Event-VLM: Three-Axis Efficient Inference for Real-time Surveillance Video Understanding

Anonymous Authors

Anonymous Institution
anonymous@eccv2026.org

Abstract. Vision-Language Models (VLMs) are attractive for intelligent surveillance, but their deployment is limited by inference economics: processing hundreds of concurrent streams with full-frame, full-cache decoding is prohibitively expensive. We show that this cost is driven by three orthogonal redundancies: *temporal* (most frames are non-events), *spatial* (most visual tokens are irrelevant background), and *decoding* (auto-regressive KV-cache access becomes memory-bandwidth bound). We present **Event-VLM**, a cascaded framework that allocates computation only *when* events occur, *where* hazards reside, and *which* cache entries matter during generation. Event-VLM combines: (1) *Event-Triggered Gating* with risk-sensitive detection loss for high recall on critical hazards; (2) *Knowledge-Guided Token Pruning* with adaptive dilation for amorphous hazards, reducing visual tokens by 75% without retraining; and (3) *Frequency-Aware Sparse Decoding*, which uses dominant RoPE frequency chunks as a training-free proxy for online token importance and focused KV access. An optional hazard-priority prompting module further improves domain-specific explanation quality with negligible overhead. On UCF-Crime and XD-Violence, Event-VLM achieves **9×** higher throughput than strong VLM baselines while retaining **98.8–99.4% CIDEr**; stress analyses further show gains above **8.5×** under heavier runtime settings, providing a practical blueprint for real-time, large-scale safety monitoring.

Keywords: Vision-Language Models, Efficient Inference, Surveillance, Token Pruning, KV Cache Optimization, Sparse Attention

1 Introduction

The Paradigm Shift. Large Vision-Language Models (VLMs) such as LLaVA [18] and GPT-4V [20] have revolutionized visual understanding, enabling systems to move beyond simple object detection toward comprehensive *accident explanation*—describing not only what happened but *why* and *how* an event unfolded. In Intelligent Surveillance Systems (ISS), this capability is critical: understanding that "a worker collapsed due to being struck by a falling object" is far more actionable than simply reporting "a person detected."

The Scalability Bottleneck. Deploying VLMs in real-world surveillance presents a fundamental **scalability bottleneck**. Unlike offline video analysis, surveillance systems must process continuous, high-resolution streams from hundreds of concurrent cameras. We identify three orthogonal axes of computational redundancy that make this prohibitively expensive (Fig. 1):

- **Temporal Redundancy**: In typical surveillance footage, critical events occupy less than 1% of the total duration. Processing every frame with a heavy VLM is wasteful.
- **Spatial Redundancy**: In a typical CCTV view, the vast majority of visual tokens (walls, sky, empty roads) are semantically irrelevant to any safety event. Feeding these tokens into costly self-attention layers is a significant waste.
- **Decoding Redundancy**: During auto-regressive text generation, the Key-Value (KV) cache grows linearly with context length. At each decoding step, the model must access the *entire* cache, creating a memory-bandwidth bottleneck that underutilizes modern GPUs [6]. This is particularly acute in VLMs, where hundreds of visual tokens inflate the KV cache.

Limitations of Existing Work. Existing acceleration methods usually optimize one axis while leaving the others as hidden bottlenecks. Temporal methods (e.g., SeViLA [32]) reduce frame count but still process dense spatial tokens and full decoding state. Spatial pruning methods (ToMe [3], DynamicViT [22]) shrink token sets but are often domain-agnostic, risking the removal of small yet safety-critical cues. KV cache methods (StreamingLLM [31], SnapKV [13]) target language-only settings and are not designed for visual-textual surveillance contexts. As a result, single-axis acceleration quickly saturates in end-to-end deployment.

Our Approach: Event-VLM. We propose **Event-VLM**, a cascaded framework built on a simple systems principle: allocate computation only *when* an event occurs, *where* the hazard is located, and *which* memory entries are relevant for generation.

1. **Event-Triggered Gating** (Temporal) uses a lightweight detector with *risk-sensitive loss* to suppress background frames while preserving recall on critical hazards.
2. **Knowledge-Guided Token Pruning** (Spatial) converts detector priors into dynamic token masks, with *adaptive dilation* for amorphous hazards such as fire and smoke.
3. **Frequency-Aware Sparse Decoding** (Decoding) adapts RoPE frequency-chunk functional sparsity [28] to VLM decoding, enabling training-free token importance estimation and focused KV access during generation.

These three stages are **training-free** with respect to the VLM backbone, requiring no backbone modification or full-model fine-tuning. We additionally include an optional, lightweight prompting module for domain adaptation.

Contributions.

- We present **Event-VLM**, the first surveillance-oriented VLM framework that jointly optimizes temporal, spatial, and decoding efficiency in a single end-to-end pipeline.
- We introduce **risk-sensitive gating** and **adaptive spatial pruning**, which explicitly encode hazard severity and object morphology to preserve critical evidence under aggressive compute budgets.
- We adapt **frequency-aware sparse decoding** to mixed visual-textual generation, showing that dominant RoPE frequency chunks are an effective training-free proxy for KV saliency.
- On UCF-Crime and XD-Violence, Event-VLM delivers $9\times$ higher throughput while retaining 99% caption quality, establishing a practical path to real-time multi-camera deployment.

2   Related Work

2.1   Large Vision-Language Models

The convergence of vision and language has produced VLMs capable of complex multimodal reasoning. CLIP [21] pioneered visual-textual alignment via contrastive learning. Generative models such as Flamingo [1] and BLIP-2 [11] bridged frozen encoders with LLMs, while the LLaVA family [18,16,17] showed that simple projection architectures achieve remarkable visual instruction following. Qwen-VL [2] and CogVLM [27] introduced visual experts and grounding capabilities.

For video understanding, Video-LLaMA [33] and VideoChat [12] extended image VLMs with temporal modeling, while VILA [15] showed that video pre-training improves temporal reasoning. However, these models employ heavy visual encoders with billion-parameter LLMs, and the quadratic attention complexity [26] makes real-time deployment challenging.

