Supplementary material – HMD^2: Environment-aware Motion Generation from Single Egocentric Head-Mounted Device

1. Technical details

Architecture and motion inference.

Our conditional motion diffusion model follows the Transformer-based architectures presented in EDGE [14] and DiT [11] with additional MLP encoder layers to gradually reduce the input dimension (which is bigger due to added CLIP and PC features) to the token latent space size. Our input consists of the motion input (as a translation, rotation, and linear and angular velocities) and PC and CLIP features, all concatenated together, representing one sequence token per frame. Following AvatarPoser [4], the model only predicts local joint rotations but not global translation. The global movement of the character is created during test time by "stitching" the predicted body motion to the ground-truth head motion, and the head motion can be directly obtained through real HMD motion obtained through SLAM, offset by a constant calibration matrix provided by the dataset. The motion output of the diffusion model is denoted as $\boldsymbol{x} \in \mathbb{R}^{T \times F}$, where $T = 240$ and $F = 23 \times 6$. The skeleton following Xsens definition has 23 ball-and-socket joints, and for each joint, the output rotation is represented as the first two columns of its local rotation matrix. Note that the definition of Xsens human skeleton is very similar to SMPL [6], with the main difference being the ordering of joints. The model is not conditioned on body size information, but during training, it is forced to see HMD input motions from different subjects covering highly diverse demographics. As such, the trained model is able to handle body size variation implicitly. However, providing size information as an explicit condition might further improve model performance and reduce visual artifacts such as floor penetration and foot sliding. To create the motion visualizations and compute position error metrics, we used ground truth body sizes (skeleton bone lengths) for each subject.

Image encoder. We use CLIP [13] variation ViT-L/14 for our experiments and compute embeddings from the timestamp-synchronized 30 FPS camera; to get the 60 FPS image feature condition, we duplicate every frame one more time. We also tried other image encoders and found that CLIP features perform best for our task – please refer to Sec. 3.2 for experimental results.

Pointcloud encoder. As mentioned in the main paper, the pointcloud encoder considers only SLAM points within the 2m x 2m x 2m volume centered around the head with 1m offset downward. The points are voxelized in a 10x10x10 voxel grid in the following way: for each voxel center, the closest point is selected and the distance is stored as a voxel value. All the distances are truncated at 10cm (so the value is clipped between 0 and 0.1). The voxel volume is rotated with the head orientation but only along the Z (gravity) axis.

The PC autoencoder consists of the encoder and decoder parts; the encoder consists of 4 convolution layers with 3×3 kernel, channel sizes 16,32,64,128 correspondingly, ReLU in between, with the average pooling in the end to produce one feature vector of size 128. Decoder is an inversion of that, consisting of 4 transposed convolution layers. It is trained on the volumes extracted using our train set's point clouds and head trajectories. We train with Adam [5] optimizer and learning rate of 10^{-3} for 10 epochs.

System runtime. Our current implementation assumes that point cloud encodings and CLIP features are precomputed or computed in parallel on a separate device. The performance will be affected if all computations need to happen on the same device. However, we observed that even in this situation, we could achieve a throughput of $\sim$ 61 FPS for our low-latency variant, therefore keeping up with real-time speed: CLIP embeddings take around 5 ms to compute per image (2.5 ms per motion frame since we are duplicating every frame), and point cloud encoder taking around 0.1 ms per motion frame. Note that the runtime performance is evaluated on a powerful GPU, which indicates a gap for our system to work in real-time on board of the HMD itself. Additionally, our current implementation assumes the access of all SLAM feature points in the around 15min window of the whole motion sequence. In a true real-time setting, this simplification would require a warm-up phase in the same environment of similar time length.

2. Dataset details

The Nymeria dataset we used [7] is captured from Project Aria glasses [12] paired with XSens [16] IMU motion cap-

	MPJPE ↓	Hand PE ↓	FID ↓	Diversity →	Physicality →	Floor Pen. ↓
Ground-truth	0	0	0	16.13	0.56	0
Ours w/ DINOv2	$8.72^{\pm 0.07}$	$17.24^{\pm 0.18}$	$2.45^{\pm 0.02}$	$15.38^{\pm 0.19}$	$\mathbf{0.91}^{\pm 0.00}$	$1.42^{\pm 0.07}$
Ours w/ VC-1	$8.54^{\pm 0.11}$	$\mathbf{16.64}^{\pm 0.22}$	$4.34^{\pm 0.06}$	$15.00^{\pm 0.42}$	$0.92^{\pm 0.01}$	$1.26^{\pm 0.10}$
Ours w/ CLIP (current)	$\mathbf{8.36}^{\pm 0.08}$	$\mathbf{16.64}^{\pm 0.21}$	$\mathbf{2.16}^{\pm 0.02}$	$\mathbf{15.74}^{\pm 0.29}$	$1.03^{\pm 0.01}$	$\mathbf{1.03}^{\pm 0.06}$

