QuEST: Quality-guided Evidence Selection and Temporal State Control for Event-based Object Detection

Anonymous authors
Paper under double-blind review

Abstract

Event activity reveals where changes occur, but its magnitude alone does not determine their relevance to object detection. Recurrent sparse detectors must therefore decide both which observations receive computation and how strongly they update temporal memory. We introduce QuEST (Quality-guided Evidence Selection and Temporal State Control), a lightweight controller that coordinates these decisions through a shared, detection-supervised quality field. Predicted from explicit event statistics, this field guides fixed-cardinality token selection, modulates selective state-space integration, and regulates cross-frame recurrent writing. Foreground-preserved consistency training regularizes predictions and control signals under changes in background event statistics, while retaining a single recurrent stream at inference. On the full Gen1 streaming test set, QuEST achieves 52.02 AP at 50% token retention, improving over an architecture-matched activity-only baseline by 1.30 AP and 1.88 AP_{75} at identical per-stage scan cardinality, with 1.57% additional parameters and 6.02% additional backbone FLOPs. Without additional pretraining, QuEST also achieves 50.6 AP on 1Mpx and 33.2 AP on eTraM. Its AP advantage over the activity-only baseline persists across all five evaluated event-statistic shifts. These results support shared quality guidance as an effective criterion for coordinating sparse computation and temporal memory in event-based detection.

1 Introduction

Event cameras asynchronously record changes in local log intensity, offering fine temporal resolution, high dynamic range, and reduced motion blur (Gallego et al., 2022). For object detection, however, *event activity measures change, whereas detection requires object-relevant evidence*. Background motion, illumination fluctuations, and sensor noise can generate strong responses, while small or distant objects may produce sparse but informative events. Recurrent sparse detectors must therefore decide both which observations receive computation and how strongly they influence temporal memory.

Recent methods strengthen event-based detection through learned representations and recurrent temporal processing (Gehrig et al., 2019; Perot et al., 2020; Gehrig & Scaramuzza, 2023; Peng et al., 2023; Zhu et al., 2025). Sparse detectors further concentrate computation on selected locations or tokens (Messikommer et al., 2020; Peng et al., 2024; Yang et al., 2025). These advances motivate a complementary question: *how should a detector coordinate the evidence it processes now with the information it retains for later?* Activity provides a useful selection prior, but its magnitude alone does not establish detection relevance. Moreover, token admission determines the evidence available to subsequent state updates, coupling spatial selection with temporal persistence.

We study this coupling under a prescribed scan budget: each stage admits a fixed number of tokens, while their membership and influence on state remain adaptive. This setting evaluates whether detection can improve through better evidence allocation without increasing scan cardinality.

We introduce QuEST (**Qu**ality-guided **E**vidence **S**election and **T**emporal State Control), a lightweight controller built around a shared, detection-supervised quality field. A compact encoder predicts this field from explicit event statistics using box-derived supervision. As illustrated in Figure 1, quality

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.

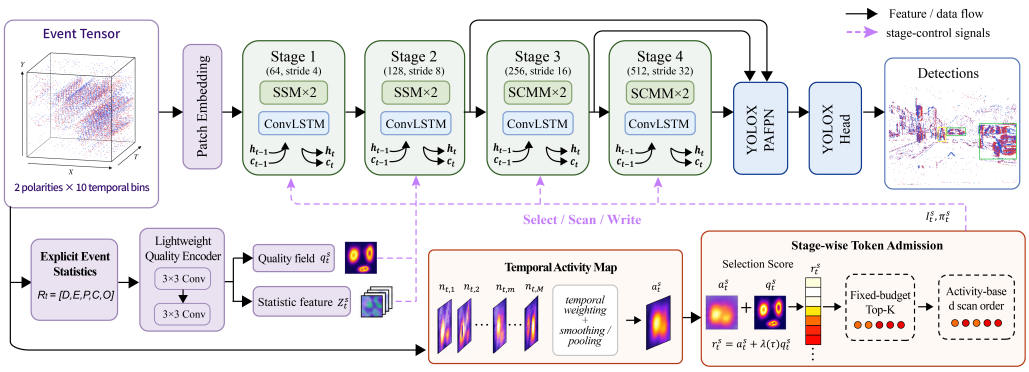


Figure 1: Overview of QuEST. A shared quality field guides fixed-cardinality selection (*Select*), scan integration (*Scan*), and recurrent writing (*Write*) across four stages. Activity provides a selection prior; multi-scale features feed the YOLOX neck and head. Solid arrows denote feature flow and dashed arrows denote stage-control signals.

guides three connected operations: *Select* combines quality with an activity prior for fixed-cardinality Top-K admission; *Scan* modulates state-space integration along the admitted token sequence; and *Write* regulates incorporation into cross-frame recurrent memory. The shared field thus links spatial computation with state control within and across event intervals.

We further introduce foreground-preserved consistency training. Paired inputs vary background event statistics while preserving the original observations within protected foreground regions. Consistency objectives regularize detection outputs and quality-control signals, encouraging stable responses to these changes. The paired branch is used only during training, leaving a single recurrent stream at inference.

On the full Gen1 streaming test set, QuEST achieves 52.02 AP while retaining 50% of tokens at every stage. At identical scan cardinality, it improves over an architecture-matched activity-only baseline by 1.30 AP and 1.88 AP_{75}, with 1.57% additional parameters and 6.02% additional backbone FLOPs. It also improves over its closest sparse SSM predecessor, SMamba, across Gen1, 1Mpx, and eTraM, reaching 50.6 AP on 1Mpx and 33.2 AP on eTraM. Controlled ablations support the contributions of selection, state control, and training regularization, while the detection advantage over the activity-only baseline persists across five evaluated event-statistic shifts.

Our contributions are:

- We introduce a shared, detection-supervised quality controller that coordinates token selection, state-space integration, and recurrent writing under a prescribed scan budget.
- We propose foreground-preserved consistency training that regularizes detection and quality-control signals under background-statistic changes without adding an inference-time branch.
- We demonstrate detection gains at matched scan cardinality, consistent improvements over SMamba across three benchmarks, and maintained advantages over an activity-only baseline under five event-statistic shifts.

2 Related Work

Event representations and recurrent detection. Event-based detection builds object evidence from sparse asynchronous measurements. Event-driven convolutions process activity directly (Cannici et al., 2019; Messikommer et al., 2020), while time surfaces, learned spatiotemporal grids, and recurrent representations provide structured inputs for downstream tasks (Sironi et al., 2018; Gehrig et al., 2019; Cannici et al., 2020). Recurrent detectors accumulate evidence across event intervals (Perot et al., 2020; Kugele et al., 2023). Subsequent methods strengthen spatiotemporal modeling (Gehrig & Scaramuzza, 2023; Peng et al., 2023; Zhu et al., 2025), adapt frame detectors (Torbunov et al., 2025), or exploit self-supervised pretraining (Mohammadi et al., 2025). Hybrid architectures and adaptive event partitioning further enrich evidence formation (Ahmed et al., 2025;

Wang et al., 2025; Sui et al., 2026). QuEST addresses a complementary allocation problem: which evidence should receive a prescribed sparse scan budget, and how strongly should it update recurrent memory?

Sparse processing and evidence allocation. State-space models provide efficient sequence modeling (Gu et al., 2022; Gu & Dao, 2024; Dao & Gu, 2024), with visual extensions adapting their computation to spatial features (Zhu et al., 2024; Liu et al., 2024). For event detection, SAST adapts window and token sparsity to scene activity (Peng et al., 2024), SMamba combines event-guided sparsification with state-space scanning (Yang et al., 2025), and SCSMamba extends sparsification across spatial and channel dimensions (Yang et al., 2026). EventMG applies Mamba to multi-level event graphs (Wu et al., 2025). In frame-based vision, DynamicViT, A-ViT, and SparseViT likewise allocate computation selectively across tokens or windows (Rao et al., 2021; Yin et al., 2022; Chen et al., 2023). QuEST connects sparse admission with temporal persistence: at fixed per-stage cardinality, one detection-supervised quality field coordinates token membership, state-space integration, and recurrent writing. This shared criterion governs both the evidence processed now and its influence on subsequent intervals.

Invariance and foreground-preserved training. Shortcut learning studies show how background context and nuisance statistics can influence learned predictions (Geirhos et al., 2020; Xiao et al., 2021). Environment-based objectives (Arjovsky et al., 2019; Sagawa et al., 2020), feature alignment (Venkataramani et al., 2024), and augmentation (Yun et al., 2019; Hendrycks et al., 2020) encourage stability under input variation. QuEST applies this principle to the statistics consumed by its quality controller. Paired training inputs perturb background statistics while preserving protected foreground observations. Consistency on predictions, features, quality estimates, and write behavior regularizes the same allocation and memory controls used by the single-stream detector at inference.