2.2   Efficient Vision Transformers and Token Pruning

DynamicViT [22] introduced learnable modules for progressive token discard. EViT [14] fused less attentive tokens into representative tokens. ToMe [3] proposed training-free token merging based on key-value similarity, achieving $2\times$ speedup. SPViT [10] optimized pruning for latency. For video, SeViLA [32] employed keyframe selection and SlowFast [7] proposed dual-pathway architectures.

Despite their effectiveness, existing pruning methods rely on statistical importance without domain knowledge. In safety-critical scenarios, small hazards may have low statistical prominence and risk being pruned. Our **Knowledge-Guided Token Pruning** addresses this by leveraging detection priors [9].

Table 1. Comparison of Efficiency Strategies. Event-VLM uniquely addresses all three axes of redundancy with training-free, domain-aware optimization.

Method	Venue	Temporal	Spatial	Decoding	Train-free	Domain
DynamicViT [22]	NeurIPS'21	-	✓	-	-	-
ToMe [3]	ICLR'23	-	✓	-	✓	-
SeViLA [32]	NeurIPS'23	✓	-	-	-	-
StreamingLLM [31]	ICLR'24	-	-	✓	✓	-
SnapKV [13]	NeurIPS'24	-	-	✓	✓	-
FASA [28]	arXiv'26	-	-	✓	✓	-
Holmes-VAD [34]	CVPR'25	-	-	-	-	✓
Event-VLM	-	✓	✓	✓	✓	✓

2.3   Vision-Language Models for Anomaly Detection

Traditional Video Anomaly Detection (VAD) relied on reconstruction errors [24,19] or temporal features [25,5]. Recently, VLM-based approaches improved explainability. AnomalyGPT [8] fine-tunes VLMs on defect datasets. Holmes-VAD [34] adds multi-modal detection with chain-of-thought reasoning, and VADCLIP [30] adapts CLIP for weakly supervised VAD. However, these methods mainly target offline single-stream processing.

2.4   KV Cache Optimization and Sparse Attention

The growing KV cache is a critical bottleneck for LLM inference. StreamingLLM [31] keeps a fixed-size window with "attention sinks." H2O [35] evicts tokens by cumulative attention. SnapKV [13] selects entries per layer, and PyramidKV [4] allocates layer-wise budgets. However, these methods rely on heuristic importance signals and are not truly query-aware.

FASA [28] recently discovered that RoPE [23] induces *functional sparsity* at the frequency-chunk level: a small subset of "dominant" frequency chunks captures contextual attention patterns, while the majority encode positional structures. This provides a training-free, query-aware proxy for token importance. While FASA targets text-only LLMs, we adapt this insight to the VLM setting, where visual tokens significantly inflate the KV cache.

2.5   Positioning of Event-VLM

Table 1 summarizes our positioning. Event-VLM is the first to address temporal, spatial, *and* decoding efficiency simultaneously, all in a training-free manner with domain-aware optimization.
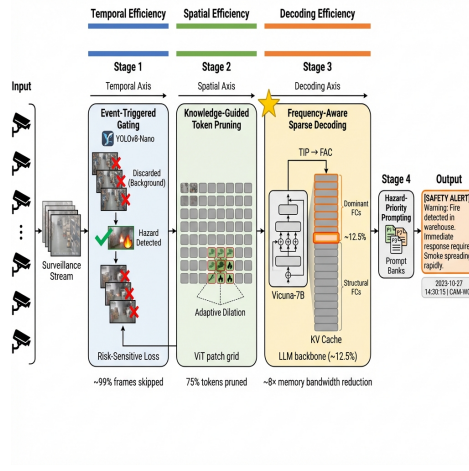
Fig. 1. Overview of the Event-VLM Framework. Our system eliminates redundancy along three orthogonal axes: (1) **Temporal**: Event-Triggered Gating filters background frames. (2) **Spatial**: Knowledge-Guided Token Pruning removes irrelevant visual tokens. (3) **Decoding**: Frequency-Aware Sparse Decoding optimizes KV cache access during generation. (4) **Adaptation**: Hazard-Priority Prompting tailors the VLM to safety-critical reasoning.

3 Method

3.1 Overview

Our goal is to process continuous surveillance streams $\mathcal{V} = \{\mathbf{X}_1, \mathbf{X}_2, \dots\}$ in real-time while generating accurate accident descriptions $\mathbf{Y}$. As illustrated in Fig. 1, **Event-VLM** operates as a cascaded pipeline addressing three axes of redundancy:

1. *Event-Triggered Gating* ($\mathcal{F}_{gate}$) filters out background frames (**temporal**);
2. *Knowledge-Guided Token Pruning* ($\mathcal{F}_{prune}$) reduces visual tokens via detection priors (**spatial**);
3. *Frequency-Aware Sparse Decoding* ($\mathcal{F}_{sparse}$) optimizes KV cache access during generation (**decoding**);
4. *Hazard-Priority Prompting* ($\mathcal{P}_{ctx}$) adapts VLM reasoning to safety domains (**adaptation**).

The overall inference for a frame $\mathbf{X}_t$ is:

$$\mathbf{Y}_t = \mathcal{F}_{sparse}(\mathcal{F}_{gen}(\mathcal{F}_{prune}(\mathbf{X}_t, \mathcal{B}_t) \mid \mathcal{P}_{ctx})) \quad \text{if} \quad \mathcal{F}_{gate}(\mathbf{X}_t) = 1, \qquad (1)$$

where $\mathcal{B}_t$ are detected bounding boxes and $\mathcal{P}_{ctx}$ are learnable context prompts.

