ARIA: ADAPTIVE REASONING FOR VIDEO ANOMALY DETECTION WITH VISION-LANGUAGE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) have demonstrated strong reasoning capabilities for video anomaly detection (VAD), enabling both anomaly prediction and human-understandable explanation. Existing VLM-based methods typically process videos segment by segment, allocating similar computation to each segment despite their varying needs for further examination. While some segments can be reliably handled with sparse visual observations, others may benefit from additional evidence, either by examining denser visual observations or by incorporating a complementary interpretation from another VLM. Uniformly applying such additional computation to all segments, however, can incur substantial inference cost. To address this issue, we propose ARIA (**A**daptive **R**eason**I**ng for Video **A**nomaly Detection), an adaptive reasoning framework that selectively allocates additional computation to video segments when needed. Given a segment, ARIA first assesses the reliability of its initial lightweight VLM response using response-level signals together with temporal and semantic context. For segments warranting further examination, ARIA then determines whether to acquire richer visual evidence through denser frame sampling or introduce complementary textual evidence from another VLM. Experiments on UCF-Crime and XD-Violence demonstrate that ARIA improves anomaly detection and explanation quality while reducing inference cost, yielding a favorable performance–computation trade-off. [1]

1 INTRODUCTION

Video anomaly detection (VAD) plays a fundamental role in enabling reliable automated decision-making in safety-critical domains, including video surveillance (Ramachandra et al., 2020), autonomous driving (Yao et al., 2022), and medical diagnosis (Fernando et al., 2021). In practice, an effective VAD system should be not only **accurate** in identifying anomalies, but also **explainable** by providing human-understandable reasoning for its predictions. Owing to the advancement in large language models (LLMs), recent research has increasingly adopted vision-language models (VLMs) (Hurst et al., 2024) for jointly improving detection performance and explanation quality.

Limitations of Existing VLM-Based Pipelines for VAD. As shown in Fig. 1, existing VLM-based VAD methods formulate VAD as a *natural language generation* task (Dosovitskiy et al., 2021), where videos are processed segment by segment, with each segment independently mapped to verbalized predictions and explanations (Ye et al., 2025; Zanella et al., 2024; Yang et al., 2024a; Zhang et al., 2024; 2025a). While this segment-wise formulation enables VLMs to perform fine-grained anomaly detection and explanation, it typically allocates a similar inference budget to every segment, overlooking substantial differences in their reasoning difficulty. In practice, not all segments require the same amount of visual evidence: many can be confidently recognized by skimming only a few frames, whereas ambiguous segments may require denser visual observations and additional attention to resolve uncertainty. For instance, in Fig. 1, the i-th segment captures only the early stage of a shooting event, where sparse observations may be insufficient to clearly establish abnormality. Uniformly increasing visual evidence for all segments could alleviate such ambiguity, but would introduce substantial redundant computation for segments that are already easy to recognize.

[1] The implementation code is available in the supplementary material and will be made publicly available.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This motivates an adaptive inference strategy that allocates visual evidence and computation according to the needs of each segment and raises a fundamental challenge: *How can we enable VLMs to adaptively allocate computation across video segments, allowing more careful examination when needed?*

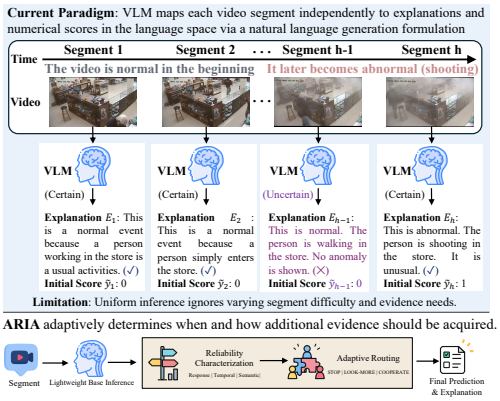

Figure 1: Existing segment-wise VLM pipelines use a uniform inference budget across segments, while ARIA adaptively decides whether and how to perform further examination.

Our Approach. Our key insight is that the initial visual evidence available to a VLM may be sufficient for some video segments but insufficient for others. When the initial response is uncertain, the missing evidence can arise in different forms. A segment may benefit from *more visual evidence*, such as denser frame sampling that reveals fine-grained temporal cues that were missed by sparse observations. Alternatively, it may benefit from a *complementary interpretation*, where another VLM provides an independent textual explanation that highlights relevant cues or alternative interpretations. The usefulness of these two forms of evidence varies across segments, making a uniform follow-up strategy potentially wasteful. This motivates an adaptive strategy that first identifies segments that would benefit from further examination and then selects an appropriate form of additional evidence.

To realize that, we propose ARIA, an adaptive reasoning framework that enhances VLMs for VAD, as shown in Fig. 1. Given a video segment, ARIA first estimates whether the initial VLM response is sufficiently reliable by jointly considering its response and the predictions of *temporally and semantically relevant* segments. Segments with sufficient evidence retain the initial response without further VLM inference. For segments that warrant additional examination, ARIA determines whether the uncertainty is better addressed by acquiring more visual evidence or by introducing a complementary model perspective. Accordingly, LOOK-MORE performs denser frame sampling and re-examines the segment with the base VLM, while COOPERATE obtains a complementary textual interpretation from another VLM. By allocating additional computation only to segments that warrant further examination and choosing between these two complementary sources of evidence, ARIA provides a flexible mechanism for refining VAD predictions and explanations while avoiding unnecessary inference.

Contributions. The key contributions of this paper are summarized as follows:

- We revisit VLM-based VAD through **adaptive computation**, recognizing that different video segments may benefit from different levels of examination.
- We propose ARIA, an **adaptive reasoning** framework that selectively acquires additional visual evidence via LOOK-MORE or complementary textual evidence via COOPERATE when needed.
- Extensive experiments on UCF-Crime and XD-Violence show that ARIA improves detection and explanation quality while reducing inference cost.

2 RELATED WORK

Video Anomaly Detection. Detecting video anomalies is inherently challenging due to the ambiguous and context-dependent nature of anomalous events. Existing VAD approaches include weakly supervised classification (Sultani et al., 2018; Wu et al., 2024b) and normal-pattern modeling through reconstruction or prediction (Liu et al., 2018; Ye et al., 2019; Lu et al., 2013). However, these methods primarily produce numerical predictions and largely overlook explainability. Recent work incorporates VLMs to support detection and explanation through prompt engineering (Zanella et al., 2024; Ye et al., 2025) or instruction tuning (Zhang et al., 2024; Lv & Sun, 2024; Yang et al., 2024a; Tang et al., 2024). Our work focuses on allocating follow-up VLM computation across video segments with heterogeneous evidence needs.

Adaptive Reasoning and Model Routing. Foundation models can benefit from structured prompting (Ning et al., 2024), self-reflection (Liu et al., 2024b; Wan et al., 2025), and self-consistency (Wang et al., 2023). To adapt inference-time computation to input difficulty, IBPO (Yu et al., 2025) learns to allocate reasoning computation under budget constraints, while AdaptThink (Zhang et al., 2025b) selects between thinking and direct answer modes. For model routing, RouteLLM (Ong et al., 2025) selects between stronger and weaker LLMs using preference data, while RADAR (Fernandez et al., 2026) jointly selects models and reasoning budgets based on query difficulty and model ability. These allocation and routing methods focus on language tasks rather than segment-level evidence acquisition for VAD. ARIA uses lightweight responses and temporal and semantic context to predict whether further examination is beneficial and, if so, to select denser visual evidence or a complementary VLM interpretation, while keeping all VLM parameters frozen.

3 THE ARIA FRAMEWORK

3.1 PROBLEM STATEMENT: VAD BY VLM-BASED NATURAL LANGUAGE GENERATION

Given a video $V = \{I_i\}_{i=1}^{F}$ with F frames, I_i denotes the i-th frame. Its frame-level ground truth is denoted by $Y = [y_1, \ldots, y_F]$, where $y_i = 1$ if I_i contains an anomaly and $y_i = 0$ otherwise. The goal is to obtain a frame-level anomaly score sequence $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_F]$, where $\hat{y}_i \in [0, 1]$, together with natural-language explanations for the detected events. To realize this, as exemplified by previous works, f follows a coarse-to-fine procedure: First, a given video V is divided into h segments $\{v_j\}_{j=1}^{h}$. Each segment v_j is represented by a fixed number of uniformly sampled frames, such that a similar visual inference budget is typically applied across segments. For each segment v_j, a VAD-specific instruction θ prompts a base VLM f to generate a binary prediction $\tilde{y}_j \in \{0, 1\}$ and a natural-language explanation E_j, where $\tilde{y}_j = 1$ denotes an anomalous segment and $\tilde{y}_j = 0$ otherwise. Formally, the initial segment-level prediction and explanation are given by

$$(\tilde{y}_j, E_j) = f(v_j; \theta). \tag{1}$$

The segment-level predictions are then aggregated and temporally smoothed to obtain continuous frame-level anomaly scores $\hat{Y}$ (Yang et al., 2024a; Ye et al., 2019). However, applying a similar inference budget to all segments overlooks substantial differences in their reasoning difficulty. Many segments can be handled reliably with sparse visual evidence, whereas ambiguous ones may benefit from further examination using either additional visual evidence or a complementary interpretation. This motivates adaptive computation that determines whether a segment warrants further examination and, if so, which form of additional evidence to acquire.

3.2 OVERVIEW OF ARIA

ARIA adaptively allocates VLM inference across video segments while keeping all VLM parameters frozen. As illustrated in Fig. 2, ARIA starts from a lightweight base inference and selectively allocates additional computation when further examination is predicted to be beneficial. Specifically, given a segment v_j, Step 1 constructs a routing representation from the lightweight base response together with temporal and semantic contextual cues. Step 2 then uses this representation to predict whether any available follow-up action is likely to improve the lightweight prediction. Step 2 routes each segment to STOP, which retains the base response, or selects between LOOK-MORE and COOPERATE to acquire denser visual evidence or a complementary textual interpretation, respectively, when further examination is predicted to be beneficial. The resulting action is STOP, which retains the lightweight response; LOOK-MORE, which re-examines the segment with denser frame sampling using the base VLM; or COOPERATE, which obtains a complementary textual interpretation from another VLM. Only the selected action is executed at inference time, avoiding VLM calls to unselected alternatives. The resulting segment-level predictions are then converted into frame-level anomaly scores through semantic aggregation and temporal smoothing. The complete process is described as pseudocode in Appendix Sec. A.1.1.

3.3 STEP 1: CHARACTERIZING LIGHTWEIGHT RESPONSES FOR ADAPTIVE ROUTING

Intuition. Given the lightweight base response from Eq. (1), ARIA constructs low-cost routing features to support the subsequent allocation of additional computation. The base response alone

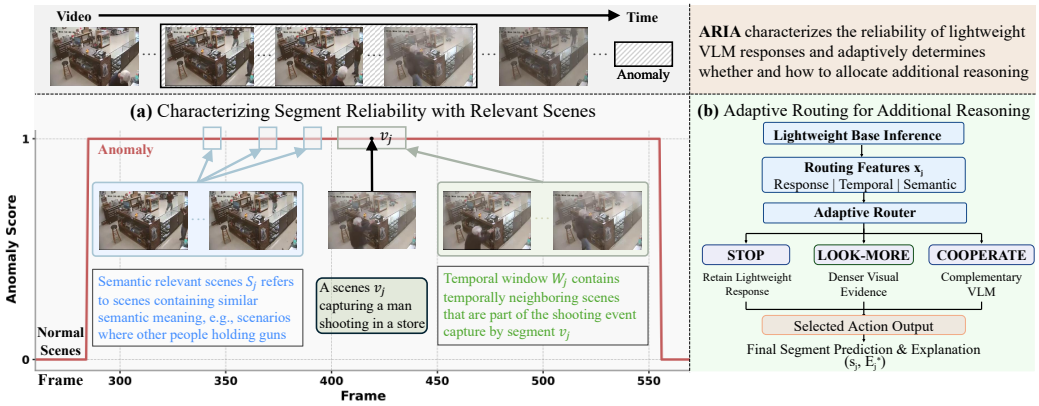

Figure 2: Overview of ARIA. (a) Lightweight responses and video context provide routing features. (b) The router retains the base response or selects LOOK-MORE for denser visual evidence or COOPERATE for a complementary textual interpretation.

may provide limited information about the potential benefit of further examination. Video data offer additional context through temporal continuity and semantic redundancy. Temporally neighboring segments may capture different stages of the same event, while visually similar content may recur at distant time steps. The lightweight predictions of these related segments can therefore be reused as contextual cues for interpreting the target response, without additional generative VLM calls. Such contextual agreement or disagreement alone, however, does not establish whether further examination would improve the target prediction. ARIA therefore combines response-level cues with temporal and semantic context into a routing representation x_j. The router described in Sec. 3.4 uses x_j to predict whether further examination would be beneficial and, if so, which form of additional evidence to acquire.

Implementation. For the lightweight base inference, we instantiate the segment-level prediction process in Eq. (1) in Sec. 3.1 using κ_b sparsely sampled frames per segment. We denote the resulting base prediction and explanation for segment v_j by $\tilde{y}_j^{(b)}$ and $E_j^{(b)}$, respectively. To characterize the lightweight response, we use an auxiliary MLP classifier that combines TF-IDF features of $E_j^{(b)}$ with numerical features to estimate whether a complementary VLM would be helpful for the segment. Its output $\mathbf{r}_j$ serves as a learned auxiliary helpfulness score in the routing representation. Computing this feature requires no additional generative VLM inference. We complement this score with explicit temporal and semantic features derived from the base predictions of related segments as follows:

1. Temporal Context. Since events often unfold over consecutive segments, we use neighboring base predictions to provide local temporal context for the target response. For each target segment v_j, we consider a temporal neighborhood $\mathcal{W}_j = \{v_i : 1 \le i \le h,\ |i-j| \le (w-1)/2\}$, where w is the window size. We arrange the base predictions $\tilde{y}_i^{(b)}$ within $\mathcal{W}_j$ in temporal order and concatenate this sequence with the normalized segment position j/h to form the temporal feature vector $\mathbf{t}_j$.

