EgoBody3M: Egocentric Body Tracking on a VR Headset using a Diverse Dataset

Amy Zhao*, Chengcheng Tang*, Lezi Wang, Yijing Li, Mihika Dave, Lingling Tao, Christopher D. Twigg, and Robert Y. Wang

Meta Platforms Inc.
{xamyzhao,chengcheng.tang,wanglezi,yijingli,mihikadave,
linglingtao,cdtwigg,rywang}@meta.com

We present the first controller-less egocentric body tracking solution that runs on an actual VR device, using the same four monochrome cameras that are used for SLAM tracking (left). We propose a novel egocentric tracking architecture that models the temporal history of body motion using multi-view latent features, and accurately infers full-body poses even when limbs are out-of-view (right).

Abstract. Accurate tracking of a user's body pose while wearing a virtual reality (VR), augmented reality (AR) or mixed reality (MR) headset is a prerequisite for authentic self-expression, natural social presence, and intuitive user interfaces. Existing body tracking approaches on VR/AR devices are either under-constrained, e.g., attempting to infer full body pose from only headset and controller pose, or require impractical hardware setups that place cameras far from a user's face to improve body visibility. In this paper, we present the first controller-less egocentric body tracking solution that runs on an actual VR device using the same cameras that are used for SLAM tracking. We propose a novel egocentric tracking architecture that models the temporal history of body motion using multi-view latent features. Furthermore, we release the first large-scale real-image dataset for egocentric body tracking, EgoBody3M, with a realistic VR headset configuration and diverse subjects and motions. Benchmarks on the dataset shows that our approach outperforms other state-of-the-art methods in both accuracy and smoothness of the resulting motion. We perform ablation studies on our model choices and demonstrate the method running in realtime on a VR headset. Our dataset with more than 30 hours of recordings and 3 million frames will be made publicly available.

Keywords: Virtual reality · Egocentric body tracking · Large-scale dataset

* Equal contribution

1 Introduction

Tracking a user's body pose with an AR/VR headset has applications in gaming, sports, avatar-based telepresence, and natural user interfaces. The current dominant method for body tracking on commercial VR devices relies on inferring body pose from only headset and controller pose, leading to interactions that feel less expressive or unnatural. While tracking more points of the body is possible by wearing additional sensors, the added expense and user friction limit adoption. Camera-based egocentric body tracking offers additional accuracy without additional friction. However, academic research in this area often employs setups with cameras positioned far from the user's face. While this improves camera visibility, it leads to a bulky mechanical design and is not ergonomic.

A practical body tracking solution should be designed to adapt to the product form factor and have cameras flush-mounted on the device's surface. Ideally, it should utilize the same cameras required for inside-out SLAM tracking of the headset's rigid pose to avoid additional expense and power. While surface-mounted cameras are preferred in product design, they present several limitations. Headset thickness is a critical design parameter, as bringing the headset closer to the user's eyes reduces torque applied to the neck [19] and hence improves ergonomics. Modern VR or Mixed Reality (MR) headsets such as the Meta Quest 3 [2], Apple Vision Pro [1], and Pico 4 [5] use pancake lenses to reduce device thickness. For these headset designs, there is no single or stereo pair of fisheye surface-mounted cameras that can cover the entire region spanned by body motion. Even the best camera setup will typically have blind spots due to the power and weight cost of adding extra cameras. An egocentric body tracking algorithm must therefore handle the following requirements: (1) support multiple cameras to cover the entire region of interest; (2) reason about body parts under occlusion or out-of-view; and (3) leverage temporal information to provide plausible and smooth motion. An algorithm that addresses these requirements is not yet available.

In addition, the difficulty of capturing data of humans wearing headsets has resulted in a paucity of real datasets for egocentric body tracking. The EgoCap dataset [33] is still the largest such dataset, with only 60,000 images and an impractical setup that places cameras far away from the headset. To unblock research in this area, multiple teams have developed synthetic datasets [6, 42]. However, human bodies are difficult to render accurately due to subtle details of anatomy, shape, deformation and clothing fit, meaning that there is still a significant domain gap between synthetic and real datasets. In addition, synthetic motion datasets such as Mixamo focus on rare, highly athletic motions instead of more subtle motions typically found in social contexts [29]. Overall, there is a lack of real datasets with practical VR camera placements, capturing a diversity of people and environments, and including ground truth 3D poses to unlock research on egocentric body tracking.

To address the above gaps in practical models and datasets, this work makes the following contributions:

1. We propose the first egocentric body tracking solution that runs on actual VR headset cameras.
2. We describe a novel egocentric tracking architecture that models temporal history of the body motion using multi-view latent features. Benchmarking on the UnrealEgo dataset shows that our model outperforms other state-of-the-art methods.
3. We release the first large-scale real dataset, EgoBody3M, for egocentric body tracking based on a realistic VR camera configuration and diverse groups of environments, subjects and motions. The dataset contains 2688 sequences from 120 subjects.

