ReactVAU: A Slow-Fast Decoupled Framework for Streaming Video Anomaly Understanding Supplementary Material

1 Grid Image Dataset Construction

To fine-tune the Fast Detection Module for Spatial Grid Folding (SGF), we construct a dedicated Grid Image Dataset. We sample frames from UCF-Crime and XD-Violence and stitch 4 consecutive frames into a 2×2 Grid Image format. All samples are drawn exclusively from the official training splits; validation and test videos are never accessed. The total grid size is 384×384 pixels. To clearly demarcate the frames, a 2-pixel wide gray boundary line is inserted between them, resulting in individual frames of 191×191 pixels.

Frame-level labels are derived from the HIVAU-70K event-interval annotations for the source videos. A grid is positive if at least one of its four frames overlaps an annotated anomaly interval; mixed grids containing both normal and anomalous frames remain positive. A grid is negative only when all four frames lie outside every anomaly interval. The original video-level labels then distinguish hard negatives from anomaly-containing videos and easy negatives from fully normal videos.

The data samples are categorized into three types to optimize training:
Positive: Frames where an anomaly is actively occurring. Because anomalies evolve rapidly, we utilize a dense sliding window sampling strategy with a 0.5-second stride at 4 FPS, creating temporal overlapping. This captures intricate dynamic details and mitigates the scarcity of anomalous data.
Hard Negative: Normal frames extracted from a video that contains anomalies (e.g., the prelude or aftermath of a crime). These frames are highly challenging distractors. We sample these at 4 FPS without temporal overlapping.
Easy Negative: Frames from completely normal videos without any anomalies. These depict static scenes with minimal variations and are sampled sparsely at 1 FPS to balance the dataset.

The negative subset contains approximately 100K hard and 30K easy negatives, a roughly 3:1 ratio. The finalized dataset comprises approximately $\sim$240K Grid Images with a 1:1 ratio of Positive to Negative samples. Figure 1 presents visual examples comparing a dense positive anomaly sample with a sparse normal negative sample, while Fig. 2 illustrates the boundary between a hard negative sampled immediately before an annotated accident and the subsequent positive grid that overlaps the event interval. These paired examples show how the same camera, scene, and moving agents can yield different labels solely according to the temporal overlap with the anomaly annotation.

Fig. 1: Examples from the constructed Grid Image Dataset. (a) A **Positive Sample** illustrating an actively evolving anomaly (e.g., two men are fighting). Utilizing a dense sliding window (4 FPS), the 2×2 grid captures intricate temporal dynamics and fine-grained variations. (b) An **Easy Negative Sample** showing a completely normal, background scene (e.g., routine city traffic). Sampled sparsely (1 FPS), the grid exhibits minimal semantic variation, providing a stable visual baseline.

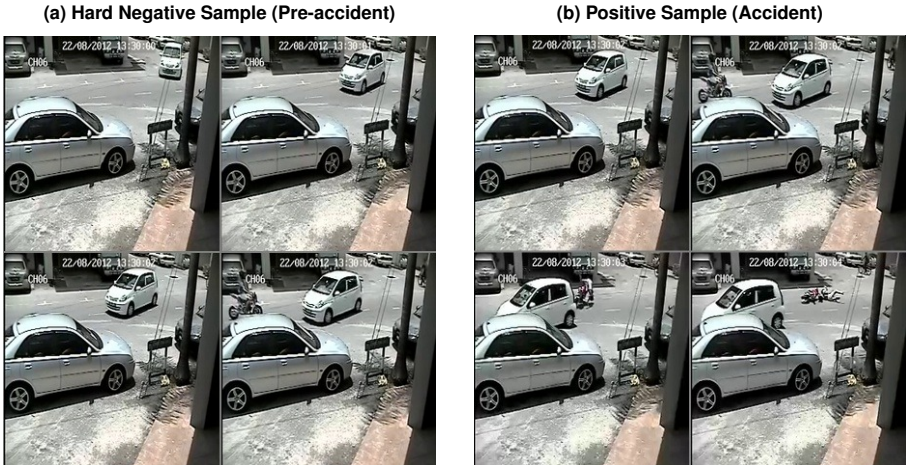

Fig. 2: Hard-negative boundary example. (a) A normal grid sampled immediately before a traffic accident from an anomaly-containing video is labeled as a hard negative because all frames lie outside the annotated event interval. (b) The subsequent grid overlaps the accident interval and is labeled positive, even though its earlier frames still depict normal activity.

