VolumetricSMPL:
VolumetricSMPL: A Neural Volumetric Body Model for Efficient Interactions, Contacts, and Collisions
– Supplementary Material –

A. Overview

In this supplementary document, we provide additional implementation details (Appendix B) and further insights into the importance of design choices in VolumetricSMPL (Appendix C). We also include additional information regarding the downstream applications of our model (Appendix D).

Next, we illustrate our easy-to-use Python interface, which seamlessly integrates with the widely used SMPL-X package (Appendix F). Finally, we discuss the limitations of VolumetricSMPL and outline potential future research directions (Appendix E).

B. Implementation Details and Comparisons

B.1. Network Architectures

The VolumetricSMPL architecture consists of two primary neural components: a shared PointNet [51] encoder for all body parts and an MLP decoder, implemented using weights predicted by the Neural Blend Weights (NBW) generator (Fig. 3).

VolumetricSMPL Encoder. After a forward pass through the SMPL-based body model, the posed skin mesh is partitioned into 15 ($K = 15$) body parts based on the kinematic chain of the human body, following [42]. Each body part is then normalized according to its respective bone transformation G_k. The local mesh of each part is resampled as a point cloud with 1k points and encoded using a shared PointNet encoder, ensuring memory-efficient alignment across all body parts.

The shared encoder follows a 128-neuron MLP PointNet architecture, interleaved with ReLU activations. It consists of an input linear layer, four ResNet blocks, and an output layer. Each ResNet block contains two linear layers with a skip connection, facilitating effective feature propagation. The output layer produces a 128-dimensional latent code $\mathbf{z}_k$, which conditions the decoder. This structure allows for efficient encoding of local shape variations while maintaining compact memory usage and fast inference speed.

VolumetricSMPL Decoder. The MLP decoder is a compact 7-layer network with 64 neurons per layer and a skip connection on the 3rd layer, interleaved with ReLU activations. This lightweight architecture ensures efficient computation while maintaining high expressivity, as the Neural Blend Weights (NBW) framework enables a large number of learnable parameters to be utilized effectively.

In addition to being conditioned on the latent code, the MLP decoder also takes a local query point as input $\mathbf{x}_k$. To enhance spatial encoding, the query point is positionally encoded [44] using only two frequency levels, providing better representation capacity for fine-scale details. Utilizing even higher frequency signals as input severely hampers the prediction accuracy and makes the training unstable.

Table B.1. **Comparison of Canonical SMPL-based vs. Flexible Direct Modeling.** We evaluate the impact of conditioning VolumetricSMPL on an explicit mesh prior versus learning a volumetric representation directly in observation space using only low-dimensional shape and pose coefficients (β, θ) as in [2]. Both methods are trained under the same protocol (Sec. 3.4). Surf. and Unif. denote IoU scores computed for points sampled near the surface and uniformly in space, respectively. Results show that incorporating explicit mesh priors significantly improves IoU and reduces SDF prediction error, demonstrating the benefits of our canonical compositional modeling approach. The experimental setup follows Tab. 1.

	IoU [%] ↑			MSE ↓	
	mean	*surf.*	*unif.*	SDF	|SDF|
Direct Modeling [2]	39.64	32.28	47.01	2.5×10^{-3}	3.5×10^{-4}
VolumetricSMPL	**94.67**	**94.25**	**95.10**	$\mathbf{3.7 \times 10^{-5}}$	$\mathbf{3.5 \times 10^{-5}}$

B.2. Volumetric Bodies

VolumetricSMPL is trained following the procedure outlined in Sec. 3.4 of the main paper. For baseline comparisons, we use the publicly released COAP [42] and LEAP [43] models, which are also trained on the AMASS subsets, for the respective SMPL [38, 49] body.

B.3. Additional Comparisons

Volumetric models such as VolumetricSMPL, LEAP [43], and COAP [42] are designed to share the same learning space as their underlying mesh-based parametric models [38, 49]. This ensures seamless integration with methods operating in the SMPL parameter space [56], as demonstrated in the applications section (Sec. 4.2).

In contrast, models such as NASA [11] and imGHUM [2] do not rely on explicit SMPL priors. Instead, they learn volumetric representations directly from scans, body meshes, or a combination of both. While this increases flexibility, it tends to be computationally expensive—NASA requires per-subject training, while imGHUM relies on a significantly larger architecture and proprietary private human scans for training.

Specifically, imGHUM requires propagating a query point through a deep stack of MLPs: an 8-layer 512-dimensional MLP, an 8-layer 256-dimensional MLP, and two 4-layer 256-dimensional MLPs, being 86% slower for inference compared to VolumetricSMPL. In contrast, our MLP decoder is significantly more lightweight, using a 7-layer 64-dimensional MLP for partitioned body parts.

Impact of Excluding SMPL Priors. To evaluate the role of explicit SMPL conditioning, we re-implement the imGHUM architecture under our training setup (Sec. 3.4). This adaptation removes the PointNet encoder that processes SMPL-derived point samples, replacing it with a large SDF decoder MLP that directly conditions on SMPL parameters (β, θ). This results in a volumetric model operating in observation space, without part-wise canonicalization.

We train both models and evaluate them following Sec. 4.1. Results in Tab. B.1 show that excluding the explicit SMPL prior significantly degrades generalization when training data is limited. Notably, our method requires training samples only within bounding boxes $\mathcal{B}$, leveraging an analytic SDF for the outer volume. Hence a model trained only with samples within $\mathcal{B}$ can produce floater artifacts in outside regions, leading to large errors.

