A. Additional Details of Experiments

This section offers a detailed overview of our experimental setup, including the implementation specifics of our method and the necessary adaptations made to baseline methods to ensure fair comparisons. We also elaborate on how our sampling strategy enables real-time control over character motion. For additional qualitative insights and extensive zero-shot transfer experiments, we direct readers to the accompanying *Supplementary Video*.

A.1. Experiment Settings

Our experimental setup for motion synthesis methods includes two distinct settings: *static* and *dynamic*. In the *static* setting, we focus on evaluating the quality of locomotion and scene-level interactions. Each test scene features five predefined pairs of start and end points, given as (x, y) coordinates in a z-up world coordinate system. These points, often located on furniture like chairs, test the method's ability to produce scene-appropriate motions. For complete trajectories, we generate midpoints using a generative A* path planning method, following Wang et al. [54].

The *dynamic* setting involves five pairs of object and human starting locations, accompanied by a trajectory leading towards the object. Each method is tasked with creating five unique motion variations that both approach and interact with the object, conforming to a designated action type.

A.2. Implementation Details

Our motion generation model utilizes a DDPM architecture with a linear variance scheduler, conditioned on scene and action embeddings. Following Guo et al. [13], we implement a Transformer encoder as the UNet structure with an embedding dimensionality of 512 for projecting body joint locations into a high-dimensional space. The Transformer consists of 6 layers, 16 heads in the multi-head attention, and uses a 0.1 dropout rate. The intermediate feedforward network's dimensionality is set to 1024. Both scene and action encoders are Transformer-based with 6 layers and 16 heads, producing 512-dimensional embeddings. These embeddings are added to the first token of the motion generation transformer, which contains timestep information. The Huber loss is used to gauge the accuracy of predicted noise.

For converting joint locations to SMPL-X parameterized meshes, a pre-trained 4-layer MLP predicts coarse SMPL-X parameters, with a 6D representation for rotation [65]. The MLP inputs three consecutive frames and outputs parameters for the middle frame. Edge cases use duplicated middle frames for input. An optimization process refines body poses using gradient descent on the L2 error between joint locations from the model and predicted SMPL-X parameters, ensuring accurate body pose representation.

Training with the Adam optimizer (learning rate 1e-4, batch size 512) on four NVIDIA A800 GPUs, our method takes 48 hours to train for 500k steps on TRUMANS.

A.3. Adaption of Baselines

We adapted baseline methods for a fair comparison in our autoregressive long-term motion generation framework. For Wang et al. [52], two milestone poses are set as the transition and subgoal at the start and end of each episode, with in-between motions generated using their original cVAE model. Methods like Huang et al. [21] and Karunratanakul et al. [23], not initially designed for long-term synthesis, were modified to incorporate $\mathbf{M}_{goal}$ and $\mathbf{M}_{trans}$ in the sampling stage, maintaining their original sampling strategies. In *dynamic* experiments involving dynamic objects, we adapted models like GOAL [47] to encompass both reaching and grasping actions, while preserving their original training pipeline.

