Bring Your Rear Cameras
for Egocentric 3D Human Pose Estimation

Supplementary Material

This supplementary material provides more details about our work. **Please also watch our supplementary video for dynamic visualizations, including the proposed datasets, qualitative results, and character animations as a future application of our method**.

A. Per-Action Evaluation

Table 6 presents the 3D error evaluation for each action category on Ego4View-RW. Our refinement method consistently brings substantial improvement across all action types. Notably, our proposed module demonstrates superior effectiveness, particularly for motions involving lower-body movements, achieving a 15.5% improvement in "stretching legs". This enhancement is likely attributed to the frequent self-occlusion of the legs in either front or rear views, which often leads to misrepresentation in joint detection by the current state-of-the-art method [4] even with four-view inputs. In contrast, our module effectively mitigates these challenges, resulting in significant advancements in egocentric 3D human pose estimation with rear-view integration.

Method	walking	kicking	boxing	crouching
EPF [4]	45.16	74.50	71.22	50.89
+ Ours	**42.10**	**68.85**	**60.81**	**46.20**
Method	kneeing	crawling	dancing	twisting body
EPF [4]	87.14	82.35	52.99	61.64
+ Ours	**73.69**	**76.92**	**48.75**	**56.03**
Method	stretching arms	stretching legs	rotating shoulders	raising legs
EPF [4]	50.03	58.14	53.56	75.62
+ Ours	**46.88**	**49.09**	**51.94**	**68.91**
Method	balancing legs up behind	sitting on the ground		all
EPF [4]	82.82	77.19		63.38
+ Ours	**74.24**	**66.79**		**56.94**

Table 6. **Per-action evaluation on Ego4View-RW** (MPJPE) with 2 front and 2 rear views. EPF represents EgoPoseFormer [4].

B. Joint Visibility Calculation

As described in Sec. 3, we obtain the visibility of end-effector joints (hands and feet) in our synthetic setup. We first generate 2D egocentric fisheye views using SMPL models with the predefined body part segmentation mesh [3]. Next, we project ground-truth 3D joints onto these images to obtain reference points, querying the nearest 2D points within a 10 × 10 pixel region around each reference. We classify a 3D joint as visible if any queried 2D point corresponds to its respective body part; otherwise, it is considered occluded.

C. Additional Details of Network Architecture

As mentioned in Sec. 4, we use several shallow networks, *i.e.,* $\mathcal{F}_{\text{O}}$, $\mathcal{F}_{\text{R}}$, $\mathcal{F}_{\text{HM}}$, $\mathcal{P}_{\text{HM}}$, $\mathcal{P}_{\text{RGB}}$, and $\mathcal{P}_{\text{Q}}$. $\mathcal{F}_{\text{O}}$ and $\mathcal{F}_{\text{R}}$ consist of two linear layers with a bilinear up-sample operation as well as with an intermediate dimension size of 64 and 128, respectively. $\mathcal{F}_{\text{HM}}$ uses one convolutional layer with a kernel size of 3, a stride of 2, and a padding size of 1, followed by two linear layers with an intermediate dimension size of 256, a bilinear up-sample operation, and a linear layer to generate heatmaps. $\mathcal{P}_{\text{HM}}$ is composed of two linear layers with intermediate and output dimension sizes of 256 whereas $\mathcal{P}_{\text{RGB}}$ and $\mathcal{P}_{\text{Q}}$ are a single liner layer with output dimension size of 256.

D. Rear-View Integration for Existing Method

As mentioned in Secs. 4 and 5, the refined heatmap features $\mathbf{R}_k$ and heatmaps $\widetilde{\mathbf{H}}_k$ can be utilised with existing 2D-to-3D lifting modules to estimate 3D poses. In this work, we integrate our module into the current state-of-the-art methods, EgoPoseFormer [4] and EgoTAP [2]. However, unlike the 3D module of EgoTAP [2] that directly uses heatmaps as inputs, EgoPoseFormer [4] uses the heatmap features (before the final heatmap output layer) instead of the heatmaps in their 2D-to-3D lifting module. To account for their methodology, we first input the refined features $\mathbf{R}_k$ from all views into a simple network consisting of four convolutional layers followed by a linear layer, yielding an initial 3D pose $\mathbf{P} \in \mathbb{R}^{16\times 3}$, which represents 16 joints including the head. This initial 3D pose is subsequently fed into the 3D updating module of EgoPoseFormer [4] to produce an updated 3D pose $\mathbf{P}_{\text{final}}$ as the final output.

E. Model Size

Here, we provide the details of our model size and inference speed with 2 front and 2 rear views. As mentioned in Sec. 4 and 5, we adopt the current state-of-the-art methods as a baseline, *i.e.,* EgoPoseFormer [1] (27M parameters) and EgoTAP [2] (242M parameters), with our refinement module (25M × the number of views). Note that EgoPoseFormer [1] consists of a simple UNet-based architecture and three transformer-based layers; therefore, they tend to have fewer trainable parameters than EgoTAP [2].

Regarding the inference speed on our setup with a single NVIDIA Quadro RTX 8000 and PyTorch, while the original EgoPoseFormer [1] runs at 67 fps, EgoPoseFormer with our refinement module can run at 30 fps with 2 front and 2 rear views. Therefore, our method can be utilized with real-time applications. Note that our focus lies in making the best use of rear views and prioritizes tracking accuracy; future work will improve inference speed with our proposed setup and novel large-scale datasets, *i.e.,* Ego4View-Syn and Ego4View-RW.

References

[1] Jiaxi Jiang, Paul Streli, Manuel Meier, and Christian Holz. Egoposer: Robust real-time egocentric pose estimation from sparse and intermittent observations everywhere. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 2

[2] Taeho Kang and Youngki Lee. Attention-propagation network for egocentric heatmap to 3d pose lifting. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 1

[4] Chenhongyi Yang, Anastasia Tkach, Shreyas Hampali, Linguang Zhang, Elliot J Crowley, and Cem Keskin. Egoposeformer: A simple baseline for stereo egocentric 3d human pose estimation. In *European conference on computer vision*, 2024. 1