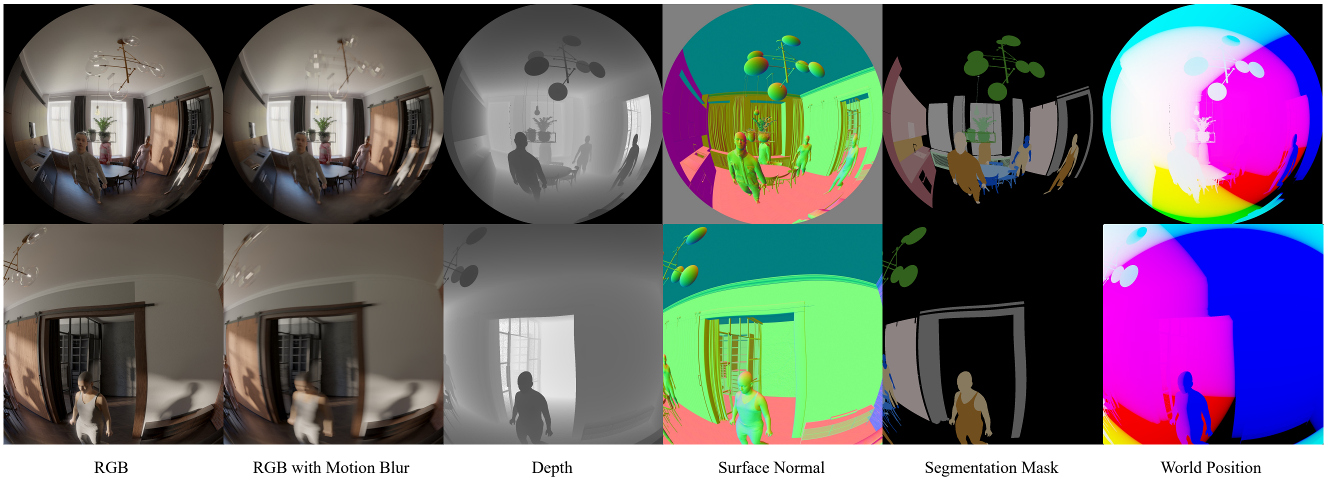
EgoGen: An Egocentric Synthetic Data Generator

Gen Li[1] Kaifeng Zhao[1] Siwei Zhang[1] Xiaozhong Lyu[1]
Mihai Dusmanu[2] Yan Zhang[1] Marc Pollefeys[1,2] Siyu Tang[1]
[1]ETH Zürich [2]Microsoft


RGB

RGB with Motion Blur

Depth

Surface Normal

Segmentation Mask

World Position

Figure 1. EgoGen: a scalable synthetic data generation system for egocentric perception tasks, with rich multi-modal data and accurate annotations. We simulate camera rigs for head-mounted devices (HMDs) and render from the perspective of the camera wearer with various sensors. Top to bottom: middle and right camera sensors in the rig. Left to right: photo-realistic RGB image, RGB with simulated motion blur, depth map, surface normal, segmentation mask, and world position for fisheye cameras widely used in HMDs.

Abstract

Understanding the world in first-person view is fundamental in Augmented Reality (AR). This immersive perspective brings dramatic visual changes and unique challenges compared to third-person views. Synthetic data has empowered third-person-view vision models, but its application to embodied egocentric perception tasks remains largely unexplored. A critical challenge lies in simulating natural human movements and behaviors that effectively steer the embodied cameras to capture a faithful egocentric representation of the 3D world. To address this challenge, we introduce EgoGen, a new synthetic data generator that can produce accurate and rich ground-truth training data for egocentric perception tasks. At the heart of EgoGen is a novel human motion synthesis model that directly leverages egocentric visual inputs of a virtual human to sense the 3D environment. Combined with collision-avoiding motion primitives and a two-stage reinforcement learning approach, our motion synthesis model offers a closed-loop solution where the embodied perception and movement of the virtual human are seamlessly coupled. Compared to previous works, our model eliminates the need for a pre-defined global path, and is directly applicable to dynamic environments. Combined with our easy-to-use and scalable data generation pipeline, we demonstrate EgoGen's efficacy in three tasks: mapping and localization for head-mounted cameras, egocentric camera tracking, and human mesh recovery from egocentric views. EgoGen will be fully open-sourced, offering a practical solution for creating realistic egocentric training data and aiming to serve as a useful tool for egocentric computer vision research. Refer to our project page.

1. Introduction

The analysis of visual input from front-facing egocentric cameras is crucial for applications that benefit from a first-person perspective, mirroring the natural human experience [26, 47, 125]. AR devices, for instance, can utilize this viewpoint to amplify user immersion. Such cameras can also cater to individual preferences, providing custom visual assistance for those with impaired vision [21, 130].

Despite its potential, egocentric perception faces challenges, primarily due to the scarcity of labeled data. Although datasets like Ego4D [26], ADT [65], Epic-Kitchen [15] and HoloAssist [110] exist, creating such datasets with rich and accurate annotations is costly and raises privacy concerns [125]. Alternatively, using graphics techniques to render synthetic multi-modal visual data has proven to be cost-effective and successful in training deep learning models, such as 3D human body estimation [9] and facial landmark detection [116].

Creating egocentric synthetic data is challenging because egocentric cameras capture the complex interplay of body movements and the environment from the camera wearer's viewpoint. Modeling the intricate details and variations in human behavior presents a significant challenge.

To tackle this problem, we introduce *EgoGen*, an egocentric synthetic data generation approach that simulates data from embodied sensors, i.e., front-facing cameras in head-mounted devices (HMD). While the ultimate goal is to simulate human behaviors that are indistinguishable from reality, in this work, we focus on creating virtual humans (i.e., camera wearers) that can explore and avoid obstacles in the 3D world that is not only complex and dynamic but could potentially include other *moving* virtual humans.

Specifically, we propose a novel generative human motion model. Our key insight is that body movement and embodied perception should be seamlessly coupled. As William Gibson aptly stated, *"We see in order to move; we move in order to see."*, our egocentric perception is crucial for identifying obstacles, navigating in an environment, and planning actions. Our body movements are not solely a response to visual stimuli; they also change our egocentric perception. Therefore, the key idea of our motion model is to enable virtual humans to *see* their environment with *egocentric* visual inputs and respond accordingly by learning a policy to control a set of collision-avoiding motion primitives (CAMPs) that are composable for synthesizing long-term, diverse human motions. Due to the unbounded and high-dimensional latent action space of our generative motion primitive model, direct policy training through rendered egocentric images is often unstable [133]. Therefore, we propose a two-stage reinforcement learning scheme using an efficient *egocentric* visual proxy to couple egocentric visual cues and body movements seamlessly. In addition, we use an "attention" reward to incentivize egocentric perception behaviors, i.e., looking in the desired direction.

Empirical results showcase the benefits of our egocentric perception-driven motion framework, which does not require a pre-calculated walking path in 3D scenes as in [33, 57, 129]. Instead, it empowers virtual humans to perceive the environment from their own viewpoint, enabling them to navigate, circumvent obstacles, and plan movements to reach the destination. Moreover, our model generalizes well to dynamic environments, even with training limited to static settings. By training virtual humans independently using CAMPs, our method synthesizes emergent multi-human behaviors without relying on multi-agent reinforcement learning algorithms. Egocentric visual cues are essential to build exploratory and generalizable motion models that unify navigational planning and movement control in complex and dynamic environments.

Building upon CAMPs, we further create a scalable data generation pipeline for *EgoGen* that outfits virtual humans with clothing, automates cloth animation, and integrates 3D assets from various sources. We validate *EgoGen*'s efficacy across three egocentric perception tasks. The high-quality synthetic data with precise ground truth annotations consistently improve the performance of state-of-the-art methods. In summary, the contributions of this work are:

1. We introduce *EgoGen*, a generative and scalable synthetic data generation approach specifically tailored for egocentric perception tasks.
2. We introduce novel motion primitives based on egocentric visual cues, enabling diverse and realistic human motion synthesis in 3D scenes. These primitives empower virtual humans to handle complex scenarios, such as dynamic environments and crowd motion without relying on multi-agent reinforcement learning.
3. *EgoGen* enables us to augment existing real-world egocentric datasets with synthetic images. Quantitative results demonstrate enhanced performance in state-of-the-art algorithms on mapping and localization for head-mounted cameras, egocentric camera tracking, and human mesh recovery from egocentric views.

2. Related Work

Human-Related Simulators and Synthetic Data. Previous works primarily focus on simulating robots [58, 75, 86, 100, 103] and autonomous cars [8, 18, 23, 81]. While some incorporated animated digital humans, like in [23] and [73], these efforts relied on pre-recorded motion sequences. Rendering images of people to train perception models has been widely studied such as [19, 20, 47, 74, 90, 94]. In particular, [105] offers large-scale synthetic data for egocentric camera wearer pose estimation but relies on mocap data, lacking realistic and spontaneous interactions with the digital world. In contrast, *EgoGen* closes the gap in egocentric synthetic data generation for head-mounted devices. Please refer to Sec. S1 for detailed comparisons.

Human Motion Synthesis. Generating high-fidelity human motions and interactions with 3D scenes is widely studied in graphics [12, 35, 36, 46, 97, 98]. While they can generate high-quality motion, it's usually deterministic. Synthesizing physically plausible human motions has been extensively studied [14, 34, 68, 80, 104, 115]. However, they struggle with generalization to different body shapes. For example, [80] explicitly created 2048 humanoids to improve body shape generalizability. Time series models [69, 102, 124] synthesize the stochastic motions of diverse people well. However, in [102, 124], their generated motion sequences have limited lengths and human-scene interactions are not explicitly considered. Autoregressive methods [49, 79, 127, 129] can produce perpetual motions. In particular, [127] can generalize to diverse body shapes and synthesize long-term human motions.

Our egocentric perception-driven motion synthesis model is closely related to [127, 129], but distinguishes itself w.r.t.: (1) Enabling virtual humans to explore using egocentric visual cues, without predefined paths. (2) Synthesizing egocentric perception behaviors beyond locomotion, e.g., looking in certain directions. (3) Handling dynamic environments and multi-agent behavior without re-training.

Mapping and Localization for AR. Localization and mapping from images is a long-standing problem known as: Photogrammetry [2, 30]; Structure-from-Motion (SfM) [25, 70, 87, 95, 114]; Simultaneous Localization and Mapping (SLAM) [16, 41, 60, 64]. Researchers have worked to make SLAM amenable for edge hand-held or head-mounted devices [7, 24, 41]. Cloud-based services like Google's Visual Positioning System [78], Niantic's Lightship [63], and Microsoft's Azure Spatial Anchors [38] have made visual localization and mapping more accessible. Benchmarking efforts have arisen for small-scale AR scenarios [40, 93], touristic landmarks [37, 88], and large-scale AR-device based localization [84, 85] to evaluate these systems.

Egocentric Human Pose Estimation. Estimating 3D bodies from RGB images is widely studied from third-person views [10, 39, 42–45, 48, 67], and egocentric views [29, 50, 51, 61, 91, 105, 106, 122, 123, 126], mostly requiring expensive real-world data paired with ground truth annotations. Besides RGB images, depth images offer explicit 3D information, mitigating scale and shape ambiguity, with the potential to enable broader AR/VR applications. However, depth-based methods, especially for the egocentric view, are underexplored. Most existing works [32, 53, 59, 76, 92, 109, 117, 121] predict 3D body skeletons without expressive body meshes, struggling with challenges like severe body truncations and scene occlusions typical in egocentric views. Such limited attention mainly stems from the scarcity of data, as obtaining high-quality human mesh annotations for real-world depth images is labor-intensive.

