Attention to Threat-Relevant Objects: Reasoning Detection in Autonomous Driving via Multimodal Large Language Models

Yulin He*, Wei Chen†*, Xinbiao Gan, Siqi Wang†, Haotian Wang, Yusong Tan

School of Computer, National University of Defense Technology, Changsha, China
{heyulin, chenwei, xinbiaogan, wangsiqi10c, wanghaotian, ystan}@nudt.edu.cn

Abstract

Perceiving threats is an innate human instinct. During driving, humans naturally focus their attention on objects that pose real potential risks. Motivated by this observation, we shift the focus from traditional class-based detection to a novel task termed **threat-oriented reasoning detection** in autonomous driving. This task aims to localize threat objects and reason about their threat levels from a driver-centric perspective. To support this task, we build a benchmark comprising diverse corner-case scenarios, annotated by multiple experienced drivers to reflect human-aligned threat cognition. Given the reasoning demands of this task, we then explore the capabilities of multi-modal large language models (MLLMs) and introduce two methods based on whether the MLLM supports object detection: 1) For MLLMs lacking detection capability, we introduce **ThreatCoT**, a plug-and-play training-free method that combines chain-of-thought (CoT) with a visual expert toolchain to support step-by-step reasoning. 2) For MLLMs with detection support, we introduce **ThreatReasoner**, an end-to-end reinforcement learning (RL)-based method built on the GRPO algorithm, which enables per-object reasoning through a fully unsupervised reward strategy. Both quantitative and qualitative experiments show that our methods can effectively unlock the new capabilities of MLLM in threat-oriented reasoning detection.

Code — https://github.com/harrylin-hyl/Threat-ReasonDet

Introduction

Object detection is a core perception task in autonomous driving and has received extensive attention (Arnold et al. 2019; Feng et al. 2021). Beyond common classes like cars, recent works on Out-of-Distribution (OOD) detection (Liu et al. 2023) and Open-World Object Detection (OWOD) (He et al. 2024) have explored the detection of corner cases and unknown obstacles that may pose serious threats to driving safety. However, as illustrated in Fig. 1(a), existing class-based detection models often fail to capture human driving preferences, *i.e.*, **the tendency to focus on objects that pose real threats**. These models rely on class semantics for object localization but lack the ability to reason from a driver's perspective and assess threat levels accordingly. To bridge this gap, we introduce a new detection task: **Threat-oriented Reasoning Detection**. Instead of detecting objects purely by class, this task requires models to mimic human threat cognition to identify potentially threat objects and assess their threat levels, as shown in Fig. 1 (b). In fact, class prediction is implicitly embedded within the reasoning process of models and is directly related to threat assessment. Moreover, thanks to the progress in foundation models (Radford et al. 2021) and multimodal large language models (MLLMs), object classification is not a main technical bottleneck. Instead, the key challenge now lies in understanding human preferences through contextual reasoning, which is a core objective for threat-oriented reasoning detection.

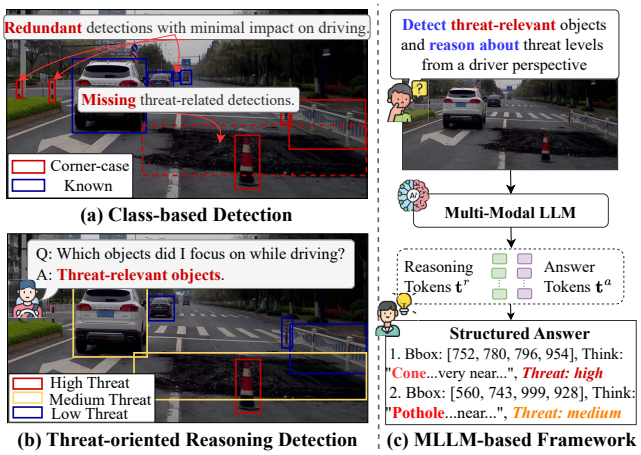


Figure 1: Illustration of threat-oriented reasoning detection. (a) shows the limitations of class-based detection in capturing human preferences, *e.g.*, redundant detections and missed threat-relevant objects. (b) presents the core idea that focuses on objects that pose real threats. (c) provides an overview of MLLM-based framework.

However, establishing a well-designed benchmark for this task is non-trivial due to three key difficulties: collecting diverse threat scenarios, obtaining consistent human-preference annotations, and designing suitable evaluation metrics. To this end, we first re-organize a dataset from mul-

*These authors contributed equally.
†Corresponding author

tiple open-source datasets (Li et al. 2022; Sun et al. 2020; Wilson et al. 2023; Geiger et al. 2013; Caesar et al. 2020), with a focus on corner-case scenarios due to their high threat potential. Second, we collect annotations from multiple experienced drivers to construct a trustworthy annotation distribution that reflects human cognitive preferences towards threat. Third, we propose a conditional recall (CRecall) metric that limits predicted boxes to the number of ground-truth annotations, effectively evaluating the accuracy of threat object detection. We also introduce L1 and L2 distance metrics to evaluate the accuracy of threat level estimation.