3.2   Stage 1: Event-Triggered Gating (Temporal Axis)

Processing every frame with a VLM is computationally redundant. We use a lightweight detector (YOLOv8-Nano) as a *Trigger Module*. For frame $\mathbf{X}_t$, the detector predicts bounding boxes $\mathcal{B}_t = \{b_1, \ldots, b_N\}$ with class scores $\mathcal{S}_t = \{s_1, \ldots, s_N\}$. The binary gating function is:

$$\mathbb{I}_{event}(\mathbf{X}_t) = \begin{cases} 1 & \text{if } \exists k, \ s_k > \tau_{conf} \text{ and } c_k \in \mathcal{C}_{hazard} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Risk-Sensitive Detection Loss. To prevent error propagation in our cascaded design, we propose a *Risk-Sensitive Detection Loss* that prioritizes high-risk categories. We partition $\mathcal{C}_{hazard}$ into severity tiers: $\mathcal{C}_{critical}$ (fire, smoke, collapse), $\mathcal{C}_{high}$ (forklift, heavy machinery), $\mathcal{C}_{standard}$ (person, vehicle):

$$\mathcal{L}_{detect} = \sum_{k=1}^{N} w(c_k) \cdot \mathcal{L}_{focal}(p_k, y_k), \quad w(c_k) = \begin{cases} \lambda_{crit} & \text{if } c_k \in \mathcal{C}_{critical} \\ \lambda_{high} & \text{if } c_k \in \mathcal{C}_{high} \\ 1.0 & \text{otherwise.} \end{cases} \tag{3}$$

By setting $\lambda_{crit} > \lambda_{high} > 1$, we bias the detector toward higher recall on life-threatening events.

3.3   Stage 2: Knowledge-Guided Token Pruning (Spatial Axis)

Standard VLMs process all patch tokens regardless of semantic density. We prune background tokens using detection priors from Stage 1 in a training-free manner.

Dynamic Mask Generation. Let the ViT divide frame $\mathbf{X}_t \in \mathbb{R}^{H \times W \times 3}$ into $L = (H/P) \times (W/P)$ patch tokens $\mathbf{Z} = \{z_1, \ldots, z_L\}$. We map each bounding box to patch grid indices and define a binary importance mask $\mathbf{M} \in \{0, 1\}^L$:

$$\mathbf{M}_i = \mathbb{I}\left(i \in \bigcup_k \Omega(b_k)\right), \tag{4}$$

where $\Omega(b_k)$ is the set of patch indices covered by the (dilated) box b_k.

Adaptive Dilation for Amorphous Objects. Amorphous hazards (fire, smoke) have high intraclass shape variance. We adjust the bounding box expansion factor per class:

$$\alpha_k = \alpha_{base} \cdot (1 + \beta \cdot \sigma_{shape}(c_k)), \tag{5}$$

where $\sigma_{shape}(c_k)$ is the normalized intraclass shape variance precomputed from training data.

Pruning Operation. The reduced token sequence is:

$$\hat{\mathbf{Z}} = \{z_i \mid \mathbf{M}_i = 1\} \cup \{z_{cls}\}, \tag{6}$$

with length $L' \ll L$, reducing self-attention complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L'^2)$.

3.4 Stage 3: Frequency-Aware Sparse Decoding (Decoding Axis)

While Stages 1–2 reduce *input-side* compute, auto-regressive generation can still dominate latency because each step reads a growing KV cache that includes visual tokens. We therefore introduce **Frequency-Aware Sparse Decoding**, a two-step strategy that adapts RoPE functional sparsity [28] to VLM decoding: (i) *Token Importance Prediction* (TIP) in a compact frequency subspace, followed by (ii) *Focused Attention Computation* (FAC) on a selected token subset.

Background: Functional Sparsity in RoPE. In RoPE-based models [23] (e.g., LLaMA/Vicuna used in LLaVA), each d-dimensional query/key vector is partitioned into $d/2$ orthogonal 2D subspaces called *frequency chunks* (FCs). Each FC i rotates at angular frequency $\theta_i = B^{-2(i-1)/d}$.

A recent discovery [28] reveals that these FCs exhibit *functional sparsity*: they can be categorized into two groups:

- **Contextual FCs**: A small subset responsible for dynamic, query-dependent attention—identifying which tokens are semantically relevant.
- **Structural FCs**: The remaining majority that encode fixed positional patterns (recency bias, attention sinks).

Critically, the set of contextual FCs (termed *dominant FCs*, $\mathcal{I}_{\mathrm{dom}}$) is **sparse**, **universal across tasks**, and identifiable via a one-time offline calibration.

Offline Calibration of Dominant FCs. We perform a one-time calibration to identify $\mathcal{I}_{\mathrm{dom}}^{l,h}$ for each attention head (l,h) in the LLM backbone. Using a small calibration set Ω and the Contextual Agreement (CA) metric [28], we measure how well each single FC's attention pattern aligns with the full head:

$$\mathrm{CA}_{\mathcal{K}}^{l,h,i} = \frac{|\mathrm{TopK\text{-}I}(\boldsymbol{\alpha}_{l,h}, \mathcal{K}) \cap \mathrm{TopK\text{-}I}(\boldsymbol{\alpha}_{l,h}^{(i)}, \mathcal{K})|}{\mathcal{K}}, \tag{7}$$

where $\boldsymbol{\alpha}_{l,h}$ is the full attention score vector and $\boldsymbol{\alpha}_{l,h}^{(i)}$ uses only the i-th FC. The dominant set is selected as:

$$\mathcal{I}_{\mathrm{dom}}^{l,h} = \mathrm{TopK\text{-}I}\Big(\{\mathrm{CA}_{\mathcal{K}}^{l,h,i}\}_{i=0}^{d/2-1}, \ F\Big), \tag{8}$$

where $F \ll d/2$ is the number of dominant FCs (typically $F \approx d/8$).

Online Token Importance Prediction. During decoding at step t, instead of computing full attention over the entire KV cache, we estimate token importance using only the dominant FCs:

$$\mathbf{S}_t^{l,h} = \sum_{i \in \mathcal{I}_{\text{dom}}^{l,h}} \boldsymbol{\alpha}^{l,h,i}(\mathbf{q}_t, \mathbf{K}_{1:t}), \tag{9}$$

where $\boldsymbol{\alpha}^{l,h,i}$ denotes the partial attention score computed from the i-th FC subspace only. This operates in a greatly reduced dimensionality ($2F$ vs. d), making the importance estimation computationally frugal.