2. Semantic Context. To capture relevant context beyond the local temporal neighborhood, we retrieve visually similar segments from the same video. To identify such segments, we use a frozen vision encoder g to obtain a visual representation $e_j = g(v_j)$ for each segment. For each target segment v_j, we retrieve the top-K segments according to cosine similarity $e_i^\top e_j/(\|e_i\|_2 \cdot \|e_j\|_2)$, which is denoted as S_j. To characterize agreement among their base predictions, we form a binary prediction distribution, where $p_j(c)$ is the proportion of retrieved segments predicted as class $c \in \{0,1\}$. We then summarize disagreement in this distribution using its entropy:

$$H_j^{(\mathrm{sem})} = -\sum_{c\in\{0,1\}} p_j(c)\log p_j(c), \tag{2}$$

Lower entropy indicates greater agreement among the retrieved predictions. We retain $H_j^{(\mathrm{sem})}$ as a continuous routing feature.

Routing Representation. In addition to the response- and context-based cues above, we include a coarse category-level feature $\mathbf{c}_j$ obtained from a frozen zero-shot visual classifier as an auxiliary

contextual prior. Concatenating these features yields the routing representation:

$$x_j = [\mathbf{r}_j \| \mathbf{t}_j \| H_j^{(\text{sem})} \| \mathbf{c}_j], \tag{3}$$

where $\|$ denotes concatenation. At inference time, all components of x_j are computed before either follow-up action is executed, without accessing outputs from LOOK-MORE or COOPERATE. Detailed feature definitions and extraction settings are provided in Appendix Sec. A.1.2.

3.4 STEP 2: ADAPTIVE ROUTING FOR FURTHER EXAMINATION

Intuition. Further examination can provide different forms of additional evidence, depending on what is missing from the lightweight response. One possibility is that the lightweight response is limited by sparse visual observations, so informative visual cues may have been missed. In this case, re-examining the segment with denser frame sampling using the same base VLM provides additional visual evidence that may correct the initial prediction. In other cases, the available visual evidence may not be the main limitation, and a complementary interpretation from another VLM can provide information not captured by the base response. Therefore, applying the same follow-up strategy to every segment selected for further examination may introduce unnecessary computation and fail to match its evidence needs. Accordingly, ARIA first determines whether further examination is likely to be beneficial and, if so, selects the follow-up action with the greater predicted corrective utility.

Implementation. To learn the two routing decisions, we construct supervision from the outcomes of the available follow-up actions during offline training. For each training segment v_j, we cache the predictions produced by the lightweight base inference, LOOK-MORE, and COOPERATE, denoted by $\tilde{y}_j^{(b)}$, $\tilde{y}_j^{(m)}$, and $\tilde{y}_j^{(\text{co})}$, respectively. These counterfactual outputs are used only to construct routing targets during training and are never accessed by the router at inference time. Let $y_j^{\text{seg}} \in \{0, 1\}$ denote the segment-level ground-truth label derived from the frame-level annotations. A follow-up action is considered beneficial only when it corrects an error made by the lightweight prediction with respect to y_j^{seg}. Accordingly, $z_j^{\text{lm}} = 1$ when LOOK-MORE changes an incorrect lightweight prediction into a correct one, and $z_j^{\text{co}} = 1$ when COOPERATE does so; otherwise, the corresponding target is 0. Since the first routing decision only needs to determine whether any available follow-up can improve the lightweight prediction, we define the stage-1 target as $z_j^{\text{s1}} = z_j^{\text{lm}} \vee z_j^{\text{co}}$, which is positive whenever at least one follow-up action can correct the lightweight prediction. Detailed construction of the segment-level labels and routing targets, together with the router training procedure, is provided in Appendix Sec. A.1.3.

Routing Heads and Training. Given the routing representation x_j constructed in Step 1, the stage-1 head estimates $p_{\text{s1}}(x_j)$, the probability that at least one available follow-up action can correct the lightweight prediction. Two action-specific heads estimate $p_{\text{lm}}(x_j)$ and $p_{\text{co}}(x_j)$, the probabilities that LOOK-MORE and COOPERATE, respectively, can correct the lightweight prediction. These events are not mutually exclusive, since both actions may correct the same base error. We therefore model them with separate binary classifiers rather than mutually exclusive training classes.

All three heads are implemented as gradient-boosted tree (GBT) classifiers with the shared input x_j and are trained using their respective binary targets defined above. The auxiliary MLP that produces $\mathbf{r}_j$ is trained using teammate-helpfulness labels. Both the MLP and the GBT heads are trained offline, while all VLM parameters remain frozen. Training and evaluation splits are constructed at the video level, keeping all segments from each video in the same split. Detailed supervision, feature-generation procedures, and evaluation configurations are provided in Appendix Sec. A.1.3.

Inference Policy. At inference time, ARIA first evaluates the stage-1 score $p_{\text{s1}}(x_j)$. If $p_{\text{s1}}(x_j) < \tau$ (τ is a preset threshold), the lightweight response is retained, and no follow-up VLM inference is performed. Otherwise, the segment is routed to further examination, and ARIA compares the two action-specific scores $p_{\text{lm}}(x_j)$ and $p_{\text{co}}(x_j)$ to select the follow-up action with the greater predicted corrective utility:

$$a(v_j) = \begin{cases} \text{STOP}, & p_{\text{s1}}(x_j) < \tau, \\ \text{LOOK-MORE}, & p_{\text{s1}}(x_j) \geq \tau \text{ and } p_{\text{lm}}(x_j) > p_{\text{co}}(x_j), \\ \text{COOPERATE}, & \text{otherwise.} \end{cases} \tag{4}$$

This policy separates the decision of whether additional computation is useful from the decision of which form of additional evidence to acquire. As a result, segments that are sufficiently handled by the lightweight inference incur no additional VLM cost, while only the selected follow-up action is executed for the remaining segments. The threshold τ controls the performance–cost trade-off: a smaller τ routes more segments to further examination, increasing follow-up computation, whereas a larger τ retains more segments after the lightweight base inference. We analyze this trade-off in Sec. 4.4 and Appendix Sec. A.4.2.

3.5 Action Execution and Aggregation

Once the routing action is determined, ARIA executes only the selected inference path. For STOP, the lightweight prediction and explanation are retained without additional VLM inference. For LOOK-MORE, the base VLM re-examines the target segment using κ_m sampled frames, where $\kappa_m > \kappa_b$, thereby acquiring denser visual evidence and yielding an updated prediction $\tilde{y}_j^{(m)}$ and explanation $E_j^{(m)}$. For COOPERATE, another VLM f_{co} provides a complementary textual interpretation of the target segment. Conditioned on the target segment and the lightweight base response, f_{co} produces an updated prediction $\tilde{y}_j^{(\text{co})}$ and explanation $E_j^{(\text{co})}$ under the cooperation prompt ξ. COOPERATE does not access any output from LOOK-MORE; the two follow-up paths are executed independently after routing. The prediction and explanation produced by the selected path are used as the final segment-level output (s_j, E_j^*), while unselected follow-up actions are never executed at inference time. For efficient execution, independent segment queries assigned to the same inference path are processed in batches of size B without sharing context across segments. Batching details and efficiency analysis are provided in Appendix Sec. A.3.5.

After obtaining the routed segment-level sequence $\{s_j\}_{j=1}^{h}$, we apply semantic aggregation as a post-processing step before converting segment-level predictions into frame-level scores as (Ye et al., 2025) does. For each target segment v_j, we retrieve its top-K semantically related segments in the visual feature space of g and combine their routed scores using normalized cosine-similarity weights to obtain an aggregated segment-level score. The resulting scores are then temporally smoothed with a Gaussian kernel and mapped to frame-level anomaly scores $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_F]$. The aggregation and smoothing configurations are detailed in Appendix Secs. A.1.2 and A.3.2.

4 Experiments and Results

We evaluate ARIA by addressing the following questions: **(Q1)** Does ARIA improve both detection performance and explanation quality? **(Q2)** Can ARIA effectively determine when further examination is beneficial and which form of additional evidence to acquire? **(Q3)** Does adaptive computation reduce unnecessary VLM inference while preserving detection performance? **(Q4)** Do these gains generalize across different VLM and prompt configurations?

4.1 Experimental Settings

Datasets. Following pioneering VLM-based VAD studies (Zanella et al., 2024; Ye et al., 2025), we evaluate ARIA on two large-scale VAD datasets, UCF-Crime (Sultani et al., 2018) and XD-Violence (Wu et al., 2020). Dataset details are provided in Appendix Sec. A.3.1.

Metrics. Following previous works, we report frame-level AUC on both datasets and AP on XD-Violence for detection performance. For explanation quality, we employ DeepSeek-R1-70B as an LLM judge and evaluate each response along three dimensions: *Reasonability*, *Novel Evidence*, and *Conciseness*. For inference efficiency, we report token consumption and wall-clock inference time. Detailed LLM-as-Judge evaluation protocols and prompts are provided in Appendix Sec. A.3.4, and additional validation of the LLM-as-Judge protocol is provided in Appendix Sec. A.5.4.

Implementation of ARIA. We use open-source VLMs from the InternVL (Chen et al., 2024b) and Qwen-VL (Wang et al., 2024) families. By default, InternVL3-8B serves as the base VLM f, while InternVL2-8B serves as the complementary VLM for COOPERATE. The lightweight base inference uses $\kappa_b = 4$ sampled frames per segment, while both LOOK-MORE and COOPERATE use 8-frame visual inputs, i.e., $\kappa_m = 8$. All VLM parameters remain frozen, and we adopt the VAD instruction

Table 1: Performance comparison of VAD measured by (a) AUC on UCF-Crime, and (b) AUC and (c) AP on XD-Violence. † in (a) indicates that the results are obtained with fine-tuning. ‡ in (c) indicates methods that use full training data to train the model.

(a) UCF-Crime (AUC)

Method	AUC (%)
Non-Explainable VAD Methods	
Anomize (Li et al., 2025)	84.49
Chen et al. (Chen et al., 2024a)	86.83
CLAWS (Zaheer et al., 2020)	83.03
CLAV (Cho et al., 2023)	86.10
CLIP-TSA (Joo et al., 2023)	87.58
DANCE-SA (Purwanto et al., 2021)	85.00
DYANNET (Thakare et al., 2023b)	84.50
GCL (Zaheer et al., 2022)	79.84
GCN (Zhong et al., 2019)	82.12
GODS (Wang & Cherian, 2019)	70.46
GS-MoE (Amicantonio et al., 2025)	**91.58**
MGFN (Chen et al., 2023)	86.98
MIST (Feng et al., 2021)	82.30
MSL (Li et al., 2022b)	85.62
OVVAD (Wu et al., 2024a)	86.40
π-VAD (Majhi et al., 2025)	90.33
RTFM (Tian et al., 2021)	84.30
S3R (Wu et al., 2022)	85.99
SSRL (Li et al., 2022a)	87.43
Sultani et al. (Sultani et al., 2018)	77.92
TPWNG (Yang et al., 2024b)	87.79
Tur et al. (Tur et al., 2023)	66.85
VadCLIP (Wu et al., 2024b)	88.02
Wu et al. (Wu et al., 2020)	82.44
WSAL (Lv et al., 2021)	85.38
Explainable VAD Methods	
AnyAnomaly (Ahn et al., 2025)	80.70
EventVAD (Shao et al., 2025)	82.03
Flashback (Anonymous, 2025)	87.29
HiProbe-VAD (Cai et al., 2025)	86.72 (88.91†)
Holmes-VAD (Zhang et al., 2024)	84.61 (89.51†)
Holmes-VAU (Zhang et al., 2025a)	88.96†
LAVAD (Zanella et al., 2024)	80.28
LLAVA-1.5 (Liu et al., 2024a)	72.84
SUVAD (Gao et al., 2025)	83.90
VADor (Lv & Sun, 2024)	85.90 (88.13†)
VERA (Ye et al., 2025)	86.55
ZS CLIP (Zanella et al., 2024)	53.16
ZS IMAGEBIND-I (Girdhar et al., 2023)	53.65
ZS IMAGEBIND-V (Girdhar et al., 2023)	55.78
ARIA	**88.32**

(b) XD-Violence (AUC)

Method	AUC (%)
Non-Explainable VAD Methods	
BODS (Wang & Cherian, 2019)	57.32
GODS (Wang & Cherian, 2019)	61.56
Hasan et al. (Hasan et al., 2016)	50.32
Lu et al. (Lu et al., 2013)	53.56
RareAnom (Thakare et al., 2023a)	68.33
Explainable VAD Methods	
EventVAD (Shao et al., 2025)	87.51
LAVAD (Zanella et al., 2024)	85.36
LLAVA-1.5 (Liu et al., 2024a)	79.62
VERA (Ye et al., 2025)	88.26
ZS CLIP (Zanella et al., 2024)	38.21
ZS IMAGEBIND-I (Girdhar et al., 2023)	58.81
ZS IMAGEBIND-V (Girdhar et al., 2023)	55.06
ARIA	**91.17**

(c) XD-Violence (AP)

Method	AP (%)
Non-Explainable VAD Methods	
Anomize (Li et al., 2025)	69.31
GS-MoE‡ (Amicantonio et al., 2025)	82.89
Wu et al.‡ (Wu et al., 2020)	78.64
OVVAD (Wu et al., 2024a)	66.53
π-VAD‡ (Majhi et al., 2025)	**85.37**
RTFM‡ (Tian et al., 2021)	77.81
MSL‡ (Li et al., 2022b)	78.58
MGFN‡ (Chen et al., 2023)	80.11
CLIP-TSA (Joo et al., 2023)	82.19
Explainable VAD Methods	
EventVAD (Shao et al., 2025)	64.04
Flashback (Anonymous, 2025)	75.13
HiProbe-VAD‡ (Cai et al., 2025)	89.51
Holmes-VAD‡ (Zhang et al., 2024)	**90.67**
Holmes-VAU‡ (Zhang et al., 2025a)	87.68
LAVAD (Zanella et al., 2024)	62.01
LLAVA-1.5 (Liu et al., 2024a)	50.26
SUVAD (Gao et al., 2025)	70.10
VERA (Ye et al., 2025)	70.54
ZS CLIP (Zanella et al., 2024)	17.83
ZS IMAGEBIND-I (Girdhar et al., 2023)	27.25
ZS IMAGEBIND-V (Girdhar et al., 2023)	25.36
ARIA	77.31

θ from VERA (Ye et al., 2025). The lightweight router is trained offline as described in Sec. 3.4. Detailed hyperparameter settings and batched inference configurations are provided in Appendix Secs. A.3.2 and A.3.5, respectively.

Baselines. Following prior VAD literature, we compare ARIA with both non-explainable VAD methods and explainable VLM-based approaches. The compared methods are summarized in Table 1.

4.2 Comparison with State-of-the-Art Methods

Detection Performance. As shown in Table 1(a), ARIA achieves 88.32% AUC on UCF-Crime, outperforming all compared explainable VAD methods without instruction tuning and improving over VERA by 1.77 percentage points. On XD-Violence, ARIA achieves 91.17% AUC and 77.31% AP, improving over VERA by 2.91 and 6.77 percentage points, respectively.

