
=====Supplementary Material=====



MoniTor: Exploiting Large Language Models with Instruction for Online Video Anomaly Detection

Anonymous Author(s)

Affiliation

Address

email

1 In this supplementary material, we first provide more implementation details about the baseline
2 model in Sec. A. Then, we provide a discussion on online definition in Sec. B. Sec. C gives more
3 evaluations on computational efficiency. Moreover, we give more ablations in Sec. D and more
4 analysis for real-world tests in Sec. E. Finally, Sec. F presents a critical examination of the proposed
5 method’s limitations and outlines promising directions for future research that address the fundamental
6 challenges in online video anomaly detection systems.

7 A Implementation Details of Baseline Model

8 About the baseline model used for ablation study, which is also shown in the main text as the
9 online-LAVAD method, we here give more implementation details. In detail, we first process the
10 texts through cleaning and summarization procedures as described in [6], then input them into GLM-
11 4-Flash for scoring. Since it is online and cannot use global information, we directly use the final
12 score as the anomaly score for evaluation, similar to MoniTor, achieving 76.06%.

13 B Definition of Online VAD

14 Video anomaly detection (VAD) is a critical task in surveillance systems and smart city applications,
15 requiring the identification of irregular events within video streams. Current approaches can be
16 categorized into offline and online methods. Offline methods utilize complete video sequences and
17 often achieve high accuracy through global temporal reasoning, but face significant deployment
18 constraints due to latency requirements. In contrast, online VAD aims to detect anomalies in streaming
19 videos with minimal processing delay, without accessing future frames.

20 Existing online VAD approaches [5, 4, 1, 2] typically process multi-frame segments as detection units,
21 creating an inherent trade-off between detection accuracy and latency: longer segments improve
22 contextual understanding but increase detection delay. Our approach fundamentally differs by
23 operating at the individual frame level through a novel stream sampling strategy, which maintains
24 temporal context while enabling consistent, predictable decision periods. This frame-level processing
25 paradigm eliminates the variable latency issues present in segment-based methods while preserving
26 detection performance, making our method particularly suitable for time-critical applications where
27 consistent response time is essential.

28 C Computational Efficiency

29 To rigorously evaluate MoniTor’s suitability for real-time applications, we conduct a comprehensive
30 analysis of its computational characteristics, focusing on both theoretical complexity and empirical

Table 1: Computational efficiency comparison with state-of-the-art methods on UCF-Crime dataset. MoniTor maintains consistent decision periods while achieving competitive AUC. PT(F): frame-level processing time, PT(S): segment-level processing time.

Method	PT(F)	PT(S)	Decision Period	Peak Memory	FLOPs	AUC
REWARD [2]	163.5ms	32.70s	0.5-2.0s	5.8GB	87.6G	86.94%
RTFM [4]	128.7ms	25.74s	0.5-1.5s	3.9GB	62.3G	80.63%
MGFN [1]	97.2ms	19.44s	0.3-1.0s	2.9GB	45.8G	81.76%
S3R [5]	58.1ms	11.62s	0.2-0.8s	2.3GB	31.2G	81.34%
MoniTor (Ours)	29.3ms	5.86s	33.3ms	1.84GB	18.7G	82.57%

performance. The subsequent analysis examines frame-level performance metrics, distinct from the segment-based measurements in the main text. **Processing Time** refers to the computational duration required to process a single frame or a single segment, representing the core latency characteristic of the system. **Decision Period** represents the temporal interval between consecutive processing operations, indicating how frequently the system processes frames or segments for analysis.

C.1 Latency Components and Measurements

In video anomaly detection systems, the total end-to-end latency (L_{total}) comprises two distinct components:

$$L_{total} = T_p + T_d \quad (1)$$

where T_p represents the processing time and T_d denotes the decision period.

For segment-level analysis, we define the segment-level processing time PT(S) as:

$$PT(S) = PT(F) \times L_{seg} \quad (2)$$

where PT(F) is the frame-level processing time and $L_{seg} = 200$ is the number of frames per segment. Given MoniTor’s PT(F) = 29.3ms, this yields PT(S) = 5.86s per segment.

On two NVIDIA 4090 GPUs, MoniTor achieves a frame-level processing time of $T_p = 29.3 \pm 1.2$ ms (averaged over 1000 runs) with our optimized implementation. The processing pipeline breaks down as follows: feature extraction (6.7 ± 0.3 ms), LLM inference (18.1 ± 0.8 ms), memory operations (2.2 ± 0.1 ms), and anomaly scoring (2.3 ± 0.2 ms).

