PRESERVE BROADLY, OBSERVE ADAPTIVELY: BUDGET-DECOUPLED EVENT-POINT LEARNING FOR ACTION RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Event cameras capture motion as asynchronous events, making them well suited to action recognition under rapid motion and challenging illumination. However, event recordings vary in cardinality, while existing point-based pipelines commonly form fixed-size network inputs by randomly sampling events from each recording or temporal clip before learning. This one-shot reduction couples event retention to per-pass computation, irreversibly discarding evidence from dense recordings while padding or duplicating events in short ones. We introduce EventVista, a preservation-aware framework that decouples the retention budget from the per-pass computation budget through three advances. First, Authentic Candidate Preservation preserves authentic evidence beyond a single forward pass without increasing per-pass computation by retaining a broader, recording-wide candidate pool while maintaining the natural cardinality of short recordings. Second, Coverage-Aware Runtime Observation constructs bounded, temporally complementary views from this candidate pool and processes them while preserving their actual cardinalities. Adaptive Evidence Integration then sequentially aggregates predictions from the shared network and stops the observation process once the accumulated prediction becomes sufficiently confident and stable. Across THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, EventVista achieves state-of-the-art Top-1 accuracies of 70.40% ($\uparrow$ 6.68 pp), 99.35%, and 50.68% ($\uparrow$ 2.38 pp), respectively. [CODE]

1 INTRODUCTION

Event cameras report pixel-wise brightness changes asynchronously rather than capturing images at a fixed frame rate. Their high temporal resolution and low latency support recognition under rapid motion (Amir et al., 2017), while their wide dynamic range is advantageous under challenging illumination (Gao et al., 2023). Existing methods encode event streams as accumulated frames (Innocenti et al., 2021), learned event grids (Gehrig et al., 2019), asynchronous graphs (Schaefer et al., 2022), or sparse spatiotemporal point sets (Wang et al., 2019; Ren et al., 2023). Cross-modal approaches further align event representations with image and text embeddings (Zhou et al., 2024a). Dense representations enable the reuse of image and video backbones but may quantize temporal structure and introduce computation over inactive regions. Point-based methods aggregate local spatiotemporal neighborhoods directly from selected events (Wang et al., 2019; Ren et al., 2023). They also support efficient hierarchical aggregation of local motion patterns and global action structure (Qi et al., 2017a;b), making the event-point domain a natural choice for accurate and resource-efficient recognition in deployment scenarios.

However, the main limitation of existing event-point pipelines often arises before the encoder processes the data. Event recordings vary substantially in event count because of motion speed, scene contrast, recording duration, and sensor activity (Wang et al., 2024b), whereas each forward pass can process only a bounded number of points. As illustrated in Fig. 1, existing pipelines commonly resolve this mismatch by sampling a fixed-size input from each recording or temporal window (Ren et al., 2023; 2024; 2025). Although convenient for batching, this single reduction couples the evidence retained from a recording to the evidence processed in one forward pass. Dense recordings may lose discriminative events, while short recordings require padding or duplication that alters

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.

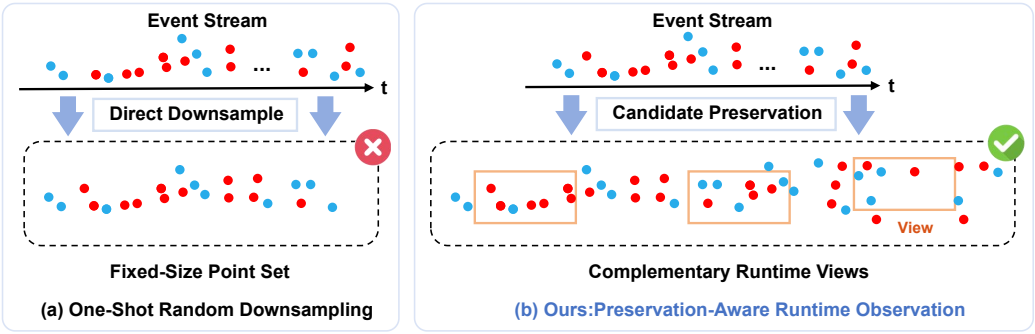

Figure 1: Comparison of event sampling strategies. (a) One-shot random downsampling produces a fixed-size point set and discards unselected events. (b) EventVista preserves recording-wide candidates and constructs complementary views under a bounded per-pass budget.

event multiplicity, local density, and neighborhood geometry. More importantly, the sampled subset is treated as the complete observation rather than one partial view of the recording. When this subset omits informative motion, the model has neither a preserved source pool nor a mechanism for acquiring complementary evidence. Consequently, recognition depends not only on representation learning, but also on which events happen to survive preprocessing. Once informative events are discarded, deeper spatiotemporal modeling cannot recover them at a later stage.

To address this premature evidence loss without increasing the computation of a single forward pass, we introduce EventVista, a preservation-aware event-point framework that separates the retention budget from the per-pass computation budget. As shown in Fig. 1(b), EventVista preserves a broader recording-wide candidate pool and accesses it through bounded runtime observations, organizing recognition around what to retain, what to observe next, and when to stop. First, to preserve authentic evidence beyond one forward pass, Authentic Candidate Preservation retains a broader set of source events while preserving the natural cardinality of short recordings. Coverage-Aware Runtime Observation constructs bounded, temporally complementary views on demand and passes them through a shared point encoder and classifier. Packed batching and offset-aware pooling preserve view boundaries and support variable-length inputs without padding or duplication. Adaptive Evidence Integration sequentially aggregates predictions from the shared network and stops once the accumulated prediction becomes confident and stable. Together, these components preserve evidence broadly, observe it selectively, encode it without cardinality distortion, and allocate computation according to sample difficulty. Across THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, EventVista achieves Top-1 accuracies of 70.40% (↑ 6.68 pp over POKER (Cao et al., 2026)), 99.35% (0.27 pp below TBR (Innocenti et al., 2021)), and 50.68% (↑ 2.38 pp over MVFNet (Wu et al., 2021)), respectively, demonstrating broad empirical effectiveness.

Our main contributions are summarized as follows:

- We identify the coupling between event retention and per-pass computation as a fundamental bottleneck in point-based event action recognition, and recast recognition as a budgeted partial-observation process concerned with what evidence remains accessible and how much of it should be processed under realistic input constraints.
- We propose EventVista, a preservation-aware framework that decouples the retention budget from the per-pass computation budget and organizes event-point recognition into a unified retain–observe–stop process.
- We develop three coordinated components: Authentic Candidate Preservation retains recording-wide source evidence, Coverage-Aware Runtime Observation exposes it through bounded, complementary views, and Adaptive Evidence Integration combines predictions and adjusts the number of observations using confidence and stability.
- EventVista achieves Top-1 accuracies of 70.40%, 99.35%, and 50.68% on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, respectively, establishing new state-of-the-art performance on THU-EACT-50-CHL and DailyDVS-200 while remaining competitive on DVSGesture under a unified framework.

2 Related Work

2.1 Event-Based Action Recognition

Event-based action recognition has explored dense representations, spike-based computation, and sparse geometric modeling. EST (Gehrig et al., 2019) learns event-to-grid mappings, TBR (Innocenti et al., 2021) encodes temporal occupancy in compact frames, and EV-ACT (Gao et al., 2023) fuses complementary event representations. ExACT (Zhou et al., 2024b) and EventBind (Zhou et al., 2024a) further introduce language-guided reasoning and event–image–text alignment, respectively. Spiking neural networks (SNNs), exemplified by Motion SNN (Liu et al., 2021), extract motion features through spike-based computation. In related asynchronous perception, AEGNN (Schaefer et al., 2022) uses graph neural networks (GNNs) to update affected features as events arrive. Point-based action recognizers operate directly on spatiotemporal coordinates: TTPOINT (Ren et al., 2023) compresses hierarchical point features through tensorization, SpikePoint (Ren et al., 2024) combines point processing with SNNs, and EventMamba (Ren et al., 2025) strengthens temporal modeling with state-space modules. More broadly, general-purpose point encoders range from PointNet++'s hierarchical neighborhood aggregation (Qi et al., 2017b) to Point Transformer V3's efficient serialized neighborhood mappings (Wu et al., 2024).

Remark. Accumulated frames enable the reuse of dense backbones, but discretize temporal activity and can incur computation over inactive regions. Additional semantic supervision does not by itself recover the individual timestamps merged during aggregation. Point inputs instead retain the coordinates and timestamps of selected source events, allowing local motion to be modeled directly as spatiotemporal geometry. This motivates our choice of event points: we avoid image reconstruction and image–text pretraining while retaining source-level temporal detail. Their explicit, variable cardinality also makes it possible to control which events are processed under a bounded per-pass budget.

