COUPLING TRAJECTORIES AND MASKS: AN EFFICIENT SPARSE-TO-DENSE ANNOTATION FRAMEWORK VIA BIDIRECTIONAL ENHANCEMENT FOR EVENT MOTION SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Event motion segmentation is bottlenecked by dense pixel-level masks, whose exhaustive annotation is labor-intensive and poorly matched to asynchronous event streams. Existing event-based segmentation approaches are poorly suited as scalable annotation tools: they either expose only discrete frame outputs or require costly mask, optical-flow, or auxiliary-modality supervision, limiting temporal coverage and annotation per interaction. We propose an efficient sparse-to-dense annotation framework that couples trajectories and masks through bidirectional enhancement. In *Motion-to-Mask*, a learned temporal trajectory field warps events to queried timestamps, forming motion-compensated Images of Warped Events (IWEs) that allow a frozen promptable segmentation model to generate masks from sparse bounding boxes. In *Mask-to-Motion*, the generated masks provide region-level constraints for mask-aware contrastive trajectory learning, refining the motion estimates used for IWE construction and mask propagation. Because the trajectory field is queryable within each annotation window and the promptable segmentation model propagates masks across query-time IWEs, the framework can produce masks at a denser temporal output rate than the box-prompt rate and transfer them to asynchronous events. Across three datasets, the generated masks enable fine-grained annotation with full or sparse prompts and no pixel-level mask or optical-flow GT; a user study shows up to 389.39% speedup at a 1:3 input-to-output frame-rate ratio across event-data experience levels.

1 INTRODUCTION

Event motion segmentation identifies independently moving regions in asynchronous event streams and is key to moving object detection, tracking, dynamic scene understanding, and event dataset construction (Stoffregen et al., 2019; Mitrokhin et al., 2019). Event cameras capture brightness changes with microsecond latency, making them useful for high-speed robotic perception and agile navigation where frame-based cameras suffer motion blur or latency limits (Gallego et al., 2020; Sanket et al., 2020). To exploit continuous sensing, downstream models require dense pixel-level masks and, in the finest case, per-event labels. Obtaining such supervision is labor-intensive because event streams are asynchronous, sparse, and rapidly displaced over time, so object boundaries evolve across short temporal intervals. A natural way to reduce the effort is to annotate a small number of event frames and derive masks from accumulated windows, but fixed windows introduce temporal quantization, blur, and boundary fragmentation (Bolten et al., 2022; Chen et al., 2025). Red-green-blue (RGB) or grayscale frames can assist labeling, but they reintroduce frame-rate limitations and cross-modal alignment costs (Mitrokhin et al., 2019; Burner et al., 2022). This annotation bottleneck limits the scalability of learning-based event vision models (Burner et al., 2022).

Dedicated annotation workflows for event-based motion segmentation remain limited, so existing event-based segmentation models provide the natural starting point for reducing annotation cost. Model-based segmentation methods avoid pixel-level mask supervision through explicit kinematic models, geometric assumptions, or global contrast maximization (Zhou et al., 2021; Yamaki et al., 2025). While annotation-efficient, their hand-crafted priors and scene-specific assumptions can

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

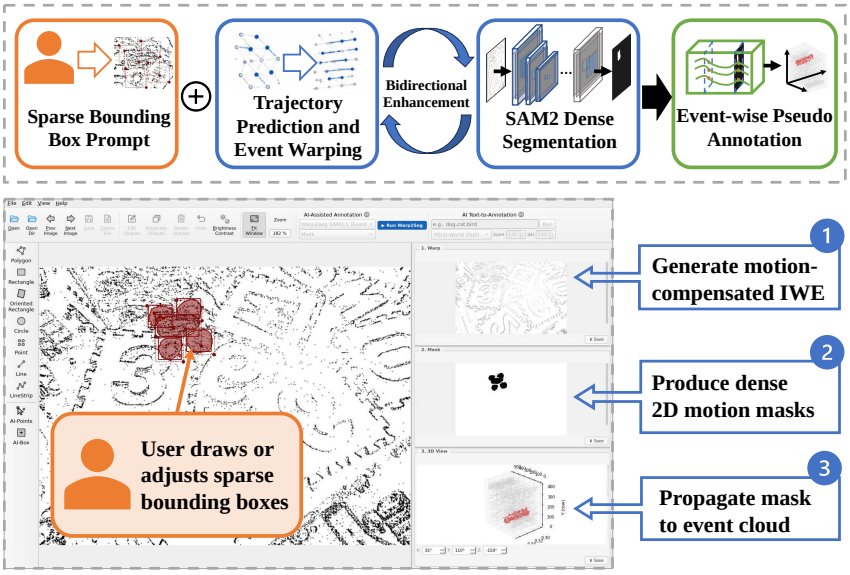


Figure 1: Annotation workflow. Bounding-box prompts guide segmentation of motion-compensated event images, and learned trajectories propagate masks to event-wise pseudo labels.

break down in scenes with multiple independent motions. Learning-based segmentation methods provide more expressive representations and robust predictions, yet remain dependent on dense pixel-level masks, synthetic supervision, or auxiliary frame-based modalities (Parameshwara et al., 2021; Li et al., 2025; Gu et al., 2026). These models are designed to produce segmentation outputs rather than support annotation, and their supervision requirements or motion assumptions make them unsuitable as a general sparse-to-dense annotation workflow. The field therefore faces a critical challenge: how to obtain accurate motion annotations at a denser temporal resolution and event-level granularity from minimal human input, without restrictive motion assumptions or dense pixel-level supervision.

Recent zero-shot and promptable segmentation models, such as Segment Anything Model 2 (SAM2), offer a possible route to this goal Ravi et al. (2025); Kirillov et al. (2023). In natural images and videos, these models can generate masks from lightweight prompts such as bounding boxes, suggesting that sparse human interaction could seed a denser mask sequence. However, directly applying them to raw events or simple event accumulation frames remains unreliable. Event frames are sparse and noisy, and rapid motion collapses events from different timestamps into the same temporal window, producing blur, ghosting, and broken contours Zhao et al. (2025); Rebecq et al. (2019). The critical challenge is constructing motion-compensated event representations that preserve fine spatial evidence, enabling dense masks at temporal resolutions exceeding the sparse manual prompts. These considerations motivate the *Motion-to-Mask* direction. We first estimate a temporal trajectory field and use it to warp events to a queried timestamp, forming a motion-compensated Image of Warped Events (IWE) with clearer object boundaries before applying promptable segmentation. The trajectory field provides the raw material for generating masks at timestamps between sparse manual prompts, while low-cost bounding boxes initialize the segmentation process. The generated masks then support the reverse *Mask-to-Motion* direction, which refines the trajectory estimates through mask-aware contrastive learning.

To realize this framework without depending on dense optical flow or trajectory labels, we build trajectory learning on Contrast Maximization (CM) Gallego et al. (2018), which can learn event alignment without flow label by maximizing the sharpness of warped event images. Yet the focus loss used in prior CM-based methods optimizes global IWE sharpness Gallego et al. (2019); Hamann et al. (2024) and can be dominated by background or majority motion, making it insufficient for separating independently moving regions. We therefore introduce the reverse direction *Mask-to-Motion*, where segmentation masks provide region layouts for mask-aware contrastive learning, encouraging events within the same mask to have compact motion representations while separating events across

different masks. This establishes bidirectional enhancement between motion estimation and mask generation (see the top of Figure 1): improved trajectories produce sharper IWEs and higher-quality masks, while the masks provide stronger region-level constraints for trajectory learning. Beyond reference-time mask generation, the same trajectory field supports masks at queried timestamps and propagates them to individual asynchronous events, producing event-wise pseudo labels for scalable event data annotation (Figure 1, bottom). Our main contributions are summarized as follows:

1. We introduce an efficient bidirectional enhancement framework for sparse-to-dense annotation that transforms temporally sparse bounding-box prompts into dense masks at queried timestamps and event-wise pseudo-labels, without requiring pixel-level masks or optical-flow ground truth (GT).
2. We introduce a bidirectional *Motion-to-Mask* and *Mask-to-Motion* pipeline, where motion-compensated IWEs connect events to a promptable segmentation model and mask-aware contrastive learning refines CM-based trajectory estimation for propagation.
3. We evaluate the proposed framework across three datasets under temporally sparse and per-frame prompt budgets. A controlled user study demonstrates higher annotation efficiency than manual mask drawing across users with different levels of event-data experience.

