# Appendix

## A. Experiment Details

### A.1. From PROX-Qualitative to PROX-E

The PROX-Qualitative (or **PROX-Q** for short) dataset comprises recordings of 20 subjects in 12 indoor scenes, including 3 bedrooms, 5 living rooms, 2 sitting booths and 2 offices. The 3D scenes were scanned with a commercial Structure Sensor RGB-D camera and reconstructed by the accompanying 3D reconstruction solution Skanect. We refer to [18] for more details of how the **PROX-Q** was established. Note that the scene meshes of **PROX-Q** do not form valid rooms, i.e. there is no ceiling and some walls are missing. Furthermore, the meshes are not semantically segmented.

To our knowledge, **PROX-Q** is the largest dataset capturing real human-scene interactions at the 3D mesh level. However, due to the incomplete room scans and lack of mesh semantics, we extend **PROX-Q** from the following perspectives, so as to serve our purposes of human-scene interaction modeling and generation from the viewpoint of an embodied agent:

**(1) Building up virtual walls, floors, ceilings.** To achieve this goal, we import the scene meshes of **PROX-Q** into Blender, which we use to enclose the original scene meshes to create rooms. When using a camera to capture the scene, we can always get a completed depth map. The completed depth maps are illustrated in Fig. 3.

**(2) Semantic annotation of the scene meshes.** The mesh semantics follow the Matterport3D dataset [3], which incorporates 40 categories of common indoor objects [3]. Our annotation is performed manually, and the mesh vertex color denotes the object labels. Therefore, we are able to capture the depth and the semantics from multiple views.

**(3) Setting up virtual cameras.** The original **PROX-Q** dataset only incorporates video recordings from a single view in each scene. This implies that we can only have 12 depth-semantics pairs to use, and hence can lead to severe overfitting when using PROX-Q for training. To overcome this drawback, for each individual frame captured by the real camera, we create a set of virtual cameras in the scene to capture the human behavior. The virtual cameras are posed according to the room structure and the human body position. Specifically, we create a 3D grid according to the room size. The range of width and length is determined by the size of the room. The range of height is between the pelvis of the human body and the ceiling height that we have created. For each camera, the X-axis is parallel to the ground, and the Z-axis is towards the human body center. Next, we add Gaussian noise on the camera translations, and discard views with no human bodies or strong body occlusions; i.e., in the image the body part around the pelvis ($\pm$ 10 pixels) is not occluded by any object in the scene. We argue that such noise is essential. Otherwise the generated human bodies will always be located in the center of the depth-semantics maps. Furthermore, we only keep the virtual cameras with the distance to the human body between 1.65m and 6.5m, so that the projected body sizes to the virtual cameras are similar to the body sizes captured by real cameras. Fig. S1 shows a set of virtual cameras before and after applying the Gaussian noise to the camera translations. Moreover, the resolution of depth and semantics is set to 480$\times$270, and the camera intrinsic parameters are

$$K = \begin{pmatrix} 233.826 & 0 & 239.5 \\ 0 & 233.826 & 134.5 \\ 0 & 0 & 1 \end{pmatrix}, \tag{9}$$

which is a default setting in Open3D [54] after specifying the depth/semantics resolution.

### A.2. Creating the MP3D-R dataset

Our **MP3D-R** dataset is extracted from the Matterport3D dataset [3]. We extract the 7 rooms by annotating bounding boxes of regions, as shown in Tab. S1. These 7 rooms have room types that are similar to **PROX-E**.

When trimming the rooms according to the annotation, we expand the annotated bounding box size by 0.5 meters to ensure that walls, ceilings and floors are incorporated. Note that, the Habitat simulator and the original Matterport3D dataset have different gravity directions. The Habitat simulator assumes that the gravity direction is along $-Y$. Thus, after loading the scene meshes from Matterport3D, we rotate the scene mesh by $-90$ degree w.r.t. the X-axis to match the bounding box annotation from Habitat. Fig. S2 shows some retrieved room meshes with the world coordinate origins.