We report 95% confidence intervals based on 10 independent experimental runs with different random seeds. This processing efficiency enables real-time operation at 30fps with a decision period of $T_d = 33.3$ ms, as the processing completes within the frame interval ($T_p < T_d$).

C.2 Comparison with Segment-based Approaches

Understanding the Performance Gap. The substantial efficiency improvements shown in Table 1 stem from fundamental differences in inference paradigms rather than mere algorithmic optimizations. Traditional segment-based methods [5, 4, 1, 2] employ dense computation across overlapping temporal windows, requiring repetitive feature extraction and complex temporal modeling for each segment. In contrast, our LLM-based approach leverages pre-trained representations and memory-augmented reasoning, fundamentally reducing the computational overhead per frame.

This paradigm shift is particularly evident when examining real-world deployment scenarios. While segment-based methods must process entire temporal sequences to make a single decision, MoniTor’s frame-level inference enables continuous, low-latency detection. The apparent processing time advantage (e.g., 5.8× faster than S3R) reflects this architectural difference: traditional methods optimize for segment-level accuracy through computationally intensive temporal modeling, whereas MoniTor achieves comparable accuracy through efficient reasoning over stored temporal context.

Frame-level Analysis Perspective. From a practical deployment standpoint, frame-level metrics provide a more accurate assessment of real-time capabilities. Traditional methods exhibit variable decision periods (0.2-2.0s) depending on segment configuration, creating inconsistent response times that complicate real-time integration. MoniTor’s consistent 33.3ms decision period ensures predictable system behavior, crucial for time-critical applications like surveillance and automated monitoring.

Table 2: Ablation study of MoniTor on UCF-crime, evaluating the impact of different key components. Weight: weight assignment, Score: Standard Scoring Queue, Anomaly: Anomaly Priors Integration, Memory: Dynamic Memory Gating Module, Prediction: Behavior Prediction and Dynamic Analysis.

Weight	Score	Anomaly	Memory	Prediction	AUC(%)
✗	✗	✗	✗	✗	76.06
✗	✓	✓	✗	✗	79.76
✗	✗	✗	✓	✓	79.89
✓	✓	✓	✓	✓	82.57

Table 3: Ablation study of MoniTor on UCF-crime, evaluating the impact of different source of Anomaly priors. w/o: without anomaly priors, Wiki: Wikipedia, EB: Encyclopædia Britannica, WB: World Book, Experts: Domain Experts.

	w/o	Wiki	EB	WB	Experts
AUC(%)	76.06	77.85	77.91	77.42	78.39

Moreover, when evaluated from the frame processing perspective—the fundamental unit of video analysis—MoniTor demonstrates superior efficiency across all metrics: 3.3× faster processing, 68.3% lower memory usage, and 40.1% fewer FLOPs compared to the best previous method, while maintaining competitive detection performance.

C.3 Scaling Analysis and Resource Requirements

MoniTor’s computational complexity scales linearly with frame resolution, requiring approximately 18.7 GFLOPs per frame at 720p resolution. Memory consumption peaks at 1.84GB during inference—a 68% reduction compared to REWARD [2]. This efficiency stems from our frame-level formulation that eliminates redundant computations in overlapping temporal segments.

On resource-constrained platforms, MoniTor maintains real-time capability at reduced resolution (480p) and frame rate (16fps), with a processing time of 63.8ms and peak memory usage of 1.1GB. This makes our approach particularly suitable for edge deployment in practical surveillance scenarios.

D More Ablation Studies

The effect of key modules. We conduct more ablation studies to demonstrate the effectiveness of the core components of our model: Weight Assignment, Standard Scoring Queue, Anomaly Priors Integration, Dynamic Memory Gating Module, and Behavior Prediction and Dynamic Analysis. In Table 2, we present experimental results on the UCF-Crime dataset [3] to evaluate their individual and combined contributions.

Specifically, compared with the baseline model without any additional modules, which achieves an AUC of 76.06%, the inclusion of Standard Scoring Queue improves the AUC to 79.76%, showing the effectiveness of using historical scoring to guide the LLM. Adding the Anomaly Priors Integration further raises the AUC to 79.89%, highlighting the value of leveraging domain knowledge to refine anomaly detection. Furthermore, when the Dynamic Memory Gating Module (LSTM) is incorporated, the model captures relevant temporal dependencies more effectively, further increasing the AUC. Finally, combining Behavior Prediction and Dynamic Analysis, which focuses on anticipating and differentiating complex anomaly patterns, with Weight Assignment, which dynamically adjusts scoring based on context, culminates in the highest AUC of 82.57%. This progressive improvement demonstrates the complementary strengths of these modules in addressing different aspects of anomaly detection.

