CIRCLE: Capture In Rich Contextual Environments

João Pedro Araújo[1], Jiaman Li[1], Karthik Vetrivel[1], Rishi Agarwal[1], Jiajun Wu[1],
Deepak Gopinath[2], Alexander Clegg[2], C. Karen Liu[1]
[1]Stanford University, [2]Meta AI

Figure 1. Example poses from CIRCLE captured from real human motion in a virtual environment.

Abstract

Synthesizing 3D human motion in a contextual, ecological environment is important for simulating realistic activities people perform in the real world. However, conventional optics-based motion capture systems are not suited for simultaneously capturing human movements and complex scenes. The lack of rich contextual 3D human motion datasets presents a roadblock to creating high-quality generative human motion models. We propose a novel motion acquisition system in which the actor perceives and operates in a highly contextual virtual world while being motion captured in the real world. Our system enables rapid collection of high-quality human motion in highly diverse scenes, without the concern of occlusion or the need for physical scene construction in the real world. We present CIRCLE, a dataset containing 10 *hours of full-body reaching motion from* 5 *subjects across nine scenes, paired with ego-centric information of the environment represented in various forms, such as RGBD videos. We use this dataset to train a model that generates human motion conditioned on scene information. Leveraging our dataset, the model learns to use ego-centric scene information to achieve non-trivial reaching tasks in the context of complex 3D scenes.*

To download the data please visit our website.

1. Introduction

Humans excel at interacting with complex environments, effortlessly engaging in everyday tasks such as getting out of a car while carrying a backpack or plugging a power cord into an outlet behind a cabinet. The remarkably flexible and compliant human body enables access to narrow or cluttered spaces where clear paths are not available. Synthesizing 3D human motion that reflects this ability to navigate in highly contextual, ecological environments, such as our homes, grocery stores, or hospital operating rooms, will significantly impact applications in Embodied AI, Computer Animation, Robotics, and AR/VR.

Machine learning models have significantly advanced the creation of 3D human motion and behaviors in recent years. However, the success of the ML-approach hinges on one condition—the human motion data for training models must be of high quality, volume, and diversity. Traditional motion capture (mocap) techniques focus on the "human movement" itself, rather than the state of the environment in which the motion takes place. While mocap can faithfully record human kinematics, capturing humans in a contextual

scene requires physical construction of a production set and specific props in the capture studio. This steep requirement limits the capability of today's mocap technologies to holistically capture realistic human activities in the real world.

We propose to eliminate the costly requirement of physical staging by capturing human motion during interactions with a *virtual reality simulation*. This allows us to capture motion like the ones shown in Figure 1, where a person reaches into cluttered spaces in a furnished apartment. Additionally, we are able to simultaneously record paired first-person perspectives of the virtual environment through VR, as illustrated in Figure 3. With paired ego-centric observation of the world, we can now train motion models to not only comprehend the *how* of certain tasks, but also the *why* behind an individual's movements.

By creating the complex scene in the virtual world and keeping the capture space in the real world empty, our method provides four crucial advantages over state-of-the-art solutions. First, creating a highly contextual environment in VR is much simpler and less costly than in actual reality. Second, capturing the state of the real world requires complex sensor instrumentation, while the state of the virtual world is readily available from the simulator. Third, because the capture space in reality is always empty, our system is not subject to occlusions that degrade the motion quality, regardless of any clutter in the perceived environment. Fourth, the data acquired by such a system provide 3D human motions and corresponding videos of the environment rendered in any camera view of choice, such as the egocentric view.

We use a Meta Quest 2 headset and the AI Habitat simulator in our experiments. However, our system is agnostic to the choice of hardware, simulator, and virtual environment. To illustrate the possibilities enabled by the availability of contextual motion capture data, we collect a dataset, CIRCLE, containing ten hours of full-body reaching motion within nine indoor household scenes. CIRCLE contains challenging reaching scenarios, including reaching for an object behind the toilet, between tightly placed furniture, and underneath the table. Finally, we use CIRCLE to train a model that generates reaching motions conditioned on scene information. Our model takes as input the starting pose of the person in the scene as well as the target location of the right hand, and automatically generates a scene-aware sequence of human motion that reaches the target location. We propose two different methods to encode the scene information and compare them against baselines.

In summary, the contributions of this work include:

- A novel motion acquisition system to collect 3D human motion with synchronized scene information,
- A novel dataset, CIRCLE, with 10 hours of human motion data from 5 subjects in 9 realistic apartment scenes,
- A data-driven model, trained on CIRCLE, for generating full-body reaching motion within an environment.

2. Related Work

Human Motion Datasets. The desire to thoroughly model human motion has contributed to many high-quality datasets. High-resolution optical motion capture datasets [7, 18, 20, 35, 42] range from the smaller CMU Motion Capture Database [7] to AMASS [20], a rich human motion collection that unifies several mocap datasets and contains over 40 hours of motion data. Other works have modeled a vast diversity of motions, including whole-body reaching, object manipulation [39], human-scene contact [12, 13], human-chair interaction [50]. While these datasets provide an excellent baseline for analyzing human motion and interaction with specific objects, they do not consider the constraints of the 3D environments humans naturally move through, a key contribution of our approach.