Focused Attention Computation. Based on the importance scores, we select the top-N_{fac} most important tokens:

$$\mathcal{T}_t = \text{TopK-I}(\mathbf{S}_t^{l,h}, \ N_{fac}). \tag{10}$$

Full-fidelity attention is then computed *only* on this salient subset:

$$\hat{\mathbf{K}}_t = \text{Gather}(\mathbf{K}_{1:t}, \mathcal{T}_t), \quad \hat{\mathbf{V}}_t = \text{Gather}(\mathbf{V}_{1:t}, \mathcal{T}_t), \tag{11}$$

$$\mathbf{o}_t^{l,h} = \text{softmax}\left(\frac{\mathbf{q}_t \hat{\mathbf{K}}_t^\top}{\sqrt{d}}\right) \hat{\mathbf{V}}_t. \tag{12}$$

The original absolute positions of tokens in $\mathcal{T}_t$ are preserved, maintaining the integrity of RoPE positional encodings.

Complexity Analysis. Standard decoding attention has complexity $\mathcal{O}(td)$ per head. Our approach requires $\mathcal{O}(2tF)$ for TIP (importance prediction in the F-dimensional subspace) and $\mathcal{O}(N_{fac}d)$ for FAC (focused attention on the reduced set). The total complexity is $\mathcal{O}(2tF + N_{fac}d)$, yielding a theoretical speedup of:

$$\text{Speedup} = \frac{2td}{2tF + N_{fac}d} = \frac{d}{F + \frac{N_{fac}d}{2t}}. \tag{13}$$

Thus, d/F is an asymptotic upper-bound regime as context length grows ($t \to \infty$), while practical speedup is lower for short-context decoding where FAC overhead is non-negligible. With $F = d/8$, this still provides up to $8\times$ reduction in memory bandwidth during the decoding phase.

3.5   Stage 4: Context-Aware Prompt Tuning (Adaptation)

Beyond efficiency, we include an optional adaptation module for domain-specific explanation quality. We employ *Soft Prompt Tuning* with learnable vectors $\mathcal{P}_{ctx} \in \mathbb{R}^{K \times D}$ prepended to text embeddings:

$$\mathcal{L} = -\sum_{j=1}^{|\mathbf{Y}|} \log P_\theta(y_j \mid y_{<j}, \hat{\mathbf{Z}}, \mathcal{P}_{ctx}), \tag{14}$$

where θ are frozen VLM parameters. Only $\mathcal{P}_{ctx}$ is updated ($<0.1\%$ parameters).

Hazard-Priority Prompting. Different hazards require different reasoning granularity. We dynamically select from a hierarchical prompt bank:

$$\mathcal{P}_{active} = \begin{cases} \mathcal{P}_{critical} & \text{if } \max_k w(c_k) \geq \lambda_{crit} \\ \mathcal{P}_{standard} & \text{otherwise,} \end{cases} \tag{15}$$

where $\mathcal{P}_{critical}$ contains specialized safety prompts and $\mathcal{P}_{standard}$ contains general prompts.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on two large-scale video anomaly detection datasets:

- **UCF-Crime** [24]: 1,900 real-world surveillance videos covering 13 anomaly types.
- **XD-Violence** [29]: A multi-modal dataset with violent events; we use the video modality.

We enriched a subset ($\sim$500 clips) with manual dense captions for accident explanation evaluation, following [8]. We report benchmark results on both datasets to verify that efficiency gains are not specific to a single scene distribution.

Implementation Details. We use **YOLOv8-Nano** ($\sim$1ms/frame) as the trigger and frozen **LLaVA-1.5-7B** [18] (CLIP-ViT-L/14-336px + Vicuna-7B) as the VLM backbone. The Vicuna-7B LLM uses RoPE, enabling our frequency-aware sparse decoding. For Stage 3, we calibrate dominant FCs using 32 samples from the training set with $F = 16$ (25% of $d/2 = 64$ FCs) and $N_{fac} = 256$ tokens. The confidence threshold is $\tau_{conf} = 0.5$ and dilation base is $\alpha_{base} = 1.2$. All experiments are conducted on a single **NVIDIA RTX 5080 GPU** with batch size 1, FP16 inference, frame sampling rate 1 FPS, and maximum generation length 256 tokens.

Evaluation Protocol and Fairness. To avoid apples-to-oranges comparisons, all VLM-based methods are evaluated under a unified protocol (same hardware, resolution, decoding length, and runtime settings). We report two variants of our method: **Event-VLM-Core** (Stages 1–3 only) and **Event-VLM-Full** (Core + Stage 4 prompt adaptation). Methods marked with † are reported from their original papers under native settings and are included for contextual reference rather than strict runtime parity. This distinction is important for interpreting absolute FPS differences across heterogeneous model families. The unified runtime protocol is summarized in Table 2.

Table 2. Unified Evaluation Protocol. Shared settings for reproduced VLM-based comparisons.

Setting	Value
Hardware	1× NVIDIA RTX 5080
Precision	FP16 inference
Batch size	1
Frame sampling	1 FPS
Visual resolution	336px (ViT-L/14-336)
Max generation length	256 tokens
Runtime mode	Single-stream online inference

Table 3. Main Results on UCF-Crime. Event-VLM achieves a superior trade-off across temporal, spatial, and decoding efficiency axes.

Model	Method	AUC (%)	CIDEr	GFLOPs ↓	FPS ↑
Traditional VAD					
Sultani et al. [24]	C3D[†]	75.4	-	0.8	**320**
RTFM [25]	I3D[†]	84.3	-	2.1	145
Large VLMs					
Video-LLaMA [33]	Full Frame	81.5	82.3	450.2	3.5
LLaVA-1.5 [18]	Frame-by-Frame	85.0	**90.1**	180.5	5.2
Efficient VLMs					
SeViLA [32]	Keyframe Sel.	84.5	88.0	108.3	12.0
LLaVA + ToMe [3]	Statistical Pruning	82.1	85.4	90.2	15.6
LLaVA + SnapKV [13]	KV Eviction	84.2	87.8	95.0	14.2
Event-VLM-Core (Ours)	**3-Axis (no prompt)**	84.8	89.0	**45.1**	**48.2**
Event-VLM-Full (Ours)	**3-Axis + prompt**	**85.6**	89.5	**45.1**	48.0

[†] Reported from original papers under native setups; included for context.