3 METHOD

QuEST uses one detection-supervised quality field to coordinate token selection and state updates at a prescribed per-stage scan cardinality. The field drives three controls: *Select* determines which tokens enter the sparse scan, *Scan* modulates state-space integration within each event interval, and *Write* regulates recurrent memory across intervals (Figure 1). The detector retains SMamba's stem and four-stage sparse backbone, including its Sparse Spatial Mamba and Spatial-Channel Mixing Mamba (SCMM) blocks (Yang et al., 2025), together with convolutional LSTM (ConvLSTM) recurrence (Shi et al., 2015) and the YOLOX neck and head (Ge et al., 2021). Foreground-preserved consistency training regularizes predictions and control signals under changes in background event statistics.

3.1 EVENT QUALITY ESTIMATION

The quality estimator learns a spatial measure of detection relevance from explicit event statistics (Figure 2(b)). Let $X_t^{p,m}(u,v)$ denote the event tensor at interval t, location (u,v), polarity $p \in \{-,+\}$, and temporal bin $m \in \{1,\dots,M\}$, with $M > 1$. Suppressing spatial and interval indices, define $x_{p,m} = \min\{|X_t^{p,m}|, 255\}$, $n_m = \sum_p x_{p,m}$, $N = \sum_m n_m$, and $N_p = \sum_m x_{p,m}$. With $\pi_m = n_m / \max\{N, \epsilon\}$ and numerical stabilizer $\epsilon > 0$, we compute

$$
\begin{aligned}
D &= \frac{\log(1+N)}{\max\{1, \max_{u,v} \log(1+N)\}}, \qquad E = -\frac{1}{\log M} \sum_m \pi_m \log\big(\max\{\pi_m, \epsilon\}\big), \\
P &= \frac{|N_+ - N_-|}{\max\{N, \epsilon\}}, \qquad C = \sum_m \pi_m \frac{m-1}{M-1}, \qquad O = \mathbf{1}[N > 0].
\end{aligned}
\tag{1}
$$

The five channels describe density, temporal entropy, polarity imbalance, temporal center, and occupancy. Together, they distinguish event patterns beyond activity magnitude; supervision relates these statistics to detection.

The statistics are computed before 4×4 average pooling. The lightweight encoder f_{stat} contains two 3×3 convolutions, each followed by group normalization and SiLU. A 1×1 convolution g_q and sigmoid σ predict the quality field from the encoded features:

$$
Z_t = f_{\text{stat}}(\text{AvgPool}_4(R_t)), \qquad q_t = \sigma(g_q(Z_t)), \qquad R_t = [D, E, P, C, O]. \tag{2}
$$

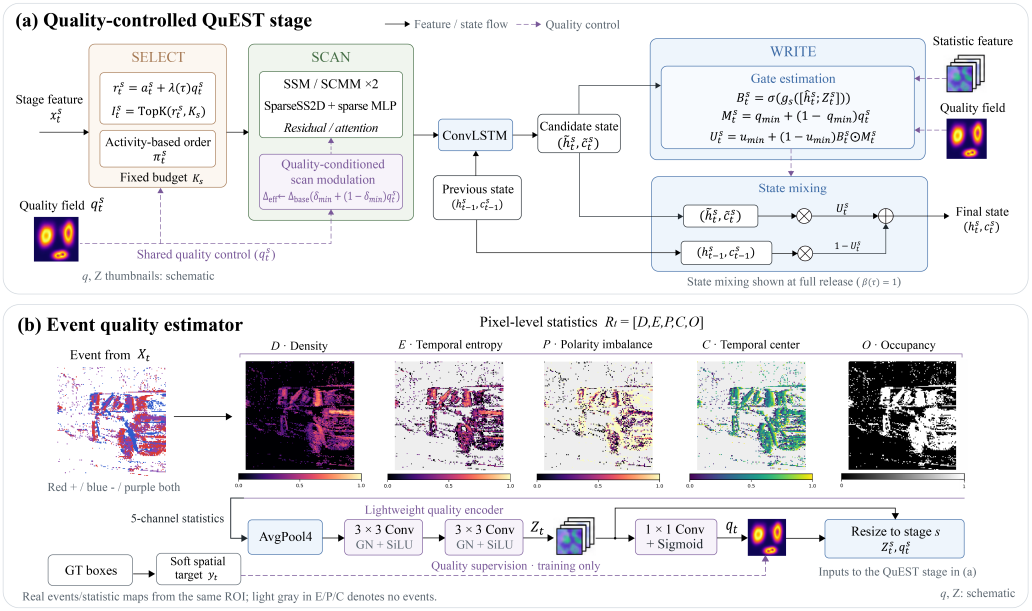


Figure 2: Shared quality estimation and the Select, Scan, and Write controls. **(a)** At each stage, the same quality field guides fixed-cardinality token selection, state-space integration, and recurrent writing. Write mixes the ConvLSTM candidate with the previous state; the illustrated mixing corresponds to $\beta(\tau) = 1$. **(b)** Five event statistics are encoded into statistical features Z_t and quality q_t, which are resized for each stage. Box-derived quality supervision is used only during training. The q and Z thumbnails are schematic.

The encoder is shared across stages. Bilinear resizing produces statistical features Z_t^s and a single-channel quality field q_t^s for stage s. Each value of q_t^s is a quality score at a spatial position. Select, Scan, and Write all use this same field; Write also uses Z_t^s to condition its gate on local event statistics.

Quality supervision uses a box-derived soft target y_t that peaks near object centers and decays with distance relative to box extent:

$$\mathcal{L}_{\text{qual}} = \mathcal{L}_{\text{focal}}(q_t, y_t) + \eta\, \mathbb{E}_{u,v}\left[(q_t - y_t)^2\right], \qquad \eta = 0.10. \tag{3}$$

The focal term addresses foreground–background imbalance; the quadratic term matches soft target values. Direct supervision trains the estimator without gradients through discrete Top-K membership. Appendix B specifies target construction and loss parameters. Inference predicts quality from events alone.

3.2 Select: Fixed-Cardinality Token Selection

Select combines the backbone's activity prior with learned quality to select a fixed number of tokens for scanning. A token is the feature vector at a spatial position of a stage. The activity prior assigns larger weights to later temporal bins:

$$V_t(u,v) = \sum_{p\in\{-,+\}} \sum_{m=1}^{M} m\, |X_t^{p,m}(u,v)|. \tag{4}$$

Pooling, smoothing, resizing, and per-stage normalization produce the activity score a_t^s. For N_s candidate tokens and prescribed token-retention ratio $\rho_s \in (0,1]$, the selection score and index set are

$$r_{t,j}^s = a_{t,j}^s + \lambda(\tau) q_{t,j}^s, \qquad \mathcal{I}_t^s = \mathrm{TopK}(r_t^s, K_s), \qquad K_s = \lceil \rho_s N_s \rceil. \tag{5}$$

Here, j indexes tokens, τ denotes the training step, and $\lambda(\tau)$ controls the quality contribution. Top-K selects the K_s highest-scoring tokens, fixing scan cardinality while quality changes token membership. K/N denotes the realized per-stage token-retention ratio K_s/N_s.

After selection, the backbone constructs the scan sequence using its activity-based ordering π_t^s, as shown in Figures 1 and 2(a). Quality changes token membership through r_t^s; the ordering rule remains activity-based. The selected quality values undergo the same gathering and ordering as their corresponding features. Appendix B specifies activity processing and the quality-weight schedule.

3.3 SCAN: QUALITY-CONDITIONED STATE INTEGRATION

Scan controls how selected features update the state-space model (SSM) state along the scan sequence within each event interval. For selected token j at stage s, let $\zeta_{t,j}^{s,d}$ denote the step-size pre-activation for SSM channel d, and let $b_\Delta^{s,d}$ be its learned bias. The shared quality field modulates the learned integration step:

$$\begin{aligned} \Delta_{\mathrm{base},t,j}^{s,d} &= \mathrm{softplus}\Big(\zeta_{t,j}^{s,d} + b_\Delta^{s,d}\Big), \\ \Delta_{\mathrm{eff},t,j}^{s,d} &= \Delta_{\mathrm{base},t,j}^{s,d}\left[\delta_{\min} + (1-\delta_{\min})q_{t,j}^s\right], \qquad \delta_{\min} = 0.5. \end{aligned} \tag{6}$$

For a fixed base step, lower quality shortens the integration step, while higher quality approaches the learned base dynamics. For a diagonal SSM transition matrix A with negative entries, a smaller step moves $\exp(A\Delta_{\mathrm{eff}})$ toward the identity, retaining more of the preceding scan state's contribution. The positive floor prevents a zero step. Quality is broadcast across SSM channels and follows feature permutations and reversals, preserving token membership and order.

