Monitoring Safety Properties for Autonomous Driving Systems with Vision-Language Models

Felipe Toledo[1], Sebastian Elbaum[1], Divya Gopinath[2], Ramneet Kaur[3], Ravi Mangal[4], Corina S. Păsăreanu[5], Anirban Roy[3], Susmit Jha[3]

***Abstract*— With the increased adoption of autonomous vehicles comes the need to ensure they reliably follow safe driving properties. Formally specifying and monitoring such properties is challenging because of the semantic mismatch between the high-level properties (e.g., assertions on spatial relationships between the ego vehicle and other entities in a road scene) and the sensed inputs of the vehicles (e.g., raw pixels). For this reason, existing monitoring methods are applicable in limited simulation settings where the ground-truth spatial relationships are available. To bridge this gap we investigate the use of Vision-Language Models (VLMs) for extracting spatial relationships from real images of driving scenes. Towards this goal, we automated the process to extract triplets of the form <subject, relation, ego> from real image datasets such as nuScenes, Waymo, and KITTI, to create DriST, a dataset of road-scene images annotated with corresponding triplets. We use DriST to evaluate the spatial reasoning capabilities of state-of-the-art VLMs in the driving domain. Our experiments show that, while standard VLMs have limited capability on this task, their performance measured using F1 score is significantly improved by fine-tuning from 0.56 to 0.93, showing the utility of DriST. We then incorporate the improved VLM into monitors of safety properties specified in formal temporal logic. The study shows the potential of the approach to detect most violations (27 out of 34) found with ground-truth data, and just four instances of false positives. We make our dataset, evaluation, and trained VLMs available at https://github.com/less-lab-uva/DriST.**

I. Introduction

The growing trend towards deploying autonomous driving systems (ADSs) on public roads has created an urgent need for developing methods to ensure the reliability of such systems. Existing systems are known to have unpredictable and risky behaviors that can lead to adverse outcomes, including loss of human and animal life [1], [2], [3], [4], [5]. While these systems are typically extensively validated through testing in simulation and in the field [6], [7], monitoring formal properties can offer a complementary approach to offer formal assurances [8], [9]. The idea is to first specify desired, safe system behaviors in a suitable formalism such as linear temporal logic on finite traces (LTL_f) [10] and then automatically translate them into *monitors* that can be used *off-line*, to check or retrieve streams of data collected during past operation, or *on-line* to check or enforce properties of a system during operation.

Driving Example. Many driving rules constitute informal specifications for ADSs. These properties typically refer to spatial relationships between entities in a road scene. Consider an ego vehicle equipped with a camera that captures images such as those shown in Figure 1. This vehicle must satisfy the following safety property to ensure reasonable distance to other vehicles: *"If there are any vehicles ahead within 25m of ego and ego speed is greater or equal than 25 mph for two consecutive time steps, then ego acceleration should be negative (braking) in the second time step"*. This property is formally specified in linear temporal logic at the top of Figure 1.

LTLf Property:

G((<vehicle, within25m, ego> ∧ egoSpeed ≥ 25 mph) ∧
X (<vehicle, within25m, ego> ∧ egoSpeed ≥ 25 mph) → X isBraking)

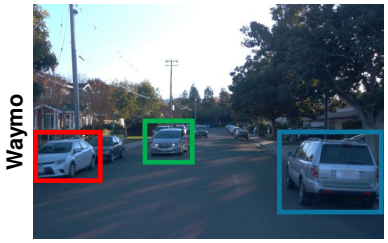


Fig. 1: LTL_f formula of ADS safety property (G, globally, and X, next, are LTL_f operators); and image with ground-truth triplets, encoding spatial relationships between ego car and objects in a scene.

Monitoring Challenge. A fundamental challenge in using formal reasoning for ADSs is the *semantic gap* between the sensor inputs (such as raw pixels in images collected with the cameras) and the high-level predicates, encoding spatial relationships between objects in a scene as in the example above. Existing monitoring works [8], [9] side-step the problem by assuming that ground-truth information about the spatial relationships (the precise pose of all other vehicles) is available for evaluating the semantic predicates. As a result, they are only applicable during simulation testing where it is possible to access the ground-truth but they cannot be deployed on real data streams collected from field operation.

Proposal. To address the challenge, we explore the use of Vision-Language Models (VLMs) for extracting spatial relationships from sensor inputs on driving scenes. VLMs [11] are powerful models trained on massive amounts of images and textual data and can be queried on vision and language modalities. In this initial exploration of VLMs viability to support monitoring, we aim to capture the *spatial relationships* found in images as triplets of the form *<subject, relation, ego>*. Figure 1 shows an image from the Waymo dataset accompanied by such ground-truth triplets where the intended property precondition holds for at least one vehicle.