2 Related Work

2.1 Event-based Motion Segmentation

Early event-based motion segmentation methods were predominantly model-based. Stoffregen et al. optimized event-to-object assignments and motion parameters by motion compensation (Stoffregen et al., 2019), while Zhou et al. alternated motion-model fitting and label assignment on a spatio-temporal event graph (Zhou et al., 2021). Recent methods further exploit event-based normal flow for joint motion segmentation and ego-motion estimation (Hua et al., 2025), or separate motion layers through variational contrast maximization (Yamaki et al., 2025). These approaches reduce annotation needs, but still depend on explicit motion models or geometric assumptions.

Learning-based methods are more expressive but commonly require pixel-level masks or auxiliary supervision. EVIMO's benchmark and pipeline use pixel-level masks, depth, and ego-motion supervision (Mitrokhin et al., 2019). Later work explored spiking networks (Parameshwara et al., 2021), unsupervised pseudo labels from geometric constraints (Wang et al., 2024), moving-object segmentation for driving scenes (Zhou et al., 2024), and auxiliary depth, pose, or optical-flow modules for complex scenes (Georgoulis et al., 2024). Pellerito et al. report strong results with motion-aware event suppression, yet require per-frame pixel-level supervision and use future masks to supervise current predictions in a continuous-frame setting (Pellerito et al., 2026). Our method instead generates dense motion masks from low-cost box prompts without pixel-level mask supervision.

2.2 Flow-Supervision-Free Event Motion Estimation

Contrast Maximization (CM) provides a natural way to learn event motion without optical-flow GT by maximizing the sharpness of Images of Warped Events (Gallego et al., 2018). Paredes-Vallés et al. extended CM to sequential low-latency optical flow learning (Paredes-Vallés et al., 2023), and MotionPriorCMax introduced nonlinear trajectory priors for dense continuous-time motion estimation (Hamann et al., 2024). However, existing approaches mainly optimize global event alignment or trajectory consistency and do not explicitly exploit object-level layout information. Our method injects region-level constraints into CM-based trajectory learning through mask-aware contrastive learning.

2.3 Promptable Segmentation and Event Annotation

Promptable segmentation models such as SAM (Kirillov et al., 2023) and SAM2 (Ravi et al., 2025) enable dense mask generation from lightweight interactions and have shown strong zero-shot transfer on images and videos. However, directly applying such models to raw event accumulations remains challenging because asynchronous events produce sparse, noisy, and motion-blurred visual evidence. EventSAM adapts SAM for event-domain semantic segmentation through task-specific

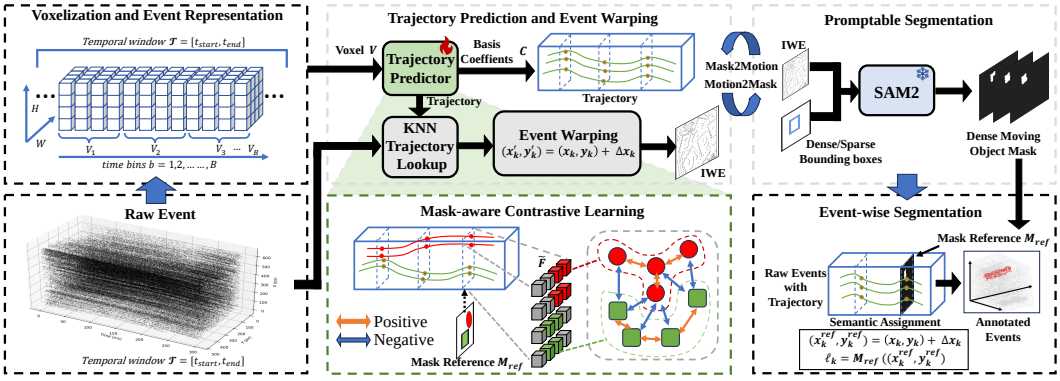


Figure 2: Overview of the proposed annotation framework. The bidirectional pipeline consists of *Motion-to-Mask* (forward), which uses CM-based trajectory prediction to generate IWE for SAM2 promptable segmentation, and *Mask-to-Motion* (reverse), which employs mask-aware contrastive learning to refine trajectory estimation.

adaptation (Chen et al., 2024). Label-efficient event vision methods have also studied reducing annotation cost, for example through pseudo-label self-training for event-based object detection (Wu et al., 2024), but they do not address temporally sparse motion annotation or trajectory-based propagation from frame masks to asynchronous events. Our method bridges this gap through motion-compensated representations, promptable segmentation, and event-wise label propagation.

3 Methodology

3.1 Overview

Our framework takes an asynchronous event stream $\{e_k\}_{k=1}^{N}$, where $e_k = (x_k, y_k, t_k, p_k)$ records a brightness change of polarity $p_k \in \{+1, -1\}$ at pixel (x_k, y_k) and time t_k, together with sparse box prompts $\mathcal{B} = \{b_j\}$ annotated at selected key frames, and outputs dense masks $\{M_{\tau_q}\}$ at queried timestamps, including a reference mask M_{ref}, plus event-wise pseudo labels $\{\ell_k\}$ assigning each event to a motion region. Its core mechanism is bidirectional enhancement between motion estimation and mask generation, as illustrated in Figure 2. In *Motion-to-Mask*, a trajectory predictor estimates per-event motion within a temporal window and warps events to a queried time, forming a motion-compensated Image of Warped Events (IWE), which SAM2 (Ravi et al., 2025) segments using box prompts. In *Mask-to-Motion*, the predicted mask provides region-level constraints for mask-aware contrastive learning, refining the trajectory predictor without dense optical flow or mask ground truth; the refined field and SAM2 propagation produce queried-time masks and event-wise annotations.

3.2 Motion-to-Mask: Event Warping and Promptable Segmentation

The forward process transforms event streams into dense motion masks through three stages: event voxelization and trajectory prediction, event warping and IWE generation, and promptable segmentation with given bounding boxes.

3.2.1 IWE Generation from Trajectory Prediction

Given a temporal window $\mathcal{T} = [t_{start}, t_{end}]$, we first convert the asynchronous events within this window into a voxel grid $V \in \mathbb{R}^{B \times H \times W}$, where B is the number of temporal bins and $H \times W$ is the spatial resolution. For each temporal bin $b \in \{1, \ldots, B\}$ spanning $[t_b, t_{b+1})$, the voxel grid is constructed by accumulating events:

$$V_b(x, y) = \sum_{k: t_k \in [t_b, t_{b+1})} p_k \cdot \delta(x - x_k, y - y_k), \tag{1}$$

where δ is the spatial Dirac delta and the sum covers all events in the temporal bin.