Early work from Hasler *et al*. [10] recovers joint configurations with scene constraints using multiple unsynchronized moving cameras. More recently, GPA [46] and SAMP [11] capture scene-conditioned human motion by placing a limited set of physical objects in the mocap area. Our work obviates the need for physical construction of environments by placing the actor in a virtual world, enabling capture in diverse and cluttered scenes. BEHAVE [4] provides multi-view RGBD videos with 3D pose and contact information to enable tracking of human-object interactions. GIMO and EgoBody [49, 51] propose motion datasets with IMU-based and egocentric image capture respectively. Moreover, both datasets offer 3D reconstructions of the scenes. However, IMU-based motions can result in drifting [14] and LiDAR and camera-based 3D reconstructions are inherently noisier than the virtual ground truth.

3D Human Motion Synthesis. In recent years, significant progress has been made in synthesizing kinematic human motion. Early progress saw the introduction of periodic motion embeddings [23, 34, 43] and non-linear autoregressive architectures for pose tracking and modeling [2, 25, 40, 45]. Recurrent neural networks [8, 21, 32, 33, 52] and similar autoregressive models have seen success on smaller sets of motion. Conditional variational autoencoders (cVAE) [5, 17, 30] are another popular architecture for generating plausible motion sequences. Lately, Transformers [1, 15, 26, 36] and diffusion models [41, 48] have seen the most success at generating unseen motions. Inspired by recent progress, we train a Transformer-based model architecture on our dataset to generate scene-conditioned motion.

Constrained Pose Generation. Our task of synthesizing 3D human motion in a contextual, ecological environment is inherently a constrained full-body pose generation chal-

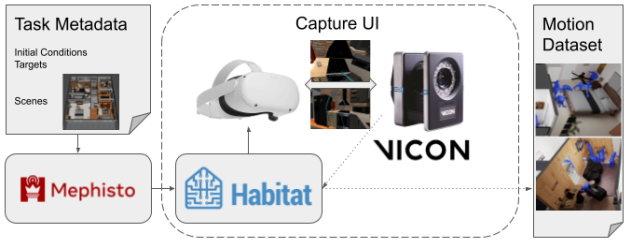


Figure 2. An overview of the data collection system, including the Mephisto experimentation framework, a VR headset (Meta Quest 2 in our case), AI Habitat simulator, and Vicon motion capture.

Figure 3. The motion capture process both in the real world (left) and the Habitat VR app (middle). Right: the corresponding SMPL-X mesh generated by MoSh and rendered in Blender.

lenge. Much of the work in constrained pose generation involves 3D hand-object manipulation [16, 47]. Although not as rigorously explored, constrained full-body pose generation is not novel. Gupta *et al*. [9] propose an observation-driven Gaussian process latent variable model for human pose estimation using information from scenes and actions. Cao *et al*. [6] tackle full-body scene-aware motion synthesis by addressing the global trajectory and local pose prediction. The authors of the previously mentioned SAMP dataset [11] also propose and train an algorithm to generate interactions with chairs, sofas, and tables. Another paper, COUCH [50], builds on the SAMP method and provides users with fine-grained control over the generation of human-chair interactions. Recent work includes using Transformer architectures for gaze-directed and context-aware human motion [51] and estimating 3D body-scene contact from a single image [13]. Another recent related work is GOAL [38], a data-driven procedure to generate the full-body grasping motion conditioned on an object's geometry and location relative to the body.

3. Data Collection

Our motion acquisition system is designed to enable collection of a contextual motion capture dataset with high volume, high diversity, and high quality.

High Volume. Using a VR headset to collect a large-scale human dataset poses a unique challenge due to the concerns of user discomfort and costly context switching. The VR app must run as fast as possible to minimize motion sickness and ensure a smooth user experience. In addition, having a highly streamlined pipeline to capture motion in large batches is crucial for minimizing overheads and scaling up the dataset. Thus, an ideal system should maximize the efficiency of data collection during the active VR time.

Figure 4. Examples of diverse poses found in CIRCLE.

Figure 5. Left: Layout of the initial state (green) and goal (purple) sampling regions within the scenes. Right: Representative images of the following scenes (left to right, top to bottom): bathroom, media room, laundry, bedroom, living room, dining room, kitchen, and closet.

High Quality. Although the data collected using optics-based motion capture are often of high quality, our system depends on the virtual world being immersive and realistic such that it does not influence the natural behaviors of the actor. Being untethered and hands-free is absolutely necessary. In addition, we need to give the actor visual feedback, such as the sight of their virtual body, whenever possible to ensure the naturalness of their performance.

High Diversity. Our system should support various subjects, scenes, and tasks. We need tools that assist the planning and preparation process for capturing diverse motion sequences. We also need a streamlined pipeline to work with a variety of virtual worlds and assets.

3.1. System Overview

With the previous design goals in mind, we built a system to collect contextual motion capture data (Fig. 2). Our system has two main components: the motion capture system and the VR app which provides the actor with the illusion of being within a virtual scene. We use AI Habitat, an open-source embodied AI research platform [31, 37], as our virtual world simulator. Habitat provides easy access to a wide variety of interaction-ready virtual environments and assets, as well as an API tailored to Embodied AI experi-

mentation. We build a WebXR app on top of Habitat's web build [29], capable of running at real-time rates on a VR headset with a modern web browser installed. This VR app streamlines all data collection logic, allowing the user to focus on quickly executing capture tasks. Data recorded by the app (headset trajectories and simulation state) are submitted to Mephisto [22] for storage.