Table 1. Comparison between different image feature encoders. MPJPE, Hand PE and Floor penetration are in cm.

ture suit. The Project Aria glasses are set to record 30fps color video at 1408×1408 pixel resolution. Data captured from the glasses are further processed with its machine perception service (MPS) [2] to output the head transformation and point clouds. The XSens motion data is recorded onboard at 1KHz and processed with Analyse Pro as 240Hz full-body motion, downsampled to 60Hz for our input. The body motion from XSens is synchronized with Aria data to high accuracy using a custom timecode device. The body motion is further calibrated to the Aria head transformation to reduce spatial drift.

The full dataset contains 1200 motion sequences totaling 300 hours of daily activities of 264 participants across 50 locations, from which we used 1040 due to spatial synchronization problems in some sequences. Participants are recruited to cover uniform demographics along the axes of gender, age, height, and weight. The locations include 47 AirBnbs, where 31 are multi-floor houses. There is also a cafeteria with an outdoor patio, a multistory office building, and a campus with a parking lot and multiple biking/hiking trails.

The dataset covers a wide range of daily activities. The highest occurrences are cooking (13.5%), searching objects (11.0%), free-form activity improvise (10.4%), and playing games (10.1%), whereas the lowest occurrences include working at a desk (1.6%), locomotion (2.2%), activities in the office (2.3%), and creating a messy home (2.3%). Outdoor activities consist approximately 15% of the data. For additional details of the dataset, we refer readers to the Nymeria paper [7].

We split the dataset for training/validation/testing as 806/10/224 sequences, corresponding to 202/3/56 hours. **The testing split does not contain any locations or subjects that appear in the training set** to ensure no data leakage. We also strive to maintain a similar distribution of activities between the training set and the test set.

3. Additional experiments

3.1. Metrics – units of measure and symbols

All the metrics shown here and in the main paper, that have units of measure, namely positional errors (MPJPE, Hand PE, Low. PE, Up. PE) and Floor Penetration, are presented in cm. The down arrow ↓ means that lower value is always better for this metric, and the right arrow → means that the value closer to Ground-truth is better.

3.2. Comparison between different images feature encoders

To explain our choice of CLIP [13] feature as a feature encoder, we additionally trained two versions of our method with image features produced by DINOv2 [10] and VC-1 [8] feature encoders. For VC-1, we chose the best performing ViT-L model, with embedding size of 1024 and input size of 250 × 250 (cropped to 224 × 224 during preprocessing); for DINOv2, we chose second to largest model ViT-L/14, providing it with the input of the same size (padded to 252 × 252) and taking the class token of the output (size 1024), which corresponds to the global image description as it gathers the information from all the image patches. The comparison is presented in Tab. 1. We found that, while methods VC-1 and DINOv2 have close generation precision and a slight advantage in Physicality (correlated to foot sliding), the model with CLIP features shows the best results on most metrics, proving our choice of the image feature encoder.

3.3. Ablation study on h parameter values and diffusion steps

In Tab. 2, we show how the error metrics change depending on the latency (h) parameter. Because experiments with $h = 1$, 3, and 5 take a long time to process on our large test split, we performed this ablation on a 9% (20 out of 224 sequences) subset of test data. To keep the subset informative and maintain the diversity of activities, we picked one random sequence from each activity scenario. The results in the table demonstrate that the top performance in terms of MPJPE is achieved at $h = 180$, which we chose as our default value. While it is not the best on all the metrics, the difference is not as significant. Our low-latency method ($h = 10$) demonstrates some performance drop, but not as big compared to the next value $h = 5$, keeping a balance between the quality and the output lag.

We also measured metrics change w.r.t. the amount of diffusion steps we taking during inference. Tab. 3 shows that FID score increases with the amount of steps – visually, this corresponds to less jittery and more realistic motion. However, the precision of the motion, measured by MPJPE metric, peaks at 5 steps for full body and 3 steps for hands. Therefore, our choice of 20 steps is a balance