Figure 2 gives an overview of the paper, highlighting our approach and three key contributions.

1) **DriST (Driving Scenes with Triplets) Dataset.** Starting with widely available autonomous driving datasets (nuScenes [12], Waymo [13], and KITTI [14]) containing driving scenes along

[1]University of Virginia, United States
[2]KBR, NASA Ames, United States
[3]SRI, United States
[4]Colorado State University, United States
[5]KBR, NASA Ames, Carnegie Mellon University, United States
*Work supported by NSF (2312487), U.S. Air Force, DARPA (FA8750-23-C-0519), U.S. Army (W911NF-17-2-0196), and Lockheed Martin.

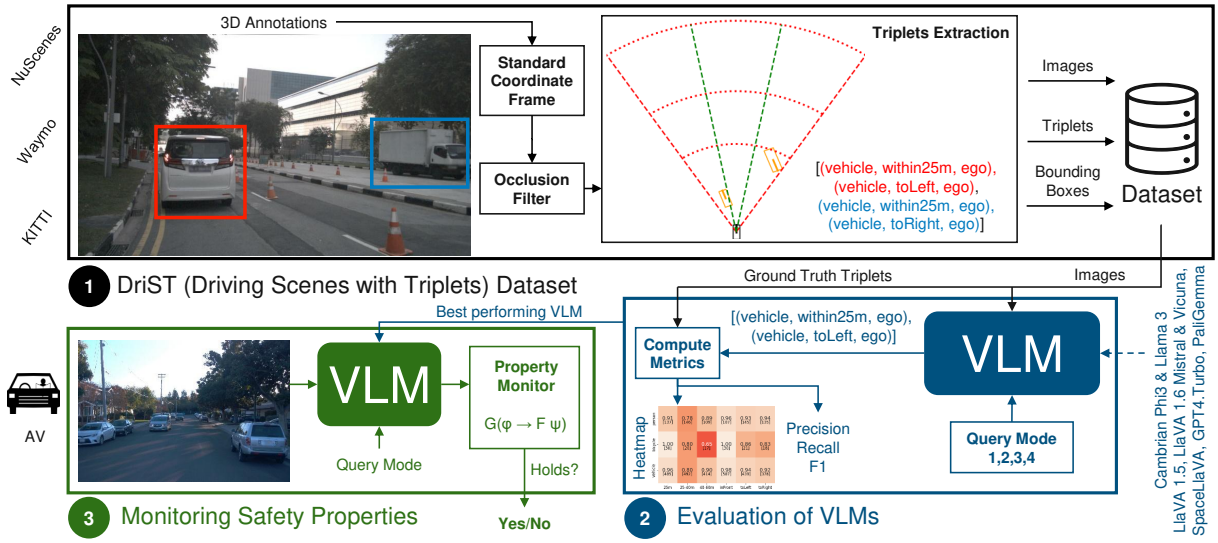


Fig. 2: Overview of paper contributions: (1) DriST dataset with images from nuScenes, Waymo, and KITTI, annotated with triplets; (2) Evaluation of state-of-the-art VLMs on DriST dataset using four query modes and producing *heatmaps* summarizing VLMs performance; (3) Monitoring of temporal logic properties with a VLM for evaluating triplets.

with 3D bounding box annotations and various meta-data, we built a dataset containing driving scenes annotated with *spatial relationships*. This process entailed developing transforms to map the original data to a standard coordinate frame based on ego, removing occlusions, and extracting triplets from the image. This dataset can be used to evaluate spatial reasoning of VLMs and also to finetune and improve VLMs for autonomous driving.

2) **Evaluation of VLMs.** Using the DriST dataset and four prompting strategies, we investigate the limitations of eight state-of-the-art VLMs for spatial reasoning. We also demonstrate the potential for improved performance through fine-tuning, raising the F1 values for a chosen model from 0.6 to 0.93.

3) **Monitoring Safety Properties.** We integrate the best performing finetuned VLM from the evaluation into monitors for safety properties specified in linear temporal logic. We then assess their performance over 199 Waymo sequences containing more than 39k frames. The results show that the monitors were able to find a total of 27 out of 34 violations. We make our implementations of the 3 contributions available at https://github.com/less-lab-uva/DriST

II. Related Work

Visual Language Models (VLMs) [11] possess a joint understanding of both visual and textual information enabling multiple vision-language tasks. This is an active area of research with a plethora of models [15], [16], [17], [18], [19], [20], [21], [22], [23], with increasing interest in spatially-aware models [24], [25]. Recent work that is close to our dataset creation contribution [26] presents a large, diverse, and spatially aware dataset to pre-train models for multi-turn question answering. However, this dataset is not yet publicly available. Unlike this work, DriST is specifically curated to detect spatial relationship triplets in road scenes.

