E-3DPSM: A State Machine for Event-Based Egocentric 3D Human Pose Estimation

Supplementary Material

Table of Contents:

A. Dataset Preprocessing

EE3D-R [25] and EE3D-W [26] datasets provide continuous event streams without natural frame boundaries. To make them suitable for training, we discretise the streams into fixed temporal windows of 20 ms. Within each window, we group events into batches of $\approx 8 \cdot 10^3$, which provides a balanced trade-off between temporal resolution and data compactness. For each discretised segment, we generate a frame-based LNES [31] event representation that preserves polarity and temporal ordering. By creating these representations beforehand, rather than during training, we ensure consistent frame counts across the continuous streams, and this greatly improves the efficiency of data loading.

B. Pose Drift under Naive Fusion

In the naive fusion approach, the 3D pose at each timestep is obtained by adding the predicted pose from the previous timestep to the current delta pose; see Eq. (11). While simple, it leads to the accumulation of errors over time, especially when delta pose estimates are noisy or when pose predictions suffer from transient uncertainties. This error accumulation results in increasing drift, causing the predicted poses to deviate from the ground truth as the sequence progresses. As shown in Fig. 6, the MPJPE steadily increases under naive fusion, highlighting the drift. In contrast, the direct pose method fluctuates due to its reliance on independent predictions at each timestep, resulting in less consistent performance. On the other hand, our learned fusion approach remains stable and produces a consistently lower error than both the naive fusion and direct pose methods, effectively mitigating drift and preserving accuracy over time.

C. Model Efficiency

We evaluate the efficiency of our approach compared to the baselines in terms of parameter count, FLOPs, GPU memory requirement, and 3D pose update rate. As shown in Tab. 5, our E-3DPSM incurs moderately higher computational cost than existing (more lightweight) baselines, yet remains within the same order of magnitude and achieves real-time performance on a single NVIDIA A6000 GPU. In Tab. 6, we report a detailed per-module computational requirement breakdown of our method, highlighting the primary contributors. Our design strikes a favourable balance that enables substantial improvements in accuracy and stability while preserving real-time responsiveness. It demonstrates that robustness under challenging motion and occlusion can be achieved without sacrificing deployment feasibility.

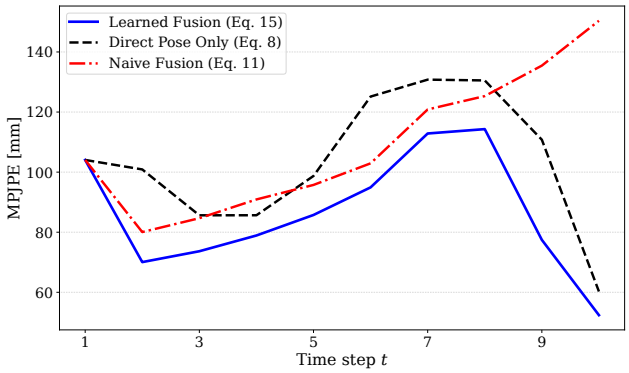

Figure 6. **Pose drift over time.** Comparison of learned fusion (Eq. (15)), direct pose only (Eq. (8)), and naive fusion (Eq. (11)) across temporal sequence length. Naive fusion leads to rapidly increasing drift, whereas our learned fusion effectively mitigates this drift, maintaining stable accuracy over time.

D. Additional Evaluations

D.1. Comparison with Kalman-Smoothed Baselines

In our method, the Kalman filter is a learned module used for pose fusion inside the network, rather than a post-hoc smoothing step. For fairness, we also apply inference-time Kalman filtering (KF) to prior baselines, where it serves only as an external temporal smoother. This experiment reveals that our improvements cannot be attributed simply to filtering, but to the way fusion is integrated and trained within the architecture. As shown in Tab. 4, our approach achieves substantially lower MPJPE and smoother predictions compared to Kalman-smoothed baselines. These results confirm that the gains primarily arise from our design for learned pose fusion and temporal modelling, which can-

Table 4. **Comparison with Kalman-smoothed baselines on the EE3D-R dataset.** We apply inference-time Kalman filtering (KF) to prior methods to rule out post-hoc smoothing as the main reason for improvements. Our method achieves substantially lower MPJPE and e_{smooth}, demonstrating that the performance is due to the proposed architecture and not filtering in post-processing.

Method	MPJPE ↓	PA-MPJPE ↓	e_{smooth} ↓
EgoPoseFormer [44] with KF	144.27	92.32	37.11
EventEgo3D [25] with KF	107.23	82.54	15.78
EventEgo3D++ [26] with KF	100.98	75.57	13.98
Ours (Causal)	**84.45**	**62.64**	**8.40**
Ours (Non-Causal)	**81.32**	**60.21**	**6.65**

Table 5. **Model efficiency comparison** in terms of parameters, FLOPs, GPU memory, and 3D pose update rate in Hz (measured on a single NVIDIA A6000 GPU).

Method	Params ↓	FLOPs ↓	GPU Memory ↓	Pose Update Rate ↑
EgoPoseFormer [44]	14.1 M	5.5 G	82 MB	130
EventEgo3D [25]	**1.25 M**	**416.84 M**	**25 MB**	**139**
EventEgo3D++ [26]	**1.25 M**	**416.84 M**	**25 MB**	**139**
Ours	6.64 M	8.16 G	74 MB	80

Table 6. Detailed module-wise FLOPs breakdown.

Modules	FLOPs
CNN Layers	7.05 G
Deformable Attention	0.89 G
SSM	0.06 G
Query Decoder	0.15 G
Pose Heads	10^{-3} G
Fusion	10^{-4} G
Total	**8.16 G**

not be replaced by external filtering applied after prediction.