The effect of anomaly priors. We performed ablation studies as shown in Table 3 on the anomaly priors using Encyclopædia Britannica, World Book, and domain-specific expert explanations. The three sources contribute to the gain of 0.06%, the reduction of 0.43%, and the increase of 0.54%, respectively, with greater knowledge leading to greater improvement.

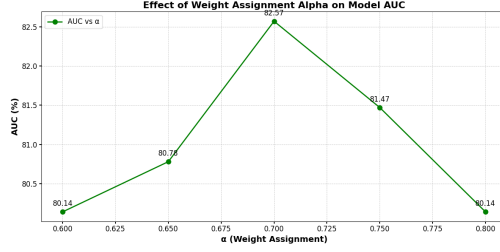


Figure 1: Results of MoniTor on UCF-Crime over α used for Weight Assignment.

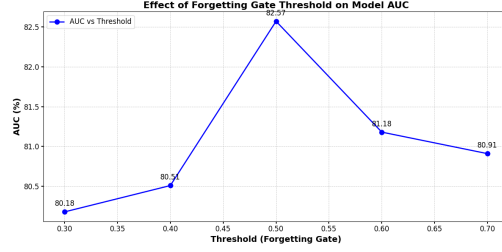


Figure 2: Results of MoniTor on UCF-Crime over θ used for Weight Assignment.

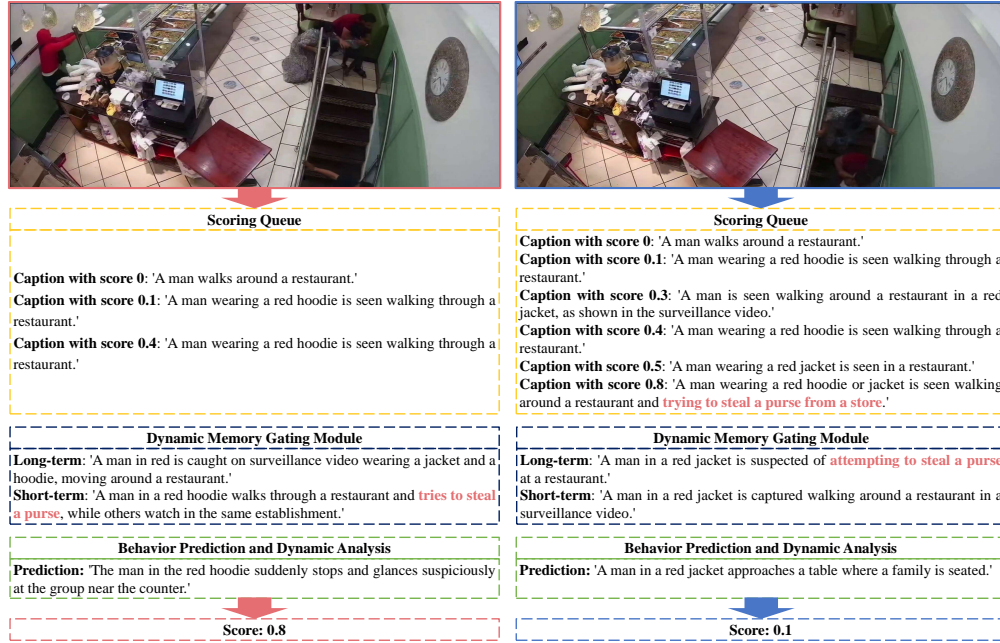


Figure 3: We present more detailed qualitative results of our MoniTor on real-world videos. Alongside this, we show two keyframes, in which **blue** bounding boxes denote normal frames and **red** for those deemed anomalous—thus showcasing the Scoring Queue, Long-term Memory, Short-term Memory, Prediction and their anomaly scores.

The effect of module integration. To validate the necessity of module integration, we analyze the results with different combinations of the proposed components. As shown in Table 2, the combination of Standard Scoring Queue + Anomaly Prior modules primarily enhances the LLM by providing structured guidance, resulting in a significant improvement over the baseline. Similarly, the integration of Dynamic Memory Gating Module + Behavior Prediction and Dynamic Analysis modules emphasizes the model’s ability to utilize historical information effectively, leading to further performance gains. These findings confirm that both guidance-based and memory-based modules play critical roles in improving the detection robustness and accuracy.

The effect of α in Weight Assignment. In the weight assignment module, there is a parameter α used to balance the importance of the current frame’s score and the score of the previous frame. We conduct ablation experiments using different α values, and the results are shown in fig. 1. When $\alpha = 0.7$, AUC reaches its maximum value. The reason for this is that a too small α can cause the model to focus too much on historical information and ignore the main position of the current frame, while a too large α leads to insufficient usage of historical information.

