of the human body (Section F). Finally, we describe additional dataset information (Section G) and implementation details (Section H). The supplemental video shows results in motion. ***Real faces and their reconstructions are blurred for anonymity***.

A Geometry Visualization

In Figure A1, we show that A-NeRF can learn convincing body geometry without relying on pre-defined template meshes. The surface-free property also allows A-NeRF to model accessories like headphones, caps and quivers (the second last and the last rows in Figure A2), which are often not included in human template meshes. Moreover, the geometry is consistent between front and back even though learned only from a monocular video, so long as the person is seen from the back. However, very fine details such as the thin arrow shaft are not captured and the surface is not regularized to be smooth.

B Additional Qualitative Results for A-NeRF

We include additional novel view synthesis results in Figure A2. A-NeRF shows plausible synthesis results from different angles. Besides the geometric detail of the density field analyzed in Section A, A-NeRF also reconstructs the appearance of small structures that are not modeled by human surface models and are most difficult to reconstruct, such as arrows, quivers, and caps (first two rows in Figure A2).

C Additional Pose Refinement Results for A-NeRF

We show additional pose refinement results in Figure A3. A-NeRF estimates human poses that align better with the training images.

D Ablation studies

In the following, we quantify the improvements brought about by our contributions concerning pose estimation accuracy and image generation quality. The results support the ablation summary point i), iii) and iv) given in the main document.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

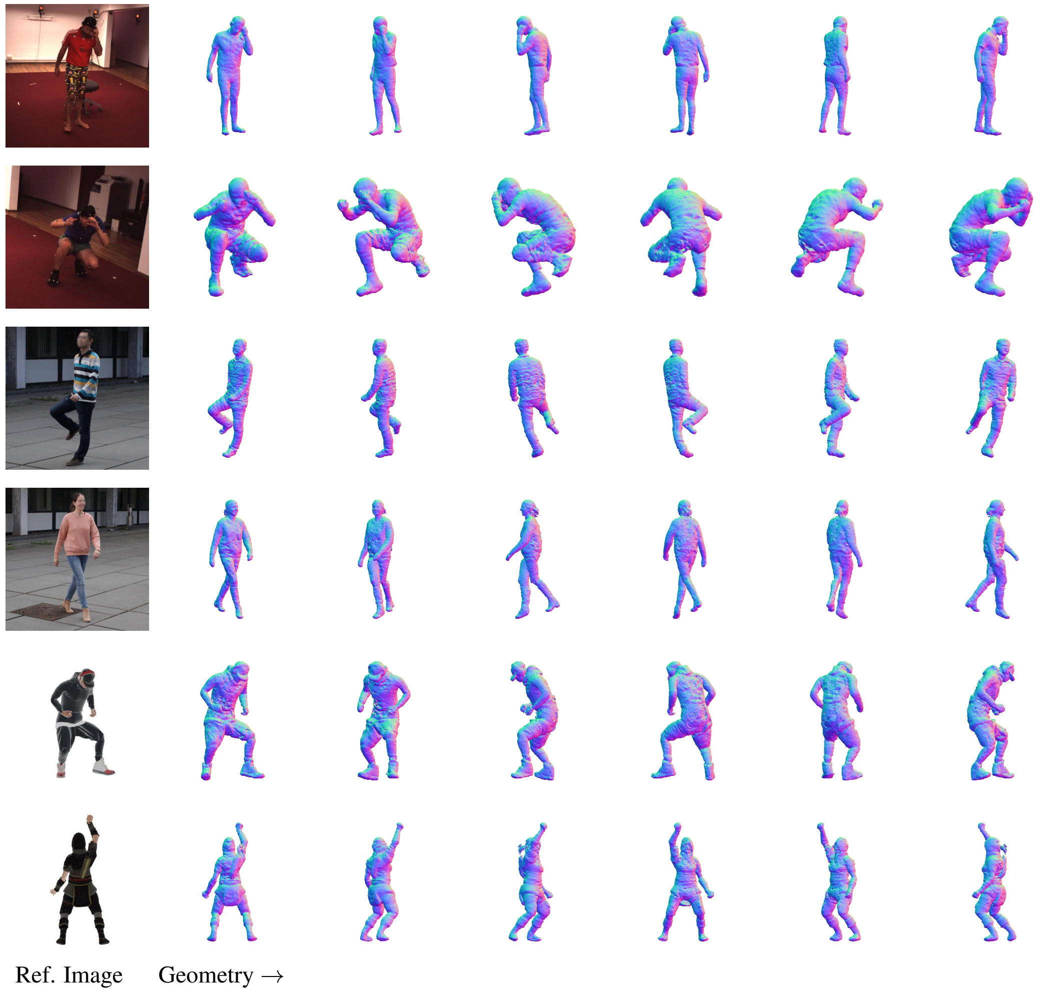

Figure A1: **A-NeRF learns plausible geometry without relying on explicit surface templates and accurate initial poses.** We visualize the geometry using Marching Cubes [9], with the resolution of the voxel grid set to 256.

