

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

Mayur Deshmukh^{1,2} Hiroyasu Akada¹ Helge Rhodin^{1,3} Christian Theobalt¹ Vladislav Golyanik¹

¹MPI for Informatics, SIC ²Saarland University, SIC ³Bielefeld University

Abstract

Event cameras offer multiple advantages in monocular egocentric 3D human pose estimation from head-mounted devices, such as millisecond temporal resolution, high dynamic range, and negligible motion blur. Existing methods effectively leverage these properties, but suffer from low 3D estimation accuracy, insufficient in many applications (e.g., immersive VR/AR). This is due to the design not being fully tailored towards event streams (e.g., their asynchronous and continuous nature), leading to high sensitivity to self-occlusions and temporal jitter in the estimates. This paper rethinks the setting and introduces E-3DPSM, an event-driven continuous pose state machine for event-based egocentric 3D human pose estimation. E-3DPSM aligns continuous human motion with fine-grained event dynamics; it evolves latent states and predicts continuous changes in 3D joint positions associated with observed events, which are fused with direct 3D human pose predictions, leading to stable and drift-free final 3D pose reconstructions. E-3DPSM runs in real-time at 80 Hz on a single workstation and sets a new state of the art in experiments on two benchmarks, improving accuracy by up to 19% (MPJPE) and temporal stability by up to 2.7 $\times$. The source code and trained models are available on our project page¹.

1. Introduction

Egocentric 3D human pose estimation from head-mounted devices (HMDs) is a key capability for immersive VR/AR applications such as real-time avatar control, fitness tracking, telepresence, and hands-free interfaces. By capturing motion directly from the wearer’s perspective, it removes the need for external cameras and constrained capture environments. Yet, fast camera motion and frequent self-occlusions also add new algorithmic challenges.

Recent RGB-based egocentric 3D pose estimation methods [1–3, 14, 16, 17, 23, 24, 27, 29, 35, 36, 38–40, 44, 45,

¹<https://4dqv.mpi-inf.mpg.de/E-3DPSM/>

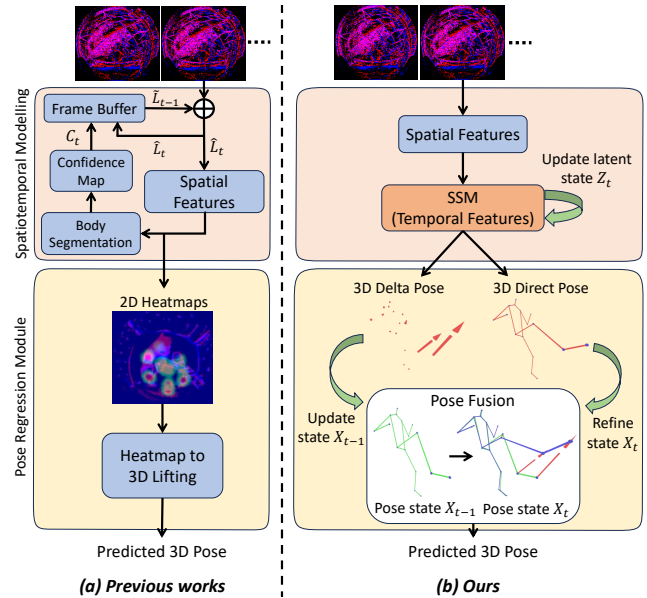


Figure 1. **Rethinking event-based egocentric 3D human pose estimation.** (a) Previous methods [25, 26] capture temporal information only through a single previous event frame stored in the frame buffer leading to jitter and drift. (b) Our E-3DPSM approach models motion as a continuous event-driven state evolution, fusing delta and direct 3D human pose updates, thereby achieving real-time and temporally stable 3D reconstruction and significantly outperforming prior approaches in the 3D accuracy.

47] achieve accurate results in controlled, well-lit environments, but struggle under real-world conditions. Low light causes underexposure and sensor noise, rapid head motion leads to blur, and continuous streaming of high-resolution video imposes heavy bandwidth and power demands on wearable devices. As a viable alternative in challenging conditions, event cameras offer millisecond-level temporal resolution and high dynamic range.

The recently introduced EventEgo3D [25, 26] is the first method for egocentric and real-time 3D human pose estimation from a single head-mounted event camera. Compared to third-person captures [41], humans remain centred and scale-consistent across input frames in egocentric

views. These properties reduce background variability and make egocentric event streams particularly suitable for continuous 3D reconstruction of fast human motions. While EventEgo3D can capture 3D human poses at high temporal resolutions, its overall accuracy is considered low in many practical scenarios. We believe the reasons are in its architecture (Fig. 1-(a)): First, EventEgo3D predicts direct 3D poses, models temporal information using only the previous event frame stored in the frame buffer through short-term feature propagation across event frames and, hence, does not fully exploit the asynchronous, continuous, and change-driven nature of event data. This often leads to jitter, drift, and substantial 3D errors under self-occlusions. Moreover, the reliance on 2D heatmaps introduces quantisation errors [46], and segmentation masks that need to be predicted at test time can serve as an additional source of inaccuracies.

This paper fundamentally rethinks egocentric event-based 3D human pose estimation with the overarching goal of improving the 3D estimation accuracy. Our key insights are threefold: 1) Since events inherently encode changes in the 2D observation space, they should correspond to and induce changes in the 3D space. The changes in the relative 3D joint locations, which we refer to as delta poses, are accumulated into smooth and robust 3D trajectories and fused with direct 3D predictions. This principle is visualised in Fig. 1-(b) and can be compared with the simultaneous prediction of 3D locations and sparse 3D scene flow. 2) Second—as events observe changes in the 2D space continuously (in the sense that their temporal resolution is substantially higher than what is required to capture the fastest human motions)—we formulate the prediction task as a continuous process, where 3D motion continuously evolves in response to asynchronous events. We demonstrate that state space modelling [9, 50] is naturally suitable for continuous prediction of 3D human poses using event streams. 3) Additional supervision with object-background segmentation masks and intermediate 2D heatmaps inherited from RGB-based vision [25, 26] can be avoided. A neural architecture can learn to extract all intermediate features and representations necessary for 3D human pose estimation with a reduced set of pre-defined design choices.

