EgoPoseFormer v2: Accurate Egocentric Human Motion Estimation for AR/VR

Supplementary Material

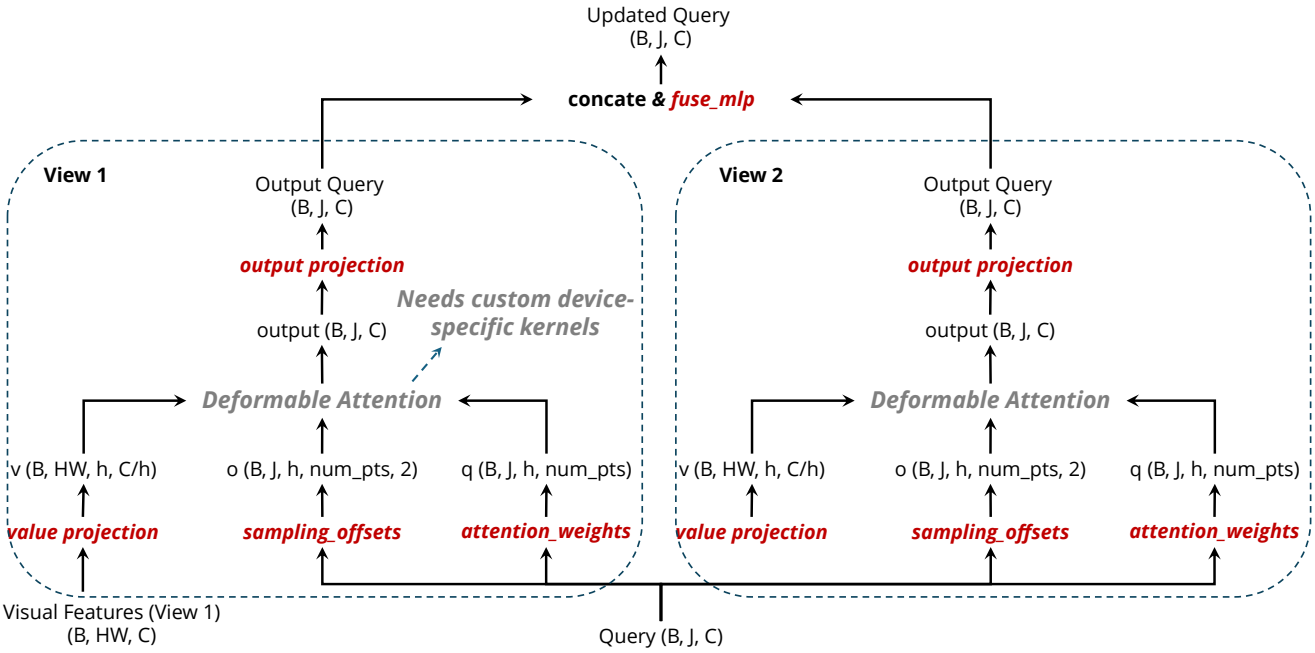

Figure 1. Deformable stereo attention module used in EPFv1 [5]. Each joint query independently attends to sampled image features via learned offsets and attention weights. Outputs from different views are sequentially fused using an MLP. This design introduces the specialized component with higher development complexity [4] and suboptimal hardware utilization. B, J, C, num_{pts}, and h denote batch size, number of joints, feature channels, reference points, and attention heads, respectively. Learnable layers are highlighted in red, with layer names matching Tab. 1.

1. Efficiency Analysis

This section provides a detailed efficiency comparison between EPFv2 and other models like EgoPoseFormer (EPFv1) [5] and EgoBody3M [6].

1.1. Cross-attention in EPFv1 and EPFv2

The key architectural difference lies in how spatial features are aggregated from multi-view images. EPFv1 uses deformable attention, which dynamically samples image features around projected keypoint locations using learned offsets. While deformable attention benefits from optimized CUDA kernels during training, achieving similar efficiency during on-device inference would incure a large amount of random memory reads [4], limiting its efficiency on edge-computing devices, *i.e.*, the AR/VR headset in our case. In contrast, EPFv2 employs standard cross-attention conditioned on concatenated projected 2D keypoints (Eq. 7 in the main paper), encoding multi-view information through the conditioning mechanism rather than explicit deformable sampling. We visualize both architectures in Fig. 1 and Fig. 2, with red labels denoting trainable components. Corresponding FLOPs and parameter breakdowns are summarized in Tab. 1. Apart from its simplicity, our method achieves even smaller cost (in terms of FLOPs and parameters) while offering significantly better deployment friendliness because of its reliance on standard operations.

1.2. The Impact of Joint Count

An efficiency bottleneck in prior transformer-based models such as EPFv1 is the one-to-one correspondence between body keypoint and query tokens. Specifically, the model uses a separate query per keypoint and each query independently attends to the feature maps. This design introduces computational overhead that scales linearly with the number of keypoints.