Following (Hamann et al., 2024), we use UNet with a polynomial basis and a RAFT-style architecture (Gehrig et al., 2024) with a Bezier basis. Both take the voxel grid as input. UNet predicts a full-resolution $2K$-channel coefficient field, while RAFT predicts the same channels on a grid downsampled by 8 using correlation-based matching and iterative refinement. Spatially sampled trajectories are reconstructed by:

$$\mathbf{x}(\tau) = \sum_{q=1}^{K} \mathbf{c}_q \phi_q(\tau), \tag{2}$$

where ϕ_q is the q-th basis function and $\mathbf{c}_q \in \mathbb{R}^2$ is its predicted two-dimensional coefficient, consistent with the $2K$ coefficient channels at each spatial sample. The polynomial basis assumes approximately constant velocity within the short window, while the Bezier basis provides trajectories for general motion patterns. Because $\mathbf{x}(\tau)$ is parameterized by temporal basis functions, it can be evaluated at any queried time τ_q within the prediction window. For a queried time τ_q, the same construction replaces τ_{ref} in Eq. 3 with τ_q, giving $\Delta\mathbf{x}_{n,b}(\tau_q) = \mathbf{x}_n(\tau_q) - \mathbf{x}_n(\tau_b)$. The resulting per-event displacements warp events to τ_q and form a query-time IWE, denoted by I_{τ_q}. We compute per-event displacement with a KNN-based flow lookup table. The spatial domain is divided into cells of size s; for each temporal bin and cell, we identify the N nearest sampled trajectories, denoted $\mathcal{N}_b^{(i,j)}$, and evaluate their displacements to the reference time:

$$\Delta\mathbf{x}_{n,b} = \mathbf{x}_n(\tau_{\text{ref}}) - \mathbf{x}_n(\tau_b), \tag{3}$$

where τ_{ref} is the normalized reference time and τ_b is the normalized mid-time of bin b. Averaging the neighborhood gives the grid-cell flow:

$$\Delta\mathbf{x}_b^{(i,j)} = \frac{1}{|\mathcal{N}_b^{(i,j)}|} \sum_{n \in \mathcal{N}_b^{(i,j)}} \Delta\mathbf{x}_{n,b}, \tag{4}$$

producing a flow lookup table indexed by (b, i, j). Each event $e_k = (x_k, y_k, t_k, p_k)$ selects a temporal bin b_k from t_k and a grid cell from its coordinates, $(i_k, j_k) = \left(\left\lfloor \frac{y_k}{s} \right\rfloor, \left\lfloor \frac{x_k}{s} \right\rfloor\right)$, and its displacement is read directly from the lookup table $\Delta\mathbf{x}_k = \Delta\mathbf{x}_{b_k}^{(i_k,j_k)}$. The warped event coordinates are $(x'_k, y'_k) = (x_k, y_k) + \Delta\mathbf{x}_k$. The Image of Warped Events at a queried time τ_q is generated by accumulating all warped events at τ_q using bilinear voting. For the reference-time case, $\tau_q = \tau_{\text{ref}}$. Formally, $I_{\tau_q}(x, y) = \sum_k p_k \cdot B(x - x'_k, y - y'_k)$, where B distributes each event to its four neighboring pixels and $p_k \in \{+1, -1\}$ is its polarity. This compensation aligns events from the same physical edge, suppresses motion blur, and produces sharper boundaries than raw accumulation. Evaluating the same trajectory field at different τ_q therefore provides the temporally dense IWE sequence used for annotation.

3.2.2 IWE-Guided Segmentation with Sparse Bounding Box Prompt

The query-time IWE I_{τ_q} is fed to frozen SAM2, which produces a dense mask M_{τ_q} from box prompts. The trajectory field supplies each motion-compensated IWE, while SAM2 propagates masks across them, decoupling the box-prompt and mask-output rates.

During trajectory training, we simulate sparse annotation by sampling N_s timestamps within each prediction window. If a sampled timestamp lacks an annotated frame, its box is interpolated from the nearest annotated frames.

Formally, let t be a sampled timestamp between the nearest preceding and succeeding annotated frames at $t_p < t < t_n$. After Hungarian matching, $(\mathbf{c}_p, \mathbf{s}_p)$ and $(\mathbf{c}_n, \mathbf{s}_n)$ denote centers and sizes at both frames. Using time weights $w_p = \frac{t_n - t}{t_n - t_p}$ and $w_n = \frac{t - t_p}{t_n - t_p}$, the interpolated center and size are

$$\mathbf{c} = w_p \mathbf{c}_p + w_n \mathbf{c}_n, \quad \mathbf{s} = w_p \mathbf{s}_p + w_n \mathbf{s}_n. \tag{5}$$

The predicted trajectory flow compensates motion between annotated and target times; outside the annotated range, we use the nearest frame with the same compensation. These boxes and the IWE generate training-time masks in SAM2.

During inference, matched-rate mode uses one box per output timestamp. Hyper-frame-rate mode sets $r_m > r_b$ and propagates the segmentation state between sparse prompts; both modes use trajectory-based IWEs.

3.3 MASK-TO-MOTION: MASK-AWARE CONTRASTIVE LEARNING AND EVENT-WISE ANNOTATION

Following CM-based methods (Hamann et al., 2024), the focus loss encourages sharp IWEs but can be dominated by background or majority motion in multi-object scenes. We use SAM2 masks to partition the reference-time IWE into motion regions; features from the same region form positive pairs, while features from different regions form negative pairs for mask-aware contrastive learning.

For the UNet architecture, the features are extracted from an intermediate decoder layer. For the RAFT architecture, we use the GRU hidden state (Cho et al., 2014) that carries iterative motion-refinement information. A flow-conditioned projection head then fuses these features with the predicted flow field via Feature-wise Linear Modulation (FiLM) (Perez et al., 2018), producing embeddings conditioned on both appearance and motion: $\tilde{F} = (1 + \gamma(F_{\text{flow}})) \cdot F_{\text{feat}} + \beta(F_{\text{flow}})$, where F_{feat} is the network feature, F_{flow} the predicted flow, and γ, β scale and shift parameters from a flow encoder, so motion directly conditions the feature representation.

For each mask time slot t_m, we compute the corresponding flow via the predicted trajectories and warp the feature grid to that time. Each warped pixel is assigned a unique label from the SAM2-generated mask based on its spatial region. Let ℓ_i and ℓ_j denote the region labels of pixels i and j, where pixels within the same connected component of the mask share the same label. The positive set for anchor i is defined as pixels sharing the same region label, $\mathcal{P}_i = \{j \mid \ell_j = \ell_i, j \neq i\}$, and the negative set as pixels from different regions, $\mathcal{N}_i = \{j \mid \ell_j \neq \ell_i\}$. The loss is computed as InfoNCE (Oord et al., 2018) over class-balanced sampled anchors:

$$\mathcal{L}_{\text{mac}} = -\frac{1}{|A|} \sum_{i \in A} \Big(\text{LSE}\Big(\frac{s_{ij}}{\tau}\Big)_{j \in \mathcal{P}_i} - \text{LSE}\Big(\frac{s_{ij}}{\tau}\Big)_{j \in \mathcal{P}_i \cup \mathcal{N}_i} \Big), \tag{6}$$

where each FiLM-modulated embedding is L_2-normalized as $\bar{f}_i = \tilde{f}_i / \|\tilde{f}_i\|_2$, and $s_{ij} = \bar{f}_i^\top \bar{f}_j$ is therefore their cosine similarity. Here, τ is the temperature, A is the anchor set, and LSE denotes the log-sum-exp operator. The total training objective is

$$\mathcal{L} = \mathcal{L}_{\text{focus}} + \mathcal{L}_{\text{smooth}} + \lambda_{\text{mac}} \mathcal{L}_{\text{mac}}. \tag{7}$$

Following (Hamann et al., 2024), $\mathcal{L}_{\text{focus}}$ sharpens IWEs and $\mathcal{L}_{\text{smooth}}$ regularizes trajectory variation; Appendix A.1 gives complete definitions and training details. Although SAM2 is frozen, mask-derived constraints improve trajectories and masks through bidirectional enhancement. Given M_{ref}, we warp each event e_k to the reference time using its trajectory displacement $\Delta \mathbf{x}_k$ and assign $\ell_k = M_{ref}(x_k^{ref}, y_k^{ref})$ with the same event-to-window and event-to-tile associations.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Following prior event motion segmentation works (Hua et al., 2026; Zhou et al., 2024), we adopt their benchmark selection and evaluation protocols on EVIMO2 (Burner et al., 2022), EVIMO Mitrokhin et al. (2019), and DSEC-MOS derived from DSEC Gehrig et al. (2021). None of these datasets provide GT bounding boxes, so we extract bounding boxes from the boundaries of the segmentation masks. For EVIMO2, which provides instance segmentation masks, we extract independent moving object masks according to the motion segmentation definitions used in prior work. The "full bboxes" setting supplies a bounding-box prompt at every available annotated frame, whereas the "1 bbox" setting uses one prompt within each 300 ms trajectory prediction window, corresponding to a prompt rate of about 3 FPS. Thus, these settings differ in temporal prompt density. At the same temporal density, a bounding box remains cheaper to annotate than a pixel-level mask because it provides coarser spatial supervision.