Direct comparison with the released pre-trained imGHUM model is infeasible, as it learns a different human shape space from proprietary data. Additionally, an additional key advantage of compositional volumetric models (*e.g.*, COAP and VolumetricSMPL) is their ability to resolve self-intersections (Sec. 4.2.4), unlike LEAP and imGHUM.

Performance Breakdown. To better isolate the contributions of each component in our efficient querying pipeline (Sec. 3.2), we evaluated the impact of the coarse analytic SDF acceleration. In Tab. 1 and Tab. 2, removing the coarse acceleration increases the runtime from 15 ms to 25 ms and memory usage from 3.1 GB to 5.0 GB—showing a $\sim$1.7$\times$ speedup and $\sim$1.6$\times$ memory reduction attributable to the analytic term.

C. Ablation studies

C.1. Impact of the Point Cloud Sampling

As described in the method section, VolumetricSMPL applies kinematic-based mesh partitioning [42] to the SMPL body mesh and normalization of each mesh part according to its respective bone transformation G_k. Each local mesh is then resampled into a point cloud with 1k points, which is subsequently encoded using a shared PointNet encoder to ensure efficient memory alignment across all body parts.

Table C.1. **Impact of the Point Cloud Sampling.** The number of point samples per-body part has moderate impact on the model performance and computational resources (inference speed and GPU memory). 1k samples is the default configuration in the main paper (denoted 1,000*). The models are trained for 10 epochs. The default setup strikes a good balance between accuracy and resource consumption.

Point	Resources		IoU [%] ↑			MSE $\times 10^{-5}$ ↓	
Samples	t [ms] ↓	GPU ↓	mean	*surf.*	*unif.*	SDF	|SDF|
250	15	2.6	87.47	84.94	90.01	6.37	7.15
500	15	2.7	90.71	87.87	93.55	6.34	6.96
750	15	2.9	91.31	88.45	94.16	6.40	6.91
1,000*	15	3.1	91.43	88.49	94.38	6.47	6.96
1,250	15	3.2	91.45	88.51	94.38	6.52	6.97
1,500	15	3.4	91.40	88.50	94.30	6.56	7.00
1,750	15	3.5	91.33	88.44	94.21	6.57	7.00
2,000	15	3.7	91.33	88.42	94.24	6.59	7.06

Table C.2. **Impact of the Bounding Box Size.** We evaluate how different bounding box sizes B_k affect SDF accuracy. Larger boxes degrade SDF predictions, as the neural network becomes less precise further from the iso-surface, making analytic SDF estimation preferable in these regions. The optimal padding of 12.5% (denoted by *) achieves the lowest mean SDF error and is used in the final model. Reported values are scaled by $\times 10^{-5}$.

	Bounding Box B_k Size (%)													
MSE	5	7.5	10	**12.5***	15	20	30	40	50	60	70	80	90	100
SDF	13.7	11.6	9.0	6.5	6.3	6.2	6.0	6.1	6.4	6.8	7.3	7.9	8.4	9.0
|SDF|	6.3	6.5	6.7	7.0	7.3	8.0	10.1	13.8	19.5	27.9	39.9	56.0	77.7	104.9
mean	10.0	9.1	7.9	**6.7**	6.8	7.1	8.1	10.0	12.9	17.4	23.6	32.0	43.0	57.0

To evaluate the impact of the number of sampled points on both model performance and computational efficiency, we conduct an ablation study, summarized in Tab. C.1. The reported results correspond to VolumetricSMPL trained for 10 epochs under the experimental setup outlined in the main paper. The findings indicate that the default setting of 1k points achieves a good balance between accuracy and resource efficiency, making it a suitable choice for practical deployment.

C.2. Impact of the Bounding Box Size

We evaluate how the bounding box size B_k affects model performance. While varying the padding level does not meaningfully affect occupancy/sign evaluation, it does influence signed distance accuracy.

To analyze this effect, we ablate padding levels from 5% to 100%, with results summarized in Tab. C.2. As shown, larger bounding boxes degrade SDF predictions, as the neural network becomes less precise further from the iso-surface, making analytic SDF estimation preferable in

Table C.3. **Impact of MLP Size on Efficiency and Accuracy.** We evaluate the effect of reducing the MLP decoder size by comparing architectures with 32, 40, 50, and 64 neurons (default). As shown, smaller MLPs significantly reduce computational cost: using 32 neurons instead of 64 reduces inference time by 33.3% (15 ms → to 10 ms) and GPU memory usage by 35.5% (3.1 GB → 2.0 GB). However, this comes at the cost of increased SDF error ($|SDF|$ rising by ∼50%) and decreased IoU. The 64-neuron configuration achieves a good balance between computational efficiency and reconstruction accuracy, making it the optimal choice for resource-intensive downstream tasks.

	Resources			IoU [%] ↑			MSE [$\times 10^{-5}$] ↓	
Neurons	t [ms] ↓	GPU ↓	#param	mean	*surf.*	*unif.*	SDF	|SDF|
32	10	2.0G	1.7M	94.12	93.73	94.50	4.5	5.2
40	11	2.2G	2.1M	94.02	93.60	94.45	4.6	5.2
50	12	2.4G	2.8M	94.54	94.25	94.82	5.3	5.5
64	15	3.1G	4.0M	**94.67**	**94.25**	**95.10**	**3.7**	**3.5**

these regions.

The optimal padding of 12.5% achieves the lowest mean SDF error which is adopted as the final model parameter.