All these insights are reflected in our new approach, *Event-based 3D Pose State Machine (E-3DPSM)*, which treats egocentric event-based 3D human pose estimation as a continuous temporal process. *E-3DPSM* maintains a latent state that evolves with the observed motion. Each incoming set of events (LNEs [31]) updates this internal state, which encodes the current pose, its uncertainty, and the learned spatiotemporal context. Predicted 3D delta changes correspond to event-level variations and (along with the evolving latent state, accumulating motion cues over time) produce temporally consistent 3D poses. To further mitigate drift and temporally stabilise 3D reconstruction, we

introduce a learnable Kalman-filter-inspired fusion module that adaptively integrates global and delta predictions. In the absence of events, the latent state retains the last estimate, eliminating the need for explicit caching mechanisms used in the previous approaches [25, 26]. In summary, the technical contributions of this paper are as follows:

- *E-3DPSM*, the first state machine architecture that maintains a continuous 3D human pose state aligned with the asynchronous dynamics of event streams, enabling real-time tracking at 80Hz on our hardware (Sec. 4);
- A spatiotemporal pose encoder that fuses spatial motion cues with long-range temporal dependencies, leveraging deformable attention and bidirectional state-space modelling to remain robust under occlusion and rapid motion;
- A learnable neural module that dynamically balances direct and delta predictions, mitigating drift and ensuring smooth trajectories even under sparse and noisy events.

E-3DPSM achieves state-of-the-art results on two egocentric event benchmarks, reducing MPJPE and PA-MPJPE by $\sim 19\%$, and jitter by up to $2.7\times$ compared to the previous state of the art [25, 26] (Sec. 5).

2. Related Work

Egocentric 3D Human Pose Estimation. Egocentric 3D human pose estimation has gained traction with the rise of VR/AR applications requiring full-body 3D motion recovery from head-mounted cameras. Early works such as EgoCap [29] used a stereo fisheye capture setup and global pose estimation using off-the-shelf SLAM, while xR-EgoPose [35] and SelfPose [36] handled monocular settings. Later methods addressed calibration and improved 3D global pose estimation from HMD views [38, 39] as well as whole-body pose refinement with diffusion models [40]. To study a variety of settings with challenging human poses and self-occlusions, UnrealEgo [1] and UnrealEgo2 [2] projects introduced large-scale synthetic datasets and stereo architectures, with algorithmic improvements shown by Ego3DPose [17] (a two-path network) and EgoTap [16] (a grid ViT). Further works explored rear-mounted cameras [3], fisheye-based self-supervision [23], motion priors, reinforcement learning, and spatio-temporal transformers [14, 24, 27, 45, 47]. Despite this progress, RGB-based methods remain reliant on frame-based processing and often suffer from issues such as blur, low-light sensitivity, and high bandwidth demands. In contrast, event cameras offer high temporal resolution and robustness in challenging observation conditions, but require fundamentally different (and often new) neural architecture designs.

Event-Based 3D Human Pose Estimation. The first event-based methods for 3D human pose estimation used exocentric views. EventCap [43] demonstrated high-speed monocular 3D motion capture by tracking a 3D human body template using a monocular event stream in a series of optimi-

sation problems, and EventHPE [49] jointly estimated 3D poses and shapes from events using a two-stage neural network trained on the DHP19 dataset [5]. EventEgo3D [25] and EventEgo3D++ [26] introduced egocentric event-based 3D human pose estimation. Their architecture includes two branches, i.e., 1) an encoder-decoder to convert events into 2D heat maps and lift them to 3D poses, and 2) an event stream segmentation module to filter out background events. This method predominantly adopts components from RGB-based methods (i.e., not specifically tailored to event streams) and does not reach the accuracy required in many applications with HMDs (suffers from temporal jitter). In contrast, we rethink the problem and tailor all design choices to the egocentric event-based setting.

The works by Gehrig et al. [8] and Zubic et al. [50] proposed a recurrent ViT and a state-space model (SSM) for event-based object detection, while PRE-Mamba [30] introduced an SSM for event stream deraining. Lang et al. [21] used a Mamba-type SSM for context-aware fusion of RGB and event features in 3D human pose estimation. SSMs are also widely used in problems with RGB and point cloud inputs [22]. Inspired by all these insights, we bring SSMs to the realm of event-based dynamic 3D vision and formulate egocentric 3D human pose estimation as a continuous dynamic process. By adopting a neural SSM, we explicitly model temporal evolution and inter-frame pose differences, leading to substantially improved accuracy and temporal consistency over prior event-based methods. This allows us to avoid event segmentation masks and intermediate 2D heatmaps [25, 26] as potential sources of inaccuracies.

3. Preliminaries

Event Cameras. Event cameras are bio-inspired sensors that record per-pixel brightness changes asynchronously instead of capturing full image frames at fixed rates. A pixel (x, y) generates an event $e = (x, y, t, p)$ whenever the change in log intensity I exceeds a threshold C , with polarity $p \in \{-1, 1\}$ denoting the sign of the change:

$$\Delta I(x, y, t) = |I(x, y, t) - I(x, y, t - \Delta t)| \geq C. \quad (1)$$

This results in sparse and high-temporal-resolution data, which is well-suited for fast motion and dynamic scenes.

LNES Representation. Given a set of events $e_i = (x_i, y_i, t_i, p_i)$ collected within a time window of T ms, we construct a Locally Normalised Event Surface (LNES) [31] $\mathbf{L} \in \mathbb{R}^{192 \times 256 \times 2}$, with two separate channels for positive and negative polarities. Each event is normalised relative to the start time t_0 of the temporal window, i.e.,

$$\mathbf{L}(x_i, y_i, p_i) = \frac{t_i - t_0}{T}, \quad (2)$$

so that recent events map to values close to 1.0 and older ones decay toward 0.0. This encodes both the spatial loca-

tion and temporal freshness of events, yielding a dense 2D grid compatible with standard neural architectures.

State-Space Models (SSMs) provide a structured way to model long-range temporal dependencies in sequential data. An SSM maintains a latent state $\mathbf{Z}_t \in \mathbb{R}^d$ that evolves over time according to a linear recurrence, while mapping inputs to outputs via learnable projections:

$$\mathbf{Z}_{t+1} = \mathbf{A}\mathbf{Z}_t + \mathbf{B}x_t, \quad \mathbf{Y}_t = \mathbf{C}\mathbf{Z}_t, \quad (3)$$

where x_t is the input at timestep t , $\mathbf{Y}_t$ is the output, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are learned matrices. Unlike recurrent networks, SSMs use an equivalent convolutional form that enables efficient parallel training while preserving the ability to run causally at inference. Recent variants such as S4 [10] and S5 [34] layers stabilise training through spectral or band-limiting constraints, making them particularly effective for long-sequence modelling tasks. We use event-specific S5 layers, similar to Zubic et al. [50], to capture long-term temporal context in event streams as a 3D reconstruction cue.