2.2 Event Preprocessing and Sampling

Event preprocessing controls noise, temporal coverage, and point cardinality. Window-based denoising combines temporal correlation filtering with learned spatial modeling (Fang et al., 2025), while geometric reduction includes farthest-point sampling (Qi et al., 2017b) and grid subsampling (Wu et al., 2024). For event-point recognition, TTPOINT (Ren et al., 2023) distributes a fixed sampling budget across temporal subwindows to improve temporal coverage; SpikePoint (Ren et al., 2024) samples a fixed number of events from sliding-window clips; and EventMamba (Ren et al., 2025) downsamples temporal windows before feature extraction. These strategies reduce the data volume entering the network and make variable-length recordings tractable.

Remark. Denoising and geometric subsampling address noise or redundancy, but do not themselves decouple retained evidence from network-input size. Random sampling can overlook temporally localized motion, whereas temporal stratification improves within-view coverage without necessarily preserving additional evidence for later observation. If preprocessing stores only a fixed-size subset, omitted events cannot be revisited; duplicating short inputs instead changes event multiplicity. We therefore separate candidate retention from runtime input formation. Voxel-based source-event selection reduces local repetition while retaining original coordinates rather than synthetic centroids, and a larger candidate budget keeps additional support accessible beyond one forward pass. Short candidate sets require no padding or duplication, while larger pools are accessed through temporally distributed complementary views. Confidence-and-stability-based stopping adjusts the number of observations, rather than requiring every recording to use a fixed view count.

3 Method

An event camera produces a sequence

$$\mathcal{E} = \{e_i\}_{i=1}^{N}, \qquad e_i = (x_i, y_i, t_i, p_i), \tag{1}$$

where N is the number of events, (x_i, y_i) are the pixel coordinates, t_i is the timestamp, and p_i is the event polarity. Following the point-based formulation, the default model represents each event by the spatiotemporal point $\mathbf{u}_i = (x_i, y_i, t_i)$. A polarity-augmented input is evaluated in Appendix B.

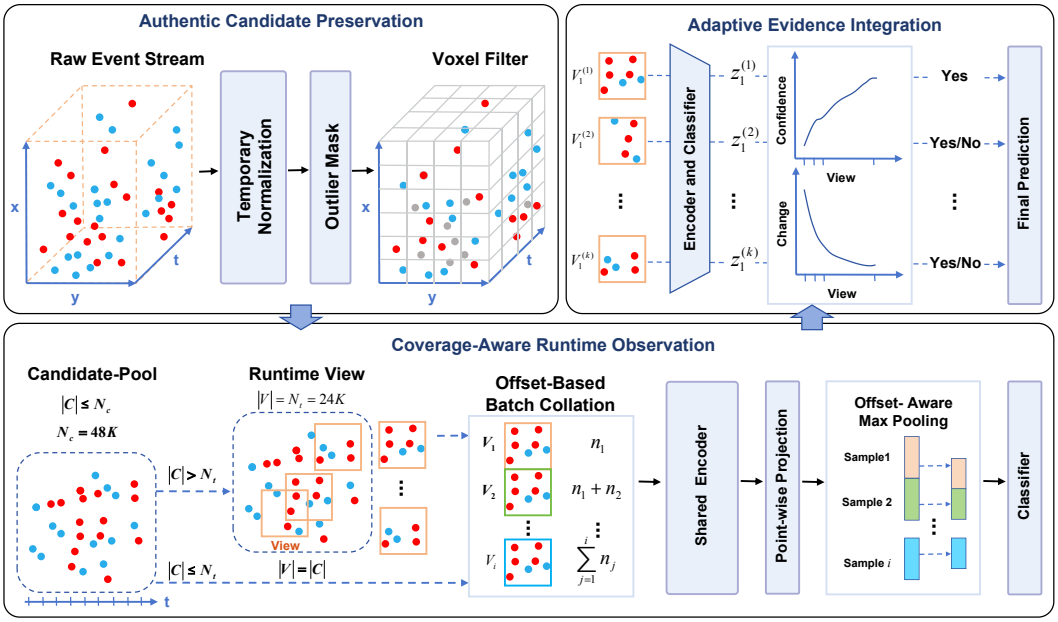

Figure 2: Overview of EventVista. Authentic Candidate Preservation retains a recording-wide pool of source events. Coverage-Aware Runtime Observation constructs bounded complementary views and processes them with a shared encoder, offset-aware global max pooling, and classifier while preserving their actual cardinalities. Adaptive Evidence Integration accumulates view-level predictions and stops when the aggregated prediction becomes confident and stable.

Event counts vary substantially across recordings, whereas one forward pass can process only a bounded number of points. We therefore distinguish the retention budget N_c, which limits the amount of source evidence that remains accessible, from the per-pass computation budget N_t. Conventional fixed-size pipelines implicitly set $N_c = N_t$, whereas EventVista permits $N_c > N_t$ and accesses the retained evidence through bounded views:

$$\mathcal{C} = \mathcal{P}(\mathcal{E}) \subseteq \mathcal{E}, \quad |\mathcal{C}| \le N_c; \qquad \mathcal{V}^{(k)} \subseteq \mathcal{C}, \quad |\mathcal{V}^{(k)}| \le N_t. \tag{2}$$

Here, $\mathcal{P}$ denotes Authentic Candidate Preservation, $\mathcal{C}$ is the retained candidate set, and $\mathcal{V}^{(k)}$ is the k-th runtime view. EventVista comprises three coordinated components: Authentic Candidate Preservation determines what remains accessible, Coverage-Aware Runtime Observation constructs and encodes bounded views, and Adaptive Evidence Integration determines how their predictions are combined and when observation stops. Fig. 2 summarizes this retain–observe–stop pipeline.

Candidate construction is performed once on the complete recording. Runtime views are drawn only from the retained pool, without revisiting the raw event stream. Their order is prediction-independent; confidence and stability determine only the processed prefix length. This separates the evidence available for recognition from the evidence evaluated before termination.

3.1 Authentic Candidate Preservation

Authentic Candidate Preservation determines *what to retain*. A normalized coordinate copy is used only to detect isolated outliers, and the mask is applied to the original event stream. The remaining events are partitioned into spatiotemporal voxels, each contributing one original event rather than a synthesized centroid. Thus, $\mathcal{C} \subseteq \mathcal{E}$ contains only source events while locally repetitive activity is reduced. If the number of voxel representatives exceeds N_c, a dataset-specific source-event subsampling rule enforces the retention cap, as detailed in Appendix A. Candidate sets below the cap are retained without padding or duplication. This stage preserves recording-wide evidence for observation without prematurely constructing the final network input.

3.2 Coverage-Aware Runtime Observation

Coverage-Aware Runtime Observation determines *what to observe next*. When $|\mathcal{C}| \leq N_t$, the complete candidate set forms the only view, avoiding padding or event repetition. Otherwise, the normalized temporal extent is divided into M non-overlapping intervals $\{\mathcal{C}_m\}_{m=1}^{M}$. Each interval receives a quota q_m proportional to its cardinality, with $\sum_m q_m = N_t$, and the k-th runtime view is

$$\mathcal{V}^{(k)} = \bigcup_{m=1}^{M} \mathcal{S}_m^{(k)}, \qquad \mathcal{S}_m^{(k)} \subseteq \mathcal{C}_m, \quad |\mathcal{S}_m^{(k)}| = q_m. \tag{3}$$

Here, m indexes temporal intervals, $\mathcal{C}_m$ is the candidate subset in interval m, and $\mathcal{S}_m^{(k)}$ is the subset selected from that interval for view k. Within each interval, later views prioritize events not selected by earlier views. Each view therefore spans the action temporally, while successive views expose complementary source evidence. All selected points remain original events drawn directly from $\mathcal{C}$.

Each runtime view is processed by a shared point encoder following established hierarchical sampling and local aggregation designs (Qi et al., 2017b; Qian et al., 2022; Lin et al., 2023). To preserve the actual cardinality of each view, we concatenate only valid points and record cumulative offsets $o_b = \sum_{j=1}^{b} n_j$, where n_j is the point count of view j. These offsets delimit individual views during sampling and neighborhood aggregation and are updated after downsampling. Following pointwise projection, offset-aware global max pooling aggregates features within each view's boundaries, yielding one descriptor per view for a shared classifier. This connects bounded observation to prediction without padding or duplicating short inputs: dense recordings can provide successive views, while short candidate sets are encoded in full. The resulting view-level logits are passed to Adaptive Evidence Integration.