Explanation Quality. As shown in Table 2, ARIA outperforms both the initial VERA response and Self-Reflection implemented with SRPO (Wan et al., 2025) across all three evaluation dimensions, with the largest gains in *Novel Evidence*: 2.79 points over the initial response and 2.25 points over Self-Reflection. UCF-Crime results are reported here, with XD-Violence results provided in Appendix Table 18.

Table 2: Explanation-quality evaluation via LLM-as-Judge (DeepSeek-R1-70B) across 14 anomaly categories in UCF-Crime. Each individual criterion is scored on a 1–10 scale.

Metric	**Initial** (VERA)	**Self-Reflection** (SRPO)	**ARIA**
Reasonability	7.16	7.49	**8.02**
Novel Evidence	4.13	4.67	**6.92**
Conciseness	6.24	6.32	**7.03**
Total	17.52	18.48	**21.97**
Win Percentage	1.4%	3.4%	**90.2%**

Table 3: Ablation of adaptive reasoning components on UCF-Crime. Δ denotes the AUC improvement over the lightweight 4-frame base inference.

Configuration	**AUC (%)**	Δ
Lightweight Base (4 frames) only	83.73	–
Adaptive + LOOK-MORE only	86.63	+2.90
Adaptive + COOPERATE only	87.28	+3.55
Full Adaptive Reasoning	**88.32**	**+4.59**

To further assess the reliability of the LLM-based evaluation, we use a position-debiased best-pick protocol across different judge configurations and conduct a blind human evaluation. ARIA achieves preference rates of 67.0%–71.4% across judge configurations, with a method ordering consistent with human preferences; full results are provided in Appendix Sec. A.5.4.

4.3 ANALYSIS OF ADAPTIVE REASONING

We examine two key aspects of the adaptive computation design: (i) whether the two follow-up actions provide useful forms of additional evidence, and (ii) whether the router can determine when further examination is beneficial and which follow-up action has greater corrective utility.

Adaptive Reasoning Components. Table 3 compares adaptive variants with one or both follow-up actions available. The LOOK-MORE-only and COOPERATE-only variants achieve 86.63% and 87.28% AUC, respectively. Allowing the router to select between both actions yields 88.32% AUC, improving over the stronger single-action variant by 1.04 percentage points. Comparisons with entropy-based segment selection are provided in Appendix Table 6, and voting- and debate-based alternatives for COOPERATE are evaluated in Appendix Sec. A.2.4.

Routing Feature Design. With adaptive routing retained, progressively adding response-level features and temporal and semantic context to x_j improves routing-head AUC and downstream VAD AUC over using only the lightweight base prediction as routing input (Table 4). Adding category-level context further increases downstream AUC to 88.32%, despite small decreases in the stage-1 and COOPERATE head AUCs. Additional diagnostic analyses are provided in Appendix Sec. A.4.3.

4.4 PERFORMANCE–COST ANALYSIS

A key motivation of adaptive computation is to avoid unnecessary follow-up VLM inference. We therefore compare ARIA with VERA, a fixed 8-frame segment-by-segment inference framework with fixed collaborative reasoning, to assess its detection performance and inference cost, measured by token consumption and wall-clock time.

Compared with fixed collaborative reasoning, ARIA reduces average token consumption per video from 1,249,200 to 716,512 (a 42.6% reduction). Visual-token consumption decreases from 983,040 to 524,288 (a 46.7% reduction). Under the same hardware environment, ARIA also achieves lower end-to-end inference time, requiring 5.9 minutes per video compared with 17.1 minutes for fixed 8-frame segment-by-segment inference and 25.7 minutes for fixed collaborative reasoning (Table 5). Detailed batching analysis is provided in Appendix Sec. A.3.5.

We further vary the routing threshold τ to examine the trade-off between detection performance and inference cost. Detection performance remains relatively stable for $\tau \in [0.10, 0.20]$, where lower

Table 4: Cumulative ablation of routing features on UCF-Crime. p_{s1} estimates whether at least one follow-up action can correct the lightweight prediction, while p_{lm} and p_{co} estimate the action-specific corrective utility of LOOK-MORE and COOPERATE, respectively. The last column reports downstream VAD AUC.

Routing Features	p_{s1} AUC	p_{lm} AUC	p_{co} AUC	Downstream AUC (%)
Lightweight base prediction	0.8242	0.8020	0.8450	84.43
+ Response features	0.8774	0.8517	0.8878	86.74
+ Temporal context	0.9054	0.8788	0.9490	87.21
+ Semantic-neighbor entropy	**0.9240**	0.8952	**0.9580**	87.57
+ Category-level context	0.9208	**0.8964**	0.9566	**88.32**

Table 5: End-to-end inference time under the same hardware environment. Relative cost is normalized to fixed 8-frame segment-by-segment inference.

Inference Strategy	Time / Video (min)	Relative Cost
Fixed 8-Frame Segment-by-Segment Inference	17.1	100%
Fixed Collaborative Reasoning	25.7	150.3%
ARIA (Adaptive Computation)	**5.9**	**34.5%**

thresholds incur higher inference cost with limited additional performance gains. We use $\tau = 0.20$ in the main experiments to limit follow-up computation within this stable performance range. The complete threshold sweep is provided in Appendix Sec. A.4.2.

4.5 GENERALIZATION

On UCF-Crime, we evaluate AUC gains over the corresponding base inference across complementary VLMs, base VLM architectures, and prompt variants. The gains range from 0.13 to 1.24 percentage points across complementary VLMs, from 1.24 to 4.83 points across base VLM architectures, and from 1.24 to 3.16 points across prompt variants. Full results are provided in Appendix Sec. A.4.6.

4.6 QUALITATIVE RESULTS AND CASE STUDIES

Fig. 3 illustrates how ARIA adapts to different evidence needs. In Fig. 3 left, LOOK-MORE corrects a false positive by acquiring denser visual evidence. In Fig. 3 right, COOPERATE recovers a missed anomaly through a complementary VLM interpretation. Additional examples are provided in Appendix Sec. A.5.2.

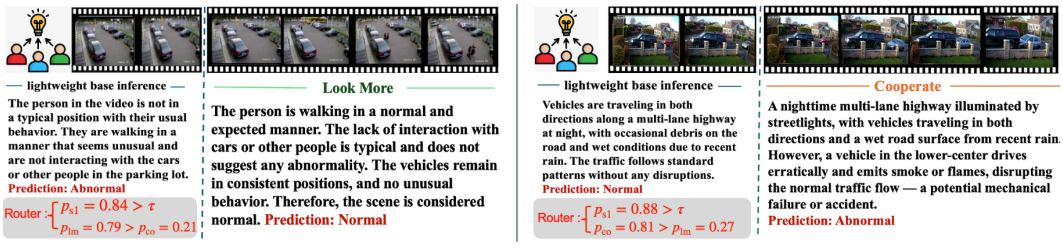

Figure 3: Representative cases showing how ARIA adaptively selects LOOK-MORE or COOPERATE to obtain the additional evidence needed for correcting base predictions.

5 CONCLUDING REMARKS

We presented ARIA, an adaptive reasoning framework for VLM-based VAD that determines whether a video segment warrants further examination and selects between denser visual evidence and a complementary VLM interpretation. Experiments on UCF-Crime and XD-Violence show that ARIA improves detection and explanation quality while reducing unnecessary inference cost. Additional evaluations demonstrate consistent gains across the tested VLM and prompt configurations.

AI USE STATEMENT

In preparing this work, we made limited use of large language models as auxiliary tools, specifically for sentence rewriting, grammar checking, and stylistic refinement to improve the clarity and readability of the manuscript. We reviewed all AI-assisted work. All core research contributions, including the experimental methodology and design, analysis of experimental results, mathematical derivations, and conclusions, were developed by the authors. The authors take full responsibility for the content of this work.

ETHICS STATEMENT

This work studies video anomaly detection using existing benchmark datasets. Because video anomaly detection may be deployed in surveillance and other safety-critical settings, practical applications should carefully consider privacy, data protection, and the potential consequences of false positive and false negative predictions. The proposed system is intended to support anomaly detection and reasoning rather than serve as the sole basis for high-stakes automated decisions.

REPRODUCIBILITY STATEMENT

To facilitate reproduction of our results, the supplementary material includes the code for the complete inference pipeline and adaptive router, as well as the datasets and test splits used in our experiments. Detailed model configurations, routing and training procedures, hyperparameter settings, data processing, and evaluation protocols are provided in the main paper and Appendix. Together, these materials provide the code, data, and experimental specifications required to reproduce the reported results.

REFERENCES

Sunghyun Ahn, Youngwan Jo, Kijung Lee, Sein Kwon, Inpyo Hong, and Sanghyun Park. Anyanomaly: Zero-shot customizable video anomaly detection with lvlm. *arXiv preprint arXiv:2503.04504*, 2025.

Giacomo D' Amicantonio, Snehashis Majhi, Quan Kong, Lorenzo Garattoni, Gianpiero Francesca, François Brémond, and Egor Bondarev. Mixture of experts guided by gaussian splatters matters: A new approach to weakly-supervised video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10275–10285, 2025.

Anonymous. Flashback: Memory-driven zero-shot, real-time video anomaly detection. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=BQwGPv4DTe`. under review.

Zhaolin Cai, Fan Li, Ziwei Zheng, and Yanjun Qin. Hiprobe-vad: Video anomaly detection via hidden states probing in tuning-free multimodal llms. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 592–601, 2025.

Junxi Chen, Liang Li, Li Su, Zheng-jun Zha, and Qingming Huang. Prompt-enhanced multiple instance learning for weakly supervised video anomaly detection. In *CVPR*, 2024a.

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. In *AAAI*, 2023.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024b.

MyeongAh Cho, Minjung Kim, Sangwon Hwang, Chaewon Park, Kyungjae Lee, and Sangyoun Lee. Look around for anomalies: Weakly-supervised anomaly detection via context-motion relational learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12137–12146, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Jia-Chang Feng, Fa-Ting Hong, and Wei-Shi Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *CVPR*, 2021.

Nigel Steven Fernandez, Branislav Kveton, Ryan Rossi, Andrew Lan, and Jack Wang. Radar: Reasoning-ability and difficulty-aware routing for reasoning llms. In *International Conference on Learning Representations*, volume 2026, pp. 109429–109457, 2026.

Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 54(7): 1–37, 2021.

Shibo Gao, Peipei Yang, and Linlin Huang. Suvad: Semantic understanding based video anomaly detection using mllm. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S. Davis. Learning temporal regularity in video sequences. In *CVPR*, 2016.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei W Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in llms. *Advances in Neural Information Processing Systems*, 37:24181–24215, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Hyekang Kevin Joo, Khoa Vo, Kashu Yamazaki, and Ngan Le. Clip-tsa: Clip-assisted temporal self-attention for weakly-supervised video anomaly detection. In *ICIP*, 2023.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 32. Association for Computational Linguistics, 05 2025.

Fei Li, Wenxuan Liu, Jingjing Chen, Ruixu Zhang, Yuran Wang, Xian Zhong, and Zheng Wang. Anomize: Better open vocabulary video anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29203–29212, 2025.

Guoqiu Li, Guanxiong Cai, Xingyu Zeng, and Rui Zhao. Scale-aware spatio-temporal relation learning for video anomaly detection. In *ECCV*, 2022a.

Shuo Li, Fang Liu, and Licheng Jiao. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In *AAAI*, 2022b.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, 2024.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DWkJCSxKU5.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024a.

Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. *Advances in Neural Information Processing Systems*, 37:97800–97825, 2024b.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013.

Hui Lv and Qianru Sun. Video anomaly detection and explanation via large language models. *arXiv preprint arXiv:2401.05702*, 2024.

Hui Lv, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Localizing anomalies from weakly-labeled videos. *IEEE transactions on image processing*, 30:4505–4515, 2021.

Snehashis Majhi, Giacomo D'Amicantonio, Antitza Dantcheva, Quan Kong, Lorenzo Garattoni, Gianpiero Francesca, Egor Bondarev, and François Brémond. Just dance with pi! a poly-modal inductor for weakly-supervised video anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24265–24274, 2025.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting llms for efficient parallel generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mqVgBbNCm9.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Didik Purwanto, Yie-Tarng Chen, and Wen-Hsien Fang. Dance with self-attention: A new look of conditional random fields on anomaly detection in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 173–183, 2021.

Bharathkumar Ramachandra, Michael J Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(5): 2293–2312, 2020.

Yihua Shao, Haojin He, Sijie Li, Siyu Chen, Xinwei Long, Fanhu Zeng, Yuxuan Fan, Muyang Zhang, Ziyang Yan, Ao Ma, et al. Eventvad: Training-free event-aware video anomaly detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2586–2595, 2025.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018.

Jiaqi Tang, Hao Lu, Ruizheng Wu, Xiaogang Xu, Ke Ma, Cheng Fang, Bin Guo, Jiangbo Lu, Qifeng Chen, and Ying-Cong Chen. Hawk: Learning to understand open-world video anomalies. In *NeurIPS*, 2024.

Kamalakar Vijay Thakare, Debi Prosad Dogra, Heeseung Choi, Haksub Kim, and Ig-Jae Kim. Rareanom: A benchmark video dataset for rare type anomalies. *Pattern Recognition*, 140:109567, 2023a.

Kamalakar Vijay Thakare, Yash Raghuwanshi, Debi Prosad Dogra, Heeseung Choi, and Ig-Jae Kim. Dyannet: A scene dynamicity guided self-trained video anomaly detection network. In *WACV*, 2023b.

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 *ICCV*, 2021.

Anil Osman Tur, Nicola Dall'Asen, Cigdem Beyan, and Elisa Ricci. Unsupervised video anomaly detection with diffusion models conditioned on compact motion representations. In *International Conference on Image Analysis and Processing*, 2023.

Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, et al. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement learning. *arXiv preprint arXiv:2506.01713*, 2025.

Jue Wang and Anoop Cherian. Gods: Generalized one-class discriminative subspaces for anomaly detection. In *ICCV*, 2019.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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 *ECCV*, 2022.

Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *ECCV*, 2020.

Peng Wu, Xuerong Zhou, Guansong Pang, Yujia Sun, Jing Liu, Peng Wang, and Yanning Zhang. Open-vocabulary video anomaly detection. In *CVPR*, pp. 18297–18307, 2024a.

Peng Wu, Xuerong Zhou, Guansong Pang, Lingru Zhou, Qingsen Yan, Peng Wang, and Yanning Zhang. Vadclip: Adapting vision-language models for weakly supervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6074–6082, 2024b.