2 Related work

2.1 Outside-in body tracking

Human pose tracking from outside-in cameras has been extensively studied. A series of neural network-based 2D keypoint estimation methods (as described in [28], [48], [10], [38], and [46]) have resulted in reliable 2D keypoint predictions that are now being used as the basis for 3D pose estimation. For 3D pose estimation, one approach involves directly regressing 3D keypoint positions from monocular images ([25,30,39,40]). Multi-view image based methods extend this further to regress 3D pose using 3D-aware representations [18,32]. Recently, "2D to 3D lifting" became a popular approach. In [27], it was demonstrated that a simple Multi-Layer Perceptron (MLP) [16] architecture for one-frame keypoint-based lifting can outperform direct regression methods. [31] demonstrated that temporal model based lifting using sequence of 2D keypoints significantly improves lifting accuracy. More recent work explored advanced temporal models such as Graph Convolutional Networks based models ([17, 45]), transformer based models (MixSTE [50] and MotionBert [53]) to further boost the temporal lifting accuracy. While keypoint-based lifting works well for outside-in body tracking, additional information about the image is needed to model the uncertainty of joints due to more frequent occlusions in egocentric views (e.g., cues about the limb could help infer wrist position) [42]. Our model addresses this issue by using latent features extracted from multi-view images, making it more similar to the VIBE framework [23].

2.2 Datasets for egocentric body tracking

EgoCap [33] was one of the first to propose computer vision-based egocentric tracking by attaching two cameras on rigid rods to a bicycle helmet. While locating the cameras far from the face maximizes the view of the body, it compromises ergonomics and form factor, making such a system impractical for commercial AR/VR devices. We follow EgoCap's approach of using outside-in cameras to estimate ground truth body joint locations and tracking the egocentric cameras' pose and orientation. EgoGlass [52], on the other hand, proposes a more realistic

camera configuration where the two cameras are located on the temples of a set of glasses. Unfortunately, their data is not available to the public.

The limited availability of egocentric data and the challenges in collecting it have led to the widespread use of synthetic data. Mo^2Cap^2 [49] released a synthetic dataset with a single camera located 8cm from the head. xR-EgoPose [41, 42] released a large dataset that positioned a single fisheye camera at the bottom of a VR headset. However, recent advancements in optics such as the pancake lenses used in Meta's Quest Pro [3], Meta's Quest 3 [2] and Apple's Vision Pro [1] reduce headset thickness, making this setup less practical. UnrealEgo [6] employs a significantly more realistic camera configuration (virtually identical to EgoGlass) and adds synthetic humans in the background to create a more realistic scenario for AR body tracking.

Wang and colleagues [43] addressed the lack of egocentric data by using a weak-supervision approach, where a single outside-in camera was paired with an egocentric camera. To further improve egocentric body by leveraging scene data, Wang [44] generated synthetic datasets EgoGTA and EgoPW-Scene, containing labels for egocentric poses and scene geometry. However, these datasets have an impractical camera location and the use of weak ground truth makes it difficult to build upon.

EgoHumans [22] is a new dataset captured with a combination of egocentric and outside-in views, allowing for accurate 3D ground truth in-the-wild. However, the camera's field-of-view captures little of the wearer's own body, so the main task presented in the dataset involves predicting the motions of other people. Ego4D is a large-scale collection of naturalistic human motion, but it does not have 3D human pose ground truth. Tab. 2 provides an overview of the various datasets utilized for egocentric body tracking.

2.3 Egocentric tracking using cameras

Existing methods in egocentric tracking mostly focused on single-frame based prediction. [33] uses a ConvNet-based 2D body-keypoint detector and a 3D body model based optimization algorithm with stereo cameras. [49] uses CNNs to predict body joint distance to cameras in addition to 2D keypoint position and back projects to get 3D position. More recent methods explore a form of 2D feature to 3D pose lifting approach [6, 42, 52]. [42] adopts a two-step approach with a dual-branch auto-encoder applied to predicted 2D heatmaps to capture uncertainty of joints in the egocentric setting. [21] proposes to use motion features and shape features extracted at frame level to predict the 3D pose and an additional 3D CNN to refine the pose with a volumetric representation. [52] aims to tackle a more practical setup with multiple partial-body view images by extending this architecture with 2 separate CNNs for two views and modeling body part information. [6] shows that using one CNN with stereo inputs for 2D prediction instead of two separate CNNs further improve the final 3D pose estimation result. [51] estimates 3D joint locations from a fisheye-camera image with automatic camera calibration. [21] predicts 3D body poses from a single egocentric fisheye camera, where the camera wearer can be seen in the

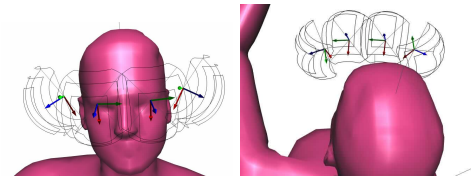


Fig. 2: Headset cameras are marked with coordinate frames, black lines trace the frustums. (left) from front (right) from back.

peripheral view. There is less work in leveraging temporal models for egocentric tracking. [9] uses a keypoint sequence-based 3D RNN to predict body pose, but keypoint representations lacks the ability to capture additional cues when joints are under occlusion or out-of-view (oov).