Our work is also related to scene-graph (SG) generation [27]. Although we do not build scene graphs explicitly, our proposed triplets can be assembled such graphs. Recent works [28], [29], [30] investigate how to use VLMs to generate scene graphs. Notably, the work in [30] employs a VLM to transform images into a sequence of relation-aware tokens and convert them into scene graphs. We use a different methodology to extract SGs from images, which consists of asking VLMs to generate a list of all the triplets in the image and then using these triplets to check safety properties via monitors. Unlike [30], we evaluate a wider range of VLMs, focus on road scenes, and use it as a part of a monitor.

A survey [31] categorizes monitoring systems based on how they are integrated with the perception pipeline. Among the listed categories, we focus on the one that inspects inputs and validates outputs, where monitors operate independently of the inner-workings of the perception system.

Several approaches have been introduced to monitor image streams. The work in [32] proposes a monitor that checks properties defined in Timed Quality Temporal Logic (TQTL) over bounding boxes produced by a perception pipeline. Similarly Anderson et al. [33] proposes Spatial Regular Expressions (SpREs) as a querying language for bounding boxes pattern matching over perception streams containing spatial and temporal data. Nonetheless, the spatial relationships in both works are constrained to the 2D plane, limiting the expressiveness of these properties to measurements of bounding box overlap, and rely on the perception pipeline's output, so faults in the pipeline also limit the monitor's effectiveness. Elhafsi et al. [34] also use bounding boxes to describe a scene to an LLM to detect semantic anomalies in robots. However, it does not support the monitoring of temporal-logic safety properties. In contrast, the approach in [8] builds SG with 3D spatial relationships as abstractions of the images, enabling the monitoring of Linear Temporal Logic (LTL) properties like the ones in the VA driving manual [35]. However, it assumes that the SGs are made available by a simulator, limiting its applicability in real-world scenarios. In this work, we leverage VLMs to extract SGs from real images and use a monitoring framework, similar to the one in [8], to check LTL properties on real world scenarios.

III. The DriST Dataset

The DriST dataset contains images and their corresponding relationship triplets. This section outlines the triplets format, the generation procedure, and the resulting dataset.

A. Scope of Road Scenes Triplets

A triplet of the form <*subject, relation, ego*> captures a single spatial relationship between an object and the ego vehicle, providing a precise description of how the two entities are positioned relative to each other.

The list of potential subjects and relations has a long-tail distribution with many rare instances. In this work, we focus on the head of that distribution. That is, we prioritize entities and relations frequently found in existing datasets and related to safety driving requirements. More specifically, we focus on three types of **subjects**: *vehicles*, *persons*, and *bicycles*. This scope also helps us to maintain compatibility across datasets; for instance, while nuScenes include a wider variety of entity types, Waymo only provides labels for the three subjects mentioned above.

We also focus on six **relations** that capture critical spatial interactions with the ego vehicle, organized into two categories: relative position and relative distance.

- Relative position describe the lateral positioning of entities within the ego vehicle's field of view, which is divided into three equal regions—*left of*, *in front of*, and *right of* the ego.
- Relative distance captures the proximity of entities to the ego vehicle, categorized into three ranges: *within 25 meters*, *between 25 and 40 meters*, and *between 40 and 60 meters*. These ranges correspond to safe braking distances when traveling at speeds of 25, 35, and 45 miles per hour [35], respectively, reflecting essential safety considerations.

We consider these subjects and relations to be sufficient to initially assess the potential of VLMs to capture critical spatial interactions in driving environments. However, we acknowledge their limitations such as not accounting for the road layout or the traffic signals, all of which influence the ego vehicle's behavior. Future work will focus on extending the dataset to include these elements. These enhancements will allow for defining and monitoring more complex properties, thereby supporting the evaluation of a wider range of safety behaviors in autonomous driving contexts.

B. Process for Extracting Triplets

Autonomous driving datasets such as nuScenes, Waymo, and KITTI contain sensor data (images, LiDAR) accompanied by 3D annotations for the entities in each scene. However, none of them include annotations of spatial relationships between entities and the ego vehicle. As depicted in Fig 2 (1), to generate triplets from the 3D annotations in each dataset, we developed an automated process that enables us to represent each scene in a common coordinate frame, perform necessary occlusion filtering, extract ground truth scene graphs as list of triplets, and associate a bounding box to each subject, to localize it in the image. We share the code in our repository to enable the application of this process to other datasets.

Standard Coordinate System. Each dataset uses a unique coordinate system for 3D bounding boxes, often based on sensor placement or proprietary conventions. To unify these diverse annotations, we convert all 3D bounding boxes to a common North-East-Up (NEU) coordinate system based on the right-hand rule. The origin of this coordinate system is set at the ego vehicle's center, allowing a consistent spatial reference across all scenes.