4. The E-3DPSM Approach

E-3DPSM estimates temporally consistent 3D human poses from a monocular egocentric event camera equipped with a fisheye lens in three stages; see Fig. 2. First, the incoming raw event stream is converted into N LNES frames $\{\mathbf{L}_t\}_{t=1}^N$ of length T ms. Next, the Spatiotemporal Pose Encoder Module (SPEM, Sec. 4.1) extracts temporally aware, joint-specific features by combining a multi-stage convolutional pyramid, per-stage deformable attention for spatial reasoning, and event-specialised SSM (S5) layers for temporal modelling, followed by a joint-query transformer decoder that reads the deepest-stage tokens. Finally, the Pose Regression Module (PRM, Sec. 4.2) predicts a direct 3D pose $\mathbf{P}_t^D \in \mathbb{R}^{J \times 3}$ and a delta pose $\mathbf{P}_t^\Delta \in \mathbb{R}^{J \times 3}$, with the number of joints $J=16$. A lightweight learned fusion module combines $\mathbf{P}_{t-1}$ and $\mathbf{P}_t^\Delta$, while using $\mathbf{P}_t^D$ as a global anchor, yielding the final temporally consistent 3D joints $\mathbf{P}_t$.

4.1. Spatiotemporal Pose Encoder Module (SPEM)

SPEM transforms LNES event frames into rich, temporally-aware and joint-specific representations. It is composed of a multi-stage convolutional encoder, spatially adaptive deformable attention blocks, state-space model blocks for long-term temporal reasoning, and a joint-query transformer decoder. Fig. 3 provides details.

Convolutional Feature Encoding. Each input LNES frame $\mathbf{L}_t \in \mathbb{R}^{192 \times 256 \times 2}$ is first passed through a strided convolution, producing $\mathbf{F}_t^0 \in \mathbb{R}^{96 \times 128 \times 16}$. The encoder then processes the features through four hierarchical stages. At stages $s \in \{1, \dots, 4\}$, we apply two residual blocks and a downsampling convolution, and obtain

$$\mathbf{F}_t^s = \text{Conv}\left(\text{ResBlock}_s^{(2)}\left(\text{ResBlock}_s^{(1)}(\mathbf{F}_t^{s-1})\right)\right), \quad (4)$$

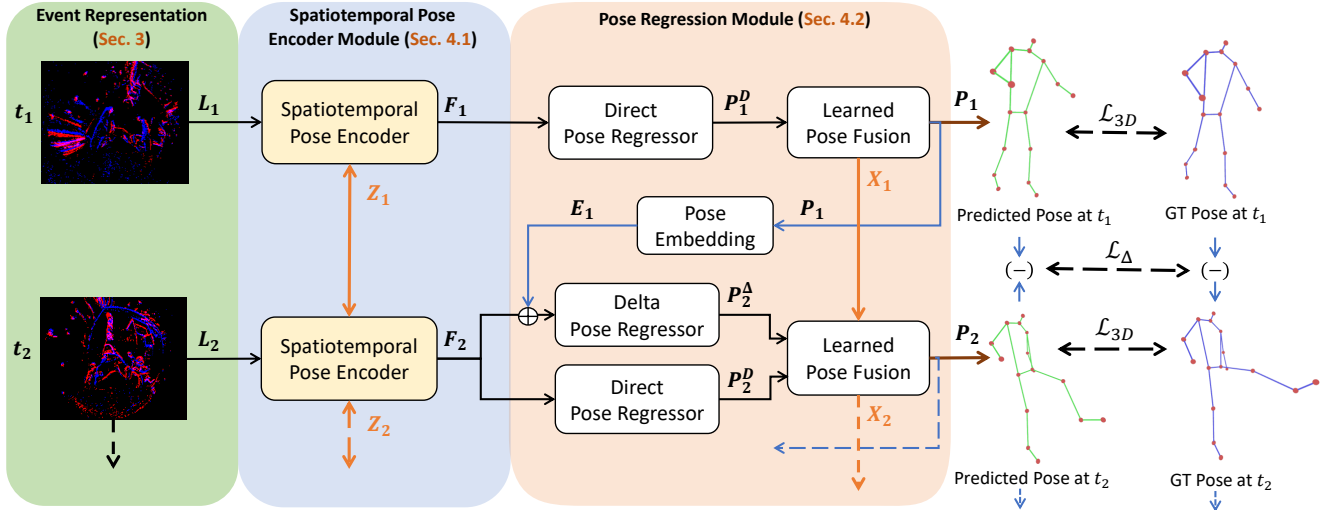


Figure 2. **Overview of the proposed E-3DPSPM approach for monocular egocentric 3D human pose estimation.** Incoming raw events e are converted into LNES frames L_t and processed by the Spatiotemporal Pose Encoder Module (SPEM, Sec. 4.1), as depicted in Fig. 3. The output of SPEM F_t is passed to the Pose Regression Module (PRM, Sec. 4.2), which estimates both the direct 3D poses P_t^D and the 3D delta poses P_t^Δ between consecutive LNES frames. These delta poses naturally correspond to the observed events (i.e., changes in the 2D LNES space). Finally, the 3D pose deltas are fused with the direct 3D pose using a learned pose fusion mechanism (a neural Kalman-style filter) to obtain the final 3D poses P_t . The predicted 3D pose P_t is supervised using absolute ground-truth 3D poses ($\mathcal{L}_{3D}$) and ground-truth 3D pose differences ($\mathcal{L}_\Delta$), respectively. At each timestep t , SPEM updates and propagates its latent state Z_t bidirectionally (the “ $\longleftrightarrow$ ” symbol), whereas PRM propagates its fused pose state X_t causally (the “ $\longrightarrow$ ” symbol) to the next timestep. During training, SPEM operates bidirectionally (non-causal) to exploit the full temporal context, while at inference, it can run in either causal (real-time) or non-causal mode. For visualisation purposes, the differences between P_1 and P_2 and the corresponding L_1 and L_2 are exaggerated.

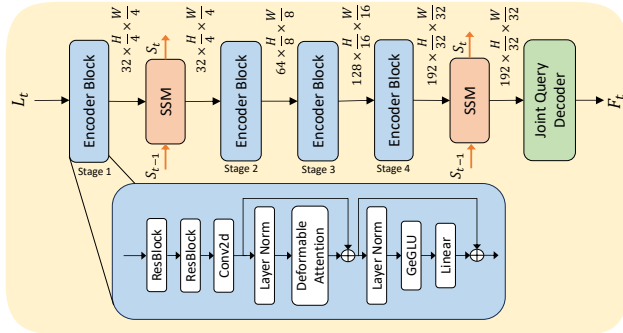


Figure 3. **Architecture of SPEM**, combining multi-stage convolutional encoding, SSM blocks, deformable attention, and a joint-query decoder for temporally-aware pose features.

where “Conv” denotes a 3×3 convolution with stride 2 that reduces spatial resolution, and each ResBlock [12] is a two-convolution residual unit with BatchNorm and SiLU [6].

