

# Observe Before You Alert: Adaptive Driver Alerting with Vision–Language Models

Anonymous Author(s)

Affiliation

Address

email

**Abstract:** Driver alerting from dashcam video is a sequential decision problem under partial observability: a system must warn early enough to support driver response, while avoiding frequent nuisance alerts. Existing approaches often reduce this problem to binary risk prediction, forcing a silent-or-alert decision even when visual evidence is uncertain. We propose VLALERT, a vision–language framework that formulates driver alerting as a tri-action policy over SILENT, OBSERVE, and ALERT. The OBSERVE action serves as an internal evidence-gathering state, allowing the system to defer uncertain warnings and adapt its subsequent observation window before committing an alert. VLALERT adapts a 4B VLM (Qwen3-VL) to generate structured safety-evidence spans and uses hidden states pooled from these spans as compact belief representations for danger estimation and policy prediction. To evaluate alerting beyond threshold-free anticipation metrics, we construct VLALERT-Bench, a unified per-tick benchmark from multiple driving-alert datasets, and introduce DAUS, a driver-aware utility score that jointly measures ranking, video coverage, false-alert burden, and lead-time utility. Experiments show that VLALERT achieves the best overall operating-point utility among representative video and VLM baselines. Ablations further indicate that structured belief-span pooling improves alert separability and that retaining OBSERVE as a separate action provides useful supervision for uncertain pre-alert states.

**Keywords:** Driver Alert System; Crash Prediction; VLM; Driving Safety

## 1 Introduction

Road traffic crashes remain a major safety concern. Many safety-critical events are associated with delayed hazard perception, driver distraction, or limited time for corrective action once a conflict becomes imminent [1, 2, 3, 4]. Dashcam-based driving alert systems aim to reduce this response gap: a camera mounted on the windshield or dashboard observes the vehicle’s first-person road scene and warns the driver before a hazardous situation becomes difficult to avoid [5, 6, 7]. Such warnings can support braking and steering in human driving, and takeover decisions when assisted driving is active [8, 9]. However, their value depends not only on detecting visual risk, but also on whether the warning is issued at the right time: early enough to support action, reliable enough to be trusted [10], and restrained enough to avoid unnecessary interruptions [11]. This suggests that driver alerting should be treated not only as visual risk recognition, but also as a temporally grounded decision problem that requires interpreting evolving traffic context and determining when the evidence is sufficient to warn [12, 13, 14, 15].

Recent work in driver alert system has gone beyond generic visual classification toward learned risk estimation from richer video and multimodal signals. Multimodal accident-understanding datasets and VLM-based analysis frameworks now use language to describe accident causes, relevant agents, prevention cues, and visual grounding [16, 17]. At the same time, video foundation and world-model representations have been adapted to ego-centric collision prediction, while self-supervised risk

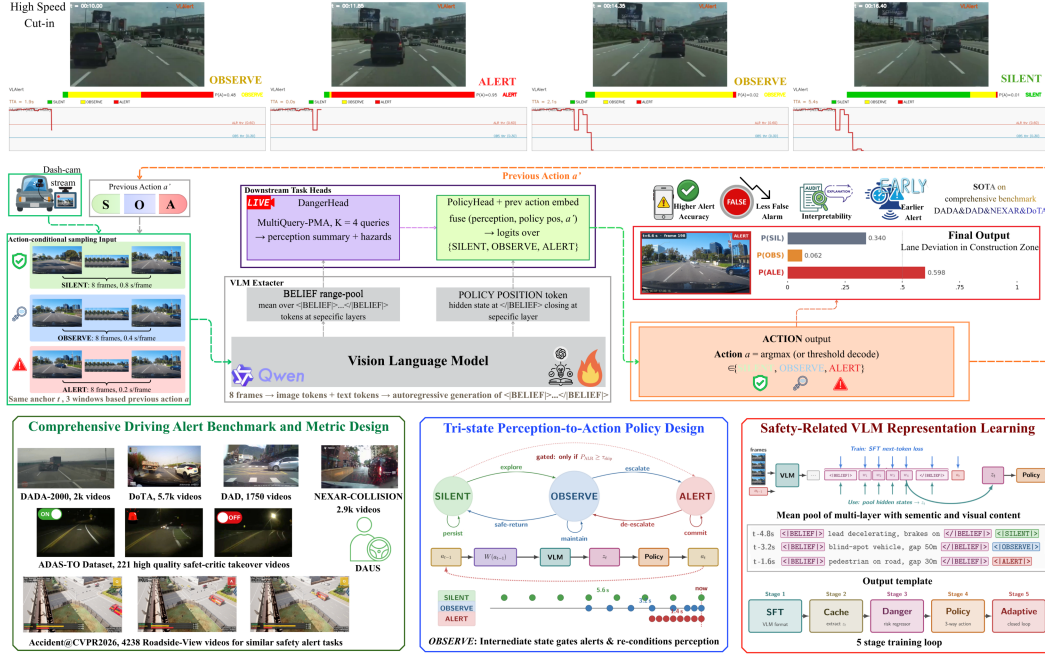


Figure 1: Overview of VLALERT.

objectives can impose temporal structure on the danger signal, such as the assumption that danger should generally increase as an accident approaches [18, 19, 20]. Complementary work on naturalistic driving further suggests that useful collision-risk signals can be learned from large-scale ordinary interactions without dense manual crash-risk labels [21]. Together, these developments make it increasingly plausible to train alert systems from visual, textual, and naturalistic evidence rather than relying only on manually designed surrogate thresholds.

These advances also leave several open questions for driver alerting. First, existing evaluations are still largely built around accident-centric datasets and ranking metrics, which are useful for anticipation but provide limited evidence about nuisance alerts, normal-driving behavior, and alignment with human-perceived risk [22, 23]. Second, most models reduce risk to a binary prediction or scalar score; thresholds can control sensitivity, but they do not provide an explicit mechanism for deferring a warning and gathering more evidence when the scene is ambiguous [24, 25]. These gaps motivate a unified alerting benchmark, an observation-aware alert policy, and a VLM interface that extracts alert-relevant representations.

To address these gaps, we propose VLALERT, a vision–language framework for adaptive driver alerting system. VLALERT adapts a VLM as a safety-evidence generator and uses the hidden states inside the generated evidence spans as compact features for danger estimation and policy prediction. Instead of reducing alerting to a binary decision or a thresholded risk score, VLALERT uses a tri-action policy over Silent, Observe, Alert, where Observe allows the system to gather more evidence before issuing an uncertain warning.

Our contributions are threefold. First, we build VLALERT-Bench, a unified per-tick benchmark that combines multiple driving-alert datasets and includes evaluation on real-world ADAS takeover clips to test whether accident-learned hazards align with human-perceived risk; we also introduce DAUS to jointly measure alert coverage, false-alert burden, and lead time. Second, we formulate alerting as an observation-aware policy over Silent, Observe, Alert, where Observe is supervised by risk-cue onset annotations and guides adaptive sampling before warning. Third, we adapt a VLM to generate structured safety-evidence spans and pool their hidden states as alert-relevant features, reducing dilution from generic scene description and improving downstream alert prediction.

## 67 2 Related Work

68 Early accident anticipation research is shaped by dashcam datasets such as DAD [26] and DADA-  
 69 2000 [27, 28], which focus on predicting whether and when a crash occurs. DoTA [29] extends this  
 70 setting to traffic anomaly detection with richer anomaly categories. More recent benchmarks move  
 71 beyond crash occurrence alone. For example, MM-AU [16] provides temporally aligned language  
 72 annotations, object boxes, accident reasons, and prevention descriptions for causal and semantic  
 73 accident understanding [16]. NEXAR-Collision focuses on ego-centric collision and near-collision  
 74 prediction, with alert-time annotations and AP-based evaluation across pre-event intervals [5].

75 A major line of work improves accident anticipation through specialized perception, attention,  
 76 and fusion designs. Recent works have used depth-enhanced 3D scene modeling [30], context-  
 77 aware and temporal-focus attention [31], LLM-assisted accident localization across when, where,  
 78 and what dimensions [13], and driver-attention-guided transformers for identifying risky traffic  
 79 participants [32]. A complementary direction is data-centric end-to-end collision prediction. BADAS  
 80 and Open-BADAS/BADAS-2.0 use V-JEPA2-style video foundation models and large ego-centric  
 81 dashcam corpora to predict threats involving the recording vehicle [19, 33, 18].

82 **Surrogate safety measures and naturalistic risk learning.** Classical driver warning systems often  
 83 rely on surrogate safety measures such as TTC, THW, post-encroachment time, or spacing-based  
 84 conflict indicators. These measures are interpretable and useful in structured settings, especially  
 85 car-following and lane-change scenarios, but they require hand-designed thresholds and may not  
 86 generalize across interaction types, scene layouts, weather, lighting, or occlusion. Recent work learns  
 87 more flexible conflict measures from trajectory data. For example, Jiao et al. model traffic conflicts  
 88 as context-dependent extreme events [34] and study missed and false alarms in vehicle-spacing-based  
 89 conflict detection [35]. GSSM further shows that large-scale naturalistic driving data can reveal  
 90 unsafe deviations even without crash labels [21].

## 91 3 Method

92 We present VLALERT, a vision–language framework for streaming driver alerting from ego-view  
 93 video. At each decision tick, VLALERT receives an 8-frame observation window and predicts a policy  
 94 over SILENT, OBSERVE, and ALERT. The method consists of four parts: a belief-state formulation,  
 95 a unified alert benchmark, a VLM-based alert architecture, and a closed-loop training procedure.