3. Ego-Sensing Driven Motion Synthesis

To close the loop for the interdependence between egocentric synthetic image data and human motion synthesis, we use deep reinforcement learning (RL), integrating egocentric vision cues to synthesize human motions as described in Sec. 3.1 and 3.2. Subsequently, we extend learned policies to generate emergent multi-agent behaviors, as in Sec. 3.3.

3.1. Ego-Sensing Driven Motion Primitives

Generating realistic egocentric data requires diverse and lifelike human motion synthesis. In this work, we consider arguably the most common everyday behaviors: navigating towards goals with egocentric perception while avoiding collisions with obstacles and people in dynamic 3D scenes.

Overview. Following recent literature [49, 120, 127, 129], we employ deep RL to train control policies on learned latent spaces that characterize natural human motions. However, unlike these previous works that only consider simple static scenes, we leverage egocentric perception and propose collision-avoiding motion primitives (CAMPs) to enable virtual humans to self-explore and navigate in a dynamic environment. Specifically, CAMPs are trained jointly to produce collision-free motion sequences. At each timestep t, the agent observes the state $\mathbf{s}_t$, performs an action $\mathbf{a}_t$, and receives a reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$, where $\mathbf{s}_{t+1}$ represents the next state of the environment due to $\mathbf{a}_t$.

Egocentric Sensing As Depth Proxy. We aim to sample actions given by a policy to synthesize realistic human motions. Egocentric perception-driven motion synthesis should arguably use egocentric vision as input. However, depth rendering is costly and RL requires billions of samples to converge [3, 113]. Besides, directly training RL with visual data can be unstable [133]. We thereby use a cheap-to-compute *egocentric sensing* $\mathcal{E}_t$ as a proxy for depth images as illustrated in Fig. 2. N rays are cast evenly from the midpoint of two eyeballs, i.e., the location of the egocentric camera. The field of view $[\theta_{min}, \theta_{max}]$, centered on the 2D projection of the viewing direction $\vec{\mathbf{v}}$, limits the agent's perception to the front area. Rays stop at collisions, with collision detection in 2D. See more details in Sec. S2.1.

Agent Representation. The agent is a virtual human represented by an SMPL-X mesh [67]. We further compact the body representation by selecting $M = 67$ body surface markers $\mathbf{x} \in \mathbb{R}^{M\times 3}$ on the mesh following [128].

Motion Primitive Environment. We implement a *finite-horizon* environment based on the generative motion primitive model from [127]. Specifically, a motion primitive is defined as a 0.5-second motion clip containing $T = 20$ frames in the canonical coordinate, and each frame contains a single agent representation. The primitive model $\mathcal{P}$ is based on the C-VAE framework [96], which takes the first $T_s = 2$ frames as the condition, and models a conditional probability of the next $T - T_s$ frames. Compared to [127] trained on the AMASS dataset [52] with many sport motion sequences, we train $\mathcal{P}$ using the SAMP dataset [33], which focuses on daily activities, better suited for HMD use cases. Our *action space* $\mathcal{A}$ is the pretrained 128D latent space of $\mathcal{P}$, and the **action** $\mathbf{a}_t$ can be randomly sampled from $\mathcal{A}$.

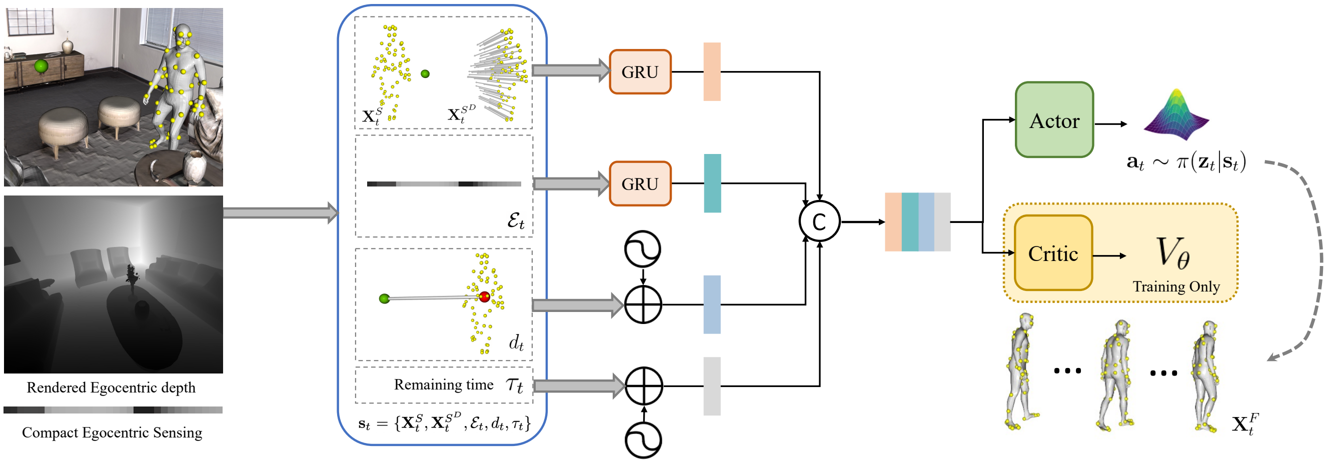

Rendered Egocentric depth
Compact Egocentric Sensing
$\mathbf{X}_t^S$
$\mathbf{X}_t^{S^D}$
$\mathcal{E}_t$
d_t
Remaining time τ_t
$\mathbf{s}_t = \{\mathbf{X}_t^S, \mathbf{X}_t^{S^D}, \mathcal{E}_t, d_t, \tau_t\}$
GRU
GRU
C
Actor
$\mathbf{a}_t \sim \pi(\mathbf{z}_t|\mathbf{s}_t)$
Critic
V_θ
Training Only
...
...
$\mathbf{X}_t^F$

Figure 2. Policy network architecture. We learn a generalizable mapping from motion seed body markers $\mathbf{X}_t^S$, marker directions $\mathbf{X}_t^{S^D}$, egocentric sensing $\mathcal{E}_t$, and distance to the target d_t to CAMPs. The policy learns a stochastic collision avoiding action space to predict future body markers $\mathbf{X}_t^F$. For illustration purposes, we visualize only one frame of $\mathbf{X}_t^S$ and $\mathcal{E}_t$. See Sec. 3.1 and 3.2 for details.

With the input of a random action $\mathbf{a}_t \in \mathbb{R}^{128}$ and a motion seed $\mathbf{X}_t^S = [\mathbf{x}_t^0, \mathbf{x}_t^{T_s-1}]$ (history frames), $\mathcal{P}$ predicts future frames $\mathbf{X}_t^F = [\mathbf{x}_t^{T_s}, ..., \mathbf{x}_t^{T-1}]$ of the current motion primitive $\mathbf{X}_t = [\mathbf{X}_t^S, \mathbf{X}_t^F] \in \mathbb{R}^{T \times M \times 3}$, which represents a short sequence of human motion spanning 0.5 s:

$$\mathbf{X}_t^F = \mathcal{P}(\mathbf{X}_t^S, \mathbf{a}_t)$$

State. To preserve Markov property [66], the state is defined as $\mathbf{s}_t = \{\mathbf{X}_t^S, \mathbf{X}_t^{S^D}, \mathcal{E}_t, d_t, \tau_t\}$, in which $\mathbf{X}_t^{S^D} \in \mathbb{R}^{T_s \times M \times 3}$ denotes the normalized direction of each marker seed to the target, $\mathcal{E}_t \in \mathbb{R}^{T_s \times N}$ denotes the egocentric sensing depth proxy, d_t denotes the distance from the pelvis to the target, and τ_t denotes the remaining time. See Fig. 2.

Reward. To synthesize egocentric perception behaviors, we use an "attention reward" to incentivize the virtual human to look in specific directions: $r_{attention} = \cos\langle\vec{\mathbf{v}}, \vec{\mathbf{a}}\rangle$, where $\vec{\mathbf{a}}$ is the attention direction from the head joint to the viewing target. The reward function is defined as:

$$r_t = r_{cont.} + r_{dist} + r_{ori} + r_{attention} + r_{pene} + r_{pose} + r_{succ},$$

where $r_{cont.}$ enforces valid foot contact and minimizes foot skating; r_{dist} encourages reaching the target; r_{ori} aligns the body forward direction with the target; r_{pene} guides collision avoidance; r_{pose} reduces unrealistic human poses; and r_{succ} is a sparse reward when reaching the target.

Episode Termination. To handle collisions beyond [129], we employ multiple *termination* signals to conclude an episode if the generated motion primitive $\mathbf{X}_t$ satisfies any of the following conditions:

- Success: The virtual human reached the target.
- Penetration: The virtual human collides with the obstacle.
- Timeout: The virtual human did not reach the target within the maximum timesteps.

3.2. Training Collision-Avoiding Stochastic Policies

Algorithm. We use Proximal Policy Optimization [89] (PPO) to learn a generalizable mapping from various egocentric sensing and body configurations to CAMPs. Instead of extensive manual data collection for all possible input combinations, we leverage the stochastic nature of the PPO policy. Through exploration and sampling actions, the agent traverses the scene and generates varied egocentric sensing and body configurations, diversifying the training data.

Instead of training each CAMP independently for every single step, we use PPO to train a sequence of CAMPs jointly in multi-step collision avoidance tasks. This approach can benefit choosing a more favorable CAMP which makes the subsequent action easier.

Network. The network architecture is shown in Fig 2. The actor and critic network share a feature extraction trunk to encode the state $\mathbf{s}_t$: the motion seed ($\mathbf{X}_t^S$ and $\mathbf{X}_t^{S^D}$) and the egocentric sensing $\mathcal{E}_t$ are encoded using RNNs; the rest of scalar states are encoded using positional encoding [108]. The actor predicts a stochastic policy $\mathbf{a}_t \sim \pi(\mathbf{z}_t|\mathbf{s}_t)$ conditioned on the current state $\mathbf{s}_t$, where $\mathbf{z}_t \sim \mathcal{N}(\mu, \Sigma)$. μ and Σ are the mean and variance of the learned action space.

Objective Function. The objective function includes the policy surrogate L^{CLIP}, the value function error term L^{VF} to evaluate the value prediction V_θ, and an entropy bonus L^S to encourage exploration:

$$L = L^{CLIP} + c_1 L^{VF} + c_2 L^S$$

where c_1, c_2 are coefficients. See more details in Sec. S2.3.

RL Pre-training and Finetuning. Training in crowded scenes, e.g. Replica [99], requires additional steps. Because the action space $\mathcal{A}$ is an unbounded Gaussian, RL exploration while predicting reasonable human poses can

Algorithm 1 Crowd motion synthesis with learned CAMPs

```
Result: Multi-human locomotion w/ collision avoidance;
Init: crowd size $C$, marker seed for each human $\mathbf{X}_c^S$;
for $step \leftarrow 1$ to $max\_steps$ do            ▷ env. finite horizon
    for $c \leftarrow 1$ to $C$ do                    ▷ for each human
        update all locations with $\{bbox(\mathbf{X}_c^S)\}_{c=1:C}$
        compute egocentric sensing $\mathcal{E}_c$;
        execute one action, produce one CAMP;
    end for
end for
```

be challenging. We first pretrain the policy with a higher r_{pene} weight without *penetration termination*. After convergence, we finetune it with strict termination constraints using a signed distance field (SDF) for penetration detection. Please refer to Sec. S2.2 for the formulation and weighting of each reward and training detail.