2 Detailed Evaluation Protocols

To ensure fair comparison while strictly adhering to our streaming constraints, we formulate two distinct evaluation protocols corresponding to the VAD and VAU tasks:

VAD Protocol (Real-time Frame-level Evaluation): The system continuously processes the video sequence at 4 FPS without observing any future frames. Because the Fast Detection Module generates a single anomaly score per 1-second interval (comprising 4 frames), we compute the interval score $S_{interval}^{(t)}$ at second t based on the gatekeeping mechanism:

$$S_{interval}^{(t)} = \begin{cases} S_{det}^{(t)}, & \text{if } S_{det}^{(t)} \leq \tau_{trigger} \\ \mu \cdot S_{det}^{(t)} + (1-\mu) \cdot S_{reason}^{(t)}, & \text{if } S_{det}^{(t)} > \tau_{trigger} \end{cases} \tag{1}$$

Trigger Threshold Selection. The trigger threshold $\tau_{trigger}$ in the VAD protocol is derived exclusively from the *training* split of each benchmark to prevent any test-set information leakage. Specifically, we precompute the Fast Detection scores S_{det} on all training videos of the respective dataset (UCF-Crime and XD-Violence independently), reconstruct query-level binary labels from the frame-level ground truth, and perform a systematic ROC-based threshold analysis. We select the operating point that guarantees at least 90% recall on the training set (denoted *Recall@90*), yielding dataset-specific thresholds for UCF-Crime and XD-Violence respectively. This recall-oriented criterion ensures that the Slow Reasoning Module is triggered for the vast majority of true anomalous intervals, deliberately accepting a moderate false-positive trigger rate in exchange for near-complete anomaly coverage—since the subsequent semantic verification by the Slow Module effectively suppresses spurious activations.

To align with the frame-level AUC and AP evaluation metrics required by UCF-Crime and XD-Violence, which demand an anomaly score for every single frame in the video, we apply a broadcasting strategy. Let N denote the original frame rate (FPS) of the given video (e.g., $N = 30$). The generated interval score $S_{interval}^{(t)}$ is duplicated and assigned to all N frames within that specific t-th second:

$$S_{frame}^{(N \cdot t+i)} = S_{interval}^{(t)}, \quad \forall i \in \{0, 1, \ldots, N-1\} \tag{2}$$

Finally, to eliminate abrupt score fluctuations and bridge short detection gaps without violating causality (i.e., no future frame observation), we replace standard offline Gaussian smoothing with a strictly causal $O(1)$ Online Smoother. For each time step t, the smoothed score $S_{smooth}^{(t)}$ is computed sequentially by combining an Exponential Moving Average (EMA) and a Peak Holding mechanism:

$$E^{(t)} = \alpha \cdot S_{frame}^{(t)} + (1-\alpha) \cdot S_{smooth}^{(t-1)} \tag{3}$$

$$S_{smooth}^{(t)} = \max(E^{(t)}, \beta \cdot S_{smooth}^{(t-1)}) \tag{4}$$

where α controls the EMA responsiveness (acting as a low-pass filter to reduce jitter), and β dictates the peak decay rate, effectively preventing sharp score drops mid-event. This strictly causal sequence S_{smooth} is then used to calculate the real-time detection performance against the ground truth labels.

VAU Protocol (Post-event Semantic Evaluation): Existing benchmarks such as HIVAU-70K are inherently structured for offline evaluation, where a global question-answer prompt is presented after the entire video has been observed. We faithfully adapt this paradigm to our streaming framework while preserving full comparability with prior offline methods.

During video ingestion, the Fast Detection Module processes each 1-second interval and produces a detection score $S_{det}^{(t)}$. Meanwhile, every sampled frame is simultaneously fed into the AAPM. Critically, the detection score $S_{det}^{(t)}$ is *not* exposed to the language prompt; instead, it is exclusively consumed by the AAPM mechanism to modulate token retention priorities. Specifically, when $S_{det}^{(t)}$ exceeds the pool admission threshold τ_{pool}, the corresponding visual tokens are compressed and injected into the dedicated Anomaly Pool, and the Anomaly Priority Score (APS) protects the associated PEMF events from aggressive temporal compression. In the VAU setting, $\tau_{trigger}$ is fixed at 0.4 to ensure that the AAPM mechanism is activated with sufficient sensitivity—capturing visually suspicious intervals even before they surpass a strict detection confidence—so that critical anomaly cues are preserved for downstream reasoning.