Table A1: **PA-MPJPE improvements for different joints on Human 3.6M Protocol I.** The higher, the better. A-NeRF helps improve pose accuracy on joints with high initial errors (elbows, wrists, knees, and ankles).

	head top ↑	neck ↑	shoulders ↑	elbows ↑	wrists ↑	hips ↑	knees ↑	ankles ↑	pelvis ↑	spine ↑	head ↑
A-NeRF w/o smoothness prior	1.90	2.35	0.49	3.77	9.04	0.90	3.10	3.78	0.87	0.86	3.32
A-NeRF (Our full model)	2.18	2.16	0.67	3.97	9.44	1.04	3.28	3.66	1.04	1.14	3.28

D.1 Pose Estimation Accuracy Evaluation

We performed the following experiments on the Human3.6M benchmark using the PA-MPJPE metric.

Joint-wise PA-MPJPE Improvements. The MPJPE metric is an average over all the joints. To provide a more detailed view, Table A1 shows PA-MPJPE improvements individually for every body part, in relation to the initialization. A-NeRF drastically improves those joints that have a high initial error (e.g., elbows, wrists, knees, ankles). The most significant is the wrist. These individual improvements highlight the importance of refining initial pose estimates, particularly for applications requiring the precise position of end effectors.

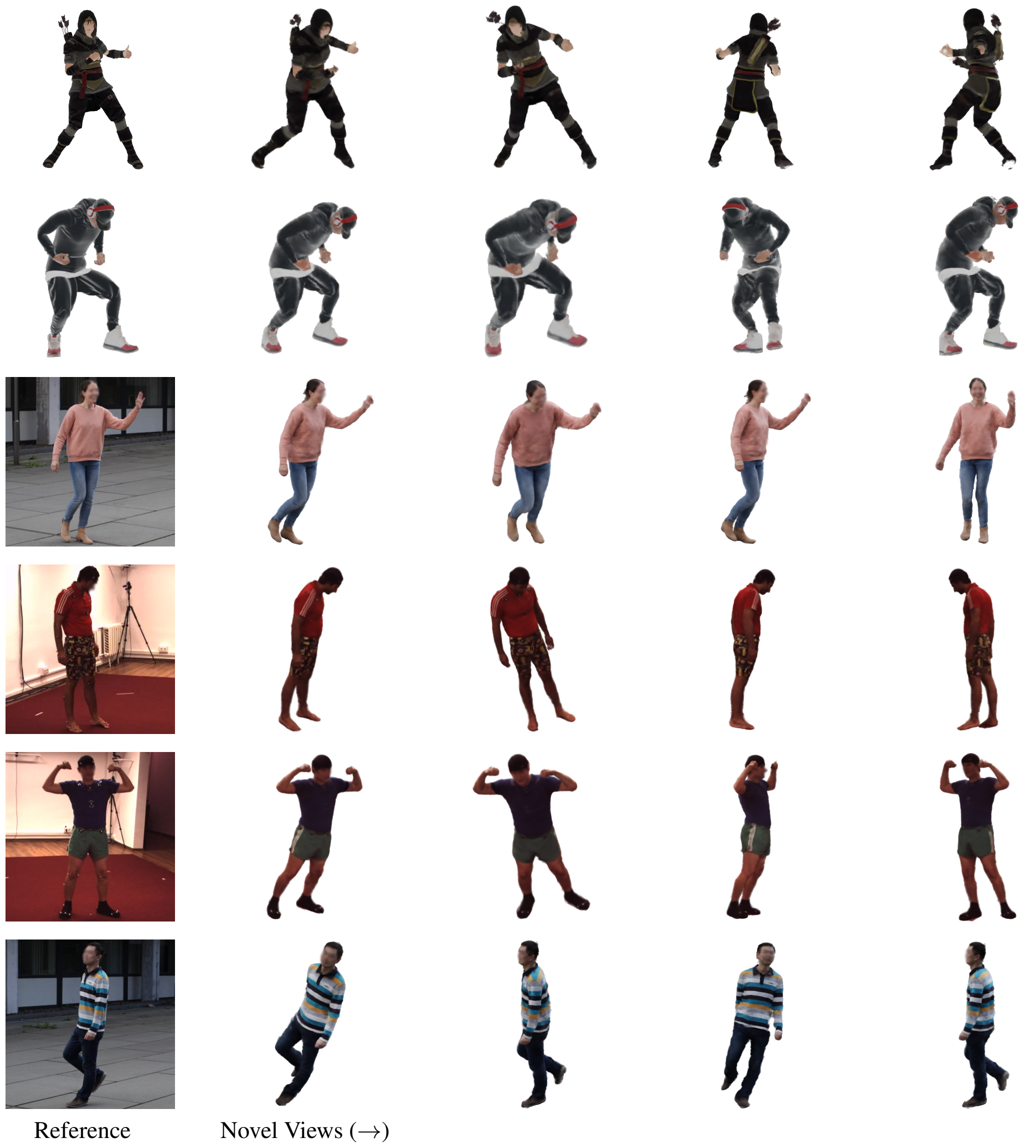

Figure A2: **Additional novel view results.** The learned volumetric body model can be viewed from different directions with consistent geometry. Furthermore, A-NeRF also captures details such as arrows, quivers, and caps (1st and 2nd row), which are usually not included in human surface models; but such thin structures remain challenging. ***Real faces and their reconstructions are blurred in all figures for anonymity.***

Impact of Radial Distance Encoding. In Table A2, we show that Rel. Pos. encoding $\tilde{\mathbf{q}}_k$ does not improve the overall PA-MPJPE, despite being able to refine end effectors. By contrast, our proposed Rel. Dist. encoding leads to a 57% improvement on the wrist over the Rel. Pos. encoding, while improving the overall PA-MPJPE by 0.56. This showcases that ***i) Embedding relative 3D position,*** $\tilde{\mathbf{q}}_k$***, yields only half as good pose refinements as our proposed Rel. Dist.,*** $\tilde{\mathrm{v}}$***, which captures radial information.***

Impact of Cutoff on Refinement. In Table A2, we show that incorporating **Cutoff** can further increase the pose accuracy, with a 15% improvement on wrists over our model without **Cutoff**.

Impact of Pose Regularization. In Table A2, we show that the pose regularizer yields moderate improvement on the pose refinement results, and is not strictly essential for our learning framework.