For capturing human motion, we use a Vicon system with 12 cameras controlled with Vicon Shogun. The cameras record at 120 FPS. We connect Shogun and the VR app through a webserver running locally on the machine capture machine. The VR headset interfaces with this webserver through the local WiFi network. This allows the app to directly control the beginning and ending of individual mocap recordings. Conversely, we use Vicon's DataStream SDK to stream the reconstructed skeleton with very low latency to the VR app, allowing users of the system to see their skeleton in the virtual world. This provides visual feedback for events such as collisions, and enables the user to interact more immersively with the virtual world.

Figure 3 shows an actor using our system to collect a sequence. The room has no obstacles, so all the actor's movements (left image) are in response to what is being shown in the headset (middle image). The image on the right shows what the same sequence looks like after post-processing. We describe in detail each component of our system in the following sections.

3.2. Preparation

We define *sequences* as individual *clips* of captured data (the atomic unit of CIRCLE). Sequence specifications are generated from pairs of manually annotated start and goal regions (see Fig. 5 for an example), from which we sample concrete instances of start and goal position pairs. We refer to each start/goal region pair as a *task*. Specifically, any two sequences with start and goal positions sampled from the same pair belong to the same task.

We frontload all task generation steps to the preparation phase at the start of each collection session. The goal of this phase is to prepare a list of sequences (individual clips) to be collected. To streamline the specification of these sequences, we develop a tool that allows users to load scenes into Blender and annotate both start and goal regions. We then sample pairs of points from these regions to generate sequence specifications. Sampled goal positions are rejected if they are unreachable (either too far from navigable surfaces or inside scene or object geometry). Finally, we sort the generated sequences to minimize the transition overhead of resetting the scene and importing assets.

3.3. Collection

Given a list of sequences, we simultaneously start the VR app and a Mephisto process. When entering VR for the first time each day, the user must calibrate the headset in order to align it to the Vicon skeleton. Since we do not know the true offset between the headset frame of reference and the captured head bone, we use an alignment heuristic. For details, please see the Supplementary Material.

After calibration, the user can start recording. Clips containing the sequences listed for collection are labeled as reaching clips. The transition between sequences can also contain valid contextual mocap data. Therefore, instead of discarding them, the system records and labels them separately. As the user completes sequences, the data are logged to the Mephisto server. When the user has finished all pregenerated sequences, the VR app exits and Mephisto closes, caching all sequences resulting from the session.

3.4. Post-processing

After collection, CIRCLE data are passed through a variety of post-processing steps.

Mocap data processing. We use Shogun Post to process and export the captured clips to BVH and C3D formats.

Offline synchronization Due to latency between the headset and the webserver communicating with the Vicon machine, the start times of the headset and mocap data are misaligned and must be synchronized. By assuming that the offset between the head bone and the headset remains constant during each sequence, this can be accomplished by solving for the time offset which maximizes the convolution between the velocity profiles of the head bone and headset. With start times aligned, we then trim the sequences to the same length and linearly interpolate the headset poses such that every mocap frame has a corresponding headset pose.

Human mesh fitting. We run MoSh++ [20] (henceforth referred to as MoSh) on the C3D files to acquire the SMPL-X parameters corresponding to each frame in the sequence (Fig. 3, right).

Synthetic sensor information. After synchronization, we load both the mocap data in BVH format and the VR trajectories in Habitat and extract synthetic sensor information such as ego-centric RGB-D videos (Fig. 3, middle). Additionally, we can use Blender to render first-person RGB-D videos with the SMPL-X meshes calculated by MoSh.

Quality assurance. Identifying and fixing sequences with artifacts is a demanding manual process. We find that our pipeline has a very high yield of data that does not need to be fixed, so our focus is on identifying sequences with problems so that they can be collected again. To help prioritize, we develop a suite of tools that automatically check for common problems, such as:

- **Task completion.** The task in the VR app is considered complete if the wrist of the live-streamed skeleton is within 1 cm of the goal location. We use this to validate the accuracy of our pipeline. All collected clips

Scene	Frames			Minutes		
	T	R	Total	T	R	Total
Bathroom	137k	188k	325k	19	26	45
Bedroom	203k	309k	512k	28	43	71
Clothes closet	137k	210k	347k	19	29	48
Dining room	213k	309k	522k	30	43	72
Kitchen	261k	309k	570k	36	43	79
Laundry room	137k	201k	338k	19	28	47
Laundry closet	128k	216k	344k	18	30	48
Living room	165k	279k	444k	23	39	62
Media room	376k	529k	904k	52	73	126
Total			4306k			598

Table 1. Breakdown of our dataset by scene and type of clip (**T**ransition and **R**eaching). We collect over 4 million frames, and 10 hours of data.

pass this test.

- **Marker swaps**. For a given marker, given its position at frame t, we find the nearest marker at frame $t+1$. If the labels of those two markers differ, we consider that those two markers are swapped. Clips that are flagged by this procedure are automatically discarded.
- **Jumps in joint values.** To help spot sequences with jumps in joint values, for each sequence we record the maximum joint linear acceleration in global space and the maximum angular velocity in local space. Sequences that score high values on either of these metrics are then manually inspected.

3.5. Dataset

We use our data acquisition system to produce CIRCLE, a dataset of whole body reaching motion within a fully furnished virtual home. Each distinct room in the apartment is considered a separate scene. We manually annotate the start and goal regions for each scene (128 tasks in total; see Fig. 5) using the tool described in Sec. 3.2 and follow the procedure outlined in Sec. 3.3 for data collection. Physics simulation was disabled during the collection of CIRCLE, which means that the environment is static in all sequences.