D.2. Per-Joint and Per-Action Evaluation

We report detailed per-joint and per-action results on EE3D-R [25] and EE3D-W [26] datasets. For each dataset, we provide MPJPE and PA-MPJPE per body part together with the mean across joints, as summarised in Tab. 14. Overall, the trends mirror the aggregate findings in the main paper. Improvements are consistent across nearly all joints, with particularly large gains on distal joints such as wrists, ankles, and feet, which are challenging due to fast motion and frequent self-occlusions.

We further break down performance by action classes in Tab. 13 using the same metrics. The results demonstrate consistent gains across diverse activities, including dance, sports, and highly articulated motions; see Figs 13 and 14. Notably, the improvements are most pronounced in occlusion-prone actions such as kicking, crawling, and crouching. Additionally, the jitter plots of the end effector joints on EE3D-R (Fig. 11) and EE3D-W (Fig. 12) highlight the reduced jitter in these occlusion-heavy joints compared to prior methods.

Table 7. **Training strategy ablation on the EE3D-R dataset.** We compare causal (forward) vs. non-causal (bidirectional) training and different sequence lengths used during training.

Training Strategy	MPJPE ↓	PA-MPJPE ↓	e_{smooth} ↓
Training Directionality			
Causal (Forward Only)	89.88	66.74	10.14
Non-Causal (Ours)	**84.45**	**62.64**	**8.40**
Pose Sequence Length (N)			
20 poses	86.25	65.62	9.76
30 poses	86.03	64.87	8.95
40 Poses (Ours)	**84.45**	**62.64**	**8.40**

D.3. Occlusion-Only Evaluation

To quantify robustness under occlusions, we evaluate only time steps and joints that are marked as occluded by the dataset-provided visibility masks. We focus on end-effectors that are most susceptible to self-occlusion and fast motion: elbows, wrists, knees, ankles, and feet. For each method, we report MPJPE, MPJPE-PA, and jitter plots restricted to the occluded subset.

Tab. 12 summarizes the per-joint results for occluded end-effectors on EE3D-R and EE3D-W. The numbers show clear and consistent gains for our approach across all end-effectors. Improvements are most pronounced at the distal joints, such as wrists, ankles, and feet, where occlusions and rapid movements typically cause large errors. This strong performance under occlusions comes from the SSM blocks used in SPEM. SSM maintains an internal latent state that evolves smoothly over time, allowing the model to integrate motion information across long temporal ranges. During occlusions—when spatial features are weak or absent—the SSM maintains and propagates a coherent motion state rather than relying solely on the current input. This temporal continuity helps preserve joint trajectories and reduce jitter for occluded joints. Overall, the occlusion-only analysis demonstrates that our method effectively mitigates failures arising from self-occlusion, leading to more reliable pose recovery under challenging visibility conditions.

E. Additional Ablations

E.1. Training Strategy

To assess the impact of the training strategy, we experiment with directionality and sequence length on EE3D-R, as summarised in Tab. 7.

Directionality. We compare causal (forward-only) training with non-causal (bidirectional) training. In the causal train-

Table 8. **Inference-time ablation on the EE3D-R dataset comparing different strategies for resetting internal states.** We evaluate resetting the SSM block states, resetting the Kalman fusion states, and using continuous state evolution without resets (ours).

State Reset Strategy	MPJPE ↓	PA-MPJPE ↓	e_{smooth} ↓
SSM Reset (40 Frames)	104.56	81.24	15.20
Fusion Reset (40 Frames)	95.40	70.90	21.50
No Reset (Ours)	**84.45**	**62.64**	**8.40**

ing setup, SSM has access only to past observations. In non-causal training, SSM incorporates both past and future context (bidirectional) during training, and when evaluated with causal inference, achieves lower errors and smoother trajectories compared to causal-only training. This demonstrates that bidirectional context at training time helps the model learn stronger motion priors that continue to generalise even when the model is deployed in a strictly causal setting.

Pose Sequence Length. In this ablation, we vary the number of poses N used during training to study its effect on temporal modelling. With $N = 20$, SPEM receives a limited temporal context, which reduces accuracy and increases jitter. Increasing the sequence length to $N = 30$ improves accuracy and reduces jitter. Training with $N = 40$ poses yields the best overall performance, indicating that longer sequences allow the model to learn richer motion dynamics and produce more stable and accurate predictions.

E.2. Internal State Reset

We study the effect of different inference-time state reset strategies on EE3D-R. Since our model maintains internal states in both the SSM blocks and the learned fusion module, one natural question is whether these states should be periodically reset to avoid drift. To investigate this, we evaluate three settings: 1) resetting the SSM states every 40 frames, 2) resetting the Kalman fusion states every 40 frames, and 3) no resets, where states evolve continuously across the entire test sequence.

As shown in Tab. 8, periodic resets do not yield benefits and in fact can degrade performance, either by increasing error or by reducing temporal smoothness. In contrast, continuous state evolution without resets achieves the best results, indicating that the model learns to regulate its internal states without the need for manual intervention. This analysis confirms that stability in our framework naturally arises from learned dynamics and fusion mechanisms.

E.3. Different Event Representations

We analyse the influence of different event representations on the pose estimation performance. Prior work by Gehrig et al. [7] introduced a versatile end-to-end trainable voxel-based event stream representation for learning. We extensively experimented with it in our framework at early and intermediate project stages and found that it leads to poor generalisation across datasets in our setting. We also experimented with a *Learned LNES* variant of the static LNES [31], where each 2D entry is assigned a learnable weight based on its spatial-temporal coordinates (x, y, t) and polarity p. These weights modulate local event aggregation, emphasising informative motion boundaries while suppressing noise. As shown in Tab. 9, the standard predeined LNES [31] yields the best accuracy and temporal smoothness. This suggests that structured, interpretable representations, such as LNES, provide a strong inductive bias for egocentric event-based pose estimation, achieving robustness without additional learnable overhead. Overall, the question of whether pre-defined or learnable event stream representations for learning are the best choices remains problem-dependent and open in the broader context of event-based vision.