3.3. Compositing Learned Motion Primitives

Although CAMPs are trained solely with static scenes, their direct application to dynamic settings is achieved by decomposing jointly trained CAMPs into individual motion primitives and re-compositing them. Our model demonstrates effective generalization, provided that the egocentric sensing is updated with the most recent obstacle location at each timestep. Furthermore, our model is directly applicable to tasks involving complex interactions with other virtual humans. To synthesize crowd motion (Alg. 1), each virtual human employs the same policy to navigate and avoid others. To a specific virtual human, others are seen as dynamic obstacles, represented by body bounding boxes for avoidance. Acknowledging the inherent delay in human reactions when avoiding dynamic obstacles [4], agents take a single CAMP sequentially, instead of in parallel, i.e. the first agent generates its first CAMP and waits for others to complete their first CAMP before all agents move on to prepare their second CAMP. To ensure successful collision avoidance, the agent's egocentric sensing is updated before taking a new action. This composition of CAMPs synthesizes emergent multi-human behaviors *without* multi-agent RL algorithms (see Sec. 5.1), enhancing the generalization and scalability.

4. Egocentric Synthetic Data Generation

Synthesizing realistic egocentric perception-driven human motions (as detailed in Sec. 3) forms the foundation of simulating egocentric synthetic data. An overview of our egocentric data generation pipeline *EgoGen*, is shown in Fig. 3.

4.1. Embodied Camera Placement

Similar to existing AR devices, we use the head pose to define the egocentric viewing direction $\vec{v}$. Our development is based on Blender [13]. We use the SMPL-X [67] mesh to position the egocentric camera between the two eyeballs. The camera's viewing direction ($\vec{v}$) is perpendicular to the plane determined by the two eye bones in the armature. We also support simulating multi-camera rigs as shown in Fig. 1. When the body moves (Sec. 3.3), we can synthesize egocentric videos with continuously updated camera poses.

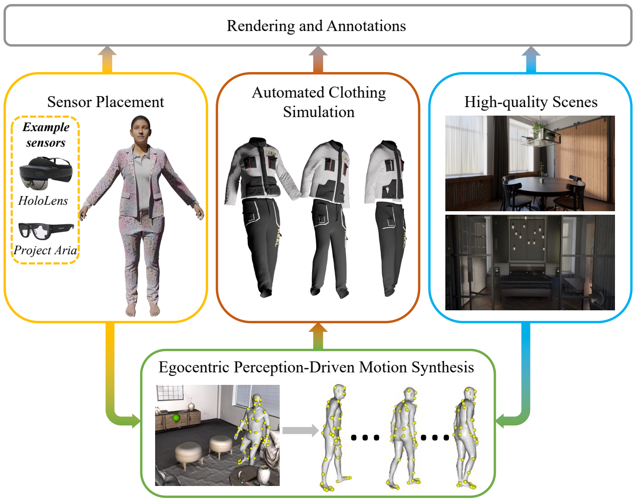

Rendering and Annotations
Sensor Placement
Example sensors
HoloLens
Project Aria
Automated Clothing Simulation
High-quality Scenes
Egocentric Perception-Driven Motion Synthesis

Figure 3. Overview of *EgoGen*. Through generative motion synthesis (Sec. 3), we further enhance egocentric data diversity by randomly sampling diverse body textures (ethnicity, gender) and 3D textured clothing through an automated clothing simulation pipeline (Sec. 4.2). With high-quality scenes and different egocentric cameras, we can render photorealistic egocentric synthetic data with rich and accurate ground truth annotations (Sec. 4.3).

4.2. Body Texture and Clothing

To enhance *EgoGen*'s synthetic data realism, we dress virtual humans using human textures and 3D clothing assets from BEDLAM [1, 9], including 50 male and 50 female skin albedo textures from seven ethnic groups.

Unlike prior works [9, 119] relying on unscalable commercial software for clothing dynamics simulation, we automate it for diverse synthesized motions and body shapes, minimizing manual effort. Each garment mesh is in a consistent rest pose, i.e., A-Pose (See Fig. 3 middle-left). For each motion sequence, we first repose it to match the body pose in the first frame using linear blend skinning. This involves initializing the clothing geometry by sampling pose and shape blend shapes, along with skinning weights from the nearest multiple SMPL-X vertices in A-Pose. Then we simulate upper and lower garments separately using a state-of-the-art clothing simulation network [28].

4.3. Rendering and Annotations

EgoGen supports simulating diverse head-mounted devices with different camera models, such as fisheye and pinhole

Table 1. Evaluation of motion synthesis in scenes with moving obstacles, multiple humans, and path diversity. ↓: lower is better; ↑: higher is better. The best results in each scenario are in boldface. * denotes an improved version for fair comparison. (Sec. 5.1)

Evaluation	Metrics	GAMMA* [127]	DIMOS* [129]	Ours
Mov. obs.	SR (%) ↑	96	83	**100**
	Dist. (m) ↓	0.29	0.55	**0.06**
	Cont. ↑	0.95	0.96	**0.97**
	Pene-S. (%) ↓	9.2	8.4	**3.4**
2 humans	SR (%) ↑	95	88	**100**
	Dist. (m) ↓	0.32	0.41	**0.07**
	Cont. ↑	0.96	**0.98**	0.97
	Pene-H. ↓	27.6	10.7	**0**
4 humans	SR (%) ↑	92	70	**100**
	Dist. (m) ↓	0.41	0.79	**0.07**
	Cont. ↑	0.94	0.95	**0.96**
	Pene-H. ↓	60.4	41.7	**0**
Diversity	SR (%) ↑	96	84	**97**
	Std Dev ↑	0.987	1.05	**1.21**

cameras. Given the camera's intrinsic parameters and relative poses within the camera rig, we can simulate AR devices like Project Aria glasses [54] and HoloLens [55], facilitating synthetic data generation for real-world applications. Camera extrinsic is determined by our generative human motion model. We use Blender [13] to render photorealistic egocentric image sequences with motion blur. We also render out a rich set of ground truth annotations, such as depth maps, surface normals, segmentation masks, world positions, optical flow, etc for egocentric perception tasks.

5. Experiments

We assess the motion quality, generalizability, and diversity of our motion model, highlighting its ability to generalize to unseen complex tasks and comparing it with recent baselines (Sec. 5.1). We evaluate our proposed egocentric sensing as a depth proxy for enhancing agent exploration (Sec. 5.2) and conduct ablation studies (Sec. 5.3).

We further demonstrate the effectiveness of *EgoGen* on three egocentric computer vision tasks in Sec. 5.4, and 5.5. By incorporating synthesized egocentric images, we can enhance the performance of the state-of-the-art algorithms.

5.1. Evaluation of Learned CAMPs

We assess CAMPs' generalizability in dynamic scenes, including scenes with moving obstacles and scenes with multiple individuals. In tests with moving obstacles, the obstacle blocks the person's path by moving between the person and the goal. In multiple human test scenes, lines from their starting and goal locations intersect in the middle, requiring solving human-human penetrations. See detail in Sec. S4.1.

In Tab. 1, we compare goal-reaching behaviors with two recent baselines: GAMMA [127] and DIMOS [129]. Baseline methods use navigation meshes and path planning for static scenes, while CAMPs can autonomously avoid dynamic obstacles (Sec. 3.3). For fair comparison in dynamic scenes, we extend the baselines by updating navigation meshes and performing on-the-fly path planning at each time step. The tree-based search as in [127] is disabled for all the methods. **Metrics**: (1) *SR*: Success rate for reaching the goal location within a 0.3m threshold. (2) *Dist.*: Average distance of the final pelvis location to the goal. (3) *Cont.*: The contact metric [127] that measures foot-floor contact and foot skating. (4) *Pene-S.*: Percentage of frames with detected human-scene penetration in moving obstacle scenes. (5) *Pene-H.*: Accurate human-human penetration evaluation metric using COAP [56] in multiple human scenes. Please refer to Sec. S4.2 for metric details.

Table 2. Evaluation of egocentric sensing. (Sec. 5.2)

Method (sensing range)	SR (%) ↑	Dist. (m) ↓
Local map [129] (0.8 m)	78	0.35
Local map* (7 m)	4	3.04
Egocentric sensing (ours) (7 m)	**95**	**0.12**

CAMPs outperform the two baselines in dynamic scenarios with moving obstacles and multiple humans, exhibiting lower human-scene and human-human penetrations and a higher goal-reaching success rate. In multiple human scenarios, we observe that in the baselines, dynamically redoing path planning for each human independently can not effectively solve human-human penetration. In contrast, composable CAMPs can generalize well in dynamic settings without using multi-agent RL to synthesize crowd motions.

We assess walking path diversity using the standard deviation of pelvis locations for the same start-target pairs in scenes with a single static box obstacle. As shown in Tab. 1 (Diversity), our approach does not require a pre-computed global path and allows agents to self-explore without being constrained by predefined paths, achieving higher walking path diversity and success rate. This fosters diverse synthetic data generation via more diverse synthesized motion.

5.2. Evaluation of Egocentric Sensing

We assess the exploration ability of our egocentric sensing $\mathcal{E}_t$ in Replica [99] scenes. In Tab. 2, we replace $\mathcal{E}_t$ with a local map [129] in our state $\mathbf{s}_t$, following their encoding method. Relying on local information can trap agents in local optima, e.g., walls beyond their sensing range, resulting in lower SR. Our egocentric sensing acts as a depth proxy, allowing the agent to avoid local optima, explore more effectively than local maps [129] or scandots [3], and achieve higher SR. In addition, our compact representation is more scalable as the sensing range increases, while quadratic local map growth can hinder the policy network's learning.

5.3. Ablation Studies

We compare our policy with several ablations in Tab. 3:
Egocentric depth: an ablation training an egocentric depth

Table 3. Ablation studies. *Note: in our observation,* $\|VP\|_2 > 15$ *indicates abnormal human poses.* (Sec. 5.3)

	SR (%) ↑	$\|\text{VP}\|_2$ ↓	$\cos(\vec{v}, \vec{a})$ ↑
Egocentric depth	8	13.64	0.049
No pretraining	90	28.77	0.918
No attention reward	90	12.26	0.891
Our policy	**92**	**10.57**	**0.940**

image-based policy without the depth sensing proxy. Egocentric depth images are encoded with a CNN;
No pretraining: an ablation training collision avoidance in crowded scenes with strict penetration termination directly;
No attention reward: an ablation for the viewing direction.

We assess pose naturalness with the maximum pose embedding norm encoded with VPoser [67] and evaluate the attention reward with the cosine similarity between the viewing direction $\vec{v}$ and the attention direction $\vec{a}$ (Sec. 3.1).

Directly training RL with egocentric depth images is ineffective due to our high-dimensional action space, emphasizing the value of the compact egocentric sensing representation. Training agents with strict penetration constraints in crowded scenes directly can result in exploring unreasonable action subspaces, given its unbounded Gaussian nature, leading to unrealistic human poses, highlighting the effectiveness of our two-stage RL training scheme. Without the attention reward, the virtual human's capability to attend to a specific direction decreases. All ablation studies are evaluated in Replica. See visuals in Sup. Vid. and Sec. S4.3.

5.4. Mapping, Localization, and Tracking for HMD

Mapping and localization. LaMAR [84] is the first mapping and localization benchmark dataset for AR in large-scale scenes. Despite over a year of extensive data collection, the dataset still lacks exhaustive scene coverage, especially in large open spaces. *EgoGen* can let virtual humans explore large-scale scenes, render dense egocentric views, and build a more complete SfM map by extracting image feature points with SuperPoint [17] and matching images with SuperGlue [83]. Despite synthetic images being noisier due to scene quality, SuperGlue [83] matching can filter out noisy feature points and yield reliable matches.