The Slow Reasoning Module is queried only upon the conclusion of the video sequence. At this point, it assembles the fully accumulated, anomaly-protected memory sequence $\mathcal{M}_{seq} = [\mathcal{M}_{PEMF}; \mathcal{M}_{ST}; \mathcal{M}_{pool}; \mathcal{M}_{RTP}]$ and receives the original open-ended questions provided by the HIVAU-70K benchmark verbatim. Here, $\mathcal{M}_{ST}$ denotes the short-term memory component of FSTW, while $\mathcal{M}_{RTP}$ denotes its current high-resolution perception tokens. No textual side-information or detection alerts from the Fast Module are injected into the language prompt, ensuring an entirely fair comparison with other generic MLLMs and offline VAU methods. The generated textual descriptions are then evaluated against the ground truth captions using standard NLG metrics (BLEU, ROUGE, CIDEr, METEOR).

3 Implementation Details

3.1 Prompt Engineering.

Since ReactVAU relies on token-level logit extraction ("Yes"/"No") to compute S_{det} and S_{reason}, the exact wording of prompts directly influences score calibration and output stability. We provide the verbatim prompt templates below.

Fast Detection Prompt ($\mathcal{Q}_{det}$). During both training and testing, the Fast Module (PaliGemma2-3B) receives the 2×2 Grid Image together with the following instruction:

```
Analyze the 2x2 video grid. Is there any anomalous behavior, safety
hazard, or event that disrupts the normal scene? Answer:
```

During training, the ground-truth response is purely formatted as either "Yes" or "No".

Slow Reasoning Prompt ($\mathcal{Q}_{reason}$). The behavior of the Slow Module explicitly differs between the pure detection (VAD) and semantic understanding (VAU) tasks to avoid prompt bias while fulfilling distinct evaluation protocols:

- **VAD Task (Real-time Anomaly Verification):** When testing on UCF-Crime and XD-Violence, the Slow Module is awakened upon detection ($S_{det} > \tau_{trigger}$) to act as a definitive semantic gatekeeper against false positive alarms. The accumulated memory sequence $\mathcal{M}_{seq}$ is provided alongside the following explicit verification prompt:

```
A detection module has flagged this timestamp with {score_pct}%
confidence. {anomaly_context}

Note: This detector frequently produces false alarms from camera
motion, lighting changes, and normal fast movements. Many of these
alerts turn out to be false alarms.

You have been monitoring this video stream from the beginning and
maintain continuous visual memory. Your task is to independently
VERIFY whether this is a true anomaly or a false alarm.

Carefully assess:

1. What is actually happening in the current scene?

2. Compare with the normal and abnormal patterns you have observed
-- is this genuinely different?

3. Is there clear evidence of anomalous behavior -- such as safety
hazards, violence, accidents, or events disrupting normal activity?

Do not rely solely on the detection confidence. Only answer 'Yes'
if you see clear visual evidence of a genuine anomaly.

Is there a confirmed anomaly happening right now? Answer 'Yes' or
'No'.
```

- **VAU Task (Training and Semantic Description):** For both training and testing on HIVAU-70K, we strictly use the original open-ended questions provided by the dataset to ensure entirely fair comparison with other generic MLLMs and offline VAU methods. No textual side-information or alerts from the Fast Module are passed into the prompt. The fast detection score (S_{det}) is exclusively used internally by the AAPM mechanism.

3.2 Training Configuration.

ReactVAU undergoes a sequential two-stage training paradigm. In the first stage, the Fast Detection Module (PaliGemma2-3B) is fine-tuned to extract spatial anomaly features and predict binary anomaly scores (S_{det}) accurately from grid

images. The second stage then focuses on instruction-tuning the Slow Reasoning Module (StreamForest-7B) on the multi-granular HIVAU-70K dataset. Crucially, these two stages are not isolated. During the training of the reasoning MLLM in the second stage, we utilize the real-time anomaly scores S_{det} precomputed by the trained Fast Module to actively trigger and update the AAPM dynamics (e.g., modulating anomaly priority and pool admission). Consequently, the primary training objective of the second stage is inherently formulated to teach the Reasoning Module how to effectively comprehend and leverage the anomaly-protected historical context stored within the AAPM for accurate semantic anomaly understanding. Both stages employ LoRA to efficiently adapt the pre-trained weights while preserving the vision tower representations. The complete hyperparameter configurations are summarized in Tab. 1.

Table 1: Training Configuration. Detailed hyperparameters for fine-tuning both modules. All experiments use mixed-precision (bf16) training.