After establishing the benchmark, we explore the potential of MLLMs due to their impressive multi-modal reasoning ability. The overall framework is shown in Fig. 1(c). For MLLMs lacking detection capability (*e.g.*, InternVL3 (Zhu et al. 2025)), we adopt a modular framework; for those with such capability (*e.g.*, Qwen2.5VL (Bai et al. 2025)), we employ an end-to-end framework. However, experimental results show that existing MLLMs struggle to accurately understand threats (see Tab. 1 and Tab. 2). Therefore, we propose solutions for these two frameworks, respectively. For the modular framework, we introduce **Threat-CoT**, a plug-and-play, training-free method that combines Chain-of-Thought (CoT) with a visual expert toolchain to perform step-by-step threat reasoning from the image level to the object level. The toolchain consists of three visual experts: GroundingDINO (Liu et al. 2025a) for object detection, DepthAnythingV2 (Yang et al. 2024) for depth estimation, and SAM (Kirillov et al. 2023) for segmentation. These models extract object-level attributes such as location, class, and depth, serving as critical references for threat estimation. For the end-to-end framework, we propose **ThreatReasoner**, an RL-based method built on the GRPO algorithm (Shao et al. 2024; Guo et al. 2025), which enables self-organized per-object reasoning. ThreatReasoner uses a fully unsupervised reward strategy, including a format reward for structured output, a non-repeat reward to encourage diverse reasoning across objects, and a prediction number reward to counteract the trivial one-box outputs potentially induced by the non-repeat constraint.

We evaluate our methods on the constructed threat-oriented reasoning detection benchmark. For the modular framework, ThreatCoT improves threat object detection by 3% in Mean CRecall (M-CRecall) and reduces threat estimation error by 0.15 in Mean L1 (M-L1), compared to InternVL3. For the end-to-end framework, ThreatReasoner outperforms Qwen2.5VL with a 31.2% gain in M-CRecall and a 0.32 drop in M-L1 error. It also surpasses the recent reasoning model VisionReasoner (Liu et al. 2025c) by 9.7% in M-CRecall and 0.17 in M-L1 error, respectively.

Our contributions are fourfold: 1) We construct the first threat-oriented reasoning detection benchmark in autonomous driving by re-organizing and re-annotating datasets, along with introducing appropriate evaluation metrics, aiming to bridge the gap between threat-relevant perception and reasoning. 2) We propose Threat-CoT, a plug-and-play, training-free method that guides MLLMs to perform step-by-step reasoning from the image level to the object level. 3) We propose ThreatReasoner, an end-to-end RL-based method that enables per-object reasoning through a fully unsupervised reward strategy. 4) Through extensive experiments and visualizations, we validate the effectiveness of our design and offer critical insights into the application of reasoning techniques in autonomous driving.

Related Work

Class-based Object Detection

Object detection (OD) plays a critical role in perception for autonomous driving (AD) (Arnold et al. 2019; Feng et al. 2021). However, AD is an open-world scenario, where corner cases or unknown objects frequently occur and pose safety risks. To address this challenge, research on Out-of-Distribution (OOD) detection and Open-World Object Detection (OWOD) in AD has gained increasing attention. Early work like Bayesian SegNet (Kendall, Badrinarayanan, and Cipolla 2015) used uncertainty estimation to identify unknown objects. VOS (Du et al. 2021) learned latent distributions to generate virtual outliers for OOD detection. UnSniffer (Liang et al. 2023) introduced a confidence scoring method based on known classes and used a negative energy suppression loss to handle background noise. ORDER (Singh et al. 2021) was the first to extend OWOD to AD, enhancing unknown object representation via feature mixing. AD-OWOD (He et al. 2024) introduced a dual-branch network and utilized GroundingDINO (Liu et al. 2025a) to detect unknown objects using a predefined vocabulary bag. In contrast to previous works, this paper introduces a new task: threat-oriented reasoning detection.

Multimodal Large Language Model

Multi-modal Large Language Model (MLLMs) extend the capability of Large Language Models (LLMs) to understand image data while maintaining human-like dialogue and reasoning skills. Due to the practical value in real-world scenarios, extensive research (Li et al. 2023; Liu et al. 2024) has focused on effectively aligning multi-modal representations. Recent advancements in MLLMs, such as Qwen2.5VL (Bai et al. 2025), and InternVL3 (Zhu et al. 2025), have shown remarkable performance in tasks such as image analysis, OCR, and visual captioning. Building on these advanced capabilities, recent studies have applied MLLMs to AD (Cui et al. 2024; Huang et al. 2024). CODA-LM (Chen et al. 2024) is the most related work to ours, which introduced a benchmark for corner-case VQA with automated evaluation using LLMs. In contrast, our proposed threat-oriented reasoning detection task differs in two key aspects: 1) It requires models to perform both perception and reasoning simultaneously. 2) It simplifies the VQA task into a quantitative threat estimation task, making it easier to align human cognitive preferences towards threats.

Reasoning Perception

Reasoning perception aims to perceive objects based on implicit textual instructions. While similar in formulation to referring segmentation, it is more challenging due to the need for deeper understanding and reasoning. LISA (Lai et al. 2024) pioneered reasoning segmentation by fine-tuning

a pre-trained MLLM to generate segmentation tokens for SAM (Kirillov et al. 2023). VISA (Yan et al. 2024) extended reasoning segmentation to videos with an added tracking module. Recently, Seg-Zero (Liu et al. 2025b) applied RL-based GRPO algorithm into this task, enhancing the inherent reasoning capability of MLLMs. VisionReasoner (Liu et al. 2025c) further addressed multi-object prediction problem using the Hungarian algorithm. In this work, we introduce reasoning perception into AD, aiming to bridge the gap between threat-relevant perception and reasoning.