Yuchen Yang, Kwonjoon Lee, Behzad Dariush, Yinzhi Cao, and Shao-Yuan Lo. Follow the rules: Reasoning for video anomaly detection with large language models. In *ECCV*, 2024a.

Zhiwei Yang, Jing Liu, and Peng Wu. Text prompt with normality guidance for weakly supervised video anomaly detection. In *CVPR*, 2024b.

Yu Yao, Xizi Wang, Mingze Xu, Zelin Pu, Yuchen Wang, Ella Atkins, and David J Crandall. Dota: Unsupervised detection of traffic anomaly in driving videos. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):444–459, 2022.

Muchao Ye, Xiaojiang Peng, Weihao Gan, Wei Wu, and Yu Qiao. Anopcn: Video anomaly detection via deep predictive coding network. In *ACM international conference on multimedia*, 2019.

Muchao Ye, Weiyang Liu, and Pan He. Vera: Explainable video anomaly detection via verbalized learning of vision-language models. In *CVPR*, 2025.

Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, et al. Think smarter not harder: Adaptive reasoning with inference aware optimization. *arXiv preprint arXiv:2501.17974*, 2025.

M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *CVPR*, 2022.

Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *ECCV*, 2020.

Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. Harnessing large language models for training-free video anomaly detection. In *CVPR*, 2024.

Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Chuchu Han, Xiaonan Huang, Changxin Gao, Yuehuan Wang, and Nong Sang. Holmes-vad: Towards unbiased and explainable video anomaly detection via multi-modal llm. *arXiv preprint arXiv:2406.12235*, 2024.

Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Xiaonan Huang, Changxin Gao, Shanjun Zhang, Li Yu, and Nong Sang. Holmes-vau: Towards long-term video anomaly understanding at any granularity. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13843–13853, 2025a.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025b.

Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, 2019.

Yuqi Zhu, Ge Li, Xue Jiang, Jia Li, Hong Mei, Zhi Jin, and Yihong Dong. Uncertainty-guided chain-of-thought for code generation with llms. *arXiv preprint arXiv:2503.15341*, 2025.

A APPENDIX

A.1 METHOD DETAILS

A.1.1 PSEUDOCODE OF ARIA

Alg. 1 summarizes the inference procedure of ARIA. Given a video divided into segments, ARIA first performs lightweight base inference for all segments. Based on the resulting responses, ARIA constructs routing features from response-level, temporal, semantic, and additional lightweight contextual signals. The router first determines whether the lightweight response is likely to benefit from further examination. If not, the lightweight response is directly retained. Otherwise, ARIA compares the action-specific scores of LOOK-MORE and COOPERATE and selects the action with the greater predicted corrective utility. LOOK-MORE acquires denser visual evidence using the same base VLM, while COOPERATE obtains a complementary textual interpretation from another VLM. Only the selected inference path is executed. Finally, the routed segment-level predictions are refined by semantic aggregation and temporal smoothing to obtain frame-level anomaly scores.

Here, $v_j^{(\kappa)}$ denotes κ frames sampled from segment v_j, where $\kappa_m > \kappa_b$.

Algorithm 1: Inference Procedure of ARIA

Input: Segments $\{v_j\}_{j=1}^{h}$; base VLM f; complementary VLM f_{co}; prompts θ and ξ; sampling sizes κ_b and κ_m; trained routing heads; routing threshold τ.

Output: Frame-level anomaly scores $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_F]$ and final segment-level explanations $\{E_j^*\}_{j=1}^{h}$.

Pass 1: Lightweight Base Inference

for $j = 1, \ldots, h$ **do**

$(\tilde{y}_j^{(b)}, E_j^{(b)}) \leftarrow f(v_j^{(\kappa_b)}; \theta)$

end for

Pass 2: Adaptive Routing for Further Examination

for $j = 1, \ldots, h$ **do**

Construct routing feature x_j from the lightweight response, temporal and semantic context, and additional lightweight contextual signals.

Compute the stage-1 probability $p_{\mathrm{s1}}(x_j)$ that further examination is beneficial.

If $p_{\mathrm{s1}}(x_j) \geq \tau$, compute $p_{\mathrm{lm}}(x_j)$ and $p_{\mathrm{co}}(x_j)$.

Select the routing action

$$a(v_j) = \begin{cases} \textsc{Stop}, & p_{\mathrm{s1}}(x_j) < \tau, \\ \textsc{Look-More}, & p_{\mathrm{s1}}(x_j) \geq \tau \text{ and } p_{\mathrm{lm}}(x_j) > p_{\mathrm{co}}(x_j), \\ \textsc{Cooperate}, & \text{otherwise.} \end{cases}$$

Execute only the selected action

$$(s_j, E_j^*) = \begin{cases} (\tilde{y}_j^{(b)}, E_j^{(b)}), & a(v_j) = \textsc{Stop}, \\ f(v_j^{(\kappa_m)}; \theta), & a(v_j) = \textsc{Look-More}, \\ f_{\mathrm{co}}\big(v_j, \tilde{y}_j^{(b)}, E_j^{(b)}; \xi\big), & a(v_j) = \textsc{Cooperate}. \end{cases}$$

end for

Segment-Level Aggregation and Frame-Level Prediction

$$\{\bar{s}_j\}_{j=1}^{h} \leftarrow \textsc{SemanticAggregate}\left(\{s_j\}_{j=1}^{h}\right).$$

$$\hat{Y} \leftarrow \textsc{TemporalSmooth}\left(\{\bar{s}_j\}_{j=1}^{h}\right).$$

Return $\hat{Y}$ and $\{E_j^*\}_{j=1}^{h}$.

A.1.2 Additional Method Implementation Details

We provide additional implementation details for the routing features and the post-processing aggregation used in ARIA.

Routing Features. As introduced in Sec. 3.3, the routing representation x_j uses information available before either follow-up action is executed. The auxiliary MLP encodes the base response using TF-IDF with unigram and bigram features, a maximum vocabulary size of 500, and a minimum document frequency of 5. The text features are concatenated with a one-hot category encoding and four numerical features: the binary base prediction, the transformed response length $\log(1+\text{len})/10$, the normalized segment position, and the mean of preceding base predictions. The concatenated vector is fed to an MLP with two hidden layers of 128 and 64 units, trained to predict whether a complementary VLM would be helpful. Its output score $\mathbf{r}_j$ is included in x_j without requiring additional generative VLM inference.

For temporal context, we consider the neighborhood $W_j = \{v_i : 1 \le i \le h,\ |i-j| \le (w-1)/2\}$, where w is the window size. We arrange the lightweight base predictions $\tilde{y}_i^{(b)}$ within W_j in temporal order and concatenate this sequence with the normalized segment position j/h to form the temporal feature vector t_j. Such signals characterize whether the current prediction is consistent with its local temporal context.

For semantic relevance, we first extract a visual representation $e_j = g(v_j)$ using the frozen vision encoder g and retrieve the top-K semantically relevant segments from the same video according to cosine similarity. We denote their index set by $\mathcal{N}(j)$. We then compute the binary prediction distribution over the retrieved segments from their lightweight predictions. Let $p_j(c)$ denote the proportion of retrieved segments predicted as class $c \in \{0,1\}$. We then summarize prediction disagreement in this distribution using its entropy:

$$H_j^{(\text{sem})} = -\sum_{c\in\{0,1\}} p_j(c)\log p_j(c). \tag{5}$$

A smaller $H_j^{(\text{sem})}$ indicates more consistent predictions across semantically related scenes, while a larger value indicates stronger disagreement. Different from using $H_j^{(\text{sem})}$ as a hard consistency test, ARIA directly includes it in x_j as a continuous routing feature.

In addition, we incorporate a coarse category-level feature c_j obtained from a frozen zero-shot visual classifier. This information serves as a lightweight scene prior and does not introduce an additional generative VLM call. Together, the response-level descriptor r_j, temporal descriptor t_j, semantic-context entropy $H_j^{(\text{sem})}$, and category-level feature c_j constitute the routing representation used by all three routing heads.

Segment-Level Aggregation. After the selected reasoning action is executed, we obtain the routed segment-level prediction sequence $\{s_j\}_{j=1}^h$. Following the semantic redundancy of video data, for each target segment v_j, we retrieve its top-K semantically related segments using the same frozen visual encoder g. Their routed scores are combined using normalized cosine-similarity weights with respect to the target segment to obtain a refined segment-level score. In this way, predictions from visually related scenes provide additional evidence without requiring further VLM inference.

Lastly, the refined segment-level scores are temporally smoothed with a one-dimensional Gaussian kernel and mapped back to the corresponding frames to obtain the final anomaly score sequence $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_F]$. The specific choices of the retrieval size K and the smoothing configuration are reported in Sec. A.3.2.

A.1.3 Router Training Details

We provide additional details on the offline training procedure of the router in ARIA. The VLM parameters remain frozen throughout this process, and only the auxiliary MLP used to produce r_j and the three lightweight routing heads are trained.

Construction of Training Targets. For each segment v_j, we cache the predictions produced by the lightweight base inference, the denser visual inference from LOOK-MORE, and the complementary inference from COOPERATE. These cached outputs are used only for constructing the router training

targets and are never accessed by the router during inference. Specifically, the LOOK-MORE target is positive when the denser visual inference corrects an error made by the lightweight inference, while the COOPERATE target is positive when the cooperation inference corrects the corresponding error. The stage-1 target is positive if either follow-up action can correct the lightweight prediction. The segment-level ground-truth labels y_j^{seg} are derived from the original frame-level annotations.

Video-Level Cross-Validation. We perform nested video-level cross-validation exclusively within the official UCF-Crime training split, keeping all segments from the same video in the same fold. For each of the five outer folds, all fitting of feature preprocessors and predictive models is restricted to the outer-training videos. Within the outer-training split, we perform inner cross-fitting to generate out-of-fold MLP scores for training the GBT routing heads. For each inner fold, the TF-IDF vocabulary, category mapping, and MLP are fitted only on the inner-training videos and applied to the inner-held-out videos. The feature preprocessors and MLP are then refitted on the full outer-training split to generate MLP scores for the outer-held-out videos, which are evaluated using the GBT routing heads trained on the inner out-of-fold features. Predictions from the five outer-held-out folds are concatenated for router evaluation. No videos from the official UCF-Crime test split are included in this procedure.

Evaluation Protocol. For the main detection experiments, the router is trained once using only videos from the official UCF-Crime training split and is then kept fixed for evaluation on both UCF-Crime and XD-Violence. The same trained router is directly applied to XD-Violence without additional router training or dataset-specific adaptation. The detection results on both datasets are reported on their respective official test splits.

Training of the Auxiliary MLP. The auxiliary MLP takes TF-IDF features of the base response, a one-hot category encoding, and the numerical features described above. TF-IDF uses unigrams and bigrams with at most 500 features and a minimum document frequency of 5. The MLP has two hidden layers of 128 and 64 units and is trained on teammate-helpfulness labels using Adam with an initial learning rate of 10^{-3} for up to 50 epochs, with early stopping on an internal validation split. The auxiliary score $\mathbf{r}_j$ is generated using the nested video-level cross-fitting procedure described above and included in x_j as an auxiliary feature for the GBT routing heads.

Training of Routing Heads. The stage-1, LOOK-MORE, and COOPERATE heads take the same routing feature x_j as input but are trained independently using their corresponding binary targets. We instantiate each head with a lightweight HistGradientBoosting classifier. Since positive samples are relatively sparse for the routing tasks, class-balanced sample weights are used during training. The same classifier configuration is applied across all folds without per-fold hyperparameter tuning.

A.2 METHOD MOTIVATION AND DESIGN

This section further discusses the design motivations behind the adaptive reasoning mechanism in ARIA. The key objective of ARIA is not simply to allocate more reasoning to every video segment, but to adaptively determine whether additional computation is needed and, when it is beneficial, what form of additional evidence should be acquired.

This adaptive reasoning process involves two closely related design considerations. First, ARIA needs low-cost signals to characterize whether the lightweight base response is likely to benefit from further examination. Second, even when additional reasoning is beneficial, different segments may require different forms of evidence. Some errors may arise because sparse observations miss important visual cues, while others may arise because the base VLM provides an incomplete interpretation of the available evidence. Accordingly, ARIA distinguishes between two follow-up actions: LOOK-MORE, which acquires denser visual evidence using the same base VLM, and COOPERATE, which introduces a complementary model perspective from another VLM.

We therefore analyze the adaptive reasoning design from these two aspects. We first examine reliability signals that can support the decision of whether further examination is beneficial. We then analyze the different forms of additional evidence provided by LOOK-MORE and COOPERATE. Finally, under the adaptive-routing setting where complementary-model reasoning is selected, we further discuss how such complementary evidence should be incorporated and why the adopted cooperation mechanism is used.

A.2.1 Reliability Signals for Adaptive Routing

Adaptive reasoning first needs to determine whether the lightweight base response is likely to benefit from additional reasoning. A natural question is whether existing uncertainty or reliability estimation methods for language models can be directly used for this purpose.

Previous studies have explored uncertainty and reliability estimation in language models using internal token-level probabilities during generation (Zhu et al., 2025; Hu et al., 2024) or response-level self-consistency across repeated generations (Lin et al., 2024). We consider these two representative strategies as alternative approaches for characterizing the reliability of VLM responses in VAD.

(Option 1) Token Probability-Based Reliability Estimation: *Given a video segment v_j and instruction θ, the VLM f generates a decision token ('0' for normal or '1' for anomaly) during the natural language generation process. Along with the token, f also produces the associated probabilities p_0 and p_1 for the two possible outcomes. A common way to quantify predictive uncertainty is to compute the entropy of this binary distribution:*

$$H_{\text{token}}(v_j) = -\big(p_0 \log p_0 + p_1 \log p_1\big). \tag{6}$$

A larger entropy indicates that the model assigns similar probabilities to the two outcomes and is therefore less confident in its prediction, whereas a smaller entropy indicates a more concentrated and confident prediction.