Table 9. **Design choice study for event stream representation for learning on the EE3D-R dataset.** We experiment with learned voxel-based representation, learned LNES and fixed LNES.

Event Representation	MPJPE ↓	PA-MPJPE ↓	e_{smooth} ↓
Learned Voxel-Based [7]	100.25	78.54	10.78
Learned LNES	93.78	71.12	9.14
LNES (Ours)	**84.45**	**62.64**	**8.40**

Table 10. **Inference-time ablation on the EE3D-R dataset comparing the use of different 3D pose update rates frequencies.**

Event Frequencies	MPJPE ↓	PA-MPJPE ↓	e_{smooth} ↓
20 Hz (50 ms)	87.45	63.74	11.90
25 Hz (40 ms)	84.57	62.80	9.49
50 Hz (20 ms) (Ours)	**84.45**	**62.64**	**8.40**

E.4. Inference-Time Event Frequencies

Tab. 10 summarises the evaluation of the robustness of our model to different event window durations (T) during inference, effectively varying the target 3D pose update rates. Although the model is trained with an event window of $T = 20$ ms (50 Hz), it maintains stable accuracy across a wide range of frequencies without significant degradation. This flexibility is valuable for real-world deployment, where event rates may vary due to motion dynamics or hardware constraints.

We attribute this robustness to the event-specific S5 blocks [8], whose learnable timescale parameters dynamically adapt to varying temporal resolutions. This capability allows the model to adjust to changes in the effec-

Table 11. **Ablation of global and input/state-dependent covariance learning for Q and R on EE3D-R dataset.**

Learning Strategy	MPJPE ↓	PA-MPJPE ↓
Input/State Dependent	91.15	69.74
Global Learned (Ours)	84.45	62.64

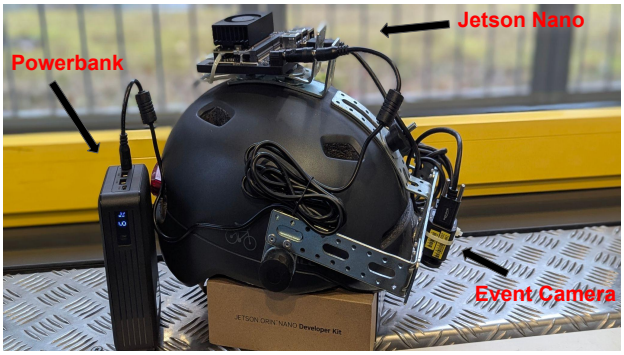

Figure 7. **Our head-mounted device setup.** The device uses a single fisheye egocentric event camera for input, NVIDIA Jetson Orin Nano for onboard processing, and a portable powerbank for standalone operation.

tive event rate, maintaining stable temporal modelling and smooth pose evolution across different inference frequencies.

E.5. Learning Strategy for Q and R

We evaluate an input-dependent formulation of the process ($\mathbf{Q}$) and measurement ($\mathbf{R}$) noise covariances by predicting them with lightweight MLPs conditioned on feature embeddings $\mathbf{F}$ (see Sec. 4.1). Since $\mathbf{F}$ depends on the internal latent state $\mathbf{Z}$, the resulting $\mathbf{Q}$ and $\mathbf{R}$ are implicitly both input- and state-dependent. As shown in Tab. 11, this variant performs worse than our proposed globally learned $\mathbf{Q}$ and $\mathbf{R}$. We observe that the latter act as stable, calibrated priors for temporal fusion, whereas input-dependent covariances introduce additional flexibility that leads to overfitting and less stable filtering.

F. Head-Mounted Device and Real-Time Demo

We build a head-mounted setup following the specifications of EventEgo3D++ [25, 26], with an additional down-facing fisheye Ximea MU050CR-SY RGB camera [42] for reference views and an NVIDIA Jetson Orin Nano Super Developer Kit for portable onboard processing (see Fig. 7).

For our real-time demo, we deploy the Jetson Orin Nano with a power bank for fully portable operation, enabling evaluation on in-the-wild sequences under low-light and fast-motion scenarios. To visualise the predicted 3D poses,

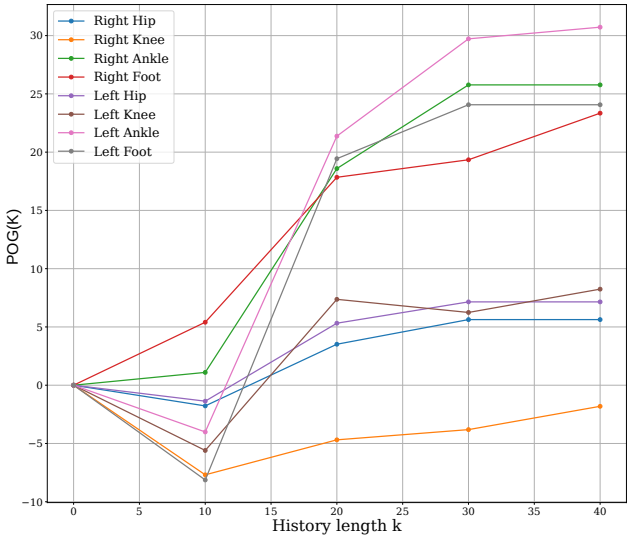

Figure 8. We plot the improvement in MPJPE obtained by increasing the duration of temporal history k, showing how a longer past context yields larger gains for occluded lower body joints.

we implement a lightweight client-server viewer over WebSockets, where an iPad acts as the client device streaming poses in real time (see Fig. 10). Our method operates reliably on this portable setup, achieving ≈30 Hz pose update rates on real event streams (see our video 8:10-9:36). When using a laptop equipped with an NVIDIA 3050 Ti carried in a backpack, our method achieves ≈50 Hz.