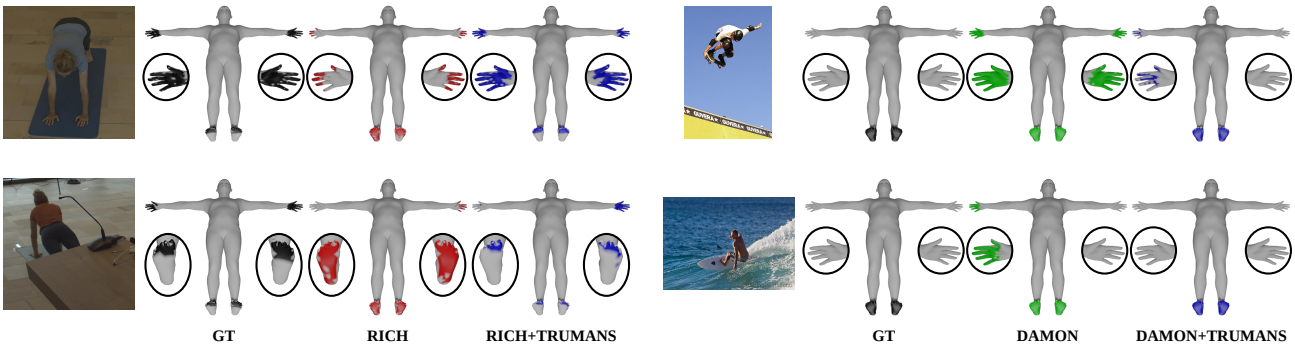
A.4. Human Study

We conducted human studies with 31 participants (17 males and 14 females) in a controlled environment, ensuring no communication among them. For each baseline and ablated version of our method, we generated 4 human motions aligned with the respective test settings (dynamic or non-dynamic). These motions were applied to the SMPL-X body mesh and rendered in the same scene from TRUMANS-test for horizontal comparison. Alongside these synthesized motions, one MoCap motion from TRUMANS-test in the same scene was also included, with careful rendering to minimize visual obstructions.

To qualitatively compare our method with SAMP [17], DIMOS [64], LAMA [25], and Wang et al. [54], we replicated their demonstrations using our model trained on TRUMANS. This involved setting similar subgoals to duplicate the trajectories. Participants were shown side-by-side comparisons of motions synthesized by our method and the baseline methods' demos and then asked to choose the more natural-looking output. The frequency of our method being preferred is reflected in the Success Rate of Discrimination (SucRateDis) reported in Tab. A1.

Table A1. **Human study results of comparisons between our method with recent work.** The Success Rate of Discrimination (SucRateDis), indicating the frequency at which our method is selected as the superior one, is reported.

Method	Success Rate of Discrimination (%)
SAMP [17]	100
Wang et al. [54]	100
LAMA [25]	80.6
DIMOS [64]	64.5



(a) BSTRO [20] trained on RICH [20] combined with TRUMANS. (b) DECO [49] trained on DAMON [49] combined with TRUMANS.

Figure A1. **Additional qualitative results of 3D contact estimation.**

A.5. Image-based Tasks

This section details additional qualitative and quantitative results for image-based tasks using the rendered images and annotations from TRUMANS.

3D Human Mesh Estimation To assess the impact of integrating TRUMANS into training, we use two additional methods, I2L [34] and SGRE [51], on the 3DPW test set. These methods are trained either solely on 3DPW or on a combination of 3DPW and TRUMANS at varying ratios. As indicated in Tabs. A2 and A3, incorporating our synthetic data with the real training set markedly enhances performance.

Table A2. **Performance of I2L [34] in 3D human mesh estimation trained on 3DPW [50] combined with TRUMANS in different ratios.**

Training Data	MPVE↓	MPJPE↓	PA-MPJPE↓
3DPW [50]	186.9	160.4	90.2
3DPW+T (2:1)	133.2	116.5	69.1
3DPW+T (1:1)	**126.1**	**110.2**	**66.2**

Table A3. **Performance of SGRE [51] in 3D human mesh estimation trained on 3DPW [50] combined with TRUMANS in different ratios.**

Training Data	MPVE↓	MPJPE↓	PA-MPJPE↓
3DPW [50]	257.0	223.0	110.6
3DPW+T (2:1)	240.6	207.2	113.5
3DPW+T (1:1)	**138.0**	**117.5**	**80.3**

3D Contact Estimation Qualitative results of baseline methods trained with and without TRUMANS are presented in Fig. A1. These results demonstrate that the incorporation of TRUMANS in training enhances the precision of contact prediction.

B. Additional Details of TRUMANS

B.1. Additional Details of Dataset Comparison

In our dataset comparison presented in Tab. 1, we have categorized similar objects into common types for a more equitable comparison. For SAMP [17], their seven reported objects, including various chairs and a table, are grouped into three types: "sofa," "chair," and "table." BEHAVE [2], with a list of 20 distinct items, is classified into 14 object types, consolidating similar items like chairs and tables. Similarly, iReplica [15]'s report of 9 objects is condensed into 5 classes.

Additionally, for iReplica, we have combined data from their two datasets, EgoHOI and H-contact, for simplicity. EgoHOI contributes 0.25 hours of HSI data with ego-view and multiview RGBD videos, while H-contact adds 0.5 hours of HSI data featuring per-frame hand-object contact.

B.2. Dataset Splits

TRUMANS is divided into training, validation, and test sets, with scenes 1 to 70 for training, 71 to 80 for validation, and 91 to 100 for testing. This distribution creates a split ratio of approximately 7:1:2 across all data frames for the respective sets.