C.3. Even smaller MLPs: Impact on Efficiency and Accuracy

We further analyze the impact of reducing the MLP decoder size by comparing architectures with 32, 40, and 50 neurons with the default setting of 64 neurons. Results are summarized in Tab. C.3.

As shown in Tab. C.3, smaller MLPs reduce computational costs. Using 32 neurons instead of 64 reduces computation time by 33.3% (15 ms → 10 ms) and GPU memory usage by 35.5% (3.1 GB → 2.0 GB). However, this comes at the cost of substantially lower accuracy, with |SDF| errors increasing by 50%. The 64-neuron setup achieves a good trade-off between computational efficiency and reconstruction accuracy while being useful for many resource intensive downstream tasks.

C.4. Alternative Architectures

We also experimented with alternative MLP architectures, including SIREN [54], HyperNetworks [16], and FiLM-based conditionings [7]. However, due to the weak supervision signal in our model—where only the global SDF prediction (after the min operation) is supervised—these approaches failed to converge in a feed-forward setting.

C.5. Comparison with Classic Techniques

We further compare our method against traditional techniques such as winding numbers [22], which have been used in human contact modeling [12, 45]. However, winding numbers are not differentiable and do not generalize well to applications that require differentiable collision loss terms [35, 59, 68, 69], unlike VolumetricSMPL.

Table C.4. **Comparison of occupancy checks using winding numbers and VolumetricSMPL.** We evaluate inference time and GPU memory usage to check whether SMPL-X vertices are inside another SMPL-X body. VolumetricSMPL achieves over 40× faster inference and reduces GPU memory usage by 50×, making it significantly more efficient for large-scale learning tasks.

	Resources	
	Inference Time ↓	GPU Memory ↓
Winding Numbers [22, 45]	464.41 ms	15.58 GB
VolumetricSMPL	**11.06** ms	**0.27** GB

To quantify the efficiency gap, we evaluate inference time and GPU memory usage for occupancy checks between two SMPL-X bodies by determining whether one body's vertices reside inside another. We adopt the implementation from [45] and report the results in Tab. C.4.

As shown in Tab. C.4, winding numbers introduce significant computational overhead, making them impractical for learning-based tasks requiring large batch sizes without down-sampling human meshes. In contrast, VolumetricSMPL is over 40× faster and consumes 50× less GPU memory, enabling efficient large-scale training and inference.

D. Downstream Tasks

In the following, we provide further implementation details for the downstream tasks illustrated in Fig. 1.

D.1. Reconstructing Human-Object Interactions from Images in the Wild

Following the two-stage methodology proposed by Wang *et al.* [59] and PHOSA [68], we first independently reconstruct humans and objects from the input image. In the second stage, a joint optimization step refines their contacts and spatial arrangements.

Unlike [59], which uses a time-consuming collision loss based on mesh-triangle intersections, we replace this step with an efficient alternative by transforming the body mesh into signed distance fields. Specifically, we use VolumetricSMPL to compute penetration loss efficiently, significantly improving computational speed while maintaining accuracy.

Initial Body Poses and Shapes Estimation. We use PARE [31] to predict the pose θ and shape β parameters of the SMPL [38] parametric body model. Next, we leverage MMPose [9] to detect 2D body joint keypoints $\mathbf{J}_{2D}$. Finally, we refine the predicted SMPL body model by fitting it to the detected 2D keypoints using SMPLify [5].

The optimization objective for SMPLify is formulated as:

$$E(\beta, \theta) = \left|\left|\Pi(\hat{\mathbf{J}}_{3D}) - \mathbf{J}_{2D}\right|\right|_2^2 + E_\theta + E_\beta, \qquad \text{(D.1)}$$