Dataset contents. CIRCLE contains 10 hours (more than 7000 sequences) of both right and left hand reaching data for five subjects across 9 scenes (Fig. 5, right). The breakdown of data per scene is listed in Table 1. The diversity of scenes induces a wide range of motions (Fig. 4), including reaching high places, bending, crawling, crouching, kneeling, and lying down.

After post-processing, each sequence in CIRCLE includes:

- The SMPL-X parameters of a body model fit to the mocap data using MoSh,
- The VR headset trajectory synchronized to the mocap skeleton,
- Egocentric RGB-D video (rendered with both Habitat and Blender),
- Task specific data, such as initial and goal conditions, and the scene where the data are collected.

4. Motion Generation

Our goal is to generate a sequence of scene-aware human poses $\boldsymbol{X} \in \mathbb{R}^{T\times D}$ given a start pose $\boldsymbol{X}_0$, target wrist joint position $\boldsymbol{g} \in \mathbb{R}^3$ and a scene point cloud $\boldsymbol{S} \in \mathbb{R}^{N\times 3}$, where T represents the sequence length, D represents the dimension of the pose state and N represents the number of scene points. The pose state at time step t, $\boldsymbol{X}_t$, consists of root translation $\boldsymbol{p}_t \in \mathbb{R}^3$, global joint positions $\boldsymbol{J}_t \in \mathbb{R}^{22\times 3}$, and local joint rotation $\boldsymbol{R}_t \in \mathbb{R}^{22\times 6}$ represented using continuous 6D vectors [53]. We use the SMPL-X model [24] with 22 joints. Our approach is to learn a motion refinement model that takes as input a rough initial motion sequence and scene features, and outputs a scene-aware and higher-quality motion sequence (Figure 6).

Motion Initialization. Given the full body pose at the first time step, $\boldsymbol{X}_0$, we generate a naive initial motion sequence $\boldsymbol{X}_0, \hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, ..., \hat{\boldsymbol{X}}_T$, using a simple procedure. We define a constant pose $\boldsymbol{X}_c$ shared by the whole dataset. We first translate $\boldsymbol{X}_c$ such that the human's wrist reaches $\boldsymbol{g}$ and make this translated pose $\hat{\boldsymbol{X}}_T$.We then linearly interpolate the root translation from $\boldsymbol{X}_0$ to $\hat{\boldsymbol{X}}_T$ to create the in-between frames, $\hat{\boldsymbol{X}}_t$, where $1 \leq t \leq T-1$.

Scene Encoding. One crucial design decision for training our model is the scene representation. The high-quality privileged 3D information provided by CIRCLE makes many choices of scene encoding possible. We explore two types of scene encoding: Basis Point Set (BPS) [27] representation and features extracted by PointNet++ [28]. Both types of scene encoding consist of human-scene interaction features and geometry features. Together, they form a scene feature $\boldsymbol{F}_t \in \mathbb{R}^{256}$ at each time step t.

We first detail our BPS representation. Given an initialized motion sequence $\boldsymbol{X}_0, \hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, ..., \hat{\boldsymbol{X}}_T$, we use SMPL-X model [24] to generate the corresponding human meshes $\boldsymbol{M}_0, \boldsymbol{M}_1, ..., \boldsymbol{M}_T$ where each $\boldsymbol{M}_t \in \mathbb{R}^{K\times 3}$ has K downsampled vertices (we use $K = 699$). To encode the human-scene interaction features, we compute $d(\boldsymbol{M}_t, \boldsymbol{S}) \in \mathbb{R}^{K\times 3}$, the difference between human vertices and their nearest neighbor points of the scene. As for the geometry scene features, we first define a cylinder with a radius $r = 0.6$ m and a height $h = 2$ m. The cylinder is centered at $(p_t^x, p_t^y, \frac{h}{2})$, where p_t^x, p_t^y denote the horizontal position of the root at time step t. We sample 1024 points from the volume of the cylinder at time t: $\boldsymbol{B}_t = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_{1024}]^T$. We compute the geometry features of BPS representation

as $d(\boldsymbol{B}_t, \boldsymbol{S}) \in \mathbb{R}^{1024\times 3}$. We concatenate $d(\boldsymbol{M}_t, \boldsymbol{S})$ and $d(\boldsymbol{B}_t, \boldsymbol{S})$, and feed it to an MLP to get a lower dimensional vector as our final scene encoding $\boldsymbol{F}_t \in \mathbb{R}^{256}$ at time step t.