B.3. Object Types

TRUMANS includes 20 types of objects commonly found in indoor scenes, categorized as either [a] (articulated) or [r] (rigid). The list with descriptions is as follows:

- Articulated chair: [a], including gaming and office chairs.
- Rigid chair: [r], encompasses chairs with/without armrests and stools.
- Table: [r], available in round and square shapes.
- Sofa: [r], varieties like single-seaters and couches.
- Bed: [r].
- Book: [a].
- Pen: [r].

- Phone: [r].
- Mouse: [r].
- Keyboard: [r].
- Handbag: [r].
- Vase: [r].
- Cup: [r].
- Bottle: [r].
- Laptop: [a].
- Oven: [a].
- Drawer: [a].
- Cabinet: [a].
- Microwave: [a].
- Door: [a].

B.4. Capture Pipeline

Aligning virtual and real environments The alignment between virtual and real environments is crucial to ensure the plausibility of actions projected into the virtual world. Our process starts with manually selecting scenes and objects from the 3D-FRONT [10] dataset and BlenderKit [6], prioritizing scenes with numerous interactable objects that fit within our motion capture area. Manual alignment is performed to match these virtual scenes with real-world counterparts. For example, a digital sofa may be replicated with a real sofa or chairs arranged to mimic its shape. When digital and physical shapes do not match perfectly, we modify the digital asset, such as scaling objects or editing meshes, or adjust our real-world setups, like placing a mat on a chair to simulate a higher seat.

To align digital characters with our real actors, we start by exporting a human armature matching the actor's bone lengths using the VICON Shogun system [41]. Then, in Character Creator 4 [38], we adjust sliders to create digital humans mirroring the actors' bone lengths. These digital characters are re-imported into Shogun for real-time IK retargeting, ensuring accurate character poses for our digital humans.

Object placeholders Our dataset addresses the limitations of previous datasets related to object visibility and complex HSIs in clustered scenes. We use an optical MoCap system with object placeholders designed for light transmission, enabling accurate capture even in complex HSIs. For example, an actor seated behind a transparent acrylic table allows for precise leg tracking.

Real-time data quality inspection Data recording is monitored in real-time to ensure quality. Inspectors watch the digital world and human avatars on screens during capture, filtering out obvious errors like untracked markers or jittering in IK solving. This real-time inspection aids in maintaining high data quality.

B.5. Motion Augmentation Implementation Details

This section delves into the specifics of our motion augmentation pipeline, illustrating the need for our refinement processes with examples and establishing theoretical bounds for the smoothness of IK target trajectories.

In our augmentation process, we first identify contact points between human joints and object meshes in a given motion sequence. For example, if joint J_1 is in contact with an object mesh at point $\mathbf{v}_m$ at time T_1, and we subsequently alter the object's shape or replace it, the new corresponding point becomes $\mathbf{v}'_m$. To preserve the interaction, the joint's target location $\mathbf{l}'$ must be adjusted accordingly to maintain the contact point:

$$\mathbf{l}'_{T_1} - \mathbf{v}'_m = \mathbf{l}_{T_1} - \mathbf{v}_m, \tag{A1}$$

such that

$$\mathbf{l}'_{T_1} = \mathbf{l}_{T_1} + \mathbf{v}'_m - \mathbf{v}_m. \tag{A2}$$

The offset $\mathbf{v}_1 = \mathbf{v}'_m - \mathbf{v}_m$ represents the change in joint position resulting from the object's shape variance. To address the abrupt trajectory change, we implement a smoothing process for the pose trajectory. Within a defined proximity window W, we apply this offset with a linearly decreasing norm to ensure a smoother transition in the joint's trajectory:

$$\mathbf{v}_{1t} = (1 - \frac{|t - T_1|}{W})\mathbf{v}_1. \tag{A3}$$

This offset application ensures a seamless blend from the original trajectory to the augmented one. To validate the smoothness of this trajectory, we establish a bound on the difference in bone target positions between consecutive frames. The notation $|\cdot|$ is used to denote the absolute value for scalars and the 2-norm for vectors, aiding in quantifying these trajectory modifications.

$$\begin{aligned} |\mathbf{l}'_{t+1} - \mathbf{l}'_t| &= |\mathbf{l}_{t+1} + \mathbf{v}_{t+1} - \mathbf{l}_t - \mathbf{v}_t| \\ &\leqslant |\mathbf{l}_{t+1} - \mathbf{l}_t| + |\mathbf{v}_{t+1} - \mathbf{v}_t| \\ &\leqslant |\mathbf{l}_{t+1} - \mathbf{l}_t| + \frac{|\mathbf{v}|}{W}. \end{aligned} \tag{A4}$$

When the smoothing window length W is set appropriately, such as 30 in our implementation, and the norms of the offset vectors are within practical limits (dictated by reasonable object variation scales and stable CCD IK results), the updated IK target trajectories exhibit sufficient smoothness.