3.4 WRITE: QUALITY-CONDITIONED RECURRENT WRITING

Write applies the shared quality field to memory updates across event intervals. ConvLSTM first produces candidate hidden and cell states $(\widetilde{h}_t^s, \widetilde{c}_t^s)$ from the current representation and previous recurrent state. A stage-specific convolutional write controller g_s combines the candidate hidden state with statistical features, and quality modulates the resulting write rate:

$$\begin{gathered} B_t^s = \sigma\Big(g_s([\widetilde{h}_t^s; Z_t^s])\Big), \qquad M_t^s = q_{\min} + (1-q_{\min})q_t^s, \\ U_t^s = u_{\min} + (1-u_{\min})B_t^s \odot M_t^s. \end{gathered} \tag{7}$$

Here, $[\cdot;\cdot]$ denotes channel concatenation and $\odot$ denotes elementwise multiplication. B_t^s is the base write gate, and the single-channel quality factor M_t^s is broadcast across channels. The entries of the resulting write gate U_t^s are write rates: they depend jointly on candidate content, statistical features, and the quality field shared with Select and Scan. We use $q_{\min} = 0.20$ and $u_{\min} = 0.10$, and cap U_t^s at 0.995 before mixing.

Let $S_t^s = (h_t^s, c_t^s)$ denote the recurrent state pair and $\widetilde{S}_t^s = (\widetilde{h}_t^s, \widetilde{c}_t^s)$ its candidate. Applying the same write gate to both components gives

$$\widehat{S}_t^s = (1 - U_t^s) \odot S_{t-1}^s + U_t^s \odot \widetilde{S}_t^s. \tag{8}$$

Smaller write rates preserve more of the previous state; larger rates incorporate more of the current candidate. During training, a release factor $\beta(\tau) \in [0,1]$ introduces this mixing progressively:

$$S_t^s = (1-\beta(\tau))\widetilde{S}_t^s + \beta(\tau)\widehat{S}_t^s. \tag{9}$$

The update uses the conventional ConvLSTM candidate at $\beta = 0$ and quality-controlled mixing at $\beta = 1$, as illustrated in Figure 2(a). The hidden state feeds the feature hierarchy, and both states persist across intervals. Scan and Write thus apply the same relevance signal at two distinct points: integration along the selected token sequence and recurrent updates between event intervals. Appendix B provides controller initialization and activation schedules.

3.5 FOREGROUND-PRESERVED CONSISTENCY TRAINING

We train the detector and quality-guided controls to produce consistent responses when background event statistics change while foreground observations remain fixed. Let $\mathcal{M}_t$ denote the union of annotated boxes expanded by a small margin and broadcast over polarity and temporal channels. This binary mask defines the protected foreground regions. The paired view is

$$X_t^{\mathrm{cf}} = \mathcal{M}_t \odot X_t + (1 - \mathcal{M}_t) \odot T_\xi(X_t), \tag{10}$$

where T_ξ combines density scaling, spatial dropout, nonnegative noise, temporal smoothing, and polarity jitter; ξ denotes the sampled perturbation parameters. Masked composition restores the original tensor values exactly inside the protected regions, preserving the observations associated with the detection targets.

The two streams share detector parameters. The original stream supplies stop-gradient targets, and its corresponding detached recurrent context initializes the paired branch. The consistency objective is

$$\mathcal{L}_{\text{pair}} = \lambda_p \mathcal{L}_{\text{pred}} + \lambda_f \mathcal{L}_{\text{feat}} + \lambda_q \mathcal{L}_{\text{q-cons}} + \lambda_g \mathcal{L}_{\text{gate}}. \tag{11}$$

$\mathcal{L}_{\text{pred}}$ aligns boxes and detection probabilities using the original stream's detection confidence as weights. Over perturbed background regions, $\mathcal{L}_{\text{feat}}$ compares channel-normalized multi-scale features, and $\mathcal{L}_{\text{q-cons}}$ compares quality fields. $\mathcal{L}_{\text{gate}}$ aligns each example's mean write rate, averaged across spatial positions, channels, and stages. The objective therefore regularizes both detection outputs and the quality and write signals that control token selection and state updates.

The complete objective on labeled frames is

$$\mathcal{L} = \mathcal{L}_{\text{YOLOX}} + \lambda_{\text{qual}} \mathcal{L}_{\text{qual}} + \mathcal{L}_{\text{pair}} + \lambda_{\text{env}} \mathcal{L}_{\text{env}}, \tag{12}$$

where $\mathcal{L}_{\text{YOLOX}}$ is the detection loss, and λ_{qual} and λ_{env} weight quality supervision and environment alignment, respectively. The paired consistency loss $\mathcal{L}_{\text{pair}}$ is applied at scheduled paired-training steps. The environment alignment loss $\mathcal{L}_{\text{env}}$ aligns group means of pooled detector representations across within-batch partitions defined by log total event magnitude, peak-bin fraction, and polarity imbalance. In the ablations, training regularization (Reg.) comprises $\mathcal{L}_{\text{pair}}$ and $\lambda_{\text{env}} \mathcal{L}_{\text{env}}$; quality supervision is shared by all quality-enabled variants. Appendix B gives the hyperparameter values and training schedules, and Appendix B.4 details the perturbations and regularization losses.

All auxiliary objectives and paired operations are training-only. At inference, the quality estimator guides Select, Scan, and Write within a single recurrent stream.

4 Experiments

We evaluate QuEST on three benchmarks and examine Select, Scan, Write, and training regularization through ablations, foreground–background diagnostics, and event-statistic shifts.

4.1 Experimental Setup

Datasets and evaluation. We evaluate on Gen1, 1Mpx (Gen4), and eTraM (Verma et al., 2024). Inputs contain ten temporal bins per polarity over 50-ms intervals. Gen1 uses 256×320 inputs; 1Mpx and eTraM use 384×640 inputs after downsampling and padding. During streaming evaluation, recurrent states persist across consecutive input chunks and reset at recording-sequence boundaries. We report COCO-style average precision (AP), averaged over intersection-over-union (IoU) thresholds from 0.50 to 0.95 in steps of 0.05. AP_{50} and AP_{75} use fixed IoU thresholds; AP_S, AP_M, and AP_L evaluate small, medium, and large objects. Appendices B and C.1 specify preprocessing, recurrent sequence lengths, and evaluation.

Training and ablations. Unless otherwise stated, each stage retains $K_s = \lceil 0.5 N_s \rceil$ tokens. The Gen1 ablations share the backbone, neck, head, and scan cardinality. Quality-enabled variants share the estimator and quality supervision, differing in Select, Scan, Write, and training regularization (Reg.): paired consistency and environment alignment (Section 3.5). Training uses the recurrent YOLOX protocol, a one-cycle learning-rate schedule, and FP16 computation (Appendix B).

Computation and runtime. Parameter counts cover the full detector. We report mean backbone FLOPs directly counted over the first 1,000 Gen1 test frames, excluding the neck, detection head, and non-maximum suppression (NMS). Inference time is reported as runtime and covers the backbone, neck, and head, excluding NMS. The Activity-only–QuEST comparison uses paired steady-state runtime measurements on an NVIDIA RTX A6000 with batch size 1 and FP16 autocast. Appendix D.2 provides the computation and runtime measurement procedures and the inference-time scan-budget study.

Table 1: Detection AP (%) across three benchmarks; parameters refer to Gen1. †Self-supervised pretraining; ‡staged adaptation; *estimated parameter count. Bold/underlined AP marks the best/second-best results among methods without † or ‡, including ours.

Method	Backbone	Params (M)	Gen1 AP	1Mpx AP	eTraM AP
RED (Perot et al., 2020)	CNN+RNN	24.1	40.0	43.0	–
ASTMNet (Li et al., 2022)	CNN+RNN	$> 100^*$	46.7	48.3	–
ERGO-12 (Zubić et al., 2023)	Transformer	59.6	50.4	40.6	–
RVT-B (Gehrig & Scaramuzza, 2023)	Transformer+RNN	18.5	47.2	47.4	29.5
GET-T (Peng et al., 2023)	Transformer+RNN	21.9	47.9	48.4	–
SAST-CB (Peng et al., 2024)	Transformer+RNN	18.9	48.2	48.7	30.0
S5-ViT-B (Zubić et al., 2024)	Transformer+SSM	18.2	47.7	47.8	29.3
SMamba (Yang et al., 2025)	SSM+RNN	16.1	50.4	<u>49.3</u>	32.6
SATE (Zhu et al., 2025)	CNN+RNN	26.4	**52.7**	49.1	<u>33.0</u>
EvRT-DETR-B (Torbunov et al., 2025)‡	Transformer+RNN	57.1	52.7	50.1	–
TESPEC (Mohammadi et al., 2025)†	Recurrent detector	46.7	51.6	50.6	–
QuEST (Ours)	SSM+RNN	16.3	<u>52.0</u>	**50.6**	**33.2**

4.2 Comparison with Existing Methods

Table 1 compares QuEST with existing detectors on Gen1, 1Mpx, and eTraM. QuEST achieves 50.6 AP on 1Mpx and 33.2 AP on eTraM, the highest values among the methods without additional pretraining or staged adaptation in the table. Relative to SMamba, its closest sparse SSM predecessor, AP increases from 50.4 to 52.0 on Gen1, from 49.3 to 50.6 on 1Mpx, and from 32.6 to 33.2 on eTraM. On Gen1, QuEST reaches 52.0 AP with 16.3M parameters, compared with 48.2 AP and 18.9M parameters for SAST-CB, and 47.2 AP and 18.5M for RVT-B. These results combine improved detection with fewer parameters; the architecture-matched ablations isolate the contribution of quality-guided control.