	MPJPE ↓	Hand PE ↓	FID ↓	Diversity →	Physicality →	Floor Pen. ↓
Ground-truth	0	0	0	16.95	0.04	0
$h = 230$	$9.53^{\pm 0.01}$	$16.15^{\pm 0.04}$	$\mathbf{13.44}^{\pm 0.01}$	$15.28^{\pm 0.01}$	$0.32^{\pm 0.00}$	$1.47^{\pm 0.02}$
$h = 220$	$9.49^{\pm 0.02}$	$16.07^{\pm 0.06}$	$13.61^{\pm 0.01}$	$15.30^{\pm 0.01}$	$0.25^{\pm 0.00}$	$1.46^{\pm 0.01}$
$h = 200$	$9.44^{\pm 0.01}$	$\mathbf{16.03}^{\pm 0.04}$	$13.74^{\pm 0.01}$	$15.32^{\pm 0.01}$	$0.23^{\pm 0.00}$	$1.45^{\pm 0.02}$
$h = 180$ (Ours)	$\mathbf{9.42}^{\pm 0.02}$	$16.05^{\pm 0.02}$	$13.76^{\pm 0.01}$	$15.43^{\pm 0.01}$	$\mathbf{0.22}^{\pm 0.00}$	$1.44^{\pm 0.01}$
$h = 120$	$9.43^{\pm 0.03}$	$16.05^{\pm 0.05}$	$14.02^{\pm 0.01}$	$15.22^{\pm 0.01}$	$0.26^{\pm 0.00}$	$1.43^{\pm 0.01}$
$h = 60$	$9.49^{\pm 0.06}$	$16.19^{\pm 0.03}$	$14.23^{\pm 0.01}$	$15.20^{\pm 0.01}$	$0.30^{\pm 0.00}$	$1.33^{\pm 0.03}$
$h = 30$	$9.61^{\pm 0.04}$	$16.42^{\pm 0.07}$	$14.39^{\pm 0.03}$	$15.57^{\pm 0.03}$	$0.40^{\pm 0.00}$	$1.26^{\pm 0.03}$
$h = 20$	$9.75^{\pm 0.10}$	$16.51^{\pm 0.08}$	$16.46^{\pm 0.04}$	$15.36^{\pm 0.04}$	$0.45^{\pm 0.00}$	$\mathbf{1.18}^{\pm 0.05}$
$h = 10$ (Ours low-lat.)	$10.19^{\pm 0.12}$	$17.13^{\pm 0.14}$	$17.00^{\pm 0.10}$	$15.66^{\pm 0.10}$	$0.73^{\pm 0.03}$	$1.41^{\pm 0.14}$
$h = 5$	$13.13^{\pm 0.46}$	$21.28^{\pm 0.45}$	$20.36^{\pm 0.33}$	$\mathbf{16.71}^{\pm 0.33}$	$0.94^{\pm 0.02}$	$1.84^{\pm 0.43}$
$h = 3$	$21.10^{\pm 1.08}$	$29.80^{\pm 1.15}$	$72.63^{\pm 0.82}$	$20.35^{\pm 0.82}$	$1.29^{\pm 0.12}$	$4.49^{\pm 0.51}$
$h = 1$	$28.96^{\pm 1.68}$	$38.13^{\pm 1.54}$	$129.94^{\pm 1.37}$	$22.74^{\pm 1.37}$	$2.22^{\pm 0.17}$	$3.75^{\pm 0.72}$

Table 2. Ablation study on the latency (h) parameter. Test is performed on a subset (9%) of the current test split. MPJPE, Hand PE and Floor penetration are in cm.

	MPJPE ↓	Hand PE ↓	FID ↓	Diversity →	Physicality →	Floor Pen. ↓
Ground-truth	0	0	0	16.95	0.04	0
2 steps	$9.54^{\pm 0.01}$	$15.94^{\pm 0.02}$	$15.04^{\pm 0.00}$	$15.45^{\pm 0.00}$	$0.50^{\pm 0.00}$	$1.87^{\pm 0.02}$
3 steps	$9.27^{\pm 0.01}$	$\mathbf{15.52}^{\pm 0.03}$	$15.28^{\pm 0.01}$	$14.85^{\pm 0.01}$	$0.32^{\pm 0.00}$	$1.64^{\pm 0.01}$
5 steps	$\mathbf{9.26}^{\pm 0.01}$	$15.57^{\pm 0.03}$	$14.94^{\pm 0.01}$	$14.97^{\pm 0.01}$	$0.25^{\pm 0.00}$	$1.54^{\pm 0.02}$
10 steps	$9.34^{\pm 0.02}$	$15.81^{\pm 0.03}$	$14.25^{\pm 0.01}$	$15.50^{\pm 0.01}$	$0.24^{\pm 0.00}$	$1.47^{\pm 0.01}$
20 steps (Ours)	$9.42^{\pm 0.02}$	$16.05^{\pm 0.02}$	$13.76^{\pm 0.01}$	$15.43^{\pm 0.01}$	$\mathbf{0.22}^{\pm 0.00}$	$1.44^{\pm 0.01}$
40 steps	$9.52^{\pm 0.02}$	$16.21^{\pm 0.02}$	$13.40^{\pm 0.01}$	$15.71^{\pm 0.01}$	$0.22^{\pm 0.00}$	$1.43^{\pm 0.02}$
80 steps	$9.60^{\pm 0.03}$	$16.38^{\pm 0.02}$	$\mathbf{13.11}^{\pm 0.01}$	$\mathbf{15.77}^{\pm 0.01}$	$0.23^{\pm 0.00}$	$\mathbf{1.41}^{\pm 0.01}$

Table 3. Ablation study on the amount of steps in reverse diffusion process. Test is performed on a subset (10%) of the current test split.

between motion precision and realism.

