Log What Matters: Safety-Aware Adaptive Logging for CAVs in Dynamic Conditions

Johora Akter Polin*, Yichen Luo*, Sumaiya†, Zheng Dong† and Sidi Lu*
*Department of Computer Science, William & Mary, Williamsburg, VA 23185, USA
†Department of Computer Sciences, Wayne State University, Detroit, MI 48202, USA
{japolin, yluo11, sidi}@wm.edu, {sum, dong}@wayne.edu

***Abstract*—Although machine learning (ML) models for connected and autonomous vehicles (CAVs) report steadily improving accuracy on curated benchmarks, these gains often fail to translate to onboard, in-field deployments. A primary reason is that the models are engineered around clean, information-dense inputs, while operational sensing data is imperfect and dominated by redundant or low-salience frames. This wastes scarce compute, creates pipeline backlog, and forces decisions to rely on stale perception and prediction outputs, degrading accuracy and increasing end-to-end latency. To bridge this gap, we propose *CAAL-VLM*, an adaptive logging pipeline that removes redundant and non-informative frames while preserving task-relevant content, thereby improving accuracy and timeliness. *CAAL-VLM* incorporates two designed reinforcement learning (RL) logging policies: one for camera data, which leverages SigLIP similarity and Qwen-VL semantics to retain safety-critical visual evidence while suppressing redundancy; and one for LiDAR data, where scene similarity and odometry stability guide adaptive logging while maintaining geometric fidelity. Building on adaptive logging, we further introduce a VLM-guided sensing module to maintain perception accuracy under challenging scenarios by adapting sensing decisions to scene context. Extensive experiments show that *CAAL-VLM* reduces I/O overhead by ∼27% and perception latency by ∼42%, while improving perception robustness under adverse conditions.**

***Index Terms*—Adaptive logging, vision-language model.**

I. Introduction

Connected and autonomous vehicles (**CAVs**) are safety-critical cyber-physical systems. Their driving stack is an machine-learning (**ML**) sensor-to-decision pipeline that maps high-rate multimodal data to perception and prediction, then to downstream driving decisions that directly govern safety.

Most CAV ML models are built around benchmark-conditioned assumptions of clean, high-quality sensor inputs, which break in the field where sensing is inherently noisy and imperfect. For example, LiDAR-only [1], [2], camera-only [3], [4], and LiDAR–camera fusion detectors [5], [6] perform well on KITTI [7], nuScenes [8], and Waymo [9], yet prior studies report up to 30–60% accuracy drops on uncurated in-field data [10]–[12], revealing a persistent benchmark-to-reality gap.

In practice, this benchmark-to-field mismatch often appears as pervasive redundancy in raw sensor streams. A Pandar64 LiDAR emits 10 point-cloud frames per second [13], about 230,000 points per frame [14], yet on highways many consecutive scans are near-duplicates and frequently contain no safety-critical objects (*e.g.*, pedestrians), with many points coming from low-value regions such as sky or distant static background. At scale, this yields over 5 TB of raw LiDAR data per day per sensor [14], much of the raw sensor stream carries limited task- or safety-relevant information.

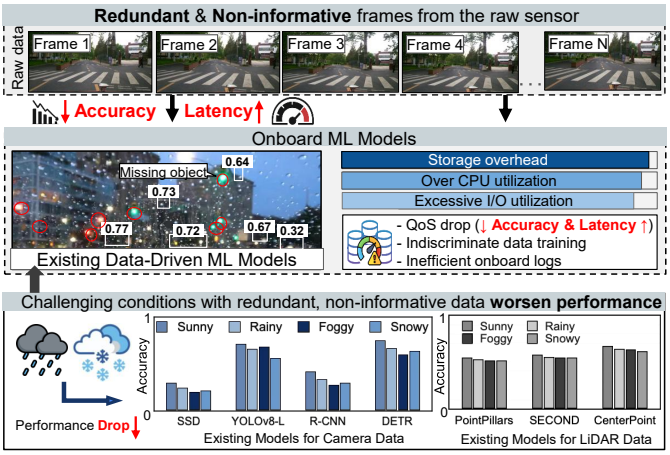

Fig. 1. The severity of redundant and non-informative sensor frames lies in their tendency to degrade perception accuracy (*e.g.*, missed detections), increase latency, and waste CPU, I/O, and storage resources. Their impact is further amplified in adverse environments such as rain, fog, and snow, where the performance of multiple camera- and LiDAR-based detection models deteriorates substantially.

These frames force the vehicle to act on outdated rather than current inputs, degrading ML accuracy (Figure 1). As real-time inference capacity is fundamentally limited, a model can process a fixed number of frames per second. When many frames are redundant or non-informative, they consume this limited budget and leave fewer timely opportunities for genuinely new evidence [15]–[17]. Thus, critical informative frames may not be processed promptly, delaying scene understanding and downstream decisions and increasing safety risk.

Moreover, redundant or non-informative frames can bias model training. Because learning is implicitly weighted by sample frequency, a dataset dominated by near-duplicate scenes repeatedly exposes the model to benign operating conditions (*e.g.*, clear-weather cruising). Prior work finds that many training samples contribute little new learning signal, whereas rare or challenging examples disproportionately shape the decision boundary [18]. Therefore, models trained on redundancy-dominated datasets degrade significantly under rare events (*e.g.*, adverse-weather visibility loss) [10], revealing weakened robustness when reliability is most critical.

Not only do redundant and non-informative frames degrade ML accuracy, but they also waste storage capacity and con-

sume scarce CPU and I/O resources. Given the hardware constraints of CAV platforms, every frame must still be received, copied, and passed through middleware queues such as the Data Distribution Service (DDS) in Robot Operating System 2 [19]–[21] before it can be processed or written to storage. As a result, these low-value frames compete for CPU cycles, cache, memory bandwidth, and I/O throughput on the vehicle's onboard computing platform [22]–[27], reducing headroom for safety-critical perception, planning, and control workloads. This contention underscores the need to suppress redundant frames before they enter the processing pipeline.

Collectively, these issues highlight the need to address redundant and non-informative frames in CAV ML pipelines. We therefore investigate the following research questions (**RQs**):

RQ_1: Can we design a value-aware mechanism that suppresses redundant or non-informative sensor frames in CAV, rather than logging all frames indiscriminately? **RQ_2:** (a) Can a value-aware logging mechanism reduce storage cost while preserving perception accuracy and end-to-end latency? (b) If so, is this robustness preserved under challenging scenarios (*e.g.*, adverse weather, complex scenes)?

Contributions: To suppress redundant and non-informative frames in ML perception pipelines, we propose ***CAAL-VLM*** (Context-Aware Adaptive Logging with Vision-Language Models), which improves storage efficiency while preserving accuracy and latency. The key contributions are as follows:

- Since redundant and non-informative frames degrade perception accuracy while wasting scarce CAV resources, we present *CAAL-VLM*, a unified, safety-aware adaptive logging pipeline for camera and LiDAR data in CAVs. The pipeline jointly reasons about (i) which frames are safety-critical or semantically valuable, (ii) whether a frame contributes new visual or geometric evidence beyond recent history, and (iii) how aggressively frames should be retained or suppressed given current conditions and resource headroom. A dedicated chain-of-thought (**CoT**) module converts scene semantics and sensor-integrity cues into structured context signals that feed the camera and LiDAR branches. Sensor-specific action policies then decide, for each frame, whether it should be kept (optionally at reduced fidelity) or safely dropped, enabling value-driven logging that removes non-informative data while preserving ML performance.
- For camera data, this logging pipeline that combines visual redundancy detection (**SigLIP**) with VLM-based semantic understanding (**Qwen-VL**), together with a reinforcement learning (**RL**) policy that selects the final logging action for each frame. Frames containing safety-critical events (*e.g.*, pedestrians, hazards, or traffic-rule changes) or substantial semantic shifts are always retained at full resolution. When no such cues exist, redundancy is determined by SigLIP similarity: highly similar frames are dropped, moderately similar frames are stored at reduced resolution, and dissimilar frames are preserved in full quality. This three-level retention strategy improves storage efficiency while maintaining safety-relevant visual evidence.
- For LiDAR data, we propose a three-part logging mechanism: (i) content-aware frame deduplication, which projects each scan into a Bird's Eye View (**BEV**) histogram and uses Chi-square distance to suppress near-identical scans over time; (ii) adaptive threshold selection, where an RL policy tunes the similarity threshold based on downstream LiDAR-odometry stability, ensuring that informative geometric change is preserved; and (iii) performance-aware retention, which keeps scans that improve localization reliability while discarding redundant ones to reduce storage and compute overhead. Together, these modules remove temporal redundancy while preserving the geometric fidelity required for accurate mapping and localization.
- Extensive evaluation across storage efficiency, Quality of Service (**QoS**), and perception robustness shows that *CAAL-VLM* reduces camera storage by $\sim$23% and LiDAR storage by $\sim$27%, while lowering camera peak bandwidth by $\sim$28%. Under the same logging budget, *CAAL-VLM* further reduces QoS violations by $\sim$13% and lowers perception latency by $\sim$42%. It also improves recall by $\sim$37%, indicating markedly stronger preservation of object-level evidence. Together, these gains reduce CPU and I/O pressure and improve runtime stability at full throughput.
- Building on *CAAL-VLM*'s value-aware logging design, we further introduce a VLM-guided, context-aware sensing module that converts scene semantics and sensor-integrity cues into transparent, verifiable sensing decisions. It performs (i) scene context extraction, (ii) modality-specific reliability scoring, and (iii) context-aware selection across camera, LiDAR, and fusion. This design enhances auditability and further improves perception accuracy and robustness in adverse visibility (*e.g.*, rainy-weather mAP improves from 0.75 to 0.86, a $\sim$1.15$\times$ relative gain over YOLOv10). We also provide an in-depth discussion of our experimental results and trends.

II. Background and Design Rationale

Maintaining onboard logging that is both resource-efficient and safety-relevant under adverse conditions requires decision mechanisms that are resource-aware and semantically interpretable [28]. While low-latency redundancy cues (pixel similarity, motion magnitude, point-count statistics) can suppress non-informative data, they often cannot explain *why* a scene is challenging (*e.g.*, intersection occlusions) or *which* frames are safety-relevant (*e.g.*, pedestrians near crosswalks) [29], [30]. This motivates semantic, scene-aware visual logging based on vision-language models (**VLMs**), which convert raw observations into structured context for adaptive control that balances logging efficiency with perception reliability [31].

Vision–Language Models for Contextual Scene Representation. VLMs map visual observations into a joint vision-text space [32], enabling the extraction of scene-level semantics that low-level similarity metrics cannot capture [33]. Rather than relying solely on pixel- or geometry-level differences, VLMs align images with linguistic concepts and produce structured descriptors of global context (*e.g.*,

weather/visibility, scene type such as highway or intersection, and coarse occlusion cues). We adopt VLMs for safety-critical logging as they provide rich, auditable semantics beyond low-level features, capturing visibility degradation, scene layout, and risk-relevant entities across conditions [34]–[37].