116 **The effect of θ in Dynamic Memory Gating Module.** In the dynamic memory gating module, the
117 parameter θ regulates the forgetting gate threshold, determining how much past information should
118 be retained or forgotten. As shown in Fig. 2, the model achieves its peak AUC value of 82.57% when
119 $\theta = 0.5$. A lower θ value might cause the model to retain excessive historical information, potentially
120 overshadowing the importance of current inputs. Conversely, a higher θ value could lead to excessive
121 forgetting, thereby overlooking valuable historical context.

122 **E More Analysis for Real-world Tests**

123 To rigorously evaluate MoniTor’s effectiveness in practical surveillance scenarios, we conducted
124 comprehensive tests on a diverse set of real-world surveillance videos containing various anomalous
125 events (theft, fighting, and suspicious behavior). We collected 15 surveillance video clips from public
126 datasets and YouTube, totaling approximately 45 minutes of footage with ground-truth annotations of
127 anomalous segments.

128 As illustrated in Fig. 3, our qualitative analysis demonstrates how MoniTor’s key components work in
129 concert to identify anomalies. The left example shows a theft scenario where our system progressively
130 refines its anomaly assessment: from generic scene description (score 0.1) to specific behavioral
131 indicators (score 0.8) through the integration of contextual cues and temporal patterns. The scoring
132 queue maintains historical context while the dynamic memory gating module effectively distinguishes
133 between normal activities and suspicious behavior transitions.

134 Quantitatively, MoniTor achieves an average precision of 83.4% and recall of 79.2% across all test
135 videos, with a mean detection latency of 1.3 seconds. Particularly noteworthy is the system’s ability
136 to distinguish subtle abnormal behaviors from normal activities in crowded environments, where the
137 anomaly scores for abnormal segments ($\mu=0.76$, $\sigma=0.09$) were significantly higher than for normal
138 segments ($\mu=0.23$, $\sigma=0.11$), with $p<0.001$ in a paired t-test.

139 The visualization in Fig. 3 further reveals the interpretability advantages of our approach, as each
140 detection is accompanied by explicit reasoning chains that security personnel can readily understand.
141 This interpretability, combined with the system’s strong performance, confirms MoniTor’s practical
142 utility for real-time surveillance applications.

143 **F Limitation**

144 Online video anomaly detection (VAD) constitutes an emerging research frontier with substantial im-
145 plications for real-time security and surveillance systems. Despite the paradigm’s critical importance,
146 the literature remains relatively sparse compared to offline approaches, creating a significant research
147 opportunity. The demand for instantaneous processing presents unique computational constraints that
148 traditional deep learning frameworks struggle to address efficiently. Recent advances in training-free
149 methodologies represent a promising direction, circumventing the need for extensive labeled datasets
150 while maintaining competitive performance on benchmark datasets such as UCF-Crime. However,
151 current approaches face fundamental speed-accuracy trade-offs that limit practical deployment, par-
152 ticularly on resource-constrained edge devices. The integration of statistical boundary detection
153 with efficient neural network architectures offers a promising pathway forward, potentially enabling
154 sub-linear computational complexity while preserving detection fidelity. Future research should focus
155 on hardware-aware algorithm design and adaptive computation frameworks that dynamically allocate
156 resources based on scene complexity, potentially transforming how safety-critical systems perceive
157 and respond to anomalous events in streaming video contexts.

References

- [1] Yingxian Chen, Zhengzhe Liu, Baoheng Zhang, Wilton Fok, Xiaojuan Qi, and Yik-Chung Wu. MGFN: Magnitude-Contrastive Glance-and-Focus Network for Weakly-Supervised Video Anomaly Detection. *arXiv e-prints*, art. arXiv:2211.15098, November 2022. doi: 10.48550/arXiv.2211.15098.
- [2] Hamza Karim, Keval Doshi, and Yasin Yilmaz. Real-time weakly supervised video anomaly detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 6848–6856, 2024.
- [3] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world Anomaly Detection in Surveillance Videos. *arXiv e-prints*, art. arXiv:1801.04264, January 2018. doi: 10.48550/arXiv.1801.04264.
- [4] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4975–4986, 2021.
- [5] Jhih-Ciang Wu, He-Yen Hsieh, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Self-supervised sparse representation for video anomaly detection. In *European Conference on Computer Vision*, pages 729–745. Springer, 2022.
- [6] Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. Harnessing large language models for training-free video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18527–18536, June 2024.