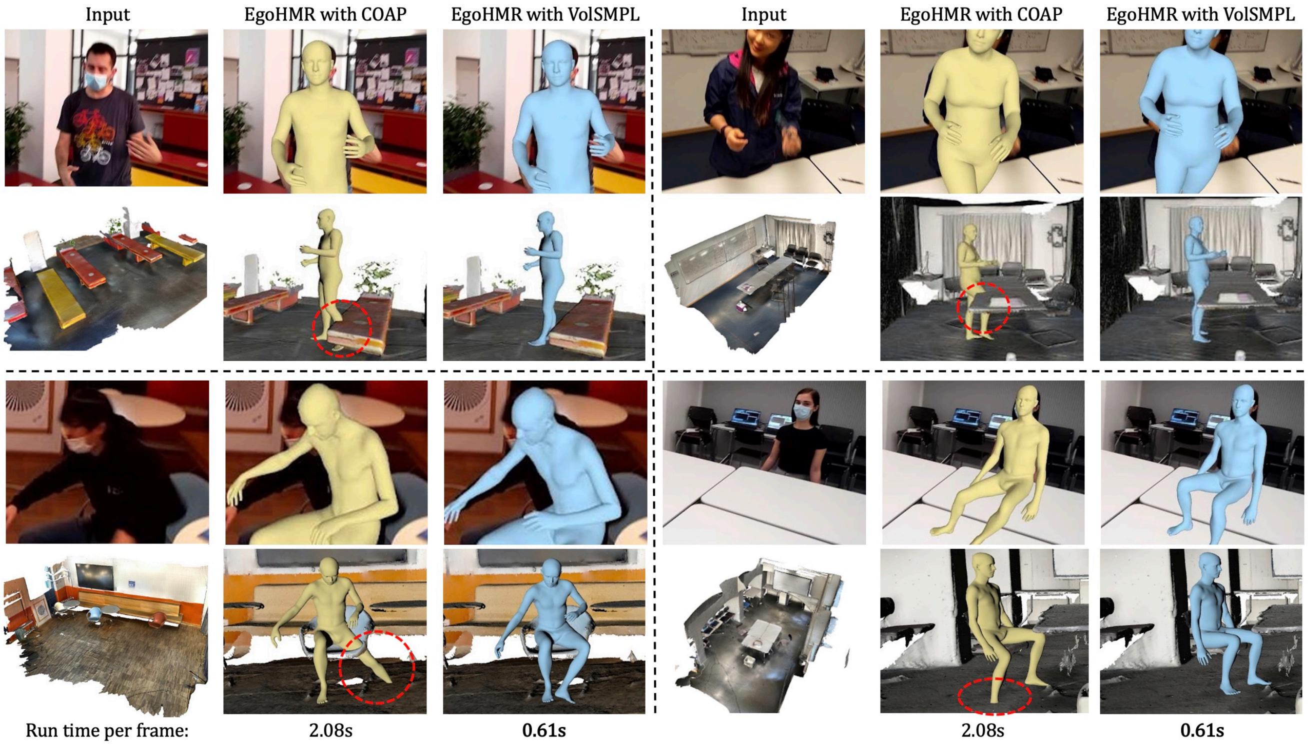

Figure C.1. **Human Mesh Recovery in 3D Scenes.** Given an egocentric image and the 3D scene mesh as input, EgoHMR with VolumetricSMPL (in blue) achieves fewer human-scene collisions than with COAP (in yellow) while being substantially faster (2.08s *vs.* 0.61s). The collisions are denoted by the red circles.

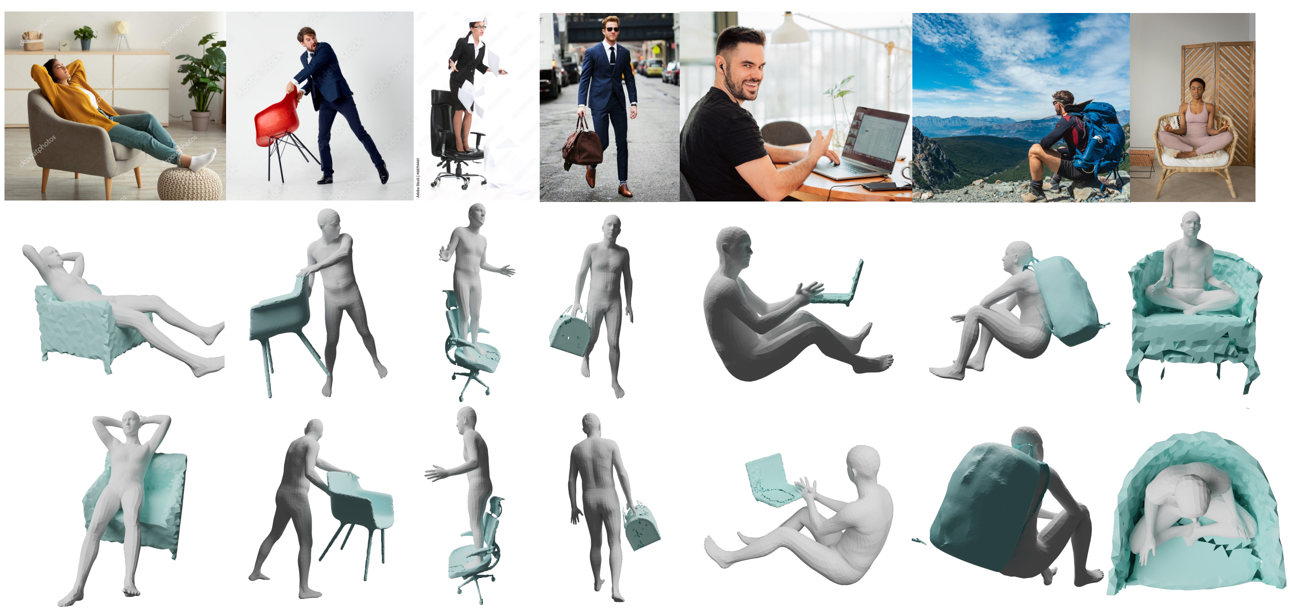

Figure C.2. **Reconstructing Human-Object Interactions from Images in the Wild.** Here we demonstrate how VolumetricSMPL can be integrated to reconstruct human-object interactions from images in the wild [59, 68]. VolumetricSMPL achieves comparable reconstruction quality while being significantly faster than calculating human mesh SDF. See more details in 4.2.1.

where E_θ and E_β are pose and shape prior terms, $\hat{\mathbf{J}}_{3D}$ represents the estimated 3D body joints, and Π is the perspective projection operator.

Initial Object Pose and Shape Estimation. We formulate object pose and shape estimation as a rendered shape-matching problem. First, we select an object mesh from a set of template meshes for each object category, choosing the one that best matches the corresponding 2D image.

Next, we use PointRend [30] to detect objects in the image, extracting their bounding boxes, segmentation masks, and semantic labels. Finally, we refine the 6-DoF object pose of the selected mesh using a differentiable renderer [25]. For further details, refer to PHOSA [68].