### 96 3.1 Problem Formulation

97 Driver alerting is a partially observed sequential decision problem. At tick  $t$ , the system observes only  
 98 an ego-view video history, while safety-critical factors such as agent intent, occluded objects, road  
 99 affordances, and driver response remain latent. We therefore use a POMDP-inspired formulation [36],  
 100 but do not explicitly model transition or observation probabilities. Instead, VLALERT learns an  
 101 amortized belief representation from video and trains a policy on top of it. The action space is

$$\mathcal{A} = \{SILENT, OBSERVE, ALERT\}. \quad (1)$$

102 SILENT suppresses warning. OBSERVE defers commitment and requests more evidence. ALERT  
 103 triggers a driver-facing warning. Unlike binary collision predictors, VLALERT treats warning as a  
 104 sequential policy decision under uncertainty. Let  $V_{\leq t}$  denote the video stream up to tick  $t$  and let  
 105  $a_{t-1}$  denote the previous action. A fixed temporal window forces all risk states to use the same  
 106 sampling scale. This is undesirable because long windows provide driving context and reduce false  
 107 alarms, while dense short windows preserve fast motion cues needed for imminent warnings. Recent  
 108 collision-prediction results show that window length changes precision, recall, and alert timing, since  
 109 overly short windows miss developing hazards and overly long windows include non-indicative  
 110 frames [19]. VLALERT resolves this trade-off with an action-conditioned sampler:

$$X_t = W(V_{\leq t}, a_{t-1}) = \{I_{t,1}, I_{t,2}, \dots, I_{t,8}\}. \quad (2)$$

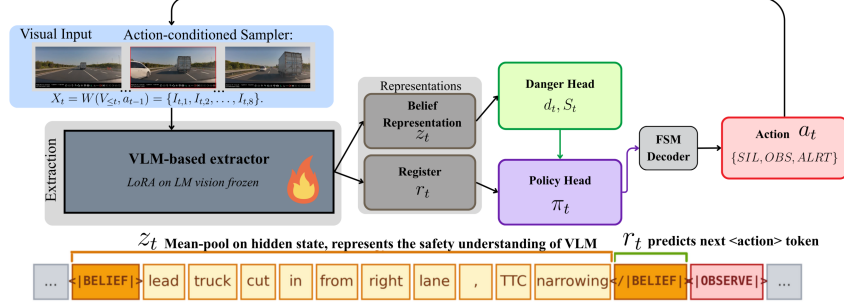


Figure 2: **Architecture of VLALERT.** VLALERT extracts belief representations from ego-view video with a VLM and predicts a tri-policy over  $\{SILENT, OBSERVE, ALERT\}$ .

111 The sampler uses a wide sparse window after SILENT, a medium dual-resolution window after  
 112 OBSERVE, and a dense short-range window after ALERT. Thus, the action does not merely emit  
 113 a decision; it also re-targets the next visual observation. Given  $X_t$ , VLALERT computes a learned  
 114 belief embedding and predicts the next action:

$$\hat{b}_t = \Phi_\theta(X_t), \quad \pi_t = \pi_\psi(a_t \mid \hat{b}_t, a_{t-1}). \quad (3)$$

115 Here  $\hat{b}_t$  is not an explicit posterior over latent driving states. It is a learned hidden-state representation  
 116 that summarizes task-relevant evidence under partial observability. This formulation gives OBSERVE  
 117 an operational role as an evidence-gathering action rather than an intermediate confidence bin.

### 118 3.2 Benchmark Construction

119 We construct **VLALERT-Bench**, a unified per-tick benchmark for driver-facing alert decisions.  
 120 The benchmark pools four real-world dashcam corpora, Nexar Collision [5], DoTA, DAD, and  
 121 DADA-2000, for in-domain training and validation. Two additional sources are held out for test-only  
 122 evaluation: ADAS-TO-Critic [37], which contains expert-reviewed human takeover events from  
 123 production ADAS logs, and ACCIDENT [38], a synthetic collision set for OOD testing.

124 All sources are converted to a common 1 Hz streaming format. Each tick contains an 8-frame obser-  
 125 vation window and one action label in  $\{SILENT, OBSERVE, ALERT\}$ . Let  $t_f$  denote the timestamp  
 126 of the tick anchor frame,  $t^*$  the event time (collision, anomaly peak, or takeover, depending on the  
 127 source), and  $t^\circ \leq t^*$  the first timestamp at which a source-specific risk-cue annotation marks the  
 128 developing hazard. For collision-style corpora, the tick action label is

$$y_f = \begin{cases} \text{ALERT}, & t^* - L_A \leq t_f < t^*, \\ \text{OBSERVE}, & t^\circ \leq t_f < t^* - L_A, \\ \text{SILENT}, & t_f < t^\circ, \\ \text{discard}, & t_f \geq t^*, \end{cases} \quad (4)$$

129 with  $L_A = 5$  s unless otherwise specified. The ALERT label therefore corresponds to the pre-event  
 130 interval in which a driver-facing warning is expected to be actionable. The OBSERVE label exposes the  
 131 earlier pre-event interval where a precursor cue is already legible, but the evidence is still insufficient  
 132 for an immediate warning. Ticks at or after  $t^*$  are discarded because they no longer represent a  
 133 predictive alerting problem. Negative clips are labelled SILENT throughout.

134 The risk-cue onset  $t^\circ$  is obtained with source-specific protocols: DoTA inherits its native anomaly  
 135 interval, DADA-2000 uses manually reviewed risk-cue onset annotations, Nexar uses an optical-  
 136 flow-based estimate, and DAD provides only clip-level labels with no usable  $t^\circ$ . Sources without a  
 137 usable  $t^\circ$  contribute zero OBSERVE ticks and split their usable pre-event interval between SILENT  
 138 and ALERT only. The full per-source protocol and the resulting class distribution are reported in  
 139 Appendix B.



| Source                 | Domain         | Train        |               | Val / Held-out |                     |
|------------------------|----------------|--------------|---------------|----------------|---------------------|
|                        |                | Clips        | Ticks         | Clips          | Ticks               |
| Nexar Collision        | dashcam        | 1,500        | 40,190        | 667            | 6,721               |
| DoTA                   | dashcam        | 2,949        | 29,763        | 326            | 3,256               |
| DAD                    | dashcam        | 1,157        | 4,628         | 127            | 508                 |
| DADA-2000              | dashcam        | 800          | 5,640         | 99             | 664                 |
| <b>In-domain total</b> | –              | <b>6,406</b> | <b>80,221</b> | <b>1,219</b>   | <b>11,149</b>       |
| ADAS-TO-Critic         | dashcam        | –            | –             | 221            | –                   |
| ACCIDENT               | synthetic CCTV | –            | –             | 2,211          | 17,224 <sup>†</sup> |

Table 1: **VLALERT-Bench composition.** In-domain sources are split by video and converted to 1 Hz ticks with 8-frame observation windows. ADAS-TO-Critic and ACCIDENT are held out for evaluation. <sup>†</sup>Pre-accident ticks only.

### 3.3 Model Architecture

VLALERT contains a VLM belief extractor, a DangerHead, a PolicyHead, and an event-gated decoder. We instantiate the extractor with Qwen3-VL-4B [39]. The visual encoder and original backbone weights are frozen. The language layers are adapted with LoRA [40]. We add two belief delimiters and three action tokens:  $\langle \text{BELIEF} \rangle$ ,  $\langle / \text{BELIEF} \rangle$ ,  $\langle \text{SILENT} \rangle$ ,  $\langle \text{OBSERVE} \rangle$ ,  $\langle \text{ALERT} \rangle$ . For each input frame, the VLM is trained to emit a short belief span followed by a frame-level action token:

$$\langle \text{BELIEF} \rangle w_1, \dots, w_n \langle / \text{BELIEF} \rangle \langle \text{ACTION} \rangle.$$

The belief span contains concise safety evidence. The special tokens make the belief region deterministic, so hidden states can be extracted without parsing free-form text. Let  $h_j^{(\ell)}$  be the hidden state at token position  $j$  and layer  $\ell$ . For frame  $f$ , let  $o_f$  and  $c_f$  be the positions of  $\langle \text{B} \rangle$  and  $\langle / \text{B} \rangle$ . We obtain a frame-level belief by span-pooling over a small set of upper layers  $\mathcal{L}$ :

$$z_t^{(f)} = \left\|_{\ell \in \mathcal{L}} \left( \frac{1}{c_f - o_f - 1} \sum_{j=o_f+1}^{c_f-1} h_j^{(\ell)} \right) \right\|. \quad (5)$$

The sequence  $Z_t = \{z_t^{(1)}, \dots, z_t^{(8)}\}$  forms the visual belief state. We also read a decision register  $r_t^{(f)}$  from the closing belief token. This design is motivated by recent evidence that transformer hidden states can encode belief-like information in partially observed sequential processes [41]. DangerHead estimates visual risk from  $Z_t$ . It applies a shared frame-level MLP and aggregates the eight frame beliefs with learned-query attention. It outputs per-frame danger scores  $d_t^{(1:8)}$ , a clip-level perception summary  $S_t$ , and a clip-level danger score  $d_t^{\text{clip}}$ . PolicyHead predicts the alert action from the decision-register sequence, the perception summary, the per-frame danger scores, and the previous-action embedding. Formally, it computes

$$u_t = \left[ \text{Temp}(r_t^{(1:8)}) \| S_t \| d_t^{(1:8)} \| e(a_{t-1}) \right]. \quad (6)$$

$$\pi_t = \text{softmax}(g_\psi(u_t)). \quad (7)$$

During policy training, DangerHead is frozen. This keeps risk perception separate from alert decision-making. At deployment, a finite-state decoder converts  $\pi_t$  into a tick-level action. A direct  $\text{SILENT} \rightarrow \text{ALERT}$  transition is allowed only when  $p(\text{ALERT})$  exceeds a high-confidence threshold. A separate event-gating module then converts the dense tick stream into sparse driver-facing alerts.

### 3.4 Training and Inference

VLALERT is trained in three stages. First, we perform belief-format supervised fine-tuning. The VLM is trained with next-token cross-entropy over the assistant response so that it emits eight belief spans and eight frame-level action tokens. Only LoRA parameters and the new special-token embeddings

are updated. This stage creates a reproducible hidden-state interface for downstream alerting. Second, we cache belief features from the fine-tuned VLM and train the downstream heads.