(Option 2) Repeated-Generation Self-Consistency: *Given a video segment v_j and instruction θ, we query the VLM f repeatedly for N trials, obtaining a set of predictions*

$$\{\tilde{y}_j^{(1)}, \tilde{y}_j^{(2)}, \ldots, \tilde{y}_j^{(N)}\}.$$

The degree of agreement among these predictions can be used as an empirical reliability signal. High agreement suggests that the model produces a stable prediction for the segment, whereas disagreement across repeated generations indicates greater uncertainty in the response. For a simple binary baseline, we mark a response as unreliable if at least two repeated generations produce different predictions:

$$U_{\text{self}}(v_j) = \mathbb{1}\left\{\exists\, i, l \text{ such that } \tilde{y}_j^{(i)} \neq \tilde{y}_j^{(l)}\right\}. \tag{7}$$

Discussion: Limitations of Existing Solutions. Despite their popularity in LLM pipelines, these strategies have limitations when used to support adaptive reasoning in VLM-based VAD. Firstly, Option 1 assesses prediction reliability from token probabilities during language decoding. Such probabilities reflect the model's confidence in its generated decision, but may not fully characterize whether the lightweight response would benefit from additional visual evidence or a complementary reasoning perspective. Secondly, for Option 2, repeatedly querying the model for each segment is impractical for video data. Due to the large number of video segments, the computation increases to N times that of a single query, making the overhead too demanding for a mechanism whose purpose is to decide whether additional computation should be allocated in the first place.

These limitations motivate ARIA to characterize the lightweight response using low-cost response-level and contextual signals that are already available after the initial inference, rather than relying on repeated generative VLM calls. We evaluate these reliability estimation strategies on UCF-Crime (Sultani et al., 2018), with the results reported in Table 11. The subsequent adaptive routing and refinement procedures are described in Secs. 3.4 and 3.5.

A.2.2 Different Forms of Additional Evidence for Adaptive Reasoning

Reliability characterization addresses whether a lightweight response is likely to benefit from further examination. However, adaptive reasoning also requires determining what kind of additional evidence should be acquired once further reasoning is predicted to be beneficial. A lightweight prediction can be incorrect for different reasons. In some cases, sparse observations may omit important visual cues; in others, the relevant visual information may already be present, but the base VLM may provide an incomplete or limited interpretation. Applying the same follow-up strategy to both cases would therefore fail to account for their different evidence needs.

In ARIA, Look-More and Cooperate serve as two complementary follow-up actions for these different situations. Look-More acquires additional visual evidence using the same base VLM,

Table 6: Comparison of segment-selection and follow-up strategies on UCF-Crime. Adaptive selection uses the learned stage-1 gate, whereas entropy selection chooses 50% of segments based on prediction entropy. Δ denotes the AUC improvement over the lightweight 4-frame base, in percentage points.

Configuration	**AUC (%)**	Δ
Lightweight Base (4 frames) only	83.73	–
Adaptive + LOOK-MORE only	86.63	+2.90
Adaptive + COOPERATE only	87.28	+3.55
LOOK-MORE only on 50% of segments	87.19	+3.46
COOPERATE only on 50% of segments	88.15	+4.42
Full Adaptive Reasoning	**88.32**	**+4.59**

whereas COOPERATE introduces an additional reasoning perspective from another VLM. The role of the adaptive router is therefore not only to determine whether further reasoning is beneficial, but also to select which form of additional evidence has greater predicted corrective utility for the current segment.

Denser Visual Evidence in LOOK-MORE. The lightweight base inference observes only κ_b sparsely sampled frames. Therefore, informative visual cues may be missed when the observed evidence is incomplete. LOOK-MORE increases the number of sampled frames to $\kappa_m > \kappa_b$, providing denser observations of the same target segment. When the initial error is caused by insufficient visual evidence, these additional observations can provide the missing cues required for a more reliable prediction. Thus, LOOK-MORE provides an adaptive follow-up action for segments whose main limitation is insufficient visual observation.

Complementary Evidence in COOPERATE. Additional visual observations are not necessarily the most useful form of follow-up evidence for every segment. In some cases, the relevant visual cues may already be present, while the limitation instead arises from how the base VLM interprets them. For such segments, ARIA introduces a second VLM f_{co} to provide a complementary reasoning perspective.

Importantly, COOPERATE is not executed after LOOK-MORE fails. The two actions represent alternative follow-up paths within the adaptive reasoning space, and the router selects between them according to their predicted corrective utility. In this sense, complementary-model reasoning is used as a targeted evidence source for segments predicted to benefit more from an alternative model perspective than from denser visual observations.

A.2.3 SEGMENT SELECTION AND FOLLOW-UP STRATEGIES

Table 6 extends the main-text ablation with two variants that select 50% of segments according to semantic-context entropy. While adaptive selection uses the learned stage-1 gate, these variants use a fixed selection rate and apply either LOOK-MORE or COOPERATE to all selected segments. Unselected segments retain their lightweight base predictions.

Full ARIA achieves 88.32% AUC, improving over adaptive selection with COOPERATE only (87.28%) by 1.04 percentage points and over the 50% selection variant with COOPERATE (88.15%) by 0.17 percentage points. The latter comparison involves different segment-selection rules and potentially different follow-up rates, and therefore does not isolate the effect of action selection or establish an advantage at matched inference cost.

A.2.4 COLLABORATIVE REASONING DESIGN UNDER ADAPTIVE ROUTING

The analysis above motivates complementary-model evidence as one possible follow-up action within adaptive reasoning. Once a segment is routed to COOPERATE, however, an additional design question remains: how should the complementary model perspective be incorporated into the existing lightweight response?

The objective here is not to construct an independent multi-agent reasoning system for every segment. Instead, the collaborative mechanism serves as a selective follow-up operation after adaptive routing has determined that complementary-model reasoning is appropriate. We therefore consider three

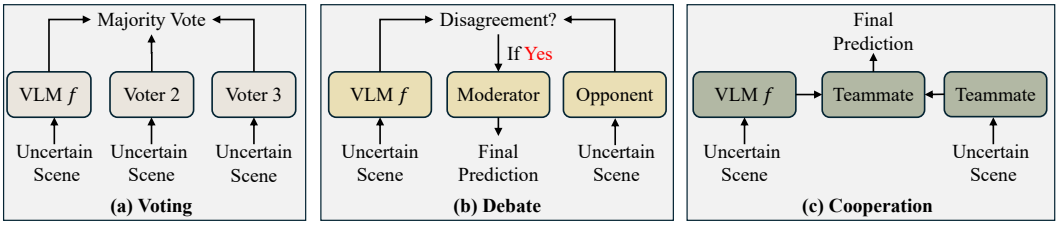

Figure 4: Comparison of voting-, debate-, and cooperation-based collaborative reasoning mechanisms for incorporating complementary model perspectives after adaptive routing. The proposed cooperation-based design directly combines the existing base response with complementary evidence from another VLM.

collaborative reasoning mechanisms that can be used for the COOPERATE branch, including voting, debate, and cooperation, as illustrated in Fig. 4. The proposed cooperation-based design has been detailed in Sec. 3.5. We further describe the other variants, which are used as baselines in Table 12.

Voting-Based Collaborative Reasoning. For this variant, we introduce two additional VLMs, f_{voter2} and f_{voter3}. For a target segment v_j routed to the COOPERATE branch, we feed v_j into each VLM, and each model independently determines whether the segment contains an anomaly. The majority class predicted by the models is used as the final prediction. The prompt used by each voter is detailed in Sec. A.3.3. This variant is an intuitive form of multi-agent collaboration as discussed in (Kaesberg et al., 2025).

From the perspective of adaptive reasoning, however, voting mainly aggregates multiple independent predictions. Although it utilizes outputs from multiple models, it does not explicitly exploit the existing lightweight response as evidence to be complemented. Therefore, the additional models largely repeat the full VAD decision process rather than specifically providing the missing perspective identified by adaptive routing.

Debate-Based Collaborative Reasoning. For this variant, when a segment is routed to the COOPERATE branch, one VLM-based agent serves as a debater f_{debate} and provides an alternative judgment to the base model f, while another agent serves as a moderator f_{moderate} to determine which opinion is more reliable. After receiving both opinions, the moderator makes the final decision. This is similar to the method by (Liang et al., 2024). The prompt used by the moderator is detailed in Sec. A.3.3.

Debate explicitly introduces an alternative model perspective, but it additionally requires a separate moderation stage to determine how the two opinions should be reconciled. Under the adaptive reasoning setting of ARIA, where follow-up computation is selectively invoked only for routed segments, such an additional fusion stage introduces a more complex reasoning path than directly incorporating complementary evidence into the existing response.

Discussion. The proposed cooperation-based design detailed in Sec. 3.5 is aligned with the role of COOPERATE as a targeted follow-up action under adaptive routing. Rather than asking multiple models to independently repeat the complete VAD decision or introducing an additional moderator to arbitrate between competing opinions, the cooperation-based design directly reuses the lightweight base response and combines it with a complementary perspective from another VLM.

This design has two desirable properties in the adaptive reasoning setting. First, it preserves and reuses the prediction and explanation already obtained from the lightweight base inference, rather than discarding the computation that has already been performed. Second, the additional VLM is used specifically to provide a complementary perspective for segments that the router has selected for COOPERATE. Thus, the additional computation is tied directly to the evidence need identified by adaptive routing.

In this sense, COOPERATE is not a standalone multi-agent module applied uniformly to the video. Instead, it is one of the adaptive follow-up actions in ARIA, designed to supplement the existing lightweight reasoning when a complementary model perspective is predicted to provide greater corrective utility. The detection-performance comparison among the collaborative reasoning variants is reported in Table 12.

A.2.5 THEORETICAL MOTIVATION FOR COMPLEMENTARY VLM EVIDENCE

We further provide an information-theoretic motivation for introducing a complementary VLM in the COOPERATE branch. The analysis below does not imply that an additional VLM improves every segment. Instead, it formalizes why complementary evidence can reduce uncertainty when it contains label-relevant information that is not already captured by the lightweight base response.

Theorem. Let y_j^{seg} denote the true binary label of segment v_j, and let $E_j^{(b)}$ and $E_j^{(\text{co})}$ denote the reasoning evidence provided by the base model and the cooperation model, respectively. Incorporating $E_j^{(\text{co})}$ cannot increase the conditional uncertainty, i.e.,

$$H(y_j^{\text{seg}} \mid E_j^{(b)}, E_j^{(\text{co})}) \leq H(y_j^{\text{seg}} \mid E_j^{(b)}).$$

If $E_j^{(\text{co})}$ provides additional information about y_j^{seg} beyond $E_j^{(b)}$, i.e.,

$$I(y_j^{\text{seg}}; E_j^{(\text{co})} \mid E_j^{(b)}) > 0,$$

the inequality is strict. Similarly, the expected posterior variance satisfies

$$\mathbb{E}[\text{Var}(y_j^{\text{seg}} \mid E_j^{(b)}, E_j^{(\text{co})})] \leq \mathbb{E}[\text{Var}(y_j^{\text{seg}} \mid E_j^{(b)})].$$

Proof. By the definition of conditional mutual information,

$$I(y_j^{\text{seg}}; E_j^{(\text{co})} \mid E_j^{(b)}) = H(y_j^{\text{seg}} \mid E_j^{(b)}) - H(y_j^{\text{seg}} \mid E_j^{(b)}, E_j^{(\text{co})}) \geq 0.$$

Therefore,

$$H(y_j^{\text{seg}} \mid E_j^{(b)}, E_j^{(\text{co})}) \leq H(y_j^{\text{seg}} \mid E_j^{(b)}),$$

with strict inequality whenever $E_j^{(\text{co})}$ contains additional information about y_j^{seg} beyond $E_j^{(b)}$. Likewise, conditioning on additional evidence cannot increase the expected posterior variance. Thus, complementary evidence from the cooperation model can reduce prediction uncertainty when it provides information not already contained in the base-model evidence.

Intuition. If the cooperation model merely repeats the evidence already provided by the base model, the additional benefit can be limited. In contrast, when it provides new label-relevant evidence through different model knowledge or reasoning paths, combining $E_j^{(b)}$ and $E_j^{(\text{co})}$ can make the prediction more reliable.

From the perspective of adaptive reasoning, this result does not imply that complementary-model reasoning should be applied to every segment. Instead, it motivates complementary evidence as one useful candidate follow-up action when the lightweight response is predicted to benefit from a different model perspective.

Interpretation for Video Anomaly Detection (VAD). In VAD, $E_j^{(b)}$ and $E_j^{(\text{co})}$ can be viewed as complementary reasoning evidence produced by the base and cooperation VLMs, respectively. Because individual video segments provide only limited context, different models may attend to different visual or semantic cues. Therefore, the cooperation VLM can provide additional evidence for ambiguous cases that are difficult to resolve using the lightweight base response alone.

Meanwhile, other segments may instead be limited primarily by sparse visual observations and therefore benefit more from LOOK-MORE. This distinction motivates ARIA to model LOOK-MORE and COOPERATE as alternative adaptive follow-up actions rather than applying complementary-model reasoning uniformly to all segments.

A.3 EXPERIMENTAL SETUP AND EXECUTION

A.3.1 DATASET DETAILS

We conduct experiments on two large-scale VAD datasets: (1) UCF-Crime (Sultani et al., 2018) and (2) XD-Violence (Wu et al., 2020). The details are as follows:

- The **UCF-Crime** dataset is collected from real-world surveillance videos (128 hours long in total), covering crime-related anomalies including abuse, arrest, arson, assault, burglary, explosion, fighting, road accident, robbery, shoplifting, shooting, stealing, and vandalism. The training set has 1610 videos (810 abnormal ones and 800 normal ones), while the test set has 290 videos (140 abnormal ones and 150 normal ones). The total number of test frames is over 1 million (1,111,808), and abnormal frames account for 7.92%. The average duration of a test video is 2.13 minutes, which is relatively long compared to common video datasets and serves as an important benchmark.
- **XD-Violence** is another representative large-scale (217 hours long in total) VAD dataset with 6 anomaly categories, i.e., abuse, car accident, explosion, fighting, riot, and shooting, which defines anomalous events as the ones related to violence. This dataset is collected from movies and YouTube videos. It has 3954 training videos and 800 test videos (500 abnormal ones and 300 normal ones). The total number of test frames is over 2 million (2,335,801), and abnormal frames account for 23.07%. The average duration of a test video is 1.62 minutes.

A.3.2 Hyperparameter Setting

During inference, following (Zanella et al., 2024; Ye et al., 2025), the interval between adjacent segment centers d is set to 16 frames, and each segment corresponds to an approximately 10-second temporal window centered at the segment location.

For lightweight base inference, ARIA sparsely samples $\kappa_b = 4$ frames from each segment. For segments routed to LOOK-MORE, the base VLM uses a denser visual sampling configuration with $\kappa_m = 8$ frames, where $\kappa_m > \kappa_b$. The COOPERATE branch also uses 8-frame visual inputs. This allows ARIA to keep the initial inference cost low while allocating additional visual observations and complementary-model reasoning only to selected segments.