Human-Object Joint Optimization. The joint optimization process refines human and object scales, translations, and rotations by minimizing the following objective function:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{contact} + \lambda_2 \mathcal{L}_{normal} + \lambda_3 \mathcal{L}_{pen} + \lambda_4 \mathcal{L}_{scale}, \tag{D.2}$$

where $\mathcal{L}_{scale}$ penalizes deviations between the current human/object scale s'_c and the prior scale $\bar{s}_c$ obtained from large language models:

$$\mathcal{L}_{scale} = \|s'_c - \bar{s}_c\| . \tag{D.3}$$

The contact loss $\mathcal{L}_{contact}$ encourages plausible human-object interactions by minimizing the one-way Chamfer distance between contact pairs:

$$\mathcal{L}_{contact} = \sum_{(i,j)\in\mathcal{I}} \mathbb{1}(\mathbf{n}_{\mathcal{P}_h^i}, \mathbf{n}_{\mathcal{P}_o^j}) d_{\mathrm{CD}}(\mathcal{P}_h^i, \mathcal{P}_o^j). \tag{D.4}$$

The surface normal consistency loss $\mathcal{L}_{normal}$ enforces alignment between interacting human and object surfaces:

$$\mathcal{L}_{normal} = \sum_{(i,j)\in\mathcal{I}} \mathbb{1}(\mathbf{n}_{\mathcal{P}_h^i}, \mathbf{n}_{\mathcal{P}_o^j})(1 + d_{\cos}(\mathbf{n}_{\mathcal{P}_h^i}, \mathbf{n}_{\mathcal{P}_o^j})), \tag{D.5}$$

where $d_{\cos}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$ is the cosine similarity between two normal vectors.

The penetration loss $\mathcal{L}_{pen}$ prevents object vertices from penetrating the body mesh:

$$\mathcal{L}_{pen} = \frac{1}{|\mathcal{O}|} \sum\nolimits_{v\in\mathcal{O}} \mathrm{ReLU}(-f_\Theta^{\text{Vol.SMPL}}(v/s_c'^h|\theta, \beta)), \tag{D.6}$$

where $f_\Theta^{\text{Vol.SMPL}}$ represents the VolumetricSMPL body model, and $\mathcal{O}$ denotes the object mesh. The object vertex v is rescaled with $\frac{1}{s_c'^h}$, where $s_c'^h$ is the current human scale. Since VolumetricSMPL predicts a scale-invariant signed distance field, we apply this rescaling when querying the signed distance.

In our experiments, we set $\lambda_{1,..,4} = [1e5, 1e3, 1e4, 1e3]$. When using SDF-based optimization [59], we adjust $\lambda_3 = 1e3$ instead of $1e4$. The joint optimization runs for 1k steps. The number of object vertices varies from 1k to 20k, depending on the object category. We use the Adam optimizer with a learning rate of $2e-3$.

Additional visual results are displayed in Fig. C.2

D.2. Human Mesh Recovery in 3D Scenes

Given an egocentric image $\mathcal{I}$ containing a truncated human body and a corresponding 3D scene point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ in the camera coordinate system, where N is the number of scene points, EgoHMR aims to model the conditional distribution of human body poses $p(\theta|\mathcal{I}, \mathbf{P})$. The goal is to generate body poses that naturally interact with the 3D scene while aligning with the image observations. The body translation $\boldsymbol{\gamma}$ and shape parameters $\boldsymbol{\beta}$ are modeled deterministically. During diffusion inference, at each sampling step t, the denoiser D predicts the clean body pose $\hat{\theta}_0$ from the sampled noisy pose θ_t at timestep t:

$$\hat{\theta}_0 = D(\theta_t, t, \mathcal{I}, \mathbf{P}). \tag{D.7}$$

For further architecture details, refer to [69]. The predicted pose $\hat{\theta}_0$ is then noised back to θ_{t-1} using the DDPM sampler [21]:

$$\theta_{t-1} \sim \mathcal{N}(\mu_t(\theta_t, \hat{\theta}_0) + a\Sigma_t \nabla \mathsf{J}(\theta_t), \Sigma_t), \tag{D.8}$$

where $\mu_t(\theta_t, \hat{\theta}_0)$ is a linear combination of θ_t and $\hat{\theta}_0$, and Σ_t is a scheduled Gaussian distribution [21]. The sampling process is guided by the gradient of a collision score $\mathsf{J}(\theta)$, which mitigates human-scene interpenetrations. The guidance is modulated by Σ_t and a scale factor a.

For EgoHMR with COAP (corresponding to the experiment setup w. COAP in Tab. 4 of the main paper), the collision score is computed by checking whether each scene vertex is inside the human volume, using COAP [42]:

$$\mathsf{J}(\theta) = \frac{1}{|\mathbf{P}|} \sum\nolimits_{q\in\mathbf{P}} \sigma(f_\Theta^{\text{coap}}(q|\mathcal{G})) \mathbb{I}_{f_\Theta^{\text{coap}}(q|\mathcal{G})>0}, \tag{D.9}$$

where f_Θ^{coap} stands for the COAP body model, and $\sigma(\cdot)$ stands for the sigmoid function.

For EgoHMR with VolumetricSMPL (corresponding to the experiment setup w. Ours in Tab. 4 of the main paper), the collision score is computed using the signed distance field predicted by VolumetricSMPL for each scene vertex:

$$\mathsf{J}(\theta) = \frac{1}{|\mathbf{P}|} \sum\nolimits_{q\in\mathbf{P}} \mathrm{ReLU}(-f_\Theta^{\text{VolumetricSMPL}}(q|\mathcal{G})), \tag{D.10}$$

where $f_\Theta^{\text{VolumetricSMPL}}$ denotes the proposed VolumetricSMPL body model.