To extract the scene features using PointNet++, we first pre-train a PointNet++ model for the semantic segmentation task on the S3DIS dataset [3]. We input the scene point cloud $\boldsymbol{S}$ to the pre-trained model to get features for each point $\boldsymbol{F}^s \in \mathbb{R}^{N\times 128}$. For an arbitrary point $\boldsymbol{e}$ in 3D space, the feature $f(\boldsymbol{e}, \boldsymbol{F}^s) \in \mathbb{R}^{128}$ of $\boldsymbol{e}$ is computed by taking the inverse distance weighted interpolation, $f(\boldsymbol{e}, \boldsymbol{F}^s) = \sum_{i=1}^{n_e} w_i \boldsymbol{F}^s(\boldsymbol{p}_i) / \sum_{i=1}^{n_e} w_i$, where $w_i = 1/||\boldsymbol{p}_i - \boldsymbol{e}||_2$, n_e represents the number of nearest neighbors for $\boldsymbol{e}$ (we use three), and $\boldsymbol{p}_i$ represents the ith nearest neighbor point of $\boldsymbol{e}$. For the interaction features, we compute the feature at time step t as $\boldsymbol{F}_t^h = [f(\boldsymbol{M}_t^0, \boldsymbol{F}^s), f(\boldsymbol{M}_t^1, \boldsymbol{F}^s), ..., f(\boldsymbol{M}_t^K, \boldsymbol{F}^s)]^T$ for each of the vertices $\boldsymbol{M}_t^k$ on the human mesh to obtain $\boldsymbol{F}_t^h \in \mathbb{R}^{K\times 128}$. For the geometry features, similarly, we use the nearest neighbor points of $\boldsymbol{B}_t$ denoted as $\boldsymbol{N}_t$ to compute the features as $\boldsymbol{F}_t^c = [f(\boldsymbol{N}_t^0, \boldsymbol{F}^s), f(\boldsymbol{N}_t^1, \boldsymbol{F}^s), ..., f(\boldsymbol{N}_t^{1024}, \boldsymbol{F}^s)]^T$ and obtain $\boldsymbol{F}_t^c \in \mathbb{R}^{1024\times 128}$. Finally, we compute the mean of $\boldsymbol{F}_t^h$ and $\boldsymbol{F}_t^c$, and concatenate them to get a 256-dimensional vector as our final scene encoding at time step t, $\boldsymbol{F}_t \in \mathbb{R}^{256}$.

Model Architecture. We adopt a Transformer-based model architecture [44] (Figure 6) to estimate human motions from the initialized motion $\boldsymbol{X}_0, \hat{\boldsymbol{X}}_1, ..., \hat{\boldsymbol{X}}_T$ and scene features $\boldsymbol{F}_0, \boldsymbol{F}_1, ..., \boldsymbol{F}_T$. Our model consists of four self-attention blocks, each containing a multi-headed attention layer followed by a position-wise feed-forward layer.

Training Loss. Our training loss consists of four terms, including L_1 loss for root translation, global joint positions, local joint rotation, and joint positions computed using forward kinematics. The loss for each time step is defined as

$$L = w_{\text{trans}}\|\boldsymbol{p}_t - \hat{\boldsymbol{p}}_t\|_1 + w_{\text{joint}}\|\boldsymbol{J}_t - \hat{\boldsymbol{J}}_t\|_1 + w_{\text{rot}}\|\boldsymbol{R}_t - \hat{\boldsymbol{R}}_t\|_1 + w_{\text{FK}}\|\boldsymbol{J}_t - \text{FK}(\hat{R}_t, \hat{\boldsymbol{p}}_t)\|_1,$$

where w_{trans}, w_{joint}, w_{rot}, and w_{FK} represent the loss weights for each term. Note that our loss function does not leverage the scene information for supervision in order to have a fair comparison with our baselines. However, in practice these supervisions can and should be leveraged.

5. Evaluation

We evaluate our model by measuring the generated motions using a set of standard metrics, by comparing against two baselines, and by inspecting the results visually. Please also view the supplemental video for more qualitative evaluation.

5.1. Setup

We train our model on CIRCLE, use two different criteria to split the data, and compare against two baselines. To keep all sequences under 240 frames (the size of our model input), we downsample every sequence to 20 FPS. We train our model for 1800 epochs using AdamW [19] with weight decay 0.01 and an initial learning rate of 0.0001 that is multiplied by 0.3 every 1000 epochs.

Dataset Split. We evaluate on two separate splits:

- **Random**: We randomly split the reaching sequences into 2565 for training and 453 for testing.
- **Task**: We hold out specific tasks (start/goal region pairs) for testing, resulting in 2578 sequences for training and 440 for testing (109 and 19 tasks, respectively).

Baselines. We compare our approach with two baselines:

- **GOAL:** The MNet described in GOAL [38], an MLP architecture that, given a start and a goal pose[1], autoregressively predicts the next pose in the sequence.
- **NO-SCENE:** Our architecture without the scene encoding $\boldsymbol{F}_0, \cdots \boldsymbol{F}_T$. Since our loss function does not depend on the scene information, our model does not have any privileged supervision over this baseline.

All baseline methods are trained on the same data.

5.2. Evaluation Metrics

We use the following metrics to evaluate our method.

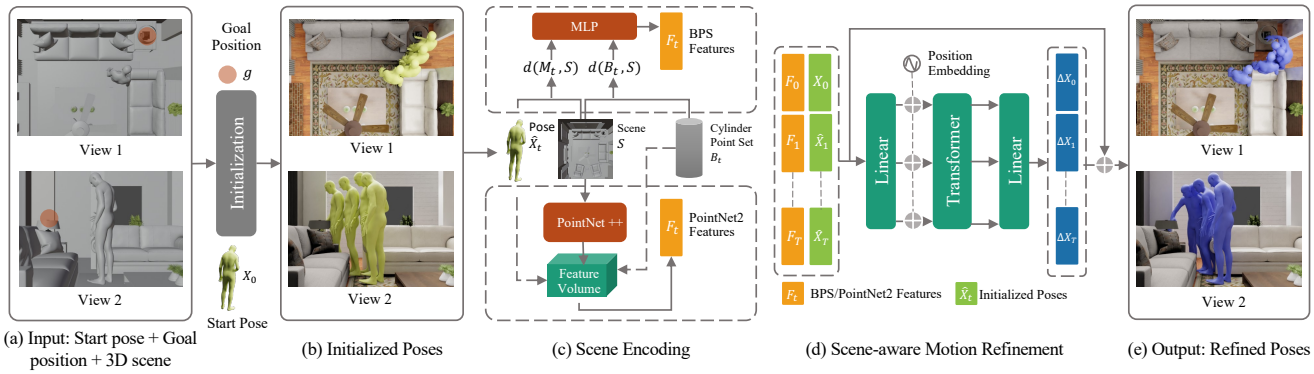
Task Completion. Task success is determined by a threshold on euclidean distance between the generated and ground truth position of the right-hand wrist at the last frame in the sequence. An instance of a task is considered successful if this distance is lower than 10 cm.