Semantic Context Extraction with Qwen-VL. Among available VLMs, we adopt Qwen-VL for its strong vision-text understanding, robust grounding, and instruction-following behavior [38]. In our pipeline, Qwen-VL acts as a context extractor that outputs structured semantics of the operating environment, including (i) high-level conditions (*e.g.*, foggy/rainy/snowy, day/night), (ii) scene type (*e.g.*, intersection/highway), and (iii) risk cues such as occlusions and vulnerable road users (**VRUs**). These signals complement visual similarity by abstracting pixel variation into scene-level context, enabling difficulty and risk reasoning.

Lightweight Redundancy Detection with SigLIP. While Qwen-VL provides semantic understanding of what is present in a scene, efficient logging also requires estimating how much new visual information a frame contributes relative to recent observations. In real driving sequences, semantic attributes often remain unchanged over long intervals even as frames arrive at a high rate, making per-frame VLM inference impractical on resource-constrained platforms. SigLIP [39] complements Qwen-VL with a lightweight embedding-based similarity score: cosine similarity filters near-duplicate frames, while VLM reasoning is applied only to likely novel frames.

Adaptive Policy Control via Reinforcement Learning. Although visual similarity and semantic context provide useful signals, optimal logging depends on time-varying conditions such as vehicle motion, sensor noise, weather severity, and available compute/I/O. Thus, fixed similarity or compression thresholds cannot consistently balance storage efficiency with downstream perception fidelity. Reinforcement learning (RL) is well suited to this setting because it frames logging as a sequential control problem, enabling adaptive parameter updates from observed system behavior rather than static heuristics. Actions tune retention policies, and rewards trade resource use against perception performance, enabling online, architecture-agnostic adaptation without per-condition tuning.

III. Adaptive Logging CAAL-VLM Pipeline

To address RQ_1, we present *CAAL-VLM*, which reduces redundant and non-informative data (as shown in Figure 2).

A. Safety-Aware Adaptive Logging for Camera

This section presents a redundant and non-informative frame removal pipeline combining two vision-language models: SigLIP for visual similarity estimation and Qwen-VL for semantic scene understanding. Each frame is encoded into visual and semantic embeddings to capture redundancy, contextual changes, and safety-critical events, and these signals decide whether to keep, discard, or store the frame at reduced resolution for efficient yet safety-preserving data handling.

1) ***SigLIP Embedding Extraction:*** SigLIP serves as the visual redundancy detector in the deduplication pipeline by measuring how visually similar the current frame I_t is to a previously stored key frame. For each incoming image, SigLIP computes a high-dimensional embedding $\mathbf{v}_t = f_{\text{SigLIP}}(I_t)$, where $\mathbf{v}_t \in \mathbb{R}^D$ captures the frame's global appearance. To make the comparison consistent, each embedding is L2-normalized (unit length) as $\hat{\mathbf{v}}_t = \mathbf{v}_t / \|\mathbf{v}_t\|$. Cosine similarity measures visual overlap between consecutive frames, and the corresponding distance decreases as redundancy increases.

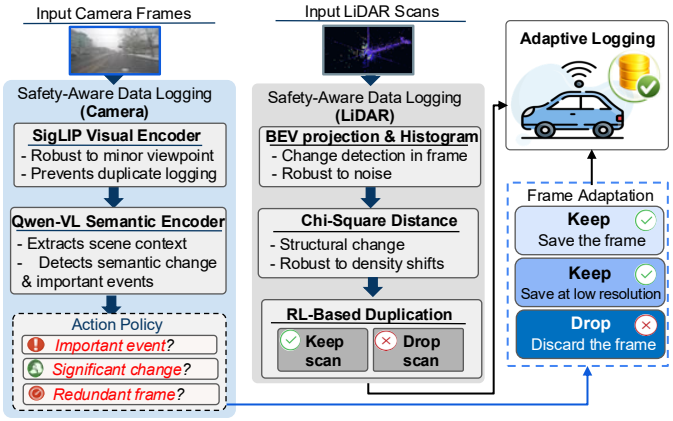

Fig. 2. Overview of the proposed *CAAL-VLM* safety-aware adaptive logging pipeline. Camera frames are processed by SigLIP for visual redundancy scoring and by Qwen-VL for semantic context and metadata for scene understanding (e.g., weather, time, critical objects such as pedestrians, cycles), then a prioritized policy based on important event, significant semantic change, and redundancy decides to KEEP the frame, KEEP the frame in low-resolution, or DROP the frame. In parallel, LiDAR scans are converted to BEV histograms and compared using dynamic Chi-square distance, with the RL-based adaptive deduplication module deciding to KEEP or DROP scans.

a) ***Semantic Relevance:*** While SigLIP captures appearance-level overlap, visual similarity alone can miss safety-critical changes (*e.g.*, a pedestrian entering view or a signal change). To prevent this, Qwen-VL acts as a semantic gate that checks for meaningful contextual change; when it raises no important-event flag and the semantic-change score is negligible, the redundancy decision is delegated to SigLIP. Here, an *important event* is a Qwen-VL safety/task indicator (*e.g.*, hazards, traffic-rule changes, near-collisions), and a *critical object* is a high-impact entity (*e.g.*, pedestrian, cyclist) whose state or motion can trigger the flag or increase the semantic-change score. When such semantic cues are absent, redundancy is decided purely by SigLIP similarity.

b) ***Similarity-Based Deduplication:*** Under this semantic-negligible condition, we apply a visual-similarity cutoff of 0.92 to identify frames that are visually indistinguishable from the previously kept frame. This choice reflects the empirical observation that SigLIP cosine similarities above 0.9 typically indicate near-duplicate frames in driving datasets, where camera motion is smooth and scene structure changes gradually. Setting the cutoff slightly above this point provides a safety margin that avoids dropping subtle but meaningful differences while still removing redundant frames [40]. Thus, frames with greater visual similarity are treated as duplicates and dropped unless Qwen-VL indicates semantic relevance or a safety-critical event occurs.

2) ***Qwen-VL Semantic Extraction:*** Qwen-VL provides high-level semantic understanding of each frame, complement-

ing SigLIP by separating visually similar frames that may still contain meaningful or safety-critical changes. For an input frame I_t, Qwen-VL produces visual tokens $\{\mathbf{h}_t^{(1)}, \ldots, \mathbf{h}_t^{(K)}\}$, projects them into the multimodal language space, and aggregates them into a semantic embedding via mean pooling, $\mathbf{u}_t = \frac{1}{K}\sum_{k=1}^{K} \mathbf{h}_t^{(k)}$. It also outputs structured metadata (time of day, weather, detected objects, the Boolean *important_event* flag) and a natural-language summary, capturing cues beyond raw pixels to avoid dropping semantically important frames.

*a) **Semantic-Change Scoring**:* To quantify contextual variation, a semantic-change score is computed by comparing metadata with the last key frame. This score increases when objects appear, disappear, or when environmental or contextual conditions shift. Semantic similarity between adjacent frames is also estimated from pooled embeddings via cosine similarity, allowing the system to capture subtle scene changes.

*b) **Thresholding and Semantic Gating**:* The semantic-change threshold separates minor fluctuations from meaningful context shifts. Qwen-VL metadata differences typically form two regimes: low scores (< 0.3) indicate negligible change, while high scores (> 0.6) reflect substantive events (object/traffic/environment changes). We set a threshold between these regimes so only significant semantic transitions trigger KEEP; otherwise, SigLIP stores moderate redundancy as low-res and drops near-duplicate frames.

*3) **Visual–Semantic Fusion Strategy for Adaptive Log**:* The final deduplication action is obtained by fusing SigLIP visual similarity, Qwen-VL semantic signals, and the PPO policy [41]. Each frame receives an action $a_t \in \{\text{KEEP, KEEP (Low-Resulation), DROP}\}$, selected from the state vector $s_t = [d_t^{(\text{vis})}, sc_t, important\_event_t]$, where $d_t^{(\text{vis})} = 1 - s_t^{(\text{vis})}$ denotes the SigLIP visual distance. The decision policy follows three key rules:

- **Safety override:** If a safety-relevant event is detected If Qwen-VL detects a safety-relevant *important event*, the frame is always KEEP the frame regardless of similarity scores, ensuring no critical evidence is lost.
- **Semantic preservation:** If the scene exhibits substantial semantic change (i.e., the semantic-change score exceeds), PPO policy assigns a higher probability to the KEEP action given the current state, to improve downstream scene understanding.
- **Visual redundancy control:** When semantics are negligible (no important event and low semantic change), the decision is driven by SigLIP visual similarity. .
  - High similarity ($s_t^{(\text{vis})} > \tau_{\text{drop}}$): the current observation is visually near-identical to the retained reference (e.g., same viewpoint, unchanged layout, no new objects), so storing it adds negligible new information $\Rightarrow$ DROP.
  - Moderate similarity ($\tau_{\text{low}} < s_t^{(\text{vis})} \leq \tau_{\text{drop}}$): the frame shows limited novelty (e.g., small viewpoint drift, minor lighting/weather variation, slight object motion) and may still support continuity or later retrieval $\Rightarrow$ KEEP the frame in low resulation to preserve coarse context while reducing storage and bandwidth.
  - Low similarity ($s_t^{(\text{vis})} \leq \tau_{\text{low}}$): the visual content differs substantially (*e.g.*, new scene region, significant viewpoint change), indicating high information gain $\Rightarrow$ KEEP at full quality to protect downstream mapping/localization and semantic interpretability.

As shown in Figure 3, when no important event is detected with low semantic change, the policy uses a visual redundancy score to drop redundant frames, store moderately redundant ones at 25% lower resolution, and keep informative frames.

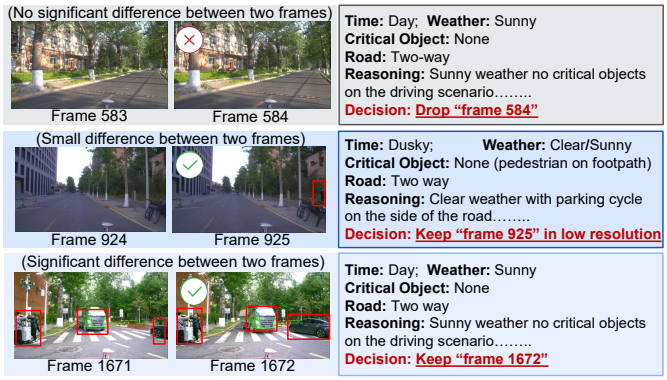

Fig. 3. Content-aware frame deduplication examples across low, moderate, and high similarity: low-similarity pairs are kept, moderate-similarity pairs are kept at lower resolution, and high-similarity pairs are dropped, guided by scene context (time/weather), critical objects (*e.g.*, pedestrian), and road type.

B. Safety-Aware Adaptive Logging for LiDAR

High-frequency 3D cloud points of LiDAR sensors generate large volumes of data in CAVs. Traditional logging strategies such as fixed-rate sampling or ego-motion triggers often retain many near-duplicate scans in slowly changing environments, leading to substantial storage overhead with little added perceptual value. We address this by measuring geometric change between consecutive scans, keeping structurally informative frames, and dropping redundant ones.