3.4. More results on error distribution

In Tab. 4, we present additional metrics, splitting per-joint average error into average error across upper (Up. PE) and lower (Low. PE) body regions. The upper region is defined as all the joints that are higher than the pelvis for the subject standing in a T-pose, namely the spine, shoulders, arms, hands, neck, and head. The lower body region is defined as the rest of the joints, excluding the root joint (hips, legs, feet). From these metrics, we can directly observe the effect of adding pointcloud and image encoders to our data. When the PC encoder is added, the lower body error is reduced significantly, and the upper body gets slightly worse (most likely due to noisy points near the upper body region). This suggests that pointcloud helps to disambiguate the lower body by providing landscape information (floor level, nearby objects, etc.). On the other hand, when CLIP image encoding is added, we notice a major reduction in the upper body error, suggesting that image features help the method better understand interactions and localize hands. At the same time, lower body error also decreases - most likely, the error is reduced when parts of the lower body are visible on camera. HMD^2, denoted as "Ours, w/ PC, w/ CLIP" in the table, combines both strengths of the methods above and achieves the lowest mean per-joint error.

3.5. More top 5% error results and metric computation algorithm

The error reduction effect discussed above can also be noted in Tab. 5, showing the top 5% error for upper and lower body error metrics. Here, we want to clarify our top error selection strategy. As shown in Sec. 3.8 and Fig. 2, the average error on the sequence greatly depends on the activity performed in that sequence. If we were to sort all the per-frame joint errors and select the top 5% (95% percentile) among them, we would only select the frames from several worse-performing sequences. To avoid such behavior, we compute the 95% error percentile within each sequence separately and average those results across all sequences.

3.6. Effects of the input variation on the generation performance

In Tab. 4, we also present a study of another, much more challenging baseline – a 3-point input method. For that, we chose the original implementation AvatarPoser [4], which takes not only the head position and orientation as an input but also the positions and orientations of the hands. With more input information, this baseline achieves better performance on average. However, we highlight that even with

	MPJPE ↓	Hand PE ↓	Low. PE ↓	Up. PE ↓	Floor Pen. ↓
EgoEgo	$16.61^{\pm 1.49}$	$34.64^{\pm 1.64}$	$26.58^{\pm 3.57}$	$11.31^{\pm 0.54}$	$2.43^{\pm 1.54}$
AvatarPoser (Head)	10.64	21.51	17.70	6.90	2.94
AvatarPoser (Head & Hands)	**7.74**	**6.29**	16.10	**3.11**	4.63
Ours, w/o PC, w/o CLIP	$9.28^{\pm 0.23}$	$19.47^{\pm 0.36}$	$15.04^{\pm 0.53}$	$6.21^{\pm 0.11}$	$3.29^{\pm 0.31}$
Ours, w/ PC, w/o CLIP	$8.97^{\pm 0.10}$	$20.38^{\pm 0.28}$	$\underline{13.59}^{\pm 0.21}$	$6.53^{\pm 0.07}$	$\mathbf{0.99}^{\pm 0.07}$
Ours, w/o PC, w/ CLIP	$8.57^{\pm 0.11}$	$\underline{16.32}^{\pm 0.22}$	$14.02^{\pm 0.25}$	$\underline{5.64}^{\pm 0.06}$	$2.15^{\pm 0.15}$
Ours, w/ PC, w/ CLIP	$\underline{8.36}^{\pm 0.08}$	$16.64^{\pm 0.21}$	$\mathbf{13.23}^{\pm 0.16}$	$5.75^{\pm 0.06}$	$\underline{1.03}^{\pm 0.06}$

Table 4. Lower and upper body error depending on the input variations. We are beating a 3-point input baseline on a lower body error and achieve close performance on average. All the metrics are in cm.