For temporal-context features, we use a local window size of $w = 5$. When computing semantic similarity, we use ImageBind as the frozen visual feature extractor g. For a test video V containing h segments, the number of semantically relevant segments is set to $K = 0.15h$. This setting is used to retrieve scenes that are semantically related to the target segment and provides relevant visual evidence for semantic-context features and subsequent segment-level aggregation.

After semantic aggregation, the refined segment-level scores are temporally smoothed with a one-dimensional Gaussian kernel and mapped to frame-level anomaly scores.

For adaptive routing, we set the stage-1 routing threshold to $\tau = 0.20$. When $p_{\mathrm{s1}}(x_j) < \tau$, ARIA retains the lightweight base response; otherwise, the segment proceeds to additional reasoning and is further routed to either LOOK-MORE or COOPERATE (the detailed routing rule is provided in Sec. 3.4).

A.3.3 Prompts Used in Inference

Prompt for Lightweight Base Inference θ. The lightweight base inference follows the individual reasoning prompt provided by VERA (Ye et al., 2025). Given the visual input of a target segment, the base VLM independently answers a set of VAD-specific guiding questions, generates a short explanation, and produces a binary anomaly prediction. This prompt serves as the common individual reasoning template θ used throughout ARIA.

Prompt for Lightweight Base Inference θ

System Prompt:
You are the model. You are designed to do binary classification. The input is a sequence of video frames for identifying whether there is an anomaly in the video; you need to output the class label, i.e., an integer in the set $\{0, 1\}$. 0 represents normal video, and 1 represents abnormal video. Please answer the prompt questions.

User:
Answer the following questions based on what you see from the video frames and provide an explanation in one sentence.

- **Guiding Questions:** [Guiding Questions]

Based on the analysis above, please conclude your answer to "Is there any anomaly in the video?" in *'Yes, there is an anomaly'* or *'No, there is no anomaly'*.

- **Input:** [Visual Data]

Output Requirements (MUST follow exactly):
Please give your output strictly in the following format:

- **Answers to Prompt Questions:** [Provide your analysis by answering the questions listed in Prompt Questions.]
- **Output:** [ONLY the integer class label; make necessary assumptions if needed]

Please ONLY reply according to this format; don't give me any other words.

Prompt for LOOK-MORE. When a segment is routed to LOOK-MORE, the same base VLM re-examines the target segment using the individual reasoning template θ above. The task formulation and guiding questions remain unchanged; the distinction is that LOOK-MORE operates on the denser visual evidence acquired by the selected follow-up path. No previous prediction or explanation is provided to the model as a self-reflection instruction. Thus, LOOK-MORE isolates the benefit of acquiring additional visual evidence rather than introducing an additional reasoning agent or a self-reflection process.

Formally, the LOOK-MORE response is generated as

$$(\tilde{y}_j^{(m)}, E_j^{(m)}) = f(v_j^{(\kappa_m)}; \theta).$$

Prompt for COOPERATE ξ. The COOPERATE action uses a complementary VLM f_{co} to provide an independent model perspective and then perform cross-model verification. It consists of two stages, both executed only after the target segment v_j has been routed to COOPERATE.

Stage 1: Independent Complementary Reasoning. The complementary VLM first observes the target visual input and independently generates a binary prediction $\tilde{y}'_j$ and an explanation E'_j using the same individual reasoning template θ, without accessing the lightweight base response:

$$(\tilde{y}'_j, E'_j) = f_{\text{co}}(v_j; \theta).$$

Here, $\tilde{y}'_j$ and E'_j are intermediate outputs of the complementary VLM rather than the final outputs of the COOPERATE action.

Stage 2: Cross-Model Verification. The same complementary VLM then performs cross-model verification using the cooperation prompt ξ. The prompt provides the target visual input, the lightweight base prediction and explanation $(\tilde{y}_j^{(b)}, E_j^{(b)})$, and the independently generated complementary prediction and explanation $(\tilde{y}'_j, E'_j)$. The model evaluates both responses, compares their reasoning, and produces the final COOPERATE prediction $\tilde{y}_j^{(\text{co})}$ and explanation $E_j^{(\text{co})}$:

$$(\tilde{y}_j^{(\text{co})}, E_j^{(\text{co})}) = f_{\text{co}}\big(v_j, \tilde{y}_j^{(b)}, E_j^{(b)}, \tilde{y}'_j, E'_j; \xi\big).$$

These two stages together constitute the COOPERATE action described in Sec. 3.5. The complementary model first forms its own interpretation and then re-evaluates the segment using both the lightweight

base response and its independently generated response. Neither stage accesses any output from LOOK-MORE.

Prompt for COOPERATE ξ

System Prompt:
You are the model. You are designed to do binary classification. The input is a sequence of video frames for identifying whether there is an anomaly in the video. You need to output the class label, i.e., an integer in the set $\{0, 1\}$. 0 represents normal video, and 1 represents abnormal video. Please follow the instruction below to make a conclusion.

User:
There are two analyses below, please conclude your answer to "Is there any anomaly in the video?" in *'Yes, there is an anomaly'* or *'No, there is no anomaly'* by watching the video carefully and pondering whether these two opinions are consistent.

- **Opinion 1:** [Base prediction $\tilde{y}_j^{(b)}$ and explanation $E_j^{(b)}$ from f]
- **Opinion 2:** [Independent prediction $\tilde{y}'_j$ and explanation E'_j from f_{co}]
- **Input:** [Video Frames]

Output Requirements (MUST follow exactly):
Please give your output strictly in the following format:

- **New Analysis and Conclusion:** [Please provide a new analysis with these two opinions and make a conclusion.]
- **Output:** [ONLY the integer class label]

Please ONLY reply according to this format. Don't give me any other words.

Prompts for Collaborative Baselines. For the voting- and debate-based variants used in the collaborative-reasoning comparison, we retain the same individual reasoning formulation wherever an agent independently performs VAD. The additional prompt design is introduced only when the collaborative mechanism requires an explicit aggregation or moderation step.

Voting-Based Collaborative Reasoning. Each voter independently analyzes the target segment using the same individual reasoning prompt θ used by the lightweight base inference. The voters do not access one another's responses. Their binary predictions are subsequently aggregated by majority voting. Therefore, no additional voter-specific prompt template is introduced beyond the individual reasoning prompt in Sec. A.3.3.

Debate-Based Collaborative Reasoning. For the debate baseline, the base model and the opponent independently generate their judgments using the individual reasoning prompt θ. A separate moderator then receives the two debaters' analyses and determines the final prediction. During implementation, the reasoning outputs from the two agents are inserted into the corresponding `[Debater's Analysis]` fields below.

Prompt for Debate-Based Collaborative Reasoning

System Prompt:
You are designed to do binary classification on whether an anomaly exists in the given video input. The input is a sequence of video frames for identifying whether there is an anomaly in the video. You need to output the class label, i.e., an integer in the set {0, 1}. 0 represents a normal video, and 1 represents an abnormal video. Please follow the instructions below to make a conclusion.

User:
There are two debaters involved in debating whether an anomaly exists. They will present their analyses and discuss their perspectives. Please evaluate both sides' analyses and decide which one is correct. Please conclude your answer to "Is there any anomaly in the video?" based on what you see in the video.

- **Debater 1's Argument:** [Debater's Analysis]
- **Debater 2's Argument:** [Debater's Analysis]
- **Input:** [Visual Data]

Output Requirements (MUST follow exactly):
Please give your output strictly in the following format:

- **New Analysis and Conclusion:** [Please provide a new analysis with these two opinions and make a conclusion.]
- **Output:** [ONLY the integer class label]

Please ONLY reply according to this format; don't give me any other words.

A.3.4 LLM-AS-JUDGE EVALUATION PROTOCOL

Prompt for LLM-as-Judge Evaluation. The LLM-as-Judge evaluation employs DeepSeek-R1-70B (Guo et al., 2025) as the judge model. For each window, the judge is given two reasoning responses and asked to assign separate scores to each response across three dimensions. The prompt template is as follows:

Prompt for LLM-as-Judge Evaluation

System Prompt:
You are an expert evaluator assessing the quality of video anomaly detection reasoning. You will be given two reasoning responses about the same video segment.

User:
Evaluate BOTH responses on each of the following three dimensions. For each dimension, assign an INTEGER score from 1 to 10 (higher = better).

- **Reasonability:** Is the causal chain complete and logically consistent, with no circular reasoning or unsupported claims?
- **Novel Evidence:** Does the response introduce specific visual observations or perspectives not present in the other response?
- **Conciseness:** Is the information dense and directly relevant, free of redundant or formulaic language?
- **Response A:** [Reasoning Response A]
- **Response B:** [Reasoning Response B]

Output Requirements (MUST follow exactly):
Output ONLY a valid JSON object in exactly this format, with no additional text or explanation:

```
{"A": {"reasonability": 7, "novel_evidence": 4,
"conciseness": 6},
 "B": {"reasonability": 8, "novel_evidence": 7,
"conciseness": 7}}
```

Guiding Questions. We replace the placeholder `[Guiding Questions]` with the guiding questions learned by VERA (Ye et al., 2025). The same guiding-question template is used by the lightweight base inference, LOOK-MORE, the independent complementary reasoning stage of COOPERATE, and the independent voters in the voting baseline.

For the default setup, InternVL3-8B serves as the base model f and InternVL2-8B serves as the complementary model f_{co}. The guiding questions are:

1. Are there any people in the video who are not in their typical positions or engaging in activities that are not consistent with their usual behavior?
2. Are there any vehicles in the video that are not in their typical positions or being used in a way that is not consistent with their usual function?
3. Are there any objects in the video that are not in their typical positions or being used in a way that is not consistent with their usual function?
4. Is there any visible damage or unusual movement in the video that indicates an anomaly?
5. Are there any unusual sounds or noises in the video that suggest an anomaly?

A.3.5 BATCHED INFERENCE AND EFFICIENCY ANALYSIS

For efficient execution, independent segment queries assigned to the same inference path are processed in batches of size B. Each segment retains its own visual input, prompt, and output, without sharing contextual information with other segments in the same batch. Therefore, batching changes the execution configuration without changing the per-segment input construction or the predefined routing policy.

In the main experiments, we use $B = 8$ for both the lightweight base inference and the follow-up VLM inference. Within COOPERATE, each segment's verification query depends on its independently generated response, while queries for different segments remain independent and can be batched within the corresponding stage.

Table 7 reports the per-segment inference time with and without batching under the same hardware environment. For the base InternVL3-8B model, batching reduces the average inference time

Table 7: Effect of batched inference on per-segment VLM runtime. Independent segment queries are executed in batches without sharing context across segments.

VLM	$B = 1$ (s/segment)	$B = 8$ (s/segment)	Speedup
InternVL3-8B (Base)	4.598	**1.138**	$4.04\times$
InternVL2-8B (Complementary VLM)	3.551	**1.298**	$2.74\times$

from 4.598 seconds to 1.138 seconds per segment, corresponding to a $4.04\times$ speedup. For the complementary InternVL2-8B model, the per-segment inference time decreases from 3.551 seconds to 1.298 seconds, yielding a $2.74\times$ speedup.

These results show that batched execution reduces the reported per-segment inference time. Combined with adaptive routing, which avoids invoking follow-up inference for every segment, batching contributes to the wall-clock efficiency reported in Table 5.

A.3.6 Running Time

The running time of ARIA depends on both the selected reasoning actions and the batched execution configuration. Every segment first undergoes one lightweight base inference. Segments routed to Stop require no additional VLM inference, while those routed to Look-More require one additional query to the base VLM with denser visual input. For segments routed to Cooperate, the complementary VLM is queried twice: first to generate an independent prediction and explanation, and then to verify this response together with the lightweight base response and produce the final prediction and explanation.

For a video containing h segments, let n_{lm} and n_{co} denote the numbers of segments routed to Look-More and Cooperate, respectively. The total number of logical segment-level VLM queries is therefore

$$N_{\mathrm{query}} = h + n_{\mathrm{lm}} + 2n_{\mathrm{co}}.$$

This count refers to individual segment queries before batching, rather than the number of batched execution calls. It also does not directly determine wall-clock time, since the queries differ in visual input size, prompt content, and generated output length.

In the main experiments, independent queries are processed with $B = 8$, as described in Sec. A.3.5. Adaptive routing reduces unnecessary follow-up inference, while batching improves the execution efficiency of the remaining queries. Consequently, the runtime of ARIA should not be inferred solely from the number of participating VLMs or from a fixed per-query latency.

As reported in Table 5, the average inference time is 17.1 minutes per video for fixed 8-frame inference, 25.7 minutes for fixed collaborative reasoning, and 5.9 minutes for ARIA. The runtime of ARIA corresponds to 34.5% of the fixed 8-frame baseline. These results describe the performance of the evaluated system configurations, including adaptive routing and batched execution, rather than attributing the entire runtime reduction to routing alone.

A.4 Additional Quantitative Analysis

A.4.1 Hyperparameter Sensitivity Test

For key hyperparameters including w and K, we conduct sensitivity tests on them on the UCF-Crime dataset, with InternVL3-8B as f and InternVL2-8B as f_{co}. The results are as follows, with other hyperparameters fixed when alternating one.

Sensitivity Test for w. From Table 8, the performance of ARIA is stable with a varying w from 3 to 11. This result shows that the VAD performance is insensitive to w. Thus, we can set $w = 5$ in general cases.

Sensitivity Test for K. Table 9 shows that if we do not use semantically similar scenes (i.e., setting $K = 0$), the performance will drop to 84.57%, which demonstrates the necessity of introducing scenes with similar semantic meaning for handling suspicious segments. Meanwhile, when K is

Table 8: Influence of the window size w on AUC (%).

w	3	5	7	9	11
AUC	88.37	88.32	88.28	88.24	88.16

Table 9: Influence of the number of appearance-similar scenes K on AUC (%).

K	0	$0.05 \cdot h$	$0.10 \cdot h$	$0.15 \cdot h$	$0.20 \cdot h$
AUC	84.57	88.14	88.30	88.32	86.54

alternated in a reasonable range around $0.05 \cdot h$ to $0.15 \cdot h$, the performance will be stable and kept high. If we set K to a relatively huge number, $0.20 \cdot h$, the performance will drop slightly. Thus, selecting $0.15 \cdot h$ for K is generally a good choice.

A.4.2 Performance–Cost Trade-off

We analyze the sensitivity of detection performance and computational cost to the stage-1 routing threshold τ. Table 10 reports AUC at seven selected thresholds, together with estimates of average token consumption and batched inference time per video. At the evaluated thresholds between 0.10 and 0.20, AUC ranges from 88.31% to 88.34%. Increasing τ to 0.35 and 0.50 reduces AUC to 87.67% and 85.52%, respectively. We use $\tau = 0.20$ in the main experiments, achieving an AUC of 88.32%, only 0.02 percentage points below the highest AUC in this sweep.