DangerHead is trained with frame-level and clip-level binary cross-entropy using the continuous danger target. After DangerHead is trained, it is frozen. PolicyHead is then trained with three-way cross-entropy over SILENT, OBSERVE, and ALERT. A transition regularizer discourages low-confidence jumps from SILENT directly to ALERT. Third, we perform closed-loop refinement. The action-conditioned sampler is activated during training. The previous action is first teacher-forced, then mixed with the model prediction, and finally taken from the model rollout. This curriculum exposes PolicyHead to the same adaptive sampling distribution used at inference time. At test time, VLALERT repeats a closed loop. It samples an 8-frame window from  $W(V_{\leq t}, a_{t-1})$ . It extracts belief states with the VLM. It estimates danger with DangerHead. It predicts  $\pi_t$  with PolicyHead. It decodes the tick-level action with the finite-state decoder. It emits sparse driver-facing alerts through the event-gating module. The decoded action is fed back as  $a_t$  for the next tick.

## 4 Experiments and Results

### 4.1 Evaluation Protocol

We evaluate all methods on the VLALERT-Bench validation split, containing 11,149 ticks, 1,219 videos, and 794 positive videos. Each method produces a scalar alert score per tick, and all threshold-dependent metrics are computed at one operating threshold  $\tau$  per method. We report tick-level ranking metrics, AUROC and  $AP_{\text{tick}}$ ; video-level coverage,  $\text{Recall}_v$ ; operating-point metrics,  $\text{Precision}_t$ ,  $F1_t$ , and balanced accuracy; and lead-time metrics,  $mTTA@2s$  and  $mTTA@4s$ . Because VLALERT-Bench is dominated by SILENT ticks, we use balanced accuracy rather than raw accuracy:

$$\text{BalAcc} = \frac{\text{TPR} + \text{TNR}}{2}.$$

This avoids rewarding constant-SILENT predictors and better reflects the deployed alerting trade-off.

### 4.2 Driver-Aware Utility Score

Standard accident-anticipation evaluation often reports  $mAP@TTA$  [42], which measures threshold-free temporal ranking across Time-To-Accident buckets. However, a deployed alerting system must operate at a fixed threshold, avoid nuisance alerts, cover dangerous videos, and warn early. We therefore report **DAUS** (Driver-Aware Utility Score), a geometric utility over ranking, coverage, precision, and timing:

$$\text{DAUS} = (MR_v P_t U_t)^{1/4}, \quad U_t = \min\left(\frac{mTTA}{L_{\text{alert}}}, 1\right), \quad (8)$$

where  $M = mAP@TTA$ ,  $R_v = \text{Recall}_v$ ,  $P_t = \text{Precision}_t$ , and  $U_t$  is normalized lead-time utility. The multiplicative form penalizes failure in any component, making DAUS sensitive to both missed hazards and disruptive false alerts.

### 4.3 Comparison with Baselines

We compare VLALERT with five representative baselines: ResNet50-LSTM [43], R3D-18 [44], [45], Open-BADAS [19], and Gemini-2.5-Flash [46]. More details are in Appendix D.

Table 2 reports the main comparison on VLALERT-Bench validation. VLALERT achieves the best overall operating-point performance, ranking first in AUROC,  $\text{Recall}_v$ ,  $F1_t$ ,  $AP_{\text{tick}}$ ,  $\text{Precision}_t$ , balanced accuracy, and DAUS. Compared with the strongest baseline, Open-BADAS, VLALERT improves AUROC from 0.610 to 0.689,  $AP_{\text{tick}}$  from 0.176 to 0.195, and DAUS from 0.4752 to 0.4878. This suggests that the proposed belief-state extraction and tri-action alert policy improve both tick-level discrimination and deployment-oriented decision quality beyond strong video representations.

| Method                | AUROC $\uparrow$ | Recall $\uparrow$ | F1 $\uparrow$ | AP $\uparrow$ | Prec $\uparrow$ | BalAcc $\uparrow$ | mTTA@2s $\uparrow$ | mTTA@4s $\uparrow$ | mAP@TTA $\uparrow$ | DAUS $\uparrow$ |
|-----------------------|------------------|-------------------|---------------|---------------|-----------------|-------------------|--------------------|--------------------|--------------------|-----------------|
| <b>VLALERT</b>        | <b>0.689</b>     | <b>0.884</b>      | <b>0.297</b>  | <b>0.195</b>  | <b>0.188</b>    | <b>0.648</b>      | <b>1.4</b>         | 3.0                | 0.503              | <b>0.4878</b>   |
| Open-BADAS            | 0.610            | 0.882             | 0.276         | 0.176         | 0.184           | 0.581             | 1.2                | 2.3                | 0.512              | 0.4752          |
| R3D-18                | 0.601            | 0.800             | 0.242         | 0.161         | 0.159           | 0.573             | 1.3                | 2.9                | 0.506              | 0.4559          |
| ResNet50-LSTM         | 0.581            | 0.800             | 0.218         | 0.147         | 0.138           | 0.541             | <b>1.4</b>         | 3.0                | 0.506              | 0.4439          |
| Gemini-2.5-Flash-Lite | 0.568            | 0.712             | 0.220         | 0.153         | 0.173           | 0.554             | 1.1                | 2.1                | 0.504              | 0.4344          |
| MViT-V2-S             | 0.566            | 0.750             | 0.206         | 0.148         | 0.129           | 0.523             | <b>1.4</b>         | <b>3.1</b>         | <b>0.530</b>       | 0.4329          |

Table 2: **Main results on VLALERT-Bench validation.** All threshold-dependent metrics in a row use the same operating threshold  $\tau$ .

Notably, MViT-V2-S obtains the highest mAP@TTA but the lowest DAUS, indicating that threshold-free temporal ranking alone can misrepresent driver-facing alert utility when video recall and tick-level precision at the deployment threshold are poor.



Figure 3: Performance comparison of VLALERT and Open-BADAS.

#### 4.4 Ablation Studies

We ablate the two components that distinguish VLALERT from a conventional binary VLM-based alerter: the structured BELIEF interface and the explicit OBSERVE action.

**BELIEF pooling.** We freeze the SFT-tuned Qwen3-VL-4B backbone and compare three ways of extracting hidden-state features from the same generated response. A 5-fold linear probe is trained for binary ALERT prediction on the VLALERT-Bench validation split. As shown in Table 3(a), pooling over the BELIEF span gives the strongest linear separability, with AP 0.459 and AUROC 0.894. Using only the opening BELIEF token remains competitive, while pooling a same-length random response span drops AP to 0.204. This supports the hypothesis that the structured BELIEF format localizes safety-relevant evidence in predictable hidden-state positions, rather than leaving it diffusely distributed across the generated text.

**OBSERVE supervision.** We retrain PolicyHead with three label spaces while keeping the backbone and DangerHead features fixed. Table 3(b) shows that the three-action formulation obtains the best

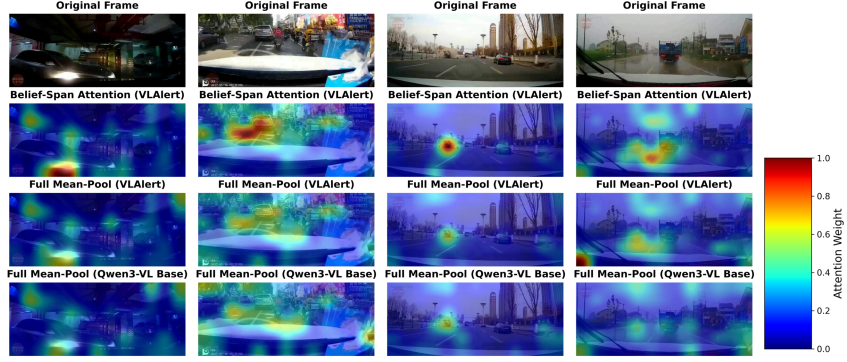


Figure 4: Comparing attention of the adaptive BELIEF-span and the default VLM output using mean-pool: Results show that BELIEF-span sees the whole scene with right focus on driving risk visual cues.

| (a) BELIEF pooling (linear probe, 5-fold CV) |               |                  |               | (b) OBSERVE action space (3 seeds, $F1^*$ -optimal $\tau$ ) |                            |                               |                            |
|----------------------------------------------|---------------|------------------|---------------|-------------------------------------------------------------|----------------------------|-------------------------------|----------------------------|
| Pooling strategy                             | AP $\uparrow$ | AUROC $\uparrow$ | F1 $\uparrow$ | Action space                                                | AP <sub>t</sub> $\uparrow$ | AUROC <sub>t</sub> $\uparrow$ | F1 <sub>t</sub> $\uparrow$ |
| <b>BELIEF range (ours)</b>                   | <b>0.459</b>  | <b>0.894</b>     | <b>0.442</b>  | <b>3-class S/O/A (ours)</b>                                 | <b>0.184</b>               | <b>0.734</b>                  | <b>0.256</b>               |
| BELIEF open-token only                       | 0.419         | 0.882            | 0.414         | Binary, OBSERVE→SILENT                                      | 0.175                      | 0.705                         | 0.255                      |
| random-span control                          | 0.204         | 0.760            | 0.222         | Binary, OBSERVE→ALERT                                       | 0.179                      | 0.726                         | 0.254                      |

Table 3: **Ablations on VLAAlert-Bench validation.** (a) BELIEF-aligned pooling yields substantially stronger alert separability than random-span pooling, indicating that the structured span localizes safety-relevant evidence in predictable hidden-state positions. (b) Keeping OBSERVE as a separate action improves tick-level ranking over collapsing it into either SILENT or ALERT, suggesting that OBSERVE provides useful supervision for uncertain pre-alert states.