Comparison sources. ASTMNet results and its estimated size follow Gehrig & Scaramuzza (2023). ERGO-12 uses augmented-training results from Zubić et al. (2023) and the parameter count from Zubić et al. (2024). The eTraM results for RVT-B, SAST-CB, and S5-ViT-B follow Yang et al. (2025).

4.3 Ablation Studies

We ablate Select, Scan, Write, and training regularization on Gen1 at 50% token retention in every stage. The architecture-matched *Activity-only* baseline shares the backbone, neck, and head with QuEST, but uses activity scores for selection and omits the quality estimator, quality-conditioned controls, and their auxiliary objectives. *QuEST w/o Reg.* enables all three controls while omitting the training regularization in Section 3.5.

Table 2: Gen1 ablations at $K/N = 0.5$. **(a)** Accuracy and inference cost; FLOPs cover the backbone, whereas runtime also includes the neck and head. Baseline denotes Activity-only. **(b)** Ablations of Select (Sel.), Scan, Write, and training regularization (Reg.). Quality-enabled variants share the estimator and quality supervision. Gains use unrounded results.

(a) Accuracy and cost

Metric	Baseline	QuEST
AP	50.72	**52.02**
AP_{75}	52.76	**54.64**
Params (M)	16.070	16.322
FLOPs (G)	2.576	2.731
Runtime (ms)	20.51	22.05
K/N	0.50	0.50

(b) Component ablations

Variant	Sel.	Scan	Write	Reg.	AP	AP_{75}
Activity-only	–	–	–	–	50.72	52.76
Select	✓	–	–	–	51.14	53.28
Scan + Write	–	✓	✓	–	51.25	53.87
Select + Scan	✓	✓	–	–	51.43	54.30
QuEST w/o Reg.	✓	✓	✓	–	51.86	54.14
QuEST	✓	✓	✓	✓	**52.02**	**54.64**

Overall improvement. QuEST improves AP from 50.72 to 52.02 and AP_{75} from 52.76 to 54.64 (Table 2(a)). The 1.88-point AP_{75} gain extends to a stricter localization threshold, while AP_S and AP_M increase by 1.04 and 1.26 points. All gains use identical per-stage scan cardinality. Appendix D.1 reports the complete metrics.

Select, Scan, and Write. Select reaches 51.14 AP, improving over Activity-only by 0.42 points. Adding Scan raises AP to 51.43 and AP_{75} to 54.30, gains of 0.29 and 1.01 points. Enabling Write as well gives QuEST w/o Reg. at 51.86 AP, a further 0.43-point gain (Table 2(b)).

Scan + Write achieves 51.25 AP with activity-based selection, improving over Activity-only by 0.53 points. Combining all three controls improves over Scan + Write by 0.62 AP and over Select by 0.72 AP, with the estimator, quality supervision, and scan cardinality fixed. This supports complementary benefits from quality-guided selection and state control.

Training regularization. Adding training regularization to QuEST w/o Reg. improves AP by 0.16 and AP_{75} by 0.50, giving QuEST the highest AP and AP_{75} among the evaluated ablations. The inference architecture is unchanged; Section 4.5 evaluates the same regularization under event-statistic shifts.

Inference cost and scan budget. The 1.30-point AP gain requires 0.253M additional parameters (1.57%) and 0.155G additional backbone FLOPs (6.02%). Both methods use 0.108134G for selective scanning, included in backbone FLOPs. Paired runtime is 22.05 ms for QuEST and 20.51 ms for Activity-only. SMamba uses 2.404G and 24.84 ms under the same FLOPs and local runtime protocols, respectively.

Using the same QuEST checkpoint and changing only the inference-time token-retention ratio, 50% token retention achieves 52.02 AP, compared with 52.06 AP at 75% token retention. The 50% setting uses one-third fewer tokens and 8.79% lower backbone FLOPs at nearly the same AP (Appendix D.2).

4.4 Quality-Guided Selection and Recurrent Writing

We examine the quality field q_t^s and write gate U_t^s using foreground–background statistics at each stage. Foreground (FG) consists of stage-grid locations inside the union of the original annotated boxes; background (BG) consists of the remaining locations. Q_{FG} and Q_{BG} denote the regional mean quality scores, while G_{FG} and G_{BG} denote the regional mean write rates obtained by first averaging U_t^s across channels. These means pool regional sums and pixel counts across all labeled test frames, giving each regional grid location equal weight. We report quality contrast $\Delta Q = Q_{\mathrm{FG}} - Q_{\mathrm{BG}}$ and write-rate contrast $\Delta G = G_{\mathrm{FG}} - G_{\mathrm{BG}}$, omitting stage indices for brevity. Appendix B.3 specifies the masks and aggregation.

Across 470 Gen1 test sequences and 30,605 labeled frames, foreground regions receive higher mean quality scores and write rates at every stage (Table 3). Both contrasts are strongest in the first two stages and remain positive at depth, with $K/N = 0.500$ throughout. This foreground preference accompanies a fixed scan budget.

Table 3: Stage-wise quality and write behavior of QuEST on Gen1.

Metric	Stage 1	Stage 2	Stage 3	Stage 4
ΔQ	0.1479	0.1368	0.1084	0.0701
ΔG	0.1215	0.1051	0.0305	0.0112
K/N	0.500	0.500	0.500	0.500

Positive write-rate contrast indicates greater average weight on the current recurrent candidate in foreground regions (Equation 8). Appendices D.4 and E provide regional means and visualization details, respectively.

Quality-guided token selection. Figure 3 compares stage-1 selection on 1Mpx, with both selection rules retaining 7,680 of 15,360 tokens. Quality guidance increases foreground coverage by 9.1–12.5 percentage points in the illustrated examples, allocating more of the fixed scan budget to annotated objects. The final column shows the corresponding QuEST detections.

4.5 Robustness Evaluation

We evaluate five event-statistic shifts: density scaling, spatial dropout, nonnegative noise, temporal smoothing, and polarity jitter. Each transformation affects the full input tensor, without ground-

truth masks or the foreground restoration used during training; scan cardinality remains unchanged. Appendix C.2 specifies transformation strengths and sampling.

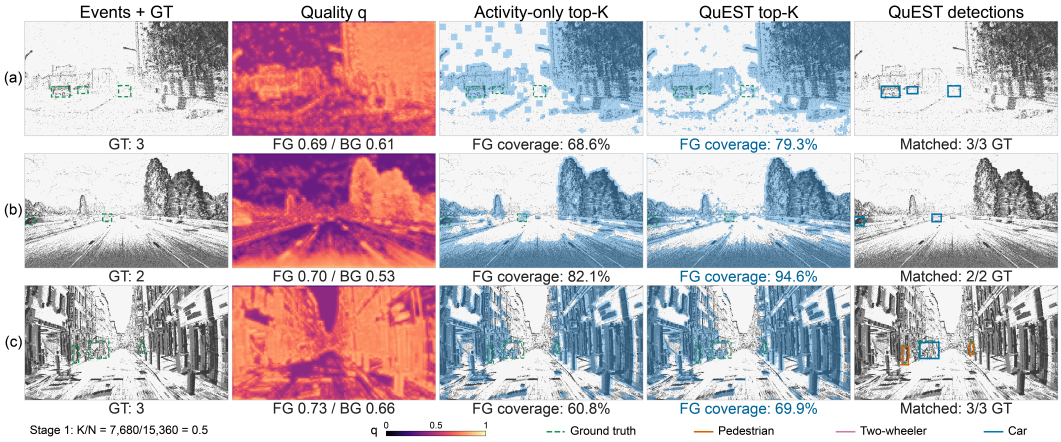


Figure 3: Quality-guided token selection on 1Mpx at $K/N = 0.5$. Columns show events with ground truth, quality, activity-only and QuEST selections, and QuEST detections. Blue shading marks selected tokens; FG coverage is the fraction of valid grid cells within annotated foreground regions that are selected. Quality uses a fixed $[0, 1]$ scale; FG/BG values below its maps are the regional mean quality scores for each example. GT denotes ground-truth annotations. The activity-only reference uses the same activity scores; all detections are from QuEST. Green dashed boxes denote ground truth; solid boxes denote detections colored by class.