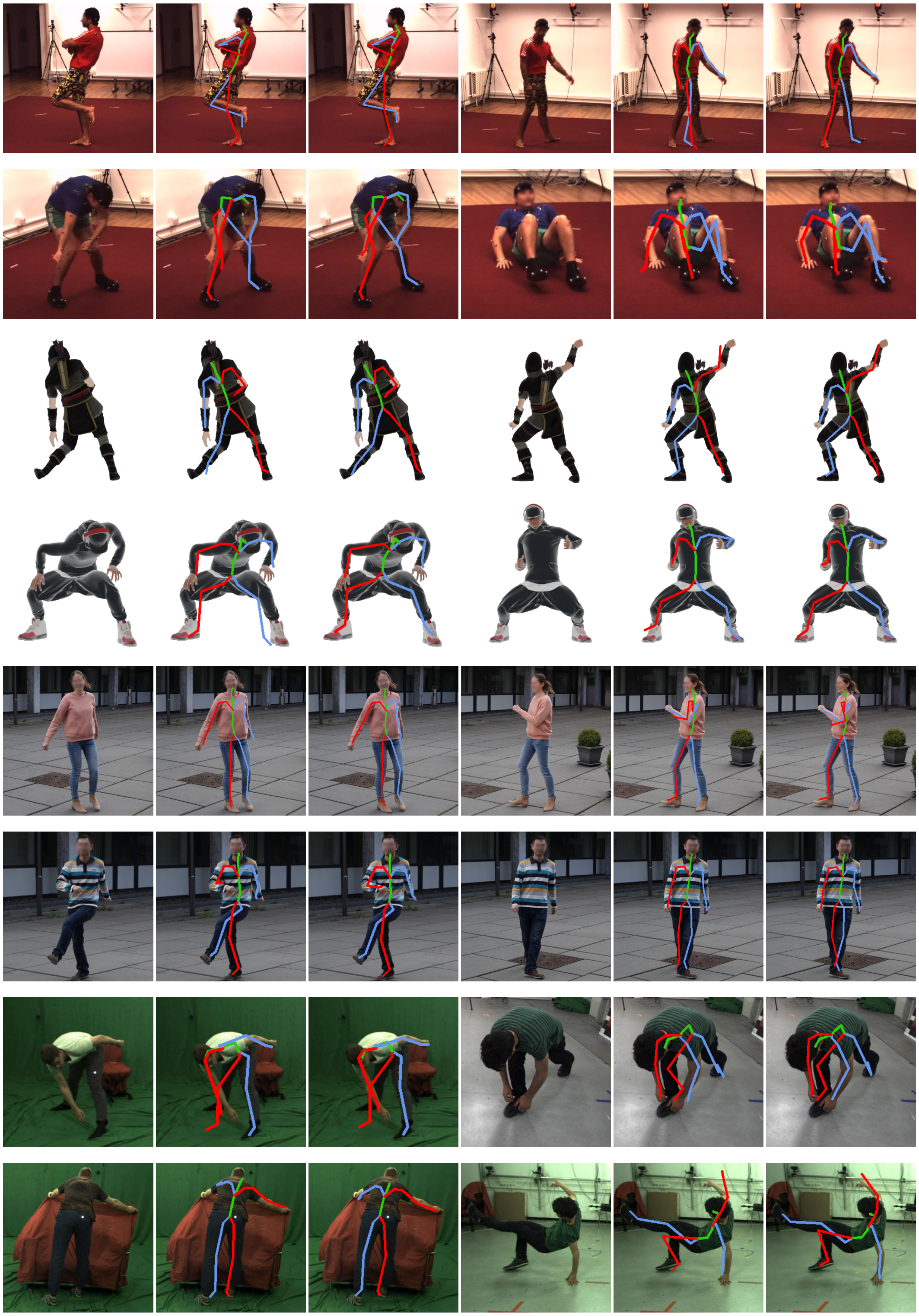

Figure A3: **A-NeRF optimizes the body poses to align better with the images.** ***Real faces and their reconstructions are blurred in all figures for anonymity.***

Table A2: **Pose refinement influence evaluated on Human 3.6M S9 Protocol II2.** Using Rel. Dist. encoding helps refine the estimated poses, and **Cutoff** can further improve the refinement.

	PA-MPJPE↓	Wrist-Improve ↑
SPIN	43.67	0.00
Rel. Pos. + Cutoff	41.90	4.98
Rel. Dist.	41.34	7.85
Rel. Dist. + Cutoff + No Reg	41.21	8.86
Rel. Dist. + Cutoff (Ours)	**40.97**	**9.02**

Table A3: **Directional encoding impact.** The influence of the directional encoding is small but noticeable. It works best to transfer the ray direction relative to the bone coordinates.

Position Rep.	Direction Rep.	PSNR ↑	SSIM ↑	#dim for each Rep.
Rel. Dist. + Rel. Dir.	World Ray Dir.	26.18	0.9456	**360 + 72 + 27**
Rel. Dist. + Rel. Dir.	Ray Ang.	30.13	0.9699	360 + 72 + 216
Rel. Dist. + Rel. Dir.	Rel. Ray (our full model)	**30.61**	**0.9727**	360 + 72 + 648

D.2 Image Generation Quality Evaluation

We evaluate the visual quality of different variants using SURREAL dataset (see Section G for dataset details). Similarly, image quality is quantified via the PSNR and SSIM within the character bounding boxes, comparing the rendered to the ground truth image. We use the test set of 1,500 images showing the training character in novel camera view and novel pose. Here, the ground-truth pose is given as input for training and is not further refined.

On these synthetic sequences, all models are trained using the L2 distance, with ground truth human poses and camera. The batch size is $N_{\text{batch}} = 2048$, and each model is trained for 150k iterations unless stated otherwise.