However, this IK solving approach is limited to simpler scenarios involving interaction with a single object. In more complex situations, typical of our dataset, we encounter additional challenges. For instance, consider an offset $\mathbf{v}_1$ introduced at time T_1 for bone J_1 due to object augmentation, with this contact ending at T_2. At T_2, another bone J_2 enters into contact, necessitating a new IK target. Post-IK process, bones without specific targets, including J_1, may shift from

their original positions. We denote the deviation for bone J_1 as $\mathbf{v}_2$. To mitigate this deviation, we employ a similar offsetting technique by blending $\mathbf{v}_1$ and $\mathbf{v}_2$. For each time t within this window of length W, the bone is assigned an offset vector that is a weighted mean of these two offsets, calculated using vector norms as weights:

$$\text{offset} = \frac{(|\mathbf{v}_{1t}|\mathbf{v}_{1t} + |\mathbf{v}_{2t}|\mathbf{v}_{2t})}{|\mathbf{v}_{1t}| + |\mathbf{v}_{2t}|}, \tag{A5}$$

where

$$\begin{aligned} \mathbf{v_{1t}} &= (1 - \frac{|t - T_1|}{W})\mathbf{v}_1, \\ \mathbf{v_{2t}} &= (1 - \frac{|t - T_2|}{W})\mathbf{v}_2. \end{aligned} \tag{A6}$$

By integrating these two stages of linear blending of offset vectors, we achieve smooth trajectories for IK targets. As outlined earlier, akin to the approach in Eq. (A4), we analyze the joint target differences between consecutive frames to further substantiate the smoothness of our approach:

$$\begin{aligned} & |\mathbf{l}'_{t+1} - \mathbf{l}'_t| \\ = \; & \Big| \mathbf{l}_{t+1} - \mathbf{l}_t \\ & + \frac{|\mathbf{v}_{1,t+1}|\mathbf{v}_{1,t+1} + |\mathbf{v}_{2,t+1}|\mathbf{v}_{2,t+1}}{|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|} \\ & - \frac{|\mathbf{v}_{1,t}|\mathbf{v}_{1,t} + |\mathbf{v}_{2,t}|\mathbf{v}_{2,t}}{|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|} \Big| \qquad \text{(A7)} \\ \leqslant \; & |\mathbf{l}_{t+1} - \mathbf{l}_t| \qquad \text{(A8)} \\ & + \left| \frac{|\mathbf{v}_{1,t+1}|\mathbf{v}_{1,t+1}}{|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|} - \frac{|\mathbf{v}_{1,t}|\mathbf{v}_{1,t}}{|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|} \right| \qquad \text{(A9)} \\ & + \left| \frac{|\mathbf{v}_{2,t+1}|\mathbf{v}_{2,t+1}}{|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|} - \frac{|\mathbf{v}_{2,t}|\mathbf{v}_{2,t}}{|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|} \right|, \qquad \text{(A10)} \end{aligned}$$