Deformable Attention for Spatial Reasoning. Inspired by recent egocentric pose estimation methods [2, 44] that use deformable attention [48] to adaptively focus on pose-critical regions under occlusions and motion, we incorporate a deformable attention block at the end of each stage to refine features at every timestep. We first flatten the feature map $F_t^s \in \mathbb{R}^{H^s \times W^s \times C^s}$ into a sequence of feature tokens $T_t^s \in \mathbb{R}^{(H^s W^s) \times C^s}$, and then compute

$$F_t^s = \text{DeformAttn}(T_t^s, T_t^s, R_s) \in \mathbb{R}^{(H^s W^s) \times C^s}, \quad (5)$$

where H^s , W^s , and C^s are stage-dependent. The reference points R_s are initialised on a normalised uniform grid and optimised end-to-end. In this setup, T_t^s acts as query, key, and value, so that each token attends to itself and its neighbours through learned deformable offsets. Deformable attention shifts sampling toward joint-critical neighbourhoods, allowing tokens to attend to relevant body parts under strong egocentric (fisheye lens) distortions.

SSM for Bidirectional Temporal Modelling. While spatial reasoning is handled within each frame, modelling motion over time is critical, especially under occlusions. For this, we insert S5 layers [50], a recent state-space model (SSM) variant specialised for event streams, at selected encoder stages $\hat{s} \in \{2, 4\}$ to aggregate long-range temporal context independently at every spatial location $(H^{\hat{s}}, W^{\hat{s}})$. Let $F_{1:N}^{\hat{s}} = [F_1^{\hat{s}}, \dots, F_N^{\hat{s}}] \in \mathbb{R}^{(H^{\hat{s}} W^{\hat{s}}) \times N \times C^{\hat{s}}}$ denote the per-location feature sequence at stage $\hat{s}$. The SSM layer transforms this sequence and returns temporally refined features $\tilde{F}$ with an internal state Z as follows:

$$\tilde{F}_{1:N}^{\hat{s}}, \text{ and } Z_t^{\hat{s}} = \text{SSM}_{\hat{s}}(F_{1:N}^{\hat{s}}). \quad (6)$$

We use the band-limited S5 variant [50] (bandlimit set as 0.5), placed after the downsampling stage. During training, SSM layers are evaluated in parallel over full sequences using their convolutional form, and we run them bidirectionally to expose past and future context. At inference, we can

switch to causal forward-only updates with recurrent state propagation by carrying internal state $\mathbf{Z}_s$ across timesteps, which enables real-time deployment without future frames.

Joint Query Decoder. After temporal modelling, we extract joint-specific features using a lightweight transformer decoder layer [37]. We follow existing works [2, 44] and define a set of J learnable joint query embeddings $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_{16}\} \subset \mathbb{R}^{192}$. These queries act as joint-identity tokens, allowing the decoder to consistently associate each query with a specific 3D body joint across timesteps. At each timestep t , we flatten the output from the encoder’s last stage $\tilde{\mathbf{F}}_t^4 \in \mathbb{R}^{6 \times 8 \times 192}$ to memory tokens $\mathbf{M}_t \in \mathbb{R}^{48 \times 192}$, and decode joint-aware features using:

$$\mathbf{F}_t = \text{TransformerDecoder}(\mathbf{U}, \mathbf{M}_t) \in \mathbb{R}^{16 \times 192}. \quad (7)$$

The decoder attends each joint query to spatial memory, enabling it to learn both joint appearance and contextual interactions between joints (e.g., elbow-wrist alignment). Subsequently, these final representations $\mathbf{F}_t \in \mathbb{R}^{16 \times 192}$ are passed to PRM (Sec. 4.2) for pose prediction.

4.2. Pose Regression Module (PRM)

Following joint-aware spatiotemporal encoding, our PRM estimates temporally consistent 3D joint positions across all LNES frames. It consists of three components: 1) a direct pose regressor for initial predictions, 2) a delta pose regressor to track motion across time, and 3) a learnable module that adaptively fuses both estimates into a unified 3D pose.

Direct Pose Regression. At each timestep t , we apply a lightweight MLP head to each joint query token output by the transformer decoder $\mathbf{F}_t \in \mathbb{R}^{16 \times 192}$ to predict the 3D position of each joint directly:

$$\mathbf{P}_t^D = \text{MLP}_{\text{Direct}}(\mathbf{F}_t) \in \mathbb{R}^{16 \times 3}. \quad (8)$$

The prediction at $t = 1$ initialises our fusion module, and subsequent estimates act as an anchor used to mitigate drift when necessary, as determined adaptively by the fusion module. This branch serves as an intermediate prediction for stabilisation rather than a standalone regression target.

Delta Pose Regression. To regress the relative offset between the current and previous frames $\mathbf{L}_t$ and $\mathbf{L}_{t-1}$, we introduce a delta pose regressor. At each subsequent timestep $t > 1$, we concatenate the current joint token $\mathbf{F}_t \in \mathbb{R}^{16 \times 192}$ with an embedding $\mathbf{E}_{t-1}$ of the previous pose, and then apply a lightweight MLP head:

$$\mathbf{E}_{t-1} = \text{MLP}_{\text{pose-emb}}(\mathbf{P}_{t-1}) \in \mathbb{R}^{16 \times 64}, \text{ and} \quad (9)$$

$$\mathbf{P}_t^\Delta = \text{MLP}_\Delta([\mathbf{F}_t; \mathbf{E}_{t-1}]) \in \mathbb{R}^{16 \times 3}. \quad (10)$$

Since event streams encode changes rather than absolute intensities, predicting relative 3D joint displacements is often more aligned with the input modality and forms an easier regression target than absolute 3D positions. The deltas

capture short-term motion cues that remain structured over time and are, therefore, especially helpful in situations with high-speed motion and occlusions, where absolute positions can fluctuate more significantly.

Learned Pose Fusion. To obtain the final pose for $t > 1$, a naive approach would be a simple addition of the direct pose from the previous timestep with the current delta pose:

$$\mathbf{P}_t = \mathbf{P}_{t-1}^D + \mathbf{P}_t^\Delta. \quad (11)$$

This would lead to error accumulation over time, especially when delta estimates are noisy or when direct predictions suffer from transient uncertainty (see App. B). To mitigate this adaptively, we introduce a learnable fusion module, implemented as a differentiable Kalman-style filter [11, 15, 20]. This module adaptively weighs the current delta update against the predicted direct pose to produce a more stable and accurate final pose. At each timestep $t > 1$, the module receives: 1) $\mathbf{P}_t^\Delta \in \mathbb{R}^{48 \times 1}$ as a motion update, and 2) $\mathbf{P}_t^D \in \mathbb{R}^{48 \times 1}$ as an observation. We treat this process as a fusion problem, i.e., we fuse these two measurements to compute a corrected 3D pose estimate where we maintain a latent internal state $\mathbf{X}_t \in \mathbb{R}^{48 \times 1}$ corresponding to the current estimate of the full pose, and a covariance matrix $\Sigma_t \in \mathbb{R}^{48 \times 48}$ encoding the uncertainty in the estimate.