[3] One can see the object categorization via: https://github.com/niessner/Matterport/blob/master/metadata/mpcat40.tsv

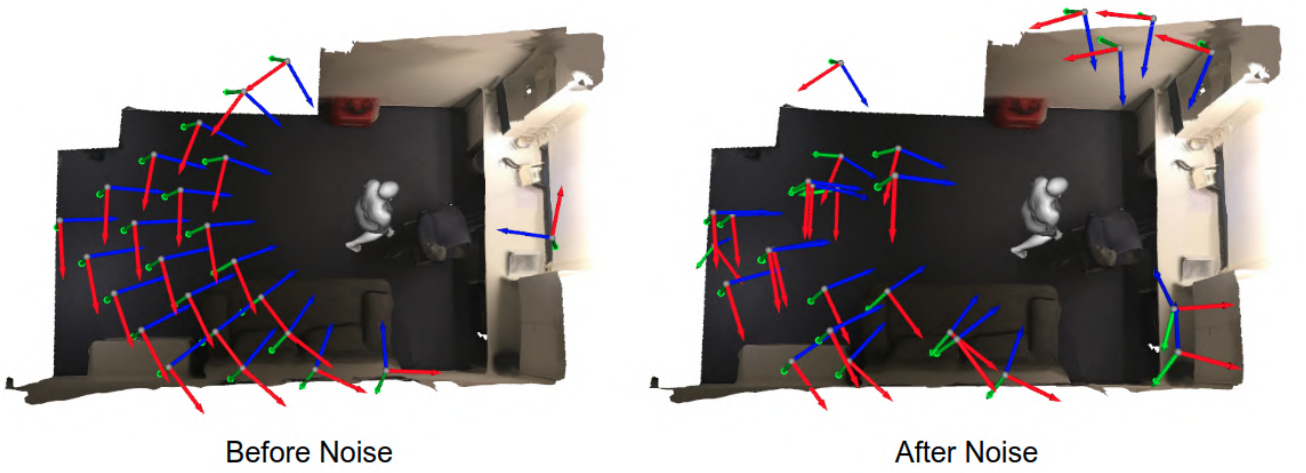


Figure S1: Illustration of the virtual cameras before and after applying the Gaussian noise to the camera translation. The X, Y and Z axes of each camera are denoted by red, green and blue, respectively.

Table S1: The seven rooms in **MP3D-R**, retrieved from the Matterport3D dataset [3].

| scan ID | region ID | room type |
|---|---|---|
| 17DRP5sb8fy | 0-0 | bedroom |
| 17DRP5sb8fy | 0-8 | family room |
| 17DRP5sb8fy | 0-7 | living room |
| sKLMLpTHeUy | 0-1 | family room |
| X7HyMhZNoso | 0-16 | living room |
| zsNo4HB9uLZ | 0-0 | bedroom |
| zsNo4HB9uLZ | 0-13 | living room |

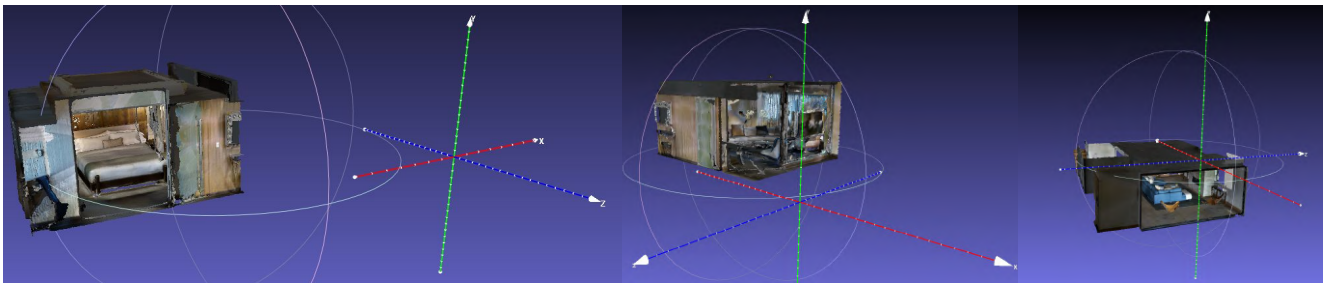

Figure S2: Three examples of the rooms in **MP3D-R**. One can see the world origins, and the gravity direction is along $-Y$.