Threat-ReasonDet

Problem Definition

Given an input image $\mathbf{x}_{img}$ and a paired textual input $\mathbf{x}_{txt}$, threat-oriented reasoning detection (Threat-ReasonDet) is designed to mimic human-like threat recognition during driving. It outputs bounding boxes of threat objects, denoted as $\mathbf{y}_{bbox} \in \{\mathbf{B}_i \mid \mathbf{B}_i = \{(x_i^1, y_i^1), (x_i^2, y_i^2)\}\}$, where (x^1, y^1) and (x^2, y^2) represent top-left and bottom-right corners of bounding boxes, respectively. Simultaneously, it outputs a threat level for each object, denoted as $\mathbf{y}_{threat} \in \{\text{High}, \text{Medium}, \text{Low}\}$. Unlike general reasoning tasks where $\mathbf{x}_{txt}$ may vary in intent, the textual input in Threat-ReasonDet consistently instructs the model to identify threat objects and assess their threat levels. However, this consistency does not simplify the task. On the contrary, Threat-ReasonDet poses greater challenges due to its inherently ambiguous semantics and the need for deep per-object reasoning in complex traffic scenes. For example, cases like "a car merging lanes" or "a partially obscured pedestrian crossing the street". These cases require taking the driver's perspective, thoroughly analyzing the surrounding environment, and carefully reasoning about each individual object.

Dataset and Annotation

Given the lack of quantitative evaluation, establishing a benchmark for the Threat-ReasonDet task is essential. We first re-organized a diverse set of images from CODA (Li et al. 2022), Waymo (Sun et al. 2020), Argoverse2 (Wilson et al. 2023), Kitti (Geiger et al. 2013), and nuScenes (Caesar et al. 2020) datasets, focusing on corner-case scenarios that pose serious risks to driving safety. However, obtaining reliable annotations for this task is non-trivial, as a single agreed-upon label is often unavailable due to inherent individual biases. For example, a progressive driver might view the cone on the road in Fig. 1 (b) as a medium threat, while a cautious driver might label it as a high threat, both of which we consider reasonable. To better reflect such diverse cognitive preferences, we construct a distribution based on annotations from multiple individuals rather than relying on a single-person label. To further ensure annotation quality, we adopt a debate strategy to revisit cases with large discrepancies, ultimately reaching a consensus. This process helps to reduce individual major biases while preserving reasonable minor differences in annotations.

Evaluation

Threat-ReasonDet involves two subtasks: threat object detection and threat level estimation. Accordingly, we design two sets of metrics to evaluate their performance separately.

Threat Object Detection. Average Precision (AP) is a widely used metric for evaluating detection accuracy, but it is not well-suited for Threat-ReasonDet, which categorizes objects by threat levels rather than by mutually exclusive semantic classes. For example, if a predicted bounding box matches the ground-truth location but assigns a medium threat level instead of the ground-truth high level, AP would treat this as entirely incorrect. However, in Threat-ReasonDet, such a prediction should incur a milder penalty, as it is only one level off and still reflects threat-relevant detection. Therefore, we propose a brand-new metric called conditional recall (CRecall), which better evaluates threat-relevant detection performance. CRecall relaxes the strict binary penalties in AP by reforming the task: recalling as many threat objects as possible using no more predictions than the number of ground-truths. Predictions are ranked by threat levels and pruned by rank. CRecall is computed as:

$$\text{CRecall}\,@\tau = \frac{|\mathcal{P}_{\text{matched}}(\tau)|}{|\mathcal{G}|}, \quad \text{subject to } |\mathcal{P}| \leq |\mathcal{G}| \tag{1}$$

$$\mathcal{P}_{\text{matched}}(\tau) = \{\mathbf{p}_i \in \mathcal{P} \mid \exists \mathbf{g}_j \in \mathcal{G}, \text{IoU}(\mathbf{p}_i, \mathbf{g}_j) \geq \tau\}. \tag{2}$$

where $\mathcal{G}$ and $\mathcal{P}$ are the sets of ground-truth and prediction, respectively. τ is the IoU threshold set to 0.5.

Threat Level Estimation. In addition to detection, Threat-ReasonDet also requires the model to predict threat levels. Since misclassifying a high-threat object as medium or low should incur different penalties, threat level estimation is more akin to a regression problem than a classification task. Therefore, we use both L1 and L2 distances to evaluate predictions across threat levels. Compared to L1, the L2 distance imposes a more severe penalty on predictions that exhibit large deviations from the ground-truths.

Method

In this section, we present the proposed ThreatCoT and ThreatReasoner methods, designed for the modular and end-to-end frameworks, respectively.

ThreatCoT Method

For MLLMs that do not support object detection (Zhu et al. 2025; Dubey et al. 2024), we propose ThreatCoT, which employs a modular framework to query threat-relevant knowledge from MLLMs. As shown in Fig. 2, ThreatCoT consists of image-level threat reasoning, visual expert toolchains, and object-level threat reasoning.