1. Motion Update (Prediction Step): First, we predict the new internal state based on the previous state $\mathbf{X}_{t-1}$ and $\mathbf{P}_t^\Delta$:

$$\mathbf{X}_t = \mathbf{A} \cdot \mathbf{X}_{t-1} + \mathbf{B} \cdot \mathbf{P}_t^\Delta. \quad (12)$$

Here, $\mathbf{A} \in \mathbb{R}^{48 \times 48}$ is a state transition matrix modelling temporal dynamics, and $\mathbf{B} \in \mathbb{R}^{48 \times 48}$ modulates the effect of $\mathbf{P}_t^\Delta$. We also update the prediction uncertainty:

$$\Sigma_{t|t-1} = \mathbf{A} \cdot \Sigma_{t-1} \cdot \mathbf{A}^\top + \mathbf{Q}, \quad (13)$$

where $\mathbf{Q} \in \mathbb{R}^{48 \times 48}$ is the learned process noise covariance, capturing uncertainty in the motion model and delta update.

2. Measurement Update (Correction Step): Next, we incorporate the direct pose observation $\mathbf{P}_t^D$ to refine the predicted state. We compute the Kalman gain $\mathbf{K}_t \in \mathbb{R}^{48 \times 48}$, which balances the confidence between the motion prediction and the observation:

$$\mathbf{K}_t = \Sigma_{t|t-1} \cdot \mathbf{H}^\top \cdot (\mathbf{H} \cdot \Sigma_{t|t-1} \cdot \mathbf{H}^\top + \mathbf{R})^{-1}. \quad (14)$$

Here, $\mathbf{H} \in \mathbb{R}^{48 \times 48}$ is the observation matrix (identity in our case), and $\mathbf{R} \in \mathbb{R}^{48 \times 48}$ is the learned observation noise covariance, representing uncertainty in the direct pose prediction. We then correct the internal state using the residual, i.e., difference between observation and prediction:

$$\mathbf{P}_t = \mathbf{X}_t + \mathbf{K}_t \cdot (\mathbf{P}_t^D - \mathbf{H} \cdot \mathbf{X}_t). \quad (15)$$

To reflect the reduced uncertainty after fusion, we update the state covariance using the Joseph form [4], which provides a numerically stable update and guarantees the positive semi-definiteness of the resulting covariance matrix:

$$\Sigma_t = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}) \cdot \Sigma_{t|t-1} \cdot (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H})^\top + \mathbf{K}_t \cdot \mathbf{R} \cdot \mathbf{K}_t^\top. \quad (16)$$

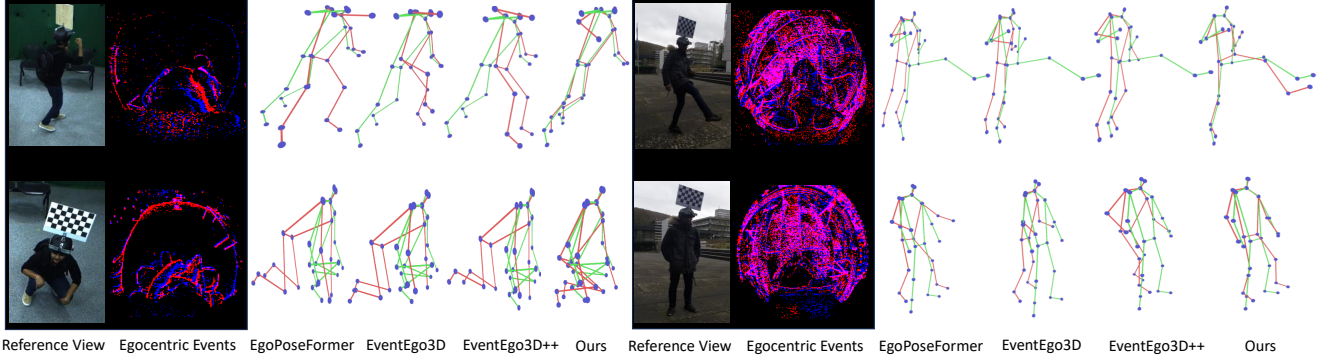


Figure 4. **Qualitative comparison of our method with prior approaches.** We compare against EgoPoseFormer [44], EventEgo3D [25], and EventEgo3D++ [26]. **Left:** EE3D-R (real dataset). **Right:** EE3D-W (in-the-wild). **Red:** Predicted pose. **Green:** Ground truth.

Table 1. **Quantitative comparisons on EE3D-R and EE3D-W.**

EE3D-R	MPJPE ↓	PA-MPJPE ↓	$e_{\text{smooth}} \downarrow$
EgoPoseFormer [44]	151.66	96.99	66.50
EventEgo3D [25]	110.39	84.52	27.06
EventEgo3D++ [26]	103.28	77.06	22.93
Ours (Causal)	84.45	62.64	8.40
Ours (Non-Causal)	81.32	60.21	6.65
EE3D-W	MPJPE ↓	PA-MPJPE ↓	$e_{\text{smooth}} \downarrow$
EgoPoseFormer [44]	220.40	130.45	79.23
EventEgo3D [25]	195.50	108.20	45.29
EventEgo3D++ [26]	172.43	98.41	40.87
Ours (Causal)	158.86	93.46	23.57
Ours (Non-Causal)	155.82	90.85	22.65

In practice, $\mathbf{A}, \mathbf{B}, \mathbf{H}$ are fixed as identities, while $\mathbf{Q}$ and $\mathbf{R}$ are learned once during training and remain constant at inference, i.e., they are not frame-dependent but shared across both the motion and measurement updates. Rather than relying on fixed noise assumptions, we learn both the process uncertainty (associated with the model update) and the observation uncertainty (associated with the direct pose prediction) in an end-to-end manner. Even though $\mathbf{Q}$ and $\mathbf{R}$ are constant at inference, learning them end-to-end provides the fusion module with calibrated priors on how much to trust delta pose changes versus direct pose estimates, thereby reducing drift and improving stability under occlusions (see Tab. 2). Rather than generic temporal smoothing, it performs reliability-aware integration of direct and delta predictions, akin to sensor fusion in classical filtering, which yields more accurate and stable 3D poses.