Occlusion Filter. Spatial reasoning in autonomous driving camera scenes must distinguish visible objects. To determine whether an entity is visible we use ray tracing from the ego vehicle to each entity in the scene. If a ray intersects with any other entity's bounding box, we mark the target entity as occluded and remove it from the list of entities in the scene.

Triplets Extraction. Using the field of view of the camera and a distance threshold of 60 meters (red cone in Fig 2-1), we prune entities whose centroids fall outside this area. We then use the sections of the cone, parameterized based on the relations definitions, to determine the distance and position of the entity relative to ego. For example, in Fig 2-1, there is a vehicle (white van) on the left and within 25m, as we can see that it is in the left bottom sub-area of the red cone. There is also another vehicle (white truck) on the right and within 25m as well, as it is depicted in the right bottom sub-area.

Finally, we associate each entity in the scene with its corresponding bounding box, representing each data point as an image, a list of triplets, and the bounding boxes. The bounding box provides two key advantages: first, it enables spatial grounding of the triplets, allowing visual validation of each entity's position and relationship in the image. Second, it offers additional contextual cues that aid in complex spatial reasoning and support advanced query formulation, as outlined in the next section.

C. The DriST dataset

DriST dataset contains **209,590** scenes labeled with triplets, with over 4 entities (2.9 vehicles, 1.6 persons, 0.06 bicycles) and 8 triplets per image on average, offering substantial diversity in driving scenarios. The most common relationships for position is in-front and for distance is between 40-60m, but all relationships appear on average of at least 1.2 times per image. This balanced distribution, combined with the diverse sensor configurations and camera resolutions in the nuScenes, Waymo, and KITTI datasets, provide us with a range of typical driving elements and interactions and realistic spatial contexts to study the potential of VLMs to help reason in this domain.

IV. Evaluating VLMs on the DriST dataset

To assess the effectiveness of VLMs in accurately capturing triplet relationships within autonomous driving scenes, we evaluate eight state-of-the-art models and four query strategies. We randomly sampled 300 images from the DriST test split for each of the three datasets, resulting in a total of 900 images for evaluation. Due to the operational cost of querying GPT-4 Turbo, for this model we further sampled 30 images from each 300-image subset, totaling 90 images for its evaluation. We made our code available at https://github.com/less-lab-uva/DriST

A. Models

We selected a range of models that represent distinct configurations in both language modeling and vision encoding. We include GPT-4 Turbo (GPT-4-T) [19] as a reference for top-tier multimodal performance, setting a high baseline for model comparison, and

RoadScene2Vec (RS2V) [36] as a scene graph generator for the driving domain. As a foundational model, LLaVA 1.5 (L1.5) [16] combines the popular CLIP [15] image encoder with Llama 2 [37]. We also include LLaVA 1.6 [38] Mistral (L1.6-Mis) and LLaVA 1.6 Vicuna (L1.6-Vic), both designed to handle higher image resolutions and fine-tuned on a more diverse dataset than LLaVA 1.5, while leveraging the latest advancements in their respective language models. Recognizing the importance of spatially-aware models, SpaceLLaVA [25] has been fine-tuned on spatial data with the objective of achieving better spatial understanding. Further diversity is introduced with PaliGemma [17], which leverages SigLIP [39] as an alternative vision encoder, potentially broadening the range of visual contexts the model can process. Lastly, the Cambrian Phi 3 (C-Phi3) [18] and Llama 3 [40] models incorporate the Spatial Vision Aggregator (SVA) to combine features from multiple vision encoders, thereby enabling more nuanced spatial reasoning through feature aggregation.

To enhance the spatial reasoning capabilities of VLMs specifically for autonomous driving scenarios, we experimented with using the DriST dataset to fine-tune and trained LoRAs [41] on LLaVA 1.5 (L1.5-FT, L1.5-L) in different query modes. We used 75% of the dataset for training, 5% for validation, and the remaining 20% for testing. While LLaVA 1.5 is somewhat dated, its repository benefits from extensive community support, and includes reliable scripts and documentation for efficient fine-tuning and LoRA training on GPUs. This model is also cost-effective; we fine-tuned our models usign 8 A40 GPUs, resulting in up to 3x improvements over baseline models.

B. Query Modes

To evaluate how well each VLM captures spatial relationships, we designed four distinct query modes to extract triplets. These modes represent progressively granular methods for prompting the models. Examples of all query modes are provided in our repository.

Mode 1 involves prompting the VLM with a general request for all spatial triplets in an image. We provide context regarding the entities and relationships of interest and specify that relationships should be expressed relative to the ego vehicle. The VLM then outputs a variable-length list of triplets corresponding to the entities and their relationships identified within the image. This open-ended querying mode allows us to capture the model's overall capacity to enumerate triplets in a driving scene, by querying the VLM only once.