In Tab. 4, we evaluate *EgoGen* by assessing the localization recall at $(1^\circ, 10cm)$ on the validation set in a lobby of ∼120 sqm of the LaMAR CAB location. In addition, we report the number of triangulated 3D points (#P3D) and track length. *EgoGen* improves the 3D reconstruction by yielding more points for a slightly improved track length and also a significantly better localization performance compared to using the real data only. Ng et al. [62] augments mapping images by perturbing real-world camera poses with noise, which may generate unrealistic camera poses (e.g., stuck in a wall or facing the ceiling), limiting egocentric localization effectiveness. Their method also assumes the availability of initial camera poses, which may not always be feasible. In contrast, *EgoGen* augments by virtual humans *randomly* exploring scenes. Our approach holds promise for creating AR mapping and localization datasets for digital twin scenes without manual data collection, providing enhanced privacy preservation, e.g. no need for anonymization. Refer to Sec. S5.1 for visualization and implementation details.

Table 4. Mapping and localization evaluation. We augment LaMAR with the same amount of images (248 frames) and report the localization recall at $(1^\circ, 10cm)$ on the validation set. *EgoGen* achieves the highest track length and recall. (Sec. 5.4)

	#P3D ↑	Track length ↑	Recall (%) ↑
LaMAR	1929739	5.1946	66.9
Ng et al. [62]	**1937758**	5.1940	74.9
EgoGen	1936169	**5.2105**	**76.7**

Table 5. Egocentric camera tracking evaluation of models trained with and without synthetic data from *EgoGen*. (Sec. 5.4)

	Pose ↓	Rotation ↓	Transl (mm) ↓
Scratch	1.83	0.74	**1303**
+ *EgoGen* pretrain	**1.67**	**0.62**	1305

Egocentric camera tracking. Egocentric camera tracking for HMD aims to yield device pose trajectories in 3D scenes given egocentric video observations. Recovering camera poses from monocular RGB videos using SLAM [101] is a challenging and ill-posed problem due to scale ambiguity. EgoEgo [47] leverages the knowledge of human motion to address the egocentric HMD tracking problem. Specifically, EgoEgo trains a neural network to infer the translation scaling and rotations from egocentric videos, which improves the HMD tracking performance. However, training this model requires jointly captured data of ground truth HMD trajectories and egocentric videos, which are costly to collect. We address this limitation by using *EgoGen* to synthesize quantities of egocentric videos with accurate camera trajectories to pretrain the model, which proves to improve the tracking performance on real data. We conduct experiments on the GIMO [131] dataset that contains ∼200 short sequences of paired motion-video data in 19 scenes. Using *EgoGen*, we synthesize ∼4k sequences of human movements in their scenes and render corresponding egocentric videos using the same camera intrinsic as GIMO and the embodied camera placement described in Sec. 4.1. We also slightly perturb the camera placement location and orientation to simulate the diversity of how people wear HMDs in real data and avoid overfitting to one specific camera placement. We first pretrain the model with synthetic data generated by *EgoGen*, then finetune it on the real GIMO data. Tab. 5 shows the egocentric camera tracking performances for models trained with and without synthetic data. Definitions of evaluation metrics can be found in Sec. S5.2. The finetuned model benefits from *EgoGen* synthetic data and predicts more accurate camera poses compared to the model trained using real data only.

5.5. Human Mesh Recovery from Egocentric views

Human mesh recovery (HMR) is the key to human behavior understanding from the egocentric view, thus crucial for applications in robotics and AR/VR. Given an egocentric RGB or depth image of a target subject, HMR aims to reconstruct the subject's 3D body pose and shape. However, acquiring and annotating real-world data is expensive, demanding, and time-consuming, with egocentric data being particularly scarce. EgoBody [125] is a recent egocentric dataset capturing two-people interactions, with egocentric depth/RGB frames annotated with SMPL-X body meshes. EgoBody provides ∼180k egocentric RGB frames, and merely ∼23k depth frames due to the low frame rate of the depth sensor, with ∼90k/∼10k in the RGB/depth training set. Such limited data is insufficient to train a learning-based model from scratch. In contrast, with *EgoGen*, large-scale synthetic egocentric data can be generated in a time-efficient way. We leverage *EgoGen* to generate quantities of training frames (300k RGB, 105k depth) of humans moving in EgoBody 3D scenes, rendered from the egocentric view, and annotated by SMPL-X parameters of the target subject. Specifically, RGB images are rendered with lifelike human body textures and 3D clothing, with random lighting.

With the recent HMR regressor, ProHMR [45], we show that pre-training with our synthetic data from *EgoGen* enhances the existing method's capability to generalize on real-world scenarios. Evaluated on the real-world EgoBody test set, we compare two training schemes: (1) trained from scratch on the real-world EgoBody training set ("-scratch"), and (2) pre-trained on synthetic data from *EgoGen* and fine-tuned on the real-world EgoBody training set ("-ft").

HMR from depth. As no existing methods were proposed for depth-based HMR task, we adapt ProHMR [45] to the depth input by changing the channel number of the first convolution layer. To mimic real-world sensor noise, synthetic noise [31] is added to the rendered depth. G-MPJPE is additionally reported for depth-based HMR as depth images provide global information. As shown in Tab. 6, compared to the model trained only with a limited amount of real-world data (Depth-scratch), errors are significantly reduced for the model pre-trained with our large-scale synthetic data (Depth-ft), in terms of global translation (22.9% lower G-MPJPE), local pose (20.7% lower MPJPE), and body shape (19.5% lower V2V).

HMR from RGB. For training with RGB images, we apply various data augmentation techniques similar to [9]. Tab. 6 indicates that the RGB-based model pre-trained with large-scale synthetic data ("RGB-ft") also outperforms the model trained only on real-world data ("RGB-scratch"), for both body pose and shape accuracy.

Table 6. Evaluation of HMR on EgoBody test set. "*-scratch" denotes the model trained from scratch with the Egobody training set, and "*-ft" denotes the model pre-trained with *EgoGen* synthetic data. The units for all metrics are in *mm*. (Sec. 5.5)

	G-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓	V2V ↓
Depth-scratch	117.7	82.2	54.1	100.6
Depth-ft	**90.7**	**65.2**	**47.3**	**81.0**
RGB-scratch	-	90.7	59.9	102.1
RGB-ft	-	**85.3**	**56.2**	**97.2**

The enhanced performance highlights that *EgoGen*'s synthetic data effectively compensates for the lack of real-world training data, boosting the performance of current methods when test on real-world data. We will release both of our synthetic EgoBody datasets. See Sec. S5.3 for dataset statistics, qualitative visualizations, and training details.

6. Conclusion

We propose a novel egocentric synthetic data generation approach, *EgoGen*, that uses embodied sensors, a parametric body model, and a generative egocentric perception-driven human motion synthesis method to create egocentric training data with accurate and rich ground truth annotations. By integrating deep reinforcement learning and collision-avoiding motion primitives with egocentric depth proxy, *EgoGen* synthesizes robust human motion and emergent multi-agent behaviors. This paves the way to an efficient and scalable data generation solution that may have a profound impact on egocentric perception tasks.

7. Future Work and Potentials

Human-scene interaction in *EgoGen* is currently coarse. We aim to extend the current method to simulate more detailed human motion driven by egocentric perception, such as hand manipulation, sitting, lying, etc, to facilitate more realistic egocentric synthetic data. We use fixed attention goals to model human attention. Predicting human intention through historical egocentric perception and synthesizing viewing directions based on predicted intention holds significant potential. Synthesizing gaze direction for predicting human intent is valuable but presently hampered by data requirements; we will revisit this when resources allow.

Egocentric perception tasks are broad. We will explore many other egocentric vision tasks with *EgoGen* as this area grows rapidly such as social understanding and forecasting.

EgoGen could also benefit human-robot interaction. For example, our generative human motion model and lifelike human appearances can be integrated into Habitat 3.0 [73] to further close the sim2real gap for robotic agents.

References

[1] Meshcapade GmbH, Tübingen, Germany. https://meshcapade.com,, 2022. 5

[2] Mohammed Abdel-Wahab, Konrad Wenzel, and Dieter Fritsch. Efficient reconstruction of large unordered image datasets for high accuracy photogrammetric applications. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia. XXII ISPRS Congress*, 2012. 3

[3] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, pages 403–415. PMLR, 2022. 3, 6

[4] M. Pilar Aivar, Eli Brenner, and Jeroen B. J. Smeets. Avoiding moving obstacles. *Experimental Brain Research*, 190 (3):251–264, 2008. 5

[5] Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik. Unrealego: A new dataset for robust egocentric 3d human motion capture. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[6] Marcin Andrychowicz, Anton Raichuk, Piotr Stanczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? A large-scale empirical study. *CoRR*, abs/2006.05990, 2020. 3

[7] Apple. ARKit. https://developer.apple.com/arkit/, 2017. 3

[8] Sonia Baltodano, Srinath Sibi, Nikolas Martelaro, Nikhil Gowda, and Wendy Ju. The rrads platform: a real road autonomous driving simulator. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 281–288, 2015. 2

[9] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 5, 8, 4, 6

[10] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578, 2016. 3

[11] Zhongang Cai, Mingyuan Zhang, Jiawei Ren, Chen Wei, Daxuan Ren, Zhengyu Lin, Haiyu Zhao, Lei Yang, and Ziwei Liu. Playing for 3d human recovery. *arXiv preprint arXiv:2110.07588*, 2021. 2

[12] Simon Clavet. Motion matching and the road to next-gen animation. In *Proc. of GDC*, 2016. 2

[13] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5, 6, 3

[14] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021. 2

[15] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021. 2

[16] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003. 3

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. 7, 6

[18] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. 2

[19] Salehe Erfanian Ebadi, Saurav Dhakad, Sanjay Vishwakarma, Chunpu Wang, You-Cyuan Jhang, Maciek Chociej, Adam Crespi, Alex Thaman, and Sujoy Ganguly. Psp-hdri+: A synthetic dataset generator for pre-training of human-centric computer vision models. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*, 2022. 2

[20] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[21] Alexander Fiannaca, Ilias Apostolopoulous, and Eelke Folmer. Headlock: A wearable navigation aid that helps blind cane users traverse large open spaces. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, pages 19–26, 2014. 2

[22] Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu. Vrkitchen: an interactive 3d virtual environment for task-oriented learning. *arXiv*, abs/1903.05757, 2019. 2

[23] Zahra Ghodsi, Siva Kumar Sastry Hari, Iuri Frosio, Timothy Tsai, Alejandro Troccoli, Stephen W. Keckler, Siddharth Garg, and Anima Anandkumar. Generating and characterizing scenarios for safety testing of autonomous vehicles, 2021. 2

[24] Google. ARCore. https://developers.google.com/ar/, 2018. 3

[25] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *CVPR*, 2001. 3

[26] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico

Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[27] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam H. Laradji, Hsueh-Ti Derek Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, A. Cengiz Öztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 3739–3751. IEEE, 2022. 2

[28] Artur Grigorev, Michael J. Black, and Otmar Hilliges. HOOD: hierarchical graphs for generalized modelling of clothing dynamics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 16965–16974. IEEE, 2023. 5, 4

[29] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 3

[30] Norbert Haala, Michael Cramer, Florian Weimer, and Martin Trittler. Performance test on uav-based photogrammetric data collection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012. 3

[31] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. *ICRA*, 2014. 8

[32] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 160–177. Springer, 2016. 3

[33] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proceedings of the International Conference on Computer Vision 2021*, 2021. 2, 3

[34] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. *arXiv preprint arXiv:2302.00883*, 2023. 2

[35] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Trans. Graph.*, 36(4):1–13, 2017. 2

[36] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020. 2

[37] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *IJCV*, 2021. 3

[38] Neena Kamath. Announcing Azure Spatial Anchors for collaborative, cross-platform mixed reality apps. https://azure.microsoft.com/en-us/blog/announcing-azure-spatial-anchors-for-collaborative-cross-platform-mixed-reality-apps/, 2019. 3

[39] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 3

[40] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DoF Camera Relocalization. In *ICCV*, 2015. 3

[41] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007. 3

[42] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regressor for 3D human body estimation. In *Proceedings International Conference on Computer Vision (ICCV)*, pages 11127–11137. IEEE, 2021. 3

[43] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2252–2261, 2019.