2.4 Egocentric tracking using sparse signals

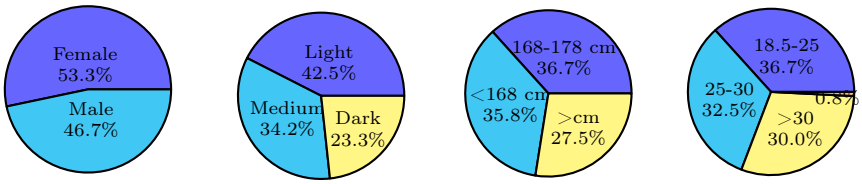
Many VR headsets use hand-held controllers to track the hand pose. These controllers typically combine on-board IMUs with IR LEDs for robust, wide field-of-view tracking, making them a highly useful signal for recovering body pose. There is a robust literature on inferring body pose from these tracked controllers. Parger and colleagues [29] proposed a heuristic approach for reconstructing body pose from headset and tracked controllers, known as "three-point tracking". QuestSim [47] combines reinforcement learning and physics simulation to produce plausible motions. The limitation of three point tracking approaches is that they assume the presence of tracked controllers (or other additional hardware), which users may find encumbering. Dittadi and colleagues [12] predicted SMPL parameters from hand, head and gaze tracking. Li and colleagues [24] use diffusion models to estimate body pose from SLAM-based headset tracking alone. In general, methods that work with sparse signals operate in an under-constrained setting, generating plausible but not necessarily accurate or expressive motions.

3 Dataset

3.1 Capture Setup

We collect data using a realistic VR headset equipped with 4 synchronized monochrome global-shutter cameras as shown in Fig. 2. Monochrome cameras are preferred over RGB cameras due to their superior low-light performance, as the Bayer grid used in RGB cameras blocks some of the light from reaching the sensor [20]. The two cameras on the sides of the headset are 640×480 VGA cameras while the two cameras in front have the higher resolution of 1280×1024. We track the subject's pose using 8 outside-in Azure Kinect [4] time-of-flight RGB-D cameras, while the 6-DOF headset pose is tracked using an Optitrack motion capture system. All cameras in the system are synchronized using a global time-code to ensure perfect frame alignment.

Fig. 3: Some sample environments captured during the data collection. The mocked-up environments spanned a gamut of indoor environments, including office environments (16%), bedrooms (6%), living rooms (20%), dining rooms (37%), and studio apartments (22%). See supplementary materials for more information.



(a) Gender **(b)** Skin Tone **(c)** Height **(d)** BMI

Fig. 4: Demographic statistics for subjects in our dataset (We combine the data of test / train / validation splits to report the demographic distribution; see supplementary materials for distributions in each split). Gender was self-reported, BMI and height were measured, and skin tone was measured as described in the text.

3.2 Diversity

Data diversity is important to ensure the model generalizes to different environments, subjects and motions. Our data collection process emphasizes this diversity, including a wide range of environments, subjects, and protocols.

Environments. We collect data in an indoor environment since our primary use-case is VR. Unlike previous work [33], we vary the indoor environment to ensure that our models can generalize to a wide range of home and office environments. During data collection, the environment is changed daily by rotating furniture, carpets, wall hangings, etc. See Fig. 3 for example environments.

Subjects. We also balance across gender, skin tone, height, and BMI in our data collection (see Fig. 4). Participants self-reported their age and gender. The skin tone was measured using a Pantone RM200 colorimeter based on guidelines in the Pantone SkinTone Guide [26] and divided into three categories: light, medium, and dark. Because clothing strongly impacts the wearer's egocentric appearance, we encourage participants to wear a variety of clothing to achieve variations in clothing fit, type and pattern. See Fig. 5 for examples of outfits.

Protocols. While [6] provides the largest motion protocol variation among existing egocentric body tracking datasets, the motions are mostly athletic body movements. In this dataset, we aim to balance AR/VR use cases across social and fitness use cases through 30 motion protocols. We show a subset in Tab. 1, full details on the protocols can be found in the Supplemental Material.

Fig. 5: Sample outfits captured during data collection. Note that each participant wore two different outfits. See supplementary materials for more information.

Table 1: Subset of motion protocols by use case. We collect motions covering social, productivity (virtual meetings), sports/fitness and gaming.