G. Past-Only Gain Analysis

To quantify how temporal history improves the accuracy of occluded joints in our method, we introduce and calculate the past-only gain (POG) metric, which is defined as follows. Let t denote a timestep where a joint is currently occluded and has been fully visible for the previous $N = 40$ frames $(t-N, \ldots, t-1)$. We define k as the history length, that is, the number of most recent visible frames before t that the model is allowed to use when predicting the pose at time t. For instance, $k = 0$ means the prediction uses no temporal history, and $k = 40$ means the prediction uses the last 40 visible frames preceding the occlusion. For a given joint and occluded timestep t, let MPJPE_t^k denote the per-joint MPJPE when using a history of length k. The POG is defined as

$$\text{POG}(k) = \text{MPJPE}_t^0 - \text{MPJPE}_t^k, \tag{25}$$

which measures how much MPJPE is reduced at the same occluded frame when the model has access to k frames of past information. A positive value indicates that temporal history improves occlusion accuracy. We compute this metric for multiple history lengths k. For each k, we evaluate

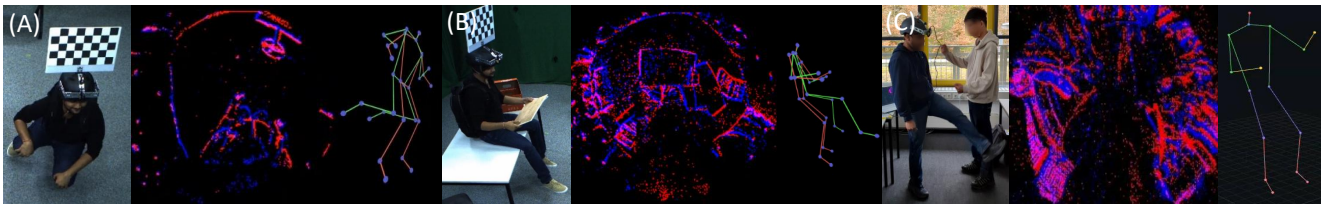

Figure 9. **Failure cases for different scenarios.** (A) Strong self-occlusion crawl action, (B) interaction with objects, (C) other humans in the FOV. External views are only for reference. **Red**: Predicted pose. **Green**: Ground truth. C visualises our prediction only (no ground truth available). *Inputs to E-3DPSM are egocentric LNES frames.*

predictions at the exact same occluded timesteps t, compute MPJPE_t^k, and pair it with the baseline error MPJPE_t^0. Averaging these paired values across all selected occlusion frames yields a POG plot for each joint.

The resulting plot Fig. 8 shows positive gains for lower-body joints. The largest improvements appear for ankles and feet, which undergo frequent occlusions in egocentric settings. Increasing the history length produces progressively lower MPJPE at occluded frames, indicating that the model benefits from a richer temporal context. This confirms that our continuous state formulation effectively preserves long-range motion structure and leverages it to recover 3D human poses under severe occlusions.

H. Limitations

Fig. 9 shows challenging scenarios involving strong self-occlusions during crawling (Fig. 9-A), interactions with objects (Fig. 9-B), and in the presence of other humans within the field of view (Fig. 9-C). Such situations can lead to degraded 3D pose accuracy due to missing or ambiguous motion cues. In addition, abrupt illumination changes such as flickering effects (see crouching in our video 7:50-8:06) can lead to occasional temporal instability, particularly during fast and complex motions. These limitations suggest several directions for future work: Modelling occlusions explicitly and generative pose refinement could improve the plausibility of 3D poses when observations are incomplete.

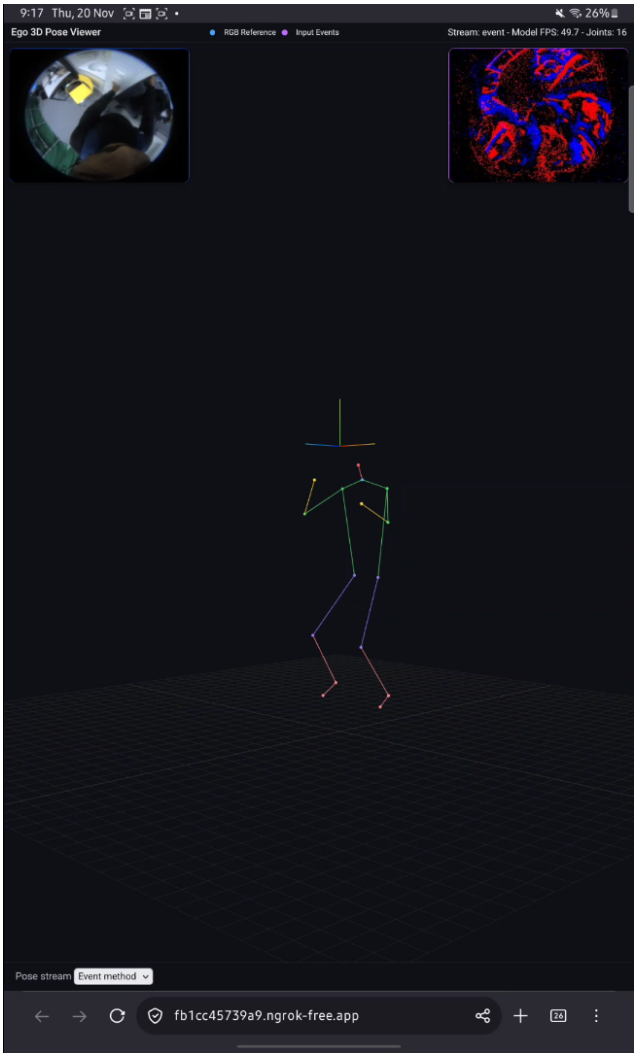

Figure 10. **Our real-time viewer.** Screenshot of our iPad-viewer showing the live event stream, reference RGB view, and the predicted 3D skeleton rendered in real time. Note that there is a transmission delay of 3–5 poses.