Table 4: Gen1 AP under full-input event-statistic shifts at $K/N = 0.5$. Mean AP and Mean Ret. average AP and AP retention, respectively, over the five shifts, excluding Clean. AP retention is $100\,\mathrm{AP}_{\mathrm{shift}}/\mathrm{AP}_{\mathrm{clean}}$ for each variant. QuEST w/o Reg. omits training regularization. Bold marks column maxima; ΔAP is QuEST minus Activity-only.

Variant	Clean	Density	Dropout	Noise	Temporal	Polarity	Mean AP	Mean Ret. (%)
Activity-only	50.72	50.82	50.49	50.72	50.73	50.67	50.69	**99.94**
QuEST w/o Reg.	51.86	**51.90**	51.56	43.72	51.86	51.87	50.18	96.77
QuEST	**52.02**	51.81	**51.78**	**52.17**	**51.95**	**52.02**	**51.95**	99.86
ΔAP	+1.30	+0.99	+1.29	+1.45	+1.22	+1.35	+1.26	–

Detection under event-statistic shifts. QuEST outperforms Activity-only across all five shifts by 0.99–1.45 AP (Table 4). Mean shifted AP rises from 50.69 to 51.95, a 1.26-point gain at the same scan cardinality, extending the detection advantage beyond clean inputs.

Effect of training regularization. With the same inference architecture, QuEST improves over QuEST w/o Reg. from 50.18 to 51.95 mean AP and from 96.77% to 99.86% mean AP retention. The approximately 1.77-point mean AP gain is driven primarily by nonnegative noise, where QuEST achieves 52.17 AP, an improvement of approximately 8.45 points. This comparison supports the combined training regularization in Section 3.5, with its clearest benefit under the evaluated noise condition and no additional inference-time branch. Appendix D.3 reports additional metrics.

5 Conclusion

We introduced QuEST, which uses a shared quality field to coordinate token selection, state-space integration, and recurrent writing under a prescribed scan budget. At 50% token retention, it achieves 52.02 AP on Gen1, improving over an architecture-matched activity-only baseline by 1.30 AP and 1.88 AP_{75} at identical scan cardinality. Consistent gains over SMamba across three benchmarks and maintained advantages under five event-statistic shifts support a common design principle: detection-relevant quality can jointly guide sparse computation and temporal memory.

AI USE STATEMENT

Writing assistance. We used large language models to assist with language polishing and manuscript organization. The authors reviewed and revised the AI-assisted text.

Research ideation and execution. We used large language models to provide feedback on experimental comparisons, assist with interpreting results, and help implement portions of the research code. The authors verified the relevant code through inspection and testing.

The authors take responsibility for the final manuscript, including its methodology, experimental results, and AI-assisted content.

ETHICS STATEMENT

This study evaluates event-based object detection on the existing Gen1, 1Mpx, and eTraM benchmarks. The method can support perception in robotics and transportation. Responsible deployment should account for privacy in recorded scenes and application-specific reliability requirements, with appropriate data handling and validation in the intended operating conditions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the quality estimator, fixed-cardinality token selection, quality-conditioned state control, and training objectives. Section 4 describes the experimental setup and controlled comparisons. Appendix A introduces event-camera sensing and the input representation. Appendix B details preprocessing, training settings, quality supervision, and foreground-preserved training interventions. Appendix C specifies the detection metrics and full-input event-statistic shift protocols. Appendix D reports additional detection metrics, control ablations, scan-budget results, and computation and runtime measurements, including their measurement conditions. Appendix E documents the qualitative visualization protocol and presents spatial and temporal examples.

REFERENCES

Soikat Hasan Ahmed, Jan Finkbeiner, and Emre Neftci. Efficient event-based object detection: a hybrid neural network with spatial and temporal attention. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13970–13979. IEEE, 2025.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1656–1665. IEEE, 2019.

Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *European Conference on Computer Vision*, pp. 136–152. Springer, 2020.

Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. SparseViT: Revisiting activation sparsity for efficient high-resolution vision transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2061–2070. IEEE, 2023.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413. URL https://doi.org/10.1109/TPAMI.2020.3008413.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

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 (ICCV)*, pp. 5633–5643, 2019.

Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13884–13893. IEEE, 2023.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gmrxHFvB.

Alexander Kugele, Thomas Pfeil, Michael Pfeiffer, and Elisabetta Chicca. How many events make an object? improving single-frame object detection on the 1 Mpx dataset. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3913–3922. IEEE, 2023.

Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31:2975–2987, 2022.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/baa2da9ae4bfed26520bb61d259a3653-Abstract-Conference.html.

Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, pp. 415–431. Springer, 2020.

Mohammad Mohammadi, Ziyi Wu, and Igor Gilitschenski. TESPEC: Temporally-enhanced self-supervised pretraining for event cameras. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7782–7793. IEEE, 2025.

Yansong Peng, Yueyi Zhang, Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. GET: Group event transformer for event-based vision. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6015–6025. IEEE, 2023.

Yansong Peng, Hebei Li, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. Scene adaptive sparse transformer for event-based object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16794–16804. IEEE, 2024.

Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16639–16652, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c213877427b46fa96cff6c39e837ccee-Abstract.html.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, pp. 802–810, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html.

Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1731–1740. IEEE, 2018.

Zhipeng Sui, Haiqing Hao, Weihua He, Seng-Hong Lee, and Wenhui Wang. Adaptive spatial-temporal window: Unlocking the potential of event cameras in heterogeneous velocity scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 946–955, 2026.

Dmitrii Torbunov, Yihui Ren, Animesh Ghose, Odera Dim, and Yonggang Cui. EvRT-DETR: Latent space adaptation of image detectors for event-based vision. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9812–9821. IEEE, 2025.

Rahul Venkataramani, Parag Dutta, Vikram Melapudi, and Ambedkar Dukkipati. Causal feature alignment: Learning to ignore spurious background features. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4666–4674, 2024.

Aayush Atul Verma, Bharatesh Chakravarthi, Arpitsinh Vaghela, Hua Wei, and Yezhou Yang. eTraM: Event-based traffic monitoring dataset. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22637–22646. IEEE, 2024.

Xiao Wang, Yu Jin, Wentao Wu, Wei Zhang, Lin Zhu, Bo Jiang, and Yonghong Tian. Object detection using event camera: A MoE heat conduction based detector and a new benchmark dataset. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29321–29330. IEEE, 2025.

Sheng Wu, Lin Jin, Hui Feng, and Bo Hu. EventMG: Efficient multilevel Mamba-graph learning for spatiotemporal event representation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/73d829353fdd9749f9b6cf26c5387f2e-Abstract-Conference.html.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=gl3D-xY7wLq.

Nan Yang, Yang Wang, Zhanwen Liu, Meng Li, Yisheng An, and Xiangmo Zhao. SMamba: Sparse Mamba for event-based object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9229–9237, 2025.

Nan Yang, Zhanwen Liu, Yang Wang, Shangyu Xie, Wenyue Liu, and Xiangmo Zhao. SCSMamba: Spatial-channel sparse Mamba for event-based object detection. *Expert Systems with Applications*, 328:132611, 2026. doi: 10.1016/j.eswa.2026.132611. URL https://doi.org/10.1016/j.eswa.2026.132611.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10799–10808. IEEE, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6023–6032, 2019.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62429–62442. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhu24f.html.

Lin Zhu, Tengyu Long, Xiao Wang, Lizhi Wang, and Hua Huang. Rethinking scale-aware temporal encoding for event-based object detection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/d450dceeacd6083d1d550247377f2320-Abstract-Conference.html.

Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12800–12810. IEEE, 2023.

Nikola Zubić, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5819–5828, 2024.

A Event Cameras and Input Representation

Event generation. An event camera asynchronously reports local changes in log intensity. An event $e_i = (x_i, y_i, t_i, p_i)$ specifies a pixel, timestamp, and polarity $p_i \in \{-1, +1\}$. Under an idealized contrast-threshold model,

$$L(x_i, y_i, t_i) - L_{\text{ref}}(x_i, y_i) = p_i C, \tag{13}$$

where $L = \log I$ and $C > 0$. Events reflect object or camera motion, illumination changes, and noise; event magnitude alone therefore does not determine detection relevance.

Representation used in this work. Our experiments use polarity-separated histograms with hard temporal assignment. Each 50-ms interval produces ten bins per polarity and 20 channels, with negative bins preceding positive bins. Within a nonempty interval, timestamps are normalized by the first and last events:

$$\begin{aligned} b_i &= \min\left\{9, \left\lfloor 10 \frac{t_i - t_{\min}}{\max\{t_{\max} - t_{\min}, \epsilon_t\}} \right\rfloor \right\}, \\ H_{p,b}(x, y) &= \sum_i \mathbf{1}[p_i = p, b_i = b, x_i = x, y_i = y]. \end{aligned} \tag{14}$$

where ϵ_t is one timestamp unit. The stored bin index $b \in \{0, \ldots, 9\}$ corresponds to $m = b + 1$ in Section 3.1; polarity signs identify the same two channels in both notations. Empty intervals yield zero tensors. Separating polarities retains the temporal and polarity statistics used by the quality estimator. The fast histogram constructor accumulates in `uint8` and clips counts to $[0, 10]$. Stored tensors are cast to the model's floating-point type without rescaling or mean/variance normalization. The quality path separately clips magnitudes at 255 without rescaling them.