Hyperparameter	**Fast Detection (PaliGemma2)**	**Slow Reasoning (StreamForest)**
LoRA Configuration		
Rank (r)	16	64
Alpha (α)	32	16
Dropout	0.05	0.05
Bias	none	none
Target Modules	LLM Linear Layers	LLM Linear Layers
Optimization		
Optimizer	AdamW (Fused)	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)
Base Learning Rate	2×10^{-6}	1×10^{-5}
Vision Tower LR	– (Frozen)	– (Frozen)
LR Scheduler	Cosine	Cosine Decay
Warmup Ratio	0.05	0.03
Weight Decay	0.01	0.01
Training Protocol		
Tunable Parts	Projector + LoRA (LLM)	Projector + LoRA (LLM)
Epochs	1	1
Per-Device Batch Size	16	1
Gradient Accumulation	2	2
Effective Batch Size	32	2
Precision	bf16 (TF32 enabled)	bf16 (TF32 enabled)
Max Sequence Length	96 (Text) + 729 (Image)	8192 tokens
Gradient Checkpointing	×	✓
Training Data		
Dataset	Grid Image Dataset (∼240K)	HIVAU-70K instruction data
Infrastructure		
Hardware	1× H100-80GB	1× H100-80GB
Distributed Strategy	None	DeepSpeed ZeRO-Stage 1
Attention Implementation	Eager	SDPA
Training Time	∼12–14 hours	∼14–16 hours

3.3 Inference Hyperparameters.

The key inference-time hyperparameters governing the Slow-Fast Decoupled Framework are listed below:

- **Trigger Threshold** ($\tau_{trigger}$): Dataset-specific for VAD; fixed at 0.4 for VAU and general inference. For the VAD task, the threshold is derived from training-set *Recall@90* analysis as described in Sec. 2 on UCF-Crime and XD-Violence independently, ensuring near-complete anomaly coverage without test-set information leakage. For the VAU task and general deployment, a fixed value of 0.4 is adopted to maintain sufficient sensitivity for triggering the AAPM mechanism and the Slow Reasoning Module upon observing visually suspicious events.
- **Pool Admission Threshold** (τ_{pool}): 0.6. Governs entry into the Anomaly Pool $\mathcal{M}_{pool}$.
- **Confirmation Threshold** ($\tau_{confirm}$): 0.5. Final gating for anomaly declaration after score fusion.
- **Score Fusion Weight** (μ): 0.4 (as described in the main paper).
- **AAPM Scaling Constant** (κ): 4.0. Controls the exponential protective weight in P_a.
- **EMA Responsiveness** (α in Eq. 3): 0.6.
- **Peak Decay Rate** (β in Eq. 4): 0.95.

4 Memory Token Budget

The assembled memory sequence $\mathcal{M}_{seq}$ combines long-term event memory, recent short-term context, anomaly-focused evidence, and dense current perception. Table 2 reports the maximum visual-token allocation when the Slow Reasoning Module is triggered.

Table 2: Memory token budget. Maximum visual-token allocation of each component in $\mathcal{M}_{seq}$ when Slow Reasoning is triggered.

Component	Composition	Max Tokens
$\mathcal{M}_{PEMF}$ (Long-term)	$C_{max} = 2048$ tokens	2,048
$\mathcal{M}_{ST}$ (Short-term)	12 frames $\times$ 128 tokens/frame	1,536
$\mathcal{M}_{pool}$ (Anomaly Pool)	8 frames $\times$ 128 tokens/frame	1,024
$\mathcal{M}_{RTP}$ (Dense Perception)	4 frames $\times$ 729 tokens/frame	2,916
Total $\lvert\mathcal{M}_{seq}\rvert_{max}$		**7,524**

The maximum visual memory occupies 7,524 tokens under the 8,192-token model input limit, leaving the remaining context for the task prompt and other text tokens. The Anomaly Pool contributes 1,024 tokens, only about 13.6% of the

triggered visual context. This allocation gives the Slow Reasoning Module explicit high-density evidence for suspicious intervals while preserving most of the context window for PEMF, short-term memory, and dense current perception. In practice, this balance lets the model compare a suspected anomaly against both normal background history and recent motion rather than reasoning from anomaly evidence alone.

5 Robustness to Fast Module Miscalibration

The two-stage training procedure introduces a one-way dependency: Fast Detection scores control AAPM retention and pool admission during Slow Module training. To test whether noisy scores corrupt this process, we replace the converged Fast Module with a checkpoint obtained at 1/10 of the Stage-1 training steps while keeping the AAPM-trained Slow Module fixed. Table 3 summarizes the standalone Fast degradation and the corresponding full-pipeline robustness.