224 AP<sub>t</sub>, AUROC<sub>t</sub>, and F1<sub>t</sub> across three seeds. It improves AP<sub>t</sub> by 5.4% and 3.1% over binary classifier,  
 225 indicating that OBSERVE provides a useful supervision signal for ranking uncertain pre-alert states  
 226 rather than acting as a redundant intermediate label.

227 The paper also validates the VLAAlert performance against two unseen datasets, ADAS-TO, which  
 228 tests if the model aligns with human takeover actions before crashes happen, and another synthesized  
 229 dataset in CARLA from a different roadside view (Appendix A).

## 230 5 Conclusion

231 We presented VLAAlert, a vision-language framework that reframes dashcam driver alerting as a  
 232 tri-action sequential policy over SILENT, OBSERVE, and ALERT, with OBSERVE serving as an  
 233 evidence-gathering deferment rather than an intermediate confidence bin. The framework consists  
 234 of three components: a unified per-tick benchmark, VLAAlert-Bench, paired with the DAUS metric  
 235 that jointly scores ranking, video coverage, precision, and lead-time utility; a tri-action policy that  
 236 adapts observation frequency; and a VLM belief-extraction interface that emits focused safety spans  
 237 to downstream heads. On VLAAlert-Bench, VLAAlert obtains the best overall operating-point utility  
 238 across all baselines. Our ablations further show that pooling hidden states over the BELIEF span  
 239 yields substantially stronger alert separability than pooling random spans of equal length, and that  
 240 retaining OBSERVE as a separate action improves tick-level ranking over collapsing it into either  
 241 SILENT or ALERT.

## References

- [1] W. H. Organization. *Global status report on road safety 2023: country and territory profiles*. World Health Organization, 2024.
- [2] National Center for Statistics and Analysis. Distracted driving in 2024. Research Note DOT HS 813 790, National Highway Traffic Safety Administration, Apr. 2026. URL <https://doi.org/10.21949/z7ma-yc97>.
- [3] Q. Qu, Y. Ali, Y. Shen, Q. Bao, and M. M. Haque. Examining collision avoidance behavior of distracted drivers: A correlated grouped random parameters accelerated failure time model with heterogeneity-in-means. *Accident Analysis & Prevention*, 215:108016, 2025.
- [4] Y. H. Omran, H. Sadeghi-Bazargani, M. H. Yarmohammadian, and G. Atighechian. Driving hazard perception tests: a systematic review. *Bulletin of Emergency & Trauma*, 11(2):51, 2023.
- [5] D. Moura, S. Zhu, and O. Zvitia. Nexar dashcam collision prediction dataset and challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2591, 2025.
- [6] W. Zhao, S. Gong, D. Zhao, F. Liu, N. Sze, and H. Huang. Effects of collision warning characteristics on driving behaviors and safety in connected vehicle environments. *Accident Analysis & Prevention*, 186:107053, 2023.
- [7] R. Yu, R. Zhang, H. Ai, L. Wang, and Z. Zou. Personalized driving assistance algorithms: Case study of federated learning based forward collision warning. *Accident Analysis & Prevention*, 168:106609, 2022.
- [8] G. Huang and B. J. Pitts. Takeover requests for automated driving: The effects of signal direction, lead time, and modality on takeover performance. *Accident Analysis & Prevention*, 165:106534, 2022.
- [9] X. Tan and Y. Zhang. The effects of takeover request lead time on drivers’ situation awareness for manually exiting from freeways: A web-based study on level 3 automated vehicles. *Accident Analysis & Prevention*, 168:106593, 2022.
- [10] A. Wang, J. Wang, C. Huang, D. He, and H. Yang. Exploring how physio-psychological states affect drivers’ takeover performance in conditional automated vehicles. *Accident Analysis & Prevention*, 216:108022, 2025. ISSN 0001-4575. doi:<https://doi.org/10.1016/j.aap.2025.108022>. URL <https://www.sciencedirect.com/science/article/pii/S0001457525001083>.
- [11] J. Zhang, Z. Zhang, T. Zhang, Y. Zhang, and S. Chen. Effects of time interval and request modality on driver takeover responses: Identifying the optimal time interval for two-stage warning system. *Accident Analysis & Prevention*, 215:108008, 2025. ISSN 0001-4575. doi:<https://doi.org/10.1016/j.aap.2025.108008>. URL <https://www.sciencedirect.com/science/article/pii/S0001457525000946>.
- [12] H. Liao, H. Sun, H. Shen, C. Wang, C. Tian, K. Tam, L. Li, C. Xu, and Z. Li. Crash: Crash recognition and anticipation system harnessing with context-aware and temporal focus attentions. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM ’24, page 11041–11050, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi:[10.1145/3664647.3680672](https://doi.org/10.1145/3664647.3680672). URL <https://doi.org/10.1145/3664647.3680672>.
- [13] H. Liao, Y. Li, C. Wang, Y. Guan, K. Tam, C. Tian, L. Li, C. Xu, and Z. Li. When, where, and what? a benchmark for accident anticipation and localization with large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8–17, 2024.



- [14] J. Zhang, H. Liao, Y. Xie, C. Wang, Y. Guan, B. Rao, and Z. Li. Eyes on the road, mind beyond vision: Context-aware multi-modal enhanced risk anticipation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12054–12063, 2025.
- [15] X. Liu, B. Rao, Y. Guan, C. Wang, H. Liao, J. Zhang, C. Lin, M. Zhu, and Z. Li. Predict and resist: Long-term accident anticipation under sensor noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 782–790, 2026.
- [16] J. Fang, L.-l. Li, J. Zhou, J. Xiao, H. Yu, C. Lv, J. Xue, and T.-S. Chua. Abductive ego-view accident video understanding for safe driving perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22030–22040, 2024. doi:10.1109/CVPR52733.2024.02080.
- [17] R. Zhang, B. Wang, J. Zhang, Z. Bian, C. Feng, and K. Ozbay. When language and vision meet road safety: leveraging multimodal large language models for video-based traffic accident analysis. *Accident Analysis & Prevention*, 219:108077, 2025.
- [18] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.
- [19] R. Goldshmidt, H. Scott, L. Niccolini, S. Zhu, D. Moura, and O. Zvitia. Badas: Context aware collision prediction using real-world dashcam data. *arXiv preprint arXiv:2510.14876*, 2025.
- [20] A. Pjetri, D. Abbondandolo, D. C. de Andrade, S. Caprasecca, F. Sambo, and A. D. Bagdanov. Self-supervised road accident anticipation with non-decreasing danger. In *European Conference on Computer Vision*, pages 65–79. Springer, 2024.
- [21] Y. Jiao, S. C. Calvert, S. van Cranenburgh, and H. van Lint. Learning collision risk proactively from naturalistic driving data at scale. *Nature Machine Intelligence*, pages 1–14, 2026.
- [22] T. Zhao, Y. Zou, Z. Mao, P. Xiao, Y. Huang, H. Yang, Y. Li, T. Li, G. Wu, and Y. Lin. Accident anticipation via temporal occurrence prediction. *Advances in Neural Information Processing Systems*, 38:9081–9100, 2026.
- [23] Z. Zhao, L. Pipkorn, B. Mehler, B. Reimer, and P. Gershon. Driver behavior in response to forward collision warnings considering driving context. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 68, pages 953–957. SAGE Publications Sage CA: Los Angeles, CA, 2024.
- [24] S. Mishra, M. Mishra, P. Shyam, and D. Har. Time-vad: Text-informed magnitude enhancement feature learning for vehicle accident detection and anticipation. *IEEE Transactions on Intelligent Transportation Systems*, 27(3):3127–3142, 2026. doi:10.1109/TITS.2025.3637912.
- [25] S. Xie, L. Kong, Y. Dong, C. Sima, W. Zhang, Q. A. Chen, Z. Liu, and L. Pan. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6585–6597, 2025.
- [26] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. In *Asian conference on computer vision*, pages 136–153. Springer, 2016.
- [27] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li. Dada-2000: Can driving accident be predicted by driver attention? analyzed by a benchmark. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4303–4309. IEEE, 2019.
- [28] J. Fang, D. Yan, J. Qiao, J. Xue, and H. Yu. Dada: Driver attention prediction in driving accident scenarios. *IEEE transactions on intelligent transportation systems*, 23(6):4959–4971, 2021.

- [29] Y. Yao, X. Wang, M. Xu, Z. Pu, Y. Wang, E. Atkins, and D. J. Crandall. Dota: Unsupervised detection of traffic anomaly in driving videos. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):444–459, 2022.
- [30] H. Liao, Y. Li, Z. Li, Z. Bian, J. Lee, Z. Cui, G. Zhang, and C. Xu. Real-time accident anticipation for autonomous driving through monocular depth-enhanced 3d modeling. *Accident Analysis & Prevention*, 207:107760, 2024.
- [31] H. Liao, H. Sun, H. Shen, C. Wang, C. Tian, K. Tam, L. Li, C. Xu, and Z. Li. Crash: Crash recognition and anticipation system harnessing with context-aware and temporal focus attentions. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11041–11050, 2024.
- [32] Y. Kumamoto, K. Ohtani, D. Suzuki, M. Yamataka, and K. Takeda. Aat-da: Accident anticipation transformer with driver attention. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1142–1151, 2025.
- [33] R. Goldshmidt, H. Scott, L. Niccolini, and H. Matzner. Beyond the beep: Scalable collision anticipation and real-time explainability with badas-2.0. *arXiv preprint arXiv:2604.05767*, 2026.
- [34] Y. Jiao, S. C. Calvert, S. van Cranenburgh, and H. van Lint. A unified probabilistic approach to traffic conflict detection. *Analytic Methods in Accident Research*, 45:100369, 2025.
- [35] Y. Jiao, S. C. Calvert, and H. Van Lint. Minimising missed and false alarms: a vehicle spacing based approach to conflict detection. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1982–1987. IEEE, 2024.
- [36] M. Lauri, D. Hsu, and J. Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2022.
- [37] Y. Wang, Y. Xu, J. Sun, and H. Zhou. Adas-to: A large-scale multimodal naturalistic dataset and empirical characterization of human takeovers during adas engagement. *arXiv preprint arXiv:2603.06986*, 2026.
- [38] L. Pícek, V. Čermák, M. Hanzl, and M. Čermák. ACCIDENT @ CVPR 2026. <https://kaggle.com/competitions/accident>, 2026. Kaggle competition.
- [39] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- [40] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- [41] E. S. Hu, K. Ahn, Q. Liu, H. Xu, M. Tomar, A. Langford, D. Jayaraman, A. Lamb, and J. Langford. The belief state transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=ThRMTCgpvvo>.
- [42] W. Bao, Q. Yu, and Y. Kong. Deep reinforced accident anticipation with visual explanation. In *International Conference on Computer Vision (ICCV)*, 2021.
- [43] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.
- [44] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi:10.1109/CVPR.2018.00675. URL <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00675>.