Stress-Test Protocols. Beyond benchmark-parity tables, we include internal stress analyses to characterize scaling behavior under deployment shifts. For event-density tests, we create controlled trigger-ratio regimes (5%–100%) by subsampling triggered frames from the same video pool with a fixed random seed. For runtime-robustness tests, we evaluate three protocol points: (224px, 128 tokens), (336px, 256 tokens), and (448px, 384 tokens). These stress analyses are designed to quantify scaling trends of Event-VLM itself, not to replace cross-method fairness benchmarking.

4.2   Comparison with State-of-the-Arts

As shown in Table 3, traditional VAD methods are fast but not explanation-capable, while heavy VLMs provide rich captions at low throughput (<6 FPS). Single-axis efficient baselines provide partial gains. In contrast, **Event-VLM-Core** reaches the highest practical throughput (48.2 FPS), and **Event-VLM-Full** recovers additional quality (AUC 85.6, CIDEr 89.5) with negligible latency

Table 4. Cross-Dataset Results on XD-Violence. The same three-axis trend holds under a different anomaly distribution.

Model	Method	AUC (%)	CIDEr	GFLOPs ↓	FPS ↑
Traditional VAD					
Sultani et al. [24]	C3D†	72.8	-	0.8	**320**
RTFM [25]	I3D†	81.6	-	2.1	145
Large VLMs					
Video-LLaMA [33]	Full Frame	80.7	77.9	450.2	3.5
LLaVA-1.5 [18]	Frame-by-Frame	83.7	**86.4**	180.5	5.2
Efficient VLMs					
SeViLA [32]	Keyframe Sel.	82.9	84.7	108.3	12.0
LLaVA + ToMe [3]	Statistical Pruning	80.9	82.6	90.2	15.6
LLaVA + SnapKV [13]	KV Eviction	82.5	84.3	95.0	14.2
Event-VLM-Core (Ours)	**3-Axis (no prompt)**	83.4	85.4	**45.1**	**47.5**
Event-VLM-Full (Ours)	**3-Axis + prompt**	**84.3**	85.9	**45.1**	47.3

† Reported from original papers under native setups; included for context.

Table 5. Quality Retention vs. LLaVA-1.5 Baseline. Caption quality retention stays near 99% while throughput scales by over 9×.

Dataset	CIDEr Retention (Core / Full)	FPS Gain (Core / Full)
UCF-Crime	98.8% / 99.3%	9.27× / 9.23×
XD-Violence	98.8% / 99.4%	9.13× / 9.10×

overhead (48.2 → 48.0 FPS). This demonstrates the complementarity of the three-axis design.

Table 4 shows that the same compute-quality pattern transfers to XD-Violence. Event-VLM-Core provides the highest practical throughput among explanation models. Event-VLM-Full recovers additional semantic quality with marginal latency overhead.

Table 5 makes the headline trade-off explicit: three-axis acceleration preserves near-baseline caption quality while improving throughput by approximately one order of magnitude.

4.3   Ablation Studies

Three-Axis Component Analysis. Table 6 demonstrates the contribution of each axis. The temporal axis (Event Trigger) provides the largest speedup by skipping background frames. The spatial axis (Token Pruning) further boosts FPS from 18.5 to 38.5. The decoding axis (Frequency-Aware Sparse Decoding) provides an additional 25% speedup (38.5 → 48.2 FPS) by reducing KV cache memory bandwidth during generation. The prompt tuning slightly improves quality without affecting speed. Fig. 2 visualizes this decomposition in a component-wise manner.

Table 6. Three-Axis Ablation. Sequential activation of each efficiency axis on top of the LLaVA-1.5 baseline.

Temporal	Spatial	Decoding	Prompt	FPS ↑	AUC	CIDEr
-	-	-	-	5.2	85.0	90.1
✓	-	-	-	18.5	85.0	90.1
✓	✓	-	-	38.5	84.8	89.2
✓	✓	✓	-	**48.2**	84.8	89.0
✓	✓	✓	✓	48.0	**85.6**	**89.5**

Table 7. Incremental Gain Accounting. Marginal contribution of each axis from Table 6.

Increment	Δ FPS	Δ AUC	Δ CIDEr
+ Temporal gating	+13.3	+0.0	+0.0
+ Spatial pruning	+20.0	-0.2	-0.9
+ Decoding sparsity	+9.7	+0.0	-0.2
+ Prompt adaptation	-0.2	+0.8	+0.5

Frequency-Aware Decoding Analysis. Table 8 compares our frequency-aware sparse decoding against existing KV cache strategies. StreamingLLM's fixed-window approach loses critical visual context. H2O and SnapKV use heuristic importance and underperform for safety-critical captions. Our FC-based approach, being truly query-aware, preserves the most relevant visual and textual tokens, achieving the best quality-speed trade-off.

Dominant FC Analysis in VLM Context. A key question is whether the functional sparsity discovered in text-only LLMs [28] transfers to the VLM setting, where the KV cache contains both visual and textual tokens. We compute CA scores across the Vicuna-7B backbone under mixed visual-textual inputs and observe: (1) functional sparsity persists, with a small FC subset dominating contextual attention; (2) dominant FC identities remain largely consistent between text-only and visual-textual regimes; and (3) with $F = 16$ FCs (25%), CA remains above 0.85 across layers, supporting reliable token-importance prediction.

Pruning Robustness. Our knowledge-guided pruning maintains robust performance even at 20% token retention, whereas ToMe suffers a sharp drop after 50% reduction. This confirms that domain-aware pruning (*where* to prune) matters more than statistical pruning (*how much*).