Impact of the Skeleton-Relative Encoding Variants In addition to our proposed Rel. Dist., Rel. Dir. and Rel. Ray. encodings, we experiment with other alternatives:

- **Relative Ray Angle (Ray Ang.)** A low-dimensional alternative to our Rel. Ray. encoding is to calculate the angle between the ray direction and the vector from query point to joint m

  $$\mathbf{d}' = [\mathbf{d}'_{k,1}, \cdots, \mathbf{d}'_{k,24}], \quad \mathbf{d}'_{k,m} = \text{arc}\cos(\tilde{\mathbf{d}}_{k,m} \bullet \tilde{\mathbf{q}}_{k,m}) \in \mathbb{R}. \tag{1}$$

  As shown in Table A3, our proposed Rel. Ray. offers better image quality.
- **Bone-relative Position (Rel. Pos.)** Table A4 shows the result of different query position encoding. Note that we do not apply **Cutoff** to these models. The straightforward baselines (World Position + Joint Position + θ) that condition on pose directly perform the worst. We find that these baselines achieve 0.7 SSIM by simply rendering background colors. On the other hand, Rel. Pos. significantly outperforms the straightforward baselines. However, it does not provide any gain over our proposed Rel. Dist. + Rel. Dir. encoding, albeit having drastically more dimensions. This shows that ***ii) Our embedding choices keep the dimensionality moderate while improving on or matching the PSNR of higher-dimensional variants.***
- **Conditioning on pose.** In Table A5, we further compare different ways of encoding the bone directions. Our proposed Rel. Dir. encoding shows result superior to the baseline that conditioned directly on θ.