- 376 [45] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. Mvitv2:  
377 Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022.
- 378 [46] Google DeepMind. Gemini 2.5 Flash-Lite Model Card. Model card, Sept.  
379 2025. URL [https://storage.googleapis.com/deepmind-media/Model-Cards/](https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Lite-Model-Card.pdf)  
380 [Gemini-2-5-Flash-Lite-Model-Card.pdf](https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Lite-Model-Card.pdf). Published/updated September 26, 2025.  
381 Accessed May 29, 2026.

## A Validation in Unseen Datasets

### A.1 ADAS-TO-Critic: Generalization to Human Takeovers

| Method                     | F1↑          | R@10s↑       | R@5s↑        | Lead@10s↑   | Lead@5s↑    | DAUS↑        |
|----------------------------|--------------|--------------|--------------|-------------|-------------|--------------|
| <b>VLALERT (Ours)</b>      | <b>0.686</b> | <b>0.941</b> | <b>0.887</b> | 6.12        | <b>3.88</b> | <b>0.520</b> |
| Open-BADAS                 | 0.585        | 0.810        | 0.742        | 5.83        | 3.52        | 0.515        |
| R3D-18                     | 0.418        | 0.602        | 0.498        | <b>7.02</b> | 3.75        | 0.485        |
| MViT-V2-S                  | 0.355        | 0.588        | 0.498        | 6.65        | 3.74        | 0.495        |
| ResNet50-LSTM              | 0.385        | 0.579        | 0.498        | 6.66        | 3.74        | 0.479        |
| Gemini-2.5-Flash-Lite (zs) | 0.013        | 0.036        | 0.032        | 5.12        | 3.43        | 0.393        |

Table 4: **ADAS-TO-Critic evaluation.** The split contains 221 expert-reviewed takeover clips centered at  $t = 10$  s. R@10s/R@5s measure pre-takeover alert coverage; Lead@10s/Lead@5s report the mean first-alert lead time. *zs* denotes zero-shot prompting.

We further evaluate VLALERT on **ADAS-TO-Critic**, a held-out set of 221 expert-reviewed dashcam clips from production ADAS logs. Each clip is centered at a confirmed human takeover at  $t = 10$  s. This setting tests whether an alert policy trained on accident and anomaly data can transfer to naturalistic takeover events, where the driver’s intervention provides an external behavioral signal of perceived risk.

Table 4 reports the results. VLALERT achieves the best F1, R@10s, R@5s, Lead@5s, and DAUS. In particular, it detects 88.7% of takeover clips within the last 5 s before intervention, compared with 74.2% for Open-BADAS. The improvement in DAUS indicates that VLALERT not only fires more often before takeovers, but does so with better timing under the deployment utility criterion.



Figure 5: **Example prediction on ADAS-TO-Critic.** Top: selected frames with predicted action and class probabilities. Bottom: 1 Hz probability trace over time; the dashed line marks the human takeover at  $t = 10$  s.

Figure 5 shows a representative case. VLALERT remains in OBSERVE while risk is developing, then switches to ALERT one second before the recorded takeover and peaks at takeover time. This illustrates the intended OBSERVE→ALERT behavior: defer under uncertainty, then warn sharply when intervention becomes imminent.

### A.2 Out-of-Distribution Generalization on Roadside Accidents

We next evaluate whether VLALERT transfers beyond ego-view dashcam data. We use a roadside CARLA accident benchmark with 2,211 collision clips captured from fixed roadside cameras. This setting changes both viewpoint and scene statistics: the camera is static, the view is third-person, and the accident often unfolds across an intersection rather than from the ego vehicle’s perspective. Since all clips are positive, AP and AUROC are not meaningful; we instead report per-clip alert rate and

| Method                | Full         |             | Last 5 s     |             | Last 2 s     |             | DAUS <sub>+</sub> ↑ |
|-----------------------|--------------|-------------|--------------|-------------|--------------|-------------|---------------------|
|                       | Rate↑        | mTTA↑       | Rate↑        | mTTA↑       | Rate↑        | mTTA↑       |                     |
| <b>VLALERT (Ours)</b> | <b>88.9%</b> | <b>7.31</b> | <b>83.1%</b> | <b>4.18</b> | <b>74.8%</b> | <b>1.49</b> | <b>0.945</b>        |
| Gemini-2.5-Flash-Lite | 2.1%         | 5.59        | 1.5%         | 3.38        | 0.4%         | <b>1.50</b> | 0.510               |

Table 5: **Roadside CARLA accident evaluation.** Rate is the percentage of clips with at least one ALERT; mTTA is the mean first-alert lead time in seconds. DAUS<sub>+</sub> is the degenerate utility defined in Appendix F.

mean time-to-accident (mTTA) over the full pre-accident interval, the last 5 s, and the last 2 s before impact.

Table 5 shows that VLALERT transfers well to this unseen viewpoint. It alerts on 88.9% of clips over the full pre-accident window with a mean lead time of 7.31 s, and still fires on 74.8% of clips within the last 2 s before impact. In contrast, zero-shot Gemini-2.5-Flash-Lite rarely issues an alert under the same three-action protocol. Figure 6 gives a representative example: VLALERT escalates from OBSERVE to ALERT several seconds before impact, while Gemini remains silent.



Figure 6: Roadside CARLA accident example. A fixed roadside camera captures an unprotected left-turn collision. VLALERT escalates from OBSERVE to ALERT before impact and sustains the alert, while Gemini remains in SILENT.

## B Benchmark Construction Details

**Event-time extraction.** Each source is converted to a common tick-level stream using its native annotation format. For Nexar Collision, the event time  $t^*$  is read from the released metadata. For DoTA, we use the annotated anomaly interval  $[t_{\text{start}}, t_{\text{end}})$ . For DADA-2000,  $t^*$  is the accident time from the clip annotation. DAD does not provide reliable per-frame event timestamps, so its clip-level labels are propagated to all usable ticks. ADAS-TO-Critic and ACCIDENT are held out for evaluation.

**Tick generation.** For each clip, we sample a 1 Hz tick stream. A tick is anchored at time  $t_f$  and exposes the eight frames immediately preceding that anchor. The tick label is the action label assigned to its anchor frame under Eq. 4. For collision-style corpora, ticks after the alert window are discarded because they no longer test predictive alerting. Negative clips are labelled SILENT throughout.

**Label distribution.** Figure 7 reports the train and validation labels distributions after filtering. Nexar contributes most SILENT ticks, while DoTA contributes most ALERT ticks and supplies nearly all OBSERVE ticks. DAD is the smallest source, and DADA-2000 provides a denser mix of normal and alert ticks.

**Risk-cue onset and OBSERVE labels.** The base labelling rule in Eq. 4 assigns SILENT and ALERT labels from the event time. The OBSERVE label is added as an intermediate pre-event state.

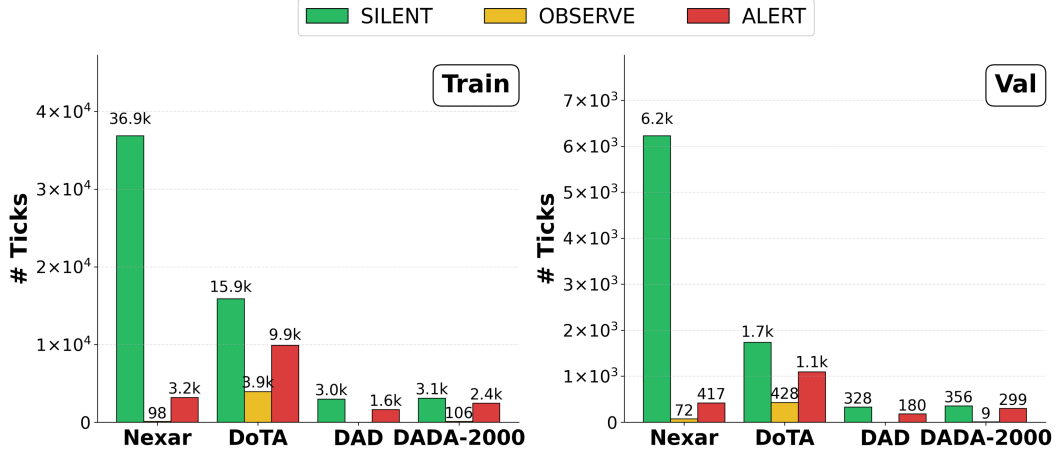


Figure 7: **VLALERT-Bench class distribution.** Per-source tick counts for train and validation after applying the label rule and post-event filtering. Bars show absolute tick counts for SILENT, OBSERVE, and ALERT.

For a clip with event time  $t^*$ , we define  $t^\circ \leq t^*$  as the first time at which the visual precursor of the event becomes legible. Ticks in  $[t^\circ, t^*)$  are relabelled as OBSERVE, replacing the SILENT label that would otherwise be assigned before the event. Because the four in-domain sources provide different levels of temporal annotation,  $t^\circ$  is obtained with source-specific rules.