3.3 Adaptive Evidence Integration

A bounded view may omit discriminative events, making predictions subset-dependent. For view k, let $\mathbf{z}^{(k)} = f_\theta(\mathcal{V}^{(k)})$ denote the logits produced by the shared encoding and classification pipeline. Adaptive Evidence Integration combines predictions from successive observations. After k observations, it computes

$$\overline{\mathbf{z}}^{(k)} = \frac{1}{k}\sum_{r=1}^{k} \mathbf{z}^{(r)}, \qquad \mathbf{a}^{(k)} = \mathrm{softmax}\Big(\overline{\mathbf{z}}^{(k)}\Big). \tag{4}$$

Here, r indexes views, $\overline{\mathbf{z}}^{(k)}$ averages their logits, and $\mathbf{a}^{(k)}$ is the aggregated class distribution. This parameter-free integration uses one shared model rather than an ensemble or learned fusion branch. Confidence and prediction change are measured by the top-two probability margin and the distance between successive distributions:

$$m^{(k)} = a_{(1)}^{(k)} - a_{(2)}^{(k)}, \qquad d^{(k)} = \left\|\mathbf{a}^{(k)} - \mathbf{a}^{(k-1)}\right\|_1, \tag{5}$$

where $a_{(1)}^{(k)}$ and $a_{(2)}^{(k)}$ denote the largest and second-largest probabilities. After at least two views, observation stops when

$$m^{(k)} \geq \tau_m, \qquad d^{(k)} \leq \tau_d, \qquad \arg\max_c a_c^{(k)} = \arg\max_c a_c^{(k-1)}. \tag{6}$$

Here, τ_m and τ_d are confidence and stability thresholds, and c indexes action classes. Otherwise, inference continues until $K_{\max}$. Recordings with $|\mathcal{C}| \leq N_t$ stop after processing their complete candidate set once. Training uses exactly one stochastic view per recording with cross-entropy; all views share the same encoder and classifier.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. We evaluate EventVista on THU-EACT-50-CHL (Gao et al., 2023), DVSGesture (Amir et al., 2017), and DailyDVS-200 (Wang et al., 2024a). The benchmarks contain

Table 1: Top-1 accuracy (%) on three event-based action-recognition benchmarks. EventMamba's 12-class Daily DVS result is excluded because it is not DailyDVS-200.

Method	Pub.	Modality	THU-EACT-50-CHL	DVSGesture	DailyDVS-200
Event-stream and point-based methods					
HMAX SNN (Xiao et al., 2020)	TNNLS'20	Event stream	32.70	–	–
Motion SNN (Liu et al., 2021)	IJCAI'21	Event stream	47.30	–	–
PointNet++ (Qi et al., 2017b)	NeurIPS'17	Event points	–	95.30	–
TTPOINT (Ren et al., 2023)	MM'23	Event points	–	98.80	–
SpikePoint (Ren et al., 2024)	ICLR'24	Event points / SNN	–	98.74	–
EventMamba (Ren et al., 2025)	TPAMI'25	Event points	59.40	99.20	–
Frame-, token-, and cross-modal methods					
EV-ACT (Gao et al., 2023)	TPAMI'23	Frame (stack)	58.50	–	–
C3D (Tran et al., 2015)	ICCV'15	Event frames	–	–	21.99
R(2+1)D (Tran et al., 2018)	CVPR'18	Event frames	–	–	36.06
TSM (Lin et al., 2019)	ICCV'19	Event frames	–	–	40.87
EST (Gehrig et al., 2019)	ICCV'19	Learned event tensor	–	–	32.23
MVFNet (Wu et al., 2021)	AAAI'21	Multi-view frame	59.56	–	48.30±0.27
MVF-Net (Deng et al., 2022)	TCSVT'22	Multi-view event frame	–	–	43.98±0.18
ESTF (Wang et al., 2024b)	AAAI'24	Event tokens	49.50	–	24.68
TimeSformer (Bertasius et al., 2021)	ICML'21	Frame tokens	–	–	44.25
Video Swin-T (Liu et al., 2022)	CVPR'22	Frame tokens	–	–	48.06
TBR + I3D (Innocenti et al., 2021)	ICPR'21	Event frames	–	**99.62**	–
IBM TrueNorth CNN (Amir et al., 2017)	CVPR'17	Event frames / SNN	–	96.50	–
SpikMamba (Chen et al., 2024)	MMAsia'24	Event frames / SNN	–	99.01	–
ExACT (Zhou et al., 2024b)	CVPR'24	Event frames + text	–	98.86	–
EMP (Cao et al., 2025)	ICCV'25	Stack + recon. frames + text	–	98.86	–
EventBind (stack) (Zhou et al., 2024a)	ECCV'24	Stack frame + text	60.77	–	–
EventBind (recon.) (Zhou et al., 2024a)	ECCV'24	Reconstructed frame + text	61.14	–	–
POKER (stack) (Cao et al., 2026)	CVPR'26	Stack frame + text	62.06	–	–
POKER (recon.) (Cao et al., 2026)	CVPR'26	Reconstructed frame + text	63.72	–	–
EventVista	–	Event points	**70.40**	99.35	**50.68**

50, 11, and 200 action categories, respectively, and exhibit substantially different event-cardinality distributions. We report recording-level Top-1 accuracy for all benchmarks. Together, these benchmarks cover both short recordings and highly dense event-stream regimes. THU-EACT-50-CHL contains 1,786 training and 544 test recordings, DVSGesture contains 1,077 training and 264 test recordings, and the adopted DailyDVS-200 split contains 14,773 training and 4,093 test recordings.

Implementation details. EventVista uses normalized spatiotemporal coordinates (x, y, t) by default. The tested polarity variant is reported in Appendix B. Unless otherwise specified, we use $N_c = 48\,\mathrm{K}$, $N_t = 24\,\mathrm{K}$, $M = 8$, $K_{\max} = 5$, $\tau_m = 0.70$, and $\tau_d = 0.10$. Candidate sets with $|\mathcal{C}| \leq N_t$ are processed once in full; the default adaptive policy evaluates at least two views for larger pools. Within each observation strategy, stopping rules use the same checkpoint and ordered view bank. The view order is prediction-independent. Training uses one stochastic view per recording.

The shared point encoder has $L = 9$ stages with output widths $32, 32, 64, 64, 128, 128, 256, 256$, and 512. Each stage retains half of the input points by farthest-point sampling and uses $K_{\mathrm{nn}} = 24$ neighbors for both cross- and intra-resolution aggregation, followed by residual refinement. Views are packed without padding, and cumulative offsets are updated after sampling to preserve their boundaries throughout feature extraction. Point-wise projection and offset-aware global max pooling produce a 512-D descriptor per view for the shared classifier. Models are trained for 350 epochs with batch size 8 using AdamW, an initial learning rate of 10^{-3}, weight decay of 10^{-4}, and cosine annealing. We apply dataset-specific point dropping and coordinate perturbations, together with label smoothing on DVSGesture and DailyDVS-200. Stopping thresholds are selected on validation data and fixed before final evaluation.

4.2 Comparison with Published Methods

Table 1 compares EventVista with representative event-stream, point-based, frame-based, token-based, spiking, and cross-modal approaches. EventVista achieves Top-1 accuracies of 70.40%,

Table 2: Controlled variants of EventVista . Each variant changes candidate retention, runtime-view construction, inference policy, or batch execution while retaining the same encoder architecture. Top-1 accuracy (%) is reported.

Variant	THU-EACT-50-CHL	DVSGesture	DailyDVS-200
w/o Candidate Preservation	65.82	95.02	44.71
w/o Runtime Observation	66.98	98.96	48.03
Per-recording execution	67.74	97.61	47.56
w/o Adaptive Integration	65.40	97.35	48.68
EventVista	70.40	99.35	50.68

Table 3: Comparison of runtime observation strategies under a common per-view cap and fixed-prefix logit aggregation. The coupled baseline retains 24K candidates; all other strategies retain 48K.

Observation strategy	THU-EACT-50-CHL	DVSGesture	DailyDVS-200
Coupled pool + 5-view TTA	65.31	95.36	45.89
Decoupled pool + random views	66.42	97.33	46.53
Temporal stratification	67.98	98.02	48.11
Coverage-aware complementarity	66.70	98.11	49.87

99.35%, and 50.68% on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, respectively. It exceeds the listed POKER reconstruction-based result by 6.68 percentage points on THU-EACT-50-CHL and the listed MVFNet result by 2.38 points on DailyDVS-200. On DVSGesture, it surpasses the listed point-based methods and remains close to the strongest listed frame-based result. These results use only sparse event coordinates, whereas several compared systems use accumulated frames, reconstructed images, or language supervision. Results follow the respective published settings for event-window construction, representation, and sequence aggregation. EventMamba's 12-class Daily DVS result is excluded because that benchmark differs from DailyDVS-200. Fig. 3 compares accuracy with reported model size: EventVista uses 2.168M parameters on DailyDVS-200, and all runtime views reuse the same encoder and classifier.

4.3 Analysis of Retention–Computation Decoupling

Table 2 examines candidate retention, runtime observation, adaptive integration, and batch execution under controlled settings. Restricting retention to a single-view budget reduces accuracy on all three benchmarks, supporting the value of keeping additional evidence accessible. Replacing coverage-aware views with random subsets also lowers accuracy, while fixed-view integration is less effective than adaptive inference. The per-recording execution control separately examines packed batching while retaining the same encoder architecture and unpadded variable-length inputs. The adaptive policy is analyzed further below because it affects both recognition accuracy and the number of evaluated views. Detailed definitions of these controls are provided in the supplementary material. Table 3 compares runtime observation strategies with a common per-view cap and fixed-prefix logit aggregation.