Category	Motion	Train sequences	Test sequences
Social	Social gesture greet shake hands	55	16
	Arm rest at sides rotate head	73	19
	Hands on hips conversation	58	13
	Stand cross arms	78	20
Productivity	Meeting discussion taking notes	64	16
	Whiteboard presentation	62	16
	Pointing whiteboard	64	16
Sports/fitness	Playing tennis	44	11
	Fitness punching	49	11
	Boxing and kickboxing	57	15
Gaming	Fighting with legs	69	17
	Elbow strike	46	17

3.3 Ground truth

The eight synchronized outside-in Azure Kinects (Fig. 6) provide RGB and depth images (Fig. 7). We compute the ground truth poses using a combination of RGB keypoint estimation and frame-to-frame deformable ICP. Manual quality annotation is used to filter out incorrect ground truth poses.

Kinematic body model. We use an anatomically inspired body model with 159 joints and 67 degrees of freedom. This model was chosen because the minimal set of DOFs makes online tracking using Gauss-Newton algorithms very efficient. The body mesh, with 7324 vertices, is deformed by the skeleton using linear blend skinning. Body shapes are represented using a 128-dimensional shape basis [7], but only the first 10 are used for shape solving. The kinematic model and related C++ tracking code are set to be open-sourced by Q3 2024.

RGB keypoint estimation. We perform 2D body keypoint estimation on the RGB images (Figure 7). We use two different keypoint estimators: MediaPipe [8] and a separate model trained on internal manually annotated data.

Fig. 6: (Left) 6 of the 8 outside-in Kinect cameras are circled in green. (Right) point cloud constructed by fusing depth images from all 8 Azure Kinects.

Fig. 7: Four of the eight synchronized Azure Kinect images used for ground truth tracking. From left: RGB image, depth image, noisy RGB keypoints, body pose after combined depth and RGB signals. Additional examples can be found in the supplemental materials.

We find that neither model is sufficiently reliable in our cluttered environments. To improve reliability, we combine the 2D keypoints from both detectors across all 8 RGB cameras and iteratively discard outliers [13]. The remaining undiscarded keypoints are triangulated to give 3D joint locations.

Frame-to-frame tracking using deformable ICP. We iteratively fit the skinned body mesh to the fused point cloud (Fig. 6, right) using a frame-to-frame deformable ICP. The tracker is initialized at the first frame by fitting the triangulated RGB keypoints using a nonlinear Gauss-Newton solver. At each subsequent frame, the pose is extrapolated and then fit to the point cloud using deformable ICP [14]. Each vertex on the mesh is matched to the closest depth point, and we use our nonlinear solver to minimize the point-to-plane error [35]. To address frame-to-frame tracking loss due to the subject's fast motion, we incorporate a re-initialization step. If we see 100 consecutive frames where 3 or

Table 2: Compared with other available dataset, our dataset has a duration of 31.8 hours of real-data capture, aggregating to more than 3.4M frames at 30 fps. Our dataset also has the feasible headset configuration based on a realistic setting balancing practicality and ergonomics.

	Mo^2Cap^2	xrEgoPose	EgoCap	EgoGlass	UnrealEgo	EgoBody3M
Camera location	8cm from head	2cm from nose	25cm from head	1cm from head	1cm from head	1.5cm from head
Num. images	530k	380k	60k	170k	450k	13.7M(3.4M frames)
Num. cameras	1	1	2	2	2	4
Num. body keypoints	15	25	17	13	32	26
Num. hand keypoints	0	40	0	0	40	0
Image quality	low	realistic	real	real	realistic	real
Environment diversity	low	moderate	none	none	high (outdoor)	high (indoor)
Motion diversity	medium	medium	low	low	high	high

more RGB keypoints are incorrect (projected joint error is greater than 700 pixels) we assume that tracking is lost and reinitialize the frame-to-frame tracking by fitting the triangulated RGB keypoints.

Shape and scale calibration. The deformable ICP fit relies on having an accurate body mesh. To improve the quality of the tracking, we ask subjects to perform a calibration sequence where they perform several simple motions designed to exercise the full range of motion. After tracking the motion frame-to-frame, we perform a final global fit using Gauss-Newton to solve for the PCA shape coefficients and joint lengths that minimize the point-to-plane error between the body mesh and the point cloud across the entire sequence.

Annotation. To filter out incorrect ground truth, we ask annotators to review data at 2 fps using our annotation tooling and mark each frame with one of the following four annotations: "Perfect", "Slightly inaccurate", "Very inaccurate" and "Wrong", corresponding to joint errors of up to 2.5, 5, 10 cm, and greater than 10cm. "Slightly inaccurate" allows errors up to roughly the width of the wrist, which we determined qualitatively to be accurate enough for training. Frames marked as "very inaccurate" or "wrong" are ignored at train time as well as when computing metrics, leaving us with 3.4M "Perfect" or "Slightly inaccurate" frames. We refer to the resulting dataset as "EgoBody3M". See supplementary materials for additional statistics about the dataset.