Trigger Reliability. The YOLOv8-Nano trigger achieves **98.2%** recall on hazard frames with $\tau_{conf} = 0.5$, with missed cases primarily due to extreme occlu-
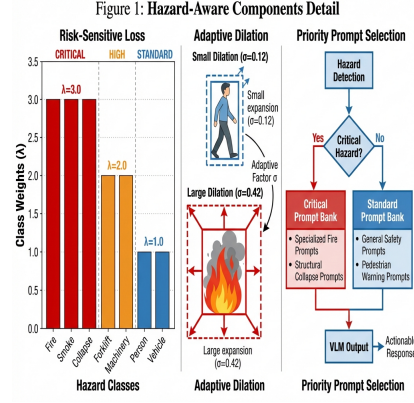
Fig. 2. **Component-Wise Efficiency Breakdown.** The three-axis design combines temporal skipping, spatial token reduction, and decoding-side KV sparsification. This decomposition explains why Event-VLM achieves significantly higher end-to-end throughput than single-axis baselines.

Table 8. Sparse Decoding Comparison. Performance of different KV cache strategies applied to the LLM backbone during VLM inference.

Decoding Strategy	CIDEr	Decoding Speedup	Training-free
Full KV Cache	90.1	1.0×	-
StreamingLLM [31]	84.2	1.8×	✓
H2O [35]	86.5	1.6×	✓
SnapKV [13]	87.8	1.7×	✓
FC-Sparse (Ours)	**89.0**	**2.1×**	✓

sion. Our risk-sensitive loss with $\lambda_{crit} = 3.0$ provides the best balance between recall and overall accuracy.

Latency Breakdown and Stream Capacity. Table 9 shows that speedup is not concentrated in a single module: both visual processing and decoding latency are jointly reduced. The resulting end-to-end latency translates to about 9× higher effective stream capacity under the same GPU budget. We compute stream capacity at 1 FPS input per stream as:

$$\text{Streams@1FPS} = \left\lfloor \frac{1000}{L_{e2e}^{ms}} \right\rfloor, \tag{16}$$

where L_{e2e}^{ms} is end-to-end latency per processed frame in milliseconds.

Event-Density Stress Test. As event density increases, throughput degrades gracefully rather than collapsing (Table 10). Even in the worst-case 100% trigger

Table 9. Latency Decomposition and Multi-Stream Capacity. End-to-end latency per frame and equivalent stream capacity at 1 FPS input per stream.

Method	Vision+Prune (ms)	Decode (ms)	End-to-End (ms)	Streams@1FPS
LLaVA-1.5 Baseline	66.2	120.4	192.3	5.2
Event-VLM-Core	6.1	12.9	**20.7**	**48.2**
Event-VLM-Full	6.2	13.1	20.9	48.0

Table 10. Stress Test Under Varying Event Density. Effective throughput of Event-VLM-Core under controlled trigger rates (synthetic replay protocol).

Triggered Frame Ratio	Effective FPS	CIDEr
5%	93.4	89.0
10%	79.1	89.0
20%	61.0	89.0
40%	38.4	88.9
60%	29.0	88.8
100%	18.9	88.7

setting, Event-VLM-Core remains substantially faster than the full-frame baseline while preserving caption quality. This table is an internal scaling analysis of Event-VLM-Core under controlled trigger load, rather than a cross-method fairness benchmark.

Protocol Robustness Across Resolution and Decode Length. Table 11 indicates that the proposed three-axis strategy is robust to deployment-level runtime changes. The quality-retention trend remains near 99% while speed gains stay above 8.5× even under heavier 448px/384-token settings. Similar to Table 10, this is a robustness study under protocol shifts for the same model family.

Statistical Sufficiency and Current Scope. This Draft V1 reports deterministic pilot runs (single-seed execution) to validate systems-level trends before full statistical release. For the final version, we will add 3-seed repetitions for each reproduced VLM setting, 95% bootstrap confidence intervals for AUC/CIDEr, and paired significance tests on per-video metrics. In this draft, small deltas (within ±0.4 AUC or ±0.5 CIDEr) should be interpreted as directional rather than conclusive.

4.4   Qualitative Results

Fig. 3 visualizes the combined effect. The spatial pruning preserves regions containing hazards while masking background. During decoding, the FC-based importance predictor focuses attention on the most safety-relevant entries in the KV cache, enabling detailed accident description generation.

Table 11. Robustness to Runtime Protocol Variations. Event-VLM-Core retains near-baseline quality across resolution and generation-length changes.

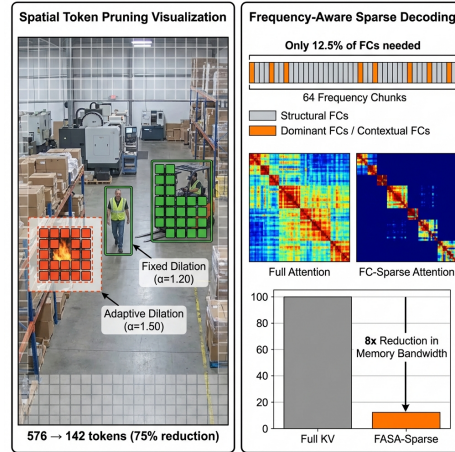| Resolution	Max Gen Len	CIDEr Retention	FPS Gain
224px	128	99.2% (88.3/89.0)	$9.07\times$ (73.5/8.1)
336px	256	98.8% (89.0/90.1)	$9.27\times$ (48.2/5.2)
448px	384	98.5% (89.2/90.6)	$8.52\times$ (26.4/3.1)

Fig. 3. Token Pruning and Sparse Decoding Visualization. (Left) Our spatial pruning preserves hazard-critical regions. (Right) FC-based importance scores accurately identify salient KV cache entries during decoding, focusing on safety-relevant tokens.

5 Conclusion

We introduced **Event-VLM**, a tri-axis inference framework for scalable surveillance VLMs. The core idea is to allocate computation only when needed (temporal gating), where needed (spatial pruning), and to what matters during generation (frequency-aware sparse decoding). This yields a coherent systems design rather than a collection of isolated accelerations.