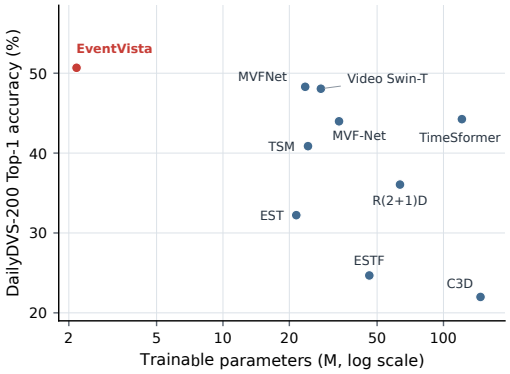

Figure 3: DailyDVS-200 accuracy versus reported parameter count for the compared models. The horizontal axis is logarithmic; marker sizes are uniform.

The coupled baseline uses $N_c = N_t$, whereas the decoupled variants retain a 48K candidate pool and process at most 24K events per view. Random views already improve accuracy over the coupled baseline. Temporal stratification further improves

Table 4: Accuracy–computation trade-off of fixed-view and adaptive inference. Acc. denotes Top-1 accuracy (%), and Views is the average number of evaluated runtime views.

	THU-EACT-50-CHL		DVSGesture		DailyDVS-200	
Inference rule	Acc.	Views	Acc.	Views	Acc.	Views
Fixed 1 view	60.70	1.00	93.45	1.00	43.11	1.00
Fixed 3 views	64.66	3.00	96.31	3.00	47.91	3.00
Fixed 5 views	65.56	5.00	97.40	5.00	48.89	5.00
Confidence only	69.25	2.31	96.17	1.86	47.65	2.69
Stability only	70.03	2.07	98.98	1.72	50.19	2.48
Confidence + stability	**70.40**	2.64	**99.35**	2.12	**50.68**	3.08

Table 5: Observation strategy crossed with inference policy. Acc. denotes Top-1 accuracy (%), and Views denotes the mean number of evaluated views. Fixed 5 averages all five view logits, whereas Adaptive applies the joint margin–stability stopping criterion.

		THU-EACT-50-CHL		DVSGesture		DailyDVS-200	
Observation	Inference	Acc.	Views	Acc.	Views	Acc.	Views
Temporal stratification	Fixed 5	67.98	5.00	98.02	5.00	48.11	5.00
Temporal stratification	Adaptive	68.12	2.78	99.05	2.20	48.92	3.16
Coverage-aware complementarity	Fixed 5	65.40	5.00	97.35	5.00	48.68	5.00
Coverage-aware complementarity	Adaptive	**70.40**	2.64	**99.35**	2.12	**50.68**	3.08

accuracy by 1.56, 0.69, and 1.58 percentage points on the three datasets, respectively. Coverage-aware complementarity prioritizes previously unobserved events, but does not consistently outperform stratification under fixed-five-view aggregation. We therefore examine its interaction with stopping below.

4.4 Adaptive Evidence Integration and Efficiency

Table 4 compares fixed-prefix inference with confidence-only, stability-only, and joint stopping. These rules share the encoder, candidate pool, and ordered view sequence. Fixed-prefix inference uses a prescribed observation limit, whereas adaptive inference selects a recording-dependent prefix. The joint criterion achieves higher accuracy than either single-criterion alternative on all three datasets. Its mean evaluated prefix is 2.64, 2.12, and 3.08 views, respectively, with a maximum of five.

Table 5 crosses observation construction with inference policy. Under adaptive inference, coverage-aware complementarity improves on temporal stratification by 2.28, 0.30, and 1.76 percentage points while evaluating 0.14, 0.08, and 0.08 fewer views on average, respectively. Together with the fixed-prefix results, this supports the combination of complementary observation and sample-dependent stopping. Adaptation controls the processed prefix length, rather than the view order.

EventVista contains 2.129M, 2.119M, and 2.168M trainable parameters on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, respectively. The profiled single-view costs are 3.857, 6.780, and 6.780 GFLOPs. Multiplying these costs by the mean evaluated view counts gives network-computation estimates of 10.18, 14.37, and 20.88 GFLOPs per recording. These estimates exclude candidate construction; exact costs for variable-length inputs require summing the computation of the individual evaluated views. Additional observations increase computation without increasing model capacity.

4.5 Qualitative Analysis

Using examples from DVSGesture, Fig. 4 shows the additional source events exposed by successive views. Fig. 5 shows the corresponding prediction evolution, obtained by applying softmax to the mean logits of the first k views. In the prediction-rescue example, the ground-truth class overtakes its competitor after View 4 and remains correct at the five-view limit. In the early-termination example,

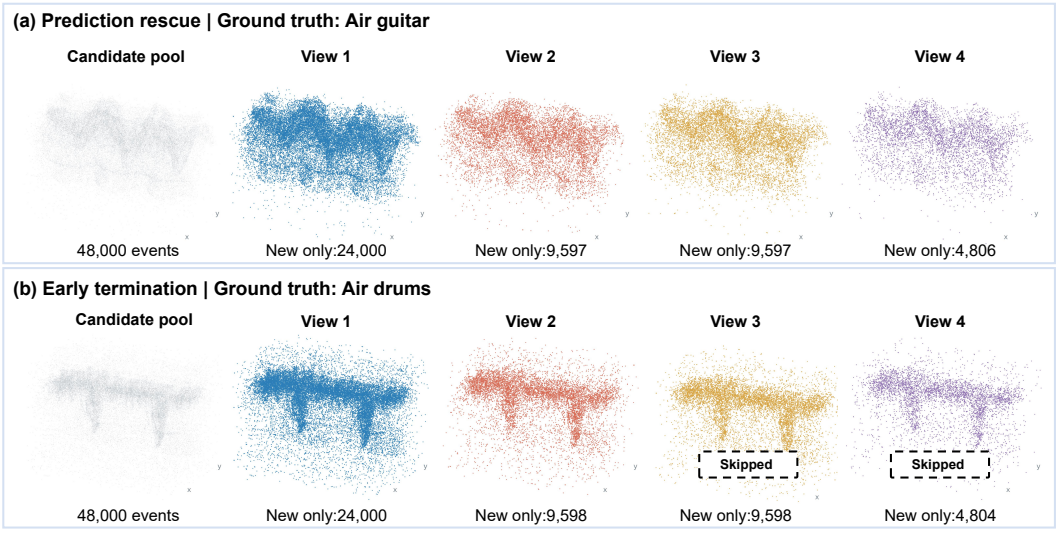

Figure 4: Complementary observation under a bounded per-view budget. (a) Successive views expose source events absent from the initial view. (b) An easy sample stabilizes after two views and skips later views.

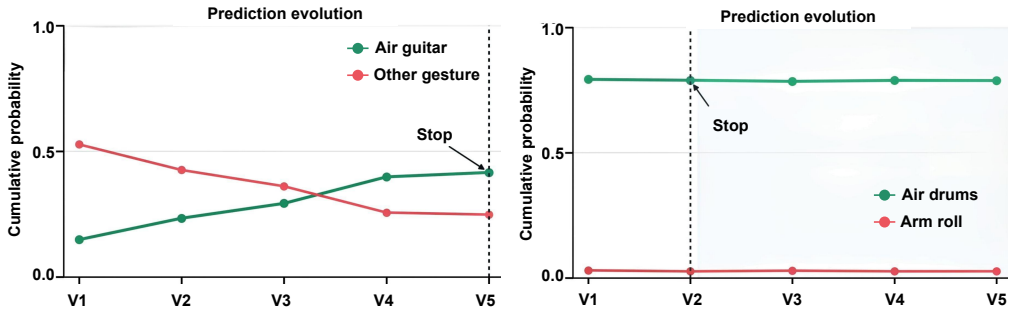

Figure 5: Prediction evolution with successive views. The hard example terminates at the view limit; the easy example stops when confidence and stability conditions hold. Dashed lines mark termination.

confidence and stability permit stopping after View 2; Views 3–5 are displayed only for fixed-$K_{\max}$ auditing. These examples illustrate how complementary observation can correct an initial prediction and how stable predictions can terminate early.

5 Conclusion

We introduce EventVista, a preservation-aware event-point framework that decouples evidence retention from per-pass computation. Authentic Candidate Preservation retains a broad recording-wide pool without expanding short recordings. Coverage-Aware Runtime Observation accesses this pool through bounded, complementary views and supports their variable cardinalities through shared encoding with offset-aware pooling. Adaptive Evidence Integration accumulates predictions from one shared model and stops when they become confident and stable, adapting computation to sample difficulty. EventVista achieves 70.40%, 99.35%, and 50.68% Top-1 accuracy on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, respectively, setting new state of the art on THU-EACT-50-CHL and DailyDVS-200 while remaining competitive on DVSGesture. These results show that event-point recognition benefits not only from better encoding, but from preserving evidence broadly and observing it adaptively.