- **DoTA.** DoTA provides frame-level anomaly annotations and anomaly categories. We use the first annotated anomaly frame as the risk-cue onset  $t^\circ$ , and use the contact or critical-proximity frame as  $t^*$  when available. This rule is applied automatically to all DoTA clips. As a result, DoTA supplies most of the OBSERVE ticks in VLALERT-Bench.
- **DADA-2000.** DADA-2000 provides accident-cause text and driver-attention maps. Two authors manually annotated  $t^\circ$  for all DADA-2000 clips used in our training and validation splits. We define  $t^\circ$  as the first frame where the at-risk actor becomes visible and matches the salient entity described by the accident-cause annotation. Annotators used a frame-level review interface with attention overlays. Agreement on a 100-clip audit subset was Cohen’s  $\kappa = 0.79$ . Disagreements were resolved by joint review.
- **Nexar Collision.** Nexar provides collision time but no anomaly interval or attention map. We therefore estimate  $t^\circ$  automatically from pre-event motion. For each clip, we compute dense optical-flow magnitude over a 1 s rolling window and mark  $t^\circ$  as the last pre-event frame where the maximum flow magnitude exceeds the clip-level 90th percentile. Clips without a clear pre-event flow peak receive no OBSERVE label and remain SILENT until the event.
- **DAD.** DAD provides clip-level binary labels but no reliable frame-level event time. We therefore do not assign risk-cue onset times for DAD. DAD contributes only SILENT and ALERT ticks.

**Annotation quality control.** The manual annotation steps were conducted by two authors with experience in traffic-safety data annotation. A third author reviewed a random 10% sample from each batch. The residual disagreement rates were 4.1% for DoTA cleanup and 6.3% for DADA-2000 risk-cue onset marking. All disagreements were resolved by joint adjudication. We did not use crowd workers, since identifying the first safety-relevant visual precursor requires domain knowledge and consistent temporal judgement.

The resulting OBSERVE distribution matches the annotation strength of each source. In the training split, the 4,152 OBSERVE ticks consist of 3,948 from DoTA, 106 from DADA-2000, 98 from Nexar, and 0 from DAD, as reflected in Figure 7.



## C VLALERT Training Details

This appendix describes the data pipeline, training stages, and inference protocol of VLALERT. All stages are run with bf16 mixed precision on a single NVIDIA RTX 5090 GPU with 32 GB memory. The full pipeline takes approximately 80 GPU-hours.

### C.1 Training-Data Construction

**Sample unit.** Each training sample follows the tick format of VLALERT-Bench: an 8-frame observation window anchored at time  $t_f$ , paired with an action label  $y \in \{SILENT, OBSERVE, ALERT\}$ . The tick label is derived from the event time  $t^*$  using the labelling rule in Eq. 4. For belief-format supervised fine-tuning, we also assign per-frame labels  $y_1, \dots, y_8$  by applying the same rule to each frame timestamp.

**Belief-span text.** Each frame is paired with a short safety-evidence span enclosed by the `<|BELIEF|>` and `</|BELIEF|>` delimiters. When a dataset provides native textual annotations, such as accident causes or attention regions, we reuse them as safety-evidence text. For datasets without rich text annotations, we use the base instruction-tuned Qwen3-VL-4B model to generate concise safety captions with the prompt:

Describe the safety-relevant evidence in this dashcam frame in no more than 20 words.  
Mention only hazards, agents, and traffic state.

Generated captions are filtered for length, generic content, and obvious failures. Failed captions are replaced with short category-based templates derived from the action label. For OBSERVE samples, risk-cue onset annotations are used to bias the belief text toward the developing hazard rather than generic scene description.

**Assistant response format.** For every 8-frame tick, the supervised target is an eight-line response:

```
<|BELIEF|> belief_1 </|BELIEF|> <action_1>
<|BELIEF|> belief_2 </|BELIEF|> <action_2>
...
<|BELIEF|> belief_8 </|BELIEF|> <action_8>
```

where `<action_i>` is one of `<SILENT>`, `<OBSERVE>`, or `<ALERT>`. This fixed format makes the belief boundaries deterministic and supports the span-pooling operation in Eq. 5.

**Prompt template.** The prompt used for belief-format training and feature extraction is:

```
[SYSTEM]
You are a driving-safety analyst. You will be shown eight frames
from a driving video. For each frame, identify the safety-relevant
evidence and decide the appropriate alert level for the driver.

[USER]
Frames: <img_1><img_2>...<img_8>
For each of the eight frames, output one line in the form
<|BELIEF|> evidence </|BELIEF|> <action>
where <action> is one of <SILENT>, <OBSERVE>, or <ALERT>.
```

We avoid specifying a fixed intra-window sampling rate in the prompt because the closed-loop sampler changes the temporal spacing of the eight frames according to the previous action.

**Final training set.** Applying this pipeline to the in-domain split of VLALERT-Bench yields 80,221 training ticks and 11,149 validation ticks. Each tick contains eight frames and the corresponding belief/action supervision.

| Source          | Clips        | Ticks         | Belief spans   | Belief origin                   | Example snippet                                       |
|-----------------|--------------|---------------|----------------|---------------------------------|-------------------------------------------------------|
| Nexar Collision | 1,500        | 40,190        | 321,520        | GPT-5.4 caption                 | <i>“lead truck brakes hard, gap closing fast”</i>     |
| DoTA            | 2,949        | 29,763        | 238,104        | GPT-5.4 caption + anomaly label | <i>“pedestrian crossing from right curb”</i>          |
| DAD             | 1,157        | 4,628         | 37,024         | GPT-5.4 caption                 | <i>“motorcycle weaving between lanes ahead”</i>       |
| DADA-2000       | 800          | 5,640         | 45,120         | Native cause + attention text   | <i>“rear-end risk; lead car decelerating sharply”</i> |
| <b>Total</b>    | <b>6,406</b> | <b>80,221</b> | <b>641,768</b> | –                               | –                                                     |

Table 6: **Belief supervision for Stage 1 SFT.** Each training tick provides eight per-frame belief spans. Native textual annotations are reused when available; otherwise, concise safety-evidence captions are generated offline with GPT-5.4 and filtered before training.

## 501 C.2 Stage 1: Belief-Format Supervised Fine-Tuning

502 **Trainable parameters.** We start from Qwen3-VL-4B-Instruct. The visual encoder and the original  
503 language-model backbone are frozen. LoRA adapters are attached to the linear projections in the  
504 language transformer blocks, including query, key, value, output, up, gate, and down projections.  
505 We add five new tokens to the tokenizer: `<|BELIEF|>`, `</|BELIEF|>`, `<SILENT>`, `<OBSERVE>`, and  
506 `<|ALERT|>`. Only the LoRA weights and the embeddings of the new tokens are trained. LoRA uses  
507 rank 64, scaling factor  $\alpha = 128$ , and dropout 0.05.

508 **Training-data composition.** Stage 1 uses the 80,221 training ticks from the in-domain VLALERT-  
509 Bench split. Each tick contains eight frames and therefore contributes eight supervised  
510 `<|BELIEF|> ... </|BELIEF|>` spans, yielding 641,768 belief spans in total. Table 6 summarizes  
511 the supervision sources. When native textual annotations are available, such as DADA-2000 accident-  
512 cause and attention text or DoTA anomaly categories, we reuse them as safety-evidence descriptions.  
513 For the remaining frames, we generate belief text offline with GPT-5.4 using the prompt:

514 Describe the safety-relevant evidence in this frame in no more than 20 words. Mention only  
515 hazards, agents, and traffic state.

516 The generated text is filtered for length, generic captions, and obvious failures. Residual failures  
517 are replaced with short class-templated fallbacks derived from the per-frame label  $y_i$ . No additional  
518 free-form manual captioning is used.

519 **Objective.** The model is trained with next-token cross-entropy over the assistant response only.  
520 System, user, and image tokens are masked out. Belief-text tokens use weight 1.0, while delimiter  
521 and action tokens use weight 2.0 to reduce collapse toward the majority SILENT label. Formally, for  
522 the target token sequence  $z_{1:T}$  and token weights  $w_t$ , the Stage 1 loss is

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{\sum_t w_t} \sum_{t=1}^T w_t \log p_{\theta}(z_t | z_{<t}, X),$$

523 where  $w_t = 0$  for masked prompt and image tokens,  $w_t = 1$  for belief-text tokens, and  $w_t = 2$  for  
524 delimiter and action tokens.

525 **Optimization.** We use AdamW with learning rate  $5 \times 10^{-5}$ , weight decay 0.05, cosine decay, 3%  
526 warm-up, gradient clipping at 1.0, and bf16 precision. The micro-batch size is one tick, with gradient  
527 accumulation over 16 steps, giving an effective batch size of 16 ticks. We train for three epochs with  
528 maximum sequence length 4096 and gradient checkpointing enabled. Stage 1 takes approximately 60  
529 GPU-hours.

### C.3 Stage 2: Belief Caching and Downstream Heads

**Belief cache.** After Stage 1, the LoRA-adapted VLM is frozen. We run it over all in-domain ticks and extract hidden states from each generated  $\langle |\text{BELIEF}| \rangle$  span. For each frame  $f \in \{1, \dots, 8\}$ , we span-pool the hidden states between  $\langle |\text{BELIEF}| \rangle$  and  $\langle /|\text{BELIEF}| \rangle$  and concatenate the last four transformer layers, following Eq. 5. This yields a 10,240-dimensional belief vector per frame. We also cache a decision register from the closing  $\langle /|\text{BELIEF}| \rangle$  token. Feature caching takes approximately 6 GPU-hours.