where

$$|\mathbf{v}_{i,t}| = (1 - |\frac{t - T_i}{W}|)|\mathbf{v}_i|. \tag{A11}$$

Thus we have

$$\begin{aligned} & \left| \frac{|\mathbf{v}_{1,t+1}|\mathbf{v}_{1,t+1}}{|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|} - \frac{|\mathbf{v}_{1,t}|\mathbf{v}_{1,t}}{|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|} \right| \\ = \; & \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{1,t}|(\mathbf{v}_{1,t+1} - \mathbf{v}_{1,t})}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \\ & + \frac{|(|\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t+1} - |\mathbf{v}_{2,t+1}||\mathbf{v}_{1,t}|\mathbf{v}_{1,t})|}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \\ \leqslant \; & \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{1,t}||\mathbf{v}_{1,t+1} - \mathbf{v}_{1,t}|}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \\ & + \left| \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t+1} - |\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t}}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \right| \\ & + \left| \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t} - |\mathbf{v}_{1,t}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t}}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \right| \\ & + \left| \frac{|\mathbf{v}_{1,t}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t} - |\mathbf{v}_{1,t}||\mathbf{v}_{2,t+1}|\mathbf{v}_{1,t}}{(|\mathbf{v}_{1,t+1}| + |\mathbf{v}_{2,t+1}|)(|\mathbf{v}_{1,t}| + |\mathbf{v}_{2,t}|)} \right| \qquad \text{(A12)} \\ < \; & \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{1,t}||\mathbf{v}_{1,t+1} - \mathbf{v}_{1,t}|}{|\mathbf{v}_{1,t+1}||\mathbf{v}_{1,t}|} \\ & + \frac{|\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|\mathbf{v}_{1,t+1} - \mathbf{v}_{1,t}|}{|\mathbf{v}_{1,t+1}||\mathbf{v}_{2,t}|} \\ & + \left| \frac{|\mathbf{v}_{1,t}||\mathbf{v}_{2,t}|(|\mathbf{v}_{1,t+1}| - |\mathbf{v}_{1,t}|)}{|\mathbf{v}_{1,t}||\mathbf{v}_{2,t}|} \right| \\ & + \left| \frac{|\mathbf{v}_{1,t}||\mathbf{v}_{1,t}|(|\mathbf{v}_{2,t+1}| - |\mathbf{v}_{2,t}|)}{|\mathbf{v}_{2,t+1}||\mathbf{v}_{2,t}|} \right| \\ = \; & \frac{3|\mathbf{v}_1| + |\mathbf{v}_2|}{W}. \end{aligned}$$

With the scaling of Eqs. (A9) and (A10) now aligned, we substitute these elements to establish a theoretical bound:

$$\begin{aligned} & |\mathbf{l}'_{t+1} - \mathbf{l}'_t| \\ < \; & |\mathbf{l}_{t+1} - \mathbf{l}_t| + \frac{3|\mathbf{v}_1| + |\mathbf{v}_2|}{W} + \frac{|\mathbf{v}_1| + 3|\mathbf{v}_2|}{W} \qquad \text{(A13)} \\ = \; & |\mathbf{l}_{t+1} - \mathbf{l}_t| + \frac{4}{W}(|\mathbf{v}_1| + |\mathbf{v}_2|), \end{aligned}$$

which ensures that our target trajectories are smooth.

B.6. Annotations

Human Motion The motion data captured from the VICON system, initially in the form of pose sequences of a custom armature, is converted into the SMPL-X format [36] using a vertex-to-vertex optimization method. This ensures accurate and smooth SMPL-X representations; please refer to Fig. A2 for examples. The conversion process involves the following steps:

1. Vertices on the SMPL-X mesh are manually selected and paired with the closest vertices on our custom mesh.
2. A loss function, defined as the Mean Squared Error (MSE) between paired vertex locations, is minimized using the Adam optimizer to refine SMPL-X parameters until convergence.