Image-level Threat Reasoning. Given input image I, we design an image-level prompt, *i.e.*, "Use a paragraph to identify objects that pose a threat to your driving", denoted as Q_I, to query the MLLM for a image-level threat-relevant information G_I:

$$G_I = \text{MLLM}(I, Q_I). \tag{3}$$

We then extract the nouns of threat objects N_O from G_I.

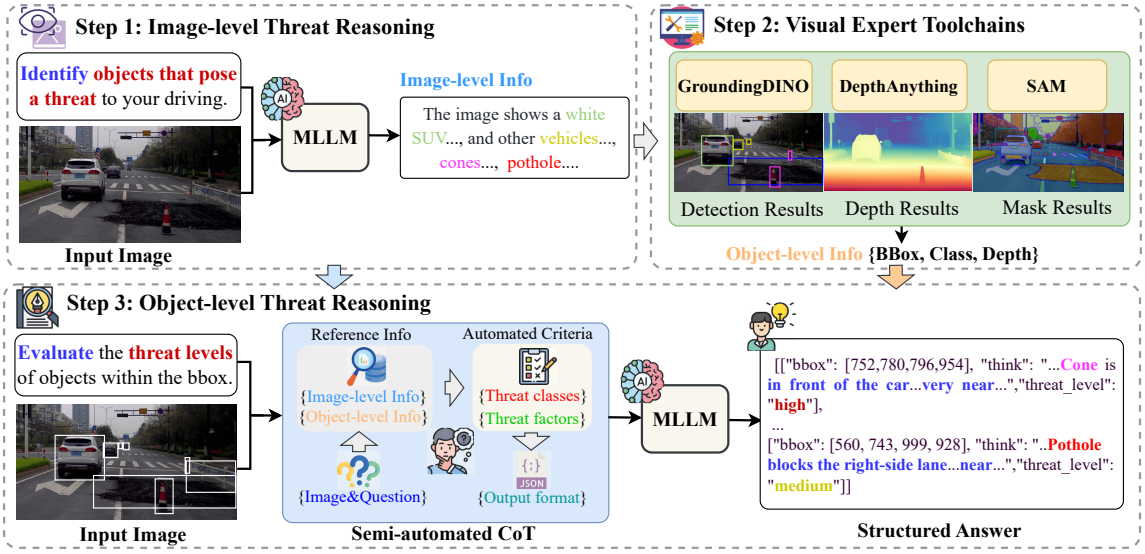


Figure 2: Overview of ThreatCoT. We first query the MLLM to identify threat objects and extract image-level information. This information, along with the input image, is then passed through toolchains to obtain object-level details. Finally, combining both image- and object-level information, we query the MLLM to estimate the threat levels of the detected objects.

Visual Expert Toolchains. To help MLLMs accurately output bounding boxes and recognize object attributes, we introduce toolchains that leverage existing visual experts. Given an input image I and the nouns N_O, the detection expert GroundingDINO (Liu et al. 2025a) generates bounding boxes of threat objects: $B = \text{GroundingDINO}(I, N_O)$. Simultaneously, the depth estimation expert DepthAnythingV2 (Yang et al. 2024) produces a depth map from the input image I: $D = \text{DepthAnything}(I)$. The bounding boxes B are then fed into the segmentation expert SAM (Kirillov et al. 2023), along with the image I, to generate masks of threat objects: $M = \text{SAM}(I, B)$. The depth of i-th object is calculated by element-wise multiplication of its mask and the depth map, followed by averaging: $D_O^i = \text{average}(M^i \cdot D)$. The object-level information is represented as $G_O = \{N_O, B, D_O\}$, which helps MLLMs to analyze the threat levels of objects.

Object-level Threat Reasoning. To minimize individual cognitive biases in prompting, we design a semi-automated CoT prompting method for threat level estimation. We manually establish the overall prompting pipeline, which includes image-level information G_I, object-level information G_O, threat classes T_C, threat factors T_F, and an output template F. The object-level query prompt can be formed as $Q_O = \{G_I, G_O, T_C, T_F, F\}$. As to the details of prompt design in T_C and T_F, we automate their generation using the MLLM itself. For T_C, we list the general categories of threat levels and then query MLLMs to provide a more detailed description of each threat level from the perspective of driver attention. For T_F, we ask MLLMs to generate ten questions to guide the evaluation of threat levels. Finally, we ask MLLMs to generate the structured answer A:

$$A = \text{MLLM}(I, Q_O), \tag{4}$$

where A includes thinking texts A_T and threat levels A_L.

ThreatReasoner Method

With advances in MLLMs, models such as Qwen2.5VL (Bai et al. 2025) now support object detection, paving the way for an end-to-end framework in Threat-ReasonDet. Inspired by recent RL techniques like GRPO (Guo et al. 2025) for enhancing LLM reasoning, we propose ThreatReasoner, which leverages a fully unsupervised reward strategy to unlock the per-object threat-relevant reasoning capabilities of MLLMs. Fig. 3 illustrates the overall pipeline.