Table 12. **Quantitative results for occlusion-only end-effector joints for the EE3D-R and EE3D-W datasets**.

Dataset	Method	Metric ↓	Elbow	Wrist	Knee	Ankle	Foot	Avg.
EE3D-R	EgoPoseFormer [44]	MPJPE	47.85	28.34	100.11	204.64	169.94	110.18
		PA-MPJPE	47.06	30.50	51.82	83.61	65.20	55.64
	EventEgo3D [25]	MPJPE	37.46	21.97	79.20	156.93	138.95	86.90
		PA-MPJPE	40.41	25.78	48.58	78.27	61.63	50.93
	EventEgo3D++ [26]	MPJPE	34.41	20.87	81.73	158.51	146.62	88.43
		PA-MPJPE	37.89	24.20	48.48	77.18	59.90	49.53
	Ours (Causal)	MPJPE	**29.91**	**16.12**	**60.53**	**120.54**	**108.37**	**67.49**
		PA-MPJPE	**31.68**	**19.02**	**41.08**	**63.02**	**52.37**	**41.85**
	Ours (Non-Causal)	MPJPE	**28.84**	**15.54**	**57.97**	**115.94**	**105.18**	**64.69**
		PA-MPJPE	**30.29**	**18.21**	**39.09**	**61.63**	**51.28**	**40.10**
EE3D-W	EgoPoseFormer [44]	MPJPE	140.34	173.93	157.29	177.07	181.12	173.01
		PA-MPJPE	104.34	119.89	102.39	124.28	132.86	113.67
	EventEgo3D [25]	MPJPE	64.56	100.27	194.68	264.90	174.81	159.04
		PA-MPJPE	61.44	101.60	76.81	105.15	79.18	84.84
	EventEgo3D++ [26]	MPJPE	61.73	84.95	177.79	242.70	167.94	147.42
		PA-MPJPE	57.28	87.80	74.33	95.53	73.24	77.64
	Ours (Causal)	MPJPE	**54.04**	**75.69**	**162.53**	**229.09**	**157.84**	**135.84**
		PA-MPJPE	**52.91**	**80.24**	**70.06**	**92.96**	**69.78**	**73.19**
	Ours (Non-Causal)	MPJPE	**54.72**	**74.84**	**160.40**	**225.28**	**156.34**	**134.72**
		PA-MPJPE	**52.57**	**78.74**	**68.90**	**92.11**	**69.14**	**72.29**

Table 13. **Per-action quantitative results for the EE3D-R and EE3D-W datasets**.

Dataset	Method	Metric ↓	Walk	Crouch	Pushup	Boxing	Kick	Dance	Inter. w/ env.	Crawl	Sports	Jump	Avg.
EE3D-R	EgoPoseFormer [44]	MPJPE	123.65	175.60	184.12	150.45	125.99	119.71	153.75	226.87	143.84	136.98	154.09
		PA-MPJPE	78.70	107.19	109.20	102.66	91.67	86.74	87.95	111.85	96.71	99.60	97.22
	EventEgo3D [25]	MPJPE	74.75	144.26	109.58	141.19	104.33	89.98	103.15	118.30	108.23	105.77	109.95
		PA-MPJPE	55.92	97.73	87.40	109.70	85.13	72.10	72.63	85.87	84.91	87.31	83.86
	EventEgo3D++ [26]	MPJPE	71.52	150.57	93.54	125.75	98.46	85.45	96.10	116.76	98.71	97.10	103.39
		PA-MPJPE	52.35	99.96	71.12	94.82	79.93	65.97	68.60	81.29	76.16	76.56	76.67
	Ours (Causal)	MPJPE	**57.12**	**124.89**	**75.77**	**97.07**	**80.83**	**71.88**	**83.54**	**91.78**	**82.48**	**77.78**	**84.31**
		PA-MPJPE	**39.49**	**86.09**	**57.63**	**74.61**	**62.25**	**52.84**	**59.37**	**65.66**	**63.02**	**60.52**	**62.14**
	Ours (Non-Causal)	MPJPE	**52.79**	**121.08**	**73.73**	**94.06**	**76.49**	**68.81**	**80.77**	**88.42**	**79.60**	**76.01**	**81.17**
		PA-MPJPE	**34.72**	**83.53**	**55.10**	**71.66**	**59.12**	**51.00**	**57.54**	**63.49**	**60.96**	**59.81**	**59.69**
EE3D-W	EgoPoseFormer [44]	MPJPE	176.96	251.27	263.46	215.26	180.34	171.24	220.04	324.68	205.83	188.64	220.40
		PA-MPJPE	106.25	144.71	147.41	138.59	123.75	117.10	118.74	151.08	130.41	128.30	130.45
	EventEgo3D [25]	MPJPE	190.79	175.79	178.25	153.63	188.65	182.25	187.92	181.10	223.55	208.59	187.05
		PA-MPJPE	105.58	107.45	112.51	77.66	101.43	108.35	101.95	98.08	122.16	112.70	104.78
	EventEgo3D++ [26]	MPJPE	166.32	155.40	166.60	141.25	157.74	159.02	163.52	149.61	206.15	179.59	164.52
		PA-MPJPE	92.97	97.60	106.26	65.18	85.14	100.38	91.47	84.32	113.47	104.90	94.16
	Ours (Causal)	MPJPE	**144.64**	**138.75**	**148.76**	**131.62**	**146.04**	**142.56**	**149.34**	**138.67**	**192.47**	**168.50**	**150.13**
		PA-MPJPE	**90.39**	**92.40**	**98.06**	**64.29**	**78.50**	**83.84**	**80.49**	**81.06**	**112.05**	**101.02**	**88.21**
	Ours (Non-Causal)	MPJPE	**141.57**	**139.74**	**146.60**	**129.55**	**140.42**	**138.85**	**145.99**	**135.83**	**189.58**	**164.45**	**147.25**
		PA-MPJPE	**88.24**	**91.40**	**96.60**	**61.15**	**74.56**	**80.73**	**77.49**	**78.10**	**109.65**	**98.21**	**85.61**