Mode 2 employs a more targeted approach, querying the VLM with yes/no questions for each possible combination of entity type and spatial relationship. For instance, it asks the model if there is a vehicle in front of the ego vehicle, and how many. Each affirmative response is used to generate a triplet, if the model indicates that there are multiple instances, the corresponding triplet is repeated accordingly, and added to the list of predicted triplets. This mode constrains the VLMs to perform a more detailed analysis of individual entity relationship pairs.

Mode 3 leverages bounding boxes to guide the VLM's attention to specific entities within the image. Here, one entity in the image is highlighted with a bounding box at a time, and the VLM is prompted to identify the triplets describing that entity's spatial relationships to the ego vehicle. This query mode assumes access to bounding box information, which may be available from object detection systems, such as Detectron2 [42], Ground DINO [43], YOLO v11 [44], or YOLO World [45]. In our experiments, we benchmark the VLMs capabilities in extracting the correct triplets given a ground-truth bounding box. Although this can potentially over-estimate the VLM capabilities, in practice, existing object detection systems are becoming highly accurate [46].

Mode 4 combines the specificity of yes/no questions with the targeted guidance of bounding boxes. In each query, every bounding box and entity type, is paired with 6 yes/no questions regarding the object's spatial relationships to the ego vehicle (one for each of the 6 relationships). For instance, given a bounding box, it asks if the object in the bounding box is in front of the ego vehicle. If the response is affirmative, the triplet is added to the list of predicted triplets. While query-intensive, Mode 4 is expected to reduce the likelihood of VLM errors and ensures that only relevant relationships are examined.

C. Metrics

To quantify the performance of each VLM and query mode, we utilize mean precision@∞, mean recall@∞ [47], and mean F1@∞ as our primary metrics (henceforth, simply referred to as mean precision, mean recall, and mean F1). We use @∞ rather than a specific k-value because VLMs do not assign confidence scores to their predicted triplets, and our goal is to evaluate all triplets present in a scene even if they are repeated. Each model-query combination produces a list of predicted triplets, which we compare against a ground-truth list available in the DriST dataset. These lists are multisets to account for instances where the same triplet may appear multiple times within a scene, such as multiple pedestrians located to the right of the ego vehicle.

Let G_x denote the ground-truth multiset of triplets for a given scene x, and V_x denote the multiset of predicted triplets generated by the VLM V for the same scene x. We use $m_A(t)$ to refer to the count (or multiplicity) of triplet t in a multiset A. $m_A(t)=0$ if t does not appear in multiset A.

Definition IV.1 (Intersection of multi-sets)**.** The intersection $G_x \cap V_x$ of multisets G_x and V_x is defined using the count $m(t)$ of each triplet t as,

$$\forall t \in \mathcal{T}.\ m_{G_x \cap V_x}(t) = \min(m_{G_x}(t), m_{V_x}(t))$$

where $\mathcal{T}$ is the set of all possible triplets.

Definition IV.2 (Mean precision and recall of VLM)**.** Let D be a set of images. If G_x and V_x are the ground-truth and predicted multiset of triplets for an image x and VLM V, the mean precision (P) and recall (R) of the VLM V with respect to D are defined as,

$$\mathrm{P} = \frac{1}{|D|} \sum_{x \in D} \frac{|G_x \cap V_x|}{|V_x|} \qquad \mathrm{R} = \frac{1}{|D|} \sum_{x \in D} \frac{|G_x \cap V_x|}{|G_x|}$$

For a scene x, precision measures the proportion of correct triplets among those predicted by the VLM V, while recall measures the proportion of ground-truth triplets correctly identified by the VLM V. The F1 score provides a harmonic mean of precision and recall. For each model, we compute the mean of these precision, recall, and F1 scores with respect to the test set.