4.3. Loss Functions

To supervise our model, we employ a multi-term loss function that balances absolute accuracy, temporal coherence, anatomical plausibility, and projection consistency:

$$\mathcal{L}_{\text{total}} = \lambda_{3D}\mathcal{L}_{3D} + \lambda_{\Delta}\mathcal{L}_{\Delta} + \lambda_{2D}\mathcal{L}_{2D} + \lambda_{BL}\mathcal{L}_{BL} + \lambda_{BA}\mathcal{L}_{BA}. \quad (17)$$

In our experiments, we set the loss weights as follows: $\lambda_{3D}=\lambda_{\Delta}=\lambda_{2D}=0.01$ and $\lambda_{BL}=\lambda_{BA}=10^{-3}$.

Delta Pose Loss ($\mathcal{L}_{\Delta}$). To help the model in capturing fine-grained temporal dynamics, we supervise delta pose predictions with ground-truth inter-frame joint displacements:

$$\mathcal{L}_{\Delta} = \frac{1}{(N-1)J} \sum_{t=2}^N \sum_{j=1}^J \left\| \mathbf{P}_{t,j}^{\Delta} - \mathbf{P}_{t,j}^{\Delta^{\text{gt}}} \right\|^2, \quad (18)$$

where $\mathbf{P}_t^{\Delta} = \mathbf{P}_t - \mathbf{P}_{t-1}$. This encourages the model to learn consistent frame-to-frame motion and supports our temporal fusion strategy.

3D ($\mathcal{L}_{3D}$) and 2D ($\mathcal{L}_{2D}$) Pose Losses. We use mean-squared error on 3D joints and their 2D projections obtained with operator $\Pi(\cdot)$ [32]:

$$\mathcal{L}_{*} = \frac{1}{NJ} \sum_{t=1}^N \sum_{j=1}^J \left\| \hat{\mathbf{P}}_{t,j}^{\text{pred}} - \hat{\mathbf{P}}_{t,j}^{\text{gt}} \right\|^2, \quad (19)$$

where $\hat{\mathbf{P}}$ is the supervision target defined as

$$\hat{\mathbf{P}} = \begin{cases} \mathbf{P}_{t,j}, & \text{for } \mathcal{L}_{3D}, \\ \Pi(\mathbf{P}_{t,j}), & \text{for } \mathcal{L}_{2D}. \end{cases}$$

Bone Length Loss ($\mathcal{L}_{BL}$). To preserve human body proportions, we compute an L1-loss on the predicted and ground-truth bone lengths. For each bone pair $(i, j) \in \mathcal{B}$ of kinematic structure [13], the bone vectors are as follows:

$$\mathbf{b}_t^{(i,j)} = \mathbf{P}_{t,i}^{\text{pred}} - \mathbf{P}_{t,j}^{\text{pred}}, \quad \tilde{\mathbf{b}}_t^{(i,j)} = \mathbf{P}_{t,i}^{\text{gt}} - \mathbf{P}_{t,j}^{\text{gt}}; \text{ and } \quad (20)$$

$$\mathcal{L}_{BL} = \frac{1}{N|\mathcal{B}|} \sum_{t=1}^N \sum_{(i,j) \in \mathcal{B}} \left| \left\| \mathbf{b}_t^{(i,j)} \right\|_2 - \left\| \tilde{\mathbf{b}}_t^{(i,j)} \right\|_2 \right|, \quad (21)$$

which stabilises joint distances, especially under occlusion.

Bone Orientation Loss ($\mathcal{L}_{BA}$). To maintain anatomically plausible limb directions, we minimise the cosine distance

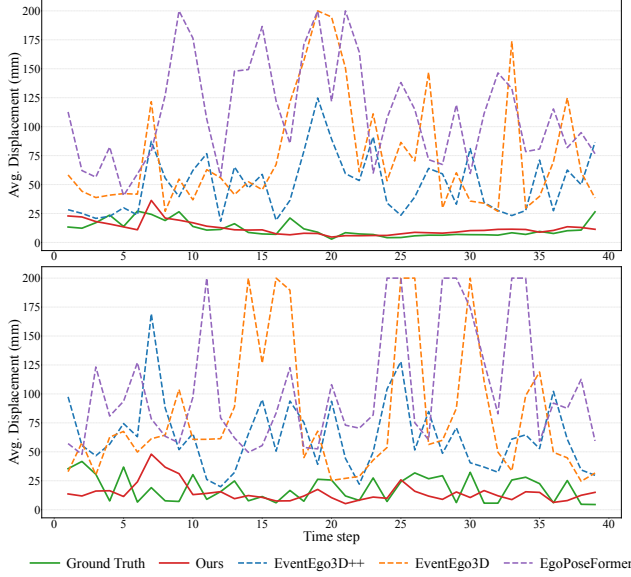


Figure 5. We plot the per-frame all-joint average displacement (Eq. (24)) for EE3D-R (top) and EE3D-W (bottom).

between predicted and ground-truth bone vectors:

$$\mathcal{L}_{\text{BA}} = \frac{1}{N|\mathcal{B}|} \sum_{t=1}^N \sum_{(i,j) \in \mathcal{B}} \left(1 - \frac{\mathbf{b}_t^{(i,j)} \cdot \tilde{\mathbf{b}}_t^{(i,j)}}{\|\mathbf{b}_t^{(i,j)}\|_2 \|\tilde{\mathbf{b}}_t^{(i,j)}\|_2} \right). \quad (22)$$

This complements the bone length loss by regularising the angular configurations of limbs.

5. Experiments

We conduct extensive experiments on the event-based ego-centric benchmarks, demonstrating the accuracy and robustness of *E-3DPSM* (Sec. 5.2). We also perform ablations to analyse the influence of each component (Sec. 5.3).

5.1. Experimental Setting

Datasets. We use two datasets: EE3D-R [25], a real-world dataset captured in a laboratory, and EE3D-W [26], a real-world in-the-wild dataset. We follow the official data splits. See App. A for data pre-processing details.

Evaluation Metrics. We report Mean Per Joint Position Error (MPJPE) and Procrustes-Aligned MPJPE (PA-MPJPE) [18] (both in mm), consistent with our baselines. To measure temporal stability, we use e_{smooth} [33] (in mm), which compares per-frame joint displacement magnitudes:

$$e_{\text{smooth}} = \frac{1}{(N-1)J} \sum_{t=2}^N \sum_{j=1}^J \left| \mathbf{d}_{t,j}^{\text{gt}} - \mathbf{d}_{t,j}^{\text{pred}} \right|, \quad \text{with} \quad (23)$$

$$\mathbf{d}_{t,j} = \|\mathbf{P}_{t,j} - \mathbf{P}_{t-1,j}\|_2. \quad (24)$$

Baselines and Evaluation Protocol. We compare against EventEgo3D [25], EventEgo3D++ [26], and the recent

Table 2. **Occlusion-only quantitative comparison on EE3D-R and EE3D-W.** Evaluation is performed only on occluded joints.