B Implementation and Training

B.1 Inputs and Optimization

Table 5 gives the dataset-specific settings. Gen1 inputs are zero-padded from 240×304 to 256×320 on the bottom and right. 1Mpx (Gen4) and eTraM inputs are downsampled from 720×1280 to 360×640 using nearest-neighbor sampling (`nearest-exact`), then zero-padded to 384×640. Box coordinates are divided by two to match the downsampled inputs. Streaming and random-sequence training samples are mixed with equal weights.

Table 5: Dataset and training settings. Sequence counts refer to stored preprocessed recordings before streaming subdivision. Training batch sizes are per GPU; k denotes 1,000 steps.

Setting	Gen1	1Mpx	eTraM
Sequences (train/val/test)	1458/429/470	681/128/120	112/25/30
Input tensor $(H \times W)$	256×320	384×640	384×640
Temporal bins per polarity	10	10	10
Recurrent sequence length	21	5	5
Training batch size per GPU	24	24	8
Number of GPUs	1	1	1
Gradient accumulation factor	1	1	1
Evaluation batch size	8	8	8
Peak learning rate	3.4641×10^{-4}	3.4641×10^{-4}	2.0×10^{-4}
Optimization horizon	400k	400k	400k
Selection warm-up	15k	20k	15k
Selection ramp duration	30k	30k	30k

We use AdamW, zero weight decay, gradient clipping at 1.0, and FP16. The peak learning rate follows square-root scaling:

$$\eta_{\max} = 2 \times 10^{-4} \sqrt{B_{\text{eff}}/8}, \qquad B_{\text{eff}} = B_{\text{device}}, \tag{15}$$

for the single-GPU runs without gradient accumulation in Table 5. The one-cycle schedule uses linear annealing, warm-up fraction 0.005, and no momentum cycling. Initial and final learning rates

are $\eta_{\max}/25$ and $\eta_{\max}/10000$. Validation runs every 5,000 training batches; model selection uses validation AP.

Horizontal flipping has probability 0.5; rotation is disabled. Random-sequence zoom is enabled with probability 0.8, with zoom-in/out weights 8:2 and ranges $[1, 1.5]$/$[1, 1.2]$. Streaming zoom-out has probability 0.5 and range $[1, 1.2]$.

B.2 Architecture and Quality Supervision

The four stages have widths $(64, 128, 256, 512)$, strides $(4, 8, 16, 32)$, and two blocks each. Stages 1–2 use Sparse Spatial Mamba blocks, and stages 3–4 use Spatial-Channel Mixing Mamba (SCMM) blocks, with ConvLSTM recurrence at each stage. Stages 2–4 feed the YOLOX neck and head; the neck's feature-pyramid depth multiplier is 0.67.

The statistical encoder f_{stat} receives density, temporal entropy, polarity imbalance, temporal center, and occupancy after 4×4 average pooling of these statistics. Two bias-free 3×3 convolutions map $5 \to 32 \to 16$ channels, each followed by eight-group normalization and SiLU. The quality prediction head g_q has zero-initialized weights and bias 2.0 and is followed by a sigmoid. Both statistical features Z_t and the quality field q_t are resized to each stage by bilinear interpolation with `align_corners=False`; numerical stabilization uses 10^{-6}.

Quality targets. For box i on the quality grid, let $(c_{x,i}, c_{y,i})$ denote its center and (w_i, h_i) its width and height, each clamped below at one grid unit. Set $\sigma_{x,i} = \max\{w_i/6, 0.8\}$ and $\sigma_{y,i} = \max\{h_i/6, 0.8\}$. Let $o_i(u, v)$ be the corresponding unit-peak Gaussian and define $d_{x,i} = \min\{2, 2|u - c_{x,i}|/w_i\}$ and $d_{y,i} = \min\{2, 2|v - c_{y,i}|/h_i\}$. Omitting the interval index t, the soft quality target is

$$y(u, v) = \max_i \sqrt{o_i(u, v) \exp[-\tfrac{1}{2}(d_{x,i} + d_{y,i})]}. \tag{16}$$

It is zero for an empty box set and has no hard box-boundary cutoff. Quality supervision combines soft-target focal loss ($\alpha = 0.75$, $\gamma = 2$) with mean squared error weighted by $\eta = 0.10$ in Equation 3. Its weight in the total objective is $\lambda_{\text{qual}} = 0.02$ (Equation 12).

B.3 Select, Scan, and Write

The activity map V_t sums absolute bin values with temporal weights $1, \ldots, 10$ for each polarity. Processing uses 4×4 average pooling, a first-stage 5×5 Gaussian blur ($\sigma = 2$), and subsequent 2×2 pooling. Normalization by the per-stage spatial maximum, floored at 10^{-6}, produces activity scores a_t^s. Select uses selection scores $r_t^s = a_t^s + \lambda(\tau) q_t^s$, with $K_s = \lceil \rho_s N_s \rceil$ and default prescribed token-retention ratio $\rho_s = 0.5$. The selected tokens retain the backbone's activity-based ordering; stages 1–2 form the ordering map with 4×4 max pooling and nearest-neighbor upsampling. Top-K membership is non-differentiable.

Let τ be the optimizer-step index used in the main text and $[z]_0^1 = \min\{1, \max\{0, z\}\}$. The schedules are

$$\lambda(\tau) = 0.25 \left[\frac{\tau - \tau_{\text{sel}}}{30000} \right]_0^1, \qquad \beta(\tau) = \left[\frac{\tau - 20000}{40000} \right]_0^1, \tag{17}$$

where τ_{sel} is the selection warm-up duration in Table 5. For Scan, quality scores follow token gathering, permutation, and reversal. Omitting stage, interval, token, and channel indices,

$$\Delta_{\text{eff}} = \text{softplus}(\zeta + b_\Delta)(0.5 + 0.5q). \tag{18}$$

The bias and softplus are applied once. Scan modulation activates when $\beta > 0$; the continuous β ramp applies to recurrent writing.

The stage-specific write controller g_s uses 1×1 convolutions with hidden widths $(64, 128, 128, 128)$, eight-group normalization, and SiLU. Its output layer has zero weights and bias 4.0. The quality floor is $q_{\min} = 0.20$, the minimum write rate is $u_{\min} = 0.10$, and the write gate U_t^s is capped at 0.995. The same gate mixes previous and candidate hidden and cell states; missing previous states are zero-initialized.

Diagnostics. Foreground (FG) masks rasterize the union of the original annotated boxes on each stage grid, without the training-time expansion; the remaining positions form the background (BG). Let $\Omega^s_{R,t}$ denote the positions in region $R \in \{\mathrm{FG}, \mathrm{BG}\}$ at stage s and labeled frame t. We average the write gate over its C_s channels at each position, $\bar{U}^s_{t,j} = C_s^{-1} \sum_{c=1}^{C_s} U^s_{t,j,c}$, and accumulate regional sums and pixel counts over the set of labeled frames $\mathcal{T}$:

$$Q^s_R = \frac{\sum_{t\in\mathcal{T}} \sum_{j\in\Omega^s_{R,t}} q^s_{t,j}}{\sum_{t\in\mathcal{T}} |\Omega^s_{R,t}|}, \qquad G^s_R = \frac{\sum_{t\in\mathcal{T}} \sum_{j\in\Omega^s_{R,t}} \bar{U}^s_{t,j}}{\sum_{t\in\mathcal{T}} |\Omega^s_{R,t}|}. \tag{19}$$

Thus Q^s_R is regional mean quality and G^s_R is regional mean write rate; each regional grid location contributes equally across the evaluated labeled frames. We report $\Delta Q^s = Q^s_{\mathrm{FG}} - Q^s_{\mathrm{BG}}$, $\Delta G^s = G^s_{\mathrm{FG}} - G^s_{\mathrm{BG}}$, and the realized token-retention ratio. Appendix D.4 provides the regional means and evaluation coverage.

B.4 Foreground-Preserved Consistency Training

For training only, expand each labeled box by two input pixels and let $\mathcal{M}_t$ be the union mask, broadcast over polarity and temporal channels. The paired view preserves protected foreground observations:

$$X^{\mathrm{cf}}_t = \mathcal{M}_t \odot X_t + (1 - \mathcal{M}_t) \odot T_\xi(X_t). \tag{20}$$

The paired branch starts after 60k optimizer steps with probability 0.25, using the last labeled time index in the batch and the corresponding detached original-stream recurrent context.