Next, we use the Habitat simulator [32] to create a virtual agent in the room. In each scene, we first put the agent in the room center, and then manipulate that virtual agent to cruise around the room. According to ranges of virtual cameras in **PROX-E**, we set the height of agent sensor to 1.8 meters from the ground. For each snapshot, we record the RGB image, the scene depth, the scene semantics, as well as the camera extrinsic parameters. The frame resolution and the camera intrinsics are identical to our settings in **PROX-E**.

Following the pipeline for creating **PROX-Q**, we also compute scene signed distance functions (SDFs) of the **MP3D-R** scenes. For each room, we first use Poisson surface reconstruction to convert the meshes to be watertight. Fig. S3 shows an example of the reconstructed scene mesh. Next, similar to [18, Sec. 3.6] we compute the SDF in a uniform voxel grid of size $256 \times 256 \times 256$ which spans a padded bounding box of the scene mesh.

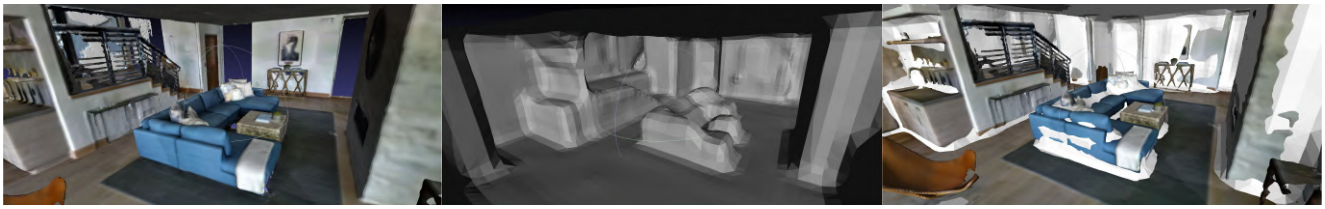


Figure S3: From left to right: The original scene mesh, the mesh after Poisson surface reconstruction, and their overlay.