EE3D-R	MPJPE ↓	PA-MPJPE ↓
EgoPoseFormer [44]	110.18	55.64
EventEgo3D [25]	86.90	50.93
EventEgo3D++ [26]	88.43	49.53
Ours (Causal)	67.49	41.85
Ours (Non-Causal)	64.69	40.10
EE3D-W	MPJPE ↓	PA-MPJPE ↓
EgoPoseFormer [44]	173.01	113.67
EventEgo3D [25]	159.04	84.84
EventEgo3D++ [26]	147.42	77.64
Ours (Causal)	135.84	73.19
Ours (Non-Causal)	134.72	72.29

RGB-based state-of-the-art method EgoPoseFormer [44], in which the input layer is adjusted to process LNES frames. To ensure a fair comparison and report e_{smooth} , all baselines are evaluated on continuous test sequences². We do not reset internal states in our method, consistent with its continuous SSM formulation. Two inference strategies are considered: *causal*, where SPEM uses only the current and past frames, and *non-causal*, where it accesses all N frames.

Implementation and Training. We implement *E-3DPSM* in PyTorch [28] with the event-S5 layer [50]. Unlike previous works [25, 26], our model requires no pre-training on EE3D-S [25] (synthetic data). The ground-truth 3D pose at the most recent event in each window of $T=20$ ms is used for supervision, with $N=40$ poses. Smaller T improve the delta regressor’s sensitivity to fine-grained motion.

All modules are optimised end-to-end with Adam [19], with the batch size of 32. We train for 15 epochs on EE3D-R with a learning rate $\eta = 10^{-3}$, and then fine-tune for 10 epochs on EE3D-W with $\eta = 10^{-4}$. The training is non-causal, with the SPEM having access to the full N -frame sequence. It takes 34 hours on four A40 GPUs, while testing is performed on one A40 in the quantitative experiments. 3D pose update rate of 80Hz is reached on a single A6000, and a more lightweight NVIDIA 3050Ti supports 52Hz.

5.2. Main Results

Tab. 1 compares our approach with prior methods [25, 26, 44] on the EE3D-R [25] and EE3D-W [26] benchmarks. Across all three metrics, *E-3DPSM* in the causal mode (supporting real-time inference) consistently outperforms existing approaches, achieving 8% and 19% reduction in MPJPE on EE3D-W and EE3D-R, respectively, and between $1.7\times$ and $2.7\times$ lower e_{smooth} . The metrics are even slightly better for the non-causal mode. While non-causal inference attains the highest overall accuracy, the causal variant performs comparably, even though the model is trained non-causally, demonstrating strong generalisation and suitability for real-time deployment. Qualitative results in Fig. 4

²originally, the baselines use a random data loading strategy

Table 3. **Ablation study on the EE3D-R dataset** evaluating the impact of each component of our *E-3DPSM* approach.

SPEM (Sec. 4.1)	MPJPE ↓	PA-MPJPE ↓	$e_{\text{smooth}} \downarrow$
w/o SSM Blocks	118.53	87.48	16.94
Single SSM Block (Stage 4)	90.18	66.06	7.84
w/o Deformable Attention	88.27	64.98	7.71
PRM (Sec. 4.2)	MPJPE ↓	PA-MPJPE ↓	$e_{\text{smooth}} \downarrow$
w/o Fusion Module	141.22	84.17	10.14
Direct Pose Only	91.26	65.43	17.22
Static Fusion (Non-learned)	88.31	65.63	9.93
Ours (full model)	84.45	62.64	8.40

further highlight these improvements: While prior methods struggle with occluded regions, particularly lower-body joints (bottom-left)—and often fail even on simple poses such as standing (bottom-right) in EE3D-W—our predictions remain consistent and anatomically plausible. A detailed per-joint and per-action breakdown of these results is provided in App. D.2.

We attribute these improvements to the synergy between our delta-based motion regression and learnable fusion. PRM models 3D motion as deltas aligned with the event stream’s change-based nature, while the fusion module anchors global stability through adaptive integration with direct pose estimates. This design reduces drift, smooths trajectories, and lowers e_{smooth} , as visualised in the representative jitter plots for walk and kick motions (Fig. 5). Furthermore, the SPEM enhances robustness to self-occlusion through bidirectional SSM-based temporal modelling. The occlusion-only evaluation (Tab. 2) shows that our method achieves the lowest average error across all occluded joints, indicating improved robustness under partial or self-occlusion. A more detailed breakdown, including end-effector analysis and evaluation protocol, is provided in App. D.3.

Additionally, we compare our approach against Kalman-smoothed baselines (see App. D.1). Even when the results of prior methods are additionally post-processed with Kalman filtering, our model maintains significantly lower MPJPE and e_{smooth} , confirming that the improvements stem from our architectural design rather than Kalman smoothing. Overall, *E-3DPSM* consistently achieves state-of-the-art performance in event-based egocentric 3D pose estimation, combining high temporal consistency, robustness to occlusions, and practical real-time applicability.

5.3. Ablation Study

We next perform an ablation study by disabling or modifying individual modules on EE3D-R to quantify the roles of temporal modelling, spatial selectivity, and pose fusion design, as shown in Tab. 3.

SPEM. Removing all SSM blocks results in a purely spatial baseline, which severely degrades accuracy and smooth-

ness, while using a single SSM block (only at stage four) also degrades accuracy and smoothness, confirming the importance of early-stage temporal modelling. Disabling deformable attention further reduces performance, highlighting the need for spatial adaptivity in egocentric views.

PRM. Without the fusion module, simply adding the current delta pose to the previous pose according to Eq. (11) causes severe drift and the worst overall performance, confirming that corrective fusion is essential for the accuracy of the full model. Using only the direct pose regressor yields poor e_{smooth} , as fine motion encoded in deltas is ignored. A static Kalman-style fusion improves stability, but underperforms our adaptive fusion, demonstrating that task-specific, learnable noise weighting is crucial for robust integration of delta and direct 3D poses across diverse actions.

Our full framework achieves the best overall performance, validating the effectiveness of our design choices. We also perform additional ablations on the training strategy (App. E.1), the state reset strategy (App. E.2), the use of different event representations (App. E.3), and the inference event frequency (App. E.4).

6. Conclusion

Our real-time formulation of event-based egocentric 3D human pose estimation as a continuous state evolution problem demonstrated superior results in the extensive experiments compared to EventEgo3D++ [26]. While MPJPE and PA-MPJPE reduced by up to 19% in the causal prediction mode, the smoothness error improved by $2.7\times$ on EE3D-R and almost $1.7\times$ on EE3D-W. The improvements are slightly greater in the non-causal operation mode.