Numbers of Poses and Views on Visual Quality. We experiment A-NeRF with varying numbers of poses and views. All models are trained for 300k iterations in this experiment. The result in Table A6 shows not only that A-NeRF can be trained with only a single view but also that adding more views does not help if the poses are not diverse enough. As evidenced by rows 2 and 3, it helps to supply additional images of new poses, nearly as much as adding new views of the same pose. We conclude that ***iii) For a fixed number of images with accurate poses, the visual quality of***

Table A4: **Query encoding trade-off.** Encoding world coordinates does not succeed on motions and has a large memory consumption. Encoding positions relative to the skeleton works yet also has high dimensionality. Our full model that combines distance and direction performs well in both aspects. Note that we do not apply cutoff to these models.

Position Rep.	Direction Rep.	PSNR ↑	SSIM ↑	#dim for each Rep.
World Pos. + Joint Positions + θ	World Ray.	14.66	0.7554	1125 + 72 + 27
World Pos. + Joint Positions + θ	Rel. Ang.	15.90	0.7602	1125 + 72 + 216
World Pos. + Joint Positions + θ	Rel. Ray.	14.40	0.7185	1125 + 72 + 648
Rel. Pos.	Ray Ang.	29.22	0.9638	1080 + 216
Rel. Pos.	Rel. Ray.	29.25	0.9631	1080 + 648
Rel. Dist. +Rel. Dir	Ray Ang.	**29.99**	**0.9702**	**360 + 72 + 216**
Rel. Dist. +Rel. Dir (our model w/o cutoff)	Rel. Ray.	29.88	0.9692	360 + 72 + 648

Table A5: **Distance-based positional encoding** is compact (360 dim) but insufficient (lower PSNR and SSIM) to encode skeleton relative query locations unless paired with direction information in our full model (72 dim, w/o positional encoding).

Position Rep.	Direction Rep.	PSNR ↑	SSIM ↑	#dim for each Rep.
Rel. Dist.	Rel. Ray	23.76	0.9137	**360 + 0 + 648**
Rel. Dist. + θ	Rel. Ray	26.33	0.9340	360 + 72 + 648
Rel. Dist. + Rel. Dir. (our full model)	Rel. Ray	**30.61**	**0.9727**	360 + 72 + 648

one long video plus diverse poses is comparable to multiple shorter multi-view recordings, which makes the simpler monocular capture set-up preferable.

E Visual Comparison to NeuralBody.

To see how our proposed A-NeRF compares to surface-based approaches in capturing articulated textured avatars, without including the uncertainty of inaccurate initial pose estimates, we train both NeuralBody and A-NeRF on SURREAL with accurate poses and ground truth camera, similar to Section D. Precisely, both models are trained on SURREAL with images from 3 out of 9 cameras in the training data, with 400 randomly sampled poses for 300K iterations (the same setting as in Table A6). We report the result in Table A7. We observe that NeuralBody produces slightly deformed/expanded body features for poses that are very different from the training data[1]. We conclude that, while NeuralBody shows impressive results on reenacting human performance with moderate sequence length, A-NeRF works better for longer sequences and for retargeting to diverse poses.

F Discussion on View-dependent Effects

To approximate the rendering equation, we need the position in space, incoming light and the outgoing ray direction. In the original NeRF, the light sources are not explicitly reconstructed, but are instead baked into the view directions. Thus, the incoming light information is represented as a black-box function. In our case, we assume a static, estimated camera for each of the images. As a result, light sources are inconsistent between frames, and therefore the model can no longer encode the incoming light information into global view directions (World Ray). To counter this issue, we use the per-image code design [10, 13] to represent the illumination effects as a black box. Furthermore, our Rel. Ray. explicitly encodes the outgoing ray direction with respect to each body part, allowing the model to better explain the illumination on the human body.

We conduct experiments on Human3.6M S9 to examine how the per-image code, Rel. Ray., and World Ray affect the performance on both pose refinement and rendering quality. As shown in Table A8,

---

[1] We have contacted the authors of NeuralBody, and confirmed that such artifacts are expected for unseen test poses.

Table A6: **Monocular vs. multi-view reconstruction.** The PSNR and SSIM scores show that our model can learn as well from a single view as from multiple ones. Having more views does not help if the poses are not diverse enough, as evidenced by rows 2 and 3.