1) Content-Aware Deduplication: A LiDAR scan at time t is represented as a point cloud $\mathcal{P}_t = \{(x_i, y_i, z_i, I_i)\}_{i=1}^{M_t}$, where (x_i, y_i, z_i) denotes 3D coordinates and I_i is return intensity. Directly comparing raw point clouds is computationally expensive, so each scan is projected into a 2D bird's-eye-view (BEV) grid. Each point maps to a discrete ground-plane coordinate (x, y), and a feature encoding $g(z, I)$ (*e.g.*, height or intensity) populates the pixel, forming the BEV image $\mathbf{B}_t(x, y) = g(z, I)$. This projection preserves scene structure while enabling efficient grid-based processing.

(i) BEV Projection and Histogram: Bird's-Eye View (BEV) is a top-down 2D projection of the 3D scene that converts irregular point clouds into a structured grid for efficient spatial reasoning; the BEV image (Figure 4) is converted into a normalized histogram $H_t(i)$ of N bins, where $H_t(i) = h_t(i)/\sum_j h_t(j)$ and $h_t(i)$ is the raw count in bin i, and normalization improves robustness to varying point densities, vehicle speed, and sensor noise while compactly summarizing geometric structure for rapid comparison.

(ii) Chi-square Distance: Geometric change between frames is quantified using the Chi-square distance $\chi_t^2 = \frac{1}{2}\sum_{i=1}^{N} \frac{(H_t(i) - H_{t+1}(i))^2}{H_t(i) + H_{t+1}(i) + \epsilon}$, where ϵ prevents division by zero. This metric is sensitive to relative differences in

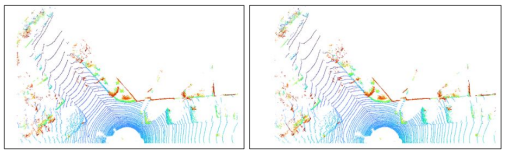

Fig. 4. Colorized bird's-eye-view (BEV) height maps generated from two consecutive LiDAR scans: (a) Frame 00 and (b) Frame 01. Pixel colors encode the normalized height (z) values using a colormap, while white pixels indicate empty BEV cells with no LiDAR returns.

histogram mass, allowing detection of structural variations such as object motion, new obstacles, or environmental changes, even under mild sensor noise. Because it captures relative change, it remains robust under variable density.

In Figure 5, we compare Chi-square thresholding for LiDAR deduplication. A *dynamic* threshold sets τ_{LiDAR} from the local distribution of $\{\chi_t^2\}$ (*e.g.*, $\tau_{\text{LiDAR}} = \text{Percentile}_p(\{\chi_t^2\})$, $p \in [70, 80]$), adapting to noise, registration error, occlusions, range effects, and scene dynamics to remain sensitive to meaningful structural change. In contrast, a static/fixed threshold assumes a stable scale for $\{\chi_t^2\}$; When the distribution drifts, a fixed τ becomes too loose in static segments and too strict in dynamic/noisy ones, causing inconsistent deduplication and degrading downstream mapping, localization, or learning.

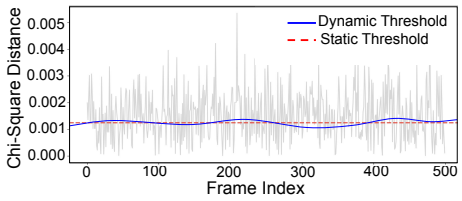

Fig. 5. Static vs. dynamic chi-square similarity thresholds for content-aware frame deduplication across a 500-frame sequence. The light-gray trace shows the per-frame chi-square distance (dimensionless) between consecutive frames, capturing short-term appearance/feature distribution changes over time (x-axis: frame index; y-axis: chi-square distance). The dynamic threshold (solid blue) adapts to local distance statistics rising in high-variability segments and relaxing in stable ones to avoid over-dropping during rapid changes while still removing redundant frames. The static threshold (red dashed) stays fixed, which can retain redundant frames in stable periods or drop informative frames during volatile segments. Overall, the figure highlights that adaptive threshold better aligns deduplication aggressiveness with time-varying scene dynamics compared to a single global cutoff.

(*iii*) RL-Based Deduplication: A new LiDAR frame is retained only when its geometric change exceeds the dynamic threshold: $\chi_t^2 > \tau_{\text{LiDAR}}$ $\Rightarrow$KEEP frame $t+1$; otherwise $\chi_t^2 \leq \tau_{\text{LiDAR}}$ $\Rightarrow$DROP frame $t+1$. Small χ_t^2 indicates nearly identical BEV-histogram distributions and minimal 3D structural change, so the frame adds little new information. Large χ_t^2 reflects meaningful spatial redistribution of returns (*e.g.*, motion, new obstacles, geometry change) and should be retained. The threshold τ_{LiDAR} controls deduplication sensitivity: smaller values preserve fine-grained motion/structure, while larger values drop more scans to save storage. When set dynamically (*e.g.*, percentile-based), it adapts to scene complexity, becoming stricter in dynamic scenes and more conservative in static ones.

2) Threshold Calibration via KISS-ICP: Selecting τ_{LiDAR} is critical, since overly aggressive frame removal can degrade odometry, mapping, and localization. We calibrate τ_{LiDAR} by evaluating LiDAR odometry under different thresholds using **KISS-ICP**, a lightweight geometry-driven scan-to-scan registration method. KISS-ICP is chosen because it **(1)** relies on scan-to-scan geometric consistency and is sensitive to missing/corrupted frames, **(2)** yields stable, interpretable trajectories without semantic priors, isolating deduplication effects, and **(3)** is a widely used CAV baseline, serving as a practical proxy for downstream localization quality.

To quantify the impact of deduplication on odometry accuracy, we measure two standard metrics:

a) Absolute Trajectory Error (ATE): $\text{ATE} = \frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{p}}_t - \mathbf{p}_t\|$ evaluates translational drift accumulated over the full trajectory. If the deduplication threshold is too high, informative frames are discarded, and KISS-ICP loses geometric constraints, causing ATE to increase.

b) Average Rotational Error (ARE): $\text{ARE} = \frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{R}}_t - \mathbf{R}_t\|_{\text{angle}}$ captures rotational misalignment between estimated and true orientations. ARE complements ATE because rotation is highly sensitive to dropping frames during turns or dynamic maneuvers.

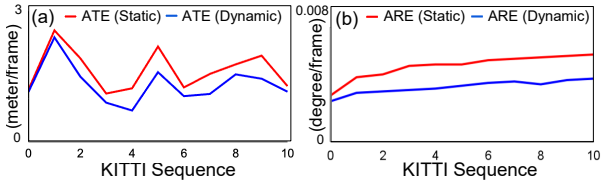

Fig. 6. Comparison of ARE and ATE of KISS-ICP application on static and dynamic Chi-square similarity thresholding across KITTI sequences for (a) Absolute Trajectory Error (m/frame) and (b) Absolute Rotation Error (deg/frame) are reported per sequence, with static (red) and dynamic (blue) thresholds; the dynamic strategy consistently yields lower error.

A static τ_{LiDAR} is suboptimal because scene structure and ego-motion vary over time: an aggressive τ drops non-redundant scans (*e.g.*, fast turns/low overlap), weakening KISS-ICP constraints and increasing ATE/ARE, while a conservative τ retains near-duplicates, wasting storage/compute and adding low-parallax updates. To address this, we use an RL agent to adapt τ_{LiDAR} to scenario demand, balancing storage, runtime, and odometry reliability. Concretely, τ_{LiDAR} is chosen from the Pareto frontier of (ATE, ARE), maximizing redundancy reduction under an accuracy constraint, reducing manual tuning and improving robustness.

3) Feature Extraction for RL State Space: Because LiDAR frames arrive sequentially, deduplication is formulated as a Markov Decision Process (MDP). Each frame is represented by a compact state vector $s_{3D} = [\text{ATE}, \text{ARE}, \chi^2]$, where ATE and ARE reflect localization stability and χ^2 measures geometric change. This 3-D state captures the key cues needed for KEEP/DROP decisions.

4) Reinforcement Learning Formulation: Fixed thresholds cannot reliably handle the diverse operating conditions of real-world LiDAR sequences, where geometric change varies with ego-motion, vibration, scene complexity, and environmental noise. To make deduplication adaptive and context-aware, we formulate the problem as a Markov Decision Process (MDP) in which an RL agent learns when a frame is informative enough to retain. At each timestep t, the agent observes a state s_t

summarizing geometric change (χ_t^2) and localization stability (ATE, ARE), and selects an action that trades off storage cost against downstream estimation accuracy.

(a) **Action space.** The agent chooses between two discrete actions KEEP and DROP, which determine whether the current frame is stored or removed. This framing casts deduplication as a sequential decision problem where choice affects subsequent localization accuracy and future rewards.

(b) **Reward function.** To balance storage efficiency with odometry performance, the reward evaluates the effect of dropping a frame on pose estimation and is defined as $r_t = 1 - (\text{ATE}_t + \lambda \cdot \text{ARE}_t)$, where ATE and ARE quantify the translational and rotational drift by KISS-ICP when the frame is removed, and controls the relative penalty on orientation errors. The reward is positive when dropping a frame preserves localization and negative when it induces drift, providing action-dependent feedback on geometric estimation quality.

(c) **Learning behavior.** Through repeated interaction with the environment, the agent implicitly learns the boundary between redundant and informative frames. Specifically, it identifies that: (1) large geometric changes (high χ^2) or rising localization errors should trigger KEEP, (2) low-change, low-error conditions justify DROP, and (3) intermediate cases require reasoning about temporal context. This results in a policy that adapts deduplication decisions to scene dynamics, noise, and vehicle motion rather than relying on fixed thresholds.

(d) **Training Procedure:** We train a Deep Q-Network (DQN) to approximate the optimal action-value function $Q(s_t, a_t)$. The agent is implemented using Stable-Baselines3 with a learning rate of 1×10^{-3}, a replay buffer of 5,000 transitions, and a batch size of 64. Training runs for 20,000 timesteps with ϵ-greedy exploration to ensure sufficient coverage of diverse geometric scenarios. Early stopping is used to avoid overfitting to specific motion patterns or noise conditions. The resulting policy selects an optimal KEEP/DROP action for each incoming LiDAR frame, enabling dynamic, performance-aware frame deduplication.

IV. VLM-Guided Adaptive Context-Aware Pipeline

To address RQ_2, the pipeline integrates a vision–language model (Qwen-VL) with a reasoning layer to enable adaptive perception under diverse weather conditions. Qwen-VL extracts weather and scene semantics from visual and textual cues and generates context annotations that guide context-driven augmentation. Building on these semantics, a chain-of-thought (CoT) module infers scene difficulty, complexity, and risk scores and converts them into actionable decisions. We present the outputs in three stages: **(1)** scene context extraction, **(2)** contextual sensor-reliability scoring, and **(3)** context-aware driving decisions. Figure 7 illustrates the CoT pipeline from visual input to reliability-aware perception for CAV.