AI USE STATEMENT

Generative AI tools assisted with literature discovery and reference checking, manuscript drafting and language editing, LaTeX preparation, and feedback on experimental reporting and result interpretation. The authors evaluated these suggestions and take responsibility for the research design, experimental results, citations, and final manuscript.

ETHICS STATEMENT

Our experiments use existing publicly available event-based action and gesture datasets and involve no new data collection or human-subject recruitment. Applications that monitor human activity should account for privacy, consent, and recognition errors across subjects and recording conditions. Deployment in consequential settings requires evaluation beyond the reported benchmarks.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the experimental setup. Appendix A documents dataset protocols, candidate construction, runtime-view formation, the shared encoder, and training settings. Additional controls, sensitivity analyses, and computation accounting are provided in Appendices B–D.

REFERENCES

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252, 2017.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824, 2021.

Meiqi Cao, Xiangbo Shu, Xin Jiang, Rui Yan, Yazhou Yao, and Jinhui Tang. Exploiting frequency dynamics for enhanced multimodal event-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5969–5979, 2025.

Meiqi Cao, Jiachao Zhang, Xin Jiang, Rui Yan, Yazhou Yao, Zechao Li, and Xiangbo Shu. Seeing motion through polarity for event-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37075–37085, 2026.

Jiaqi Chen, Yan Yang, Shizhuo Deng, Da Teng, and Liyuan Pan. SpikMamba: When SNN meets Mamba in event-based human action recognition. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, pp. 43:1–43:8, 2024. doi: 10.1145/3696409.3700204.

Yongjian Deng, Hao Chen, and Youfu Li. MVF-Net: A multi-view fusion network for event-based object classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8275–8284, 2022. doi: 10.1109/TCSVT.2021.3073673.

Huachen Fang, Jinjian Wu, Qibin Hou, Weisheng Dong, and Guangming Shi. Fast window-based event denoising with spatiotemporal correlation enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1381–1394, 2025. doi: 10.1109/TPAMI.2024.3467709.

Yue Gao, Jiaxuan Lu, Siqi Li, Nan Ma, Shaoyi Du, Yipeng Li, and Qionghai Dai. Action recognition and benchmark using event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14081–14097, 2023. doi: 10.1109/TPAMI.2023.3300741.

Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5633–5643, 2019.

Simone Undri Innocenti, Federico Becattini, Federico Pernici, and Alberto Del Bimbo. Temporal binary representation for event-based action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10426–10432, 2021. doi: 10.1109/ICPR48806.2021.9412991.

Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17682–17691, 2023.

Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7083–7093, 2019.

Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1743–1749, 2021. doi: 10.24963/ijcai.2021/240.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3202–3211, 2022.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017b.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hongwei Ren, Yue Zhou, Haotian Fu, Yulong Huang, Renjing Xu, and Bojun Cheng. TTPOINT: A tensorized point cloud network for lightweight action recognition with event cameras. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8026–8034, 2023. doi: 10.1145/3581783.3612258.

Hongwei Ren, Yue Zhou, Xiaopeng Lin, Yulong Huang, Haotian Fu, Jie Song, and Bojun Cheng. SpikePoint: An efficient point-based spiking neural network for event cameras action recognition. In *International Conference on Learning Representations*, 2024.

Hongwei Ren, Yue Zhou, Jiadong Zhu, Xiaopeng Lin, Haotian Fu, Yulong Huang, Yuetong Fang, Fei Ma, Hao Yu, and Bojun Cheng. Rethinking efficient and effective point-based networks for event camera classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6228–6241, 2025. doi: 10.1109/TPAMI.2025.3556561.

Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. AEGNN: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12371–12381, 2022.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4489–4497, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Qi Wang, Zhou Xu, Yuming Lin, Jingtao Ye, Hongsheng Li, Guangming Zhu, Syed Afaq Ali Shah, Mohammed Bennamoun, and Liang Zhang. DailyDVS-200: A comprehensive benchmark dataset for event-based action recognition. In *Computer Vision – ECCV 2024*, pp. 55–72, 2024a. doi: 10.1007/978-3-031-72907-2_4.

Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From RGB cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1826–1835, 2019. doi: 10.1109/WACV.2019.00199.

Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. HARDVS: Revisiting human activity recognition with dynamic vision sensors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5615–5623, 2024b. doi: 10.1609/aaai.v38i6.28372.

Wenhao Wu, Dongliang He, Tianwei Lin, Fu Li, Chuang Gan, and Errui Ding. MVFNet: Multi-view fusion network for efficient video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2943–2951, 2021.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024.

Rong Xiao, Huajin Tang, Yuhao Ma, Rui Yan, and Garrick Orchard. An event-driven categorization model for AER image sensors using multispike encoding and learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3649–3657, 2020. doi: 10.1109/TNNLS.2019.2945630.

Jiazhou Zhou, Xu Zheng, Yuanhuiyi Lyu, and Lin Wang. EventBind: Learning a unified representation to bind them all for event-based open-world understanding. In *Computer Vision – ECCV 2024*, pp. 477–494, 2024a. doi: 10.1007/978-3-031-72897-6_27.

Jiazhou Zhou, Xu Zheng, Yuanhuiyi Lyu, and Lin Wang. ExACT: Language-guided conceptual reasoning and uncertainty estimation for event-based action recognition and more. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18633–18643, 2024b. doi: 10.1109/CVPR52733.2024.01763.

A Reproducibility Details

This section summarizes the complementary roles of the evaluated datasets and records implementation choices omitted from the main paper. Framework equations and main-paper figures are not repeated.

A.1 Datasets and evaluation coverage

We evaluate EventVista on three complementary event-based action recognition benchmarks. For all datasets, we strictly follow the officially released splits without constructing additional random or subject-level partitions.

THU-EACT-50-CHL. THU-EACT-50-CHL (Gao et al., 2023) contains 2,330 recordings of 50 actions performed by 18 subjects. It includes whole-body motions, fine-grained actions, human–object interactions, healthcare activities, and two-person interactions. Recordings were captured using a DAVIS346 camera from four viewpoints in corridor and open-hall environments, with variations in illumination and action magnitude. We follow the official cross-subject split of 14 training and 4 test subjects, corresponding to 1,786 training and 544 test recordings. Its viewpoint variation, confusing action classes, and highly variable event-stream cardinalities make it our primary benchmark for evaluating broad evidence preservation and authentic variable-length processing.

DVSGesture. DVSGesture (Amir et al., 2017) is a widely used gesture benchmark recorded using a 128×128 DVS sensor. It contains 11 arm and hand gestures performed by 29 subjects under three illumination conditions. We strictly use its official cross-subject protocol, with subjects 1–23 for training and subjects 24–29 for testing; the segmented recordings processed by our pipeline contain 1,077 training and 264 test samples. Its relatively compact action space provides a complementary setting for examining whether adaptive inference can terminate early and reduce redundant observations without sacrificing recognition accuracy.

DailyDVS-200. DailyDVS-200 (Wang et al., 2024a) is a large-scale real-world benchmark containing more than 22,000 recordings from 47 subjects across 200 daily-action categories. It covers fine-grained, limb-level, and whole-body activities collected under diverse viewpoints, illumination conditions, backgrounds, camera motions, and subject distances. Its released split files contain 14,773 training, 3,735 validation, and 4,093 test entries. We train on the official training list and report final recognition results on the official test list. Its large label space and substantial real-world variation test whether budget-decoupled observation remains effective for complex and fine-grained action recognition.

Together, DVSGesture evaluates efficient inference on compact gestures, THU-EACT-50-CHL emphasizes heterogeneous and challenging action streams, and DailyDVS-200 examines scalability to large-scale fine-grained daily activities. These settings also place EventVista in the broader line of direct space–time event-cloud and event-point recognition methods (Wang et al., 2019; Ren et al., 2023; 2024; 2025).

A.2 Candidate-pool construction

For each recording, we first remove non-finite events and construct a normalized coordinate copy solely for outlier detection. The resulting inlier mask is then applied to the original events. For every retained event, we compute the integer voxel coordinate $\lfloor (t, x, y)/\mathbf{s} \rfloor$, where division and flooring are applied element-wise and $\mathbf{s}$ is the dataset-specific voxel size, and group identical voxel coordinates by sorting their FNV hash keys.

During training, one authentic source event is sampled uniformly from each occupied voxel. The random generator is determined jointly by the global seed, sample index, and a split-specific offset. During validation and testing, the retained representative is the first member of each hash-sorted group. Because `np.argsort` is called without requesting a stable sorting algorithm, this choice is deterministic in the current NumPy environment but has no guaranteed temporal, spatial, or original source-order semantics. The canonical preprocessors for all three datasets follow the same training-random and evaluation-first-member rule.