Collision Avoidance. We check each frame for collisions between the human mesh and the scene and compute the sum of interpenetration depths as our collision metric. To empirically correct for MoSh fitting inaccuracies, we filter out collisions with a depth lower than 2 cm.

Foot Sliding. Our chosen metric is percentage of frames in a sequence with sliding. Heuristically, a frame is considered to have sliding if the velocity of the vertex with the lowest height is greater than 1 cm per frame. Critically, this heuristic applies to sequences where the human is kneeling or lying down rather than standing.

Similarity to Dataset. To measure the similarity to the ground truth motion, we calculate the mean vertex error, mean joint position error, mean root translation error, and mean joint orientation error as the Frobenius norm of the rotation matrix representation of the root's orientation.

[1]We provide the ground truth goal pose instead of training GOAL's GNet module to make the baseline stronger. The original architecture conditions the output on the BPS representation of an object that is to be grasped. We modify it to condition on the BPS representation of the



(a) Input: Start pose + Goal position + 3D scene (b) Initialized Poses (c) Scene Encoding (d) Scene-aware Motion Refinement (e) Output: Refined Poses

Figure 6. Method overview. Given a start pose, goal position, and 3D scene (a), we first initialize the input poses using constant local joint rotation and linearly interpolated root translation (b). We extract scene features for each time step using BPS or PointNet++ from the initialized poses, a fixed point set sampled from a cylinder, and scene point clouds (c). We then concatenate scene features and initialized poses and feed them to a transformer-based model (d) to generate final poses (e).

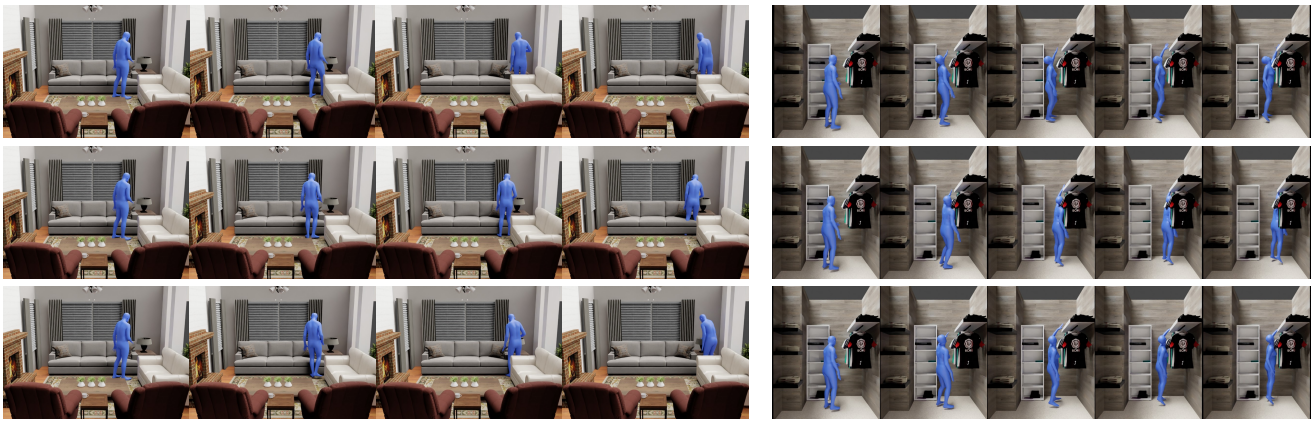


Figure 7. Comparison of the ground truth motion (top row) and the output generated by our model both with (bottom row) and without (middle row) scene conditioning for two different scenes (we show BPS scene condition only). We see that introducing scene information reduces collisions. For the full-resolution image and comparison with GOAL and PointNet scene representation please see the Supplementary Material.

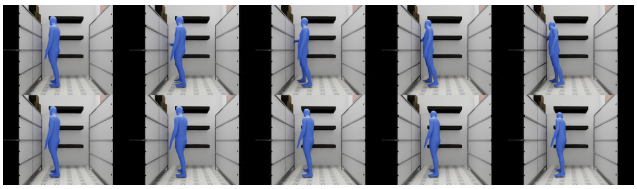


Figure 8. GOAL baseline test sequence (bottom) and corresponding ground truth motion (top). The model moves the character very little (especially the feet), resulting in poor task and similarity metrics, despite low prevalence of foot sliding artifacts.

5.3. Results

We calculate each metric from 5.2 for every sequence and report its average value over all sequences in Tab. 2 and Tab. 3.

environment.