In contrast, our method adopts a single holistic pose query that aggregates all necessary information. This design decouples the model's computational complexity from the number or type of predicted joints. As a result, our architecture improves inference efficiency while also enabling seamless support for different parametric body models.

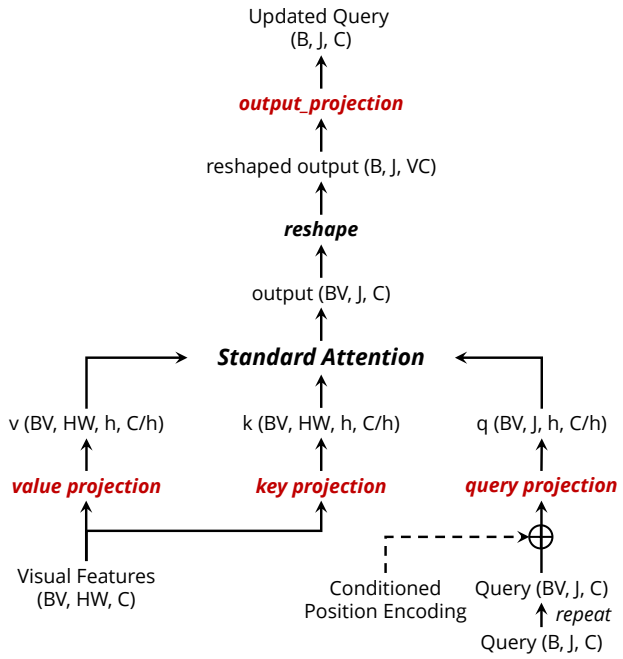

Figure 2. Our simplified multi-view cross-attention module built on standard attention. A single holistic query attends to all view features using conditioned positional encoding in a batch manner, enabling more efficient and scalable spatial fusion. V denotes the number of views. Learnable layers are highlighted in red, with naming aligned to Tab. 1.

Tab. 1 quantifies the benefits of this design. Compared to a 16-query baseline using standard attention, our single-query variant reduces FLOPs by over 4× (622K vs. 131K) while maintaining the same number of parameters. The reduced computational cost makes the model more suitable for deployment on edge devices, and the unified query design simplifies implementation and integration across different body models.

1.3. Efficiency Comparison with EgoBody3M

EPFv2 and EgoBody3M [6] are both temporal models, enabling a fair, end-to-end efficiency comparison. Here, we focus the comparison on the pose estimation head as the image encoders are subject to change. As shown in Tab. 2, our transformer-based design significantly reduces the parameter count from 14.96M to 0.83M and FLOPs from 39.76G to 10.52G. This reduction in parameters and drop in FLOPs highlights the effectiveness of our proposed architecture, making it highly suitable for real-time deployment on resource-constrained devices.

1.4. Latency measurement

We provide the latency measurement of EPFv2. Specifically, we report latency for both plain PyTorch models (common in research) and ONNX models (suitable for deployment). For ONNX, CPU latency was measured using FP32 with `onnxruntime`, and GPU latency of both FP32 and FP16 were measured using TensorRT's `trtexec` with CudaGraph enabled at optimization *lvl.* 0 (least optimized) and *lvl.* 5 (most optimized). All experiments use 4-view 256×320 images (our baseline setting) on an Intel Xeon Platinum 8339HC CPU and NVIDIA A100 GPU.

Table 1. Layer-wise comparison of parameter count and FLOPs for different spatial attention modules. (1) EPFv1 uses deformable attention with 16 joint queries and reference points, introducing moderate compute and high development complexity. (2) A baseline standard attention setup with 16 queries shows 4.75× higher FLOPs due to dense spatial interactions. (3) Our final design adopts a single holistic query and standard attention, achieving the lowest computation (131K FLOPs) while maintaining similar parameter count to other variants.

Method	Layer	# of *params*	FLOPs
(1) EPFv1 Deformable Attention w/ 16 Queries w/ 16 reference points	sampling offsets	16512	32768
	attention weights	8256	16384
	value projection	16512	32768
	output projection	16512	32768
	fuse mlp	32896	32768
	norm	256	0
Total	-	90944	147456
(2) EPFv2 w/ 16 Queries Standard Attention	query projection	16512	32768
	key projection	16512	32768
	value projection	16512	32768
	output projection	32896	524288
	norm	256	0
Total	-	82688	622592
(3) EPFv2 w/ 1 Query Standard Attention **(Our Final Design)**	query projection	16512	32768
	key projection	16512	32768
	value projection	16512	32768
	output projection	32896	32768
	norm	256	0
Total	-	82688	131072

Table 2. Efficiency Comparison with EgoBody3M. We compare only the pose estimation heads, assuming a shared backbone. EPFv2 significantly reduces parameter count and FLOPs, demonstrating the efficiency of its streamlined transformer design.