Experiment Details. In Eq. (D.8), we set the scale factor a to 0.4 for EgoHMR with COAP and 30 for EgoHMR with VolumetricSMPL. The diffusion sampling process consists of 50 steps, with collision score guidance applied only during the last 10 steps. In the final 5 denoising steps, we scale $\nabla \mathrm{J}(\theta_t)$ by a only, omitting Σ_t to prevent the collision guidance from diminishing too early in the process.

To ensure a fair comparison between COAP and VolumetricSMPL, we compute collision scores in Eq. (D.8) using 20k scene vertices sampled within a 2×2m cube centered around the human body.

For evaluation, we use the official checkpoint from [69] to perform diffusion sampling and evaluate on the EgoBody [71] test set, which consists of 62,140 frames. For the further details about the evaluation metrics we refer the reader to [69].

Additional visual results are displayed in Fig. C.1.

D.3. Scene-Constrained Human Motion Synthesis

We use DartControl [78] to generate scene-constrained navigation motion in 15 scanned scenes from the Egobody [71] dataset. Given the starting location and goal location in a 3D scene, we initialize the human with a standing pose and use the optimization-based motion synthesis method of DartControl to drive the human to reach the goal location while avoiding scene obstacles. The 3D scenes are represented as point clouds for collision evaluation, with each point cloud containing 16384 points sampled from the original scan using farthest point sampling. The motion sequences vary in length, ranging from 80 to 120 frames. We condition the locomotion style of all sequences using the text prompt "walk". The optimization objective for scene-constrained motion synthesis encourages the body pelvis of the last frame to reach the goal location and penalizes all detected human-scene collisions as follows:

$$\mathcal{L} = \mathcal{F}(\mathbf{p}, \mathbf{g}) + w * \mathcal{L}_{coll}, \tag{D.11}$$

where $\mathbf{p}$ denotes the last frame body pelvis, $\mathbf{g}$ denotes the goal location, $\mathcal{F}$ denotes the smooth L1 loss [14], w is a tunable weight for collisions, and $\mathcal{L}_{coll}$ is the scene collision term that we separately implement using COAP and VolumetricSMPL following prior task (Sec. 4.2.2).

We use a collision weight of $w = 1$ for the VolumetricSMPL collision term and conduct experiments with varying collision weights for the COAP baseline. Our observations reveal that the COAP baseline struggles to effectively balance accurate goal-reaching and collision avoidance, leading to performance that is inferior to VolumetricSMPL, as demonstrated in Tab. D.1. Notably, applying a large collision weight $w = 1$ for COAP disrupts the optimization process, leading to a failure to resolve collisions and causing deviations from the intended goal location.

Table D.1. Comparison of indoor navigation motion synthesis using COAP-based collision term with different weights and VolumetricSMPL-based collision term.

	Per-Frame ↓		Motion Quality ↓	
	Memory	Time	Collision	Goal Dist.
w. COAP ($w = 0.01$)	4.44 GB	26.48 ms	4.92 cm	0.02 m
w. COAP ($w = 0.1$)	4.44 GB	26.53 ms	2.78 cm	0.16 m
w. COAP ($w = 1.0$)	4.44 GB	26.77 ms	4.92 cm	0.57 m
w. Ours ($w = 1.0$)	**0.19 GB**	**3.78 ms**	**0.24 cm**	**0.01 m**

D.4. Self-Intersection Handling with VolumetricSMPL via Volumetric Constraints

When resolving self-intersections using volumetric constraints (Sec. 4.2.4), the model first detects potential collisions by enclosing each body part within a 3D bounding box $\mathcal{G}$. For every pair of intersecting boxes, the overlapping volume is identified, and 300 points are uniformly sampled within this region. These points are further refined by retaining only those that reside inside at least two body parts, as determined through part-wise SDF evaluations. The final set of valid intersection points is denoted as $\mathcal{S}$, and the self-intersection loss is computed according to Eq. (8).

In this experiment, we minimize the final loss term (Eq. 8) using the SGD optimizer to iteratively refine the pose parameters and resolve intersections effectively. The computational resources reported in Tab. 6 are estimated on an NVIDIA RTX 3090 GPU card.

E. Limitations and Future Work

While VolumetricSMPL achieves a 10× speedup and 6× lower memory usage compared to prior work [42], further optimizations remain an important direction. Currently, it supports batch sizes up to 80 for human-scene interaction tasks (Sec. 4.2.3) on a 24GB GPU, but this remains a bottleneck when modeling longer human motion sequences. Future work could explore memory-efficient architectures to further scale motion synthesis.

Additionally, similar to other volumetric body models [42, 43], VolumetricSMPL does not explicitly model detailed hand articulation, primarily due to limitations in available training data. A potential extension involves developing a specialized volumetric hand model and integrating it into our framework, enabling more precise hand-object interactions, particularly in fine-grained manipulation tasks.

Beyond human modeling, our NBW formulation is inherently generic and can be applied to non-human shapes. Exploring its potential for learning articulated animal models, robotic structures, or generic deformable objects could extend its applicability beyond human-centric tasks. We leave this exploration for future work.

By addressing these limitations, VolumetricSMPL could further improve efficiency, extend its scope to finer interactions, and generalize beyond human body modeling to broader applications in graphics, robotics, and virtual environments.

Broader Impact. Beyond the immediate applications explored in this work, VolumetricSMPL has the potential to serve as a valuable tool for the broader research community. Its efficiency, scalability, and ease of integration make it suitable for a wide range of interaction tasks. By providing an open-source implementation, we aim to facilitate further research into volumetric representations, encourage new applications in dynamic human-scene interactions, and inspire extensions to non-human shapes.