Task Completion. Our model clearly outperforms the GOAL baseline in the task completion metric (Tab. 2). We attribute this to the fact that GOAL has to predict the sequence autoregressively, and so all the steps need to contribute towards reaching the goal position. In contrast, our method can, by design, accomplish long-term, scene conditioned planning via full sequence prediction. Note that, despite the absence of a goal loss term in our training procedure, our models are still able to score high in the task metrics with joint position losses. Our model also moderately outperforms the NO-SCENE baseline.

Collision Avoidance. Our model clearly outperforms NO-SCENE in collision avoidance (Tab. 2). Figure 7 demonstrates the visual comparison of the differences. Since our model is not supervised by any measure of collision avoidance in the loss function, this gain can be attributed to scene state observations. Despite having access to the same scene information, GOAL sequences have

Test split	Model	Task		Motion quality		Similarity to dataset			
		Success rate (%)	Dist. to goal (cm)	Cumulative collision depth (cm)	Foot sliding (%)	MVPE (cm)	MJPE (cm)	Root trans. error (cm)	Joint orn. error
Random	Ground truth	100.00	0.00	11.02	2.14	0.00	0.00	0.00	0.00
	NO-SCENE	74.56	**7.94**	15.87	18.89	11.36	8.18	13.45	0.22
	GOAL	12.14	44.48	20.89	**12.78**	17.78	21.48	19.99	1.97
	Ours (BPS)	74.12	8.23	12.09	17.76	**10.45**	**7.62**	**11.96**	0.22
	Ours (PN2)	**79.20**	8.22	**11.63**	18.83	11.09	8.05	12.60	0.22

Table 2. Evaluation metrics on the random split ("PN2" refers to PointNet++ scene encoding).

Test split	Model	Task		Motion quality		Similarity to dataset			
		Success rate (%)	Dist. to goal (cm)	Cumulative collision depth (cm)	Foot sliding (%)	MVPE (cm)	MJPE (cm)	Root trans. error (cm)	Joint orn. error
Task	Ground truth	100.00	0.00	8.17	2.11	0.00	0.00	0.00	0.00
	NO-SCENE	64.92	9.08	10.18	18.95	11.61	8.38	14.11	0.23
	GOAL	8.43	47.60	18.05	**11.24**	17.27	20.94	20.40	1.95
	Ours (BPS)	**76.08**	**8.30**	**7.04**	20.66	11.47	8.34	**13.13**	0.23
	Ours (PN2)	70.84	9.19	9.22	18.97	**11.32**	**8.14**	13.47	0.23

Table 3. Evaluation metrics on the task split ("PN2" refers to PointNet++ scene encoding).

deeper collisions.

Foot Sliding. GOAL produces significantly fewer foot sliding artifacts than our approach. However, visualizing the GOAL predicted sequences (Fig. 8) reveals that the model often outputs a nearly static pose for all frames, explaining this deceptive contrast. Our approach with scene conditioning generally produces more foot sliding artifacts than NO-SCENE. This is possibly attributable to the greater overall adaptation of source data to fit scene constraints.

Similarity to Dataset. Although similarity to ground truth is not the primary goal, it is a loose metric that indicates the overall quality of the human motion. Our model and NO-SCENE perform similarly, while GOAL is trailing behind. We encourage readers to view the supplemental videos for additional evaluation of the motion quality.

Unseen Tasks Table 3 illustrates generalization to unseen tasks. As expected, overall performance for all models have degraded on this split. However, we continue to observe our approach notably outperforming GOAL in task completion. Additionally, our approach with BPS continues to outperform NO-SCENE on collision depth.

6. Conclusion

We introduce an efficient new method of capturing contextual motion data within a realistic cluttered environment in virtual reality, leverage this system to generate a novel contextual motion dataset, CIRCLE, and use this dataset to train models that generate scene-aware human motion.

While our first application of CIRCLE demonstrates its potential, generalized contextual motion generation remains a challenging problem. Our evaluation indicates that more work will be necessary from the research community to produce better adapted motion with fewer artifacts.

In addition to improving on our results, we would like to understand how the data collected with our system compares with human motion in real physical environments. We are also interested in studying how generalization improves with more data. Fortunately, our streamlined capture pipeline will enable us to continue expanding upon the initial scale and diversity of CIRCLE.

Looking to the future, we are interested in exploring more interactive contextual tasks. For example, expanding our system to capture physical and virtual object trajectories in addition to human motion. This next step toward interactive data collection has the potential to fill a critical need in Embodied AI, fueling efforts to understand how humans interact with objects in their everyday lives and pass those skills to artificial, embodied agents.

Acknowledgments. This work is in part supported by Meta AI, NSF CCRI #2120095, ONR MURI N00014-22-1-2740, and the Stanford Institute for Human-Centered AI.

References

[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021. 2

[2] Okan Arikan, David A Forsyth, and James F O'Brien. Motion synthesis from annotations. In *ACM SIGGRAPH 2003 Papers*, pages 402–408, 2003. 2

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 6

[4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15935–15946, 2022. 2

[5] Yujun Cai, Yiwei Wang, Yiheng Zhu, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Chuanxia Zheng, Sijie Yan, Henghui Ding, et al. A unified 3d human motion synthesis model via conditional variational auto-encoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11645–11655, 2021. 2

[6] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*, pages 387–404. Springer, 2020. 3

[7] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database, 2009. 2

[8] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015. 2

[9] Abhinav Gupta, Trista Chen, Francine Chen, Don Kimber, and Larry S Davis. Context and observation driven latent variable model for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[10] Nils Hasler, Bodo Rosenhahn, Thorsten Thormahlen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 224–231. IEEE, 2009. 2