A. Scene Context Extraction via VLM-Guided Module

1) Context-Aware Annotation: To achieve semantically consistent and context-aware environmental understanding, we employ the QwenVL to align visual embeddings with linguistic features for multimodal reasoning over weather

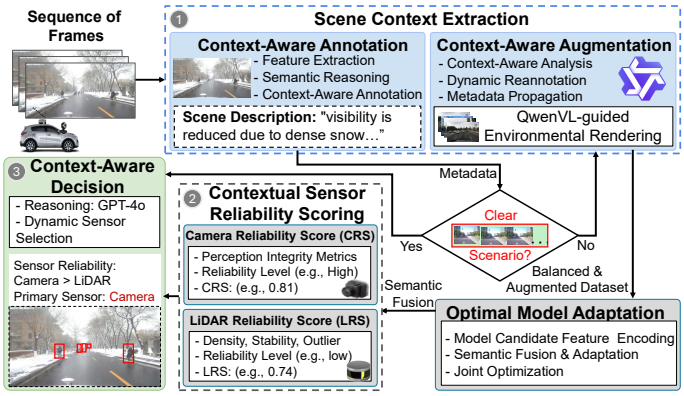

Fig. 7. Overview of the proposed VLM-guided adaptive context-aware pipeline. A chain-of-thought (CoT) reasoning pipeline enables a context-aware module under diverse driving conditions. Given a sequence of frames, we present the per-scene reasoning output in three stages: ① scene context extraction via context-aware annotation using Qwen-VL, which extracts scene understanding, *e.g.*, weather, visibility, objects semantics, and generates natural-language scene descriptions; these semantics further enable context-aware augmentation (environmental rendering, dynamic re-annotation, and metadata propagation) to improve robustness across operating conditions. If the metadata indicates a clear, normal driving scenario, no further computation is performed. Else, building on the extracted semantics, ② contextual Sensor Reliability Scoring derives scene-level difficulty, complexity, and risk scores for camera and LiDAR, and ③ the context-aware decision module converts them into actionable decisions for driving scenarios. The resulting decisions guide sensor prioritization (camera, LiDAR, or fusion) and downstream semantic fusion and model adaptation to maintain perception fidelity under challenging scenarios and visibility for connected autonomous vehicles.

semantics. Given an input frame x_i, it predicts $y_i = f_{\text{QwenVL}}(x_i) = \{\hat{W}_i, \hat{V}_i, \hat{P}_i\}$, where $\hat{W}_i$ denotes the inferred weather label, $\hat{V}_i$ quantifies visibility, and $\hat{P}_i$ represents the *Perceptual Alignment Confidence (PAC)*. The weather type is determined via $\hat{W}_i = \arg\max_{W \in \mathcal{W}} P(W \mid x_i, \theta)$, where $\mathcal{W} = \{\text{sunny}, \text{rainy}, \text{foggy}, \text{snowy}\}$ and θ are model parameters. PAC is estimated using a cross-modal alignment score $A_i = \sigma(F_v^\top W_a F_l)$ between visual features F_v and linguistic embeddings F_l, with W_a as a learned projection and $\sigma(\cdot)$ as the sigmoid function. The overall confidence is $\hat{P}_i = \lambda_1 A_i + \lambda_2 e^{-\kappa \, \text{Var}(L_i)}$, where $\text{Var}(L_i)$ denotes luminance variance, κ is a scaling factor, and $\lambda_1 + \lambda_2 = 1$. High luminance variance under adverse conditions (*e.g.*, rain or fog) decreases $\hat{P}_i$, indicating reduced perceptual consistency.

Additionally, Qwen-VL generates captions $\text{Caption} * i = \text{Decoder} * \phi(\text{Encoder}_\psi(x_i))$ (*e.g.*, "dense fog reduces lane visibility"), adding interpretable context for dataset enrichment. Together, dual encoding couples semantic confidence with linguistic cues, supporting adaptive, weather-aware perception.

Clear-driving condition workflow Under clear or normal conditions, the VLM offers minimal accuracy gains since baseline detectors like YOLOv10-M and RF-DETR already perform optimally with clear visuals (Figure 7, Table VI). Well-defined edges, textures, and colors enable traditional models to operate effectively without additional semantic reasoning. Thus, the VLM's strength lies primarily in compensating for perception loss under challenging scenarios, where ambiguity and degradation are more significant.

2) Context-Aware Augmentation: To mitigate dataset imbalance and strengthen adverse-weather coverage, the pro-

posed QwenVL-Based Weather Augmentation Layer synthesizes realistic environmental effects under semantic control, simulating rain, fog, snow, and low illumination to improve robustness to unseen conditions. For each frame x_i with weather label W_i, QwenVL extracts contextual tokens (*e.g.*, scene type, visibility, illumination, surface texture) to sample augmentation parameters $\mathcal{A}_i = \{\rho_{\text{fog}}, \rho_{\text{rain}}, \rho_{\text{snow}}, \delta_{\text{illum}}\}$, which regulate weather intensity and lighting. The frame is then transformed using physics-inspired models for visibility attenuation, rain streaks, and snow occlusion, while semantically inconsistent outputs are filtered to preserve coherence. This process yields a balanced augmented dataset $\mathcal{D}'$ with realistic and diverse conditions, improving weather-invariant training.

Optimal Model Adaptation for Sensors: The enriched dataset $\mathcal{D}'$ from the QwenVL weather augmentation layer trains complementary perception models balancing real-time speed and weather robustness. For camera stream, we use *YOLOv10* (fast, anchor-free) and *RF-DETR* (weather-resilient transformers) (Table VI). YOLOv10 is trained with $\mathcal{L}_{\text{YOLO}} = \mathbb{E}_{(x,y)\sim\mathcal{D}'}[\lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{bbox}}\mathcal{L}_{\text{bbox}} + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}}]$, and uniform weather sampling ($p(W_i) = 1/|\mathcal{W}|$) reduces clear-weather bias. RF-DETR injects weather cues into embeddings, $\mathbf{E}_i = \text{Embed}(x'_i) + \psi(W_i, \text{visibility}_i)$, and is trained with $\mathcal{L}_{\text{RF-DETR}} = \mathbb{E}_{(x,y)\sim\mathcal{D}'}[\lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{bbox}}\mathcal{L}_{\text{bbox}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}}]$. We optimize $\mathcal{L}_{\text{total}} = \eta_1\mathcal{L}_{\text{YOLO}} + \eta_2\mathcal{L}_{\text{RF-DETR}}$ and fuse features as $F_{\text{fusion}} = \Phi(F_{\text{YOLO}}, F_{\text{RF-DETR}})$ for robust camera representations under low light and adverse weather [42]–[45],.

For LiDAR, we use *CenterPoint* for its accuracy-efficiency trade-off and BEV-centric predictions (Table VIII) *under dense traffic and occlusion*. Frame-level LiDAR reliability is the mean detection confidence, $C_m^l(f_i) = \frac{1}{M_i}\sum_{j=1}^{M_i} c_{ij}^l$, where $c_{ij}^l \in [0,1]$ is the confidence of the j-th detection and M_i is the number of detections in frame f_i.

B. Contextual Sensor Reliability Scoring

1) Camera Reliability Score (CRS): It quantifies the trustworthiness of camera-based perception under varying lighting and weather conditions. It integrates three principal factors:

(*i*) Model-based confidence (C_m): For each frame f_i, the mean detection confidence is given by $C_m(f_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} c_{ij}$, where c_{ij} is the confidence of the j^{th} object and N_i is the total detections.

(*ii*) Image quality (Q_i): Visual clarity is assessed through sharpness (S), brightness (B), and contrast (K), normalized as $Q_i(f_i) = \frac{1}{3}\left(\frac{S(f_i)}{S_{\max}} + \frac{B(f_i)}{B_{\max}} + \frac{K(f_i)}{K_{\max}}\right)$, where each term lies in $[0,1]$ after adverse scenarios dataset normalization.

(*iii*) Weather influence (W_w): Each weather type is assigned an empirical reliability value $W_w \in [0,1]$ to account for visual degradation from phenomena such as rain, fog, or snow.

Reliability fusion and classification: The overall reliability is computed as $CRS(f_i) = \alpha C_m(f_i) + \beta Q_i(f_i) + \gamma W_w$, where $\alpha + \beta + \gamma = 1$ and $(\alpha, \beta, \gamma) = (0.6, 0.2, 0.2)$ ensure robustness across conditions. Based on the resulting score, reliability is categorized as *High* ($CRS \geq 0.8$), *Medium* ($0.65 \leq CRS < 0.8$), *Low* ($0.3 \leq CRS < 0.65$), and *Very Low* ($CRS < 0.3$). These levels guide adaptive sensor weighting in the Weather-Aware Reasoning Layer, ensuring consistent perception across changing environmental conditions. Overall, CRS offers a unified and interpretable measure of weather-aware camera reliability.

2) LiDAR Reliability Score (LRS): LiDAR scan quality from raw point cloud. Scan quality is computed directly from the point cloud L_i (*e.g.*, `.bin` containing $(x, y, z, \text{intensity})$). After filtering a fixed region of interest (ROI), we compute three normalized scan-health metrics in $[0,1]$:

(*i*) Completeness/density: For a LiDAR frame f_i, point-return completeness is $D(f_i) = \text{clip}\big(N_{\text{pts}}(f_i)/N_{\text{pts}}^{\text{ref}}, 0, 1\big)$, where $N_{\text{pts}}(f_i)$ is the ROI point count and $N_{\text{pts}}^{\text{ref}}$ is a clear-weather reference (*e.g.*, median over clean segments). Weather attenuation reduces valid returns, lowering $D(f_i)$ and indicating degraded geometric observability; $\text{clip}(\cdot)$ bounds the score to $[0,1]$ for stability and fusion with other normalized terms.

(*ii*) Range validity: In the next stage of the LiDAR frame, $R(f_i) = (1/N_{\text{pts}}(f_i))\sum_{k=1}^{N_{\text{pts}}(f_i)} \mathbb{1}(d_{\min} \leq d(p_k) \leq d_{\max})$, where $d(p_k)$ is the range of point p_k and $[d_{\min}, d_{\max}]$ is the valid sensing interval; the logic is simple: valid scans keep most returns within the sensor range, while weather noise and dropouts increase out-of-range or spurious points, so a higher $R(f_i)$ implies a more stable LiDAR scan.

(*iii*) Noise robustness $N(f_i) = 1 - r_{\text{out}}(f_i)$, where $r_{\text{out}}(f_i) \in [0,1]$ is the estimated outlier ratio (*e.g.*, using k-NN statistical outlier removal); the logic is that a clean scan has few isolated/spurious points, while rain/snow backscatter and multipath increase outliers, so higher r_{out} reduces reliability. The overall LiDAR scan-quality score is then $Q_i^l(f_i) = \frac{1}{3}(D(f_i) + R(f_i) + N(f_i)) \in [0,1]$, which averages completeness, valid-range consistency, and noise robustness into a single bounded quality measure.

VLM-guided weather penalty for LiDAR. LiDAR reliability degrades in adverse weather due to rain/snow backscatter and fog attenuation, so we apply a VLM-conditioned penalty that captures context severity in a bounded, interpretable manner. Given QwenVL outputs $\{\hat{W}_i, \hat{V}_i, \hat{P}_i\}$ for frame f_i, we define $W_w^l(f_i) = \omega^l(\hat{W}_i)\, g(\hat{V}_i)\, h(\hat{P}_i) \in [0,1]$, where $\omega^l(\hat{W}_i) \in [0,1]$ is a per-weather prior, $g(\hat{V}_i)$ scales with VLM-estimated visibility $\hat{V}_i \in [0,1]$, and $h(\hat{P}_i)$ down-weights the penalty when the VLM context is uncertain ($\hat{P}_i \in [0,1]$). We use bounded affine gates $g(\hat{V}_i) = \delta + (1-\delta)\hat{V}_i$ and $h(\hat{P}_i) = \epsilon + (1-\epsilon)\hat{P}_i$, with $\delta = 0.3$ to prevent collapse under extremely low visibility and $\epsilon = 0.5$ to avoid over-penalization when $\hat{P}_i$ is low. Thus, W_w^l decreases with worsening visibility and adverse weather priors while remaining stable to noisy context estimates.