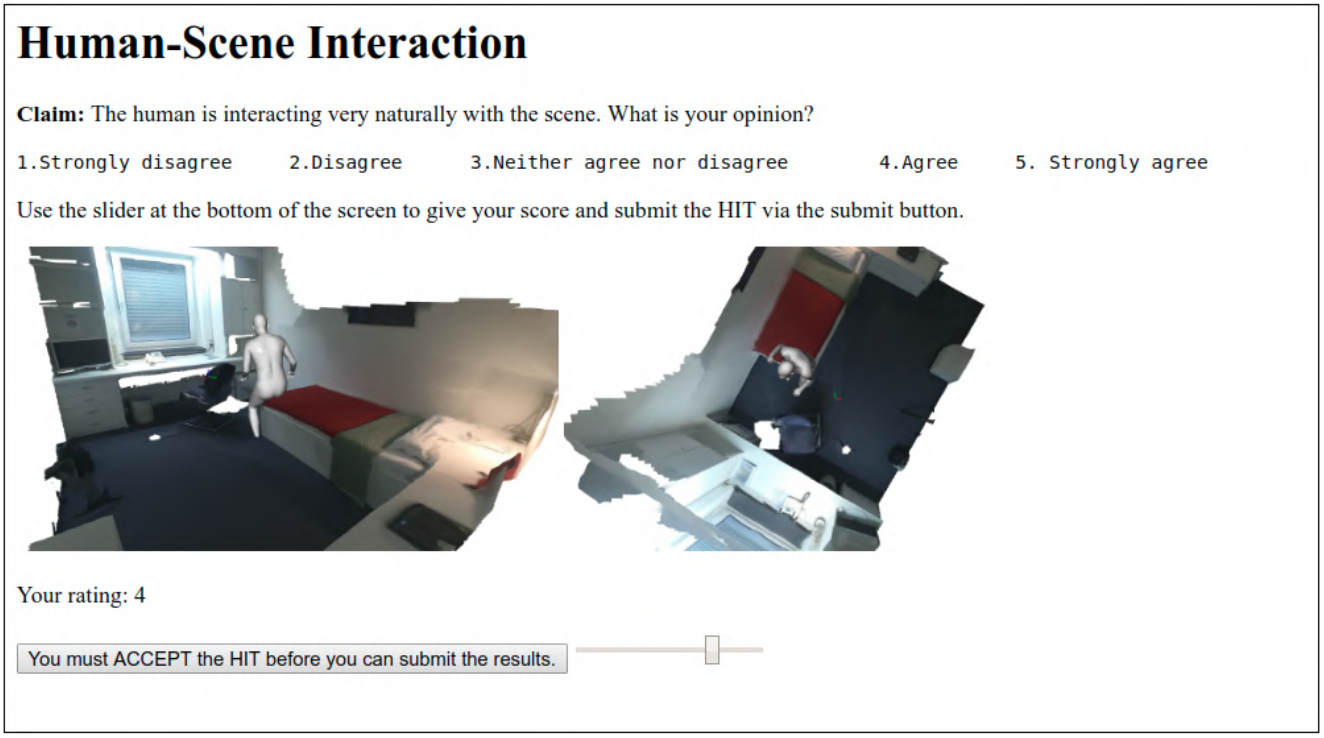


Figure S4: The user interface of our user study, in which the users are requested to rate how naturally the human is interacting with the environment.

### A.3. Details of the baseline method

To our knowledge, the most related work is Li et al. [28], which aims to put humans in a scene and infer the affordance of 3D indoor environments. In this work, the authors first propose an efficient and fully-automatic 3D human pose synthesizer to generate stick figures, using a pose prior learned from a large-scale 2D dataset [46] and the physical constraints from the target 3D scenes. With this pose synthesizer, the authors create a dataset incorporating synthesized human-scene interactions. Next, based on the synthesized dataset, the authors develop a generative model for 3D affordance prediction, which is able to generate body stick figures based on the scene images.

Compared to the method of Li et al. [28], our solution has the following key differences: (1) Our **PROX-E** dataset contains *real* human-scene interactions rather than synthesized ones. This is highly beneficial to model the distribution of human-scene interactions in the real world. (2) Our solution is to generate body meshes rather than 3D body stick figures (See Fig. 5 in [28]). Therefore, the results can be directly used in applications like VR, AR and others. (3) We use the SMPL-X model [36] in our work, hence our methods can generate various body shapes and fine-grained hand poses, beyond the body global configurations and local poses. (4) SMPL-X can be regarded as a differentable function mapping from human body features to human body meshes, so the physical constraints applied on the body mesh surfaces can be back-propagated to the

body features like in [18]. (5) We use scene depth and semantics to represent the scene, rather than using RGB (or RGBD) images as in [28]. In our study, the RGB images are only available from the real cameras of **PROX-E**, because using RGB images can increase the risk of overfitting. In addition, the benefits of scene depth and semantics are revealed in [52].

Therefore, in our work, we modify the method of Li at al. [28] as mentioned in Sec. 4.1.2, so that their model can generate body meshes like our method, and a fair comparison can be conducted. We treat the modified version of [28] as our baseline. We train the baseline model with the **PROX-E** dataset like training our models. After generating body meshes in test scenes, we also apply our scene geometry-aware fitting to refine the results of the baseline model. The qualitative results of the baseline with fitting are shown in **MP3D-R**. We argue that our modification is necessary and favorable to the baseline to produce high quality 3D human bodies. For the quantitative comparison, please refer to Tab. 2, Tab. 3 and Tab. 4.

### A.4. Details of the user study

To evaluate how naturally the human body meshes are posed in the scenes, we perform a user study via Amazon Mechanical Turk (AMT). For each generated body pose, we render images of the body-and-scene mesh from two different views. Fig. S4 shows our AMT user interface. We propose a hypothesis that the human body interacts with the scene in a very natural manner, and then let users judge this hypothesis. Their judgements are recorded on a 5-point Likert scale.

Unlike the user study in [46, 28], we do not show pairs of results from different methods in the user interface. Instead, we have multiple methods to compare, and hence let the Turker evaluate each individual result in order to keep the user interface clear. Also, we report the standard errors of the user study results in addition to the mean values, which indicate how reliable the semantic scores are. One can see in Tab. 4 that the scene geometry-aware fitting can reduce the the standard error, indicating that users tend to give more consistent judgements. The ground truth has the lowest standard error, which indicates that Turkers are able to judge when the human-scene interaction is natural.