#views	#poses	#imgs	PSNR ↑	SSIM ↑
1	1200	1200	28.32	0.9607
3	400	1200	29.10	0.9655
9	134	1206	29.00	0.9644
9	1200	10800	31.30	0.9750

Table A7: **Visual quality comparison between NeuralBody and A-NeRF.** .

Method	PSNR ↑	SSIM ↑
NeuralBody [13]	23.86	0.9304
Ours	29.10	0.9655

learning with World Ray led to reduced performance, and we observe flickering results for novel view rendering (e.g., bullet-time effect). This indicates that the camera (world) causes overfitting in our single/estimated camera setting and that it is beneficial to encode all quantities relative to the skeleton. Not using any directional information (No Ray) yields the worst results. Finally, the per-image code shows visible improvement in all cases, and it attains the best performance in combination with Rel. Ray. encoding. The outcomes support our claim that, by modeling the outgoing ray direction explicitly in relative space, we enable A-NeRF to better recover the illumination effect. Finally, we acknowledge that, while the lighting effect looks plausible in our novel view synthesis results (Figure A2), the illumination is not physically correct. But since our goal is to model an articulated human avatar instead of a recreating 3D scene with perfect light sources, we argue that our proposed Rel. Ray. serves our purpose well. It is a promising direction for future work to model illumination more explicitly and to enable relighting applications.

G Dataset Details

In addition to the real-world datasets introduced in the main paper, we experiment A-NeRF on two synthetic datasets.

- **Human 3.6M [4]** For Protocol I, we follow [5–7] to evaluate PA-MPJPE on every 5^{th} frame of the frontal camera. For Protocol II, PA-MPJPE is calculated on every 64^{th} frame from all cameras, following [12, 14].
- **SURREAL [15]** We generate a synthetic dataset using [15] to examine how each factor affects the visual quality. We use 1500 3D poses from [3], divide them into a 4-1 train-test split. Each training pose is rendered with 9 different cameras, resulting in a total of 10,800 512×512 training images. We render the 300 testing poses with 5 different cameras to construct a test set of 1,500 images.
- **Mixamo [1]** We generate two synthetic subjects, “James” and “Archer” from Mixamo, and render each subject from varying camera angles with three provided motions, “Thriller Part 3”, “Robot-Hip-Hot Dance”, and “Shoved Reaction with Spin”. Each subject has 1,130 training images. We extract pose and camera parameters with [7].

H Implementation Details

Our A-NeRF model is learned without supervision on a single or multiple videos of the same person. Camera intrinsics, bone lengths for setting $\mathbf{a}_m$, and pose θ_k are initialized with [7] for every frame k. These poses are then optimized for 500k iterations on objective Eq.2, together with the neural parameters ϕ, which model shape and appearance. We then stop optimizing pose and continue to train ϕ with only the data term for additional 200k iterations, which further improves visual fidelity. We form a training batch by randomly sampling $N_{\text{batch}} = 3072$ rays from all available images. Therefore, an image k can have no or only very few samples, leading to noisy updates of θ_k. To counteract, we

Table A8: **Comparison of different strategies for encoding illumination effect on Human 3.6M S9 Protocol II and validation split.** Using per-image codes and Rel. Ray. results in the best performance. World Ray causes overfitting in our static/estimated camera setting.

	w/ per-image code	PSNR↑	SSIM ↑	PA-MPJPE↓
No Ray		25.89	0.9132	41.99
World Ray		26.04	0.9140	41.75
Rel. Ray. + World Ray		26.25	0.9170	41.00
Rel. Ray. (our model w/o per-image code)		26.26	0.9167	41.03
No Ray	✓	26.60	0.9164	41.54
World Ray	✓	26.55	0.9161	41.41
Rel. Ray. + World Ray	✓	26.64	0.9192	41.00
Rel. Ray (Ours)	✓	**27.27**	**0.9245**	**40.97**

Table A9: **Hyperparameters.** We tune the hyperparameters on Human 3.6M subject 1.

	Mixamo	Human 3.6M	MonoPerfCap	MPI-INF-3DHP	SURREAL
G (#gradient accumulation)	20	50,125†	20	10	n/a*
λ_θ	2.0				n/a*
λ_t	0.05				n/a*
N_{batch}	3072				2048
Coarse samples	64				
Fine samples	16				
L input (#PE frequencies)	7				
L view (#PE frequencies)	4				
t (cutoff distance)	500mm				
τ (cutoff temperature)	$-20 \xrightarrow{\text{exp dec. 250k}} -200$				
Learning rate ϕ	$5\text{e-}4 \xrightarrow{\text{exp dec. 500k}} 5\text{e-}5$				
Learning ϕ rate (finetune)	$2\text{e-}4 \xrightarrow{\text{exp dec. 200k}} 5\text{e-}5$				n/a*
Learning rate θ	5e-4			1e-4	n/a*