The largest accuracy gains have been observed for lower-body joints in occlusion-heavy actions such as crawling and crouching, where prior methods struggled and our *E-3DPSM* produced significantly more stable and anatomically consistent predictions. We attribute all these improvements to the avoidance of event segmentation maps and intermediate 2D heatmaps, as well as our design choices that are natural for and tailored to event streams and that were validated in the ablation study, i.e., bidirectional SSM (integrating long-range motion cues even when events were sparse or missing), deformable attention, learned fusion of delta and directly regressed 3D poses.

As event-based 3D human pose estimation is a nascent research field, *E-3DPSM* is not without limitations, such as sensitivity to strong occlusions and highly dynamic environments, as further discussed in App. H. All in all, we believe the design principles and insights gained in this work can benefit many other problems in event-based 3D vision.

Acknowledgment. This work was partially supported by the Nakajima Foundation scholarship.

References

- [1] Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik. Unrealego: A new dataset for robust egocentric 3d human motion capture. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2
- [2] Hiroyasu Akada, Jian Wang, Vladislav Golyanik, and Christian Theobalt. 3d human pose perception from egocentric stereo videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 4, 5
- [3] Hiroyasu Akada, Jian Wang, Vladislav Golyanik, and Christian Theobalt. Bring your rear cameras for egocentric 3d human pose estimation. In *International Conference on Computer Vision (ICCV)*, 2025. 1, 2
- [4] Richard S. Bucy and Peter D. Joseph. *Filtering for Stochastic Processes with Applications to Guidance*. AMS Chelsea Publishing, 2nd edition, 2005. 5
- [5] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. Dhp19: Dynamic vision sensor 3d human pose dataset. In *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 3
- [6] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 4
- [7] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *International Conference on Computer Vision (ICCV)*, 2019. 3
- [8] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [9] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state-space layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2
- [10] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022. 3
- [11] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: learning discriminative deterministic state estimators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 5
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 4
- [13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *Pattern Analysis and Machine Intelligence (PAMI)*, 36(7):1325–39, 2014. 6
- [14] Hao Jiang and Vamsi Krishna Ithapu. Egocentric Pose Estimation from Human Vision Span. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2
- [15] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *J. Fluids Eng.*, 82(1):35–45, 1960. 5
- [16] Taeho Kang and Youngki Lee. Attention-propagation network for egocentric heatmap to 3d pose lifting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2
- [17] Taeho Kang, Kyungjin Lee, Jinrui Zhang, and Youngki Lee. Ego3dpose: Capturing 3d cues from binocular egocentric views. In *SIGGRAPH Asia Conference Papers*, 2023. 1, 2
- [18] David G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989. 7
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 7
- [20] Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, 45(4):561–578, 2021. 5
- [21] Bo Lang and Mooi Choo Chuah. Event-guided fusion-mamba for context-aware 3d human pose estimation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 950–960, 2025. 3
- [22] Jiuming Liu, Jinru Han, Lihao Liu, Angelica I. Aviles-Rivero, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Mamba4d: Efficient 4d point cloud video understanding with disentangled spatial-temporal state space models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3
- [23] Yuxuan Liu, Jianxin Yang, Xiao Gu, Yijun Chen, Yao Guo, and Guang-Zhong Yang. Egofish3d: Egocentric 3d pose estimation from a fisheye camera via self-supervised learning. *IEEE Transactions on Multimedia (TMM)*, 2023. 1, 2
- [24] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2
- [25] Christen Millerdurai, Hiroyasu Akada, Jian Wang, Diogo Luvizon, Christian Theobalt, and Vladislav Golyanik. Eventego3d: 3d human motion capture from egocentric event streams. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 6, 7, 4, 9, 10
- [26] Christen Millerdurai, Hiroyasu Akada, Jian Wang, Diogo Luvizon, Alain Pagani, Didier Stricker, Christian Theobalt, and Vladislav Golyanik. Eventego3d++: 3d human motion capture from a head-mounted event camera. *International Journal of Computer Vision (IJCV)*, 2025. 1, 2, 3, 6, 7, 8, 4, 9, 10
- [27] Jinman Park, Kimathi Kaai, Saad Hossain, Norikatsu Sumi, Sirisha Rambhatla, and Paul Fieguth. Domain-guided spatio-temporal self-attention for egocentric 3d pose estimation. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2023. 1, 2
- [28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit

- Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 7
- [29] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 2016. 1, 2
- [30] Ciyu Ruan, Ruishan Guo, Zihang Gong, Jingao Xu, Wenhan Yang, and Xinlei Chen. Pre-mamba: A 4d state space model for ultra-high-frequent event camera deraining. In *International Conference on Computer Vision (ICCV)*, 2025. 3
- [31] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. Eventhands: Real-time neural 3d hand pose estimation from an event stream. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 1
- [32] Davide Scaramuzza. Omnidirectional camera. In *Computer vision: A reference guide*, pages 900–909. Springer, 2021. 6
- [33] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *Transactions on Graphics (TOG)*, 2020. 7
- [34] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 3
- [35] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xR-EgoPose: Egocentric 3D Human Pose From an HMD Camera. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2
- [36] Denis Tome, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando de la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *Pattern Analysis and Machine Intelligence (PAMI)*, 45(6):6794 – 6806, 2023. 1, 2
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5
- [38] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. Estimating egocentric 3d human pose in global space. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2
- [39] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, Diogo Luvizon, and Christian Theobalt. Estimating egocentric 3d human pose in the wild with external weak supervision. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [40] Jian Wang, Zhe Cao, Diogo Luvizon, Lingjie Liu, Kripasindhu Sarkar, Danhang Tang, Thabo Beeler, and Christian Theobalt. Egocentric whole-body motion capture with fisheyevit and diffusion-based motion refinement. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2
- [41] Ziyun Wang, Ruijun Zhang, Zi-Yan Liu, Yufu Wang, and Kostas Daniilidis. Continuous-time human motion field from event cameras. In *International Conference on Computer Vision (ICCV)*, 2025. 1
- [42] Ximea MU050CR-SY. <https://www.ximea.com/products/miniature-compact/ximu-smallest-industrial-usb-cameras/sony-imx675-usb3-color-ximu-smallest-camera>, 2025. 4
- [43] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [44] 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 (ECCV)*, 2024. 1, 4, 5, 6, 7, 2, 9, 10
- [45] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2
- [46] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [47] Dongxu Zhao, Zhen Wei, Jisan Mahmud, and Jan-Michael Frahm. EgoGlass: Egocentric-View Human Pose Estimation From an Eyeglass Frame. In *International Conference on 3D Vision (3DV)*, 2021. 1, 2
- [48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 4
- [49] Shihao Zou, Chuan Guo, Xinxin Zuo, Sen Wang, Hu Xiaoqin, Shoushun Chen, Minglun Gong, and Li Cheng. Eventhpe: Event-based 3d human pose and shape estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 3
- [50] Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4, 7