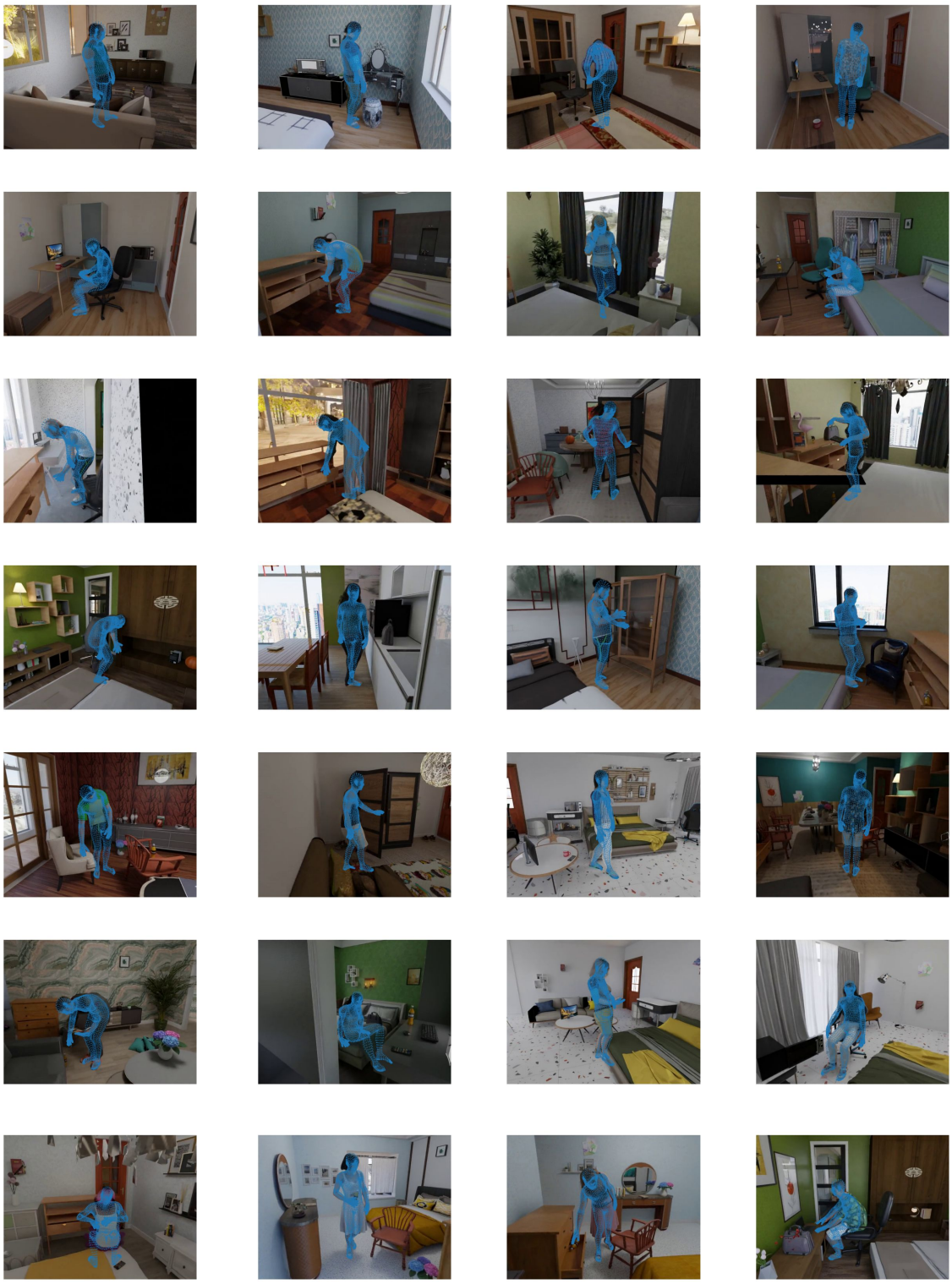


Figure A2. **Examples of SMPL-X annotations in TRUMANS.**

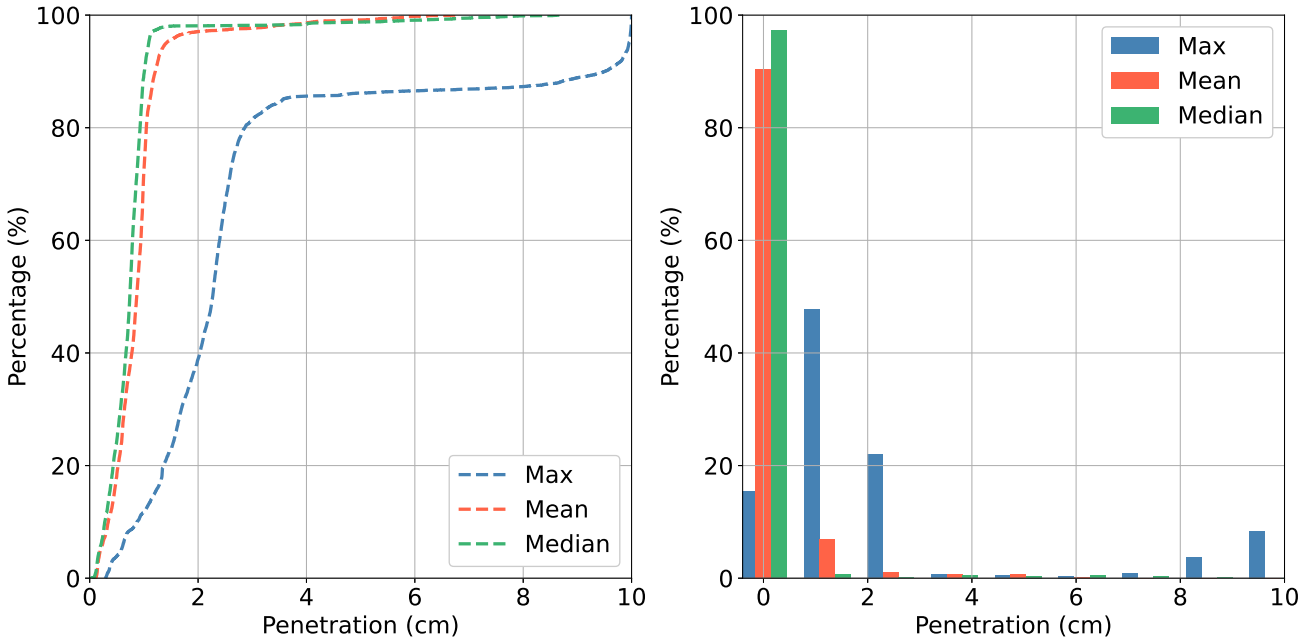


Figure A3. **Penetration statistics in TRUMANS.** This analysis covers maximum, mean, and median penetration distances per vertex per frame. The left graph displays the proportion of frames with penetration below various thresholds (X-axis), and the right bar plot categorizes frames by specific penetration distance ranges (X-axis). Notably, in more than 95% of frames, both the mean and median penetration distances stay below 2cm.

3. Inaccuracies in the SMPL-X mesh are manually corrected by adjusting bone rotations.

After aligning the SMPL-X mesh with our custom mesh, we record the mapping for future use.

In the second phase, we enhance the custom motion data by adding interpolated frames from the T-pose to the start pose. This ensures a smooth transition for each bone in the sequence.

Finally, we optimize the SMPL-X parameters frame by frame, starting from the pose established in the first phase. We first refine the body shape parameters and then adjust the pose parameters, including global translation. The optimization of each frame starts from the pose of the previous frame and continues until convergence. This method relies on minimal mesh changes between frames, supported by our high-quality motion data. A typical MSE value at convergence ranges between 5e-5 and 1e-4, indicating an average point distance of less than 1cm.

Contact Following the fitting of SMPL-X meshes, we compute per-vertex contact annotations. The contact for each human mesh vertex is determined based on its proximity and orientation relative to scene or object meshes. A vertex is deemed in contact if it fulfills either of the following conditions: (i) it resides inside an object's mesh, or (ii) while outside an object's mesh, it is within a specified threshold distance and the angle between its normal and the vector pointing towards the object is under 60 degrees. The latter criterion is particularly vital for accurate contact annotation, as it prevents misidentification in scenarios like a hand holding a bottle. Penetration statistics, as detailed in Fig. A3, reveal that in over 95% of the frames, both the mean and median penetration distances remain below 2cm. For examples of contact annotation, please refer to Fig. A4.