We hope that VolumetricSMPL will enable researchers and practitioners to advance human body modeling research and its downstream applications.

F. Seamless SMPL Code Integration

VolumetricSMPL is a lightweight and user-friendly add-on module for SMPL-based body models, enabling seamless volumetric extension.

With just a single line of code, users can extend SMPL models with volumetric functionalities. After completing the forward pass, they gain access to key volumetric functionalities, including SDF queries, self-intersection loss, and collision penalties. This implementation maintains full compatibility with existing SMPL-based reconstruction and perception applications.

The following code snippet demonstrates how to install VolumetricSMPL and integrate it with an SMPL model to utilize its volumetric functionalities:

```
pip install VolumetricSMPL
```

Listing 1. VolumetricSMPL Installation via PyPi

```
import smplx
from VolumetricSMPL import attach_volume

# Create an SMPL body
model = smplx.create(**smpl_parameters)

attach_volume(model) # extend with VolumetricSMPL

# SMPL forward pass
smpl_output = model(**smpl_data)

# Access volumetric functionalities
# 1) Query SDF for given points
model.volume.query(smpl_output, scan_points)

# 2) Compute self-intersection loss
model.volume.selfpen_loss(smpl_output)

# 3) Compute collision loss
model.volume.collision_loss(smpl_output, points)
```

Listing 2. Integrating VolumetricSMPL with SMPL.

The `attach_volume()` function seamlessly extends any *SMPL, SMPL-H, or SMPL-X* model with volumetric capabilities. Once the full forward pass is completed, users can efficiently compute signed distance field (SDF) queries, self-penetration loss, and collision penalties for physically plausible human interactions.

VolumetricSMPL is released under the MIT license and will be publicly available to the research community.

References

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, 2015. 5, 6

[2] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imGHUM: Implicit generative models of 3d human shape and articulated pose. In *ICCV*, 2021. 2, 3, 5, 6, 1

[3] Rayan Armani, Changlin Qian, Jiaxi Jiang, and Christian Holz. Ultra inertial poser: Scalable motion capture and tracking from sparse inertial sensors and ultra-wideband ranging. In *ACM SIGGRAPH*, 2024. 2, 3

[4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *CVPR*, 2022. 3

[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 *ECCV*, 2016. 3

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, 2017. 5, 6

[7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 3

[8] Zhi-Quan Cheng, Yin Chen, Ralph R Martin, Tong Wu, and Zhan Song. Parametric modeling of 3d human body shape—a survey. *Computers & Graphics*, 2018. 3

[9] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 3

[10] Sisi Dai, Wenhao Li, Haowen Sun, Haibin Huang, Chongyang Ma, Hui Huang, Kai Xu, and Ruizhen Hu. Interfusion: Text-driven generation of 3d human-object interaction. In *ECCV*, 2025. 2, 3

[11] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA: Neural Articulated Shape Approximation. In *ECCV*, 2020. 3, 5, 1

[12] Mihai Fieraru, Mihai Zanfir, Teodor Szente, Eduard Bazavan, Vlad Olaru, and Cristian Sminchisescu. Remips: Physically consistent 3d reconstruction of multiple interacting people under weak supervision. *NeurIPS*, 2021. 2, 3

[13] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. MoVi: A large multipurpose motion and video dataset. *arXiv preprint arXiv:2003.01888*, 2020. 5, 6

[14] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 6

[15] Hengtao Guo, Benjamin Planche, Meng Zheng, Srikrishna Karanam, Terrence Chen, and Ziyan Wu. Smpl-a: Modeling person-specific deformable anatomy. In *CVPR*, 2022. 2

[16] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *ICLR*, 2017. 3

[17] Si Hang. Tetgen, a delaunay-based quality tetrahedral mesh generator. *TOMS*, 2015. 3

[18] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*, 2019. 2, 3, 8

[19] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. Populating 3d scenes by learning human-scene interaction. In *CVPR*, 2021. 2

[20] Nikolas Hesse, Sergi Pujades, Michael J Black, Michael Arens, Ulrich G Hofmann, and A Sebastian Schroeder. Learning and tracking the 3d body shape of freely moving infants from rgb-d sequences. *PAMI*, 2019. 2

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 5

[22] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *TOG*, 2013. 2, 3

[23] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020. 3

[24] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *CVPR*, 2024. 3

[25] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 5

[26] Rawal Khirodkar, Shashank Tripathi, and Kris Kitani. Occluded human mesh recovery. In *CVPR*, 2022. 3

[27] Hyeonwoo Kim, Sookwan Han, Patrick Kwon, and Hanbyul Joo. Beyond the contact: Discovering comprehensive affordance for 3d objects from pre-trained 2d diffusion models. In *ECCV*, 2024. 2, 3

[28] Meekyoung Kim, Gerard Pons-Moll, Sergi Pujades, Seungbae Bang, Jinwook Kim, Michael J Black, and Sung-Hee Lee. Data-driven physics for human soft tissue animation. *TOG*, 2017. 3

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[30] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 5

[31] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regressor for 3D human body estimation. In *ICCV*, 2021. 3

[32] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru, and Cristian Sminchisescu. Dreamhuman: Animatable 3d avatars from text. In *NeurIPS*, 2023. 2