A.4.3 Diagnostic Analysis of Reliability Signals and Reasoning Strategies

We conduct additional diagnostic experiments on UCF-Crime to better understand two design choices of ARIA: which signals are informative for characterizing segment-level reasoning under limited context, and which forms of additional reasoning can provide useful complementary evidence. Unless otherwise specified, InternVL2-8B is used as the base VLM in these diagnostic experiments.

Reliability Signal Analysis. As shown in Table 11, we compare several signals for characterizing segment-level reasoning under context-limited inputs. Token probability yields a 0.25 percentage-point AUC improvement, while self-consistency provides a larger gain of 0.71 percentage points but requires repeated VLM inference ($N = 3$). Temporal and semantic relevance improve AUC by 0.82 and 0.98 percentage points, respectively, and their combination further increases the gain to 1.24 percentage points. These results suggest that temporally and semantically related scenes provide useful contextual evidence beyond the prediction of an isolated segment. Motivated by this observation, ARIA incorporates temporal and semantic context as routing signals for adaptive reasoning.

Reasoning Strategy Analysis. We further compare different strategies for providing additional reasoning under context-limited inputs in Table 12. Self-Reflection and Skeleton-of-Thought decrease detection performance in this diagnostic setting, whereas Voting, Debate, and Cooperation improve over the base inference.

Among the evaluated strategies, Cooperation achieves the highest AUC, showing that a complementary model perspective can provide useful additional evidence when the initial response is insufficient. However, this result does not imply that complementary-model reasoning is always the appropriate follow-up action. In some cases, the limitation instead arises from sparse visual observations and can be addressed by acquiring denser evidence with the same VLM. This observation motivates ARIA to determine both whether additional reasoning is needed and which form of additional evidence is more helpful.

A.4.4 Additional Analysis of Reasoning Behavior

We further conduct a controlled diagnostic analysis to understand how different forms of follow-up reasoning modify the initial VLM prediction. In particular, we compare Self-Reflection with Cooperate, which introduces the complementary model perspective used in ARIA. This analysis

Table 10: Threshold sensitivity at seven selected operating points. AUC values are taken from the threshold sweep. Token consumption and batched inference time are estimates scaled to match the main-experiment costs at $\tau = 0.20$, as described in the text. The default threshold is $\tau = 0.20$.

τ	AUC (%)	Tokens/Video	Time/Video (min)
0.05	88.21	770,322	6.42
0.10	88.31	749,597	6.21
0.12	**88.34**	742,231	6.15
0.15	88.31	732,118	6.04
0.20	88.32	**716,512**	**5.90**
0.35	87.67	668,695	5.43
0.50	85.52	627,369	5.03

Table 11: Diagnostic comparison of reliability signals under context-limited VLM inference on UCF-Crime. Improvements over the base VLM are shown in parentheses.

Operation	AUC (%)
InternVL2-8B (f)	86.33
+ Token Probability (Hu et al., 2024)	86.58 (+0.25)
+ Self-Consistency (Lin et al., 2024)	87.04 (+0.71)
+ Temporal Relevance	87.15 (+0.82)
+ Semantic Relevance	87.31 (+0.98)
+ Temporal & Semantic Relevance	**87.57** (+1.24)

is intended to characterize their reasoning behavior rather than to evaluate the full adaptive routing pipeline.

Prediction Transition Analysis. We categorize all 69,634 segments in UCF-Crime according to the correctness of the prediction before and after follow-up reasoning. As shown in Table 13, among all 69,634 segments, Self-Reflection changes 11.78% from correct to incorrect, while 9.66% change from incorrect to correct. Thus, harmful changes occur more frequently than beneficial ones.

COOPERATE exhibits a different transition pattern. Among all segments, 7.14% change from correct to incorrect, while 7.27% change from incorrect to correct. Although both strategies can modify the initial prediction, COOPERATE introduces substantially fewer harmful revisions. This provides additional evidence that a complementary VLM can supply useful information when the lightweight response is insufficient.

Importantly, this diagnostic result does not suggest that COOPERATE should be applied to every segment. COOPERATE constitutes only one possible follow-up action in ARIA; other segments can instead benefit from denser visual evidence through LOOK-MORE. The role of the adaptive router is therefore to determine when additional reasoning is beneficial and which form of additional evidence is more appropriate.

A.4.5 ORACLE UPPER BOUND FOR SUSPICIOUS SEGMENT IDENTIFICATION

To measure how much of the available selection headroom is captured by the adaptive router, we introduce an Oracle baseline that uses the ground-truth error of the lightweight base prediction to identify segments requiring refinement. We further compare it with an Always-On baseline that flags all segments and a Random 50% baseline that selects half of the segments at random. Table 14 reports the AUC of each selection rule together with the headroom recovered relative to the Oracle upper bound, defined as

$$\frac{\text{method} - \text{Always-On}}{\text{Oracle} - \text{Always-On}}.$$

ARIA recovers 42% of the selection headroom, compared with 19% recovered by random selection. This result shows that the learned adaptive routing strategy selects more informative segments for refinement than random selection under the evaluated setting.

Table 12: Diagnostic comparison of additional reasoning strategies under context-limited VLM inference on UCF-Crime. Improvements over the base VLM are shown in parentheses.

Operation	AUC (%)
InternVL2-8B (f)	86.33
+ Self-Reflection via SRPO	84.54 (-1.79)
+ Skeleton-of-Thought	85.00 (-1.33)
+ Voting (Kaesberg et al., 2025)	86.89 (+0.56)
+ Debate (Liang et al., 2024)	87.00 (+0.67)
+ Cooperation	**87.57** (+1.24)

Table 13: Segment-level prediction transitions on UCF-Crime. Percentages are computed over all 69,634 segments. Correct→Incorrect represents harmful revisions, while Incorrect→Correct represents beneficial corrections.

Prediction Transition	Self-Reflection	COOPERATE
Correct → Correct	42,449 (60.96%)	**45,680 (65.60%)**
Incorrect → Incorrect	12,255 (17.60%)	13,920 (19.99%)
Correct → Incorrect	8,205 (11.78%)	**4,974 (7.14%)**
Incorrect → Correct	6,725 (9.66%)	5,060 (7.27%)

The remaining gap between the Oracle and ARIA ($90.19\% - 88.32\% = 1.87\%$ AUC) indicates that suspicious-segment identification remains non-trivial and that there is still room for improving the adaptive routing mechanism.

A.4.6 ADDITIONAL GENERALIZATION ANALYSIS

We further evaluate the robustness of ARIA across three aspects: the complementary VLM used by COOPERATE, the base VLM architecture, and the VAD instruction prompt. Unless otherwise stated, all remaining components and inference settings are kept unchanged.

Across Complementary VLMs. We first examine whether the benefit of COOPERATE depends on a particular complementary VLM. As shown in Table 15, all four tested VLMs improve over the corresponding base inference, with gains ranging from 0.13 to 1.24 percentage points. The improvement is therefore not restricted to a specific complementary model, suggesting that COOPERATE is robust to different complementary VLM choices.

Across Base VLM Architectures. We next replace the base VLM with models from different InternVL and Qwen-VL families. ARIA improves over the corresponding base inference for all evaluated architectures. In particular, the gains reach 4.59 percentage points with InternVL3-8B and 4.83 percentage points with Qwen2-VL-7B, indicating that the benefit of adaptive reasoning is preserved across different base VLM architectures.

Across Prompt Variants. We further examine the robustness of ARIA to different VAD instruction prompts. Across all three prompt variants, ARIA improves over the corresponding base inference, with gains ranging from 1.24 to 3.16 percentage points. These results show that the improvement from adaptive reasoning is preserved under different prompt formulations.

A.4.7 RESULTS OF SCALING ARIA

We can scale up the collaborative reasoning mechanism in ARIA with more VLMs. We further use InternVL3-8B as f and add InternVL2-8B, InternVL3-8B, and Qwen2-VL-7B gradually to build a team from 2 VLMs to 4 VLMs. We average the re-evaluation score with 3 or more agents for the final prediction.

As shown in Table 16, both the three-VLM and four-VLM configurations outperform the two-VLM configuration, with the three-VLM configuration achieving the highest AUC. Specifically, the AUC increases from 88.32% with two VLMs to 88.81% with three VLMs, and remains high at 88.72%

Table 14: Oracle upper-bound analysis for suspicious segment identification. Headroom recovered is the fraction of the 3.20-AUC selection headroom (Oracle − Always-On) captured by each rule.

Method	Selection rule	UCF-Crime AUC (%)	Headroom recovered
Always-On	None (all flagged)	86.99	0% (baseline)
Random 50%	Random	87.59	19%
ARIA	Adaptive routing	**88.32**	**42%**
Oracle (upper bound)	Ground-truth error	90.19	100%

Table 15: Robustness analysis on UCF-Crime across complementary VLMs for COOPERATE, base VLM architectures, and VAD instruction prompts. Δ denotes the improvement over the corresponding base inference.

Configuration	**AUC (%)**	Δ
Complementary VLM for COOPERATE		
InternVL2-8B	86.46	+0.13
InternVL3-8B	87.11	+0.78
Qwen2-VL-7B	86.66	+0.33
InternVL2.5-8B	**87.57**	**+1.24**
Base VLM Architecture		
InternVL2-8B	87.57	+1.24
InternVL2.5-8B	87.71	+2.49
Qwen2-VL-7B	87.09	**+4.83**
InternVL3-8B	**88.32**	+4.59
VAD Instruction Prompt		
Prompt 1	**87.57**	+1.24
Prompt 2	86.45	**+3.16**
Prompt 3	86.06	+2.57

with four VLMs. These results show that collaborative scaling can provide additional performance gains, although the improvement is not monotonic as more VLMs are added.

A.5 QUALITATIVE AND EXPLANATION ANALYSIS

A.5.1 RELEVANT SCENES DEMONSTRATION

We further illustrate how ARIA retrieves relevant scenes that provide useful context for adaptive routing. As shown in Fig. 5, we take a scene from *Burglary076_x264* in UCF-Crime as an example. On the left, the retrieved temporal-relevance scenes are closely related to the target segment in time and capture the same evolving event: a person is walking from the car and attempting to break into a building, with slight changes in the temporal window. On the right, the retrieved semantic-relevance scenes are measured by appearance similarity. Although these scenes do not necessarily occur nearby in time, they share similar visual patterns with the target segment, such as a person appearing near the door.

By examining these temporally and semantically related scenes, we can observe whether the lightweight base response is supported by consistent contextual evidence. In ARIA, these relevant scenes are not used as a hard decision rule for suspicious-segment identification. Instead, they provide low-cost temporal and semantic context for the routing feature x_j, helping the adaptive router assess whether additional reasoning may be beneficial and, if so, which follow-up action is more appropriate.

A.5.2 CASE STUDIES OF ADAPTIVE REASONING PATHS

We present four case-study figures to qualitatively illustrate how ARIA selects an action based on the stage-1 benefit estimate and the action-specific corrective utilities.

- **Fig. 6: Correcting false negatives with different evidence sources.** In the left example, the lightweight base inference only observes a sparse glimpse of a person walking in an

Table 16: Effect of scaling up the number of VLMs in ARIA.

# VLMs	2	3	4
AUC (%)	88.32	**88.81**	88.72



Figure 5: We take "Burglary076_x264" from UCF-Crime as an example to showcase the relevant scenes obtained from ARIA via temporal and semantic relevance.

indoor corridor and predicts the segment as Normal. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{lm}} > p_{\mathrm{co}}$, ARIA routes the segment to LOOK-MORE. The additional frames reveal that the person leaves a backpack unattended and quickly walks away, which changes the prediction to Abnormal. In the right example, the lightweight base inference observes nighttime traffic and predicts the segment as Normal. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{co}} > p_{\mathrm{lm}}$, ARIA selects COOPERATE, where the complementary perspective highlights erratic driving together with smoke or sparks, correcting the prediction to Abnormal.

- **Fig. 7: Distinguishing suspicious behavior from routine activity.** In the left example, the lightweight base inference predicts the segment as Normal because the person appears to be simply walking through a parking lot. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{lm}} > p_{\mathrm{co}}$, ARIA routes the segment to LOOK-MORE. Denser visual evidence then reveals repeated checking of car door handles and leaning toward vehicle windows, supporting an Abnormal prediction. In the right example, the lightweight base inference over-interprets rapid motion between two people near stacked boxes and predicts Abnormal. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{co}} > p_{\mathrm{lm}}$, ARIA selects COOPERATE, whose complementary interpretation recognizes that the two people are jointly moving heavy boxes in a coordinated manner, correcting the prediction to Normal.

- **Fig. 8: Correcting false positives with additional reasoning.** In the left example, the lightweight base inference predicts Abnormal because the person appears unusually close to a row of parked bicycles. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{lm}} > p_{\mathrm{co}}$, ARIA selects LOOK-MORE. The additional frames reveal that the person is unlocking their own bicycle, adjusting the seat, and riding away, which is consistent with ordinary activity and changes the prediction to Normal. In the right example, the lightweight base inference predicts Abnormal because a person is crouching near packages at night. Since $p_{\mathrm{s1}} > \tau$ and $p_{\mathrm{co}} > p_{\mathrm{lm}}$, ARIA selects COOPERATE, whose complementary perspective identifies the person as a delivery worker scanning and placing a parcel at the correct door, correcting the prediction to Normal.

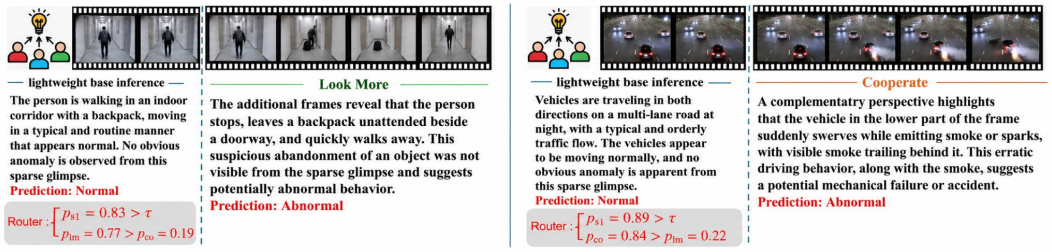

Figure 6: Case study 1: ARIA selects different follow-up actions according to the required evidence type. In the left example, LOOK-MORE reveals an unattended backpack beside a doorway and corrects the prediction from Normal to Abnormal. In the right example, COOPERATE provides a complementary interpretation that a vehicle is swerving while emitting smoke or sparks, correcting the prediction from Normal to Abnormal.