Model	QM	Time	Total	K	W	N
C-Llama3	1	15.29	0.19	0.15	0.18	0.23
C-Phi3	1	9.85	0.26	0.31	0.26	0.21
GPT-4-T	1	5.89	0.45	0.45	0.37	0.53
L1.5	1	3.95	0.36	0.44	0.35	0.28
L1.5-FT	1	2.57	**0.66**	0.72	**0.67**	**0.59**
L1.5-L	1	2.58	0.65	**0.73**	0.66	0.55
L1.6-Mis	1	3.15	0.36	0.35	0.38	0.35
L1.6-Vic	1	3.65	0.25	0.26	0.27	0.23
PaliGemma	1	1.02	0.33	0.38	0.32	0.30
RS2V	1	0.05	0.27	0.00	0.31	0.51
SpaceLLaVA	1	11.16	0.29	0.39	0.24	0.25
C-Llama3	2	8.71	0.54	0.55	0.56	0.51
C-Phi3	2	8.20	0.52	0.51	0.56	0.49
GPT-4-T	2	108.81	0.42	0.46	0.44	0.37
L1.5	2	5.13	0.45	0.44	0.46	0.44
L1.5-FT	2	4.81	**0.74**	**0.84**	**0.74**	**0.64**
L1.5-L	2	4.84	0.67	0.69	0.69	0.61
L1.6-Mis	2	8.93	0.50	0.47	0.52	0.49
L1.6-Vic	2	8.25	0.45	0.35	0.48	0.50
PaliGemma	2	1.69	0.27	0.31	0.32	0.20
SpaceLLaVA	2	14.44	0.42	0.44	0.47	0.34
C-Llama3	3	4.59	0.47	0.45	0.40	0.55
C-Phi3	3	4.01	0.45	0.42	0.37	0.56
GPT-4-T	3	27.63	0.52	0.49	0.38	0.71
L1.5	3	9.58	0.42	0.40	0.35	0.50
L1.5-FT	3	4.30	0.89	0.87	**0.90**	0.88
L1.5-L	3	4.35	**0.90**	**0.89**	0.90	**0.92**
L1.6-Mis	3	5.31	0.45	0.44	0.39	0.53
L1.6-Vic	3	5.60	0.38	0.29	0.34	0.50
PaliGemma	3	2.82	0.39	0.35	0.36	0.47
SpaceLLaVA	3	9.39	0.30	0.26	0.30	0.33
C-Llama3	4	10.20	0.50	0.63	0.27	0.59
C-Phi3	4	9.19	0.25	0.17	0.19	0.37
GPT-4-T	4	169.72	0.61	0.59	0.46	0.78
L1.5	4	6.40	0.56	0.50	0.57	0.61
L1.5-FT	4	5.42	**0.93**	**0.93**	**0.93**	**0.93**
L1.5-L	4	5.44	0.81	0.78	0.84	0.79
L1.6-Mis	4	26.91	0.52	0.45	0.51	0.61
L1.6-Vic	4	9.72	0.55	0.50	0.54	0.63
PaliGemma	4	2.30	0.56	0.50	0.57	0.61
SpaceLLaVA	4	28.72	0.56	0.50	0.57	0.61

TABLE I: F1 scores using 4 query modes (QM) for **K**itti, **W**aymo, **N**uScenes.

D. Heatmaps

To gain insights into the VLMs' specific strengths and weaknesses in identifying triplets, we further evaluate each model and query mode combination on a per-triplet basis. This analysis generates a "heatmap" that visualizes the mean precision and recall for each unique triplet across all scenes. We calculate the mean precision and recall per triplet as follows

Definition IV.3 (Mean precision and recall per triplet)**.** Let D be a set of images, G_x denote the set of ground truth triplets for an image x, and V_x denote the set of triplets returned by VLM V for same image x. Let $D_t^V = \{x \in D | m_{V_x}(t) > 0\}$ and $D_t^G = \{x \in D | m_{G_x}(t) > 0\}$. The mean precision P_t and mean recall R_t of t with respect to VLM V and dataset D is defined as,

$$\mathrm{P}_t = \frac{1}{|D_t^V|} \sum_{x \in D_t^V} \frac{m_{G_x \cap V_x}(t)}{m_{V_x}(t)} \qquad \mathrm{R}_t = \frac{1}{|D_t^G|} \sum_{x \in D_t^G} \frac{m_{G_x \cap V_x}(t)}{m_{G_x}(t)}$$

We compute these values offline based on a given set of images. Note that P_t and R_t are zero if $|D_t^V|$ and $|D_t^G|$ are zero, respectively.

E. Results

Summary. The results from our evaluation (Table I) highlight several key insights into the performance of VLMs when using various querying approaches to capture spatial relationships triplets in autonomous driving scenes. An extended table with additional information is provided in the repository. Notably, the fine-tuned LLaVA 1.5 models achieved the highest performance across query modes, outperforming all other models, including GPT-4 Turbo. This outcome underscores the effectiveness of DriST in enhancing VLMs capabilities to capture spatial relationships for driving scenarios. Furthermore, the LoRA-trained models—optimized for each query mode—also performed competitively, especially in Mode 3, where LoRA-trained LLaVA 1.5 marginally outperformed the fully fine-tuned variant.

The choice of query mode significantly impacted model performance, as observed in the total column of the results table. The incremental improvement from Mode 1 to Mode 2, and similarly from Mode 3 to Mode 4, suggests that structuring the prompts differently can help the model better capture the desired spatial triplets. Specifically, the more targeted querying approaches used in Modes 2 and 4 lead to better identification of spatial relationships, though they come at the expense of longer processing times due to the increased number of queries. The role of query mode in enhancing VLM performance points to the potential of exploring additional querying strategies to further optimize spatial relationship extraction.