### A.5. More discussions on model training

We discussed loss weights and training schemes in Sec. 3.4. The weights are determined empirically: First, the KL-divergence weight is 0.1 for better representation power of the latent variables, as indicated in [20, 2]. The annealing scheme effectively avoids a collapsed VAE posterior, which outputs a constant result no matter how the latent variable varies. Second, the VPoser weight 0.001 is determined referring to [36, 18], to balance plausibility and variability of generated body poses. A too small weight increases body pose variations but can lead to implausible body poses (e.g. twisted legs). Additionally, in our trials the $\mathcal{L}_{HS}$ weights are set to avoid overfitting. Also, enabling $\mathcal{L}_{HS}$ earlier during training causes bad body reconstruction, since the modified Chamfer loss in Eq. (7) can pull the preliminary reconstructed body mesh to the closest scene mesh vertices. Moreover, the scene geometry-aware fitting loss weights are larger than the training loss weights, so as to reduce number of iterations in optimization while retaining the quality.

## B. Generative Model Latent Space Analysis

We show how the body smoothly changes in Fig. S5. Note that the results are *without* scene geometry-aware fitting. First, Fig. S5 indicates that our model effectively learns natural human-scene interactions. It shows that the generated body tends to stand when located on the floor, touches the desk (Fig. S5 (2)), and sits on the bed (Fig. S5 (4)) when located close to the furnitures. Second, the body configurations are disentangled in the latent space to some extent. For example, the body in Fig. S5 (2) mainly moves along the X direction in the world coordinate, while the body in Fig. S5 (3) mainly moves along the Y direction. Third, the body pose is less plausible when its latent variable is far away from zero. This is similar to VPoser [36], Tab. 5 in the manuscript shows the benefits of our model used as a scene-dependent pose prior.

## C. More Qualitative Results and Failure Cases

Fig. S6 and Fig. S7 show qualitative results in the test scenes of **PROX-E**. Fig. S8 and Fig. S9 show qualitative results of the generative models and the scene geometry-aware fitting in **MP3D-R**.

We find that failure cases can be categorized to two cases: First, the generative model is not always reliable in test scenes, since samples from the model are not always plausible. Some results sampled from the generative model cannot match the geometric structures in the test scenes, and hence the body floats in the air, or collides with the scene mesh. See Fig. S10 for examples. Such failure cases can occur in both the baseline and our methods. Second, although the scene geometry-aware fitting can effectively resolve floating and collision, its optimization process cannot simulate all real physics such as gravity and elasticity. Therefore, it could hurt the human-scene interaction semantics of the results produced by the generative model.