Preliminaries of GRPO. Group Relative Policy Optimization (GRPO) (Shao et al. 2024) is an advanced RL algorithm designed to enhance policy optimization by incorporating group-wise relative comparisons. GRPO first samples a batch of prompts and generate a set of completions $G \in \{o_1, o_2, ...o_G\}$, each containing both reasoning and answer tokens. For each o_i, we compute the reward using the rule-based reward strategy $r_i = \mathcal{R}(o_i)$ and normalize it to obtain the advantage of the candidate response o_i:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, ..., r_G\})}{\text{std}(\{r_1, r_2, ..., r_G\})} \tag{5}$$

GRPO encourages the model to generate high-advantage responses within each group while ensuring that the model π_θ remains close to the reference policy π_{ref}. Consequently, the optimize objective is defined as follows:

$$\begin{aligned} &\mathcal{L}_{GRPO}(\theta) = \mathbb{E}\left[\{o_i\}_{i=1}^{N} \sim \pi_{\theta_{old}}(q)\right] \\ &\frac{1}{N}\sum_{i=1}^{N}\left\{\min\left[d_1 \cdot \hat{A}_i, d_2 \cdot \hat{A}_i\right] - \beta \mathbb{D}_{KL}\left[\pi_\theta || \pi_{ref}\right]\right\} \end{aligned} \tag{6}$$

$$d_1 = \frac{\pi_\theta(o_i \mid q)}{[\pi_\theta(o_i \mid q)]_{\text{no grad}}}, \quad d_2 = \text{clip}\left(d_1, 1-\epsilon, 1+\epsilon\right) \tag{7}$$

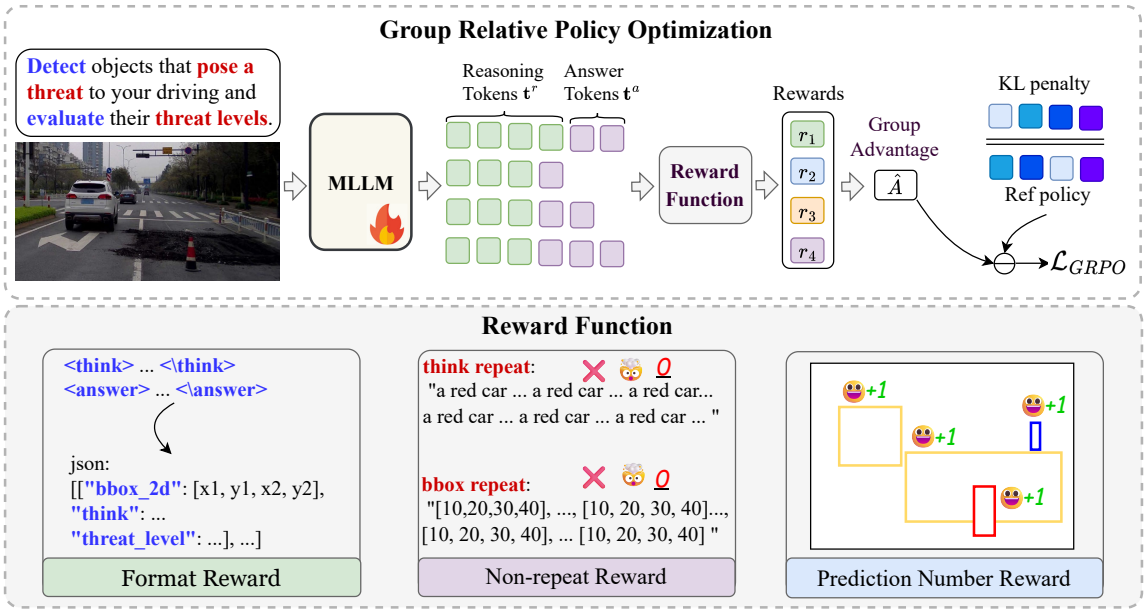


Figure 3: Overview of ThreatReasoner. Given the input image and instruct text, MLLM generates tokens in <think> and <answer> tags, guided by a rule-based reward strategy. A Kullback–Leibler (KL) divergence regularization term is also introduced to penalize deviations of the updated model from the reference policy.

The first term in Eq. 6 represents the scaled advantage, while the second term penalizes deviations of π_θ from π_{ref} via KL divergence. ϵ clips extreme advantages for stability.

Reward Functions. ThreatReasoner employs a fully unsupervised reward strategy with three reward functions:

1) **Format Reward.** This reward guides the model to output a reasoning process within <think> and </think> tags, and a JSON-format final answer within <answer> and </answer> tags, containing "bbox_2d", "think", and "threat_level" keys for each prediction.

2) **Non-repeat Reward.** Repeated outputs often suggest a lack of careful thinking and may generate hallucinations. To address this issue, we introduce a non-repeat reward. For both image-level and object-level reasoning content, we split it into sub-sentences and prioritize those that are non-redundant. We also extract bounding boxes from the answers and penalize duplicated locations. This reward encourages the model to reason uniquely for each object, promoting more deliberate and context-aware analysis.