Objects In TRUMANS, we include the watertight mesh for all objects and their 6D poses for each frame. For articulated objects, part-level annotations are provided along with the URDF (Unified Robot Description Format) files to represent their kinematic structure.

Actions For every sequence within TRUMANS, multi-hot action labels are assigned on a frame-by-frame basis. This approach allows for the representation of multiple concurrent actions in a single frame, with each action identified by a distinct label.

B.7. Video Rendering

To enhance video diversity and accurately capture HSI details, we developed an adaptive camera tracking algorithm that maintains footage smoothness and consistency. The camera, set at a constant height of 1.4 meters, moves within a 2-meter radius around the human, horizontally oriented towards the body. The camera's pose is exclusively determined by its rotation around the z-axis in the human coordinate system.

We set keyframes at intervals of 30 frames. For each keyframe, 20 camera proposals adhering to the constraints are pre-defined and evenly distributed around the ring. To identify active hand interactions, we calculate the minimum distance between hand joints and dynamic objects. If this distance exceeds 20 centimeters, we default to tracking the right hand. For the identified interacting hand, rays emitted from each camera proposal towards the hand joints help measure visibility. A joint is considered visible to a camera if the intersection point with the scene's mesh can be projected onto the camera's imaging plane, and the distance to the joint is less than 10 centimeters. The number of visible joints determines whether a camera effectively captures the interaction. The visibility threshold for different keyframes is dynamically adjusted to ensure at least one camera captures the interaction, except when all joints are invisible.