For segmentation, we report mask mean Intersection over Union (mIoU) on EVIMO2 and EVIMO, as well as Jaccard mean ($\mathcal{J}$ Mean), F-measure mean ($\mathcal{F}$ Mean), and their average ($\mathcal{J}\&\mathcal{F}$ Mean) on DSEC-MOS. In addition to segmentation, we evaluate trajectory estimation in the ablation study. We use the full EVIMO2 and DSEC datasets, which provide ground-truth optical flow, and report End-Point Error (EPE) on DSEC and Trajectory EPE (TEPE) Hamann et al. (2024) on EVIMO2 to measure the quality of the learned motion trajectories. To ensure comparability with Hamann

Table 1: Segmentation performance on EVIMO2.

Method	Per-Frame Annotation	Input Modality	mIoU
Model-based			
EMSMC	N/A	E	16.99
EMSGC	N/A	E	64.38
MSEE	N/A	E	77.40
VCM	N/A	E	84.00
Learning-based			
UDA	√	R/E	63.40
SemanticAided	√	E/S/D	79.82
MBO	√	E	85.12
SAM2-only (full bboxes)	√	E	71.83
Ours (full bboxes)	√	E	**85.42**
SAM2-only (1 bbox)	×	E	52.03
Ours (1 bbox)	×	E	**70.04**

Table 2: Segmentation performance on EVIMO.

Method	Per-Frame Annotation	Input Modality	Table	Box	Floor	Wall	Fast
Model-based							
EMSGC	N/A	E	55	24	18	24	43
Learning-based							
SpikeMS	√	E	50	65	53	63	38
GConv	√	E	51	60	55	**80**	39
EVDodgeNet	√	E/F	70	67	61	72	60
EVIMO	√	E/F/D	**79**	70	59	78	**67**
MBO	√	E	74	**77**	69	65	63
SAM2-only (full bboxes)	√	E	61	68	62	64	63
Ours (full bboxes)	√	E	66	72	**71**	73	**67**
Un-EVIMO	×	E/F/D	50	45	56	53	44
SAM2-only (1 bbox)	×	E	34	40	47	35	27
Ours (1 bbox)	×	E	**53**	**57**	**66**	**65**	**48**

Table 3: Segmentation performance on DSEC-MOS.

Method	Per-Frame Annotation	Input Modality	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	$\mathcal{J}\&\mathcal{F}$ Mean
RTNet	√	R/F	57.8	71.7	64.8
AMCNet	√	R/F	60.6	76.1	68.4
TMO	√	R/F	56.2	73.0	64.6
Isomer	√	R/F	55.9	72.2	64.1
F2Net	√	R	45.1	66.9	56.0
TMO	√	R	53.5	70.3	61.9
MED-VT	√	R	58.0	81.0	69.5
EmoFormer	√	R/E	**66.5**	82.2	74.4
SAM2-only (full bboxes)	√	E	63.9	82.9	73.4
Ours (full bboxes)	√	E	65.6	**83.3**	**74.5**
SAM2-only (1 bbox)	×	E	56.1	77.1	66.6
Ours (1 bbox)	×	E	**57.2**	**77.8**	**67.5**

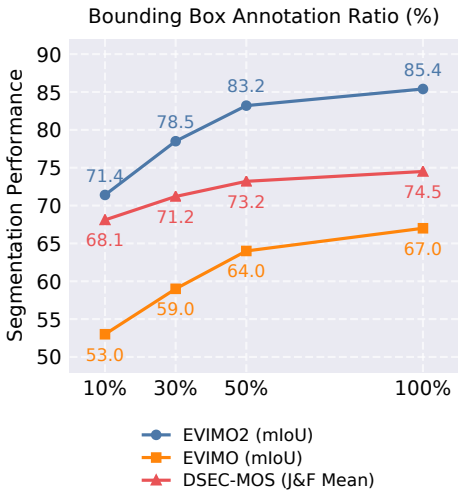


Figure 3: Segmentation performance under varying bounding box annotation ratios.

et al. (2024) and test the applicability of our method across trajectory architectures, we use a UNet with a polynomial basis on DSEC and RAFT-Spline with a Bezier basis on EVIMO2 and EVIMO, using 300 ms prediction windows for these three benchmarks during inference. The number of fixed mask-time samples for contrastive learning is set to $N_s = 2$ for DSEC and $N_s = 8$ for EVIMO2. We set $\lambda_{\text{mac}} = 0.005$. The trajectory models are trained on a single A800 GPU with the AdamW optimizer and a learning rate of 1e-4.

4.2 Main Results: Moving Object Segmentation

Tables 1–3 contextualize results across heterogeneous annotation and input regimes. *Per-Frame Annotation* marks frame-wise training labels or prompts; *Input Modality* lists events (E), RGB (R), frame-based optical flow (F), semantic labels (S), and depth (D). EVIMO2 baselines include EMSMC, EMSGC, MSEE, and VCM (Stoffregen et al., 2019; Zhou et al., 2021; Hua et al., 2025;

Table 4: User-study results on five EVIMO2 samples.

Experience	Annotation mode	Time (s) ↓	mIoU (%) ↑
Unfamiliar	Manual mask	32.28	68.82
Unfamiliar	BBox + tool	**16.93** (2.38)	**76.62**
Familiar	Manual mask	28.14	72.57
Familiar	BBox + tool	**12.49** (2.38)	**77.41**

Table 5: Speedup under different Input and Output frame rates.

Experience	Input:Output 1:1	1:2	1:3
Unfamiliar	90.67%	234.33%	346.47%
Familiar	125.30%	278.48%	389.39%

Yamaki et al., 2025), plus UDA, SemanticAided, and MBO (Jeryo & Harati, 2025; Jiang et al., 2024; Hua et al., 2026). DSEC-MOS baselines include RTNet, AMCNet, TMO, Isomer, F2Net, MED-VT, and EmoFormer (Ren et al., 2021; Yang et al., 2021; Cho et al., 2023; Yuan et al., 2023; Liu et al., 2021; Karim et al., 2023; Zhou et al., 2024). EVIMO baselines include EMSGC, SpikeMS, GConv, EVDodgeNet, EVIMO, MBO, and Un-EVIMO (Zhou et al., 2021; Parameshwara et al., 2021; Mitrokhin et al., 2020; Sanket et al., 2020; Hua et al., 2026; Wang et al., 2024).