4) LiDAR model confidence: To quantify LiDAR detection reliability, we employ *CenterPoint* as the LiDAR 3D object detector and compute a frame-level confidence score as the mean detection confidence over all predicted objects, $C_m^l(f_i) = \frac{1}{M_i}\sum_{j=1}^{M_i} c_{ij}^l$, where $c_{ij}^l \in [0,1]$ denotes the CenterPoint confidence score of the j^{th} detected object in frame f_i and M_i is the total detections in that frame.

5) Final LiDAR Reliability Score (LRS) The final LiDAR

reliability is computed as $LRS(f_i) = \alpha_l C_m^l(f_i) + \beta_l Q_i^l(f_i) + \gamma_l W_w^l(f_i)$ with $\alpha_l + \beta_l + \gamma_l = 1$, and we use default weights $(\alpha_l, \beta_l, \gamma_l) = (0.6, 0.25, 0.15)$ to emphasize detector confidence and scan health while still incorporating VLM-based contextual attenuation.

6) Reliability categories. For decision-making and explainability, we discretize the LiDAR Reliability Score $LRS \in [0, 1]$ into four non-overlapping, safety-conservative levels: *Very Low* (< 0.30; near-failure, LiDAR should not dominate), *Low* (0.30–0.65; degraded scans, down-weight LiDAR and rely more on other sensors), *Medium* (0.65–0.80; usable but partially degraded, balanced fusion), and *High* (≥ 0.80; confident scans, prioritization). Thresholds are tuned on a validation set to avoid false LiDAR prioritization under weather while keeping decisions stable and interpretable; binning follows common calibration practice [46]–[49], and the conservative low-reliability regimes reflect known adverse-weather LiDAR degradation and uncertainty-aware fusion [50]–[54]. Overall, this formulation provides a unified, context-aware LiDAR reliability score by combining QwenVL-based weather semantics with point-cloud statistics from raw LiDAR frames.

C. Context-Aware decision

The proposed reasoning layer integrates weather awareness, frame-level reliability estimation, and adaptive sensor prioritization into a context-aware perception policy. Using **GPT-4o**, it performs multimodal reasoning over QwenVL semantics and reliability cues to assign fusion weights. Given weather W_i, time context t_i, and sensor reliabilities (R_c, R_l), the decision is Decision $= f_{\text{LLM}}(W_i, R_c, R_l, t_i) \rightarrow [w_c, w_l]$, and the fused output is $O_t = w_c F_c + w_l F_l$. For each frame x_i, the system outputs $z_i = \{\text{frame}_i, W_i, R_i, D_i, E_i\}$, where R_i is overall reliability, D_i is the prioritized sensor, and E_i is an explanation. Reliability is computed as $R_i = \alpha V_i + \beta(1 - B_i) + \gamma E_i^{(\text{stability})}$ with $(\alpha, \beta, \gamma) = (0.4, 0.3, 0.3)$, where V_i encodes visibility, B_i captures degradation cues (*e.g.*, blur/contrast loss or LiDAR outliers), and $E_i^{(\text{stability})}$ measures temporal consistency. Under clear conditions, the policy prioritizes the camera when $R_c \geq \tau_W$; under fog/rain/snow it increases LiDAR weight for geometric robustness. The LLM also outputs concise rationales (*e.g.*, High visibility supports camera" or Fog reduces contrast; prioritize LiDAR"). Overall, integrating reliability scores with contextual reasoning yields a transparent, weather-adaptive fusion strategy that is both robust and practical for deployment.

V. Experiment Results and Analysis

To perform the experiment and address our research questions, we use the following experimental setup.

Dataset Collection and Preparation: We evaluate the proposed framework on the V2X-Radar dataset [55], which contains high-resolution driving scenes across diverse weather, lighting, and geographic conditions. The dataset provides synchronized RGB images, LiDAR point clouds, and radar measurements for multimodal perception in adverse conditions. We denote it as $\mathcal{D} = \{(x_i, W_i, t_i, s_i)\}_{i=1}^N$ with $x_i = \{x_i^{\text{Camera}}, x_i^{\text{LiDAR}}\}$. Camera frames are normalized as $x_i'^{\text{Camera}} = (x_i^{\text{Camera}} - \mu_x)/\sigma_x$ improving feature stability under illumination changes. Figure 8 illustrates representative day/night scenes with varied road structures and objects (*e.g.*, pedestrians, buses, cyclists) under real-world weather.

Fig. 8. Dataset diversity across weather and time-of-day. Representative camera frames illustrating diverse driving conditions used in our evaluation: three adverse-weather scenarios (Sunny, Foggy, Rainy, Snowy) and three illumination regimes (Day, Dusky, Night).

Hardware and Software Configuration: Experiments were conducted on a high-performance workstation optimized for machine learning and rendering, while large-scale data processing and training were executed on a Linux-based GPU cluster equipped with four NVIDIA GeForce RTX 2080 Ti GPUs with 11 GB of VRAM and 256 GB of system memory. The software stack is built on CUDA 12.2 and cuDNN 8.9 for efficient GPU acceleration, with PyTorch 2.3 adopted as the primary deep learning framework. This configuration provides the computational capacity and reliability required for advanced autonomous driving applications.

A. Performance Evaluation of Adaptive Logging (CAAL-VLM)

We evaluate the proposed *CAAL-VLM* pipeline through a series of experiments. Since *Camera* and *LiDAR* produce different data rates and burst behaviors, an effective logging policy must reduce total storage while also limiting bandwidth spikes that affect real-time communication and I/O.

TABLE I: **Storage and bandwidth statistics for adaptive logging policies on Camera and LiDAR streams.**

Sensor	Policy	Total (GB)↓	Avg (MB/s)↓	Peak (MB/s)↓	Ratio
Camera Stream					
Camera	Baseline	7.5721	6.81	12.12	1.000
	Heuristic	4.3236	3.41	12.12	0.571
	Adaptive-logging	5.8221	4.87	8.67	0.769
LiDAR Stream					
LiDAR	Baseline	24.5357	22.06	30.20	1.000
	Heuristic	15.1385	18.03	28.18	0.617
	Adaptive-logging	17.9060	18.98	26.01	0.732

As shown in Table I, *Adaptive-logging* reduces total storage by 23.1% for *Camera* and 27.0% for *LiDAR*, and lowers the camera peak rate from 12.12 to 8.67 MB/s. While the heuristic achieves a lower average camera rate (3.41 MB/s), it does not reduce peak bursts but drop informative frames. In contrast, Adaptive-logging smooths bandwidth by removing scenario-dependent redundancy while preserving informative data.

To assess *deployability* beyond storage reduction, we measure CPU utilization as runtime overhead and camera/LiDAR I/O utilization as data pipeline pressure. These metrics determine whether *Adaptive-logging* operates in real time without saturating compute or I/O, enabling fair comparison with baseline and heuristic policies under identical constraints (Table II).

Table II shows that the baseline incurs the highest load due to continuous logging, while the heuristic yields the lowest utilization by aggressively dropping data and discarding information. By contrast, *Adaptive-logging* (VLM-guided) retains

scenario-relevant data with much lower overhead than the baseline (CPU 6.24%, Cam I/O 4.01%, LiDAR I/O 11.31%), reducing resource usage while better preserving safety-critical events than heuristic gating.

TABLE II: **System resource utilization across logging policies.**

Policy	CPU Util (%)↓	Cam I/O Util (%)↓	LiDAR I/O Util (%)↓
Baseline	10.31	4.54	14.71
Heuristic	5.12	3.27	9.35
Adaptive-logging	6.24	4.01	11.31

While Table II shows that *Adaptive-logging* reduces CPU and I/O overhead compared to continuous logging, Table III confirms that these savings do not come at the expense of critical content: out of 98,430 baseline objects, Adaptive-logging attains a recall $\sim$1.37$\times$ higher than the heuristic, indicating markedly better preservation of object-level evidence under the same logging constraints. Overall, the policy reduces resource usage primarily by filtering redundant segments rather than discarding informative, event-relevant frames.

TABLE III: **Object retention under different logging policies**

Policy	Objects		Retention Recall		
	Logged	Rec. Frac.	Ped	Cyc	Veh
Heuristic	69,392	0.705	0.687	0.731	0.698
Adaptive-logging	**94,893**	**0.964**	**0.932**	**0.901**	**0.961**

Building on this object-level retention, Table IV further verifies that the proposed *Adaptive-logging* improves *task-level utility* under realistic conditions, achieving higher vehicle IoU (0.7/0.5) than the heuristic. By stratifying results across weather and time-of-day for LiDAR and camera detectors, the table shows that heuristic logging drops informative frames, especially under low visibility, while context-aware logging better preserves cross-domain performance. Overall, it confirms that VLM-guided adaptive logging is data-efficient and maintains higher perception accuracy than heuristic logging.

With gains in storage reduction, bandwidth control, detection performance, and resource utilization established, we next evaluate *real-time* behavior. Table V summarizes runtime quality using standard metrics: mean latency, p95/p99 tail latency, QoS violation rate, throughput (FPS), and jitter.

TABLE IV. **Performance across diverse scenarios for vehicle detection (IoU $= 0.7/0.5$).**

Method	Sunny	Rainy	Snowy	Day	Dusk	Night
Adaptive-logging						
LiDAR-based Detectors (Vehicle)						
PointPillars	75.03/83.69	73.24/82.87	73.31/82.12	78.00/88.71	71.97/80.06	71.24/79.16
SECOND	81.23/90.98	74.34/83.47	74.18/81.21	83.61/91.06	71.22/81.23	70.02/78.76
CenterPoint	86.44/94.04	78.98/84.26	75.43/83.09	88.12/95.22	73.08/83.11	71.29/82.10
PV-RCNN	85.83/94.11	81.39/86.61	83.02/88.25	88.83/95.67	82.49/86.49	81.09/84.02
SQDNet	87.29/93.32	?83.48/86.94	82.08/85.05	89.12/95.10	81.48/88.34	80.26/85.91
Fade3D	81.03/90.43	73.56/81.72	74.77/83.31	82.48/92.37	76.98/83.12	74.87/82.03
Camera-based Detectors (Vehicle)						
SMOKE	62.05/78.43	65.72/71.36	69.69/74.22	64.20/75.21	61.11/66.06	60.65/63.20
BEVDepth	75.01/83.25	72.23/77.81	73.29/81.75	75.23/84.01	72.13/76.56	72.22/78.54
BEVHeight	77.91/82.45	75.53/80.48	75.49/77.14	77.98/82.85	72.19/76.70	70.19/77.01
BEVHeight++	76.48/83.81	73.92/82.36	72.38/74.22	79.48/83.56	71.22/74.29	70.01/73.92
Heuristic logging						
LiDAR-based Detectors (Vehicle)						
PointPillars	69.23/74.76	66.03/70.28	65.91/70.07	70.15/75.66	67.87/71.35	65.96/71.54
SECOND	76.65/81.05	69.34/76.47	68.87/77.65	75.09/81.34	70.22/76.01	63.37/69.82
CenterPoint	77.03/84.96	71.76/78.92	70.29/76.81	76.28/83.78	71.89/72.86	69.82/73.65
PV-RCNN	75.08/83.09	74.86/81.52	74.23/80.07	77.34/94.54	73.87/79.23	72.01/78.09
SQDNet	76.89/85.01	74.98/81.76	75.79/82.76	79.87/86.66	75.92/82.20	75.23/80.29
Fade3D	72.27/77.65	68.28/73.28	69.54/74.27	74.32/80.43	70.17/77.43	70.91/76.02
Camera-based Detectors (Vehicle)						
SMOKE	49.98/56.01	47.70/54.21	46.24/54.08	48.10/56.92	45.65/54.48	44.72/50.28
BEVDepth	63.27/69.05	61.98/67.81	62.17/67.97	64.98/69.83	62.26/66.04	61.06/65.91
BEVHeight	65.14/70.99	63.02/68.29	62.91/68.95	66.43/71.07	63.65/69.52	61.26/67.61
BEVHeight++	69.30/74.97	64.65/69.92	65.28/69.97	74.06/75.92	65.12/69.43	64.31/69.02