**DangerHead.** DangerHead estimates visual risk from the eight frame-level belief vectors. It first applies a shared two-layer MLP,  $10240 \rightarrow 1024 \rightarrow 512$ , with GELU, LayerNorm, and dropout 0.1. A learned-query attention pool aggregates the eight frame features into a clip-level summary, and linear heads output both per-frame danger scores and a clip-level danger score. The loss is the sum of per-frame and clip-level binary cross-entropy losses using the ALERT indicator as the target. We train with AdamW, learning rate  $10^{-3}$ , weight decay  $10^{-4}$ , batch size 256 ticks, and early stopping on validation AP. Training takes approximately 3 GPU-hours. DangerHead is frozen after this stage.

**PolicyHead.** PolicyHead predicts the three-way action distribution. Its input consists of the decision-register sequence, the clip-level perception summary, the per-frame danger scores, and a 16-dimensional embedding of the previous action. The temporal module is a two-layer Transformer with 4 heads and hidden dimension 256, followed by a three-way linear classifier. The objective is class-balanced cross-entropy over  $\{\text{SILENT}, \text{OBSERVE}, \text{ALERT}\}$ , plus a transition regularizer that discourages low-confidence direct jumps from SILENT to ALERT. We train with AdamW, learning rate  $5 \times 10^{-4}$ , batch size 128 ticks, and 40 epochs. Training takes approximately 3 GPU-hours.

### C.4 Stage 3: Closed-Loop Refinement

The final stage exposes PolicyHead to the same action-conditioned observation distribution used at inference time. The sampler  $W(V_{\leq t}, a_{t-1})$  chooses the next 8-frame window according to the previous action: a wide sparse window after SILENT, a medium dual-resolution window after OBSERVE, and a dense short-range window after ALERT. The previous action follows a three-phase curriculum:

1. teacher forcing in the first epoch, using the ground-truth previous action;
2. scheduled sampling in the second epoch, linearly mixing ground-truth and model-predicted previous actions;
3. full rollout in the final epoch, using the model’s own previous prediction.

The VLM backbone and DangerHead are frozen; only PolicyHead is updated. We use learning rate  $10^{-4}$  and the same transition regularizer as in Stage 2. This stage takes approximately 8 GPU-hours.

### C.5 Finite-State Decoder and Event Gating

The finite-state decoder and event-gating module are rule-based and contain no learned parameters. Their hyperparameters are calibrated once on the VLALERT-Bench validation split and then frozen for all evaluations.

**Finite-state decoder.** The decoder maintains the previous decoded state  $s_{t-1} \in \{\text{SILENT}, \text{OBSERVE}, \text{ALERT}\}$ , which is also passed to the action-conditioned sampler. At tick  $t$ , it receives the PolicyHead distribution  $\pi_t = (p_S, p_O, p_A)$  and outputs the current action  $a_t$ . If  $s_{t-1} = \text{SILENT}$ , a direct transition to ALERT is allowed only when  $p_A > \tau_{\text{jump}}$ . Otherwise, the decoder switches to OBSERVE when  $p_O + p_A > 0.5$  and remains in SILENT otherwise. If  $s_{t-1} \in \{\text{OBSERVE}, \text{ALERT}\}$ , the decoder simply takes  $\arg \max_a \pi_t(a)$ .

The only threshold in this decoder is  $\tau_{\text{jump}}$ . We sweep  $\tau_{\text{jump}} \in \{0.30, 0.35, \dots, 0.95\}$  on the validation split and select the value that maximizes DAUS under the same operating constraint used in

the main experiments. The selected value is  $\tau_{\text{jump}} = 0.60$ , which is used unchanged on all held-out datasets.

**Event gating.** The decoded tick-level actions are converted into sparse driver-facing alerts by a deterministic event gate. An alert event is emitted only after at least  $H = 2$  consecutive ALERT ticks, which suppresses isolated one-tick flickers. After an alert is emitted, the system enters a refractory window of  $R = 3$  seconds during which no new alert event can be emitted. The values  $H = 2$  and  $R = 3$  are selected by the same validation sweep used for  $\tau_{\text{jump}}$ .

**Deployment use.** The decoder and event gate add no trainable parameters and no training cost. The calibrated triple  $(\tau_{\text{jump}}, H, R) = (0.60, 2, 3)$  is fixed for all reported results, including ADAS-TO-Critic and ACCIDENT, with no dataset-specific re-tuning.

## C.6 Inference Protocol

At inference time, VLALERT runs at 1 Hz. Given the previous action  $a_{t-1}$ , the action-conditioned sampler selects an 8-frame window  $X_t$ . The VLM produces belief spans and action tokens, and the hidden states inside the belief spans are pooled into frame-level belief vectors. DangerHead estimates per-frame and clip-level risk. PolicyHead then predicts  $\pi_t$  over SILENT, OBSERVE, and ALERT. A finite-state decoder blocks low-confidence SILENT  $\rightarrow$  ALERT transitions unless  $p(\text{ALERT}) > 0.6$ . Finally, an event-gating module merges consecutive ALERT ticks into sparse driver-facing alerts. The measured latency is approximately 180 ms per tick on the RTX 5090, within the 1 Hz update budget.

## D Baseline Architectures and Training Details

All learned baselines are trained on the same VLALERT-Bench in-domain training split of 80,221 ticks from 6,406 clips and validated on the same 11,149-tick validation split. Training uses bf16 mixed precision on a single NVIDIA RTX 5090. For each learned baseline, we tune one operating threshold on the validation set using the same calibration protocol as VLALERT. Zero-shot baselines are not trained.

### D.1 ResNet50-LSTM

**Architecture.** A ResNet-50 image encoder is applied independently to each of the eight frames. The 2048-dimensional global-average-pooled feature from each frame is projected to 512 dimensions and passed to a single-layer LSTM with hidden size 512. The final hidden state is fed to a two-layer MLP,  $512 \rightarrow 256 \rightarrow 2$ , producing binary alert logits.

**Training.** Frames are resized to  $224 \times 224$  and normalized with ImageNet statistics. The ResNet-50 backbone is frozen during a two-epoch warm-up and then unfrozen. We optimize binary cross-entropy with AdamW. The learning rate is  $10^{-4}$  for the LSTM and classifier and  $10^{-5}$  for the ResNet backbone. Weight decay is  $10^{-4}$ . We use cosine decay, 5% warm-up, batch size 16 ticks, and class-balanced sampling. Training runs for 12 epochs, and the best checkpoint is selected by validation AP. Training takes approximately 18 GPU-hours.

### D.2 R3D-18

**Architecture.** We use a Kinetics-pretrained R3D-18 backbone. The eight frames in a tick are stacked into a short video clip and resized/cropped to  $112 \times 112$ . The global average-pooled 512-dimensional feature is passed to a linear binary classifier.

**Training.** The full backbone is trained from the first epoch. We use SGD with momentum 0.9, learning rate  $10^{-3}$ , weight decay  $5 \times 10^{-4}$ , cosine decay, 5% warm-up, and batch size 32 clips. Data

617 augmentation includes horizontal flips, brightness jitter, and small temporal jitter. Training runs  
 618 for 20 epochs, with the best checkpoint selected by validation AP. Training takes approximately 14  
 619 GPU-hours.

### 620 D.3 MViT-V2-S

621 **Architecture.** We use a Kinetics-pretrained MViT-V2-S backbone. Each 8-frame tick is resized  
 622 and center-cropped to  $224 \times 224$ . The pooled CLS feature is passed through LayerNorm and a linear  
 623 binary classifier.

624 **Training.** The full backbone is fine-tuned from epoch 0. We use AdamW with learning rate  $10^{-4}$ ,  
 625 weight decay  $5 \times 10^{-2}$ , layer-wise learning-rate decay 0.75, cosine decay, 10% warm-up, drop-path  
 626 rate 0.1, batch size 32 clips, and class-balanced sampling. Training runs for 25 epochs, and the best  
 627 checkpoint is selected by validation AP. Training takes approximately 22 GPU-hours.

### 628 D.4 Open-BADAS

629 **Architecture.** Open-BADAS uses a frozen V-JEPA2 ViT-L/16 video encoder. Each 8-frame tick is  
 630 resampled to 16 frames at  $256 \times 256$ . The encoder output is passed to a lightweight collision head  
 631 consisting of attention pooling followed by a two-layer MLP for binary collision logits.

632 **Training.** The V-JEPA2 backbone is frozen, and only the collision head is optimized. We use  
 633 AdamW with learning rate  $5 \times 10^{-4}$ , weight decay  $10^{-2}$ , cosine decay, 5% warm-up, batch size 16  
 634 ticks, and class-balanced sampling. Training runs for 30 epochs, with the best checkpoint selected by  
 635 validation AP. Training takes approximately 6 GPU-hours.

### 636 D.5 Gemini-2.5-Flash-Lite

637 **Setup.** Gemini-2.5-Flash-Lite is used as a zero-shot multimodal baseline. For each tick, the eight  
 638 frames are encoded as JPEG images and sent in one multimodal request. We use temperature 0 and a  
 639 100-token output cap. No fine-tuning or in-context examples are used.

640 **Prompt.** The prompt is:

641 You are a driving-safety classifier. You will see eight frames  
 642 from a vehicle-mounted dashcam. Decide the appropriate alert level  
 643 for the last frame:  
 644 SILENT - normal driving, no hazard developing  
 645 OBSERVE - risk developing, but more evidence is needed  
 646 ALERT - a collision or near-collision is imminent or in progress  
 647  
 648 Respond only with a JSON object:  
 649 {"action": "SILENT|OBSERVE|ALERT",  
 650 "p\_silent": 0..1,  
 651 "p\_observe": 0..1,  
 652 "p\_alert": 0..1}  
 653 where the three probabilities sum to 1.

654 For the roadside-CCTV evaluation, the first sentence is changed to:

655 You will see eight frames from a fixed roadside camera overlooking a road or intersection.

656 All other prompt content is unchanged. The predicted class is the argmax over the three probabilities,  
 657 and the scalar alert score is  $p_{ALERT}$ .