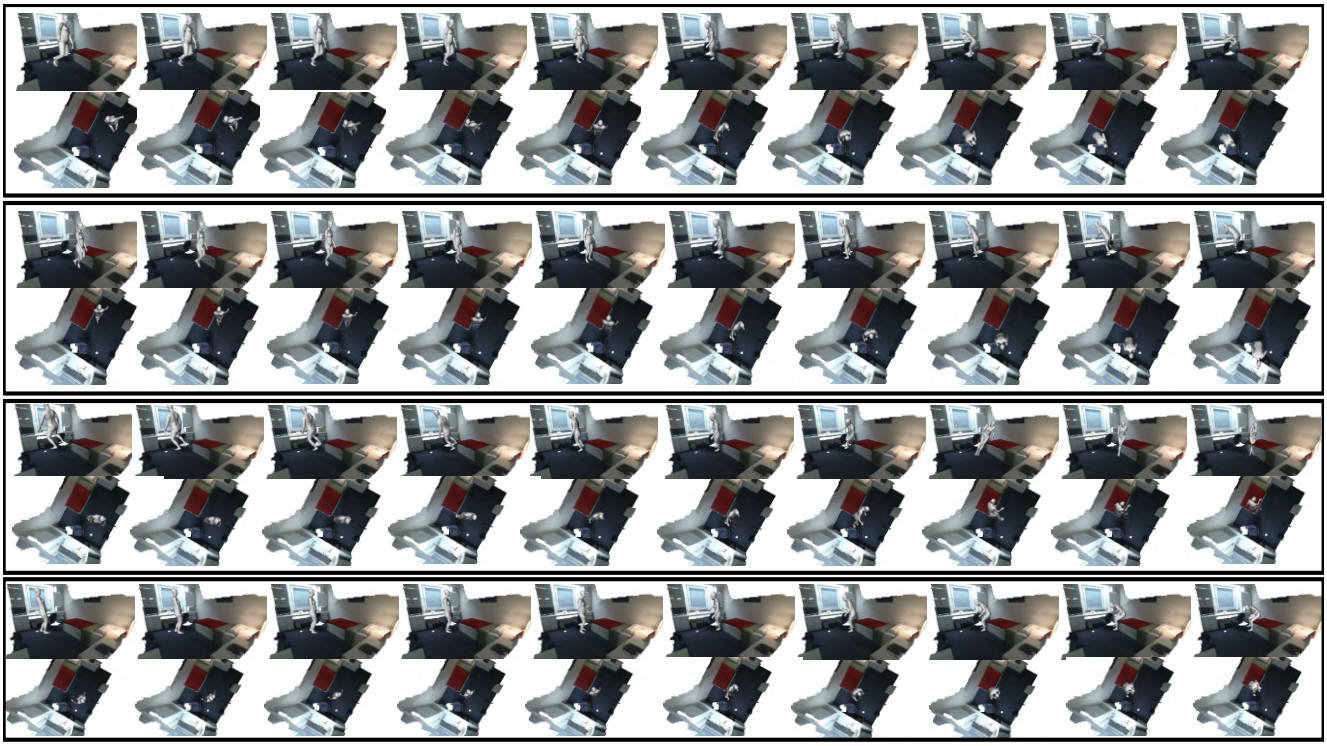

Figure S5: Illustration of the 32D latent space in the one-stage model. We regularly sample points along a line ranging from -3 to 3, and show the body meshes from two views. From top to bottom: (1) All dimensions of the line change. (2) The first 16 dimensions change, and the rest are zero. (3) Only the last 16 dimensions change. (4) Only the middle 16 dimensions change.

Fig. S11 shows some examples of such failure cases, which contain abnormal body global configurations and human-scene contact caused by our scene geometry-aware fitting.

Moreover, we discover that the quality of generated bodies also depends on the test scene data quality. For example, we have observed many low-quality generations in **MP3D-R**, which has more complex scene structures and noisy scans (unknown surfaces floating in the air) than **PROX-E**. Noisy scans can lead to noisy depth and semantic segmentation, and complex geometric structures can make geometry-aware fitting fail.

## D. Details of Scene-aware 3D Body Pose Estimation

In the experiment presented in Sec. 4.2, we follow the work of [18], and modify its Eq. (1) to incorporate our learned scene-dependent pose prior. The equation (1) in [18] is given by

$$\begin{aligned} E(\beta, \theta, \varphi, \gamma, M_s) = & E_J + \lambda_D E_D + \lambda_{\theta_b} E_{\theta_b} + \lambda_{\theta_f} E_{\theta_f} + \lambda_{\theta_h} E_{\theta_h} \\ & + \lambda_\alpha E_\alpha + \lambda_\beta E_\beta + \lambda_\epsilon E_\epsilon + \lambda_\epsilon E_\epsilon \\ & + \lambda_P E_P + + \lambda_C E_C, \end{aligned} \tag{10}$$

the notations of which are referred to [18]. In our work, we only modify the Vposer regularizer, i.e., $E_{\theta_b} = \|\theta_b\|_2^2$, and leave the other terms unchanged. Specifically, we change it to

$$E_{\theta_b} = \|\theta_b - \theta_b^s\|_2^2, \tag{11}$$

in which $\theta_b^s$ is our scene-dependent pose prior. We have demonstrated how to derive the $\theta_b^s$ in Sec. 4.2. During optimization, the initial pose feature is set to $\theta_b^s$, instead of a zero vector as in [18]. In our trials, changing the weight to $1.5\lambda_{\theta_b}$ yields a better performance.


Figure S6: Qualitative results of the baseline method in **PROX-E**. The results before and after the scene geometry-aware fitting are shown.


Figure S7: Qualitative results of S1 in **PROX-E**. The results before and after the scene geometry-aware fitting are shown.