[44] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019.

[45] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11605–11614, 2021. 3, 8

[46] Lucas Kovar, Michael Gleicher, and Frédéric H. Pighin. Motion graphs. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2008, Los Angeles, California, USA, August 11-15, 2008, Classes*, pages 51:1–51:10. ACM, 2008. 2

[47] Jiaman Li, C. Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, pages 17142–17151. IEEE, 2023. 1, 2, 7

[48] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying location information in full frames into human pose and shape estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 590–606. Springer, 2022. 3

[49] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Trans. Graph.*, 39(4), 2020. 3

[50] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4d human body capture from egocentric video via 3d scene grounding. In *2021 International Conference on 3D Vision (3DV)*, pages 930–939. IEEE, 2021. 3

[51] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, Shun Iwase, and Kris M Kitani. Kinematics-guided reinforcement learning for object-aware 3d ego-pose estimation. *arXiv preprint arXiv:2011.04837*, 2020. 3

[52] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 3

[53] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Residual pose: A decoupled approach for depth-based 3d human pose estimation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10313–10318. IEEE, 2020. 3

[54] Meta. Project Aria Glasses. https://www.projectaria.com/, 2023. 6

[55] Microsoft. HoloLens 2. https://www.microsoft.com/en-us/hololens, 2019. 6

[56] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. COAP: Compositional articulated occupancy of people. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 6, 5

[57] Aymen Mir, Xavier Puig, Angjoo Kanazawa, and Gerard Pons-Moll. Generating continual human motion in diverse 3d scenes. *CoRR*, abs/2304.02061, 2023. 2

[58] Michael Montemerlo, Nicholas Roy, and Sebastian Thrun. Perspectives on standardization in mobile robot programming: the carnegie mellon navigation (CARMEN) toolkit. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, October 27 - November 1, 2003*, pages 2436–2441. IEEE, 2003. 2

[59] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 3

[60] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real time localization and 3d reconstruction. In *CVPR*, 2006. 3

[61] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. You2me: Inferring body pose in egocentric video via first and second person interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9890–9900, 2020. 3

[62] Tony Ng, Adrian Lopez-Rodriguez, Vassileios Balntas, and Krystian Mikolajczyk. Reassessing the limitations of cnn methods for camera pose regression, 2021. 7, 5, 6

[63] Niantic. Niantic Expands Developer Platform and AR Tools with Niantic Lightship. https://nianticlabs.com/news/lightship/, 2021. 3

[64] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *CVPR*, 2004. 3

[65] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar M. Parkhi, Richard A. Newcombe, and Carl Yuheng Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. *CoRR*, abs/2306.06362, 2023. 2

[66] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4045–4054, Stockholmsmässan, Stockholm Sweden, 2018. PMLR. 4

[67] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 3, 5, 7, 2

[68] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021. 2

[69] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae, 2021. 3

[70] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *IJCV*, 2004. 3

[71] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018. 1, 2

[72] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B. Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-{ai} collaboration. In *International Conference on Learning Representations*, 2021. 1, 2

[73] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimir Vondrus, Théophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots. *CoRR*, abs/2310.13724, 2023. 2, 8, 1

[74] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *International Conference in Computer Vision (ICCV)*, 2019. 2

[75] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, page 5. Kobe, Japan, 2009. 2

[76] Umer Rafi, Juergen Gall, and Bastian Leibe. A semantic occlusion model for human pose estimation from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 67–74, 2015. 3

[77] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12630–12641, 2023. 2

[78] Tilman Reinhardt. Google Visual Positioning Service. `https://ai.googleblog.com/2019/02/using-global-localization-to-improve.html`, 2019. 3

[79] 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. 3

[80] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion, 2023. 2, 3

[81] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020. 2

[82] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Robotics: Science and Systems (RSS)*, 2020. 2

[83] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020. 7, 6

[84] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*, 2022. 3, 7

[85] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DoF outdoor visual localization in changing conditions. In *CVPR*, 2018. 3

[86] 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, 2019. 2

[87] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 3, 6

[88] Johannes L. Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, 2017. 3

[89] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 4

[90] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Synthetic training for accurate 3d human pose and shape estimation in the wild. In *British Machine Vision Conference (BMVC)*, 2020. 2

[91] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. Motion capture from body-mounted cameras. *ACM Trans. Graph.*, 30(4):31, 2011. 3

[92] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2821–2840, 2012. 3

[93] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *CVPR*, 2013. 3

[94] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, 2013. 2

[95] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006. 3

[96] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 3

[97] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 2

[98] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4): 1–13, 2022. 2

[99] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard New-

combe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4, 6

[100] 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. 2, 1

[101] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021. 7

[102] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 3

[103] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. 2

[104] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 2

[105] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7728–7738, 2019. 2, 3

[106] Denis Tome, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando De la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *arXiv preprint arXiv:2011.01519*, 2020. 3

[107] Unity Technologies. Unity Perception package. `https://github.com/Unity-Technologies/com.unity.perception`, 2020. 2

[108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[109] Keze Wang, Shengfu Zhai, Hui Cheng, Xiaodan Liang, and Liang Lin. Human pose estimation from depth images via inference embedded multi-task learning. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1227–1236, 2016. 3

[110] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20270–20281, 2023. 2

[111] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2023. 2

[112] Jiayi Weng, Huayu Chen, Dong Yan, Kaichao You, Alexis Duburcq, Minghao Zhang, Yi Su, Hang Su, and Jun Zhu. Tianshou: A highly modularized deep reinforcement learning library. *Journal of Machine Learning Research*, 23 (267):1–6, 2022. 3

[113] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 3

[114] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 3

[115] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Trans. Graph.*, 41(4), 2022. 2

[116] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021. 2

[117] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019. 3

[118] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 2

[119] Zhitao Yang, Zhongang Cai, Haiyi Mei, Shuai Liu, Zhaoxi Chen, Weiye Xiao, Yukun Wei, Zhongfei Qing, Chen Wei, Bo Dai, Wayne Wu, Chen Qian, Dahua Lin, Ziwei Liu, and Lei Yang. Synbody: Synthetic dataset with layered human models for 3d human perception and modeling, 2023. 5, 2, 4

[120] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Trans. Graph.*, 41(6), 2022. 3

[121] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738. IEEE, 2011. 3

[122] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. *arXiv preprint arXiv:2302.12827*, 2023. 3

[123] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10082–10092, 2019. 3

[124] 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. 3

[125] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In *European conference on computer vision (ECCV)*, 2022. 1, 2, 8, 6

[126] Siwei Zhang, Qianli Ma, Yan Zhang, Sadegh Aliakbarian, Darren Cosker, and Siyu Tang. Probabilistic human mesh recovery in 3d scenes from egocentric views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[127] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20481–20491, 2022. 3, 6, 5

[128] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021. 3

[129] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, , and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. In *International conference on computer vision (ICCV)*, 2023. 2, 3, 4, 6

[130] Yuhang Zhao, Elizabeth Kupferstein, Brenda Veronica Castro, Steven Feiner, and Shiri Azenkot. Designing ar visualizations to facilitate stair navigation for people with low vision. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pages 387–402, 2019. 2

[131] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, C Karen Liu, and Leonidas J Guibas. Gimo: Gaze-informed human motion prediction in context. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 676–694. Springer, 2022. 7

[132] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 2

[133] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023. 2, 3

EgoGen: An Egocentric Synthetic Data Generator

Supplementary Material

S1. Related Work

EgoGen addresses the gap in egocentric synthetic data generation specifically tailored for head-mounted devices, situated at the intersection of three key areas: 1) General synthetic data generation; 2) Egocentric simulation for embodied agents; 3) Human-related synthetic data generation. For a more detailed understanding of the distinctions between *EgoGen* and existing methods, refer to Tab. S1, where these three areas are clearly outlined within distinct blocks.

In particular, VirtualHome [71, 72] also provides rendered egocentric views from head-mounted cameras. However, their egocentric videos lack fluctuating patterns due to the absence of natural human motion; instead, they display robotic-like patterns as Habitat 2.0 [100]. We advance closer to synthesizing more realistic data for head-mounted devices. In addition, they lack a generative human motion model, whereas ours can generate more diverse human motion and trajectories. A very recent work Habitat 3.0 [73] introduced virtual humans to robotic simulation. However, their human locomotion is synthesized by cyclically replaying a walking motion clip from MoCap data along a pre-calculated path with rigid rotations to transition to the next walking direction. Both VirtualHome and Habitat 3.0 have a limited number of human agents and fall short in representing diverse human characters, with limitations in body shapes, ethnic variation, and clothing options.

Our synthetic data generation achieves increased diversity by incorporating a walking path-free generative human motion model, diverse body shapes, various body textures, and varied 3D textured clothing.

S2. Ego-Sensing Driven Motion Synthesis

S2.1. Egocentric Sensing Calculation

As a compact and cheap-to-compute representation of depth maps, egocentric sensing resembles the calculation of depth information but is simplified into 2D.

As shown in Fig. S3, the location of the egocentric camera is the midpoint of two eyeballs, and the viewing direction $\vec{\mathbf{v}}$ is visualized as the red arrow. N rays are cast from the location of the egocentric camera, with the central direction of these rays determined by the 2D projection of $\vec{\mathbf{v}}$. The starting points of these rays are identical, while their endpoints form a semicircle in front of the virtual human, representing the field of view $[\theta_{min}, \theta_{max}]$ to the human. Each ray has the potential to extend infinitely. In our implementation, $N = 32$, $\theta_{min} = -90°$, $\theta_{max} = 90°$.

The 2D collision detection of rays leverages the 2D layout of the 3D scene. For illustration purposes, we simplify the obstacles in 3D scenes with grey rectangles and visualize the collision detection of rays in Fig. S1. The egocentric sensing encodes the simplified depth of obstacles.

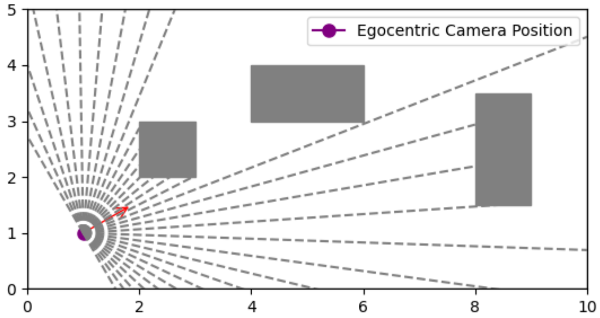


Figure S1. The 2D projection of the egocentric camera location is represented by the purple point, while the 2D projection of the viewing direction $\vec{\mathbf{v}}$ is indicated by the red arrow. The field of view changes due to the head pose.

S2.2. Reward, Weighting, and Training Detail

In our motion primitive environment, we design an intuitive set of rewards to encourage the agent to perform realistic human motions (all vectors are normalized in the following equations):

Attention reward that encourages the human to look at the goal:

$$r_{attention} = \frac{\langle \vec{\mathbf{v}}, g - h \rangle + 1}{2}, \tag{1}$$

where $\vec{\mathbf{v}}$ denotes the viewing direction, g and h denote the goal location and current head location and $\langle \cdot \rangle$ denotes the inner product.