EgoBody3M is the first large-scale real-image dataset for egocentric body tracking with a realistic VR camera configuration and a diverse group of subjects and motions. Tab. 2 presents a comparison with other datasets, showing that our dataset stands out as the largest real dataset with high motion diversity and multi-view of 4 cameras.

4 Method

The goal of egocentric body pose estimation is to learn the mapping $f(\cdot)$ from multi-view image streams $\{F_t\}_{t=1}^{T}$ (retrieved and processed from K headset-mounted cameras) to the user's pose in 3D space $\{p_t\}_{t=1}^{T}$,

$$f : \{F_t\}_{t=1}^{T} \rightarrow \{p_t\}_{t=1}^{T} \tag{1}$$

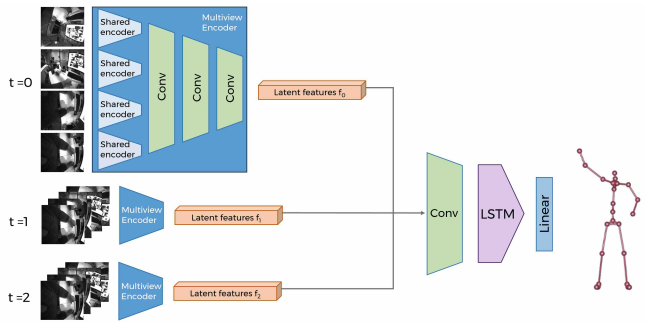


Fig. 8: Our model consists of a multi-view image encoder that extracts salient information from the image streams, and an LSTM that applies temporal reasoning. The multi-view image encoder is pre-trained with a 2D heatmap task.

where $F_t = \{I_t^k\}_{k=1}^K, p_t \in \mathbb{R}^{Jx3}$, K = number of camera views, and J = the number of body joints. We use a neural network to learn the function $f(\cdot)$, consisting of an image encoder that extracts information from the input images $\{I_t^k\}_{k=1}^K$ and a temporal model that performs temporal reasoning on the sequence of latent features $\{l_t\}_{t=1}^T$ (Fig. 8).

4.1 Encoder

Our encoder is inspired by encoder-decoder architecture, similar to a U-Net without skip connections [34]. The images from the 4 cameras are resized to 160×128 resolution using a Gaussian pyramid. The encoder initially processes each image independently (i.e., with shared weights applied to each image), and then concatenates the features. The combined features are convolved and downsampled together for several steps, generating a final latent representation l_t of the four camera views at each frame F_t. We pre-train this network to predict 2D heatmaps for each camera view. After that, we remove the decoder part and feed the latent code to a temporal model for 3D keypoints prediction.

4.2 Temporal model

We use a Long Short-Term Memory network (LSTM) [36] to apply temporal reasoning to the image features. To further promote the use of temporal information, we first take the latent features from the encoder and then feed them through a 1-layer temporal convolutional network (TCN). The TCN has a 3-frame receptive field and runs in a fully causal fashion to avoid introducing latency at runtime. The TCN output are then fed to the LSTM, which is a standard model with a 256-dimensional hidden feature size. The output is 26 3D keypoints.

4.3 Training and Optimizations

Our model is trained end-to-end, using a standard L2 pose loss to promote accuracy, and a constant pose jerk loss (via L2 regularization on the 4th order

pose derivative in time) to encourage temporally smooth predictions. During training, we break the longer sequences into smaller chunks, where each data sample consists of 31 sequential frames. During inference, the temporal model runs recurrently and updates only based on the latest frame. To ensure the hidden state at train time is representative, we use the first 27 frames to warm up the LSTM and only supervise the last 5 frames. Restricting supervision to the last 5 frames also ensures the gradients from all (batch size $\times$ 31 frames $\times$ 4 images) instances of the encoder can be stored in GPU memory. One issue we encountered was that because we always trained with 31-frame sequences, our network was unable to predict reasonable body poses for shorter sequences. To address this and ensure reasonable predictions even when starting with small initial values, we added an L2 regularization to the LSTM's internal hidden and cell state (i_h, i_c).

$$L_{pose} = L_2(\hat{p}_t - p_t) \tag{2}$$

$$L_{hidden} = L_2(i_h) + L_2(i_c) \tag{3}$$

$$L_{smooth} = L_2(p_t'''') \tag{4}$$

This results in the total training loss:

$$L_{total} = \lambda_{pose} L_{pose} + \lambda_{smooth} L_{smooth} + \lambda_{hidden} L_{hidden} \tag{5}$$

where $\hat{p}_t$ is the predicted 3D body joint positions, and L_2 denotes the L_2 norm. During training, we use $\lambda_{pose} = 1$, $\lambda_{smooth} = 0.1$ and $\lambda_{hidden} = 0.1$.