	MPJPE ↓	Hand PE ↓	Low. PE ↓	Up. PE ↓	Floor Pen. ↓
EgoEgo	$30.91^{\pm 4.82}$	$60.81^{\pm 2.98}$	$58.63^{\pm 12.17}$	$19.26^{\pm 1.16}$	$10.33^{\pm 5.90}$
AvatarPoser (Head)	22.09	43.19	44.18	13.01	18.96
AvatarPoser (Head & Hands)	16.48	**11.23**	37.91	**5.63**	18.15
Ours, w/o PC, w/o CLIP	$18.31^{\pm 0.89}$	$40.15^{\pm 1.17}$	$34.35^{\pm 2.20}$	$11.75^{\pm 0.37}$	$12.91^{\pm 1.75}$
Ours, w/ PC, w/o CLIP	$16.65^{\pm 0.44}$	$41.68^{\pm 1.05}$	$28.72^{\pm 1.02}$	$12.29^{\pm 0.30}$	$\mathbf{3.97}^{\pm 0.32}$
Ours, w/o PC, w/ CLIP	$16.30^{\pm 0.55}$	$34.25^{\pm 0.90}$	$29.98^{\pm 1.35}$	$10.58^{\pm 0.26}$	$8.28^{\pm 0.78}$
Ours, w/ PC, w/ CLIP	$\mathbf{15.49}^{\pm 0.38}$	$34.86^{\pm 0.92}$	$\mathbf{27.35}^{\pm 0.81}$	$10.80^{\pm 0.26}$	$4.22^{\pm 0.28}$

Table 5. Lower and upper body error study on top 5% errors (mean of 95% percentiles across all sequences). Here, we are beating 3-point error baseline on mean per-joint positional error. All the metrics are in cm.

additional motion input, it is worse than Ours at generating lower body motion, as Lower body PE is higher. It is important to note that HMD^2 achieves *best performance* on the most challenging frames of the sequences even when compared to a 3-point input baseline, as shown in the top 5% error study in Tab. 5.

3.7. Diversity of results given the same input

Fig. 1 shows 4 random motion samples given the same input for two sequences (1st sequence indoor, 2nd sequence outdoor). A few observations worth highlighting: 1. EgoEgo is also capable of generating diverse predictions, sometimes more diverse than Ours; 2. However, EgoEgo generations tend to be of lower quality - possibly due to model architecture not being as scalable to a massive dataset as Ours and autoregressive long sequence inference not working as well; 3. Moreover, EgoEgo samples often do not satisfy floor height constraints (1st seq. 3rd frame; 2nd seq. 1st frame), and cannot utilize image observation when certain body parts are visible (1st seq., see the right arm in 1st frame and left arm in 2nd frame); 4. Samples from Our method are "conditionally diverse". This is unseen in previous papers. E.g. when the egocentric camera sees only one arm, Ours will generate samples with this arm doing the motion seen (not perfectly accurate partially due to CLIP) and generate motions for the unseen arm and legs with diversity (see arms in 1st&2nd frames on the 1st sequence, see legs in all frames on the second sequence).

3.8. Variation of an error depending on the activity

Our test dataset consists of diverse activities, and each sequence is dedicated to a certain type of activity according to the assigned scenario. In total, there are 20 scenarios, with indoor and outdoor activities featuring walking, sitting, laying, exercising, interacting with household objects, playing sports games, and more. If we group the sequences and measure the MPJPE in each group (Fig. 2), we can observe that the error is not distributed evenly – while for most scenarios the error does not exceed 8cm, there is a chunk of challenging scenarios that have an error almost twice as high. To understand the reasons behind this, we selected and studied different metrics for the scenario, including the best, the worst, and median MPJPE. Results are presented in tables 6, 7, 8.

The best-performing scenario (Tab. 6) consists of multi-terrain outdoor walking (hiking up and downhill) but does not feature any interactions. Small lower body error demonstrates that multi-level motion is, in general, not a significant challenge for our method – in contrast to AvatarPoser, whose lower body error is higher on this scenario than on the mostly flat scenario from Tab. 7.

The scenario with the median method performance (Tab. 7) consists of mostly flat-ground indoor multi-room interactions with the objects in the house (grabbing clothes, throwing pillows, opening doors). The subject often stays in the standing position, occasionally bending to reach some objects. As interactions with the objects appear more often here, we notice higher hand positional errors for our method. This can be explained by the inability of the CLIP-encoded image features to localize the hands precisely dur-