On EVIMO2, Ours (full bboxes) reaches 85.42% mIoU, comparable to mask-supervised MBO (85.12%) and model-based VCM (84.00%). The full-box result serves as a mask-quality reference under dense prompting, whereas the one-box setting evaluates quality obtained from temporally sparse interaction. With one prompt per window, Ours reaches 70.04% mIoU, 18.01 points above SAM2-only, showing the value of motion-compensated IWEs and mask-aware contrastive learning. These initialization masks reduce the effort required to obtain accurate event annotations, as confirmed by the user study in Table 4. On DSEC-MOS (Table 3), Ours (full bboxes) reaches 74.5% $\mathcal{J}\&\mathcal{F}$ Mean using events alone, comparable to RGB+Event EmoFormer (74.4%). With one box per window, it reaches 67.5%, showing effectiveness under sparse annotation in driving scenes. On EVIMO (Table 2), Ours (full bboxes) is competitive across scenes and best on Floor and Fast. With sparse prompts, it exceeds SAM2-only by 21.2 points and Un-EVIMO by 8.2 points on average, using only events.

4.3 Annotation Efficiency and Prompt-Quality Trade-off

Figure 3 shows a monotonic performance increase as the annotation ratio rises from 10% to 100%, exposing the cost-quality trade-off. For EVIMO, each point averages all evaluated samples, not the five scene categories. Gains from 50% to 100% are smaller than from 10% to 30%, suggesting that moderate prompt density captures most benefits. EVIMO2 is most sensitive to annotation density (14 points), whereas DSEC-MOS is most robust (6.4 points), likely because of their indoor and driving domains.

To measure annotation efficiency directly, we conducted a controlled user study in which 20 participants, divided into unfamiliar and familiar event-data groups, each annotated five EVIMO2 samples using both manual mask drawing and the box-assisted workflow; Appendix B.1 details the full protocol. The study uses matched prompt and output rates, so each manual box corresponds to one output mask and the timing comparison isolates the cost of obtaining that mask. Table 4 shows that, compared with frame-by-frame manual mask drawing, the proposed box-assisted annotation method reduces the average annotation time by 47.6% for unfamiliar participants and by 55.6% for familiar participants. The parenthesized 2.38 s is the average model inference time measured on an NVIDIA A40 GPU. To complement this participant study, Appendix C simulates scale and center deviations that can arise during manual box placement and evaluates their effect on segmentation quality. More importantly, Table 5 shows that hyper-frame-rate mode directly addresses the scalable annotation bottleneck: each sparse prompt interval yields multiple masks, enabling dense annotation from minimal human input without pixel-level masks or optical-flow annotations, and achieving up to 389.39% speedup at a 1:3 input-to-output frame-rate ratio. This capability is described in Appendix A.2, illustrated in Figure 5, and demonstrated in the supplementary video.

4.4 Event-wise Annotation Quality

We construct pseudo-GT from EVIMO2's native masks and optical flow. GT flow warps each event to the nearest mask timestamp, where it receives the mask label. Events outside valid mask re-

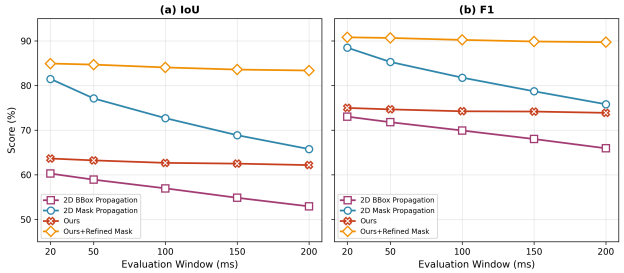


Figure 4: Event-wise annotation quality on EVIMO2 eval split. (a) IoU and (b) F1 macro-averaged over four methods across multiple evaluation windows.

Table 6: Ablation study on EVIMO2 and DSEC.

Dataset	Warp.	CL	CL type	mIoU or $\mathcal{J}\&\mathcal{F}$ Mean ↑	(T)EPE ↓
EVIMO2	×	×	N/A	71.83	N/A
	√	×	N/A	82.77	6.14
	√	√	Pred.	85.42	4.63
	√	√	GT	85.48	4.47
DSEC	×	×	N/A	73.40	N/A
	√	×	N/A	74.10	3.20
	√	√	Pred.	74.50	2.79
	√	√	GT	74.40	2.66

gions, including occluded boundaries and ambiguous-flow areas, are flagged invalid and excluded, so metrics use only reliable correspondences.

We compare four event-labeling methods. **2D BBox Propagation** (Chen et al., 2025) extrudes the center-frame bounding box over time and labels events inside the resulting cylinder. **2D Mask Propagation** (Bolten et al., 2022) temporally stretches a GT mask and queries event coordinates against the nearest anchor mask, requiring dense per-frame masks. **Ours** uses predicted trajectories to warp events to the SAM2 mask time and query labels there using only box prompts. **Ours+Refined Mask** follows the same pipeline but manually aligns the SAM2 mask with GT boundaries, matching the annotation quality assumed by 2D Mask Propagation. This refined condition is an upper-bound diagnostic and is not the standard box-only annotation workflow.

Figure 4 reports macro-averaged IoU and F1 over 20, 50, 100, 150, and 200 ms windows on the EVIMO2 eval split. **Ours+Refined Mask** achieves the highest IoU at all windows (84.93% at 20 ms and 83.37% at 200 ms), showing that motion-compensated propagation preserves boundaries over long spans. **2D Mask Propagation** degrades rapidly because temporal stretching cannot handle non-rigid motion or occlusion changes. Among box-prompt methods, **Ours** reaches 63.64% at 20 ms and 62.17% at 200 ms, consistently outperforming **2D BBox Propagation**. F1 follows the same trend, indicating that trajectory-based annotation yields more stable labels than rule-based methods. The supplementary video provides a continuous visualization of the proposed annotation workflow, including box prompting, mask generation, and propagation to asynchronous events.

4.5 Ablation Studies

Table 6 isolates motion compensation and mask-aware contrastive learning. On EVIMO2, motion compensation raises mIoU from 71.83% to 82.77%, a 10.94-point gain, and predicted-mask contrastive learning reaches 85.42% while reducing TEPE from 6.14 to 4.63; GT masks add only 0.06 points, indicating accurate region constraints. On DSEC, the same components improve $\mathcal{J}\&\mathcal{F}$ Mean from 73.40% to 74.50% and EPE from 3.20 to 2.79, with consistent gains across indoor and driving scenes. Together, these results show that both components improve mask initialization and trajectory quality for the sparse-to-dense annotation workflow.

5 Conclusion

We present an efficient sparse-to-dense annotation framework for event motion segmentation using sparse bounding-box prompts. It couples trajectory-based IWE construction with promptable mask generation and mask-aware contrastive trajectory learning. Queryable trajectories and SAM2 propagation produce masks at denser timestamps and event-wise labels without pixel-level mask or optical-flow supervision. Across three benchmarks, the generated masks provide useful initialization for fine-grained annotation under full and sparse prompting, while the user study shows reduced annotation time across experience levels. The box-perturbation analysis further supports robustness to moderate prompt variation.

AI USE STATEMENT

In this work, generative AI tools were used to assist with project environment setup, the implementation of basic software modules, and the development of experimental variants for ablation studies. They were also used to assist with and polish the writing of the manuscript and to support the retrieval and discovery of related work. All AI-assisted code was reviewed, tested, and integrated by the authors, and the corresponding ablation configurations and experimental outputs were independently verified. The authors also verified the accuracy and appropriateness of all AI-assisted text, retrieved literature, and citations. The authors take full responsibility for the final code, experimental results, and content of the paper.

REFERENCES

Tobias Bolten, Felix Lentzen, Regina Pohle-Fröhlich, and Klaus D Tönnies. Evaluation of deep learning based 3d-point-cloud processing techniques for semantic segmentation of neuromorphic vision sensor event-streams. In *VISIGRAPP (4: VISAPP)*, pp. 168–179, 2022.

Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. Evimo2: an event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv preprint arXiv:2205.03467*, 2022.