A closer examination of the results across datasets reveals some variability, particularly in Modes 1 and 2. The best models generally performed worse on nuScenes, with lower F1 scores compared to KITTI and Waymo. This drop in performance likely arises from the poorer quality of some nuScenes images, which can be blurry or low-light, making it challenging for VLMs to accurately identify entities and their spatial relationships. Interestingly, in Modes 3 and 4, where bounding boxes were provided, this variability across datasets largely disappeared, with the best-performing models achieving consistent F1 scores across all three datasets. This suggests that in challenging visual conditions, the presence of bounding boxes assists the VLMs by focusing their attention on specific entities, thereby mitigating the impact of image quality.

Additionally, the poor performance of RoadScene2Vec, particularly on KITTI where it scored an F1 of 0, further highlights the sensitivity of some SGGs to dataset-specific camera configurations. RoadScene2Vec's reliance on camera information in its configuration file means that its performance is notably affected by differences in field of view, as KITTI images are wider than those in nuScenes and Waymo. Consequently, RoadScene2Vec performed relatively better on the other two datasets.

In terms of computational efficiency, the times recorded for each model and query mode reveal important trends. For Mode 1, RoadScene2Vec achieved the fastest times given its architecture, but its F1 score was less than a third of the best-performing model, suggesting a trade-off between speed and accuracy.

The fine-tuned VLM and LoRA models struck a balance between speed and F1 score, achieving both the best times and highest accuracy in this mode. This efficiency is likely due to the VLMs generating concise text with minimal repetition or unnecessary

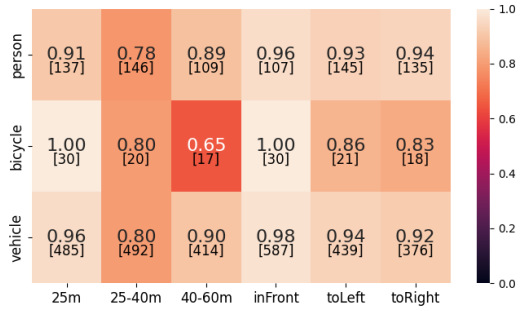


Fig. 3: Precision heatmap for LLaVA 1.5 finetuned for mode 4.

explanations, which kept generation time low. In Mode 2, despite the 18 individual questions asked, the fine-tuned VLM and LoRA models once again proved the fastest, as the responses only required simple yes/no answers. In contrast, other models were more verbose in their responses, which increased generation times. In Mode 3, Cambrian Phi3 demonstrated the fastest processing times among all models, though its F1 score was only half that of the top-performing fine-tuned VLM. Following Cambrian Phi3, the fine-tuned VLM and LoRA models were also relatively fast in this mode. Cambrian Phi3's shorter inference times likely stem from its smaller parameter size, which, combined with concise descriptions of the triplets, resulted in efficient processing. Finally, in Mode 4, the fine-tuned VLM and LoRA models once again emerged as the fastest, providing similar advantages in processing as in Mode 2, where concise yes/no responses minimized generation time.

Heatmaps. Fig 3 shows the heatmap for LLaVA 1.5 fine-tuned for mode 4, as it was the best performing across VLMs and query modes. The heatmap indicates that this VLM is not very precise at detecting bicycles between 40 and 60 meters, but the model is precise, and almost never misses a vehicle to the left of ego. The value in square brackets represents the number of samples that the triplet appeared in the prediction. In other words, it is the number of samples over which the average was calculated, providing an insight on how big is the sample size for each precision score. More heatmaps for different models and query modes can be found in our repository.

V. Monitoring Safety Properties with a VLM

Our goal is to use a VLM to evaluate temporal logic safety properties expressed in terms of propositions (triplets) about spatial relationships in driving scenes.

A. Building monitors

Specifying properties. We focus on properties expressed in linear temporal logic on finite traces (LTL_f) [10], a logic commonly used for monitoring. Let us recall our example specification in Fig. 1: *"'If there are any vehicles ahead within 25m of ego and ego speed is greater or equal than 25 mph for two consecutive time steps, then ego acceleration should be negative (braking) in the second time step"*. It can be expressed as the LTL_f formula $G((< vehicle, within25m, ego > \wedge egoSpeed \geq 25mph \wedge X(< vehicle,within25m,ego > \wedge egoSpeed \geq 25mph)) \rightarrow X\ isBraking)$. Here G (globally) and X (next) are standard operators in LTL_f.