Table 6: Dataset-specific candidate-pool construction. In every dataset, training selects a seeded uniform source event from each occupied voxel, whereas validation and testing retain the first member of the hash-sorted group. All representatives are authentic source events. Evaluation is deterministic in the current NumPy environment, although the retained member has no guaranteed temporal or source-order interpretation. Temporal voxel sizes reflect dataset-specific timestamp units. FPS denotes farthest-point sampling.

Dataset	Voxel size (t, x, y)	(N_c, N_t) (10^3)	Post-voxel capacity rule
THU-EACT-50-CHL	$(5000, 1, 1)$	$(48, 24)$	Open3D FPS to N_c; seeded no-replacement sampling only if FPS is unavailable or fails.
DVSGesture	$(0.005, 1, 1)$	$(48, 24)$	Deterministic no-replacement capping to N_c.
DailyDVS-200	$(5000, 1, 1)$	$(48, 24)$	Deterministic no-replacement capping to N_c.

Table 7: Candidate cardinality after outlier filtering, voxel reduction, and candidate capping. The short THU-EACT-50-CHL recordings motivate authentic variable-length processing.

Dataset	Split	Recordings	Min.	Median	Max.	$\#\{\lvert\mathcal{C}\rvert < 24\,\mathrm{K}\}$
THU-EACT-50-CHL	Train	1,786	1,058	48,000	48,000	519
THU-EACT-50-CHL	Test	544	583	29,908	48,000	234
DVSGesture	Train	1,077	31,151	48,000	48,000	0
DVSGesture	Test	264	48,000	48,000	48,000	0
DailyDVS-200	Train	14,773	16,058	48,000	48,000	14
DailyDVS-200	Test	4,093	25,587	48,000	48,000	0

Consequently, every voxel representative is an authentic recorded event: we do not use voxel centers, coordinate averages, temporal medians, earliest or latest events, or reliability-based selection.

For THU-EACT-50-CHL, if the number of voxel representatives exceeds N_c, Open3D farthest-point sampling reduces them to N_c. Seeded sampling without replacement is used only as an exception fallback when Open3D cannot be imported or the FPS operation raises an error. The retained coordinates are then normalized per sample and standardized using the global training-set statistics. Short recordings are neither padded nor duplicated.

As an implementation check, replaying the first capped THU test recording (sample 5) reduced 92,095 raw events to 75,060 inliers and then to 48,000 candidates. The regenerated $48{,}000 \times 3$ array exactly matched the stored array (`array_equal=true`, maximum absolute difference 0.0).

Table 6 summarizes the dataset-specific preprocessing and capacity rules, while Table 7 reports the resulting candidate cardinalities on the official data splits.

A.3 Quota rounding and refill

For a dense candidate pool, the real-valued temporal-stratum quota is $\widetilde{q}_m = N_t|\mathcal{C}_m|/|\mathcal{C}|$. We first assign $\lfloor \widetilde{q}_m \rfloor$ to every non-empty stratum, then distribute the remaining slots in descending fractional-remainder order, breaking ties by temporal-stratum index. A quota is capped by the number of available events in that stratum; any unused slots are repeatedly redistributed to later strata with remaining capacity.

Within each stratum, a deterministic permutation is indexed by dataset, split, recording identifier, view-bank seed, and stratum index. Successive views use circularly shifted slices. Events are not repeated within one view. When previously unobserved support is exhausted, the slice wraps within the same permutation; thus later views may overlap earlier views, with each view drawing from strata assigned a positive quota. If $|\mathcal{C}| \leq N_t$, the complete pool is processed once and no refill is performed.

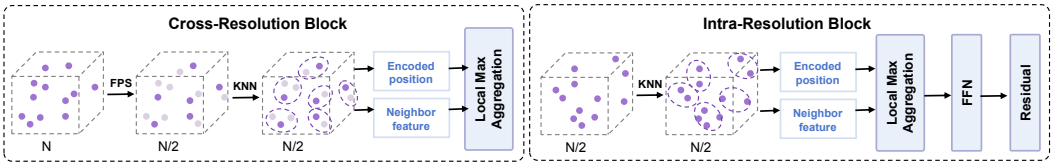

Figure 6: Local blocks of the shared point encoder. Cross-resolution aggregation transfers features from a denser point set to sampled centers. Intra-resolution aggregation refines features among centers with a residual update.

A.4 Shared Point Encoder

The shared encoder follows established hierarchical sampling and local aggregation designs (Qi et al., 2017b; Qian et al., 2022; Lin et al., 2023). We describe its stages and batching to specify the feature extractor used within Coverage-Aware Runtime Observation.

The encoder contains nine stages with output widths $32, 32, 64, 64, 128, 128, 256, 256,$ and 512. Each stage downsamples the points by a factor of two using farthest-point sampling and uses $K_{\mathrm{nn}} = 24$ neighbors in both local aggregation operations. Let $\mathbf{u}_i^\ell$ and $\mathbf{h}_i^\ell$ denote the coordinate and feature of point i at resolution ℓ. Cross-resolution aggregation is written as

$$\widetilde{\mathbf{h}}_i^\ell = \max_{j \in \mathcal{N}_{\mathrm{cross}}^\ell(i)} \Phi_\ell \left(\mathbf{h}_j^{\ell-1}, \mathbf{u}_j^{\ell-1} - \mathbf{u}_i^\ell \right), \tag{7}$$

where $\mathcal{N}_{\mathrm{cross}}^\ell(i)$ contains neighbors in the preceding resolution and Φ_ℓ is a learned mapping of neighbor features and relative coordinates. The maximum is taken channel-wise.

A same-resolution neighborhood then provides a residual update:

$$\mathbf{h}_i^\ell = \widetilde{\mathbf{h}}_i^\ell + \Psi_\ell \left(\max_{j \in \mathcal{N}_{\mathrm{intra}}^\ell(i)} \Gamma_\ell \left(\widetilde{\mathbf{h}}_j^\ell, \mathbf{u}_j^\ell - \mathbf{u}_i^\ell \right) \right), \tag{8}$$

where Γ_ℓ transforms local features and relative coordinates and Ψ_ℓ projects the pooled update.

For a batch of views, valid points are concatenated without padding. At resolution ℓ, cumulative offsets $o_b^\ell = \sum_{j=1}^{b} n_j^\ell$, with $o_0^\ell = 0$, identify the end of each view, where n_j^ℓ is its current point count. Sampling and neighborhood queries respect these boundaries, and offsets are updated after downsampling. For projected final-stage point features $\widehat{\mathbf{h}}_i$, the descriptor of view b is

$$\mathbf{g}_b = \max_{o_{b-1}^L < i \le o_b^L} \widehat{\mathbf{h}}_i. \tag{9}$$

The 512-D descriptor is passed to the shared classifier. Each runtime view therefore yields one logit vector for recording-level integration.

A.5 Training and Controlled Ablations

Training draws one stochastic stratified view per recording and iteration. The within-stratum selection changes across epochs, so the shared encoder sees different retained subsets without multi-view training cost. Inference uses a fixed ordered view bank.

The main-paper controls change candidate retention, runtime-view construction, inference policy, or batch execution while retaining the same encoder architecture. *w/o Candidate Preservation* sets $N_c = N_t = 24\,\mathrm{K}$ before learning. *w/o Runtime Observation* replaces the ordered stratified views with independently sampled random $24\,\mathrm{K}$ subsets of the same $48\,\mathrm{K}$ pool, without replacement within each subset. *w/o Adaptive Integration* evaluates all five ordered views and averages their logits without early stopping.

For *Per-recording execution*, the DataLoader returns a list of variable-length tensors instead of concatenating points from multiple recordings. Each tensor passes independently through the same nine-stage encoder, point-wise projection, and global max pooling to obtain a 512-D view descriptor. The descriptors are stacked and processed by the same classification head.

Table 8: Controls for the number of views used during training and the inclusion of event polarity. All rows use the same adaptive mean-logit inference with at most $K_{\max} = 5$ views; only the training policy or input features are changed. Train cost denotes relative forward/backward computation.