3) **Prediction Number Reward.** Although the non-repeat reward helps per-object reasoning, it often leads to trivial one-box outputs, as a single prediction naturally avoids repetition. Therefore, we introduce the prediction number reward to encourage multiple predictions. However, producing too many predictions can result in overly long outputs and compromise the accuracy of individual objects. To balance this trade-off, we adopt a piecewise linear reward function that softly constrains the prediction number:

$$\mathcal{R}_{\text{num}} = \begin{cases} \frac{n}{N}, & 1 \le n \le N \\ 2 - \frac{n}{N}, & N < n \le 2N \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Method	1 (%) ↑	2 (%) ↑	3 (%) ↑	Mean (%) ↑
Modular Framework				
Qwen2.5VL	36.1±3.2	56.3±3.2	71.9±1.7	54.7±2.7
Qwen+ThreatCoT	35.7±5.0	63.1±3.2	78.5±1.8	59.1±3.3 (↑ **4.4**)
InternVL3	46.3±1.2	65.7±3.0	78.6±1.9	63.5±2.0
Intern+ThreatCoT	45.0±4.0	71.1±3.2	83.4±2.3	**66.5**±3.1 (↑ **3.0**)
End-to-end Framework				
Qwen2.5VL	13.6±3.1	32.4±1.9	51.4±3.7	32.4±2.9
VisionReasoner	34.3±4.2	55.5±4.7	72.1±2.1	53.9±3.6
ThreatReasoner	42.3±4.2	66.5±5.4	82.0±0.5	**63.6**±3.3 (↑ **9.7**)

Table 1: Threat object detection benchmark. Evaluation reports CRecall scores for high (3), medium (2), and low (1) threat levels, as well as overall mean.

where n denotes the number of predictions, and N is the breakpoint of $\mathcal{R}_{\text{num}}$ heuristically set to 5 based on the average prediction count of Qwen2.5VL (Bai et al. 2025) model.

Experiment

Experimental Settings

Datasets. The test dataset of Threat-ReasonDet consists of 936 scene images, with an average of 4,550 threat objects annotated by five experienced drivers. It includes 500 images from CODA test set (Li et al. 2022), 122 from Waymo (Sun et al. 2020), 201 from Argoverse2 (Wilson et al. 2023), 69 from KITTI (Geiger et al. 2013), and 44 from nuScenes (Caesar et al. 2020), all carefully selected as corner-case scenarios. For training the ThreatReasoner model, we randomly sample 1,000 unlabeled images from

Method	L1 Distance					L2 Distance
	0 ↓	**1 ↓**	**2 ↓**	**3 ↓**	**Mean ↓**	**Mean ↓**
Human	0.31±0.29	0.3±0.16	0.43±0.09	0.36±0.05	0.37±0.05	0.37±0.05
Modular Framework						
Qwen2.5VL	0.23±0.04	0.78±0.02	1.15±0.03	1.55±0.06	0.93±0.01	1.68±0.03
Qwen+ThreatCoT	0.22±0.05	0.98±0.02	1.03±0.03	0.95±0.03	0.80±0.02 **(↓ 0.13)**	1.38±0.05 **(↓ 0.30)**
InternVL3	0.51±0.10	0.97±0.05	1.01±0.02	0.85±0.06	0.84±0.03	1.45±0.08
Intern+ThreatCoT	0.23±0.05	0.83±0.01	0.74±0.05	0.95±0.05	**0.69**±0.03 **(↓ 0.15)**	**1.12**±0.07 **(↓ 0.33)**
End-to-end Framework						
Qwen2.5VL	0.05±0.01	0.96±0.01	1.44±0.03	1.95±0.08	1.10±0.02	2.24±0.07
VisionReasoner	0.13±0.02	0.79±0.02	1.31±0.04	1.59±0.07	0.95±0.01	1.80±0.05
ThreatReasoner	0.31±0.05	0.80±0.01	0.81±0.08	1.18±0.03	**0.78**±0.02 **(↓ 0.17)**	**1.25**±0.05 **(↓ 0.55)**

Table 2: Threat level estimation benchmark. The evaluation reports L1 and L2 distances across varying threat levels: high (3), medium (2), low (1), and minor (0). "Human" is human performance obtained by cross-validation from multiple annotations.

Image-level Inf.	Object-level Inf.	M-CRecall (%)	M-L1	M-L2
		63.5	0.84	1.45
✓		62.8	0.80	1.37
✓	✓	**66.5**	**0.69**	**1.12**

Table 3: Ablation studies on ThreatCoT. Experiments are conducted using the InternVL3-8B model as the baseline.

Format	Non-repeat	Num.	M-CRecall (%)	M-L1	M-L2
			53.9	0.95	1.80
✓			58.3	0.91	1.66
✓	✓		51.8	1.06	2.08
✓	✓	✓	**63.6**	**0.78**	**1.25**

Table 4: Ablation studies on ThreatReasoner. Experiments are conducted using the VisionReasoner-7B model as the baseline. "Format", "Non-repeat", and "Num." are format, non-repeat, and prediction number reward functions.

the CODA validation set (Li et al. 2022).

Implementation Details. For the modular framework, we employ InternVL3-8B (Zhu et al. 2025) and Qwen2.5VL-7B (Bai et al. 2025) as the base models, and deployed them locally using vLLM (Kwon et al. 2023). For the end-to-end framework, we use Qwen2.5VL-7B as the base model and train ThreatReasoner on 8×L40 GPUs using the DeepSpeed library (Rasley et al. 2020). During training, we adopt a total batch size of 16 with 8 samples per training step. The initial learning rate is set to 1e-6, and the weight decay is 0.01. The default values for β in Eq. 6 and ϵ in Eq. 7 are set to 0.01 and 0.2, respectively. The pretrained weights of ThreatReasoner are from VisionReasoner-7B (Liu et al. 2025c).