Table 3: Robustness to Fast Module miscalibration. An under-trained Fast checkpoint substantially degrades standalone detection, whereas Slow-side verification limits the degradation of the full ReactVAU pipeline.

Configuration	Fast Checkpoint	UCF-Crime	XD-Violence	
		AUC (%)	AP (%)	AUC (%)
Baseline	StreamForest†	85.26	75.92	92.82
Fast Module only	Converged	**85.37**	**79.55**	**91.18**
	1/10-step	81.85	68.55	85.42
ReactVAU (Ours)	Converged	**88.44**	**88.50**	**95.25**
	1/10-step	85.86	85.52	93.84

Although the under-trained detector loses 3.52 points on UCF-Crime AUC and 11.00 points on XD-Violence AP, ReactVAU retains 85.86% AUC and 85.52% AP, respectively, remaining above StreamForest†. This gap between the standalone Fast degradation and the full-system degradation is important: noisy Fast scores can admit imperfect evidence into AAPM, but the Slow Module still observes the accumulated memory and performs semantic verification before the final decision. On XD-Violence, the under-trained full pipeline loses only 2.98 AP points relative to converged ReactVAU, compared with the 11.00-point AP drop of the Fast Module alone. Thus the Stage-2 dependency on Fast scores is real but bounded by Slow-side reasoning; a well-calibrated Stage-1 detector remains the recommended operating point for full performance.

6 Cross-Dataset Trigger Threshold Transfer

Only $\tau_{trigger}$ is calibrated per VAD benchmark using Recall@90 on its training split; the fusion, AAPM, and smoothing hyperparameters remain fixed. We transfer each training-derived threshold directly to the other test benchmark without further tuning. Table 4 reports the resulting cross-dataset performance.

Table 4: Cross-dataset transfer of $\tau_{trigger}$. Each threshold is calibrated only on the training split of the indicated source dataset.

Threshold Source	UCF-Crime	XD-Violence	
	AUC (%)	AP (%)	AUC (%)
UCF-Crime	**88.44**	**88.52**	**95.32**
XD-Violence	88.29	88.50	95.25

The XD-Violence threshold reduces UCF-Crime AUC by only 0.15 points. Conversely, the lower UCF-Crime threshold triggers Slow verification slightly more often on XD-Violence and changes AP/AUC by no more than 0.07 points. The slight gain on XD-Violence follows the same efficiency–accuracy trade-off shown in the main paper: a lower threshold awakens the Slow Module on a few additional suspicious intervals, recovering marginal anomalies at the cost of more computation. The small bidirectional change indicates that the Recall@90 operating point is not narrowly tied to one benchmark and remains near a stable knee of the threshold–efficiency Pareto curve across surveillance domains.

7 Per-Query Latency on Consumer Hardware

The main-paper efficiency table profiles aggregate throughput on the full UCF-Crime test set using a single H100-80GB GPU. To complement that datacenter setting, we profile per-query latency on a consumer RTX 4090-24GB GPU using the 151.8-second UCF-Crime video *Shooting022*. ReactVAU follows the deployed streaming protocol: it ingests at 4 FPS, folds four frames into one grid, and issues one query per second. Every query invokes the Fast Module and updates AAPM, while the Slow Module is invoked only on triggered intervals. Figure 3 visualizes the temporal latency trace, and Tab. 5 summarizes the corresponding path-level statistics.

Table 5 shows that ReactVAU exposes two distinct latency paths. The normal-state path accounts for 118 of 163 queries and averages only 76.8 ms/query, with a 96.8 ms P95 and 183.0 ms maximum. This corresponds to approximately 13 queries/s of throughput, far above the deployed one-query-per-second streaming rate. The Slow-verification path is heavier, averaging 966.6 ms/query with a 1090.8 ms maximum, but it is invoked on only 45 queries (27.6%). As a result, the overall mean latency is 322.5 ms/query, and the median remains 75.0 ms because most streaming intervals follow the normal-state path. In contrast, the always-on StreamForest† baseline pays the Slow cost at every second, yielding a 943.2 ms mean latency and no low-latency operating mode.