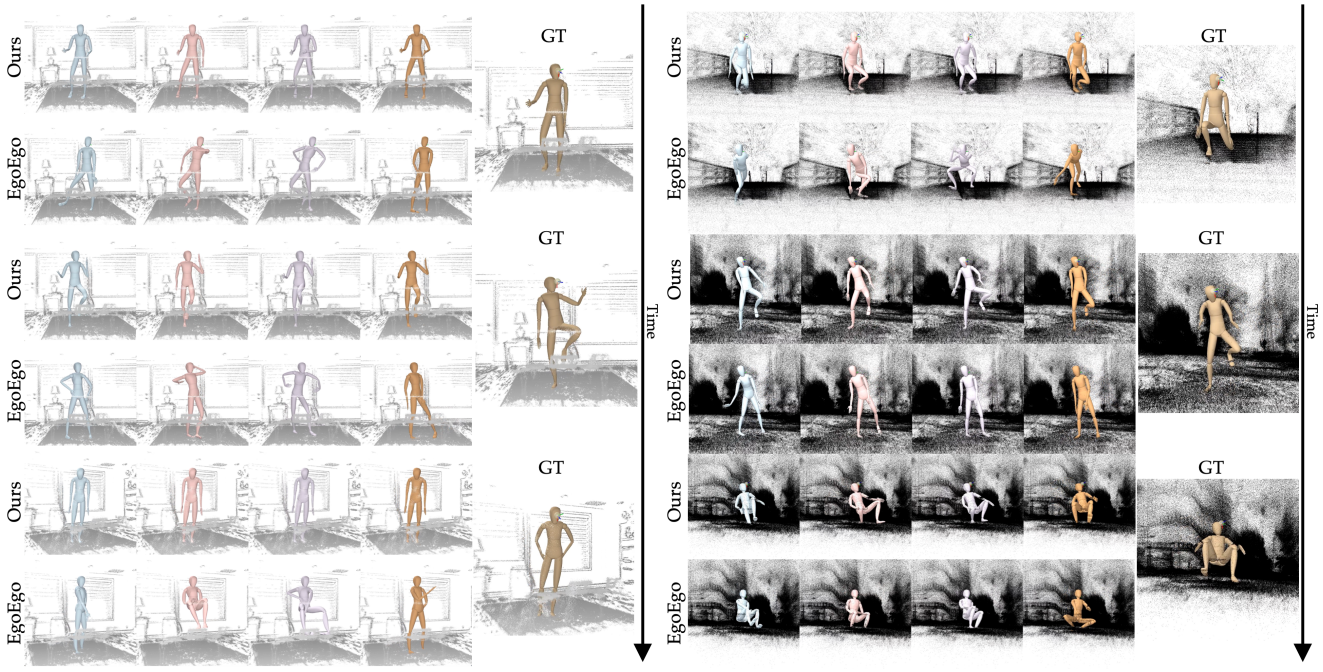

Figure 1. Range of possible results given the same input for HMD^2 and EgoEgo. Colors denote different runs, sequence frame time is increasing from top to bottom.

ing the interactions. Occasional bending can also be misinterpreted for a different motion sometimes, which explains higher floor penetration error.

The worst performing scenario (Tab. 8) consists mainly of yoga and body stretching motions, which proved to be the most challenging for all the methods. While the upper body error is higher than usual, the error is primarily increasing due to very high lower body error. This is caused by a high position uncertainty: most of the time, lower body parts are not observed by the camera, and the floor estimation from a SLAM point cloud might be noisy. Future work on improving the performance in such scenarios might benefit from: enhancing the reconstructed SLAM pointcloud quality to provide reliable terrain information; including more of these challenging motions in the dataset; using cameras with a higher field of view, like fisheye cameras, to increase the body parts visibility.

4. Limitations, future work and ethical implications

As mentioned in the main paper, our system is limited by the data encoded in the features - the limbs localization precision is less than desired sometimes. Features that contain more precise positional information than CLIP may improve performance: one potential direction for future work is to additionally condition the method on the results of the hand-tracking algorithm. However, even without explicit positional information, CLIP-encoded images improve upper body tracking. The effect on the lower body is less apparent. This, of course, can be explained by the fact that the lower body is much less visible from the camera, especially since we use a camera with the standard FOV looking outwards. Additional information from the downward-looking wide-angle cameras can improve the performance, as shown in *e.g.* [15].

Even with the point cloud context provided, our method can sometimes produce visual artifacts such as floor penetration (as measured by the Floor. Pen. metric in tables). This means that the network occasionally misses or ignores the PC context. It can happen due to the noise presented in the pointcloud data and large distances between the points, especially in untextured regions like floors or walls. One way to improve the performance here is to use the more advanced point cloud/mesh reconstruction solution, potentially using the depth sensor (*e.g.* Depth-based fusion [3]). Another way is to use a more advanced point cloud encoder; such an encoder can be trained on a different task, *e.g.*, point-to-mesh [1]. Note that we only capture static point clouds and do not yet handle dynamic environment changes such as opening doors, moving a chair, etc. – this is a great future work direction.