TABLE V: **Latency and QoS statistics across policies.**

Policy	Mean (ms)↓	p95 (ms)↓	p99 (ms)↓	QoS Viol.↓	FPS↑	Jitter↓
Baseline	16.62	19.97	21.12	0.030	10.0	8.5
Heuristic	13.63	19.77	21.24	0.029	5.0	11.2
Adaptive-logging	**9.53**	**17.78**	**18.07**	**0.026**	**10.0**	**6.8**

Note: Lower is better for latency, jitter, and QoS violations; higher is better for FPS.

Adaptive-logging achieves the lowest mean and tail latencies (9.53 ms mean; 17.78 ms p95), the lowest QoS violation rate, full throughput, and lower jitter. In contrast, the heuristic lowers mean latency but degrades throughput and increases jitter, reflecting instability from non-context-aware pruning. Overall, *Adaptive-logging* reduces resource usage while improving responsiveness and stability for safety-critical deployment.

Overall, the results validate $\mathbf{RQ}_1$, our context-aware policy improves efficiency, accuracy, and real-time responsiveness, consistently outperforming the heuristic baseline.

RQ_1 examines whether risk-aware multi-sensor logging can reduce storage and write costs under strict compute and I/O constraints while preserving online perception performance and safety-critical fidelity. RQ_2 focuses on operationalizing VLM-guided prioritization as a stable and interpretable policy with consistent retention and verifiable explanations under the same constraints. In essence, RQ_1 defines *what to preserve*, while RQ_2 ensures it is *reliable and accountable*.

B. Testing of VLM-Guided Adaptive Context-Aware Pipeline

Table VI reports camera-based object detection across weather conditions, where mAP, latency, and FPS capture accuracy, cost, and efficiency; 1S, 2S, and T denote one-stage, two-stage, and transformer-based detectors, respectively.

(1) One-Stage Detectors (1S): Among one-stage baselines in Table VI, EfficientDet-D5 offers a moderate accuracy–speed trade-off, while SSD-VGG and RetinaNet [67], [68] degrade in fog and snow. In contrast, YOLO models remain robust and real-time; notably, *YOLOv10-M* achieves the best one-stage accuracy (mAP $= 0.78$) at $>$80 FPS, making it well-suited.

(2) RCNN Family (2S): Two-stage R-CNN variants achieve only moderate clear-weather mAP but incur high overhead from region proposals and per-region refinement, leading to

TABLE VI. **Perception detector performance across environmental conditions.(↑ /↓ /↑).**

Model	Sunny	Rainy	Foggy	Snowy
Metric: mAP / Lat. (ms) / FPS (↑ /↓ /↑)				
One-stage (1S)				
SSD (1S) [56]	0.30/12/83	0.25/14/71	0.20/16/62	0.22/15/66
RetinaNet (1S) [57]	0.40/142/7	0.34/150/7	0.28/158/6	0.30/155/6
EfficientDet-D5 (1S) [58]	0.52/67/15	0.47/72/14	0.41/78/13	0.43/74/14
YOLOv8-L (1S) [59]	0.71/10/100	0.66/10/94	0.68/10/90	0.56/10/85
YOLOv9-C (1S) [60]	0.76/12/100	0.75/13/90	0.71/13/95	0.65/11/80
YOLO-NAS (1S)	0.72/9/110	0.69/9/110	0.70/9/110	0.59/9/110
YOLOv10-M (1S) [61]	**0.78/9/111**	**0.75/10/100**	**0.72/11/91**	**0.69/11/83**
Two-stage (2S)				
R-CNN (2S)	0.42/2000/1	0.34/2100/0.5	0.28/2250/0.9	0.30/2150/0.93
Faster R-CNN (2S) [62]	0.44/100/10	0.38/108/9	0.33/115/9	0.35/110/9
Mask R-CNN (2S) [63]	0.41/280/4	0.36/295/3	0.31/310/4	0.33/300/4
Transformer-based (T)				
Deform. DETR (T) [64]	0.75/14/57	0.67/16/58	0.60/16/57	0.64/17/57
RT-DETR (T) [65]	0.76/15/67	0.70/16/63	0.63/18/56	0.66/17/59
RF-DETR (T) [66]	**0.81/6/71**	**0.75/7/66**	**0.69/8/62**	**0.72/7.5/60**

Note: Each cell reports mAP/latency (ms)/FPS.

high latency and low FPS; even Faster R-CNN, while lighter, remains ill-suited for real-time CAV [69], [70].

(3) Transformer-Based Detectors: Transformer-based detectors generalize better across weather due to global attention and richer semantics. Notably, *RF-DETR* achieves the highest accuracy (mAP > 0.80) at low latency (6 ms), outperforming others, and its attention-driven fusion better preserves spatial consistency under occlusions and low-contrast scenes.

Table VI shows that fog and rain cause the largest accuracy drops due to contrast loss and noise, and reveals a clear accuracy–efficiency trade-off: two-stage detectors are slow, one-stage models (*e.g.*, YOLOv10-M) remain real-time with strong mAP, and transformers (*e.g.*, RF-DETR) are more robust across weather at low latency. Therefore, combining YOLOv10-M and RF-DETR in our pipeline (RQ_2) balances accuracy, runtime, and all-weather reliability.

Subsequently, we assess whether VLM guidance improves robustness under weather variation. Table VII shows that integrating the VLM strengthens detection consistency across conditions while maintaining feasible real-time performance.

TABLE VII. **Performance comparison under diverse scenarios.**

Scenario	YOLOv10-M mAP/Lat./FPS	RF-DETR mAP/Lat./FPS	VLM-Enhanced mAP/Lat./FPS
Sunny	0.78/9/111	0.80/6/71	0.78/9/105
Rainy	0.75/10/100	0.77/7/62	**0.86**/12/91
Foggy	0.72/11/91	0.69/8/76	**0.78**/15/83
Snowy	0.69/11/83	0.72/7/60	**0.80**/14/77

Note: Each entry is mAP↑/latency(ms)↓/FPS↑. VLM-Enhanced leverages visibility context for robust perception.

Integrating the VLM significantly improves robustness, contextual awareness, and efficiency. For *YOLOv10-M*, VLM-guided reasoning boosts accuracy by up to 11% under low visibility, rain, and fog, while preserving real-time performance with minimal latency increase and throughput above 80 FPS. Similarly, RF-DETR gains up to +0.09 accuracy in rain and fog while sustaining 60–70 FPS. By incorporating weather semantics and reliability-aware attention, the VLM reduces ambiguity and stabilizes detection, improving accuracy.

Table VIII validates LiDAR robustness and the benefit of VLM-guided adaptation for *vulnerable road users* (pedestrians and cyclists). Stratifying results across weather isolates environmental degradation effects on LiDAR, while two IoU thresholds (0.5/0.25) separate strict localization from detection sensitivity. Across PointPillars, SECOND, and CenterPoint, the *VLM-Enhanced* setting improves accuracy and stability under adverse conditions, supporting safety-critical perception.

TABLE VIII. Class-wise LiDAR 3D detection mAP (IoU $0.5/0.25$) across weather.

Scenario	PointPillars	SECOND	CenterPoint	VLM-Enhanced
Pedestrian (IoU $0.5/0.25$)				
Sunny	54.82/69.61	57.98/73.45	67.34/79.65	**77.86/85.63**
Rainy	52.94/74.64	55.23/70.58	64.09/75.22	**74.12/82.36**
Foggy	51.65/73.53	54.76/69.45	63.76/76.82	**73.18/82.08**
Snowy	51.98/74.21	55.12/71.39	61.92/75.22	**75.93/84.26**
Cyclist (IoU $0.5/0.25$)				
Sunny	71.01/78.48	73.71/78.03	75.35/78.39	**82.90/92.26**
Rainy	69.54/75.93	71.26/75.09	71.30/77.26	**76.31/85.87**
Foggy	67.67/74.65	70.34/75.19	72.45/78.37	**76.36/85.80**
Snowy	68.27/74.94	71.26/76.34	72.39/77.96	**77.68/87.42**

Table VIII shows that *VLM-Enhanced* outperforms existing models for both *Pedestrian* and *Cyclist* across all weather, indicating stronger robustness to sparse, cluttered point clouds. The largest gains occur under adverse weather at IoU 0.5: for Pedestrian, +10.0 (Rainy) and +14.0 (Snowy); for Cyclist, +5.0 (Rainy) and +5.3 (Snowy). Similar gains at IoU 0.25 indicate improved recall and reduced degradation.

VLM-guided CoT enables auditable, scenario-consistent sensor prioritization aligned with visibility and risk. Across foggy, snowy-night, and rainy scenes (Table IX), the VLM extracts context, assigns reliability and occlusion scores, and yields consistent policies: as visibility worsens, sensing shifts from camera-heavy to LiDAR-dominant or fused modes. Dense fog favors LiDAR & Camera fusion, while snowy nights prioritize LiDAR alone. This context-to-decision trace supports transparent auditing in safety-critical settings.

The results validate RQ_2, showing that our method balances

TABLE IX. An illustrative example of VLM-guided context-aware CoT reasoning output. For each scene, the output is shown in three steps: Step 1 (scene context extraction), Step 2 (contextual sensor reliability scoring), and Step 3 (context-aware decision).