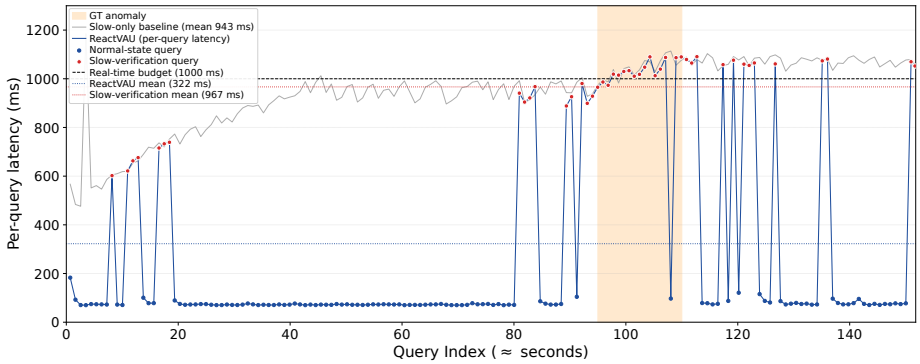

Fig. 3: Per-query latency timeline on RTX 4090-24GB. Blue points denote the normal-state path (Fast + Memory), red points denote the Slow-verification path (Fast + Memory + Slow), and the gray curve denotes the always-on Slow baseline. The dashed line marks the one-second query budget.

Table 5: Per-query latency statistics on RTX 4090-24GB. Measurements are taken on UCF-Crime *Shooting022* with 163 streaming queries. Backlog accumulates latency above the one-second query budget across consecutive queries and is reported only for the overall pipeline.

Configuration	#Queries	Per-query Latency (ms)					Backlog (ms)	
		Mean	P50	P95	P99	Max	Max	Final
ReactVAU (27.6% Slow-trigger rate)								
Normal-state path (Fast + Memory)	118	**76.8**	73.1	96.8	120.1	183.0	—	—
Slow-verification path (Fast + Memory + Slow)	45	966.6	1018.3	1089.3	1090.5	1090.8	—	—
Overall	163	322.5	75.0	1073.2	1089.9	1090.8	**411.0**	**122.3**
StreamForest† (Slow-only)	163	943.2	973.9	1091.0	1109.4	1184.8	4003.3	4003.3

Figure 3 explains the system-level effect of these statistics. Although consecutive Slow-verification calls can briefly push individual queries near or slightly above the one-second budget, the surrounding normal-state queries run far below the budget and recover the accumulated delay. ReactVAU therefore keeps the maximum backlog bounded at 411.0 ms and ends with only 122.3 ms residual backlog. The always-on baseline instead accumulates a persistent 4.0-second backlog because every query is near the Slow path latency. These results show that event-gated Slow activation preserves practical streaming responsiveness on consumer hardware, while asynchronous Slow inference and edge-device profiling remain valuable future work.

8 Fast and Slow Module Complementarity

The anomaly-score timeline in Fig. 4 visualizes how the two modules contribute different temporal behaviors.

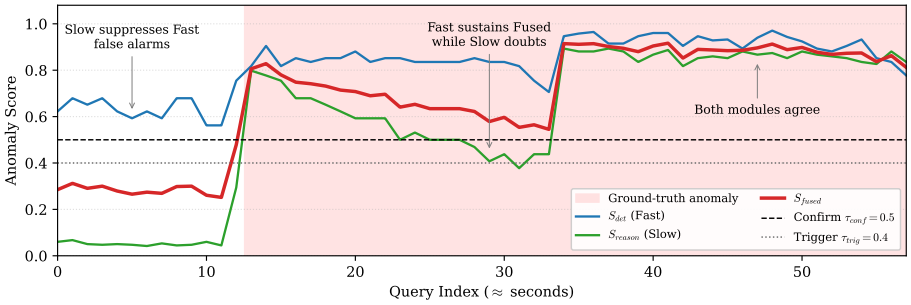

Fig. 4: Fast/Slow complementarity on an XD-Violence explosion clip. Before the anomaly, Slow verification suppresses Fast-side false alarms. During the event, the Fast score preserves an abrupt threat response when long-context Slow reasoning briefly drops, and their weighted fusion maintains the confirmed detection.

The Fast Module reacts strongly to local visual disruptions, whereas the Slow Module verifies them against accumulated context. Before the event, S_{det} produces several elevated responses, but S_{reason} remains low and suppresses 12 Fast-side false alarms after fusion. During the explosion, the situation reverses: around query 28, S_{reason} briefly falls below $\tau_{confirm}$ as the long-context reasoning score smooths over a sudden visual change, but S_{det} remains high and keeps S_{fused} above the confirmation threshold. This qualitative timeline matches the main-paper score-fusion ablation: replacing the final score with Slow-only reasoning loses abrupt threat sensitivity, while weighted fusion retains both local reactivity and semantic verification.