Training perturbations use density scaling sampled from $[0.60, 1.40]$, spatial dropout 0.10, and nonnegative noise of amplitude 0.03 times the current sample's mean absolute value, floored at 10^{-6}. Temporal smoothing and polarity jitter each activate with probability 0.5; smoothing uses weights $(0.04, 0.92, 0.04)$, and polarity amplitudes are sampled from $[-0.08, 0.08]$. Dropout masks are shared across channels. Foreground is restored and values are restricted to $[0, 3\max\{1, \|X\|_\infty\}]$; non-finite values are replaced by zero.

Regularization. The paired consistency loss uses the weights λ_p, λ_f, λ_q, and λ_g in Equation 11, with values as follows:

$$\mathcal{L}_{\mathrm{pair}} = 0.01\mathcal{L}_{\mathrm{pred}} + 0.005\mathcal{L}_{\mathrm{feat}} + 0.002\mathcal{L}_{\text{q-cons}} + 0.001\mathcal{L}_{\mathrm{gate}}. \tag{21}$$

Original-stream targets are detached. Prediction consistency combines probability MSE and normalized-box SmoothL1 (relative weight 0.25), weighted by original-stream detection confidence. It retains the union of the top 5% of locations (at least 64, capped by the available count) and locations with detection confidence at least 0.01; retained weights are floored at 0.02. Feature consistency compares channel-normalized features over background, and quality consistency uses background-masked MSE. Empty masks use the unmasked mean. Gate consistency compares per-example mean write rates across channels, spatial positions, and stages. The paired branch has no additional supervised detection loss.

The environment alignment loss $\mathcal{L}_{\mathrm{env}}$ has weight $\lambda_{\mathrm{env}} = 0.001$ in Equation 12. Each of log total event magnitude, peak-bin fraction, and polarity imbalance splits the batch at its median. The objective averages the MSE between high/low group means of normalized, concatenated stage features, skipping empty partitions. It is independent of the paired branch's warm-up and sampling probability. Training regularization (Reg.) comprises $\mathcal{L}_{\mathrm{pair}} + \lambda_{\mathrm{env}}\mathcal{L}_{\mathrm{env}}$; quality supervision remains enabled in all quality-enabled variants.

C Evaluation Protocols

C.1 Streaming Evaluation and Detection Metrics

We use the preprocessed train/validation/test partitions summarized in Table 5, with event tensors and annotations stored in the v2 preprocessing layout. The evaluator maps Gen1 class IDs 0/1 to car/pedestrian, and 1Mpx/eTraM IDs 0/1/2 to pedestrian/two-wheeler/car. Streaming states persist across consecutive input chunks and reset at recording-sequence boundaries. Random training sequences start from reset states, and training states are detached between chunks. Evaluation includes all eligible labeled time indices.

COCO metrics. We use standard COCO matching and 101-point interpolated precision at intersection-over-union (IoU) thresholds $0.50 : 0.05 : 0.95$:

$$\mathrm{AP}_{c,\theta} = \frac{1}{101} \sum_{r \in \{0, 0.01, \ldots, 1\}} P^{\mathrm{interp}}_{c,\theta}(r), \qquad \mathrm{AP} = \frac{1}{10|\mathcal{C}|} \sum_{c \in \mathcal{C}} \sum_{\theta = 0.50:0.05:0.95} \mathrm{AP}_{c,\theta}, \tag{22}$$

where $P^{\mathrm{interp}}(r) = \max_{\tilde{r} \geq r} P(\tilde{r})$, unattained recall levels contribute zero, and $\mathcal{C}$ contains categories with valid ground truth. AP and mAP denote the same quantity here. AP_{50} and AP_{75} fix the IoU threshold and average over classes; reported values are multiplied by 100. Size-specific AP uses COCO area boundaries 32^2 and 96^2, with standard ignore conventions and `maxDets=100`. Areas use the box coordinates supplied to evaluation, including spatial downsampling. Metrics are accumulated over the evaluated stream.

Event-dataset conventions. Predictions are associated with annotations using a ± 50-ms tolerance. Evaluation samples require valid ground truth after filtering; fully empty timestamps are excluded by this wrapper. Both predictions and annotations must have timestamps greater than 0.5 s. Minimum box diagonal and side lengths are 30/10 pixels for Gen1 and 60/20 pixels for 1Mpx/eTraM at native resolution, scaled by two for downsampled evaluation. Initial inputs can still initialize recurrent states.

C.2 Full-Input Robustness Evaluation

Each transformation in Table 6 is evaluated separately on the full event tensor, including foreground and background, without the foreground protection used during training.

Table 6: Full-input event-statistic transformations used for all compared variants. Each row defines a separate condition. Temporal smoothing and polarity jitter are always active in their respective conditions.

Condition	Transformation before range handling
Density scaling	$\widetilde{X} = 1.40X$.
Spatial dropout	$\widetilde{X}_{c,u,v} = X_{c,u,v}\mathcal{D}_{u,v}$, $\mathcal{D}_{u,v} \sim \mathrm{Bernoulli}(0.90)$, shared across channels.
Nonnegative noise	$\widetilde{X} = X + 0.03\mu(X)\Xi$, $\mu(X) = \mathrm{mean}(\lvert X \rvert)$; entries of the noise tensor Ξ are uniform on $[0, 1]$.
Temporal smoothing	$\widetilde{X}_{p,b} = 0.92X_{p,b} + 0.04X_{p,b-1} + 0.04X_{p,b+1}$ within each polarity stack; endpoints are replicated.
Polarity jitter	$\widetilde{X}_{-} = 0.92X_{-}$ and $\widetilde{X}_{+} = 1.08X_{+}$.

For each sample, all conditions finish with

$$c_{\mathrm{clip}}(X) = 3 \max\{\|X\|_\infty, 1\}, \qquad X' = \mathrm{nan_to_num}\big(\mathrm{clip}_{[0, c_{\mathrm{clip}}(X)]}(\widetilde{X})\big), \tag{23}$$

where the bound uses the original tensor and non-finite values are converted to finite values. Density scaling, temporal smoothing, and polarity jitter are deterministic. Dropout masks and noise are sampled independently for each event tensor and evaluation run under the same transformation distributions and strengths. Dropout masks are shared across channels within each time step. All methods use the same sequence order and batch/worker configuration.

Clean AP uses the complete, unperturbed test evaluation. For each variant, AP retention R_j measures AP under shift j relative to its own clean AP. Across five separately evaluated shifts, we report

$$R_j = 100 \frac{\mathrm{AP}_j}{\mathrm{AP}_{\mathrm{clean}}}, \tag{24}$$

$$\text{Mean AP} = \frac{1}{5} \sum_{j=1}^{5} \mathrm{AP}_j, \qquad \text{Mean Ret.} = \frac{1}{5} \sum_{j=1}^{5} R_j. \tag{25}$$

Mean Ret. denotes mean AP retention; both means exclude Clean and use unrounded evaluator outputs.

D Additional Results

D.1 Detection and Control Ablations

Tables 7 and 8 report Gen1 metrics and controls at per-stage $K/N = 0.5$, using Table 2's variant names. Activity-only selects by activity without quality estimation or control. Quality-enabled variants share the estimator and quality supervision; Scan + Write retains activity-based selection. QuEST w/o Reg. enables all three controls; QuEST also uses paired consistency and environment alignment (Appendix B.4).

At matched scan cardinality, QuEST gains 1.30 AP, 1.88 AP_{75}, 1.04 AP_S, and 1.26 AP_M over Activity-only. With the estimator and supervision fixed, AP rises from 51.14 for Select to 51.43 for Select + Scan and 51.86 for QuEST w/o Reg.; Scan + Write reaches 51.25 AP. Training regularization adds 0.16 AP and 0.50 AP_{75} without changing inference architecture. Table 9 gives all three benchmarks.

Table 7: Gen1 detection metrics at per-stage $K/N = 0.5$ (AP %). Size groups follow Appendix C.1; bold marks column maxima.

Variant	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Activity-only	50.72	80.63	52.76	43.02	57.20	50.14
Select	51.14	80.41	53.28	43.88	57.15	48.18
Scan + Write	51.25	80.45	53.87	43.58	57.56	51.17
Select + Scan	51.43	80.65	54.30	43.70	57.94	47.14
QuEST w/o Reg.	51.86	**81.04**	54.14	**44.48**	58.06	**52.25**
QuEST	**52.02**	80.93	**54.64**	44.06	**58.46**	47.99

Table 8: Gen1 control ablations at per-stage $K/N = 0.5$. Reg. combines paired consistency and environment alignment. Quality-enabled variants share quality supervision; a dash denotes a disabled component.

Variant	Select	Scan	Write	Reg.	AP	AP_{75}
Activity-only	–	–	–	–	50.72	52.76
Select	✓	–	–	–	51.14	53.28
Scan + Write	–	✓	✓	–	51.25	53.87
Select + Scan	✓	✓	–	–	51.43	54.30
QuEST w/o Reg.	✓	✓	✓	–	51.86	54.14
QuEST	✓	✓	✓	✓	52.02	54.64

Table 9: Complete QuEST detection metrics (AP %) to four decimals. Main-text comparisons report AP to one decimal.

Dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Gen1	52.0197	80.9308	54.6368	44.0620	58.4647	47.9880
1Mpx	50.6426	82.2145	52.7385	37.2853	55.0969	63.3689
eTraM	33.1791	65.0602	29.4664	17.7454	38.4876	36.7382

D.2 Scan Budget and Efficiency

We evaluate the default $\rho = 0.50$ checkpoint at inference-time token-retention ratios $\rho \in \{0.25, 0.50, 0.75\}$, without retraining or fine-tuning. All other settings follow Appendix C.1. Each stage uses $\rho_s = \rho$, with $(5120, 1280, 320, 80)$ candidates and $K_s = \lceil \rho N_s \rceil$ selected tokens. Table 10 reports detection metrics.

At $\rho = 0.50$, AP is 52.0197 versus 52.0597 at $\rho = 0.75$, using one-third fewer tokens and 8.79% fewer backbone FLOPs. At $\rho = 0.25$, AP is 51.1984 with half the default token count and 9.64% fewer backbone FLOPs. Thus, the default budget retains nearly the AP of 75% token retention with less sparse computation.

Table 10: Gen1 AP (%) from one QuEST checkpoint with only inference-time token-retention ratio varied; no retraining or fine-tuning.

Budget ρ	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0.25	51.1984	80.3025	53.5070	42.6122	58.1187	47.9963
0.50 (default)	52.0197	80.9308	54.6368	44.0620	58.4647	47.9880
0.75	52.0597	80.8104	54.7242	44.0062	58.5854	47.5997

Computation. All settings use the same 16.322454M-parameter detector. We directly count backbone operations over the first 1,000 Gen1 test frames and report mean FLOPs per frame. Table 11 separates scanning from the backbone total, which includes it. Token totals $\sum_s K_s$ sum stage selections without multiplying by block depth or scan directions; backbone FLOPs include all executed backbone operations.

Table 11: Token counts and mean backbone FLOPs for one QuEST checkpoint over the first 1,000 Gen1 test frames. Scan is included in the backbone total.

ρ	(K_1, K_2, K_3, K_4)	$\sum_s K_s$	Scan FLOPs (G)	Backbone FLOPs (G)
0.25	$(1280, 320, 80, 20)$	1700	0.089702	2.467591
0.50	$(2560, 640, 160, 40)$	3400	0.108134	2.730882
0.75	$(3840, 960, 240, 60)$	5100	0.126566	2.994172

Runtime. Paired steady-state measurements use an NVIDIA RTX A6000, batch size 1, and FP16 autocast. Runtime covers the backbone, neck, and detection head forward pass, excluding non-maximum suppression (NMS): 20.51 ms for Activity-only, 22.05 ms for QuEST, and 24.84 ms for SMamba with adaptive token allocation. Runtime and backbone FLOPs have different module coverage.

D.3 Results under Event-Statistic Shifts

Tables 12 and 13 report Activity-only metrics and variant comparisons under the full-input shifts in Appendix C.2. Mean AP and mean AP retention use unrounded outputs, exclude Clean, and follow Equations (24) and (25).

Table 12: Activity-only AP and AP retention (%) under full-input Gen1 shifts.

Condition	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP Ret. (%)
Clean	50.72	80.63	52.76	43.02	57.20	50.14	100.00
Density	50.82	80.80	52.94	43.08	57.25	49.78	100.20
Dropout	50.49	80.45	52.62	42.60	56.97	49.48	99.56
Noise	50.72	80.62	52.75	43.10	57.16	50.45	100.00
Temporal	50.73	80.61	52.70	43.08	57.15	50.32	100.03
Polarity	50.67	80.64	52.48	43.02	57.07	50.16	99.91

Table 13: Gen1 AP under shifts at $K/N = 0.5$. Mean AP and mean AP retention (Mean Ret.) exclude Clean. Bold marks column maxima.

Variant	Clean	Density	Dropout	Noise	Temporal	Polarity	Mean AP	Mean Ret. (%)
Activity-only	50.72	50.82	50.49	50.72	50.73	50.67	50.69	**99.94**
QuEST w/o Reg.	51.86	**51.90**	51.56	43.72	51.86	51.87	50.18	96.77
QuEST	**52.02**	51.81	**51.78**	**52.17**	**51.95**	**52.02**	**51.95**	99.86

Across all five shifts, QuEST gains 0.99–1.45 AP over Activity-only (mean AP: 51.95 versus 50.69). Relative to QuEST w/o Reg., regularization raises mean AP from 50.18 to 51.95 and mean AP retention from 96.77% to 99.86%, with unchanged inference architecture. The largest gain occurs under nonnegative noise (52.17 versus 43.72 AP), supporting regularization under this evaluated condition.

D.4 Stage-Wise Quality and Write Diagnostics

Table 14 gives the regional means behind Table 3, with pixel-weighted aggregation (Appendix B.3) over all 470 Gen1 test sequences: 30,605 labeled frames in 2,996 streaming batches.

Table 14: Gen1 regional mean quality and write rates. FG/BG denote foreground/background; stage indices on Q, G are omitted. K/N is the realized per-stage token-retention ratio.

	Quality			Recurrent writing			
Stage	Q_{FG}	Q_{BG}	ΔQ	G_{FG}	G_{BG}	ΔG	K/N
1	0.622906	0.475028	0.1479	0.598407	0.476951	0.1215	0.500
2	0.610046	0.473232	0.1368	0.529498	0.424414	0.1051	0.500
3	0.578743	0.470361	0.1084	0.451379	0.420878	0.0305	0.500
4	0.550453	0.480356	0.0701	0.443859	0.432676	0.0112	0.500

At fixed scan cardinality, mean quality scores and write rates are higher in foreground regions at every stage, with the strongest contrasts in stages 1–2.

E Qualitative Results

Visualizations use 1,744 eligible labeled frames from 16 evenly spaced 1Mpx test sequences, with up to 300 event intervals per sequence and at least 40 context intervals. Class-aware one-to-one matching uses IoU ≥ 0.5; quality and write-rate maps share $[0, 1]$ and exclude bottom padding.

Figure 3 compares stage-1 QuEST and activity-only Top-K masks using the same activity scores. Both select 7,680 of 15,360 padded-grid tokens. Foreground coverage—the selected fraction of valid grid cells inside eligible ground-truth boxes—increases from 68.6% to 79.3%, 82.1% to 94.6%, and 60.8% to 69.9% in the three examples. All detections are from QuEST.

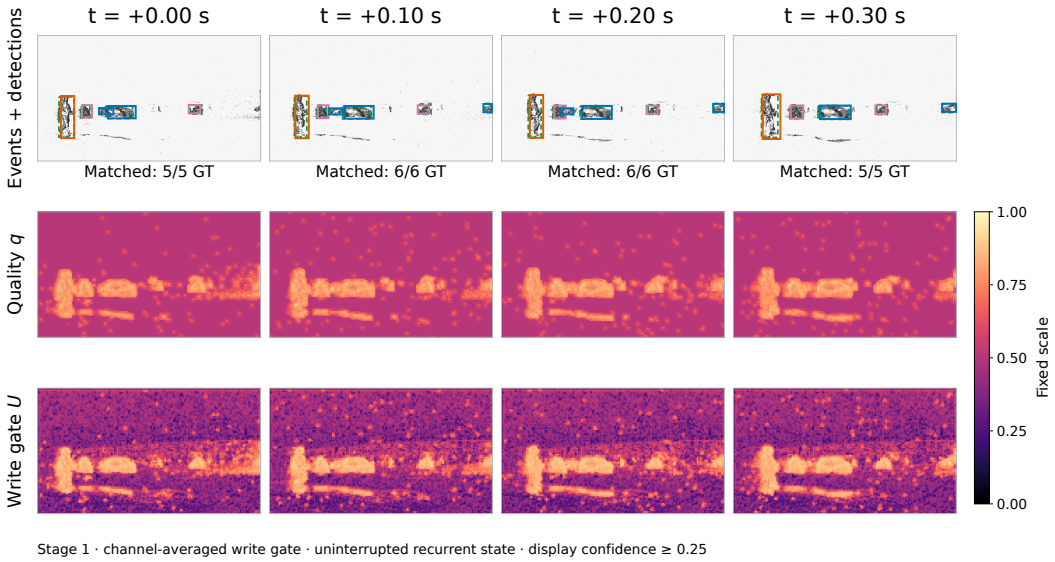


Figure 4: **Quality and recurrent writing over time on 1Mpx.** Four annotated instants from one sequence, approximately 0.10 s apart; times are relative to the first. Rows show detections over events, quality q, and channel-averaged write rates $\bar{U}$. States persist through all intervals from the loader-defined start; all eligible annotations shown are matched. Green dashed boxes mark ground truth; solid orange, pink, and blue boxes denote pedestrians, two-wheelers, and cars, respectively.