Our model, implemented in Pytorch, is trained on 6 servers with 6 GPUs each. We utilize a 1-cycle learning rate [37] for faster convergence, with a maximum rate of 0.004, and employ the ADAM optimizer and batch size of 32.

5 Results

We use mean per-joint position error (MPJPE) to evaluate the accuracy of our model. MPJPE computes the average 3D Euclidean distance (in centimeters) between the estimated and ground truth keypoints in the world space. Lower MPJPE means better accuracy. The other metric we track is mean per-joint *velocity* error (MPJVE). Velocities are computed using finite differences to for both the ground truth and prediction, and we compute average Euclidean error.

5.1 Results on EgoBody3M

We compare our model to the state-of-the-art model UnrealEgo [6] and also conduct an ablation study using common architecture components. The training process for UnrealEgo follows the recipe used by Hakada et al. [6], involving the separate training of UnrealEgo's 2D and 3D modules for the prediction of body keypoints, where the UnrealEgo model is initialized with pre-trained weights

Table 3: Comparison of our method to the state-of-the-art model UnrealEgo [6], and common architectures in the literature. UnrealEgo was initialized with ImageNet weights; all other models were initialized with a pretrained encoder-decoder (trained on our training set for 40,000 iterations). All models were then further trained on our dataset for 20,000 iterations. 2-stage training refers to models where the encoder is frozen to train the later MLP or temporal model; E2E means by end-to-end training where the encoder and MLP or temporal model are trained jointly.

Model	Training method	Overall MPJPE (cm)	Out-of-view Wrists MPJPE (cm)	Overall MPJVE (cm/frame)
UnrealEgo [6]	2-stage	7.41	23.7	1.27
2D keypoints + MLP (lifting)	2-stage	17.5	62.5	0.80
Latent + MLP	E2E	5.49	14.1	0.97
Latent + temporal	2-stage	6.54	20.9	0.57
Latent + temporal (ours)	E2E	**5.18**	**12.2**	**0.54**

from ImageNet [11]. Further details are reported in the supplementary material. Results show that for 3D pose estimation, latent features capture more salient information than 2D heatmaps, attaining lower MPJPE. The temporal model outperforms the per-frame lifting model MPJPE (5.18 vs 5.49 cm) and is much smoother, having *44% lower* MPJVE. The improvement in MPJVE can easily be seen in the supplemental video and the resulting smoothness is important for users to find the motion believable. Furthermore, the end-to-end trained model outperforms the 2-stage version, suggesting its ability to extract image information in addition to heatmaps.

Out-of-View vs In-View. One challenge for egocentric tracking involves having joints out of view of all the cameras, such as when the hands are behind the head. To capture this important case, we additionally evaluate wrist MPJPE on out-of-view wrists, where a given wrist joint is found to be outside the FOV of all four tracking cameras. As seen in Tab. 3, temporal models get 12% lower error on these challenging cases than single frame models.

Example results of different protocols. As shown in Fig. 9, our model is capable of accurate tracking even in challenging cases, such as arms crossed and out-of-view, squatting, resting hand on a cheek, etc. Notice that in these cases, a keypoint based method would fail at predicting the occluded body joints, while our method is able to reason about the subtle image cues and fill in the missing information to predict 3D body pose.

5.2 Comparison with SOTA methods on UnrealEgo

We compare our result against previous methods on the synthetic UnrealEgo dataset in Tab. 4. We used the same model architecture with modifications to the input and encoder. Specifically, input was changed to be two 256×256 images and the encoder depth was increased from 5 to 6 layers, which prevents the latent feature size from growing too large. We initialized with random weights and trained for 18000 iterations on a single machine with 8 GPUs. Our method

Fig. 9: Our model is capable of accurate tracking even in challenging cases, such as (a) arms crossed and out-of-view, (b) scratching the chin, (c) kicking and (d) squatting. Blue denotes ground truth and pink denotes model predictions.

improved on the results of UnrealEgo without pre-training by 2cm (23%) in MPJPE. Despite not using pre-training with ImageNet data, we also improved on the UnrealEgo model using ImageNet pre-training by 1.2cm (15%).

6 Demonstration on a VR headset

To demonstrate the practicality of our method, we showcase the body tracking algorithm in action on a VR headset (see supplementary material). To enable users to feel self-presence and control their avatar, we need to address three main aspects: body representation instead of keypoints, hand tracking, and representative body scale.

Body representation. To provide a realistic representation of the user's body, we need to re-target the tracking results onto an actual skinned body model. We use an internal model with 81 degrees of freedom and fit it at every frame using a Gauss-Newton solver. This allows us to show the user an actual body instead of just keypoints floating in space.

Table 4: Comparison of our approach vs previous methods on the UnrealEgo dataset.