Scenes	Step 1 (Scene Context Extraction)	Step 2 (Contextual Sensor Reliability Scoring)	Step 3 (Context-Aware Decision)
	Environment: Snowy, Time: Day, Road: Intersection, Lane: Multi-lane, Critical Objects: Car, Bike, Pedestrian. Risk: Moderate Alert!	CRS: High (≥ 0.80), LRS: Medium ($0.65 \leq$ LRS< 0.80), Occlusion: Low (clear visualization) Reasoning: Snowy weather with clear visibility and with critical objects contribute to risk in the driving scene at the intersection.	Sensor Score: Camera $\geq$ LiDAR Primary Sensor: Camera Reasoning: Snowy scene but clear visualization. The camera provides strong texture and color cues for lanes, traffic lights, and VRUs, enabling stable recognition; LiDAR serves as complementary depth verification.
	Environment: Foggy (dense haze), = Time: Day, Road: Urban Intersection, Lane: Multi-lane, Critical Objects: Bike, Pedestrian. Risk: High Alert!	CRS: Medium ($0.65 \leq$ CRS< 0.80), LRS: Medium ($0.65 \leq$ LRS< 0.80), Occlusion: High (fog veil) Reasoning: Foggy conditions and the presence of critical objects jointly elevate the risk level for driving.	Sensor Score: LiDAR $\approx$ Camera Primary Sensor: LiDAR & Camera Reasoning: fog lowers contrast and weakens camera-only cues, while LiDAR preserves geometric structure; fusing both improves robustness for detecting critical objects.
	Environment: Snowy (moderate snowfall) Time: Night, Road: Highway (slippery surface), Lane: Multi-lane, Critical Objects: Car, Bus. Risk: High Alert!	CRS: Low (< 0.65), LRS: High (≥ 0.80), Occlusion: Medium (snow streaks, spray) Reasoning: Snowy low-visibility with potential icy surface, occlusions and reduced contrast elevate risk.	Sensor Score: LiDAR $\geq$ Camera Primary Sensor: LiDAR Reasoning: : Snowy conditions reduce camera contrast and add glare, while snowfall can introduce sparse outliers in LiDAR; fusing LiDAR geometry with camera semantics improves robustness for detecting vulnerable objects.
	Environment: Rainy (raindrops on lens) Time: Day, Road: Urban Arterial (wet surface), Lane: Multi-lane, Critical Objects: Car, Bus. Risk: Moderate Alert!	CRS: Medium ($0.65 \leq$ CRS< 0.80), LRS: High (≥ 0.80), Occlusion: Medium (spray/raindrops) Reasoning:Rainy conditions reduce contrast and add glare/spray on a wet road, increasing braking risk.	Sensor Score: LiDAR $\geq$ Camera Primary Sensor: LiDAR Reasoning: Rain and spray reduce camera contrast and introduce glare/blur, while LiDAR preserves stable geometric depth cues; thus LiDAR provides more reliable object localization under degraded visibility.

efficiency, real-time feasibility, and all-weather reliability via VLM-guided adaptation and dynamic sensor prioritization.

VI. Key Observations and Discussions

⋆ **Observation$_1$:** *CAAL-VLM reduces data volume and smooths bandwidth spikes, without being as "lossy" as the heuristic* (Table I, II and III).

Discussion: Adaptive-logging reduces *camera storage* by $\sim$1.3×, *LiDAR storage* by $\sim$1.37×, and *camera peak bandwidth* by $\sim$1.4× relative to the baseline, yielding less bursty I/O and easing real-time pipeline load (Table I and II). Importantly, these gains are achieved *without over-pruning*: compared to the heuristic, Adaptive-logging retains more informative stream content, reducing the risk of missing rare but critical moments (*e.g.*, complex scenes, occlusions). Thus, heuristics compress more aggressively but lose more information, while Adaptive-logging balances efficiency and fidelity.

⋆ **Observation$_2$:** *Adaptive-logging improves downstream detector performance, with the largest gains in challenging diverse scenarios and low light* (Table IV).

Discussion: Across both LiDAR- and camera-based detectors, Adaptive-logging consistently outperforms heuristic logging (Table IV). For instance, in rain, CenterPoint gains accuracy at both IoU thresholds, and at night, the SMOKE shows an even larger improvement, indicating that Adaptive-logging better preserves challenging frames. Qualitatively, heuristic dropping can create a *distribution shift* by removing hard-but-informative scenes (low light, precipitation, complex interactions), whereas Adaptive-logging retains these cases and improves downstream detection under stress.

⋆ **Observation$_3$:** *VLM-guided semantics improves robustness in challenging conditions and enables auditable, scenario-grounded decisions* (Table IX).

Discussion: VLM-guided semantic reasoning improves robustness under adverse visibility while remaining practical online. As shown in Table VII and Table VIII, the VLM-Enhanced variant achieves higher mAP than the baseline in challenging conditions with only modest latency increases, maintaining real-time throughput and a favorable accuracy–efficiency trade-off. The approach is also *auditable*: Table IX provides an interpretable reasoning trace linking scene context and risk scoring to the final action and sensor choice (*e.g.*, foggy intersections with VRUs trigger elevated alerts and prioritized sensing/retention).

⋆ **Observation$_4$:** *Automated context labeling boosts performance efficiency and reducing human bias* (Table IX).

Discussion: QwenVL enables scalable labeling of weather, lighting, and visibility directly from raw images, reducing manual effort, preparation time, and subjective bias. Its multimodal reasoning outputs consistent, interpretable JSON-style metadata (*e.g.*, {*weather: "foggy", visibility: "low"*}; Table IX), which increases dataset diversity, supports adaptive retraining, and improves perception performance.

⋆ **Observation$_5$:** *Minimal performance gain observed under normal weather scenarios with VLM integration pipeline for camera-based detection* (Table VII).

Discussion: Under clear weather, the Vision Language Model offers minimal accuracy gain since baseline detectors like YOLOv10-M and RF-DETR already perform optimally with well-defined edges, textures, and colors. With little visual ambiguity or environmental degradation to correct, the VLM provides limited benefit, its primary strength emerging clearly under adverse weather conditions.

VII. Related Work

ML-based models for CAVs have been extensively studied, with a large body of work reporting steadily improving accuracy on curated benchmarks. For camera-based detection, one-stage models such as SSD and YOLO deliver real-time performance [67], [68], while two-stage detectors such as Faster R-CNN improve precision through region-based processing [69]–[72]; transformer-based detectors further enhance context reasoning and global feature modeling [45]. Beyond vision-only pipelines, CAV stacks increasingly pair cameras with LiDAR to exploit geometry-rich 3D perception, supported by modern BEV-centric detectors such as PointPillars [73]–[75]. However, the performance of these models can degrade under real-world adaptive scenarios (*e.g.*, adverse weather).

To mitigate such effects, prior work has explored robustness-oriented techniques including augmentation, domain adaptation, and multi-sensor fusion [42], [53], [76]–[78], together with LiDAR denoising and radar-optical fusion [79]–[82]. Although these ML models achieve steadily improving accuracy on curated benchmarks, most prior work focuses on sensing distortion and domain shift, while largely overlooking the impact of redundant and non-informative frames on ML performance in real CAV deployments. To bridge this gap, we design *CAAL-VLM*, a pipeline that reduces redundant and non-informative frames while preserving task-relevant content, thereby improving model accuracy and timeliness.

VIII. Conclusion

We presented *CAAL-VLM*, an adaptive logging framework that addresses the mismatch between benchmark-trained ML models and real-world CAV deployments, where operational sensing streams contain large fractions of redundant or low-salience frames. By jointly modeling redundancy and semantic risk, *CAAL-VLM* removes non-informative inputs while preserving safety-critical content, thereby suppressing wasted computation and reducing resource load without sacrificing perception fidelity. Through two RL–driven logging policies for camera and LiDAR data, and a VLM-guided sensing module that adapts sensing behavior to scene context, *CAAL-VLM* improves both perception timeliness and robustness under challenging conditions. Across extensive experiments, it reduces I/O overhead by $\sim$27% and perception latency by $\sim$42%, while maintaining accurate and reliable perception.

Acknowledgement

This work is supported in part by the National Science Foundation (NSF) grant CNS-2348151, CNS-2140346, CNS-2231523, and Commonwealth Cyber Initiative (CCI) grant HC-2Q26-032.

REFERENCES

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12 697–12 705.

[2] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10 529–10 538.

[3] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 7074–7082.

[4] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 913–922.

[5] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 641–656.

[6] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4604–4612.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012, pp. 3354–3361.

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11 621–11 631.

[9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine et al., "Scalability in perception for autonomous driving: Waymo open dataset," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2446–2454.

[10] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," arXiv preprint arXiv:1907.07484, 2019.

[11] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 4693–4700.

[12] Y. Dong, C. Kang, J. Zhang, Z. Zhu, Y. Wang, X. Yang, H. Su, X. Wei, and J. Zhu, "Benchmarking robustness of 3d object detection to common corruptions," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1022–1032.

[13] Y. Liu, B. Sun, Y. Tian, X. Wang, Y. Zhu, R. Huai, and Y. Shen, "Software-defined active LiDARs for autonomous driving: A parallel intelligence-based adaptive model," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 8, pp. 4047–4056, 2023.

[14] Y. Wang, Y. He, R. Wang, and W. Shi, "Quantitative analysis of storage requirement for autonomous vehicles," in Proceedings of the 16th ACM Workshop on Hot Topics in Storage and File Systems, 2024, pp. 71–78.

[15] M. Liu, X. Ding, and W. Du, "Continuous, real-time object detection on mobile devices without offloading," in 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2020, pp. 976–986.

[16] J. Yoon and M.-K. Choi, "Exploring video frame redundancies for efficient data sampling and annotation in instance segmentation," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 3308–3317.

[17] K. Hu, F. Gao, X. Nie, P. Zhou, S. Tran, T. Neiman, L. Wang, M. Shah, R. Hamid, B. Yin et al., "M-llm based video frame selection for efficient video understanding," in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 13 702–13 712.

[18] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," arXiv preprint arXiv:1812.05159, 2018.

[19] D. D. S. D. Community, "Ros 2: What is dds," https://community.rti.com/page/ros-2-what-dds, 2026, accessed: 2026-01-03.

[20] Y. Maruyama, S. Kato, and T. Azumi, "Exploring the performance of ros2," in Proceedings of the 13th international conference on embedded software, 2016, pp. 1–10.

[21] Y. He and W. Shi, "A faster and more reliable middleware for autonomous driving systems," arXiv preprint arXiv:2510.11448, 2025.

[22] J. Wang, Real-time embedded systems. John Wiley & Sons, 2017.

[23] Y. Ye, Z. Nie, X. Liu, F. Xie, Z. Li, and P. Li, "Ros2 real-time performance optimization and evaluation," Chinese Journal of Mechanical Engineering, vol. 36, no. 1, p. 144, 2023.

[24] K. S. Dong, D. Nikiforov, W. Soedarmadji, M. Nguyen, V. Jain, C. W. Fletcher, and Y. S. Shao, "Characterizing and optimizing real-time optimal control for embedded socs."

[25] M. Komorkiewicz, A. Chin, P. Skruch, and M. Szelest, "Intelligent data handling in current and next-generation automated vehicle development—a review," IEEE Access, vol. 11, pp. 32 061–32 072, 2023.

[26] A. Arooj, M. S. Farooq, A. Akram, R. Iqbal, A. Sharma, and G. Dhiman, "Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges," Archives of Computational Methods in Engineering, vol. 29, no. 2, pp. 793–829, 2022.

[27] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in 2018 26th Telecommunications Forum (TELFOR). IEEE, 2018, pp. 420–425.

[28] J. You, Z. Jiang, Z. Huang, H. Shi, R. Gan, K. Wu, X. Cheng, X. Li, and B. Ran, "V2X-VLM: End-to-end V2X cooperative autonomous driving through large vision-language models," Transportation Research Part C: Emerging Technologies, vol. 183, p. 105457, 2026.