Main Results

We compare ThreatCoT and ThreatReasoner methods with two recent open-source MLLMs (*i.e.*, Qwen2.5VL (Bai et al. 2025) and InternVL3 (Zhu et al. 2025)) and a recent reasoning MLLM (*i.e.*, VisionReasoner (Liu et al. 2025c)), under both modular and end-to-end frameworks. In the modular framework, Qwen2.5VL and InternVL3 rely on GroundingDINO (Liu et al. 2025a) to generate bounding boxes, while in the end-to-end framework, all models directly output bounding boxes and estimate threat levels. We also report human performance as a reference, obtained via cross-validation across multiple annotations.

Threat object Detection. As shown in Tab. 1, the proposed ThreatCoT significantly enhances recall accuracy, with improvements of 4.4% and 3.0% in Mean C-Recall over Qwen2.5VL and InternVL3, respectively. Furthermore, in the end-to-end framework, ThreatReasoner yields substantial gains of 31.2% and 9.7% in Mean C-Recall compared to Qwen2.5VL and VisionReasoner. These results demonstrate the effectiveness of our methods and highlight the importance of threat-relevant reasoning in threat object detection.

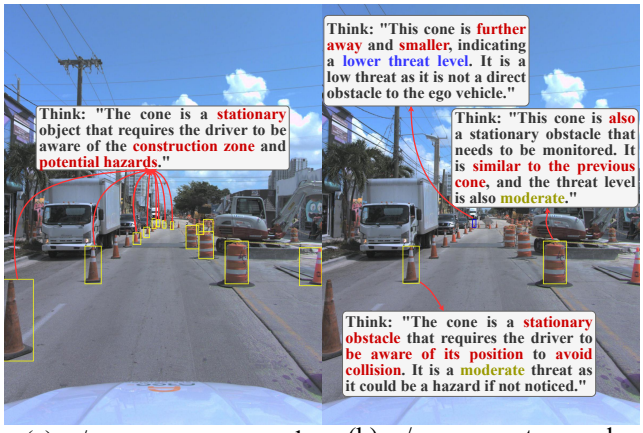
Threat level estimation. In addition to evaluating perception accuracy, we also compare the performance of threat level estimation. For a more comprehensive analysis of false positives, the evaluation also considers objects annotated as minor threats. As shown in Tab. 2, existing MLLMs struggle to accurately estimate threat levels when relying solely on their inherent capabilities, with a notable gap compared to human performance. These findings highlight the challenging nature of the Threat-ReasonDet task. Notably, significant performance gains are achieved by incorporating the proposed Threat-CoT and ThreatReasoner methods. In the modular framework, after applying Threat-CoT, QWen2.5VL reduces Mean L1 and L2 errors by 0.13 and 0.30, respectively, while InternVL3 achieves reductions of 0.15 and 0.33. In the end-to-end framework, ThreatReasoner outperforms QWen2.5VL by 0.32 in Mean L1 and 0.99 in Mean L2, and surpasses VisionReasoner by 0.17 and 0.55 in Mean L1 and L2, respectively.

Ablation Studies

We analyze the contributions of each component in ThreatCoT, with results shown in Tab. 3. Incorporating image-level information helps reduce errors in threat level estimation, while object-level cues (*e.g.*, class and depth) significantly improve accuracy in both threat object detection and threat

(a) One of GT (b) Qwen2.5VL (c) VisionReasoner (d) Qwen+ThreatCoT (e) ThreatReasoner

Figure 4: Visualization comparison with (a) one of ground truth, (b) Qwen2.5VL, (c) VisionReasoner, (d) Qwen+ThreatCoT, and (e) ThreatReasoner. Bounding boxes in red, yellow, and blue indicate high, medium, and low threat levels, respectively.



(a) w/o non-repeat reward (b) w/ non-repeat reward

Figure 5: Visualization comparison with and without the non-repeat reward.

level estimation. These results highlight the importance of spatial awareness (*e.g.*, depth) for the Threat-ReasonDet task, revealing a key limitation in current MLLMs.

We report the ablation study results of ThreatReasoner in Tab. 4. Introducing the format reward increases M-CRecall by 4.4%, while reducing M-L1 and M-L2 errors by 0.04 and 0.14, respectively. These results underscore the critical role of object-level reasoning via the answer format. However, as illustrated in Fig. 5(a), we observe that the model tends to generate repetitive reasoning for objects of the same class, revealing a lack of deep and differentiated understanding of individual objects. To address this issue, we introduce the non-repeat reward, which effectively reduces repetitive reasoning but unfortunately causes a significant drop in overall performance. This decline results from the non-repeat reward encouraging the model to predict fewer objects, leading to a degenerate solution where only one box is output to maximize the reward. Therefore, we further adopt the prediction number reward, which encourages the model to generate multiple predictions. With this combined reward, all metrics improve significantly: M-CRecall increases by 7.9%, and M-L1 and M-L2 errors are reduced by 0.15 and 0.44, respectively. Importantly, it drives the model to generate distinct object-specific reasoning, reflecting a deeper threat-relevant understanding, as shown in Fig. 5 (b).