After assessing the interaction capture capability of the 20 cameras at each keyframe, dynamic programming is used to select the optimal keyframe camera pose sequence that minimizes total rotation and maximizes interaction coverage. "Camera pose" here specifically refers to rotation about the z-axis. Camera poses for frames between keyframes are interpolated using cubic spline interpolation of the rotation angles.

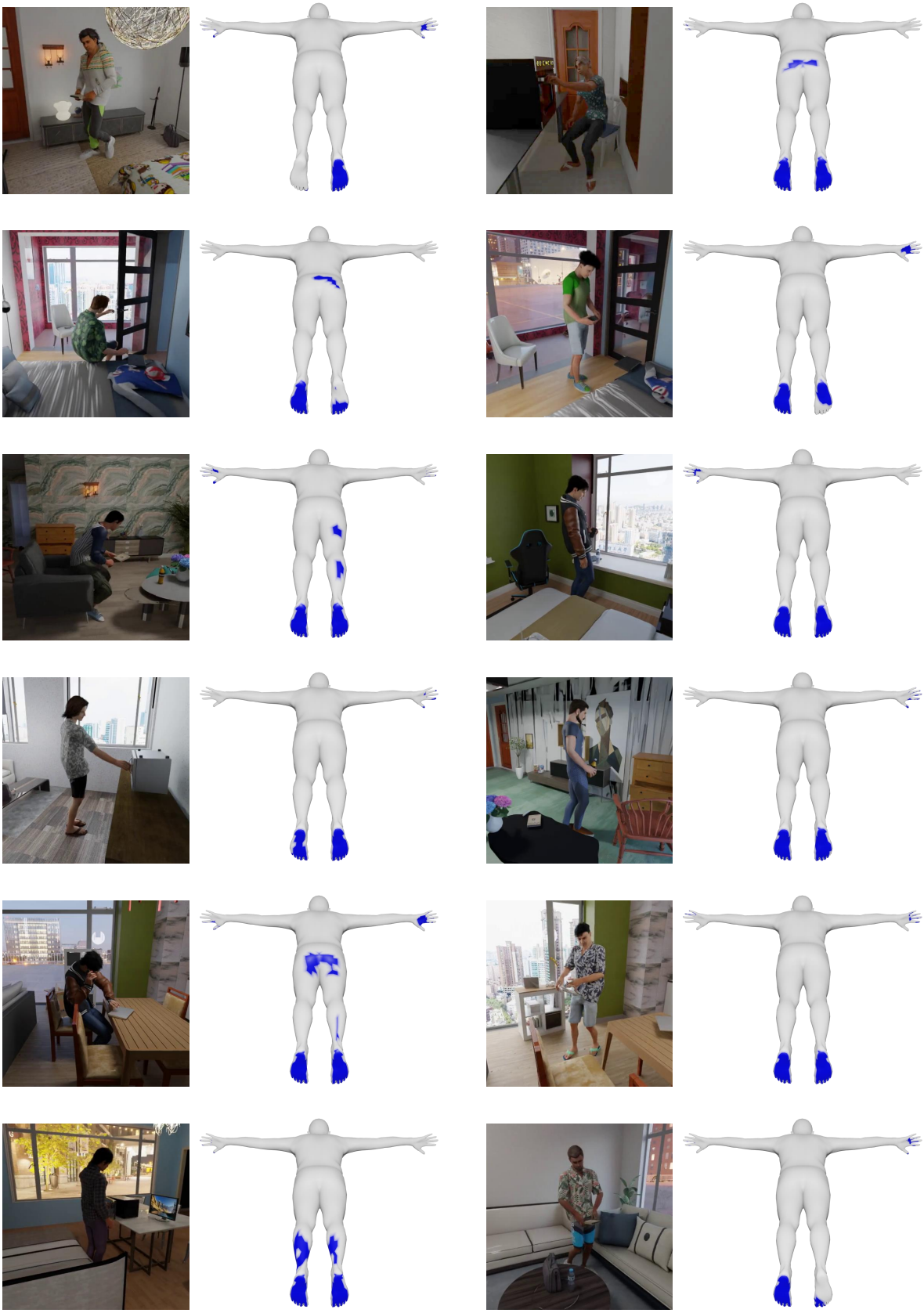


Figure A4. **Examples of contact annotations in TRUMANS.**