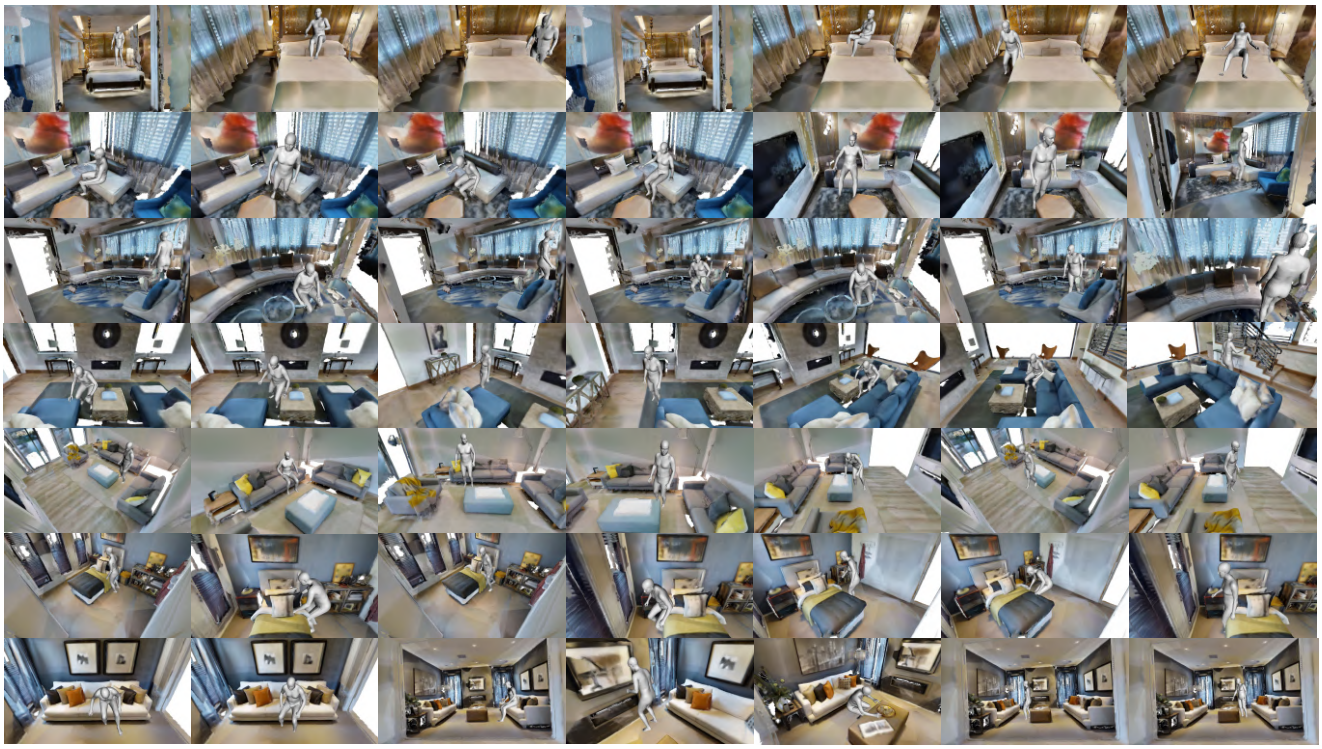

Figure S8: Qualitative results of the baseline with fitting in **MP3D-R**. We argue that our modifications to [28] are necessary and favorable to produce high quality 3D human bodies. For the quantitative comparison, please refer to Tab. 2, Tab. 3 and Tab. 4