Foot contact reward $r_{cont.}$ contains two components: Foot floor distance reward and foot skating reward.

$$r_{cont.} = r_{floor} + r_{skate} \tag{2}$$

$$r_{floor} = e^{-(|\min_{x \in F} x_z| - 0.02)_+}, \tag{3}$$

$$r_{skate} = e^{-(\min_{x \in F} \|x_{vel}\|_2 - 0.075)_+}, \tag{4}$$

where F denotes foot markers, x_z denotes the marker height, x_{vel} denotes the marker velocity, and $(\cdot)_+$ denotes clipping negative values. There are tolerance thresholds of 0.02m for foot-floor distance and 0.075m/s for skating.

Goal Distance reward that encourages the agent to get closer to the goal at each step:

$$r_{dist} = d^{t-1} - d^t, \tag{5}$$

Table S1. Comparison of existing synthetic datasets or generators. (Sec. S1)

	Domain	Egocentric	Head-mounted	Multi-Camera rigs	Virtual Humans	Automated Clothing Simulation	Generative Human Motion Synthesis
Kubric [27]	Scattered Objects	✗	✗	✗	✓	✗	✗
PointOdyssey [132]	Point Tracking	✓	✓	✗	✓	✗	✗
InfiniGen [77]	Natural	✗	✗	✗	✗	✗	✗
RoboGen [111]	Robotics	✗	✗	✗	✗	✗	✗
UniSim [118]	Real-world Interaction	✓	✓	✗	✗	✗	✗
UnityPerception [19, 107]	Object Detection Pose Estimation	✗	✗	✗	✓	rigged clothing	✗
uHumans2 [82]	Scene Graphs	✓	✗	✗	✓	✗	✗
Carla [18]	Driving	✓	✗	✗	✓	rigged clothing	✗
VirtualHome [71, 72]	Household Simulation	✓	✓	✗	✓	rigged clothing	✗
VRKitchen [22]	Cooking Simulation	✓	✓	✗	✗	✗	✗
Habitat 2.0 [100]	Embodied Robots	✓	✗	✗	✗	✗	✗
Habitat 3.0 [73]	Human-robot Interaction	✓	✗	✗	✓	✗	✗
UnrealEgo [5]	Ego-pose Estimation	✓	✓	✓	✓	rigged clothing	✗
GTA-Human [11]	Pose Estimation	✗	✗	✗	✓	✗	✗
BEDLAM [9]	Pose Estimation	✗	✗	✗	✓	✗	✗
SynBody [119]	Pose Estimation	✗	✗	✗	✓	✗	✗
ADT [65]	Digital Twin	✓	✓	✓	✗	✗	✗
EgoGen (ours)	Head-mounted Devices	✓	✓	✓	✓	✓	✓

Here d^t denotes the body-goal distance at step t.

Body orientation reward that encourages the body forward direction to be aligned with the goal location direction:

$$r_{ori} = \frac{\langle o_b, g - p \rangle + 1}{2}, \tag{6}$$

where o_b denotes the body forward orientation, g and p denote the goal location and current pelvis location, and $\langle \cdot \rangle$ denotes inner product. Different from the *attention reward* that drives the head motion, this penalizes backward movement toward the goal.

Penetration reward that penalizes the intersection of the human body and obstacles. We use different penetration rewards in different settings.

When training in sparse scenes, e.g., a single static box obstacle, penetration detection is simplified into 2D to accelerate calculation:

$$r_{pene}^{sparse} = \begin{cases} 0.05, & |\mathcal{M}_0 \cap \mathcal{B}_{xy}(X)| < thres \\ 0, & otherwise \end{cases} \tag{7}$$

where $\mathcal{M}_0$ denotes the non-walkable cells on the ground plane, $\mathcal{B}_{xy}(.)$ denotes the 2D bounding box of the body markers X, $\cap$ denotes their intersection, and $|\cdot|$ denotes the number of non-walkable cells within the bounding box of the human. $thres$ is set to 3 and the cell dimension is $0.1m \times 0.1m$.

When training in crowded scenes, we use the signed distance field (Ψ_O) for precise penetration detection:

$$r_{pene}^{crowded} = e^{-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|V|}|(\Psi_O(\mathbf{v}_{ti}))_-|} \tag{8}$$

where $|V|$ denotes the number of SMPL-X mesh vertices, T denotes the number of frames in our motion primitive ($T = 20$), $\mathbf{v}$ denotes SMPL-X mesh vertex, and $(\cdot)_-$ denotes clipping positive values. The penetration reward penalizes body vertices with negative SDF values within a motion primitive.

Pose reward that penalizes generating unrealistic human poses using VPoser [67] body pose prior:

$$r_{pose} = \begin{cases} 0.05, & \|VP\|_2 < thres \\ 0, & otherwise \end{cases} \tag{9}$$

where $\|\text{VP}\|_2$ denotes the pose embedding inferred by the VPoser encoder $\mu(\cdot)$, where $\|\text{VP}\|_2 = |\mu(\theta)|_2$. θ denotes the SMPL-X body pose parameter representation. The VPoser pose prior learns a probabilistic pose distribution where vectors closing to zero have a high probability and correspond to realistic human poses. In our observation, $\|\text{VP}\|_2 > 15$ produces unrealistic human poses. $thres$ is set to 11.

Success reward for reaching the goal location:

$$r_{succ} = \begin{cases} 1, & d < thres \\ 0, & otherwise \end{cases} \tag{10}$$

where d is the body-goal distance. $thres$ is set to 0.1.

The weights for each reward are listed in Tab. S2. The weighting of each reward is determined according to the reward value. For example, the goal distance reward measures the distance change in one motion primitive spanning 0.5s, which is approximately 10 times smaller than other rewards. As a result, its weight is 10 times bigger than others. We observe high foot skating weight helps to reduce foot skating. Higher success rewards encourage the agent to reach the goal. But on the other hand, the weight can not be too big. Because we did not do reward normalization, too large values may lead to big errors in value estimation and training instabilities.

Reward	Weight
Foot floor distance	0.1
Foot skating	0.3
Goal distance	1
Body orientation	0.1
Attention	0.3
Penetration pretraining	1
Penetration finetuning	0.1
Pose	0.1
Success	0.5

Table S2. Reward weights.

Penetration Termination. We terminate an episode due to penetration using different criteria. In sparse scenes, an episode is terminated if $r_{pene}^{sparse} = 0$.

As mentioned in Sec. 3.2 in crowded scenes, we employ a two-stage RL training scheme. In stage I, we pretrain the policy with a penetration weight of $w_{r_{pene}} = 1$ to more effectively encourage the virtual human to avoid obstacles and explicitly **not** perform penetration termination. After convergence, in stage II, we proceed to fine-tune the policy with a strict penetration termination using a reduced penetration weight of $w_{r_{pene}} = 0.1$. Penetration detection involves considering the maximum number of body vertices in penetration within a motion primitive. An episode is terminated if:

$$\max_t \sum_{i=1}^{|V|} |(\Psi_O(\mathbf{v}_{ti}))_-| \geq thres \tag{11}$$

where $thres$ is set to 40.

This design has several reasons: 1) Our action space is an unbounded Gaussian, direct training with strict termination can lead the policy to explore unreasonable spaces and produce unrealistic human poses, see Fig. S7 for illustration. 2) Reducing penetration weight during fine-tuning can amplify the significance of the goal-reaching weight, encouraging goal-reaching behaviors.

S2.3. PPO

Our PPO implementation is based on Tianshou [112]. We list the hyperparameters of PPO in Tab. S3. c_1, c_2 are defined in Sec. 3.2. “Repeat per Collect” is the training iterations with the same collected rollouts.

The majority of the hyperparameters were set to their default values. To note, adopting smaller values of “PPO Clip Threshold” and “Repeat per Collect” will not update parameters too drastically and thus stabilize the training. We use advantage normalization without value function clipping.

Param	Value
Learning Rate	3e-4
γ Discount	0.99
PPO Clip Threshold	0.1
Repeat per Collect	1
Value Function Coefficient (c_1)	1
Entropy Coefficient (c_2)	0.01
GAE (λ)	0.95
Max Grad. Norm	0.1

Table S3. PPO hyperparameters.

Another trick we adopted is that we performed the last policy layer weight scaling, which makes initial actions close to the standard normal distribution, which can boost the performance [6].

The training time is roughly 20 hours on a GeForce RTX 3090 GPU with batch size 256, 20000 steps per epoch.

The key difference between our environment with others is that our action space is not strictly bounded. The motion primitive model $\mathcal{P}$ is based on VAE and is pretrained with a KLD loss w.r.t. a standard normal distribution. As a result, we do not do any action scaling or clipping during training. Due to the nature of our action space, the learned policy can deviate too much from the standard normal distribution. As a result, we select the best model using the best test reward and minimum KL divergence between the learned policy action space and the standard normal distribution.

S3. Egocentric Synthetic Data Generation

S3.1. Embodied Camera Placement

We support various camera placements in *EgoGen*.

For egocentric sensing-driven motion synthesis (Sec. 3), we place one camera at the midpoint of two eyeballs and the viewing direction $\vec{\mathbf{v}}$ is shown in Fig. S3. We use the SMPL-X [67] armature in Blender [13] to calculate $\vec{\mathbf{v}}$. The two eye bones are visualized in Fig. S2.

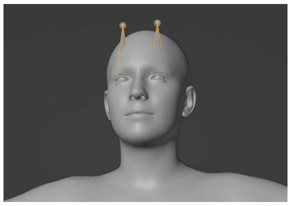


Figure S2. Eye bones are located at the eyes and are highlighted with orange edges.

EgoGen also supports multi-camera rigs simulation. With the information about the relative poses of cameras within a rig, we have the flexibility to position the camera at various locations on the head.

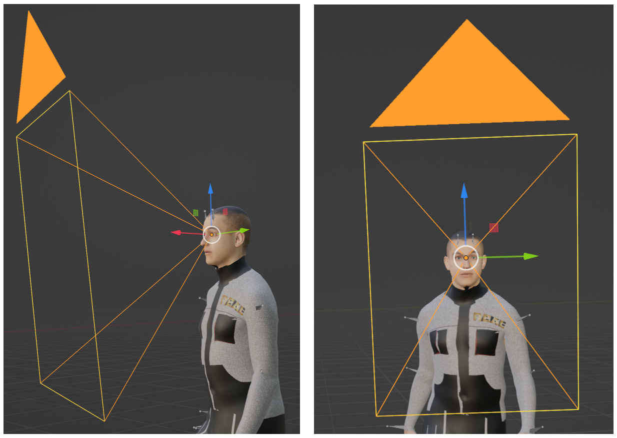


Figure S3. Illustration of embodied camera placement. The camera axes are determined by 1) The blue arrow of the eye bone (from *root* to *tip*); 2) The green arrow from one eyeball to another; 3) The red arrow representing the viewing direction $\vec{v}$. (Sec. S2.1, Sec. S3.1)

S3.2. Automated Clothing Simulation

As shown in Tab. S1, many prior works resort to generating synthetic data with rigged clothing, with unrealistic clothing deformations. In contrast, BEDLAM [9] and Synbody [119] incorporate physics-based clothing simulation to enhance realism and allow for dressing a diverse range of body shapes in a wide array of clothing. However, their approaches are not scalable for handling arbitrary motion sequences produced by our generative human motion model.

We further automate clothing dynamics simulation with the state-of-the-art clothing simulation network HOOD [28]. HOOD treats each garment as a single graph and predicts graph deformations due to both gravity and collisions with the human body mesh.