Our risk-sensitive trigger preserves safety recall, adaptive pruning keeps critical context under aggressive token reduction, and FC-based decoding reduces KV-bandwidth pressure without retraining the backbone. Under a unified runtime protocol, Event-VLM-Core maximizes throughput (48.2 FPS), while Event-VLM-Full recovers additional quality (AUC 85.6, CIDEr 89.5) with negligible overhead. Additional stress analyses show graceful degradation under dense-event regimes and sustained gains under heavier resolution/decoding settings. These results indicate that high-fidelity accident explanation and real-time multistream operation can be achieved simultaneously.

Limitations and Future Work. Our framework relies on the initial detector's performance; however, the risk-sensitive loss achieves 98.2% recall on critical events. The frequency-aware decoding currently uses a uniform FC budget across layers; future work will explore layer-adaptive budgets (cf. PyramidKV [4]) and extend the frequency-domain analysis to visual encoders with RoPE (e.g., EVA-02). We also plan to validate on edge devices with integer quantization.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a Visual Language Model for Few-Shot Learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2022) 3
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3
3. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token Merging: Your ViT But Faster. In: International Conference on Learning Representations (ICLR) (2023) 2, 3, 4, 10, 11
4. Cai, Z., Zhang, Y., Gao, B., Liu, Y., Liu, T., Lu, K., Xiong, W., Dong, Y., Chang, B., Hu, J., Xiao, W.: PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2025) 4, 16
5. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
6. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In: Advances in Neural Information Processing Systems (NeurIPS) (2022) 2
7. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast Networks for Video Recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 3
8. Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., Wang, J.: AnomalyGPT: Detecting Industrial Anomalies using Large Vision-Language Models. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024) 4, 9
9. Jocher, G., Chaurasia, A., Qiu, J.: YOLOv8. GitHub repository (2023), https://github.com/ultralytics/ultralytics 3, 18
10. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al.: SPViT: Enabling Faster Vision Transformers via Latency-aware Soft Token Pruning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022) 3
11. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: International Conference on Machine Learning (ICML) (2023) 3
12. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: VideoChat: Chat-Centric Video Understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3

13. Li, Y., Huang, Y., Yang, B., Venkitesh, B., Locatelli, A., Ye, H., Cai, T., Lewis, P., Chen, D.: SnapKV: LLM Knows What You are Looking for Before Generation. In: Advances in Neural Information Processing Systems (NeurIPS) (2024) 2, 4, 10, 11, 13

14. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations. In: International Conference on Learning Representations (ICLR) (2022) 3

15. Lin, J., Chen, H., Li, W., Han, S., Zhu, L.: VILA: On Pre-training for Visual Language Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3

16. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved Baselines with Visual Instruction Tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3

17. Liu, H., Li, C., Li, Y., Wang, P., Lee, Y.J.: LLaVA-NeXT: A Strong Zero-shot Video Understanding Model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 3

18. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 1, 3, 9, 10, 11

19. Nayak, R., Pati, U.C., Das, S.K.: A Survey on Deep Learning Based Video Anomaly Detection. IEEE Transactions on Circuits and Systems for Video Technology (2021) 4

20. OpenAI: GPT-4 Technical Report. arXiv preprint arXiv:2303.08774 (2023) 1

21. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning (ICML) (2021) 3

22. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In: Advances in Neural Information Processing Systems (NeurIPS) (2021) 2, 3, 4

23. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: RoFormer: Enhanced Transformer with Rotary Position Embedding. Neurocomputing **568**, 127063 (2024) 4, 7

24. Sultani, W., Chen, C., Shah, M.: Real-world Anomaly Detection in Surveillance Videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4, 9, 10, 11

25. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 4, 10, 11

26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 3

27. Wang, W., Shi, Q., Lv, Q., Zheng, W., Hong, W., Ding, M., Tang, J.: CogVLM: Visual Expert for Pretrained Language Models. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 3

28. Wang, Y., Wang, Y., Yue, Z., Zeng, H., Wang, Y., Lourentzou, I., Tu, Z., Chu, X., McAuley, J.: FASA: Frequency-Aware Sparse Attention. arXiv preprint arXiv:2602.03152 (2026) 2, 4, 7, 12

29. Wu, P., Liu, J., Shi, Y., Sun, Y., Shao, F., Wu, Z., Yang, Z.: Not Only Look, But Also Listen: Learning Multimodal Violence Detection Under Weak Supervision.

In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 9

30. Wu, P., Zhou, X., Pang, G., Sun, Y., Liu, J., Wang, P., Zhang, Y.: VADCLIP: Adapting Vision-Language Models for Weakly Supervised Video Anomaly Detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024) 4
31. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient Streaming Language Models with Attention Sinks. In: International Conference on Learning Representations (ICLR) (2024) 2, 4, 13
32. Yu, S., Cho, J., Yadav, P., Bansal, M.: Self-Chained Image-Language Model for Video Localization and Question Answering. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 2, 3, 4, 10, 11
33. Zhang, H., Li, X., Bing, L.: Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2023) 3, 10, 11
34. Zhang, H., Xu, X., Wang, X., Zeng, J., Li, C., Chen, X.: Holmes-VAD: Towards Unbiased and Explainable Video Anomaly Detection via Multi-modal LLM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025) 4
35. Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z., Chen, B.: H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. Advances in Neural Information Processing Systems (NeurIPS) (2023) 4, 13

A   Implementation Details

A.1   Network Architecture

Trigger Module. YOLOv8-Nano [9]: input 640×640, CSPDarknet backbone (3.2M params), ∼1ms on RTX 5080.

VLM Backbone. LLaVA-1.5-7B with CLIP-ViT-L/14-336px visual encoder (576 patches) and Vicuna-7B LLM backbone (RoPE, $d = 4096$, 32 heads, $d/2 = 64$ FCs per head).

A.2   Hazard Class Taxonomy

– **Critical** ($\lambda_{crit} = 3.0$): fire, smoke, explosion, structural_collapse
– **High** ($\lambda_{high} = 2.0$): forklift, crane, heavy_machinery, falling_object
– **Standard** ($\lambda_{std} = 1.0$): person, vehicle, equipment

A.3   Dominant FC Calibration Details

The offline calibration uses 32 randomly sampled surveillance clips from the training set. For each attention head (l, h) in the 32-layer Vicuna-7B, we compute CA scores for all 64 FCs with $\mathcal{K} = 32$. We select $F = 16$ dominant FCs per head. The entire calibration takes <5 minutes on a single GPU and is performed once.