Method	Overall MPJPE (cm)
xR-EgoPose (w/o pretrain)	12.32
xR-EgoPose (w/ pretrain)	11.29
EgoGlass (w/o pretrain)	10.77
EgoGlass (w pretrain)	9.14
UnrealEgo (w/o pretrain)	8.73
UnrealEgo (w/ pretrain)	7.91
Ours	6.70

Hand tracking. To create a plausible sense of self-presence, it is important to accurately track the user's hands. However, our body network does not attempt to predict hand pose as it would require increased network capacity and image resolution. Instead, we rely on an existing hand tracking solution [15] to estimate the hand poses. Since the hand network uses a higher resolution image of the hands, we assume that the resulting wrist position and orientation are more accurate than those predicted by the body tracking network. For frames with tracked hands, we use the hand wrist position and orientation as constraints in the Inverse Kinematics (IK) solver, generating a fused hand-body pose.

Body scale. During the IK fitting step, we allow the limb lengths to vary to fit the user's true limb lengths. To prevent sudden changes in the limb lengths, we apply a strong temporal regularization to the scale parameters. It's important to note that we do not attempt to guess the user's body shape, as it would be difficult to do so under clothing. Since our goal is to enable users to drive their avatar, precise body shape is not a primary concern of our work.

To reduce latency, the model is quantized to 8-bit and run on the Qualcomm Hexagon Tensor Accelerator. The final output layer of the LSTM is kept in 32-bit floats to reduce jitter in the final outputs.

7 Conclusion and Future Work

A novel ego-centric body tracking architecture is proposed that models the temporal history of body motion using multi-view latent features. Additionally, a new large-scale real egocentric body dataset, EgoBody3M, is released. The compelling body tracking experience is demonstrated on a real VR headset.

Limitations and Future Work: Our method faces challenges with near-field objects such us long hair or hats occluding the tracking cameras (see visual examples in the supplementary materials). The strong occlusions like a tabletop completely covering both legs increases the prediction errors much. Improvements can be made by varying training data and using motion priors from large datasets to regularize the body pose under challenging cases. In addition, since AR/VR headsets are accurately tracked in world-space by a combination of IMUs and computer vision, incorporating this additional signal can potentially help the model predict body motion based on the headset trajectory information.

Acknowledgements The authors would like to acknowledge the anonymous reviewers for their comments and corrections; Xuetong Sun and Fan Bu for their work on productizing body tracking; Steve Olsen, Kevin Harris, Steve Miller, Kaichen Sun, Ben Watson, Matthew Prasak, Daniel Frey, Gunnar Grismore, Andrew Anderson, Mark Hogan, and Neha Chachra for their help with data collection; David Dimond and Weijie Yu for their help with annotation; and Anastasia Tkach for machine learning development and experiments.

References

1. Apple Vision Pro. https://www.apple.com/apple-vision-pro/, accessed: 2023-11-17
2. Meta Quest 3. https://www.meta.com/ie/quest/quest-3/, accessed: 2023-11-17
3. Meta Quest Pro. https://www.meta.com/quest/quest-pro/, accessed: 2023-11-17
4. Microsoft Azure Kinect. https://azure.microsoft.com/en-us/products/kinect-dk, accessed: 2023-11-17
5. Pico 4. https://www.picoxr.com/global/products/pico4, accessed: 2023-11-17
6. Akada, H., Wang, J., Shimada, S., Takahashi, M., Theobalt, C., Golyanik, V.: UnrealEgo: A new dataset for robust egocentric 3D human motion capture. In: European Conference on Computer Vision (ECCV) (2022)
7. Allen, B., Curless, B., Popović, Z.: The space of human body shapes: reconstruction and parameterization from range scans. ACM Trans. Graph. (jul 2003)
8. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: BlazePose: On-device real-time body pose tracking. In: CVPR Workshop on Computer Vision for Augmented and Virtual Reality (2020)
9. Cha, Y.W., Price, T., Wei, Z., Lu, X., Rewkowski, N., Chabra, R., Qin, Z., Kim, H., Su, Z., Liu, Y., Ilie, A., State, A., Xu, Z., Frahm, J.M., Fuchs, H.: Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. IEEE Transactions on Visualization and Computer Graphics **24**(11), 2993–3004 (2018). https://doi.org/10.1109/TVCG.2018.2868527
10. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7103–7112 (2018)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Dittadi, A., Dziadzio, S., Cosker, D., Lundell, B., Cashman, T., Shotton, J.: Full-body motion from a single head-mounted device: Generating SMPL poses from partial observations. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (oct 2021)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6) (jun 1981). https://doi.org/10.1145/358669.358692
14. Hähnel, D., Thrun, S., Burgard, W.: An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In: Proceedings of IJCAI. IJCAI (2003)
15. Han, S., Wu, P.c., Zhang, Y., Liu, B., Zhang, L., Wang, Z., Si, W., Zhang, P., Cai, Y., Hodan, T., et al.: UmeTrack: Unified multi-view end-to-end hand tracking for VR. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–9 (2022)

16. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR (1994)
17. Hu, W., Zhang, C., Zhan, F., Zhang, L., Wong, T.T.: Conditional directed graph convolution for 3d human pose estimation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 602–611 (2021)
18. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7718–7727 (2019)
19. Ito, K., Tada, M., Ujike, H., Hyodo, K.: Effects of the weight and balance of head-mounted displays on physical load. Applied Sciences **11**(15) (2021)
20. Jeon, H.G., Lee, J.Y., Im, S., Ha, H., Kweon, I.S.: Stereo matching with color and monochrome cameras in low-light conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4086–4094 (2016)
21. Jiang, H., Ithapu, V.K.: Egocentric pose estimation from human vision span. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10986–10994. IEEE (2021)
22. Khirodkar, R., Bansal, A., Ma, L., Newcombe, R., Vo, M., Kitani, K.: Egohumans: An egocentric 3d multi-human benchmark. arXiv preprint arXiv:2305.16487 (2023)
23. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: Video inference for human body pose and shape estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5253–5263 (2020)
24. Li, J., Liu, C., Wu, J.: Ego-body pose estimation via ego-head pose estimation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society (jun 2023)
25. Li, S., Chan, A.B.: 3d human pose estimation from monocular images with deep convolutional neural network. In: Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12. pp. 332–347. Springer (2015)
26. LLC, P.: Pantone SkinTone Guide (2012)
27. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE international conference on computer vision. pp. 2640–2649 (2017)
28. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. pp. 483–499. Springer (2016)
29. Parger, M., Mueller, J.H., Schmalstieg, D., Steinberger, M.: Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality. In: Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology. VRST '18 (2018)
30. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7025–7034 (2017)
31. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7753–7762 (2019)
32. Remelli, E., Han, S., Honari, S., Fua, P., Wang, R.: Lightweight multi-view 3d pose estimation through camera-disentangled representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6040–6049 (2020)

33. Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.P., Schiele, B., Theobalt, C.: EgoCap: Egocentric marker-less motion capture with two fisheye cameras. ACM Trans. Graph. **35**(6) (dec 2016)
34. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (2015)
35. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proceedings Third International Conference on 3-D Digital Imaging and Modeling. pp. 145–152 (2001)
36. Sak, H., Senior, A., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128 (2014)
37. Smith, L., Topin, N.: Super-convergence: very fast training of neural networks using large learning rates. p. 36 (05 2019). https://doi.org/10.1117/12.2520589
38. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5693–5703 (2019)
39. Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., Fua, P.: Structured prediction of 3d human pose with deep neural networks. arXiv preprint arXiv:1605.05180 (2016)
40. Tekin, B., Rozantsev, A., Lepetit, V., Fua, P.V.: Direct prediction of 3d body poses from motion compensated sequences. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
41. Tome, D., Alldieck, T., Peluse, P., Pons-Moll, G., Agapito, L., Badino, H., De la Torre, F.: SelfPose: 3D egocentric pose estimation from a headset mounted camera. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3029700
42. Tome, D., Peluse, P., Agapito, L., Badino, H.: xR-EgoPose: Egocentric 3D human pose from an HMD camera. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7728–7738 (2019)
43. Wang, J., Liu, L., Xu, W., Sarkar, K., Luvizon, D., Theobalt, C.: Estimating egocentric 3d human pose in the wild with external weak supervision. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society (jun 2022)
44. 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 (CVPR) (2023)
45. Wang, J., Yan, S., Xiong, Y., Lin, D.: Motion guided 3d pose estimation from videos. In: European Conference on Computer Vision. pp. 764–780. Springer (2020)
46. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020)
47. Winkler, A., Won, J., Ye, Y.: QuestSim: Human motion tracking from sparse sensors with simulated avatars. In: SIGGRAPH Asia 2022 Conference Papers (2022)
48. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV). pp. 466–481 (2018)
49. Xu, W., Chatterjee, A., Zollhoefer, M., Rhodin, H., Fua, P., Seidel, H.P., Theobalt, C.: Mo^2Cap^2 : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. IEEE Transactions on Visualization and Computer Graphics (2019)

50. Zhang, J., Tu, Z., Yang, J., Chen, Y., Yuan, J.: Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13232–13242 (2022)
51. Zhang, Y., You, S., Gevers, T.: Automatic calibration of the fisheye camera for egocentric 3d human pose estimation from a single image. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1772–1781 (2021)
52. Zhao, D., Wei, Z., Mahmud, J., Frahm, J.M.: Egoglass: Egocentric-view human pose estimation from an eyeglass frame. In: 2021 International Conference on 3D Vision (3DV). pp. 32–41 (2021)
53. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15085–15099 (2023)