[11] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021. 2, 3

[12] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2282–2292, 2019. 2

[13] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and inferring dense full-body human-scene contact. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13274–13285, June 2022. 2, 3

[14] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W Winkler, and C Karen Liu. Transformer inertial poser: Attention-based real-time human motion reconstruction from sparse imus. *arXiv preprint arXiv:2203.15720*, 2022. 2

[15] Seong Uk Kim, Hanyoung Jang, Hyeonseung Im, and Jongmin Kim. Human motion reconstruction using deep transformer networks. *Pattern Recognition Letters*, 150:162–169, 2021. 2

[16] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 3

[17] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Trans. Graph.*, 39(4), 2020. 2

[18] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (ToG)*, 33(6):1–13, 2014. 2

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[20] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 2, 4

[21] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017. 2

[22] The Mephisto contributors. GitHub - facebookresearch/Mephisto: A suite of tools for managing crowdsourcing tasks from the inception through to data packaging for research use. https://github.com/facebookresearch/Mephisto. 4

[23] Dirk Ormoneit, Hedvig Sidenbladh, Michael Black, and Trevor Hastie. Learning and tracking cyclic human motion. *Advances in Neural Information Processing Systems*, 13, 2000. 2

[24] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 5

[25] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. *Advances in neural information processing systems*, 13, 2000. 2

[26] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 2

[27] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4332–4341, 2019. 5

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5

[29] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022. 4

[30] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[31] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[32] Jongmin Kim Seong Uk Kim, Hanyoung Jang. Human motion denoising using attention-based bidirectional recurrent neural network. In *SIGGRAPH Asia Posters*, 2019. 2

[33] Xiangbo Shu, Liyan Zhang, Guo-Jun Qi, Wei Liu, and Jinhui Tang. Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3300–3315, 2021. 2

[34] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European conference on computer vision*, pages 702–718. Springer, 2000. 2

[35] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1):4–27, 2010. 2

[36] Ziyang Song, Dongliang Wang, Nan Jiang, Zhicheng Fang, Chenjing Ding, Weihao Gan, and Wei Wu. Actformer: A gan transformer framework towards general action-conditioned 3d human motion generation. *arXiv preprint arXiv:2203.07706*, 2022. 2

[37] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[38] Omid Taheri, Vasileios Choutas, Michael J. Black, and Dimitrios Tzionas. GOAL: Generating 4D whole-body motion for hand-object grasping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 6

[39] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[40] Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. Modeling human motion using binary latent variables. *Advances in neural information processing systems*, 19, 2007. 2

[41] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 2

[42] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John P. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*, 2017. 2

[43] Raquel Urtasun, David J Fleet, and Pascal Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer vision and image understanding*, 104(2-3):157–177, 2006. 2

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 6

[45] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. 2

[46] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. *arXiv preprint arXiv:1905.07718*, 2019. 2

[47] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. Manipnet: Neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (ToG)*, 40(4):1–14, 2021. 3

[48] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2

[49] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape, motion and social interactions from head-mounted devices. *arXiv preprint arXiv:2112.07642*, 2021. 2

[50] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*. Springer, October 2022. 2, 3

[51] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, Karen Liu, and Leonidas Guibas. Gimo: Gaze-informed human motion prediction in context. In *ECCV*, 2022. 2, 3

[52] Shi-hong Xia Zhiyong Wang, Jinxiang Chai. Combining recurrent neural networks and adversarial training for human motion synthesis and control. In *IEEE Transactions on Visualization and Computer Graphics*, 2018. 2

[53] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 5

A. Supplemental

A.1. Skeleton/headset alignment

We use the headset's built-in calibration tool to align its forward direction with the forward direction of the motion capture volume. We also calibrate the mocap system and the headset such that they measure the same floor height. This reduces the alignment problem to finding a planar offset $\vec{o}$ (we do not modify the skeleton's height) that is used to align the livestreamed skeleton to the actor in VR. To achieve this, we assume that the midpoint of the eyes (measured by the headset),

$$\vec{r}_e = \frac{\vec{r}_{\text{left eye}} + \vec{r}_{\text{right eye}}}{2},$$

lies along a line that has the direction of the head bone's local forward vector and contains the midpoint of the head bone's top face ($\vec{f}_h$ and $\vec{r}_{ht}$, both respectively measured by the mocap system). We can write this as a linear system with 3 constraints and 3 unknowns,

$$\vec{r}_e = \vec{r}_{ht} + \lambda \vec{f}_h + \vec{o},$$

where

$$\vec{o} = \begin{bmatrix} o_x \\ o_y \\ 0 \end{bmatrix},$$

that we solve during calibration.

B. High resolution result images

For the complete versions of the images in Fig. 7, see Fig. 9 and Fig. 10, respectively.

Figure 9. Comparison of the i) ground truth motion (top row), and outputs generated by ii) GOAL (second row), iii) our model without scene information (third row), iv) scene information using pointnet (fourth row), v) scene information using BPS (fifth row). We see that the sequence generated by GOAL fails to achieve the objective, and that introducing scene information reduces collisions.

Figure 10. Comparison of the i) ground truth motion (top row), and outputs generated by ii) GOAL (second row), iii) our model without scene information (third row), iv) scene information using pointnet (fourth row), v) scene information using BPS (fifth row). We see that the sequence generated by GOAL fails to achieve the objective, and that introducing scene information reduces collisions.