First, we perform preprocessing on the 3D clothing mesh from [9] to separate the upper garment and lower garment into distinct clothing meshes because HOOD can not handle disconnected graphs as input. Second, we sample pose blend shapes, shape blend shapes, and average skinning weights from the closest n SMPL-X mesh vertices in A-Pose, where $n = 1$ for tight garments such as pants and $n = 1000$ for loose garments such as dresses. We repose the clothing meshes in A-Pose to match the body pose in the first frame of a synthesized motion sequence. Then, for lower garments, the vertices in the top ring are fixed to the body to prevent dropping due to gravity. Finally, we simulate the upper and lower garments separately using HOOD.

S3.3. More Examples of Available Annotations

In addition to the fisheye cameras shown in the teaser, here we show more ground-truth annotations with perspective cameras, including RGBD, optical flow, bounding boxes, segmentation masks, and surface normals in Fig. S4.

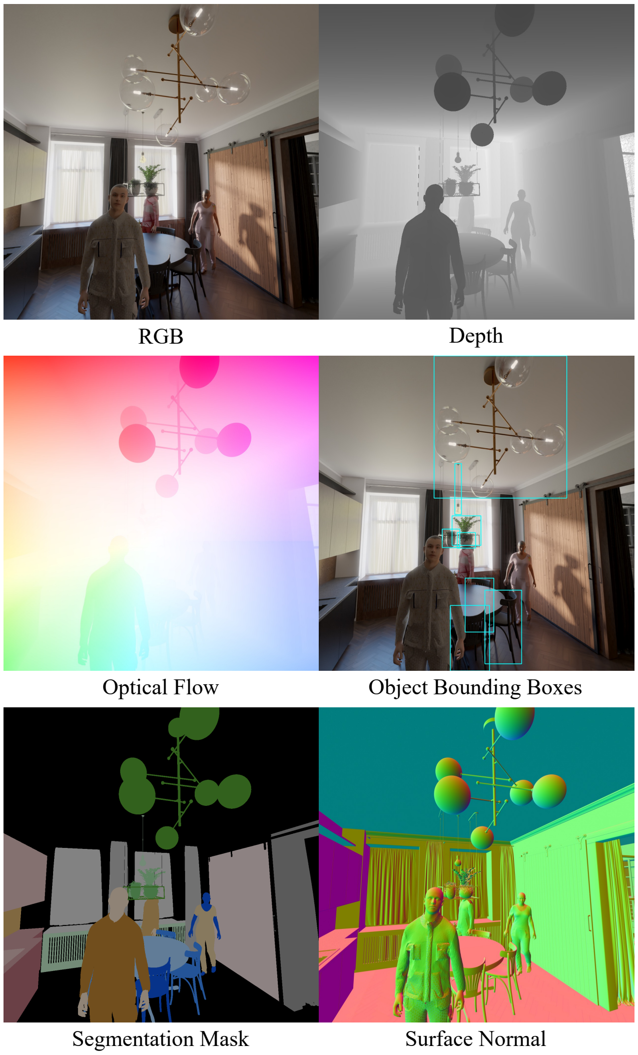


Figure S4. Ground-truth annotations from perspective cameras.

S4. Experiments

S4.1. Test Scenarios in Evaluation of CAMPs

We provide visualizations of how we built test scenarios. Please refer to Sup. Vid. for qualitative results.

Moving obstacle. Refer to Fig. S5 for an illustration of the evaluation in scenes with moving obstacle. The moving obstacle will move between the human and its goal location.

Multiple humans. To visually demonstrate, as depicted in Fig. S6, we initiate four virtual humans from distinct points in the figure. We require they walk to the opposite location across the origin, either from A to B or from B to

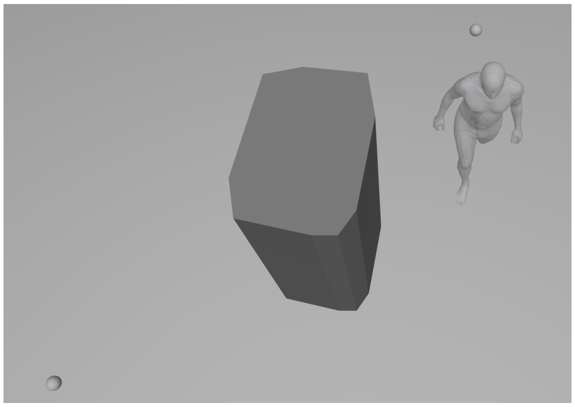


Figure S5. Moving obstacle test scenario illustration.

A. There are no other obstacles. Please refer to Sup. Vid. for qualitative results.

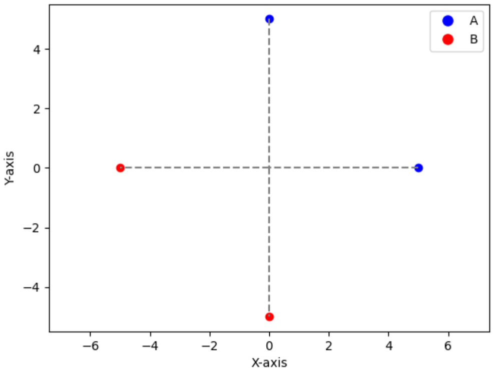


Figure S6. Multiple humans crowd motion test scenario illustration.

Path diversity. We assess walking path diversity with a static obstacle fixed at the midpoint between the start and target locations. Similar to Fig. S5, but the obstacle is not moving anymore.

S4.2. Evaluation Metrics

The Foot contact metric [127] reaches 1 when there is foot-floor contact and no foot skating, defined as:

$$s_{contact} = e^{-(|\min x_z|-0.05)_+} \cdot e^{-(\min \|x_{vel}\|_2-0.075)_+}, \tag{12}$$

where x_z and x_{vel} denote the marker height and velocity, 0.05 and 0.075 are tolerance thresholds, and $(\cdot)_+$ denotes clipping with the lowerbound of 0.

For moving obstacle scenes, we evaluate human-scene penetration by detecting all frames where the floor plane projections of any body parts and obstacles have intersections. For multiple human scenes, we measure the accurate human-human penetration using the implicit human body occupancy model COAP [56], which predicts the body occupancy given spatial location queries. Since the articulated human bodies are complex and require accurate penetration detection, we detect whether one human collides with other humans by querying its body vertices using the occupancy field of all other humans at that frame and report the occupancy values for human-human penetration.

S4.3. Ablation Studies

No pretraining. Without our two-stage RL training scheme, direct penetration termination in crowded scenes will result in unrealistic predicted human poses. As shown in Tab. 3, where $\|\text{VP}\| = 28.77 > 15$, we provide visualizations of the corresponding unnatural poses in Fig. S7. In contrast, our full model works well. Please refer to Sup. Vid.

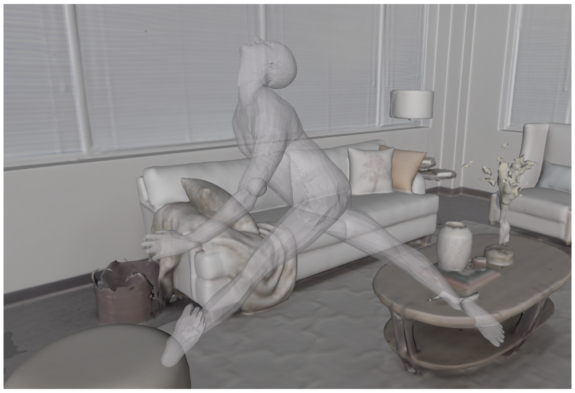


Figure S7. Ablation of our two-stage RL training. Without pretraining, the model can produce unrealistic human poses.

S5. Egocentric Perception Tasks

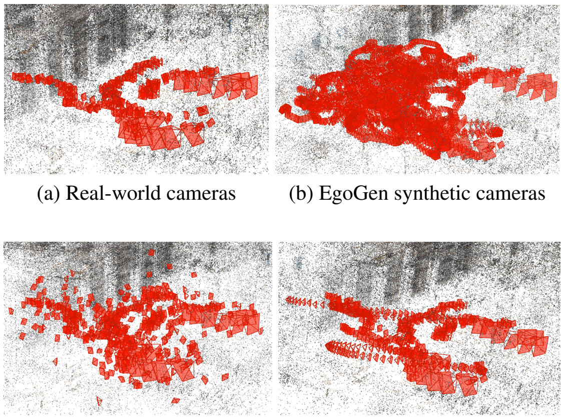
S5.1. Mapping and Localization for AR

As shown in Fig. S8, we can leverage *EgoGen* to explore the large-scale scene, add synthetic egocentric images into the dataset, and build a more complete Structure-from-Motion (SfM) map (Fig. S8b). In our implementation, we randomly set the starting and target locations of virtual humans. Compared with [62] that perturbs real-world cameras with random noise (Fig. S8c) that may result in unrealistic camera poses, *EgoGen* can simulate human trajectories and motion (Fig. S8d).

The efficacy of synthetic data for a task relies on the domain gap between synthetic and real-world images. In the SfM pipeline in our experiment, the render-to-real gap can influence the result of the feature extraction. As shown in Fig. S9 on the left, detected feature points with SuperPoint [17] are much noisier in synthetic images due to scene quality, which can make feature matching challenging. In addition, the feature matcher SuperGlue [83] exhibits overfitting behaviors: visually similar images are preferred to be matched first, i.e., it tends to match sim-sim and real-real pairs only. As a result, simply adding synthetic images into the real-world dataset will result in no matches between synthetic and real-world images, making it impossible to improve localization recall.

To ensure valid matching between synthetic and real-world mapping images, during the pair selection process using SuperGlue, we force synthetic images to match with real images only. By implementing this approach, we can achieve a denser SfM map by establishing matches between synthetic and real-world 2D image feature points (see Fig. S9) and thereby triangulating more 3D points.

To enhance the localization performance of real-world query images using the augmented SfM map, we enforce matches with both synthetic and real mapping images for all query images. This ensures that real-world query images can be paired with synthetic mapping images, leveraging a denser SfM map and enhancing localization recall.



(a) Real-world cameras (b) EgoGen synthetic cameras

(c) Perturbing existing cameras (d) EgoGen same # of cams as (c)

Figure S8. EgoGen addresses the issue of sparsity by populating the dataset with synthetic images. In Fig. S8a, the sparsity of real-world mapping images is apparent, where each red object represents a camera and each colored dot represents a triangulated 3D point. After applying EgoGen, mapping images are more densely distributed, resulting in denser 3D triangulated points, as shown in Fig. S8b. In Fig. S8c and Fig. S8d, we augment S8a with the same amount of synthetic images using [62] and EgoGen respectively. EgoGen generates synthetic data with a similar distribution as human trajectories as illustrated in S8d. Results are visualized using Colmap [87]. Note that we only visualize a subset of cameras here.

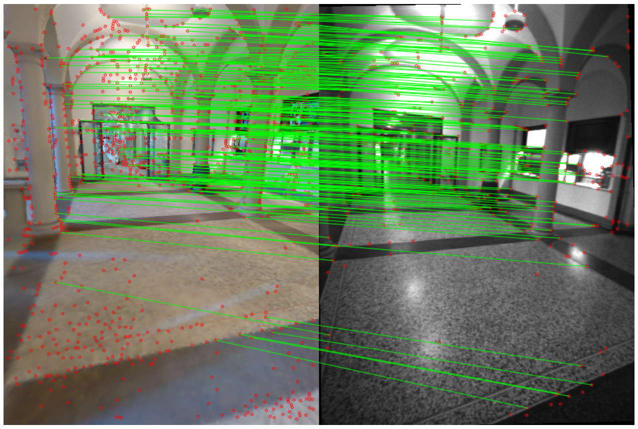


Figure S9. Feature matching visualization for a render-to-real image pair.