Nuo Chen, Chao Xiao, Yimian Dai, Shiman He, Miao Li, and Wei An. Event-based tiny object detection: A benchmark dataset and baseline. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7209–7218, 2025.

Zhiwen Chen, Zhiyu Zhu, Yifan Zhang, Junhui Hou, Guangming Shi, and Jinjian Wu. Segment any event streams via weighted adaptation of pivotal tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3890–3900, June 2024.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 5129–5138, 2023.

Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3867–3876, 2018.

Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12272–12281, 2019.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7): 4736–4746, 2024.

Stamatios Georgoulis, Weining Ren, Alfredo Bochicchio, Daniel Eckert, Yuanyou Li, and Abel Gawel. Out of the room: Generalizing event-based dynamic motion segmentation for complex scenes. In *2024 International Conference on 3D Vision (3DV)*, pp. 442–452. IEEE, 2024.

Fuqiang Gu, Yuanke Li, Xianlei Long, Kangping Ji, Chao Chen, Qingyi Gu, and Zhenliang Ni. Mambaseg: harnessing mamba for accurate and efficient image-event semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4302–4310, 2026.

Friedhelm Hamann, Ziyun Wang, Ioannis Asmanis, Kenneth Chaney, Guillermo Gallego, and Kostas Daniilidis. Motion-prior contrast maximization for dense continuous-time motion estimation. In *European Conference on Computer Vision*, pp. 18–37. Springer, 2024.

Zhiyuan Hua, Dehao Yuan, and Cornelia Fermüller. Motion segmentation and egomotion estimation from event-based normal flow. *arXiv preprint arXiv:2507.14500*, 2025.

Zhiyuan Hua, Cornelia Fermüller, and Yiannis Aloimonos. Moving border ownership for event-based motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37043–37052, 2026.

Mohammed Jeryo and Ahad Harati. Learning domain-invariant representations for event-based motion segmentation: An unsupervised domain adaptation approach. *Journal of Imaging*, 11(11): 377, 2025.

Chenao Jiang, Julien Moreau, and Franck Davoine. Event-based semantic-aided motion segmentation. In *19th International Conference on Computer Vision Theory and Applications (VISAPP 2024)*, volume 4, pp. 159–171, 2024.

Rezaul Karim, He Zhao, Richard P Wildes, and Mennatullah Siam. Med-vt: Multiscale encoder-decoder video transformer with application to object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6323–6333, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Hebei Li, Yansong Peng, Jiahui Yuan, Peixi Wu, Jin Wang, Yueyi Zhang, and Xiaoyan Sun. Efficient event-based semantic segmentation via exploiting frame-event fusion: A hybrid neural network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18296–18304, 2025.

Daizong Liu, Dongdong Yu, Changhu Wang, and Pan Zhou. F2net: Learning to focus on the foreground for unsupervised video object segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 2109–2117, 2021.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Anton Mitrokhin, Chengxi Ye, Cornelia Fermüller, Yiannis Aloimonos, and Tobi Delbruck. Evimo: Motion segmentation dataset and learning pipeline for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6105–6112. IEEE, 2019.

Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14414–14423, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Chethan M Parameshwara, Simin Li, Cornelia Fermüller, Nitin J Sanket, Matthew S Evanusa, and Yiannis Aloimonos. Spikems: Deep spiking neural network for motion segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3414–3420. IEEE, 2021.

Federico Paredes-Vallés, Kirk YW Scheper, Christophe De Wagter, and Guido CHE De Croon. Taming contrast maximization for learning sequential, low-latency, event-based optical flow. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9695–9705, 2023.

Roberto Pellerito, Nico Messikommer, Giovanni Cioffi, Marco Cannici, and Davide Scaramuzza. Motion-aware event suppression for event cameras. In *Robotics: Science and Systems 2026*, 2026.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3852–3861, 2019.

Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15430–15439, 2021.

Nitin J Sanket, Chethan M Parameshwara, Chahat Deep Singh, Ashwin V Kuruttukulam, Cornelia Fermüller, Davide Scaramuzza, and Yiannis Aloimonos. Evdodgenet: Deep dynamic obstacle dodging with event cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10651–10657. IEEE, 2020.

Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7244–7253, 2019.

Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7283–7293, 2021.

Ziyun Wang, Jinyuan Guo, and Kostas Daniilidis. Un-evimo: Unsupervised event-based independent motion segmentation. In *European Conference on Computer Vision*, pp. 228–245. Springer, 2024.

Ziyi Wu, Mathias Gehrig, Qing Lyu, Xudong Liu, and Igor Gilitschenski. Leod: Label-efficient object detection for event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16933–16943, 2024.

Ryo Yamaki, Shintaro Shiba, Gallego Guillermo, and Yoshimitsu Aoki. Iterative event-based motion segmentation by variational contrast maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4957–4966, 2025.

Shu Yang, Lu Zhang, Jinqing Qi, Huchuan Lu, Shuo Wang, and Xiaoxing Zhang. Learning motion-appearance co-attention for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1544–1553, 2021.

Yichen Yuan, Yifan Wang, Lijun Wang, Xiaoqi Zhao, Huchuan Lu, Yu Wang, Weibo Su, and Lei Zhang. Isomer: Isomerous transformer for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 966–976, 2023.

Yucheng Zhao, Gengyu Lyu, Ke Li, Zihao Wang, Hao Chen, Zhen Yang, and Yongjian Deng. Eseg: event-based segmentation boosted by explicit edge-semantic guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10510–10518, 2025.

Yi Zhou, Guillermo Gallego, Xiuyuan Lu, Siqi Liu, and Shaojie Shen. Event-based motion segmentation with spatio-temporal graph cuts. *IEEE transactions on neural networks and learning systems*, 34(8):4868–4880, 2021.

Zhuyun Zhou, Zongwei Wu, Danda Pani Paudel, Rémi Boutteau, Fan Yang, Luc Van Gool, Radu Timofte, and Dominique Ginhac. Event-free moving object segmentation from moving ego vehicle. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8960–8965. IEEE, 2024.

APPENDIX

A TRAINING AND ANNOTATION WORKFLOW

This section first details the optimization of the proposed framework and then presents the complete annotation workflow built around the optimized framework.

A.1 TRAINING PROCEDURE FOR THE PROPOSED FRAMEWORK

Trajectory objectives. Following (Hamann et al., 2024), the focus loss maximizes the spatial gradient magnitude of the IWE.

$$\begin{aligned} G_{\mathrm{IWE}} &= \frac{1}{HW} \sum_{x,y} \Big(|\partial_x \mathrm{IWE}(x,y)| \\ &\qquad + |\partial_y \mathrm{IWE}(x,y)| \Big), \\ \mathcal{L}_{\mathrm{focus}} &= G_{\mathrm{IWE}}^{-1}. \end{aligned} \tag{8}$$

To discourage spatially inconsistent trajectories, the smoothness loss applies a robust spatial-gradient penalty to the lookup-grid displacement $\Delta \mathbf{x}_b^{(i,j)}$ defined in the main paper.

$$\begin{aligned} \mathcal{L}_{\mathrm{smooth}} = \frac{\lambda_{\mathrm{smooth}}}{2B|\mathcal{G}|} \sum_{b=1}^{B} \sum_{d \in \{i,j\}} \\ \sum_{(i,j) \in \mathcal{G}} \rho\Big(\partial_d \Delta \mathbf{x}_b^{(i,j)} \Big), \\ \rho(z) = \sqrt{z^2 + \epsilon^2}. \end{aligned} \tag{9}$$

Here, $\mathcal{G}$ is the spatial lookup grid, and the Charbonnier penalty is applied element-wise to the displacement components. Depending on the trajectory setting, $\Delta \mathbf{x}_b^{(i,j)}$ represents displacement to the reference time or to the next temporal sample.