Method	# of *params*	FLOPs
EgoBody3M [6]	14.96M	39.76G
EPFv2	0.83M	10.52G

Table 3. Latency measurement of EPFv2.

Backbone	FP16	Plain PyTorch	ONNX CPU	TensorRT *lvl.* 0	TensorRT *lvl.* 5
ResNet18		9.8 ms	59.2 ms	3.3ms	1.3ms
	✓	8.3 ms	–	1.6ms	1.0ms
MobileNetV4S		14.9 ms	23.8 ms	3.0ms	1.1ms
	✓	13.3 ms	–	1.3ms	0.8ms

2. Additional Benchmark on Ego4View-Syn

In the main paper, we focus our evaluation and ablation studies on the EgoBody3M [6] dataset, since it contains long egocentric video sequences with 3D ground-truth poses. To further validate our architecture, we conduct an

Table 4. **Comparison on Ego4View-Syn [1].** We evaluate a single-frame variant of EPFv2 on the Ego4View-Syn dataset. Despite removing temporal modeling, our method achieves strong results, outperforming prior works in PA-MPJPE and remaining competitive in MPJPE.

Method	MPJPE	PA-MPJPE
EgoPoseFormer [5]	27.36	23.31
EgoRear [1]	**27.04**	23.18
EPFv2	27.94	**22.53**

Table 5. Impact of temporal sequence length. Using only two frames leads to significant performance degradation, while longer sequences offer stable and improved accuracy. We use 16 as the default setting for fair comparison with prior work [6].

Length	Overall MPJPE
2	4.30
16 **(default)**	**4.17**

additional benchmark on the Ego4View-Syn dataset proposed in the recent SoTA method EgoRear [1].

For a fair comparison with existing single-frame methods on this dataset, we adapt our model by removing the causal temporal attention module, effectively reducing EPFv2 to a single-frame baseline method. Following EgoRear's architecture, we use three transformer layers: one for pose proposal and two for refinement. Since Ego4View-Syn does not provide auxiliary inputs (e.g., headset pose), we initialize the holistic query using an MLP over the image features, following EPFv1 [5].

All training implementations are kept consistent with EgoRear. As shown in Tab. 4, our "simplified" single-frame version achieves competitive MPJPE and the best PA-MPJPE among all compared methods. This confirms that our architectural design offers strong performance even in the absence of temporal modeling.

3. Additional Ablation Study

We present two additional ablation studies to further investigate the impact of temporal context length and image backbone capacity on the performance of EPFv2.

3.1. Sequence Length

We vary the causal temporal attention window length to assess the contribution of temporal modeling. As shown in Tab. 5, using only two frames leads to a substantial performance drop, indicating that temporal context is crucial for accurate 3D pose estimation. For longer sequences, performance remains stable. We adopt a length of 16 frames as the default setting, following EgoBody3M [6], to ensure both fair comparison and efficient training.

Table 6. Comparison across different image backbones. Larger backbones, such as DINOv3-B, significantly improve pose estimation accuracy, demonstrating the benefit of strong visual features in egocentric settings.

Variants	# of *params.*	Overall MPJPE
ResNet-18 [2]	12.5M	4.17
ResNet-50 [2]	43.3M	4.03
DINOv3-ViT-B [3]	119.2M	**3.96**

3.2. Image Encoders

We also evaluate the effect of different image backbone architectures, ranging from lightweight (ResNet-18 [2]) to large-scale vision transformers (DINOv3-B [3]). Results in Tab. 6 show a clear trend: stronger backbones consistently yield better 3D pose estimation, with DINOv3-B achieving the best performance, highlighting the benefit of high-quality features for challenging egocentric scenarios.

References

[1] Hiroyasu Akada, Jian Wang, Vladislav Golyanik, and Christian Theobalt. Bring your rear cameras for egocentric 3d human pose estimation. *arXiv preprint arXiv:2503.11652*, 2025. 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[3] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 3

[4] Chenhongyi Yang, Tianwei Lin, Lichao Huang, and Elliot J Crowley. Widthformer: Toward efficient transformer-based bev view transformation. In *IROS*, pages 8457–8464. IEEE, 2024. 1

[5] 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 *ECCV*, pages 401–417, 2024. 1, 3

[6] Amy Zhao, Chengcheng Tang, Lezi Wang, Yijing Li, Mihika Dave, Lingling Tao, Christopher D Twigg, and Robert Y Wang. Egobody3m: Egocentric body tracking on a vr headset using a diverse dataset. In *ECCV*, pages 375–392, 2024. 1, 2, 3