Our method is not aware of the shape of the body and, therefore, does not correct self-interpenetration of body parts, which can happen sometimes. That can be fixed during the postprocessing stage with self-contact optimization

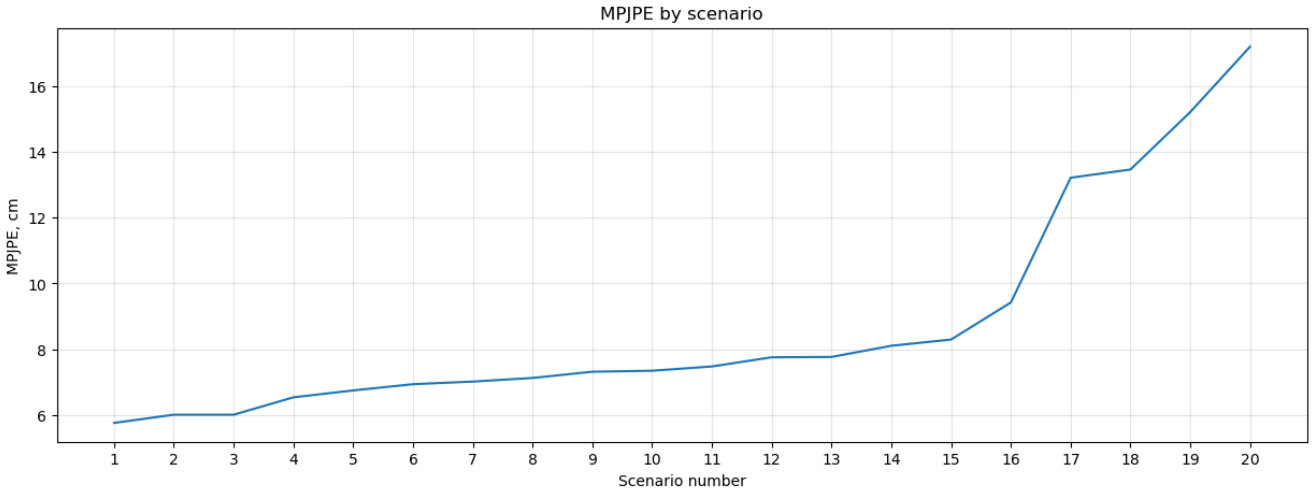

Figure 2. MPJPE depending on the action scenario (sorted in increasing order).

	MPJPE ↓	Hand PE ↓	Low. PE ↓	Up. PE ↓	Floor Pen. ↓
EgoEgo	$12.06^{\pm 0.33}$	$31.31^{\pm 1.13}$	$17.24^{\pm 0.75}$	$9.40^{\pm 0.30}$	$\mathbf{0.01}^{\pm 0.00}$
AvatarPoser (Head)	7.39	14.81	12.58	4.64	0.11
Ours ($h = 180$)	$\mathbf{5.75}^{\pm 0.03}$	$\mathbf{11.98}^{\pm 0.13}$	$\mathbf{8.84}^{\pm 0.07}$	$\mathbf{4.06}^{\pm 0.03}$	$0.02^{\pm 0.00}$
Ours ($h = 10$)	$6.19^{\pm 0.04}$	$12.16^{\pm 0.07}$	$9.97^{\pm 0.10}$	$4.13^{\pm 0.01}$	$0.02^{\pm 0.00}$

Table 6. Results for the scenario with the best HMD^2 performance. Scenario is consisting of the multi-terrain outdoor walking (hiking up- and downhill), mostly sightseeing. All the metrics are in cm.

	MPJPE ↓	Hand PE ↓	Low. PE ↓	Up. PE ↓	Floor Pen. ↓
EgoEgo	$12.29^{\pm 0.25}$	$32.32^{\pm 0.50}$	$16.40^{\pm 0.64}$	$10.16^{\pm 0.16}$	$\mathbf{0.31}^{\pm 0.14}$
AvatarPoser (Head)	8.39	20.94	11.44	6.78	0.80
Ours ($h = 180$)	$\mathbf{6.53}^{\pm 0.06}$	$\mathbf{15.66}^{\pm 0.17}$	$\mathbf{8.86}^{\pm 0.10}$	$\mathbf{5.29}^{\pm 0.05}$	$0.42^{\pm 0.05}$
Ours ($h = 10$)	$7.32^{\pm 0.05}$	$17.30^{\pm 0.17}$	$10.05^{\pm 0.10}$	$5.87^{\pm 0.04}$	$0.45^{\pm 0.02}$

Table 7. Results for the scenario with the median across all 20 scenarios HMD^2 performance. Scenario is consisting of flat-ground indoor multi-room interactions with the objects in the house (grabbing clothes, throwing pillows, opening doors), mostly upright standing with occasional bending (to reach for the next object). All the metrics are in cm.