Alternating mask and trajectory updates. Training begins without a pseudo-mask cache. During the first epoch, and until the first scheduled refresh after the second epoch, the trajectory predictor is optimized using only the focus loss and smoothness regularization. Mask-to-Motion contrastive learning remains inactive because no SAM2-generated masks are available. This warm-up allows the trajectory predictor to produce sufficiently coherent IWEs before the masks are used as region-level guidance.

At every refresh, we uniformly sample 20% of all training windows without replacement within that refresh. A new subset is drawn at each refresh, so a window selected previously may be selected again in a later refresh. For every selected window, the current trajectory predictor constructs IWEs at $\{t_m\}_{m=1}^{N_s}$ and SAM2 generates all N_s pseudo masks from the corresponding bounding-box prompts. SAM2 remains frozen throughout training and inference. The new masks overwrite the cache entries of selected windows. Existing entries of unselected windows remain unchanged, while windows that have never been selected remain without cached masks. The refresh is performed every two epochs to reduce the training-time cost of repeated SAM2 inference.

Mask time contrastive optimization. Ordinary mini-batches are sampled independently of mask-cache availability. Every sample contributes to the focus loss and smoothness regularization. For each sample, the training step queries the cache using its window identifier. If one or more valid non-empty masks are available, Mask-to-Motion aligns features to those mask times and computes the contrastive loss over the valid slots. Samples without a cached mask, or with no valid mask slots, do not contribute to the contrastive term in that step. The contrastive loss is averaged over valid mask times rather than over all samples in the batch.

At each valid mask time t_m, the trajectory predictor evaluates the corresponding flow and aligns its deep motion features to that time. FiLM fuses the trajectory features with the predicted flow before

Stage	Operation	Updated component or state
Initialization	Initialize the trajectory predictor, mask-aware projection module, and an empty pseudo-mask cache. Keep SAM2 fixed.	Mask cache
Warm-up	Until the first mask refresh, estimate trajectories and construct IWEs without Mask-to-Motion contrastive learning.	Trajectory predictor
Refresh sampling	After the second epoch and every two epochs thereafter, sample 20% of all training windows without replacement. Redraw the subset at each refresh, allowing the same window to be selected in later refreshes.	Refresh subset
Mask generation	For every selected window, construct IWEs at $\{t_m\}_{m=1}^{N_s}$ and regenerate all N_s masks from the corresponding bounding-box prompts.	Selected masks
Cache update	Overwrite cache entries for selected windows. Keep existing entries for unselected windows. Windows not yet selected remain without cached masks.	Mask cache
Batch optimization	Sample ordinary mini-batches without enforcing mask availability. Apply focus and smoothness losses to every sample. Add the mask-aware contrastive loss only for samples with valid cached mask slots.	Trajectory predictor and mask-aware projection
Repeat	Continue batchwise trajectory optimization and periodic cache refresh until training finishes.	Trained trajectory predictor

Table 7: Overall training procedure for trajectory learning in the proposed framework.

the feature grid is warped to the mask coordinates. Connected components in the SAM2-generated mask define region labels. Features assigned to the same region form positive pairs, while features from different regions form negative pairs. We use class-balanced anchor sampling (Wang et al., 2021) and an InfoNCE objective with temperature 0.1. The resulting mask-aware contrastive loss is combined with the focus loss and smoothness regularization as

$$\mathcal{L} = \mathcal{L}_{\text{focus}} + \mathcal{L}_{\text{smooth}} + \lambda_{\text{mac}}\mathcal{L}_{\text{mac}}, \tag{10}$$

where $\lambda_{\text{mac}} = 0.005$. Only the trajectory predictor and mask-aware feature projection are optimized by this objective. The periodic mask refresh is an alternating training schedule rather than an inference-time loop.

Table 7 summarizes the complete training process, including trajectory optimization and periodic pseudo-mask updates.

A.2 Annotation Workflow

Based on the optimized framework, we implement an annotation workflow that converts raw event streams and sparse box prompts into dense masks at selected timestamps and pseudo labels for individual events.

Hyper-frame-rate mode for temporal densification. The workflow separates the box-prompt rate r_b from the mask-output rate r_m. In matched-rate mode, $r_m = r_b$, so each prompted timestamp produces one output mask. In hyper-frame-rate mode, $r_m > r_b$: the user annotates boxes only at sparse prompt timestamps, the continuous trajectory field constructs an IWE at each queried output timestamp, and frozen SAM2 propagates the mask state between prompted IWEs. The resulting masks are then transferred to asynchronous events using their individual timestamps. Thus, one sparse prompt interval can produce multiple output masks without requiring a box at every output timestamp. The corresponding interface configuration is shown in Figure 5.

1. **Event frame construction and box annotation.** The user uploads a raw event stream and specifies the prompt rate, the mask-output rate, and the event accumulation window. The

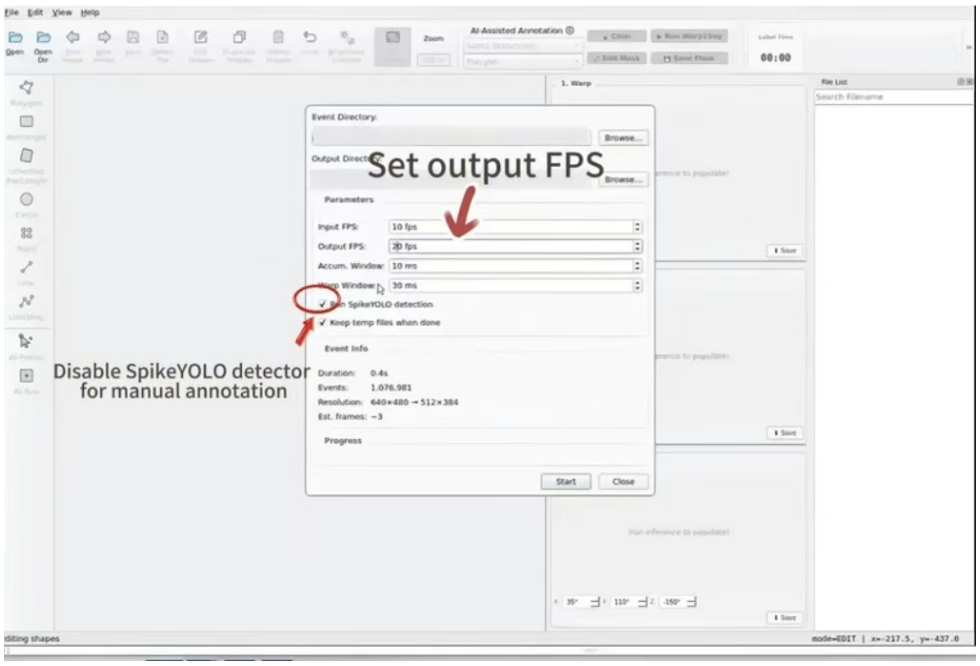


Figure 5: Hyper-frame-rate mode for temporal densification. The mask-output rate is set to 20 FPS while the box-prompt rate is 10 FPS, allowing multiple masks to be generated between sparse manual prompts.

prompt rate determines the timestamps presented for annotation. At each timestamp, the tool accumulates events within the selected window to construct a raw event frame. The user then draws bounding boxes around the motion regions of interest.

2. **Framework optimization and optional detection.** The manually annotated boxes guide training of the proposed framework through the bidirectional enhancement between trajectory estimation and mask generation. The preceding subsection provides the complete optimization procedure. The tool also provides an optional interface for an external object detector. The constructed event frames and manual boxes can train the detector independently. In our video demonstration, this interface uses SpikeYOLO (Luo et al., 2024). The trained detector can propose boxes at a higher prompt frame rate or on new event streams with similar data distributions. Boxes generated by the detector do not participate in framework training.
3. **Mask and event label generation.** After optimization of the proposed framework, either manual boxes or boxes generated by the detector can provide prompts for annotation generation. At timestamps determined by the mask-output rate, the optimized trajectory predictor estimates motion and constructs IWEs. Frozen SAM2 then generates the corresponding 2D masks. In hyper-frame-rate mode, the tool uses the video propagation interface provided by SAM2 to propagate segmentation between prompted frames. The source of the boxes does not change this propagation process. The learned trajectories finally transfer the 2D masks to individual asynchronous events and produce pseudo labels for each event.