[29] G. Liao, J. Li, and X. Ye, "VLM2Scene: Self-supervised image-text-LiDAR learning with foundation models for autonomous driving scene understanding," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 4, 2024, pp. 3351–3359.

[30] S. M. H. Abidi, S. M. Raza, and S. Y. Shin, "Safevision: Vision-language reasoning for context-aware safety monitoring," Neurocomputing, p. 132479, 2025.

[31] Y. Xu, Y. Hu, Z. Zhang, G. P. Meyer, S. K. Mustikovela, S. Srinivasa, E. M. Wolff, and X. Huang, "VLM-AD: End-to-end autonomous driving through vision-language model supervision," arXiv preprint arXiv:2412.14446, 2024.

[32] F. Bordes, R. Y. Pang, A. Ajay, A. C. Li, A. Bardes, S. Petryk, O. Mañas, Z. Lin, A. Mahmoud, B. Jayaraman et al., "An introduction to vision-language modeling," arXiv preprint arXiv:2405.17247, 2024.

[33] Y. Gao, J. Chen, and M. Li, "VLMs bridging-enhanced scene semantic reasoning framework for image-text matching," in Proceedings of the 2025 International Conference on Multimedia Retrieval, 2025, pp. 330–339.

[34] H. Zhu, S. Liang, W. Wang, B. Li, T. Yuan, F. Li, H. Wang, S.-L. Wang, and Z. Zhang, "Revisiting data auditing in large vision-language models," in Proceedings of the 33rd ACM International Conference on Multimedia, 2025, pp. 11 337–11 346.

[35] Q. Liu, C. Shang, L. Liu, N. Pappas, J. Ma, N. A. John, S. Doss, L. Marquez, M. Ballesteros, and Y. Benajiba, "Unraveling and mitigating safety alignment degradation of vision-language models," in Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 3631–3643.

[36] F.-Y. Sun, W. Liu, S. Gu, D. Lim, G. Bhat, F. Tombari, M. Li, N. Haber, and J. Wu, "LayoutVLM: Differentiable optimization of 3d layout via vision-language models," in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 29 469–29 478.

[37] Y. Xia, Y. Jiang, Y. Tan, X. Zhu, X. Yue, and B. Zheng, "MSR-Align: Policy-grounded multimodal alignment for safety-aware reasoning in vision-language models," arXiv preprint arXiv:2506.19257, 2025.

[38] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge et al., "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," arXiv preprint arXiv:2409.12191, 2024.

[39] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa et al., "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," arXiv preprint arXiv:2502.14786, 2025.

[40] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning. PmLR, 2021, pp. 8748–8763.

[41] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl ^{2}: Fast reinforcement learning via slow reinforcement learning," arXiv preprint arXiv:1611.02779, 2016.

[42] L. Rutten, "Deep learning for weather condition adaptation in autonomous vehicles," Journal of Artificial Intelligence Research and Applications, vol. 3, no. 1, pp. 274–306, 2023.

[43] S. Lu and W. Shi, "Vehicle computing: Vision and challenges," Journal of Information and Intelligence, vol. 1, no. 1, pp. 23–35, 2023.

[44] Y. Luo, D. Xu, G. Zhou, Y. Sun, and S. Lu, "Impact of raindrops on camera-based detection in software-defined vehicles," in 2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST), 2024, pp. 193–205.

[45] W. He, Y. Zhang, T. Xu, T. An, Y. Liang, and B. Zhang, "Object detection for medical image analysis: Insights from the rt-detr model," in Proceedings of the 2025 International Conference on Artificial Intelligence and Computational Intelligence, 2025, pp. 415–420.

[46] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in Icml, vol. 1, no. 05, 2001.

[47] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in Proceedings of the AAAI conference on artificial intelligence, vol. 29, no. 1, 2015.

[48] C. Gupta and A. Ramdas, "Distribution-free calibration guarantees for histogram binning without sample splitting," in International conference on machine learning. PMLR, 2021, pp. 3942–3952.

[49] S. Kuzucu, K. Oksuz, J. Sadeghi, and P. K. Dokania, "On calibration of object detectors: Pitfalls, evaluation and baselines," in European Conference on Computer Vision. Springer, 2024, pp. 185–204.

[50] M. Dreissig, D. Scheuble, F. Piewak, and J. Boedecker, "Survey on lidar perception in adverse weather conditions," in 2023 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2023, pp. 1–8.

[51] C. Goodin, D. Carruth, M. Doude, and C. Hudson, "Predicting the influence of rain on lidar in adas," Electronics, vol. 8, no. 1, p. 89, 2019.

[52] W. Y. Pao, J. Howorth, L. Li, M. Agelin-Chaab, L. Roy, J. Knutzen, A. Baltazar-y Jimenez, and K. Muenker, "Investigation of automotive lidar vision in rain from material and optical perspectives," Sensors, vol. 24, no. 10, p. 2997, 2024.

[53] E. Palladin, R. Dietze, P. Narayanan, M. Bijelic, and F. Heide, "Samfusion: Sensor-adaptive multimodal fusion for 3d object detection in adverse weather," in European Conference on Computer Vision. Springer, 2025, pp. 484–503.

[54] M. Fadili, L. Lecrosnier, S. Pechberti, and R. Khemmar, "Evaluation of an uncertainty-aware late fusion algorithm for multi-source bird's eye view detections under controlled noise," in Intelligent Robotics and Control Engineering, 2025.

[55] L. Yang, X. Zhang, J. Li, C. Wang, J. Ma, Z. Song, T. Zhao, Z. Song, L. Wang, M. Zhou et al., "V2x-radar: A multi-modal dataset with 4d radar for cooperative perception," arXiv preprint arXiv:2411.10962, 2024.

[56] F. Yang, L. Huang, X. Tan, and Y. Yuan, "Fasternet-ssd: A small object detection method based on ssd model," Signal, Image and Video Processing, vol. 18, no. 1, pp. 173–180, 2024.

[57] M. N. Alhasanat, M. H. Alsafasfeh, A. E. Alhasanat, and S. G. Althunibat, "Retinanet-based approach for object detection and distance estimation in an image," International Journal on Communications Antenna and Propagation (IRECAP), vol. 11, no. 1, pp. 1–9, 2021.

[58] N. Kandavel, S. Vinod, B. Shalini, R. Pavithra, S. Thangam et al., "Comparative analysis of yolov8 and efficientdet for object detection in autonomous vehicles," in 2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI). IEEE, 2025, pp. 1–6.

[59] Z. Afrin, F. Tabassum, H. B. Kibria, M. R. Imam, and M. R. Hasan, "Yolov8 based object detection for self-driving cars," in 2023 26th International Conference on Computer and Information Technology (ICCIT). IEEE, 2023, pp. 1–6.

[60] P. Saini, A. Dixit, and D. K. Sharma, "Enhancing object detection in adverse weather for autonomous driving with yolov9." International Energy Journal, vol. 25, 2025.

[61] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han et al., "Yolov10: Real-time end-to-end object detection," Advances in Neural Information Processing Systems, vol. 37, pp. 107 984–108 011, 2024.

[62] T. Mostafa, S. J. Chowdhury, M. K. Rhaman, and M. G. R. Alam, "Occluded object detection for autonomous vehicles employing yolov5, yolox and faster r-cnn," in 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2022, pp. 0405–0410.

[63] S. Fang, B. Zhang, and J. Hu, "Improved mask r-cnn multi-target detection and segmentation for autonomous driving in complex scenes," Sensors, vol. 23, no. 8, p. 3853, 2023.

[64] Z.-p. JIANG, Z.-q. WANG, Y.-s. ZHANG, Y. YU, B.-b. CHENG, L.-h. ZHAO, and M.-w. ZHANG, "A vehicle object detection algorithm in uav video stream based on improved deformable detr," Computer Engineering & Science, vol. 46, no. 01, p. 91, 2024.

[65] S. S. Reddy, M. Janarthanan, and I. U. Khan, "Rt-detr with attention-free mechanism: A step towards scalable and generalizable traffic sign recognition," SGS-Engineering & Sciences, vol. 1, no. 2, 2025.

[66] Y. Guo, Y. Yamamoto, H. Yaginuma, and Y. Taniguchi, "Vehicle detection in cctv with global-guided self-attention and convolution," Complex & Intelligent Systems, vol. 11, no. 10, p. 458, 2025.

[67] A. Vijayakumar and S. Vairavasundaram, "Yolo-based object detection models: A review and its applications," Multimedia Tools and Applications, vol. 83, no. 35, pp. 83 535–83 574, 2024.

[68] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using yolo: challenges, architectural successors, datasets and applications," multimedia Tools and Applications, vol. 82, no. 6, pp. 9243–9275, 2023.

[69] B. Zhang, M. Simsek, M. Kulhandjian, and B. Kantarci, "Enhancing the safety of autonomous vehicles in adverse weather by deep learning-based object detection," Electronics, vol. 13, no. 9, p. 1765, 2024.

[70] F. Sezgin, D. Vriesman, D. Steinhauser, R. Lugner, and T. Brandmeier, "Safe autonomous driving in adverse weather: Sensor evaluation and performance monitoring," in 2023 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2023, pp. 1–6.

[71] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu, "An improved faster r-cnn for small object detection," Ieee Access, vol. 7, pp. 106 838–106 846, 2019.

[72] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster r-cnn," Applied Sciences, vol. 8, no. 5, p. 813, 2018.

[73] M. Hahner, C. Sakaridis, D. Dai, and L. Van Gool, "Fog simulation on real lidar point clouds for 3d object detection in adverse weather," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 15 283–15 292.

[74] V. Kilic, D. Hegde, A. B. Cooper, V. M. Patel, and M. Foster, "Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection," in ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025, pp. 1–5.

[75] T. Prasanth, R. P. Padhy, and B. Sivaselvan, "Pcnet3d: A pillar based cascaded 3d object detection model using lidar point cloud," in International Conference on Computer Vision and Image Processing. Springer, 2024, pp. 12–24.

[76] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 196, pp. 146–177, 2023.

[77] M. Kutila, P. Pyykönen, M. Jokela, T. Gruber, M. Bijelic, and W. Ritter, "Benchmarking automotive lidar performance in arctic conditions," in 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2020, pp. 1–8.

[78] H. Delecki, M. Itkina, B. Lange, R. Senanayake, and M. J. Kochenderfer, "How do we fail? stress testing perception in autonomous vehicles," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022, pp. 5139–5146.

[79] K. Garg and S. K. Nayar, "Vision and rain," International Journal of Computer Vision, vol. 75, pp. 3–27, 2007.

[80] T. Brophy, D. Mullins, A. Parsi, J. Horgan, E. Ward, P. Denny, C. Eising, B. Deegan, M. Glavin, and E. Jones, "A review of the impact of rain on camera-based perception in automated driving systems," IEEE Access, 2023.

[81] Q. Al-Haija, M. Gharaibeh, and A. Odeh, "Detection in adverse weather conditions for autonomous vehicles via deep learning. ai 2022, 3, 303-317," 2022.

[82] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," IEEE vehicular technology magazine, vol. 14, no. 2, pp. 103–111, 2019.