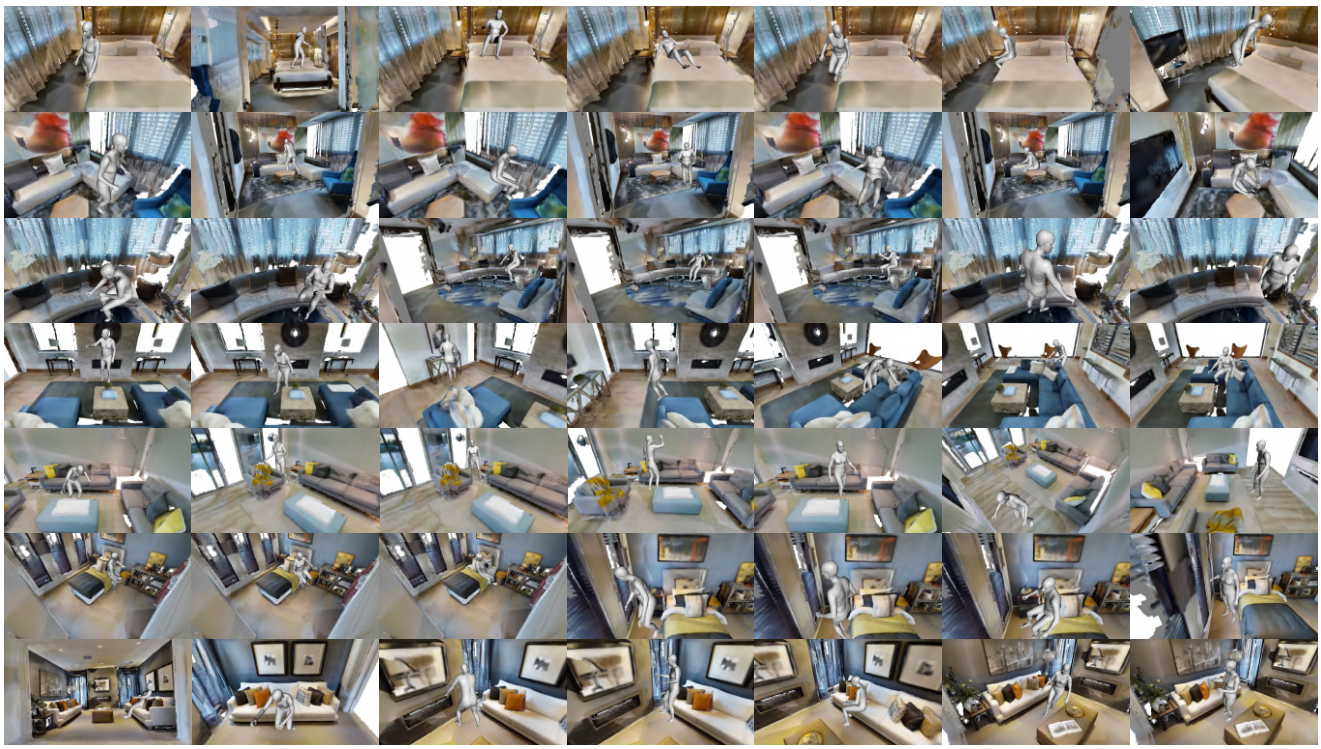

Figure S9: Qualitative results of S1 with fitting in **MP3D-R**.

Figure S10: Failure cases of body mesh generation. One can see the body floating and colliding with the scene mesh, which are implausible in the real world.

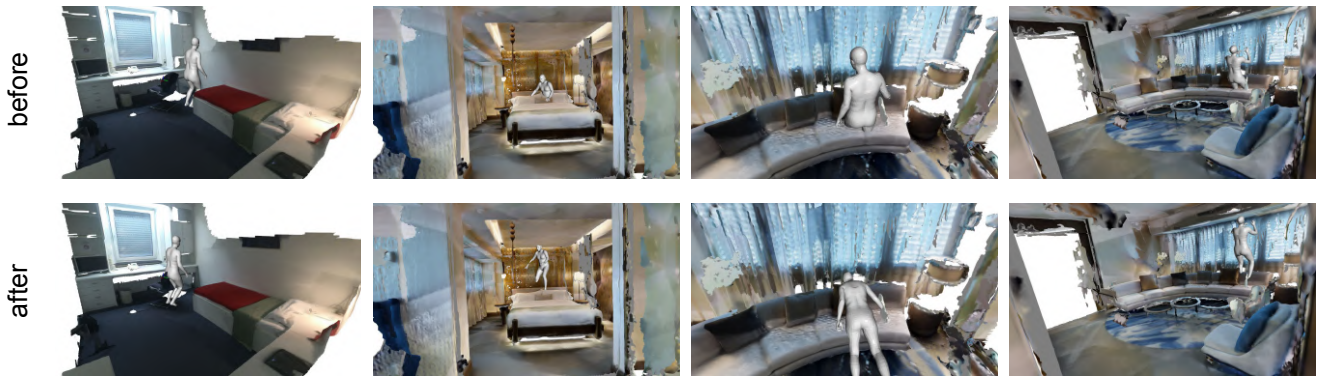


Figure S11: Failure cases of the scene geometry-aware fitting, for which results before and after the fitting are presented. One can see abnormal body translation, rotation and body-scene contact in the real world.