Table 14. **Per-joint quantitative comparison for EE3D-R and EE3D-W datasets.**

Dataset	Method	Metric ↓	Head	Neck	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Foot	**Avg.**
EE3D-R	EgoPoseFormer [44]	MPJPE	140.3	173.9	157.3	177.1	181.1	212.6	169.8	144.8	207.6	173.00
		PA-MPJPE	104.3	119.9	102.4	124.3	132.9	111.9	111.9	88.9	120.2	113.70
	EventEgo3D [25]	MPJPE	23.49	29.87	40.20	91.67	153.17	69.99	120.96	179.02	201.42	110.39
		PA-MPJPE	49.27	41.63	44.48	79.58	133.53	55.66	83.45	108.78	125.26	84.52
	EventEgo3D++ [26]	MPJPE	22.66	29.36	40.03	77.86	125.41	63.02	119.86	177.03	196.58	103.29
		PA-MPJPE	43.01	36.52	40.54	71.09	112.89	48.17	79.38	105.13	119.58	77.07
	Ours (Causal)	MPJPE	**22.40**	**28.43**	**39.12**	**68.75**	**102.01**	**60.35**	**92.31**	**135.29**	**152.37**	**84.45**
		PA-MPJPE	**34.79**	**29.57**	**36.70**	**59.08**	**85.87**	**44.28**	**64.16**	**82.27**	**96.64**	**62.65**
	Ours (Non-Causal)	MPJPE	**22.06**	**28.44**	**37.95**	**66.43**	**98.21**	**57.79**	**87.87**	**130.06**	**147.01**	**81.32**
		PA-MPJPE	**33.41**	**28.57**	**35.16**	**56.71**	**82.20**	**41.57**	**61.19**	**79.94**	**93.95**	**60.21**
EE3D-W	EgoPoseFormer [44]	MPJPE	200.10	210.15	198.70	220.13	215.29	190.40	202.70	230.08	225.25	210.50
		PA-MPJPE	130.50	140.20	128.92	135.42	139.70	120.62	125.11	140.22	138.0	133.20
	EventEgo3D [25]	MPJPE	46.54	61.64	82.57	145.81	228.19	161.96	239.96	315.86	335.56	195.50
		PA-MPJPE	69.16	55.87	65.48	102.11	184.56	76.49	94.26	134.92	145.26	108.20
	EventEgo3D++ [26]	MPJPE	44.87	56.67	74.02	127.95	185.59	136.72	215.69	285.08	303.66	172.43
		PA-MPJPE	59.46	49.49	59.01	90.39	157.96	64.35	93.59	128.17	139.41	98.42
	Ours (Causal)	MPJPE	**41.41**	**49.36**	**73.62**	**114.31**	**166.56**	**125.08**	**193.57**	**267.04**	**285.31**	**158.86**
		PA-MPJPE	**56.83**	**46.48**	**58.72**	**88.40**	**144.05**	**59.57**	**87.36**	**123.67**	**134.30**	**93.46**
	Ours (Non-Causal)	MPJPE	**43.74**	**51.32**	**73.01**	**112.67**	**163.55**	**122.79**	**189.22**	**259.93**	**277.88**	**155.82**
		PA-MPJPE	**55.95**	**45.28**	**57.46**	**85.13**	**138.42**	**58.55**	**85.02**	**121.06**	**130.62**	**90.86**

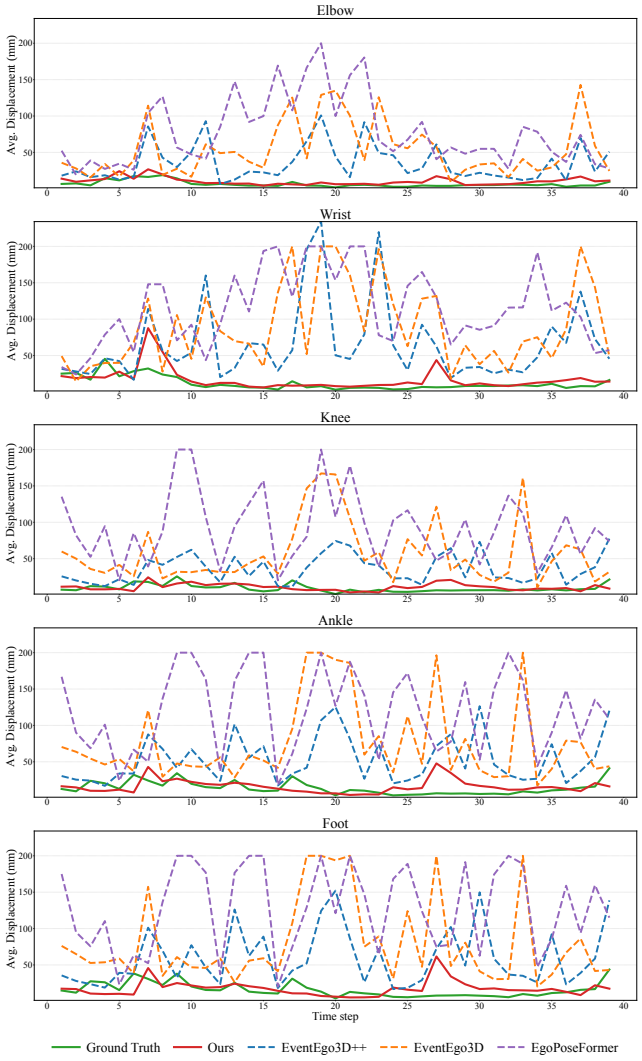

Figure 11. The per-frame average end-effector joint displacements (Eq. (24)) for EE3D-R. Zoom recommended.