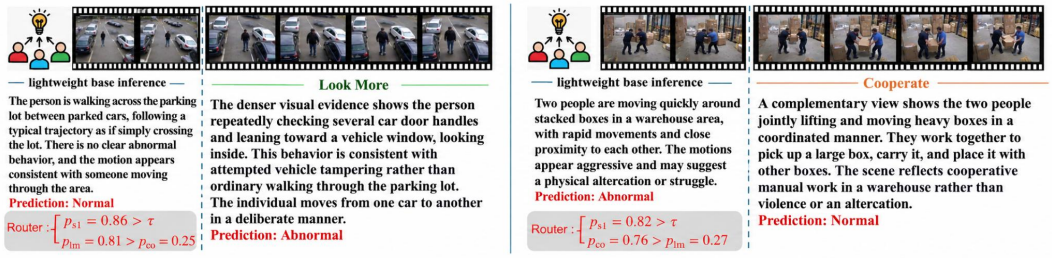

Figure 7: Case study 2: ARIA continues to select the appropriate reasoning path according to the evidence need. In the left example, LOOK-MORE reveals repeated checking of car door handles and vehicle windows, supporting an Abnormal prediction. In the right example, COOPERATE provides a complementary interpretation that two people are jointly moving boxes in a coordinated manner, correcting the lightweight false alarm and producing a Normal prediction.

- **Fig. 9: Retaining reliable lightweight predictions.** In this example, the lightweight base inference observes a person walking down a hallway in a typical manner and predicts the segment as Normal. The router assigns a low stage-1 probability, $p_{s1} < \tau$, indicating that further reasoning is unlikely to provide meaningful corrective utility. Therefore, ARIA selects the STOP action and retains the lightweight prediction directly, without invoking either LOOK-MORE or COOPERATE.

A.5.3 PER-CATEGORY REASONING QUALITY RESULTS

Table 17 reports the per-category breakdown of the LLM-as-Judge evaluation described in Table 2. For each label category, including Normal, 100 windows are randomly sampled (seed = 42) and evaluated on three dimensions: Reasonability, Novel Evidence, and Conciseness (1–10 each). The total score is the sum of the three dimensions (max = 30). A score-based win for ARIA requires its total score to be strictly higher than those of both Initial and Self-Reflection on the same segment.

Table 18 reports the overall reasoning-quality evaluation on XD-Violence via LLM-as-Judge (DeepSeek-R1-70B). For each of the seven label categories (Fighting, Shooting, Riot, Abuse, Car Accident, Explosion, and Normal), 100 windows are randomly sampled (seed = 42) and evaluated on the same three dimensions: Reasonability, Novel Evidence, and Conciseness (1–10 each). The total score is the sum of the three dimensions (max = 30). ARIA outperforms both Initial and Self-Reflection across all three dimensions and achieves a score-based win rate of 88.7%.

Table 19 reports the corresponding per-category breakdown. ARIA achieves higher total reasoning scores across all seven categories, with the highest score-based win rates on Normal (97.0%) and Fighting and Riot (92.0%), showing that the explanation-quality gains are observed across diverse anomaly types.

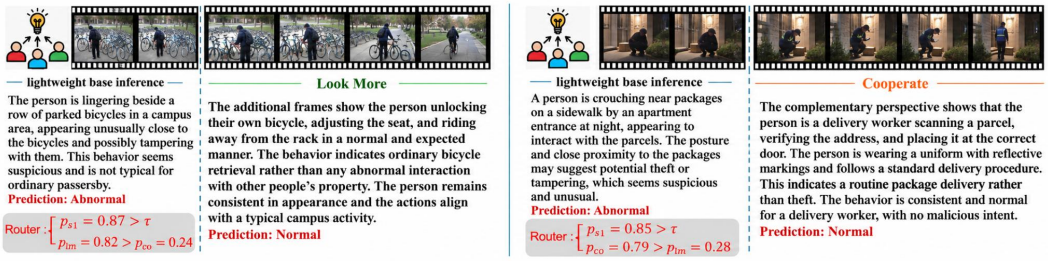

Figure 8: Case study 3: ARIA corrects false alarms through both LOOK-MORE and COOPERATE. In the left example, LOOK-MORE shows that the person is retrieving and riding their own bicycle, leading to a Normal prediction. In the right example, COOPERATE identifies the person near the packages as a delivery worker following a routine delivery procedure, also leading to a Normal prediction.

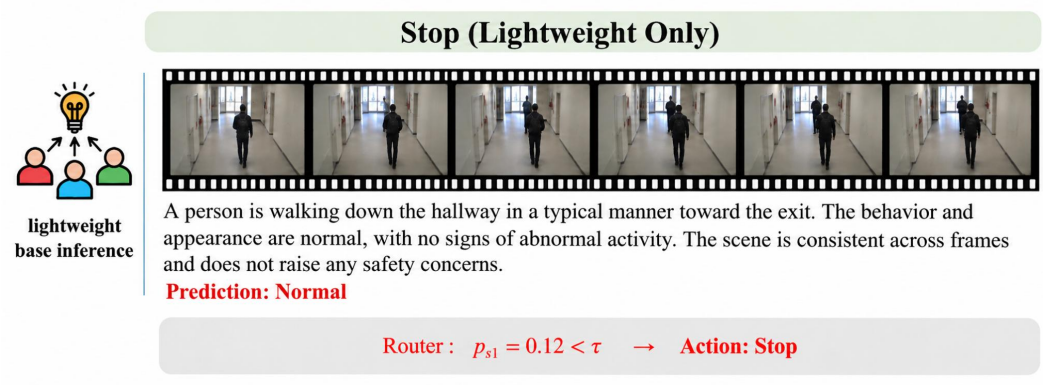

Figure 9: Case study 4: ARIA selects the STOP action when the lightweight base response is already sufficiently reliable. Since $p_{s1} < \tau$, no additional follow-up reasoning is triggered and the lightweight prediction is retained directly.

A.5.4 RELIABILITY OF THE LLM-AS-JUDGE PROTOCOL

Since the explanation-quality evaluation in Table 2 relies on an LLM judge, we further examine the reliability of this evaluation protocol from two complementary perspectives: (i) robustness across anomaly categories, judge models, prompt variants, decoding settings, and random seeds; and (ii) agreement with human preferences.

For these analyses, we adopt a three-way best-pick protocol with randomized presentation order. For each video segment, the explanations generated by Initial VERA, Self-Reflection via SRPO, and ARIA are randomly shuffled, and the judge directly selects one explanation as the winner. This differs from the score-based evaluation, which determines winners by comparing the total scores across the three evaluation dimensions. The two protocols use different decision rules, so their reported score-based win rates and best-pick preference rates are not directly comparable.

Robustness Across Anomaly Categories. Table 20 reports the per-category best-pick results on UCF-Crime using DeepSeek-R1-70B with greedy decoding and the original evaluation prompt. ARIA achieves the highest preference rate in all 14 label categories, with its preference rate ranging from 60.0% to 69.7%. The highest rates are observed for *Abuse* (69.7%), *Shoplifting* (68.3%), and *Normal* (68.3%), while the lowest are observed for *Fighting* and *Arson* (60.0%). Importantly, the improvement is not concentrated in a small subset of anomaly types: even in the categories with the lowest preference for ARIA, its preference rate remains substantially higher than those of Initial VERA and Self-Reflection. These results indicate that the explanation quality improvement of ARIA is consistent across diverse anomaly categories.

Table 17: Per-category reasoning quality scores (Reasonability + Novel Evidence + Conciseness, max = 30) evaluated by DeepSeek-R1-70B. ARIA is counted as a score-based winner when its total score is strictly higher than those of both Initial and Self-Reflection.

Category	n	Initial	Self-Reflection by SRPO	ARIA	ARIA win (%)
Abuse	100	17.58	18.88	21.75	85.0
Arrest	100	17.46	18.25	21.72	88.0
Arson	100	17.51	18.40	21.99	90.9
Assault	100	17.47	18.81	22.13	91.0
Burglary	100	17.43	18.23	22.03	93.0
Explosion	100	17.72	18.44	22.07	91.0
Fighting	100	17.44	18.22	22.03	89.0
Normal	100	17.87	18.32	22.27	93.9
RoadAccidents	100	17.51	18.30	22.10	92.9
Robbery	100	17.28	18.25	21.90	93.0
Shooting	100	17.67	18.49	21.99	90.0
Shoplifting	100	17.14	19.03	21.99	89.8
Stealing	100	17.56	18.70	21.56	84.9
Vandalism	100	17.61	18.41	22.01	90.9
Average	**1400**	**17.52**	**18.48**	**21.97**	**90.2**

Table 18: Reasoning quality evaluation via LLM-as-Judge (DeepSeek-R1-70B) across seven categories in XD-Violence. Each criterion is scored on a 1–10 scale, and the total score is the sum of the three criterion scores (max = 30). Scores are averaged over 100 randomly sampled windows per category. ARIA's Advantage is computed against Self-Reflection; the win-rate difference is expressed in percentage points (pp). A method is counted as a score-based winner when its total score is strictly higher than those of both other methods.

Metric	Initial	Self-Reflection by SRPO	ARIA	ARIA's Advantage
Reasonability	6.95	6.42	**7.99**	**+1.57**
Novel Evidence	3.97	3.96	**6.47**	**+2.51**
Conciseness	6.02	5.00	**6.98**	**+1.98**
Total	16.94	15.38	**21.44**	**+6.06**
Score-based Win Rate	0.1%	7.0%	**88.7%**	**+81.7 pp**

Robustness Across Judge Configurations. We further evaluate whether the best-pick results are sensitive to the choice of judge model, prompt wording, decoding strategy, or random seed. Specifically, we consider representative configurations involving two judge models (DeepSeek-R1-70B and Qwen2.5-72B), two prompt variants, greedy and temperature-based decoding, and three random seeds for the sampling-based setting.

As shown in Table 21, ARIA achieves the highest preference rate under every evaluated configuration, with its preference rate ranging from 67.0% to 71.4%. Under temperature-0.7 decoding, the variation across the three random seeds is only 3.0 percentage points. These results suggest that the explanation-quality advantage of ARIA is robust to changes in judge model, prompt wording, decoding strategy, and sampling randomness.

Validation Against Human Preferences. To further examine whether the LLM-as-Judge ranking aligns with human preferences, we conduct a blind three-way forced-choice study on a stratified subset of UCF-Crime segments. The evaluation subset is constructed to cover diverse anomaly categories and different judge-preferred outcomes. For each segment, the explanations generated by Initial VERA, Self-Reflection, and ARIA are normalized to reduce surface-format cues, randomly shuffled into anonymous positions, and presented to independent human annotators. Annotators select the explanation that best accounts for the observed video content based on reasoning coherence, visual grounding, and overall explanation quality.

As shown in Table 22, ARIA receives the highest human preference on the stratified evaluation subset, followed by Self-Reflection and Initial VERA. Importantly, this ordering is consistent with

Table 19: Per-category reasoning quality scores on XD-Violence (Reasonability + Novel Evidence + Conciseness, max = 30) evaluated by DeepSeek-R1-70B. ARIA is counted as a score-based winner when its total score is strictly higher than those of both Initial and Self-Reflection.

Category	n	**Initial**	**Self-Reflection by SRPO**	**ARIA**	**ARIA win (%)**
Abuse	100	16.99	16.04	20.96	78.0
CarAccident	100	16.86	15.85	21.10	84.0
Explosion	100	16.82	15.10	21.45	91.0
Fighting	100	16.94	15.52	21.58	92.0
Normal	100	16.98	14.43	21.81	97.0
Riot	100	16.99	15.32	21.70	92.0
Shooting	100	16.97	15.39	21.43	87.0
Average	**700**	**16.94**	**15.38**	**21.44**	**88.7**

Table 20: Per-category best-pick preference rates on UCF-Crime under the position-debiased protocol using DeepSeek-R1-70B with greedy decoding. ARIA achieves the highest preference rate across all 14 label categories.

Category	Initial (%)	Self-Reflection (%)	ARIA (%)
Abuse	13.3	17.0	**69.7**
Shoplifting	3.3	28.3	**68.3**
Normal	6.7	25.0	**68.3**
Arrest	6.9	26.0	**67.1**
Stealing	3.3	30.0	**66.7**
Robbery	3.3	30.0	**66.7**
Assault	6.7	27.7	**65.7**
Shooting	10.0	24.9	**65.1**
RoadAccidents	6.7	28.3	**65.0**
Burglary	6.7	30.0	**63.3**
Vandalism	6.7	30.0	**63.3**
Explosion	10.0	26.7	**63.3**
Fighting	20.0	20.0	**60.0**
Arson	16.7	23.3	**60.0**

Table 21: Robustness of the position-debiased best-pick evaluation across judge models, prompt variants, decoding settings, and random seeds. ARIA achieves the highest preference rate under all evaluated configurations.

Judge / Decoding / Prompt	Initial (%)	Self-Reflection (%)	ARIA (%)
DeepSeek-R1-70B / greedy / original	9.3	23.2	**67.5**
DeepSeek-R1-70B / greedy / variant A	8.7	23.5	**67.8**
Qwen2.5-72B / greedy / original	5.7	27.3	**67.0**
DeepSeek-R1-70B / temp=0.7 / seed 1	5.2	23.3	**71.4**
DeepSeek-R1-70B / temp=0.7 / seed 2	9.5	22.0	**68.4**
DeepSeek-R1-70B / temp=0.7 / seed 3	7.1	22.1	**70.8**

Table 22: Human validation of the LLM-as-Judge best-pick evaluation on a stratified subset of UCF-Crime. Explanations are anonymized and randomly shuffled before blind human evaluation.

Method	Human Preference (%)
Initial VERA	8
Self-Reflection	25
ARIA	**67**

the position-debiased LLM-as-Judge results. Across different judge configurations, the best-pick preference rate for ARIA ranges from 67.0% to 71.4% (Table 21).

The absolute rates obtained under the score-based and best-pick protocols should be interpreted separately because the protocols use different decision rules. Both evaluations favor ARIA under their respective protocols, and the human study yields the same aggregate method ordering.

Takeaway. Overall, the explanation-quality advantage of ARIA remains consistent across anomaly categories and a range of judge configurations. Moreover, the best-pick evaluation and the human study yield the same aggregate method ordering. These results provide additional evidence that the explanation-quality improvements reported in the main paper are not driven by a particular anomaly category, judge model, prompt wording, decoding setting, or sampling seed.