S5.2. Egocentric Camera Tracking

The egocentric camera tracking task is evaluated using the head rotation error, translation error, and pose error that jointly accounts for both rotation and translation. The head rotation error calculates the Frobenius norm of the difference between the matrix representations of the predicted rotation R_{pred} and the ground truth rotation R_{gt}, which is defined as:

$$e_{rotation} = \|R_{pred}R_{gt}^{-1}\|_2, \tag{13}$$

The head translation error is computed as the mean Euclidean distance of two sequences of head translations. The results are reported in the unit of millimeter.

The head pose error calculates the Frobenius norm of the difference between the transformation matrix of the predicted head pose and ground truth head pose, which is given by:

$$e_{pose} = \|T_{pred}T_{gt}^{-1}\|_2, \tag{14}$$

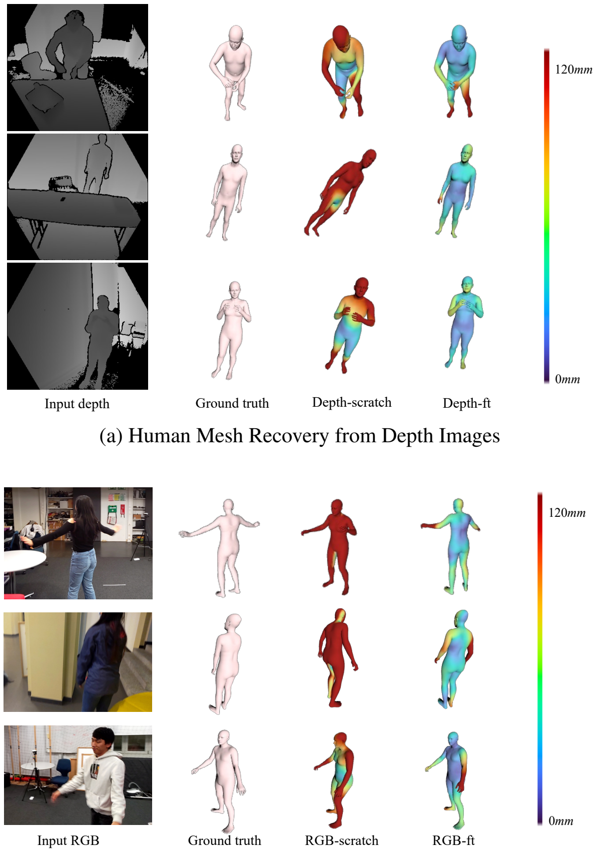
S5.3. Human Mesh Recovery from Egocentric Views

We simulate the data collection process of Egobody [125] and let two virtual humans walk in the scanned scene meshes from Egobody. We randomly sample *gender*, *body shape*, and *initial body pose* and synthesize human motions with our proposed generative human motion model to increase data diversity.

The egocentric camera is attached to both humans and we render the interactee from the camera wearer's egocentric view. Camera intrinsic is set similarly to the real-world camera. For depth data generation, we omit the clothing because the simulated depth sensor noise will remove detail. For RGB data generation, to further increase data diversity and close the sim-real gap, we randomly sample body texture and 3D textured clothing meshes from BEDLAM [9] and perform automated clothing simulation

(Sec. S3.2) given arbitrary synthesized human motion sequences from our generative human motion model. In addition, we adopt random lighting in the rendering. In total, we synthesized 105k depth images and 300k RGB images with diverse body shapes, poses, skin textures, and clothing, along with ground-truth SMPL-X annotations. We will release both of our synthetic datasets as a complement to Egobody.

Qualitative results. We visualize our qualitative results for HMR from depth in Fig. S10a and HMR from RGB in Fig. S10b on real-world test data. With large-scale synthetic data from *EgoGen*, we can compensate for the lack of real-world data and improve the performance of current models. "*-scratch" denotes models trained only with limited real-world data. "*-ft" denotes models pretrained with our large-scale synthetic data and then finetuned with real-world data.

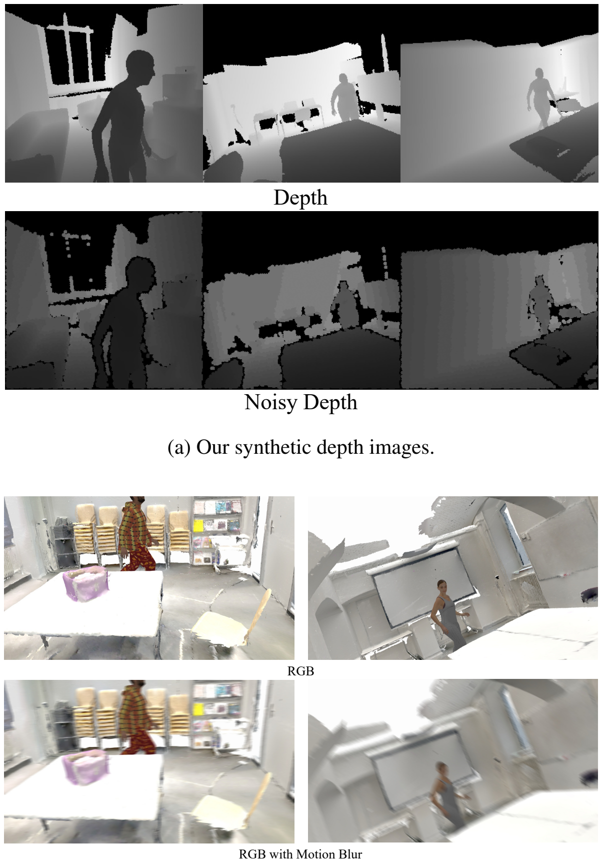


(a) Human Mesh Recovery from Depth Images

(b) Human Mesh Recovery from RGB Images

Figure S10. Qualitative results of HMR on EgoBody test set. The body mesh color of the last two columns denotes the per-vertex error between the predicted body and the ground truth.

Synthetic Data Samples. We show some examples of synthetic data from *EgoGen* in Fig. S11.



(a) Our synthetic depth images.

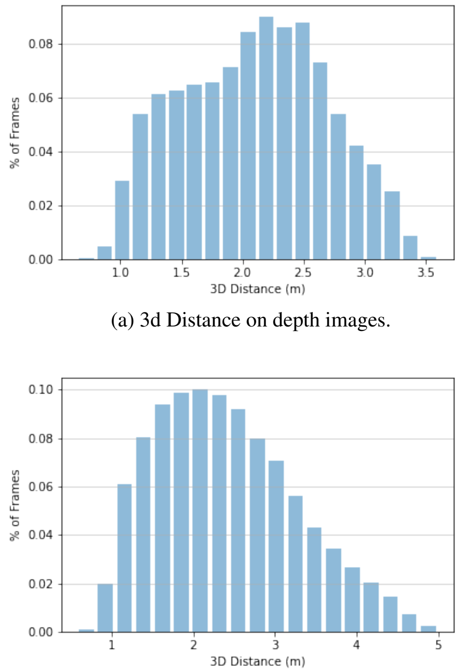
(b) Our synthetic RGB images.

Figure S11. Synthetic data samples from *EgoGen*.

Synthetic Dataset Statistics. The generated depth dataset consists of 105000 depth images with 47107 male and 57893 female images. The generated RGB dataset consists of 301073 depth images with 147862 male and 153211 female images. Both datasets cover a large range of indoor interaction distances ranging from $0.60m$ to $5.02m$. Fig. S12a shows the distribution of the interaction distance of the depth dataset and Fig. S12b shows the distribution of the RGB dataset.

Additionally, we consider two types of "invisibility" of the joints: frame-wise invisibility and joint-wise invisibility ratio. The frame-wise invisibility ratio calculates the percentage of joints that are not on the image plane among all body joints. The joint-wise invisibility ratio calculates the ratio of frames when the joint is out of the image plane among all frames.

An analysis of the invisibility distribution of the depth dataset and the RGB depth distribution can be seen



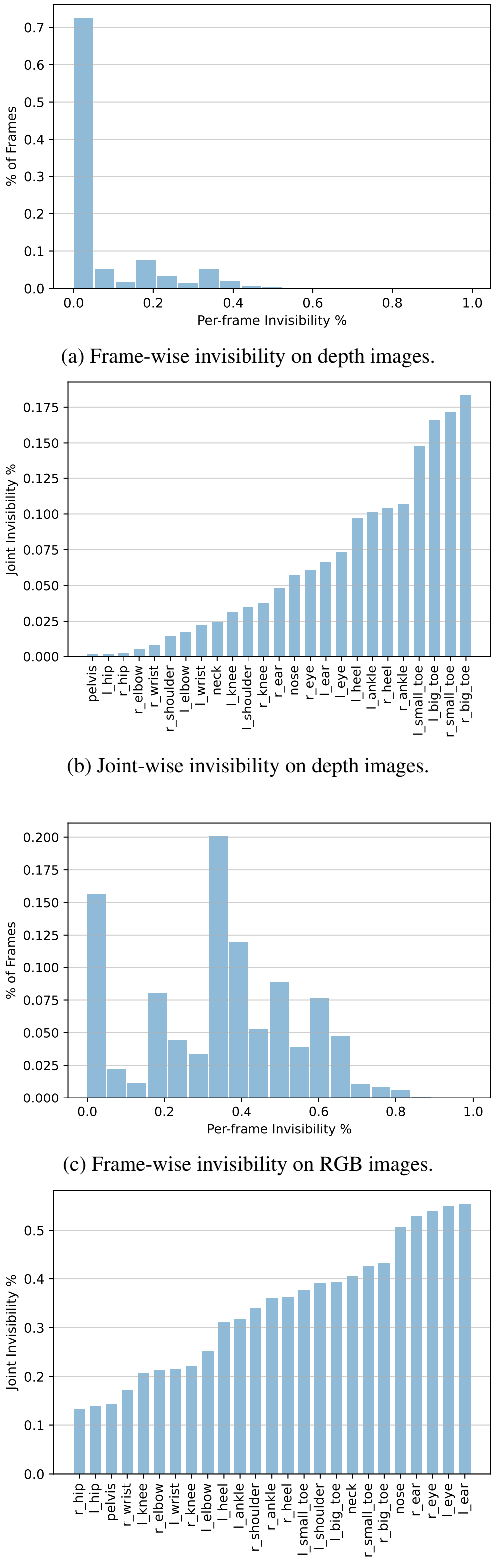
(a) 3d Distance on depth images.

(b) 3d Distance on RGB images.

Figure S12. Interaction Distance of synthetic samples from *EgoGen*.

in Fig. S13. Due to the different camera intrinsic of depth and RGB sensor, the invisibility distribution is different. From Fig. S13a, we can see that over 79% of the depth frame contains more than 90% joints. This means most depth images contain the full body. While from Fig. S13c we can see that the RGB dataset yields higher invisibility. The detailed invisibility of the joints is shown in Fig. S13d. It illustrates that even though RGB images have a higher invisibility, the most frequently missing joints are the upper or lower part of people (eyes and toes). In more than 85% of the images, the pelvis joint can be found.

Training Details. We use data augmentation on the training dataset besides adding the motion blur. These methods include using different kinds of image compression, brightness and contrast modification, noise addition, gamma, hue and saturation modification, conversion to grayscale, and downscaling techniques. During training, we set the batch size to 64 for the training on the depth dataset and 128 for the training on the RGB dataset. We use the AdamW Optimizer in the training process.



(a) Frame-wise invisibility on depth images.

(b) Joint-wise invisibility on depth images.

(c) Frame-wise invisibility on RGB images.

(d) Joint-wise invisibility on RGB images.

Figure S13. Invisibility Statistics.