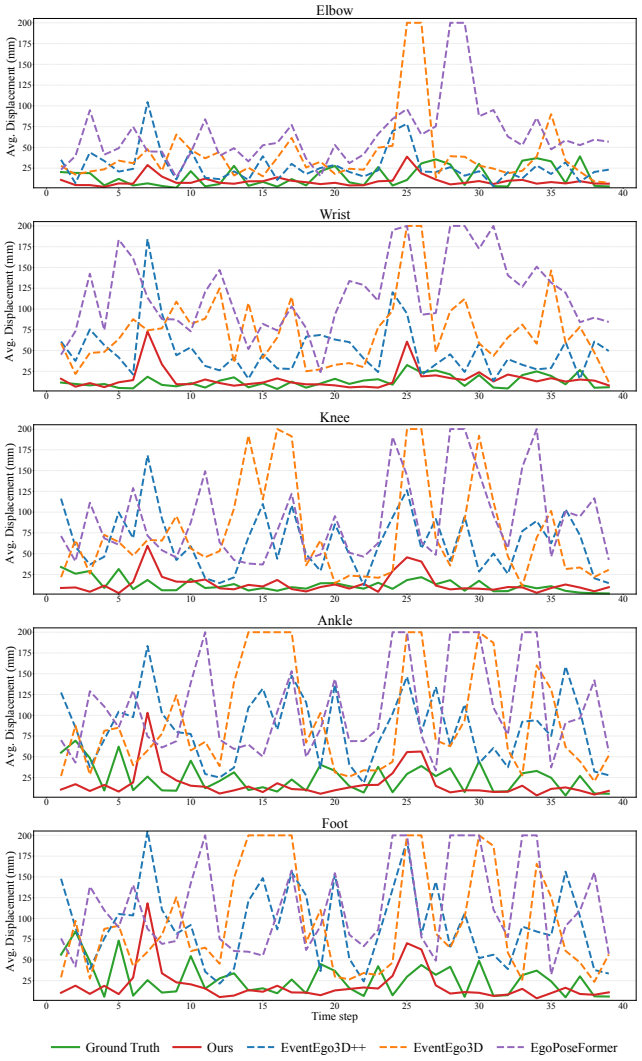

Figure 12. The per-frame average end-effector joint displacements (Eq. (24)) for EE3D-W. Zoom recommended.