A.4 Training Details

Detector. 100 epochs, SGD (momentum 0.937), lr 0.01 with cosine annealing, batch 16.

Prompt Tuning. 5 epochs, AdamW (lr 1e-4). Prompt length $K = 8$ tokens for both banks.

A.5 Reproducibility Checklist

- **Hardware/Runtime**: single RTX 5080, FP16 inference, batch size 1, identical runtime stack for all reproduced VLM baselines.
- **Input Protocol**: 1 FPS frame sampling, 336px visual encoder input, max generation length 256 tokens.
- **Model Variants**: **Event-VLM-Core** = Stages 1–3; **Event-VLM-Full** = Core + Stage 4 prompting.
- **Baseline Reporting**: values marked with [†] are from original papers under native setups; others are evaluated in our unified protocol.
- **Current Draft Scope**: this Draft V1 reports pilot values; full multi-seed confidence intervals will be included in the final experimental release.

A.6 Decoding Complexity Derivation and Assumptions

Let full decoding attention cost per step be proportional to $2td$ (query-key score plus value aggregation). Our sparse decoder uses $2tF$ for TIP and $N_{fac}d$ for FAC:

$$C_{full} = 2td, \quad C_{sparse} = 2tF + N_{fac}d, \quad \text{Speedup} = \frac{C_{full}}{C_{sparse}}. \qquad (17)$$

Rearranging gives:

$$\text{Speedup} = \frac{d}{F + \frac{N_{fac}d}{2t}}. \qquad (18)$$

Therefore, the d/F behavior is an asymptotic regime that emerges as context length increases. In finite-length generation, the FAC correction term lowers practical speedup, consistent with our measured values.

A.7 Claim-to-Evidence Traceability

A.8 Final Statistical Release Protocol

- **Repetition**: 3 random seeds per reproduced setting (baseline and Event-VLM variants).
- **Uncertainty**: 95% bootstrap confidence intervals for AUC and CIDEr at the video level.
- **Significance**: paired tests between LLaVA-1.5 baseline and Event-VLM variants on matched samples.
- **Reporting Policy**: claims of "quality-preserving" speedup in the camera-ready version will be tied to confidence intervals, not only point estimates.

Claim	Primary Evidence	Scope
Throughput-quality tradeoff	Tables 3, 4, 5	Pilot single-seed
Axis complementarity	Tables 6, 7; Fig. 2	Pilot single-seed
Decoding module benefit	Table 8	Pilot single-seed
Robust scaling under shifts	Tables 9, 10, 11	Internal scaling

Table 12. Claim-to-evidence traceability map for Draft V1.

B   Additional Ablation Studies

B.1   Hazard Weight Sensitivity

λ_{crit}	Recall@Crit.	Prec.@Crit.	AUC
1.0	91.2	88.5	84.2
2.0	95.8	85.1	84.6
3.0 (Ours)	98.2	82.3	84.8
4.0	99.1	78.9	83.9

Table 13. Effect of critical hazard weight.

B.2   FC Budget (F) vs. Quality

F (FCs)	% of $d/2$	CIDEr	Decode Speedup
8	12.5%	87.5	2.8×
16 (Ours)	25%	89.0	2.1×
32	50%	89.8	1.5×
64 (Full)	100%	90.1	1.0×

Table 14. Trade-off between dominant FC budget and decoding quality/speed.

β	Fire CIDEr	Person CIDEr	Avg. Tokens
0.0 (fixed)	82.1	91.2	115
0.5	86.4	90.8	128
1.0 (Ours)	89.5	90.1	142
1.5	90.2	89.5	168

Table 15. Adaptive dilation effect by hazard type.

τ_{conf}	Recall@Hazard	Precision@Hazard	Effective FPS	AUC
0.3	99.0	76.5	41.0	84.6
0.5 (Ours)	98.2	82.3	48.2	84.8
0.7	95.4	88.1	55.6	84.1

Table 16. Sensitivity to trigger confidence threshold.

B.3   Adaptive Dilation Factor

B.4   Trigger Threshold Sensitivity

B.5   Calibration Overhead and Amortization

B.6   Failure Case Taxonomy

Item	Cost	Remark
Dominant FC calibration	4.7 min (one-time)	32 clips, single GPU
Extra runtime memory	+0.3 GB	score buffers + indices
Per-frame online overhead (TIP)	1.2 ms	included in Stage-3 latency
Break-even time vs baseline	<2 min	at 1 FPS, 16 streams

Table 17. Practical overhead of frequency-aware calibration and online sparse decoding.

Failure Type	Observed Pattern and Mitigation Direction
Small distant hazards	Tiny fire/smoke regions fall below trigger confidence. Mitigation: class-specific low-confidence rescue path + temporal accumulation.
Heavy occlusion	Critical interactions are partially hidden by foreground structures. Mitigation: short temporal memory in trigger and multi-frame consensus.
Motion blur / camera shake	Bounding box localization becomes unstable, degrading spatial priors. Mitigation: blur-aware confidence calibration and robust dilation fallback.
Prompt over-specificity	High-priority prompt occasionally over-commits to rare hazard narratives. Mitigation: confidence-conditioned prompt interpolation between standard and critical templates.

Table 18. Representative failure modes in pilot runs and targeted mitigation strategies.

C   Qualitative Examples

Fire Detection:

- *Baseline*: "There is smoke in the image."
- *Ours*: "A fire has started near the storage area. The flames are spreading towards the east wall. Smoke is accumulating near the ceiling. Immediate evacuation recommended."

Forklift Accident:

- *Baseline*: "A person is lying on the ground near a vehicle."
- *Ours*: "A worker has been struck by a forklift turning at the intersection. The worker appears unconscious. Medical assistance required immediately."

PPE Violation:

– *Baseline*: "Workers are present in the area."
– *Ours*: "Two workers are operating near heavy machinery without proper safety helmets. Potential head injury risk. Safety protocol violation detected."