658 **Score-to-action conversion.** All non-VLALERT learned baselines, except Gemini-2.5-Flash-Lite,  
 659 output a scalar danger score  $s \in [0, 1]$  at each tick rather than an explicit alert action. To evaluate  
 660 them under the same threshold-dependent protocol as VLALERT, we convert each score into a

661 binary decision in  $\{SILENT, ALERT\}$  using a single global threshold. For each baseline, we  
 662 sweep  $\tau \in \{0.01, 0.02, \dots, 0.99\}$  on the VLALERT-Bench validation split and select the threshold  
 663  $\tau^*$  that maximizes balanced accuracy under the same video-recall constraint used for VLALERT.  
 664 The selected threshold is then frozen for all subsequent evaluations: a tick is labelled ALERT if  
 665  $s \geq \tau^*$  and SILENT otherwise. For held-out datasets such as ADAS-TO-Critic, we reuse the same  
 666  $\tau^*$  without re-tuning, so the reported results reflect the fixed operating point that would be used in  
 667 deployment.

## 668 E ADAS-TO-Critic Dataset Details

669 **Source and curation.** ADAS-TO-Critic is a held-out, test-only set built from the public ADAS-TO  
 670 collection. ADAS-TO provides 285 candidate critical dashcam clips collected during real production  
 671 ADAS operation. Each clip follows a fixed temporal structure: the vehicle is under L2 ADAS control  
 672 for the first 10 s, the human driver takes over at  $t = 10$  s, and the vehicle is manually driven for the  
 673 remaining 10 s. Two authors independently reviewed all 285 clips and removed cases where the  
 674 takeover was not primarily safety-motivated, such as comfort-driven overrides, construction-zone  
 675 protocol takeovers, or disengagements caused by known system limits rather than external hazards.  
 676 The remaining 221 clips form ADAS-TO-Critic. Inter-reviewer agreement on the keep/discard  
 677 decision is Cohen’s  $\kappa = 0.84$ ; disagreements were adjudicated by a third author. All retained clips  
 678 preserve the original takeover anchor at  $t^* = 10$  s.

679 **Why this split tests alerting.** ADAS-TO-Critic provides a behavioral reference for driver alerting.  
 680 The takeover at  $t = 10$  s is an external signal that the driver perceived the situation as requiring  
 681 immediate intervention. This makes the split different from accident-only evaluation: the target is  
 682 not collision occurrence itself, but whether the alert model fires before a human intervention that  
 683 was motivated by perceived safety risk. The split is test-only and shares no clips with the in-domain  
 684 training corpora described in Appendix B.

685 **Tick generation.** Each 20 s clip is converted into a 1 Hz tick stream. We keep ticks anchored at  
 686  $t_f \in \{1, 2, \dots, 12\}$  s, where each tick exposes the eight frames immediately preceding  $t_f$ . Ticks  
 687 after  $t_f > 12$  s are discarded because they fall inside the post-takeover manual-driving portion and  
 688 no longer test predictive alerting. Evaluation is binary at the clip level: a method either fires at least  
 689 one ALERT before takeover or it does not. No OBSERVE sub-label is used for scoring on this split.

690 **Evaluation metrics.** For each clip, we record the first ALERT prediction before takeover, denoted  
 691  $\tau_{\text{fire}}$  when it exists. We report:

- 692 • **R@10s:** the fraction of clips with at least one ALERT in  $[0, 10]$  s;
- 693 • **R@5s:** the fraction of clips with at least one ALERT in  $[5, 10]$  s;
- 694 • **Lead@10s:** the mean lead time  $10 - \tau_{\text{fire}}$  over clips that fire in  $[0, 10]$  s;
- 695 • **Lead@5s:** the same lead time, restricted to clips that fire in  $[5, 10]$  s.

696 F1 is computed from the binary fire/no-fire decision over the full pre-takeover window. No thresh-  
 697 old is tuned on ADAS-TO-Critic. All learned baselines use the operating threshold selected on  
 698 VLALERT-Bench validation, and VLALERT uses the fixed decoder and event-gating parameters  
 699 from Appendix C.5.

## 700 F CARLA Accident OOD Evaluation Details

701 **Benchmark composition.** The CARLA accident split is used only for out-of-distribution evaluation.  
 702 It contains 2,211 clips, each ending in a labelled collision with known accident time  $t^*$ . All clips are  
 703 captured by a static roadside camera at 20 fps. The accident labels cover five categories: rear-end,  
 704 head-on, sideswipe, t-bone, and single-vehicle. Compared with the in-domain dashcam data, this





Figure 8: VAlert performance on roadside Carla accident dataset.

split changes the viewpoint, field of view, camera motion, and accident geometry. Figure 8 shows the class distribution and an example failure case.

**Evaluation protocol.** All clips in this split are positive collision clips, so standard binary AP, AUROC, and the multiplicative DAUS in Eq. 8 are not well-defined for this evaluation. In particular, there are no negative clips to rank against or to estimate a false-alert burden, and clip-level precision becomes degenerate once a method fires. We therefore report positive-set alerting metrics:

- *Alert rate*: the fraction of clips with at least one ALERT inside the evaluation window;
- *mTTA*: the mean time-to-accident of the first ALERT, computed over clips that fire;
- *DAUS<sub>+</sub>*: a positive-only utility summarising coverage and lead time,

$$\text{DAUS}_+ = \frac{1}{2}U_+ + \frac{1}{2}(1 - U_-), \quad U_+ = \text{Rate}_{\text{full}} \cdot \min\left(\frac{\text{mTTA}_{\text{full}}}{5}, 1\right).$$

Here  $U_+$  is the product of full-window alert coverage and capped lead-time utility. Since this split contains no negative clips, we set  $U_- = 0$ . Thus,  $\text{DAUS}_+$  preserves the same  $[0, 1]$  range as Eq. 8 and is monotone in both alert coverage and lead time. However, it is not directly comparable to the validation DAUS in Section 4.3; it should be read only as a compact OOD summary of the per-window alert-rate and mTTA results.

We report three windows: the full pre-accident interval, the last 5 s before impact, and the last 2 s before impact. All decisions use argmax over the three-action output together with the finite-state decoder and event-gating parameters fixed on VAlert-Bench validation (Appendix C.5). No threshold is tuned on this split.

**Baseline.** We compare with Gemini-2.5-Flash-Lite under a zero-shot CCTV-aware prompt. Each tick is represented by eight consecutive frames, and the model is asked to choose among SILENT, OBSERVE, and ALERT. Gemini was evaluated on 937 clips due to scoring coverage; VLALERT was evaluated on all 2,211 clips.

**Per-type results.** Table 7 reports results by accident type. VLALERT performs best on sideswipe, rear-end, and head-on crashes, where the risk is visible through sustained closing or lateral motion. T-bone cases are harder because the crossing vehicle often enters the camera view shortly before impact. Single-vehicle crashes are the main failure mode: without a second actor, the useful signal is mainly vehicle kinematics, which is less represented in the current belief supervision.

| Method                | Type           | Clips        | Full/mTTA           | Last 5 s/mTTA      | Last 2 s/mTTA      | DAUS         |
|-----------------------|----------------|--------------|---------------------|--------------------|--------------------|--------------|
| VLALERT (Ours)        | rear-end       | 794          | <b>94.0 / 7.10</b>  | <b>87.0 / 4.25</b> | <b>78.2 / 1.49</b> | <b>0.970</b> |
|                       | head-on        | 588          | <b>93.5 / 6.72</b>  | <b>84.7 / 3.82</b> | <b>77.6 / 1.47</b> | <b>0.968</b> |
|                       | sideswipe      | 405          | <b>100.0 / 9.85</b> | <b>99.8 / 4.50</b> | <b>98.8 / 1.52</b> | <b>1.000</b> |
|                       | t-bone         | 358          | <b>69.8 / 5.17</b>  | <b>65.1 / 4.21</b> | <b>47.8 / 1.44</b> | <b>0.849</b> |
|                       | single-vehicle | 66           | <b>21.2 / 6.28</b>  | <b>16.7 / 3.92</b> | <b>7.6 / 1.34</b>  | <b>0.606</b> |
|                       | <b>Overall</b> | <b>2,211</b> | <b>88.9 / 7.31</b>  | <b>83.1 / 4.18</b> | <b>74.8 / 1.49</b> | <b>0.945</b> |
| Gemini-2.5-Flash-Lite | rear-end       | 794          | 1.9 / 5.75          | 1.0 / 2.22         | 0.6 / 1.75         | 0.510        |
|                       | head-on        | 588          | 1.6 / 7.41          | 0.8 / 4.28         | 0.0 / –            | 0.508        |
|                       | sideswipe      | 405          | 0.9 / 5.45          | 0.9 / 2.45         | 0.9 / 1.45         | 0.505        |
|                       | t-bone         | 358          | 5.3 / 3.91          | 5.3 / 3.63         | 0.8 / 1.05         | 0.521        |
|                       | single-vehicle | 66           | 0.0 / –             | 0.0 / –            | 0.0 / –            | 0.500        |
|                       | <b>Overall</b> | <b>937</b>   | <b>2.1 / 5.59</b>   | <b>1.5 / 3.38</b>  | <b>0.4 / 1.50</b>  | <b>0.510</b> |

Table 7: **CARLA accident OOD evaluation by accident type.** Each window reports **Rate / mTTA**, where Rate is per-clip alert coverage in percent and mTTA is the mean first-alert lead time in seconds. Dashes indicate that no ALERT was fired in that window.

**Limitations.** The main failure modes are single-vehicle loss-of-control and late-entry t-bone cases. The former lacks a second interacting actor, while the latter may provide less than two seconds of visible pre-impact evidence. Both failures suggest that future versions should include longer temporal context, explicit ego-motion or vehicle-kinematic belief fields, and a small amount of CCTV-style adaptation data.