Visualization Analysis

Fig. 4 shows the qualitative results of ground-truth and different methods. Our proposed ThreatCoT and ThreatReasoner demonstrate more comprehensive and accurate detection of threat objects, along with appropriate threat level estimations. For example, they correctly identify the "walking person" and "aerial work platform" in the first row, the "running dog" in the second, and the "car with headlights on at night" in the third. However, some inaccurate predictions remain. For instance, the "crane" in the third row is assigned a medium threat level by ThreatCoT and ThreatReasoner, and even a high threat level by Qwen2.5VL, whereas annotators consistently label it as minor or low threat since it is off the main road. This reveals that understanding inter-object relations in complex scenes remains highly challenging, which is an important direction for future work.

Conclusion

In this paper, we introduce the threat-oriented reasoning detection task, which models human driving preferences by focusing on threat-relevant objects. We first establish the benchmark by reorganizing and re-annotating datasets, and designing suitable evaluation metrics. Then, we propose two methods: ThreatCoT, a training-free plug-and-play method for MLLMs without detection capabilities; and ThreatReasoner, an end-to-end reinforcement learning method for MLLMs with such capabilities. Experimental results show that both methods significantly improve performance, providing valuable insights into the application of reasoning techniques in autonomous driving.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (No. 62476280), the Science and Technology Innovation Program of Hunan Province of China (No. 2024RC3137), and the National Key Research and Development Program of China (No. 2018YFB0204301).

References

Arnold, E.; Al-Jarrah, O. Y.; Dianati, M.; Fallah, S.; Oxtoby, D.; and Mouzakitis, A. 2019. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10): 3782–3795.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, K.; Li, Y.; Zhang, W.; Liu, Y.; Li, P.; Gao, R.; Hong, L.; Tian, M.; Zhao, X.; Li, Z.; et al. 2024. Automated evaluation of large vision-language models on self-driving corner cases. *arXiv preprint arXiv:2404.10595*.

Cui, C.; Ma, Y.; Cao, X.; Ye, W.; Zhou, Y.; Liang, K.; Chen, J.; Lu, J.; Yang, Z.; Liao, K.-D.; et al. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 958–979.

Du, X.; Wang, Z.; Cai, M.; and Li, Y. 2021. VOS: Learning What You Don't Know by Virtual Outlier Synthesis. In *International Conference on Learning Representations*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Feng, D.; Harakeh, A.; Waslander, S. L.; and Dietmayer, K. 2021. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 9961–9980.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11): 1231–1237.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

He, Y.; Wang, S.; Chen, W.; Xun, T.; and Tan, Y. 2024. Sniffing Threatening Open-World Objects in Autonomous Driving by Open-Vocabulary Models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9067–9076.

Huang, Z.; Feng, C.; Yan, F.; Xiao, B.; Jie, Z.; Zhong, Y.; Liang, X.; and Ma, L. 2024. Drivemm: All-in-one large multimodal model for autonomous driving. *arXiv preprint arXiv:2412.07689*.

Kendall, A.; Badrinarayanan, V.; and Cipolla, R. 2015. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; and Jia, J. 2024. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9579–9589.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Li, K.; Chen, K.; Wang, H.; Hong, L.; Ye, C.; Han, J.; Chen, Y.; Zhang, W.; Xu, C.; Yeung, D.-Y.; et al. 2022. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European Conference on Computer Vision*, 406–423. Springer.

Liang, W.; Xue, F.; Liu, Y.; Zhong, G.; and Ming, A. 2023. Unknown sniffer for object detection: Don't turn a blind eye to unknown objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3230–3239.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2025a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 38–55. Springer.

Liu, Y.; Ding, C.; Tian, Y.; Pang, G.; Belagiannis, V.; Reid, I.; and Carneiro, G. 2023. Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1151–1161.

Liu, Y.; Peng, B.; Zhong, Z.; Yue, Z.; Lu, F.; Yu, B.; and Jia, J. 2025b. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*.

Liu, Y.; Qu, T.; Zhong, Z.; Peng, B.; Liu, S.; Yu, B.; and Jia, J. 2025c. VisionReasoner: Unified Visual Perception and Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2505.12081*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Rasley, J.; Rajbhandari, S.; Ruwase, O.; and He, Y. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 3505–3506.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Singh, D. K.; Rai, S. N.; Joseph, K.; Saluja, R.; Balasubramanian, V. N.; Arora, C.; Subramanian, A.; and Jawahar, C. 2021. ORDER: Open World Object Detection on Road Scenes. In *Proc. NeurIPS Workshops*.

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

Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J. K.; et al. 2023. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*.

Yan, C.; Wang, H.; Yan, S.; Jiang, X.; Hu, Y.; Kang, G.; Xie, W.; and Gavves, E. 2024. Visa: Reasoning video object segmentation via large language models. In *European Conference on Computer Vision*, 98–115. Springer.

Yang, L.; Kang, B.; Huang, Z.; Zhao, Z.; Xu, X.; Feng, J.; and Zhao, H. 2024. Depth anything v2. *Advances in Neural Information Processing Systems*, 37: 21875–21911.

Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; et al. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.