* When GT poses are used for training, we do not enable refinement and scratch the associated hyperparameters.
† for Protocol I, which has around 3 times more images than Protocol II.

accumulate the gradient update $\Delta\theta$ for $G \geq (30\dot{N}_{\text{batch}})/N$ iterations, so that each θ_k is expected to be sampled 30 $\geq$ times before the gradient update. To speed up the rendering part, we use the same coarse-to-fine importance sampling strategy of [11] and restrict samples to be within a coarse human silhouette estimated with [2] unless the background is monochrome. The training is done on 2 Nvidia Tesla V100 32GB GPUs, which takes approximately 60 hours. It takes around 1-4 seconds to render one $512 \times 512 \times 3$ image using a single Nvidia Tesla V100 GPU. We tuned the hyperparameters on Subject 1 of the Human3.6M training set.

Skeleton Representation. We use a skeleton representation that encodes the connectivity and static bone lengths via a rest pose of the 3D joint locations $\{\mathbf{a}_m\}_{m=1}^{24}$, with the root at $\mathbf{0}$ that are connected via B bones. Dynamics are modeled with per-frame skeleton poses $\theta_k = [\omega_{k,0}, \cdots, \omega_{k,24}]$ that include the relative rotation of 24 joints, $\omega_{k,m}$, to their parents as well as the global position $\omega_{k,0}$ as a special case. For rotation, we experiment with the recently proposed overparametrized representation of [16] ($\omega_{k,m} \in \mathbb{R}^6$) and the traditional axis-angle representation ($\omega_{k,m} \in \mathbb{R}^3$). The subscript $_{k,m}$ indicates that a variable is related to the m-th joint of image $\mathbf{I}_k$. We will exploit that every bone defines a local coordinate system. We write the mapping of a 3D position $\mathbf{p}_{k,m} \in \mathbb{R}^3$ in the m-th local bone coordinates to world coordinates $\mathbf{q} \in \mathbb{R}^3$ via forward kinematics using homogeneous coordinates,

$$\begin{bmatrix}\mathbf{q}\\1\end{bmatrix} = T(\theta_k, m)\begin{bmatrix}\mathbf{p}_{k,m}\\1\end{bmatrix}, \text{ with transformation } T(\theta_k, m) = \prod_{l \in A(m)} \begin{bmatrix}\mathbf{R}(\omega_{k,l}) & \mathbf{a}_{l,l-1}\\ \mathbf{0} & 1\end{bmatrix} \in \mathbb{R}^{4\times4}, \tag{2}$$

$A(m)$ the ordered set of the joint ancestors of m, and $\mathbf{a}_{l,l-1} \in \mathbb{R}^3$ is the translation between joint l and its child joint $l-1$. Conversely, $T(\theta_k, m)^{-1}$ maps world to local bone coordinates. The rotation

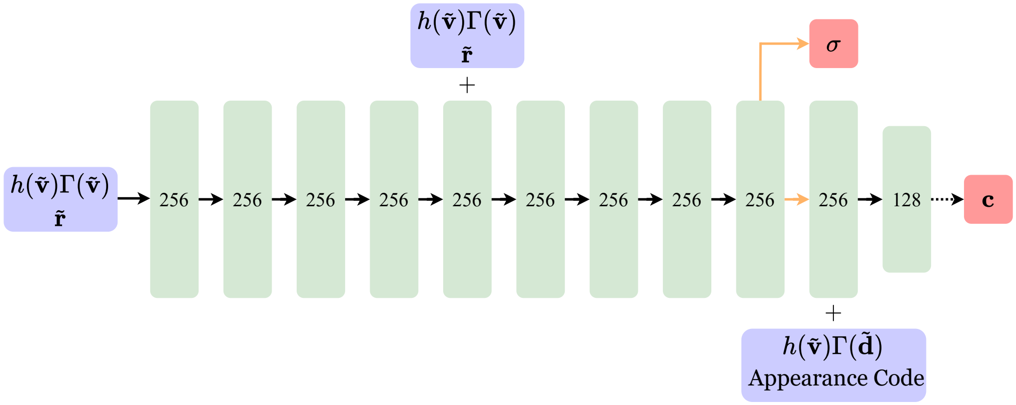
$h(\tilde{\mathbf{v}})\Gamma(\tilde{\mathbf{v}})$
$\tilde{\mathbf{r}}$
+
σ
$h(\tilde{\mathbf{v}})\Gamma(\tilde{\mathbf{v}})$
$\tilde{\mathbf{r}}$
256
256
256
256
256
256
256
256
256
256
128
$\mathbf{c}$
+
$h(\tilde{\mathbf{v}})\Gamma(\tilde{\mathbf{d}})$
Appearance Code