	MPJPE ↓	Hand PE ↓	Low. PE ↓	Up. PE ↓	Floor Pen. ↓
EgoEgo	$28.67^{\pm 1.97}$	$42.85^{\pm 1.46}$	$52.11^{\pm 4.52}$	$15.75^{\pm 0.64}$	$12.76^{\pm 3.55}$
AvatarPoser (Head)	23.30	31.11	45.01	11.32	21.79
Ours ($h = 180$)	$\mathbf{17.21}^{\pm 0.20}$	$\mathbf{24.39}^{\pm 0.36}$	$\mathbf{31.27}^{\pm 0.50}$	$\mathbf{9.45}^{\pm 0.13}$	$\mathbf{3.32}^{\pm 0.24}$
Ours ($h = 10$)	$18.74^{\pm 0.65}$	$26.28^{\pm 0.50}$	$33.37^{\pm 1.41}$	$10.55^{\pm 0.27}$	$5.01^{\pm 0.39}$

Table 8. Results for the scenario with the worst HMD^2 performance. Scenario is consisting of challenging body stretching and yoga motions, mostly on done the floor, recorded indoors. All the metrics are in cm.

methods like TUCH [9]. Another problem that affects the visual quality is motion jitter, which can be observed mostly during online low-latency inference – this can be smoothed during motion postprocessing. However, we decided not to apply the smoothing to show the raw performance of the method.

As our method uses the head-mounted first-person view camera, there are privacy concerns related to that; one of the major ones is the leaking of the raw video frames. Our current effort to mitigate this involves using the built-in functionality of Aria glasses [2] to blur the faces during the data capture. We can improve the privacy aspect even more by moving CLIP and PC encoding computation on the capturing device itself. As our method uses only the encoded image and pointcloud features instead of raw data, on-device precomputed features would work just as well. We also believe that after some optimization efforts, there is a potential to perform the full inference pipeline on the mobile device itself, therefore eliminating the potential data leak problem completely.

References

[1] Chibane, J., Pons-Moll, G., et al.: Neural unsigned distance fields for implicit function learning. Advances in Neural Information Processing Systems **33**, 21638–21652 (2020) 5

[2] Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., Peng, C., Sweeney, C., Wilson, C., Barnes, D., DeTone, D., Caruso, D., Valleroy, D., Ginjupalli, D., Frost, D., Miller, E., Mueggler, E., Oleinik, E., Zhang, F., Somasundaram, G., Solaira, G., Lanaras, H., Howard-Jenkins, H., Tang, H., Kim, H.J., Rivera, J., Luo, J., Dong, J., Straub, J., Bailey, K., Eckenhoff, K., Ma, L., Pesqueira, L., Schwesinger, M., Monge, M., Yang, N., Charron, N., Raina, N., Parkhi, O., Borschowa, P., Moulon, P., Gupta, P., Mur-Artal, R., Pennington, R., Kulkarni, S., Miglani, S., Gondi, S., Solanki, S., Diener, S., Cheng, S., Green, S., Saarinen, S., Patra, S., Mourikis, T., Whelan, T., Singh, T., Balntas, V., Baiyya, V., Dreewes, W., Pan, X., Lou, Y., Zhao, Y., Mansour, Y., Zou, Y., Lv, Z., Wang, Z., Yan, M., Ren, C., Nardi, R.D., Newcombe, R.: Project Aria: A new tool for egocentric multi-modal AI research (2023) 2, 6

[3] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: ACM symposium on User interface software and technology. pp. 559–568. ACM (2011) 5

[4] Jiang, J., Streli, P., Qiu, H., Fender, A., Laich, L., Snape, P., Holz, C.: Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In: European Conference on Computer Vision. pp. 443–460. Springer (2022) 1, 3

[5] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015) 1

[6] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Transactions on Graphics (2015) 1

[7] Ma, L., Ye, Y., Hong, F., Guzov, V., Jiang, Y., Postyeni, R., Pesqueira, L., Gamino, A., Baiyya, V., Kim, H.J., Bailey, K., Fosas, D.S., Liu, C.K., Liu, Z., Engel, J., De Nardi, R., Newcombe, R.: Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In: European Conference on Computer Vision (ECCV) (2024) 1, 2

[8] Majumdar, A., Yadav, K., Arnaud, S., Ma, J., Chen, C., Silwal, S., Jain, A., Berges, V.P., Wu, T., Vakil, J., et al.: Where are we in the search for an artificial visual cortex for embodied intelligence? Advances in Neural Information Processing Systems **36** (2024) 2

[9] Muller, L., Osman, A.A., Tang, S., Huang, C.H.P., Black, M.J.: On self-contact and human pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9990–9999 (2021) 6

[10] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 2

[11] Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023) 1

[12] Project Aria (accessed January 7, 2025), https://www.projectaria.com/ 1

[13] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021) 1, 2

[14] Tseng, J., Castellon, R., Liu, K.: Edge: Editable dance generation from music. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 448–458 (2023) 1

[15] Wang, J., Luvizon, D., Xu, W., Liu, L., Sarkar, K., Theobalt, C.: Scene-aware egocentric 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13031–13040 (2023) 5

[16] Xsens MVN Link (accessed January 7, 2025), https://www.movella.com/products/motion-capture/xsens-mvn-link 1