Input	Training views	Train cost	THU-EACT-50-CHL	DVSGesture	DailyDVS-200
(t, x, y)	1 stochastic	1.0×	70.40	99.35	50.68
(t, x, y)	2 stochastic	2.0×	67.53	98.38	49.79
(t, x, y)	3 stochastic	3.0×	67.49	99.39	49.80
(t, x, y, p)	1 stochastic	1.02×	62.74	95.41	46.92

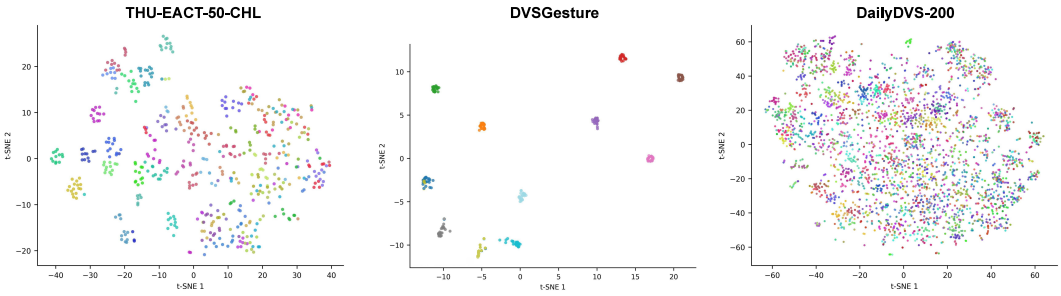

Figure 7: t-SNE projections of EventVista's final recording-level representations on the official evaluation splits of THU-EACT-50-CHL, DVSGesture, and DailyDVS-200. Each point denotes one recording, and colors encode the ground-truth action categories. The two axes are t-SNE coordinates and have no direct semantic meaning.

This control retains the 48 K candidate pool, 24 K per-view budget, authentic cardinalities of short recordings, encoder architecture, channel widths, parameter count, classification head, training recipe, and adaptive inference. It removes multi-sample concatenation, multi-sample cumulative offsets, and the corresponding batched FPS, KNN, and pooling operations. The single terminal count $[N]$ required by the single-sample pointops interface is retained as a length argument. Thus, this is a batch-execution control with the encoder preserved.

Implementation environment. All experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory. All models were trained for 350 epochs with a batch size of 8. The implementation uses Python 3.8.20, PyTorch 2.1.0, CUDA 11.8, cuDNN 8.7.0, Open3D 0.19.0, and pointops 1.0.

B Core Controls and Representation Analysis

B.1 Training-view policy and polarity

Table 8 addresses two orthogonal questions. The first three rows vary the number of training views while keeping adaptive inference fixed, directly testing the one-view-training/multi-view-inference gap. The last row adds event polarity as a fourth input feature under the original one-view training schedule.

Multi-view training does not provide consistent gains. Using two training views reduces accuracy by 2.87, 0.97, and 0.89 points on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200, respectively. Three-view training improves DVSGesture by only 0.04 points, while remaining 2.91 and 0.88 points below one-view training on THU-EACT-50-CHL and DailyDVS-200. Adding polarity reduces accuracy by 7.66, 3.94, and 3.76 points, respectively. One stochastic training view with (t, x, y) therefore provides the best overall accuracy–cost trade-off.

B.2 Feature-space visualization

Fig. 7 visualizes the final recording-level representations learned by EventVista using t-SNE (van der Maaten & Hinton, 2008). DVSGesture forms compact and clearly separated clusters, consistent with

Table 9: Top-1 accuracy (%) under retention- and per-view-budget sweeps. Only one budget changes within each block. Bold denotes the default $48\,\mathrm{K}/24\,\mathrm{K}$ operating point; the larger settings show that performance has entered a saturation region.

Sweep	(N_c, N_t) (10^3)	THU-EACT-50-CHL	DVSGesture	DailyDVS-200
Retention budget	$(24, 24)$	65.82	95.02	44.71
	$(36, 24)$	68.72	97.84	48.72
	$(48, 24)$	**70.40**	**99.35**	**50.68**
	$(64, 24)$	70.46	99.38	50.76
Per-view budget	$(48, 12)$	68.91	98.72	49.43
	$(48, 18)$	69.76	99.08	50.18
	$(48, 24)$	**70.40**	**99.35**	**50.68**
	$(48, 32)$	70.48	99.37	50.80

Table 10: Sensitivity to temporal strata and maximum prefix length. Acc. is Top-1 accuracy (%); Views is the mean number of evaluated views. Bold denotes the default $M = 8$, $K_{\max} = 5$ operating point.

		THU-EACT-50-CHL		DVSGesture		DailyDVS-200	
Sweep	Setting	Acc.	Views	Acc.	Views	Acc.	Views
Temporal strata	$M = 4,\ K_{\max} = 5$	65.85	2.69	95.18	2.16	46.31	3.13
	$M = 8,\ K_{\max} = 5$	**70.40**	2.64	**99.35**	2.12	**50.68**	3.08
	$M = 16,\ K_{\max} = 5$	67.28	2.66	93.30	2.14	47.55	3.10
Maximum prefix	$M = 8,\ K_{\max} = 3$	69.72	2.44	98.98	2.05	50.02	2.75
	$M = 8,\ K_{\max} = 5$	**70.40**	2.64	**99.35**	2.12	**50.68**	3.08
	$M = 8,\ K_{\max} = 7$	69.74	2.71	97.38	2.15	49.74	3.17

its constrained 11-class gesture vocabulary and high recognition accuracy. THU-EACT-50-CHL exhibits multiple coherent local clusters together with visible overlap among several categories, reflecting its larger action space and variations in viewpoint, illumination, and action magnitude. DailyDVS-200 produces a substantially denser embedding with partially overlapping neighborhoods because of its 200 fine-grained categories and broader real-world acquisition variation. Nevertheless, local class-consistent structures remain visible across the more challenging benchmarks. These projections provide a qualitative view of the learned representation geometry; t-SNE distances are not treated as a quantitative measure of class separability.

C Budget and Stopping Robustness

C.1 Retention and per-view budgets

Table 9 varies N_c with $N_t = 24\,\mathrm{K}$ and then varies N_t with $N_c = 48\,\mathrm{K}$. All configurations retain the same encoder, training duration, and stopping thresholds. The sweep tests whether accuracy improves as accessible evidence grows and whether the default $48\,\mathrm{K}/24\,\mathrm{K}$ point is near the saturation region rather than an isolated optimum.

C.2 Temporal strata and view limit

Table 10 varies M and $K_{\max}$ separately around the default. Accuracy and average views must be reported together: a larger $K_{\max}$ is only useful if the extra prefix improves difficult recordings without forcing easy recordings to consume it.

C.3 Threshold sensitivity

Unless otherwise specified, we use the default margin and stability thresholds. Table 11 reports neighboring settings and their accuracy–average-view trade-offs.

Table 11: Sensitivity to the stopping thresholds. Accuracy and the average number of evaluated views are reported together. Bold denotes the default operating point used in the reported experiments.

		THU-EACT-50-CHL		DVSGesture		DailyDVS-200	
τ_m	τ_d	Acc.	Views	Acc.	Views	Acc.	Views
0.60	0.10	70.12	2.42	97.23	2.00	48.41	2.89
0.70	0.05	70.39	2.86	99.35	2.30	50.70	3.33
0.70	**0.10**	**70.40**	**2.64**	**99.35**	**2.12**	**50.68**	**3.08**
0.70	0.15	70.22	2.51	99.25	2.02	50.46	2.92
0.80	0.10	69.43	2.92	99.37	2.37	48.71	3.39

Table 12: Top-1 accuracy (%) and mean evaluated views over three independent runs, reported as mean $\pm$ standard deviation.

Dataset	Top-1 (%)	Views
THU-EACT-50-CHL	70.28 ± 0.24	2.66 ± 0.05
DVSGesture	99.29 ± 0.08	2.13 ± 0.03
DailyDVS-200	50.55 ± 0.22	3.10 ± 0.04

Table 13: Recording-level computation under adaptive prefix inference. GFLOPs per recording are computed from single-view computation and the mean number of evaluated views; candidate construction is excluded. The view counts correspond to the default run used for the headline results.

Dataset	GFLOPs/view	Views	GFLOPs/recording
THU-EACT-50-CHL	3.857	2.64	10.18
DVSGesture	6.780	2.12	14.37
DailyDVS-200	6.780	3.08	20.88

C.4 Independent seeds

Table 12 reports three independent runs. Each run changes the model initialization, data-loader order, training-view sampling, and fixed inference view-bank seed together. The same stopping thresholds are used for all runs.

D Recording-Level Computation and Scope

The main-paper figure compares recognition accuracy with parameter count. Table 13 reports network-computation estimates obtained by multiplying the profiled single-view cost by the mean number of evaluated views. Candidate construction is excluded. For variable-length inputs, exact recording-level network computation requires summing the costs of the evaluated views.

D.1 Candidate storage and construction

A dense $48\,\mathrm{K}$ candidate tensor containing `float32` (t, x, y) values occupies $48{,}000 \times 3 \times 4 = 576{,}000$ bytes (0.55 MiB). One `int32` ordering index per event adds at most 0.18 MiB, giving at most 0.73 MiB for the candidate tensor and view-order metadata, excluding framework-object overhead. Short recordings require proportionally less storage.

Candidate construction reads the complete recording once before classification. The reported method is therefore an offline recording-level recognizer rather than a causal streaming recognizer. A streaming deployment would require an online candidate reservoir, causal temporal strata, and a causal stopping policy; these extensions are outside the present scope.