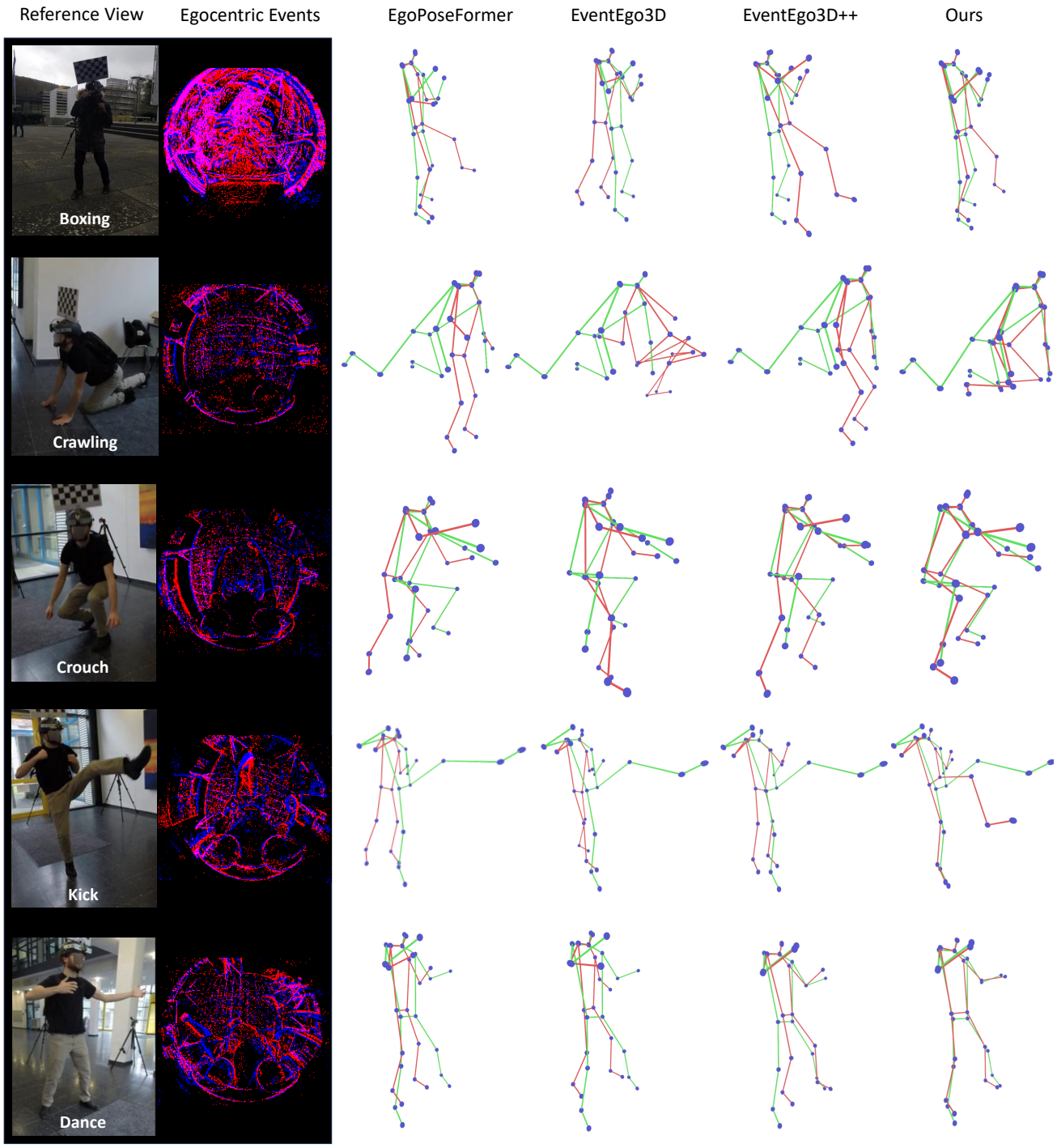

Figure 13. **Per-action qualitative comparison of our method with prior approaches on EE3D-W (challenging sequences).** We compare against EgoPoseFormer [44], EventEgo3D [25], and EventEgo3D++ [26]. **Red:** Predicted pose. **Green:** Ground truth.

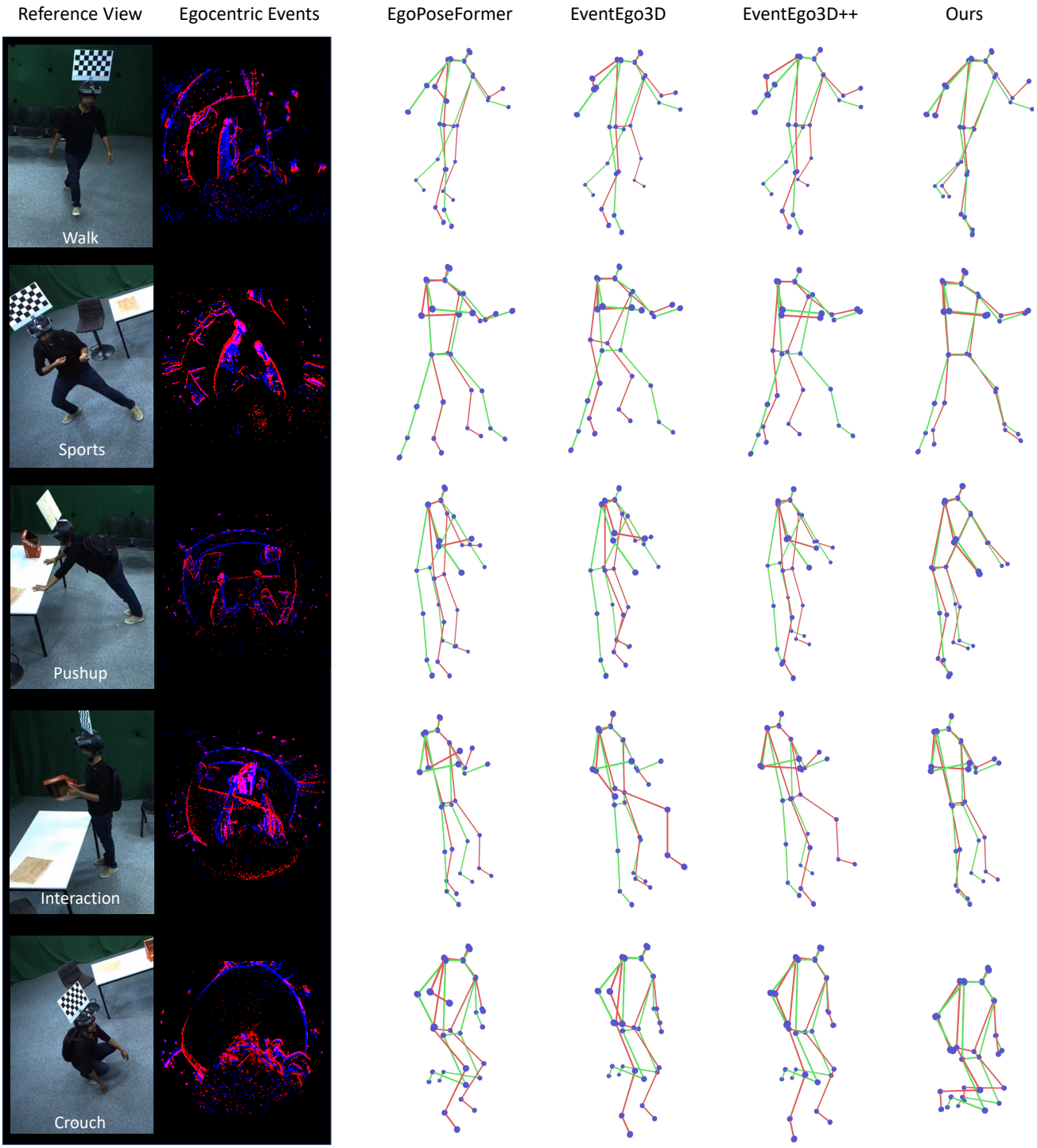

Figure 14. **Per-action qualitative comparison of our method with prior approaches on EE3D-R (walk and further challenging sequences).** We compare against EgoPoseFormer [44], EventEgo3D [25], and EventEgo3D++ [26]. **Red:** Predicted pose. **Green:** Ground truth.