Framework optimization and optional detector training are offline stages that require no participant interaction. In our tests, the optimized inference pipeline of the proposed framework, including trajectory prediction and SAM2 segmentation, uses approximately 2 GB of GPU memory.

B User Study on Annotation Cost

We conducted a controlled user study to compare conventional manual mask annotation with the bounding-box-assisted annotation workflow used by our method. The study included 20 partici-

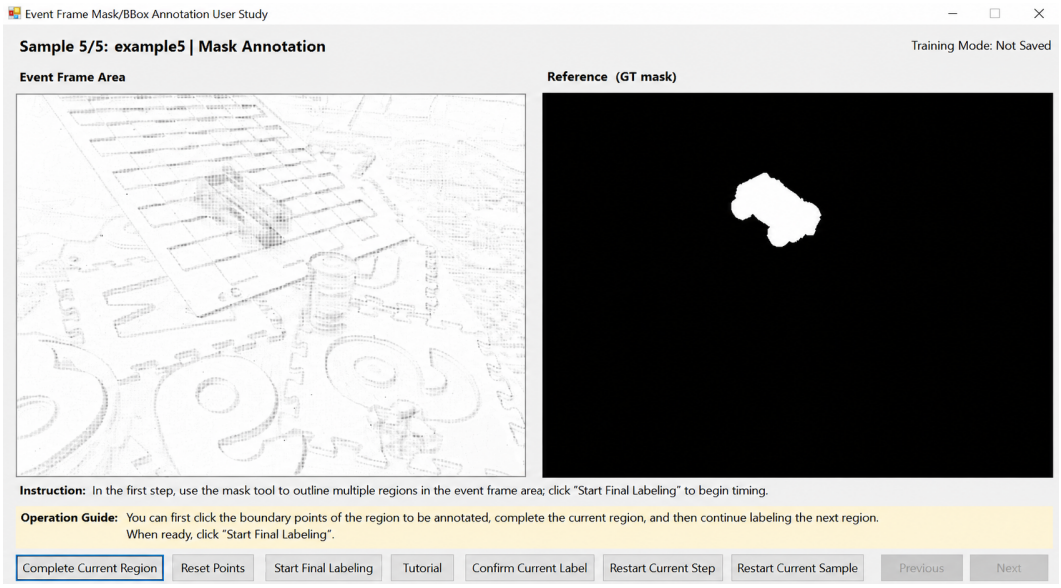


Figure 6: Interface used in the annotation user study. Participants annotate the event frame on the left while referring to the GT mask on the right. The same interface supports pixel-level mask drawing and bounding-box annotation.

pants divided equally into two groups. Ten participants were unfamiliar with event data and ten participants had prior experience working with event data. All participants annotated the same five samples randomly selected from EVIMO2 (Burner et al., 2022). Because event-based motion segmentation was unfamiliar to most participants, directly identifying object regions from sparse event frames was difficult. Even participants with prior event-data experience were not necessarily familiar with this segmentation task. We therefore displayed the ground-truth (GT) mask as a visual reference to make the target region unambiguous and to keep annotation quality and completion time comparable across participants.

B.1 Annotation Protocol

Each participant completed two annotation tasks on the same five samples. In the *manual mask* task, the participant traced the object region directly on the event frame to produce a pixel-level mask. We recorded the interaction time and evaluated the annotated mask against the GT mask using mean intersection over union (mIoU). In the *bounding box + annotation tool* task, the participant instead drew a bounding box around the target region. The box was then passed to our annotation tool to generate the dense mask. For this task, the recorded time includes both bounding-box interaction and model inference after optimization. It excludes offline framework and detector training because neither requires participant interaction. The generated mask was evaluated against the same GT mask. To balance annotation quality and time, we counted a trial as successful only when the submitted mask or bounding box achieved an IoU above 65% with the corresponding GT region. Reported time and mIoU values are averaged over the five samples and all successful annotations in the corresponding group.

Figure 6 shows the study interface. The event frame is displayed in the annotation area on the left, while the GT reference mask is shown on the right. The interface provides separate controls for outlining a pixel-level region or specifying a bounding box and records the final annotation time for each sample.

Numerical user-study results and their analysis are reported in Sec. 4.

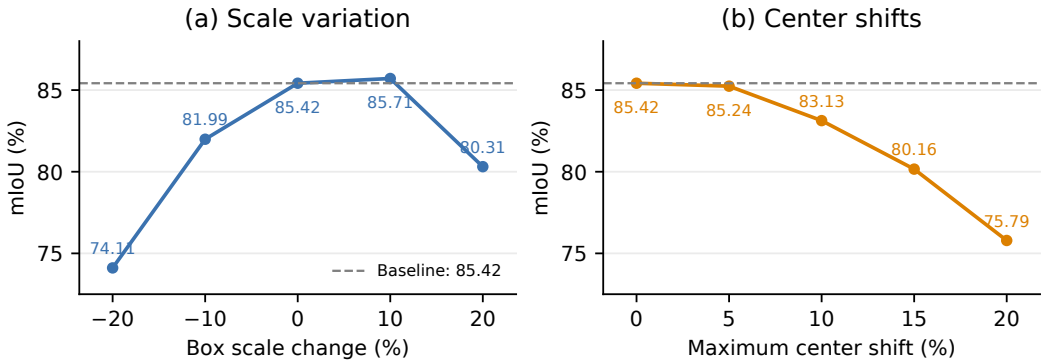


Figure 7: Robustness to bounding-box perturbations on EVIMO2. (a) Scale changes are applied about the original box center. (b) Random center offsets are bounded relative to the box width and height. The dashed line marks the unperturbed 85.42% mIoU baseline.

C Robustness to Bounding-Box Perturbations

Manual bounding boxes inevitably vary across annotators. To evaluate the resulting sensitivity, we introduce box-scale and random-center perturbations on the EVIMO2 eval split. The RAFT-Spline trajectory model first warps events to construct the warp-inter IWE, which SAM2 (Hiera-B+) (Ravi et al., 2025) segments using a perturbed GT box as its prompt. We perturb only this prompt. The trajectory model, event frame, IWE, and GT mask remain unchanged, and binary foreground and background mIoU is computed against the original GT mask. Disabling perturbations exactly reproduces the 85.42% baseline.

For a prompt box of width w and height h, scale perturbation resizes it about its center as $(w', h') = (sw, sh)$ for $s \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$. Center perturbation independently samples offsets $d_x \sim \mathcal{U}(-r, r)w$ and $d_y \sim \mathcal{U}(-r, r)h$ for $r \in \{0, 0.05, 0.10, 0.15, 0.20\}$. We evaluate the two perturbations separately to isolate their effects.

Figure 7 shows that the proposed framework tolerates moderate prompt variation. Scaling by $\pm 10\%$ changes mIoU by at most 3.43 points, and a random center shift bounded by 5% reduces it by only 0.18 points. A 10% expansion even improves mIoU slightly, suggesting that modest surrounding context can preserve useful boundary events. The asymmetric scale curve also shows that overly tight prompts are more harmful. A 20% reduction loses 11.31 points, compared with 5.11 points for a 20% expansion. Center translation causes monotonic degradation, reaching a 9.63-point drop at 20%. Overall, the framework is stable under small variations typical of manual box annotation, although large offsets and boxes that exclude object boundaries remain challenging.