E Additional Qualitative Analysis

Fig. 8 extends the event-space visualization to all three datasets. For each dataset, we present a hard prediction-rescue example and an easy early-termination example. Each view panel displays only events that have not been covered by the preceding views. The fifth view is omitted from this visualization because it introduces no previously unseen event for the selected examples. For hard samples, successive views expose complementary motion evidence. For easy samples, a short prefix is sufficient for a confident and stable prediction, allowing subsequent views to be skipped.

Fig. 9 shows the corresponding prefix-wise prediction trajectories. At prefix length k, class probabilities are obtained by applying softmax to the mean logits of Views 1 through k. The ground-truth class overtakes its strongest competitor at View 3 on THU-EACT-50-CHL, at View 4 on DVSGesture, and at View 2 on DailyDVS-200. All three early-termination examples satisfy the stopping conditions at View 2. Their later predictions are shown only for fixed-$K_{\max}$ auditing and are not evaluated during adaptive inference. Although View 5 is omitted from the event-space visualization because it adds no previously unseen event, its prediction is retained here because the wrapped view is still evaluated for difficult samples and is included in fixed-$K_{\max}$ auditing for early-stopped samples.

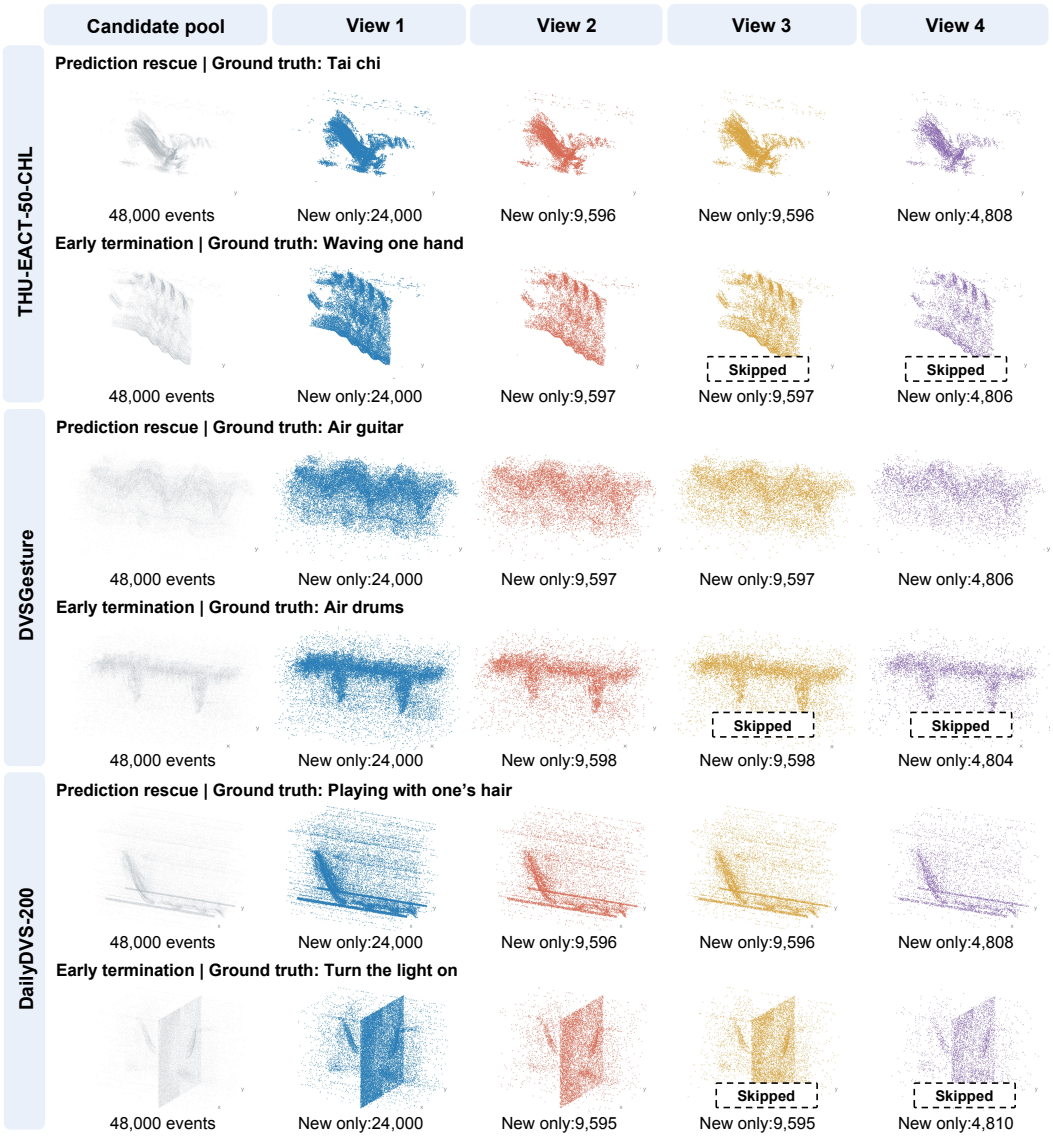

Figure 8: Event-space qualitative analysis on THU-EACT-50-CHL, DVSGesture, and DailyDVS-200. For each dataset, the upper example illustrates prediction rescue and the lower example illustrates early termination. The candidate pools and runtime views contain only authentic source events. Each view panel displays events not covered by any preceding view, and the number below it reports the corresponding newly covered event count. View 5 is omitted because it contains no previously unseen event for these examples. Dashed *Skipped* tags mark views omitted during adaptive inference. Point sets under the *Skipped* tags are shown only to visualize the complete fixed-$K_{\max}$ view bank and are not evaluated during adaptive inference.

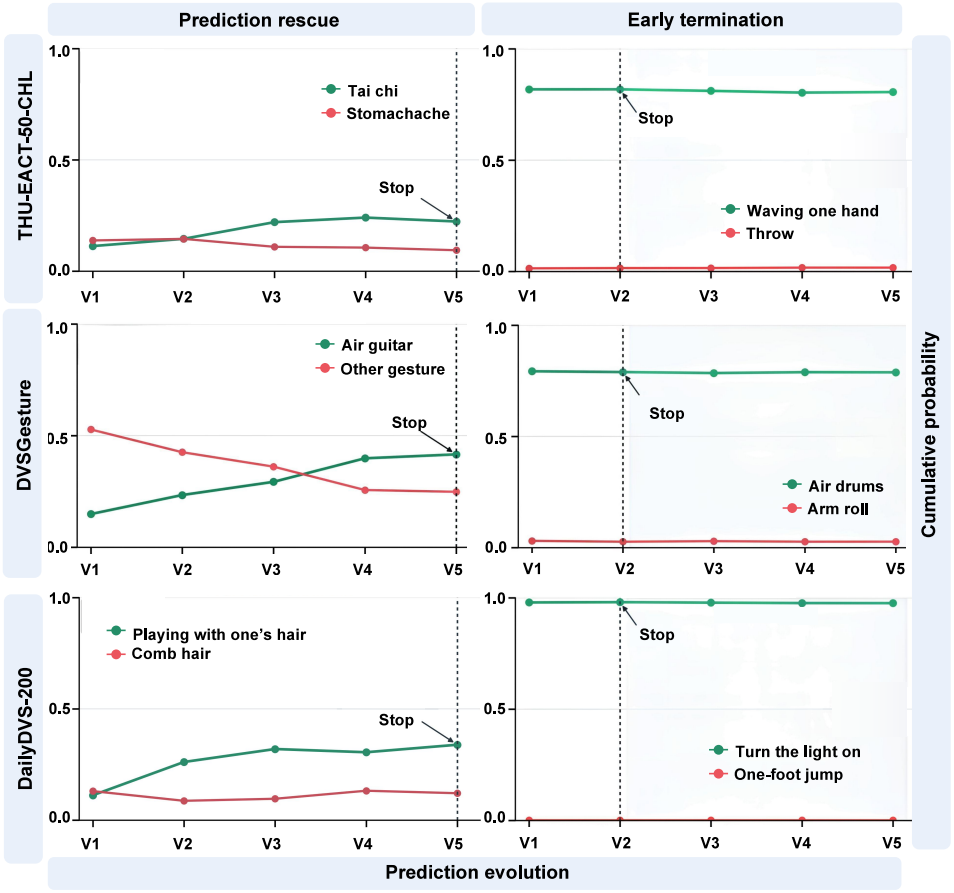

Figure 9: Prefix-wise prediction evolution for the examples in Fig. 8. Each panel compares the ground-truth class with the highest-probability non-ground-truth prediction. Probabilities at prefix length k are computed by applying softmax to the mean logits of Views 1 through k. The dotted vertical line marks the adaptive stopping point. Points following an early stopping decision are displayed only for fixed-$K_{\max}$ auditing. Examples are all drawn from their official test splits.