9 Ablation: Online Smoother Sensitivity Analysis

The Online Smoother introduces two hyperparameters: the EMA responsiveness α and the peak decay rate β (Eqs. 3–4). Rather than tuning these on the test set, we motivate their values from the temporal dynamics of real-world anomalies, then verify via exhaustive grid search that the design generalizes robustly across datasets. The full sensitivity landscape is shown in Fig. 5.

Domain-Driven Parameter Design. The EMA coefficient $\alpha = 0.6$ weights the current observation at 60% and retains 40% of prior context, yielding an effective time constant of $\frac{1}{1-\alpha} = 2.5$ query steps (≈ 2.5 seconds at 1 query/s). This matches the minimal observable anomaly duration in surveillance benchmarks. The peak-hold factor $\beta = 0.95$ governs the post-peak decay rate: the score halves in $\frac{\log 0.5}{\log 0.95} \approx 13.5$ steps (≈ 13 seconds), consistent with typical anomalous clip lengths in UCF-Crime and XD-Violence. Both values were determined by reasoning about anomaly temporal structures, not by maximizing any test-set metric.

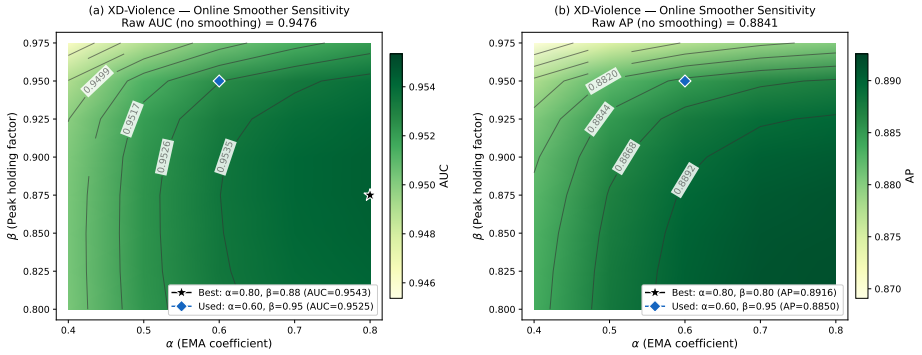

Fig. 5: Online Smoother parameter sensitivity under dual-metric evaluation on XD-Violence. Each cell in the 5×8 grid reports the VAD performance for a specific (α, β) pair. The $\star$ marks the test-set optimum for the respective metric; the $\diamond$ marks our globally deployed configuration ($\alpha = 0.6$, $\beta = 0.95$). Focusing extensively on XD-Violence provides a rigorous assessment, as AP and AUC present competing optimization surfaces. The smooth, monotonically varying contours confirm the structural EMA + Peak Hold design's effectiveness. The deployed point gracefully balances both criteria within margins of 0.18% and 0.66% to their localized optima, generalizing effectively without test-set tuning.

Parameter Sensitivity Landscape. To verify this choice empirically, we sweep a 5×8 parameter grid ($\alpha \in \{0.4, 0.5, 0.6, 0.7, 0.8\}$, $\beta \in \{0.800, 0.825, \ldots, 0.975\}$) evaluating all 40 combinations. For this comprehensive analysis, we deliberately focus exclusively on the XD-Violence benchmark. As XD-Violence necessitates a rigorous dual-metric evaluation, models are forced to simultaneously balance both Area Under the ROC Curve (AUC) and Average Precision (AP), providing a highly challenging proxy for general robustness. As visualized in Fig. 5, the performance landscape is smooth and nearly monotonically increasing towards larger (α, β), exhibiting no sharp peaks or narrow valleys. The total performance variation across all 40 configurations is only **0.79%** (AUC) and **2.15%** (AP), confirming that the causal EMA + Peak Hold structure—rather than any particular parameter value—is the primary driver of improvement over unsmoothed scores.

Crucially, our deployed configuration ($\alpha = 0.6$, $\beta = 0.95$, shown as $\diamond$ in Fig. 5) is *not* co-located with any per-metric test-set optimum. It falls within **0.18%** (AUC) and **0.66%** (AP) of the respective best-found settings without being tuned toward any of them. Moreover, the per-metric optima are mutually inconsistent—AUC favors (α=0.80, β=0.88) while AP favors (α=0.80, β=0.80), confirming that no single configuration can simultaneously maximize all metrics. The fact that a single shared setting achieves near-optimal performance across these competing evaluative axes strongly indicates that the robustness derives from the causality-driven design itself, not from test-set memorization.

Behavioral Analysis of the Causal Smoother. Figure 6 presents three representative score trajectories to illustrate how each structural property of the smoother manifests on real anomaly sequences.