Figure A4: A-NeRF architecture follows the original NeRF [11], with an additional appearance code similar to concurrent works [10, 13] for handling illumination. We use purple blocks for inputs, green blocks for intermediate features with numbers representing their sizes, red blocks for outputs. Black and dashed arrows imply the use of ReLU and sigmoid activations respectively. Orange arrows indicate that no activation is applied. Appearance code and our Rel. Ray. encoding are fed into the network in the second last fully-connect layer so that these representation do not affect human body geometry (as in the original NeRF).

matrix $\mathbf{R}(\omega_{k,l})$ is inferred as in [16]. Note that our skeleton is equivalent to SMPL [8] and others, but without their parametric surface model, and can therefore be initialized with any skeleton pose estimator.

Multi-view extension. We can incorporate a multi-view constraint to improve pose refinement when the motion is captured from multiple cameras. For initializing θ_k, we average the individual joint rotation estimates from all V views $v \in [1, \ldots, V]$. Since rotations are relative to the parent and root, this works without calibrating the cameras. Only the global position and orientation remain specific to view v. To this end, we extend our single-view notation with subscripts, position $\omega_{k,1}^{(v)}$ and orientation $\omega_{k,2}^{(v)}$ are estimated relative to camera v. With slight abuse of notation we rewrite Eq. 2,

$$\mathcal{L}^{\text{MV}}(\theta, \phi) = \sum_{v} \sum_{k} d(C_\phi(\theta_k, \omega_{k,0}^{(v)}, \omega_{k,1}^{(v)}), \mathbf{I}_k^{(v)}) + \lambda_\theta d\left(\theta_k - \hat{\theta}_k\right) + \lambda_t \left\| \frac{\partial^2 \theta_k}{\partial t^2} \right\|_2^2, \tag{3}$$

with θ_k shared across all views v, except for $\omega_{k,0}^{(v)}$ and $\omega_{k,1}^{(v)}$ which are estimated independently per camera view v since the relative camera position and orientation is unknown in our setting.

Efficient Sampling. To increase sampling efficiency, we define a cylinder surrounding the initial skeleton pose (as sketched in Figure 2). The radius of the cylinder is defined as the distance between the root joint and the joint farthest away from it, plus a 250mm buffer length. We sample points along the ray segment that lies inside the cylinder. Furthermore, as explained in the main paper, we restrict samples to be within a foreground mask. Because the mask is only approximate, we dilate the estimated foreground mask by 5 pixels.

Hyperparameter tuning. Table A9 shows our hyperparameters for A-NeRF on different datasets. All parameters are tuned on Subject 1 of H3.6M, with only the learning rate changed to apply to the 3DHP dataset with much shorter sequence sizes and the number of gradient accumulations computed relative to the number of frames as defined in the main document.

Network architecture. We start from the same network architectures as in the original NeRF [11] (see Figure A4). Following concurrent works on handling illumination changes [10, 13], we add a 16-dimensional appearance code to the second last layer of the NeRF network F_ϕ. It is individually stored and optimized for every frame. Note that the NeuralBody baselines use 128-dimensional codes. Due to its position at the end of the network and its low dimensionality, it helps learning these global effects while not deteriorating the benefits of the relative encoding.

NeuralBody baseline For a fair comparison of A-NeRF to NeuralBody, we increase the batch size for NeuralBody so that the number of training samples per batch will be the same (which improves its performance). We keep other hyperparameters the same as provided in the official implementation and re-train models on our datasets.

Novel view synthesis. The novel viewpoints in all our qualitative visualization are generated by regular sampling from two different camera trajectories: (1) cameras spaced from 0 to 360 degrees (bullet-time effect), and (2) a circular trajectory that rotates from -25 to 25 degrees along the y-axis and -15 to 15 degrees along the x-axis (elevation), with an additional zoom-in and zoom-out effect.

References

[1] Adobe. Mixamo. `https://www.mixamo.com/`, 2020.

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[3] CMU Graphics Lab Motion Capture Database. CMU Graphics Lab Motion Capture Database. `http://mocap.cs.cmu.edu`.

[4] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014.

[5] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018.

[6] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[7] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[8] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM TOG (Proc. SIGGRAPH)*, 34(6):1–16, 2015.

[9] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH*, 1987.

[10] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[12] A. Nibali, Z. He, S. Morgan, and L. Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *WACV*, 2019.

[13] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.

[14] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *ICCV*, pages 2602–2611, 2017.

[15] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[16] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019.