[33] Gen Li, Kaifeng Zhao, Siwei Zhang, Xiaozhong Lyu, Mihai Dusmanu, Yan Zhang, Marc Pollefeys, and Siyu Tang. Egogen: An egocentric synthetic data generator. In *CVPR*, pages 14497–14509, June 2024. 2, 3

[34] Ronghui Li, Hongwen Zhang, Yachao Zhang, Yuxiang Zhang, Youliang Zhang, Jie Guo, Yan Zhang, Xiu Li, and Yebin Liu. Lodge++: High-quality and long dance generation with vivid choreography patterns. *arXiv preprint arXiv:2410.20389*, 2024. 3

[35] Ronghui Li, Youliang Zhang, Yachao Zhang, Yuxiang Zhang, Mingyang Su, Jie Guo, Ziwei Liu, Yebin Liu, and Xiu Li. Interdance: Reactive 3d dance generation with realistic duet interactions. *arXiv preprint arXiv:2412.16982*, 2024. 2, 3

[36] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and ex-

pression from 4D scans. *SIGGRAPH Asia*, 2017. 3
[37] Yuwei Li, Longwen Zhang, Zesong Qiu, Yingwenqi Jiang, Nianyi Li, Yuexin Ma, Yuyao Zhang, Lan Xu, and Jingyi Yu. Nimble: a non-rigid hand model with bones and muscles. *TOG*, 2022. 3
[38] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *TOG*, 2015. 2, 3, 4, 5, 1
[39] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *NeurIPS*, 2021. 3
[40] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 5
[41] Marko Mihajlovic, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. ResFields: Residual neural fields for spatiotemporal signals. In *ICLR*, 2024. 3, 5
[42] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. Coap: Compositional articulated occupancy of people. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 8
[43] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *CVPR*, 2021. 2, 3, 5, 6, 1
[44] 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. 4, 5, 6, 1
[45] Lea Müller, Vickie Ye, Georgios Pavlakos, Michael Black, and Angjoo Kanazawa. Generative proxemics: A prior for 3d social interaction from images. In *CVPR*, 2024. 2, 3
[46] Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *ECCV*, 2020. 2
[47] Pablo Palafox, Aljaz Bozic, Justus Thies, Matthias Nießner, and Angela Dai. Neural parametric models for 3d deformable shapes. In *ICCV*, 2021. 3
[48] Pablo Palafox, Nikolaos Sarafianos, Tony Tung, and Angela Dai. Spams: Structured implicit parametric models. In *CVPR*, 2022. 3
[49] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 2, 3, 4, 8, 1
[50] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 2017. 3
[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2, 4, 1
[52] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, 2021. 2
[53] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *SIGGRAPH Asia*, 2017. 3
[54] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 3
[55] Cristian Sminchisescu and Alexandru C Telea. Human pose estimation from silhouettes. a consistent approach using distance level sets. In *WSCG*, 2002. 2
[56] Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3d human mesh from monocular images: A survey. *PAMI*, 2023. 2, 3, 1
[57] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *ECCV*, 2018. 3
[58] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. In *NeurIPS*, 2021. 2, 3
[59] Xi Wang, Gen Li, Yen-Ling Kuo, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. Reconstructing action-conditioned human-object interactions using commonsense knowledge priors. In *3DV*, 2022. 1, 2, 3, 6, 7, 4, 5
[60] Songpengcheng Xia, Yu Zhang, Zhuo Su, Xiaozheng Zheng, Zheng Lv, Guidong Wang, Yongjie Zhang, Qi Wu, Lei Chu, and Ling Pei. Envposer: Environment-aware realistic human motion estimation from sparse observations with uncertainty modeling. *arXiv preprint arXiv:2412.10235*, 2024. 2, 3
[61] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, 2022. 5
[62] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *NeurIPS*, 2021. 2
[63] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *CVPR*, 2020. 2, 3, 4
[64] Ye Yuan, Xueting Li, Yangyi Huang, Shalini De Mello, Koki Nagano, Jan Kautz, and Umar Iqbal. Gavatar: Animatable 3d gaussian avatars with implicit mesh learning. *CVPR*, 2024. 2
[65] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *CVPR*, 2023. 2, 3
[66] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *CVPR*, 2021. 3
[67] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE TNNLS*, 2012. 2
[68] 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*, 2020. 1, 2, 3, 6, 4, 5
[69] Siwei Zhang, Qianli Ma, Yan Zhang, Sadegh Aliakbarian, Darren Cosker, and Siyu Tang. Probabilistic human mesh recovery in 3d scenes from egocentric views. In *ICCV*, 2023. 1, 2, 3, 7, 5, 6
[70] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In *ECCV*, 2022. 3
[71] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein

Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In *ECCV*, 2022. 7, 6

[72] Siwei Zhang, Yan Zhang, Federica Bogo, Pollefeys Marc, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *ICCV*, 2021. 2

[73] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *3DV*, 2020. 2

[74] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *ECCV*, 2022. 3

[75] Yan Zhang, Michael J. Black, and Siyu Tang. We are more than our joints: Predicting how 3D bodies move. In *CVPR*, 2021. 2

[76] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J Black, and Siyu Tang. Generating 3d people in scenes without people. In *CVPR*, 2020. 2, 3

[77] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *CVPR*, 2022. 3

[78] Kaifeng Zhao, Gen Li, and Siyu Tang. A diffusion-based autoregressive motion model for real-time text-driven motion control. In *ICLR*, 2025. 1, 2, 7, 8, 6

[79] Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *ECCV*, 2022. 3

[80] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. In *ICCV*, 2023. 3