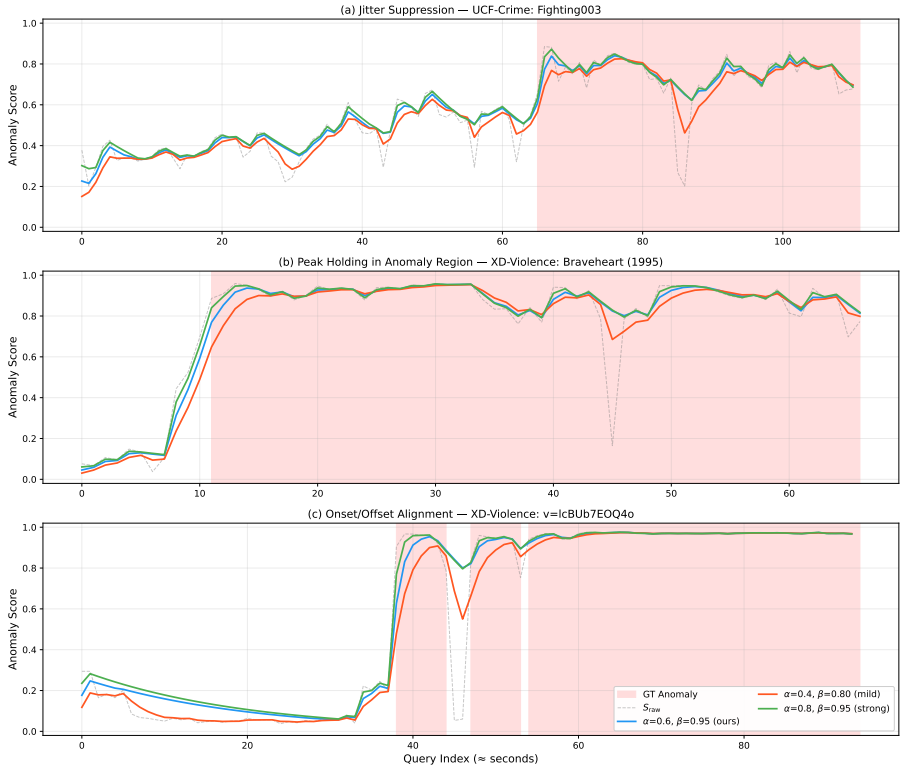

Fig. 6: Qualitative behavior of the Online Smoother on representative videos. Pink shading indicates ground-truth anomaly intervals. The dashed gray curve is the raw unsmoothed score S_{raw}; colored curves correspond to three (α, β) configurations: mild (orange), ours (blue), and strong (green). (a) The smoother suppresses high-frequency jitter in normal segments, improving signal clarity. (b) Peak holding ($\beta = 0.95$) bridges intra-event score dips, providing continuous anomaly coverage throughout sustained events. (c) A known limitation: when a genuine normal interval separates two anomalous events, peak holding may over-hold, conflating distinct event boundaries. Our chosen configuration ($\alpha = 0.6$, $\beta = 0.95$) provides the best practical balance between jitter suppression and onset-offset responsiveness.

(a) Jitter Suppression (UCF-Crime: Fighting003): Unsmoothed raw scores exhibit high-frequency fluctuations in nominally normal background segments, where minor motion changes cause erratic score oscillations. Our smoother ($\alpha = 0.6$, $\beta = 0.95$) acts as a causal low-pass filter, yielding a stable and monotonically rising trajectory that significantly increases the signal-to-noise ratio prior to the confirmed anomaly onset near $t = 65$s. **(b) Peak Holding in Sustained Anomaly Regions (XD-Violence: Braveheart):** Within a continuous anomalous event, transient score dips caused by occlusions or camera cuts (visible as the raw score drop around $t \approx 45$s) would otherwise fragment a single anomaly into a sequence of disjointed short detections. The peak-hold mechanism ($\beta = 0.95$) effectively bridges these intra-event gaps, sustaining a high-confidence

score throughout the full anomaly duration and preventing false-negative interruptions. **(c) Trade-off in Rapid Anomaly–Normal–Anomaly Sequences (XD-Violence: v=icBUb7EOQ4o):** When a genuine brief normal interval separates two distinct anomalous events, the unsmoothed raw score correctly recovers toward zero during the gap. In this case, aggressive peak holding can be counterproductive by preventing the score from dropping during the intervening normal segment. This reveals a known trade-off: while beneficial for bridging spurious intra-event drops, extensive peak holding slightly reduces proper onset-offset alignment precision on sequences with rapidly alternating event structures.