Evaluating propositions on images. Propositions such as $< vehicle,within25m,ego >$ are evaluated on sensor inputs (i.e., images). To do that we leverage a VLM to predict the triplets in a scene. Then, we use the list of predicted triplets to determine if the proposition is true or false. Other propositions, such as speed and acceleration can typically be obtained accurately from the ADS, without the need for visual sensor data.

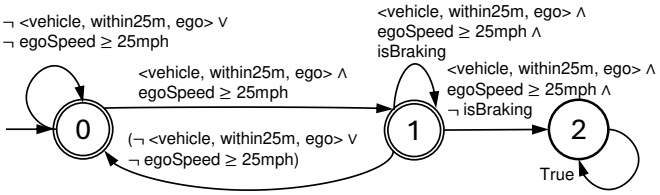


Fig. 4: DFA Monitor for the LTL_f Property.

Monitor synthesis. The property specified above, can be automatically converted to a deterministic finite automaton (DFA) that serves as a monitor for checking whether a finite trace violates the property [48], [8]. Figure 4 shows the DFA that was synthesized from the formula. States 0 and 1 are *accepting*, meaning that the property holds, and state 2 is a *bad* state, meaning that the property is violated. The transitions of the DFA require evaluating the propositions using the VLM or accessing ADS information.

B. Case study

This case study aims to show the effectiveness of monitors using the best performing VLM from Sec. IV-E, L1.5-FT, to check three safe driving properties illustrated in Table II.

Evaluation Dataset. While the VLMs evaluations in Sec. IV used 300 images from our Waymo test set (not seen during the VLM training), here we analyze our entire Waymo test set, composed of 199 sequences containing over 39k images. These sequences were collected in 3 cities: San Francisco, Mountain View, and Phoenix, at different times of the day, containing 196 frames on average. These frames contain 67,566 vehicles, 1,698 bicycles, and 22,027 persons. We leverage the ground truth triplets from these 199 sequences to verify the properties and compare the violations detected in the ground truth to those identified using the VLM's predictions. If we take a closer look at the properties in Table II, the first proposition is *anythingWithin25m*, *anythingBetween25_40m*, and *anythingBetween40_60m* for φ_1, φ_2, and φ_3, respectively. These propositions are a shortname for the disjunction between the different entities at a given range. For example: $anythingWithin25m = < vehicle,within25m,ego > \vee < bicycle,within25m,ego > \vee < person,within25m,ego >$.

Metrics. We present the results in terms of the following metrics: *True Violations (True Positives)*, property violations successfully identified by the monitor; *False Violations (False Positives)*, property violations predicted by the VLM that were not actual violations; *Missed Violations (False Negatives)*, property violations present in the ground truth but missed by the VLM; finally, *True Non-Violations (True Negatives)*, denote cases where no violation occurred in both the ground truth and the VLM's predictions.

Results. Table III summarizes the results. The monitors successfully identified 74% true violations (TP) for φ_1, 78% for φ_2, and 100% for φ_3; while reporting 1% of false violations (FP) for φ_1, 1% for φ_2; and none for φ_3. The low number of false violations (FP) suggests that the monitors are precise in their predictions,

Property	Definition
φ_1	$G(anythingWithin25m \wedge egoSpeed \geq 25mph \wedge X(anythingWithin25m \wedge egoSpeed \geq 25mph) \rightarrow XisBraking)$
φ_2	$G(anythingBetween25\_40m \wedge egoSpeed \geq 35mph \wedge X(anythingBetween25\_40m \wedge egoSpeed \geq 35mph) \rightarrow XisBraking)$
φ_3	$G(anythingBetween40\_60m \wedge egoSpeed \geq 45mph \wedge X(anythingBetween40\_60m \wedge egoSpeed \geq 45mph) \rightarrow XisBraking)$

TABLE II: Safe driving properties analyzed in the case study.

		Ground Truth	
		Violation	Non-Violation
VLM	Violation	True Positives (TP) $\varphi_1 = 14$ $\varphi_2 = 7$ $\varphi_3 = 6$	False Positives (FP) $\varphi_1 = 2$ $\varphi_2 = 2$ $\varphi_3 = 0$
	Not Violation	False Negatives (FN) $\varphi_1 = 5$ $\varphi_2 = 2$ $\varphi_3 = 0$	True Negatives (TN) $\varphi_1 = 179$ $\varphi_2 = 188$ $\varphi_3 = 193$

TABLE III: Case study results.

and the low number of missed violations (FN) indicates that the monitors do not miss many entities relevant to the properties.

Furthermore, the high count of true non violations (TN) suggests that the monitors correctly identified sequences without violations. We note that 15/179 (8%) in φ_1, 4/188 (2%) in φ_2, and 4/193 (2%) in φ_3 are the cases where the speed of the ego vehicle is higher than the threshold. For those cases, it means that the VLM accurately predicted the relevant triplets, without leading to false violations.

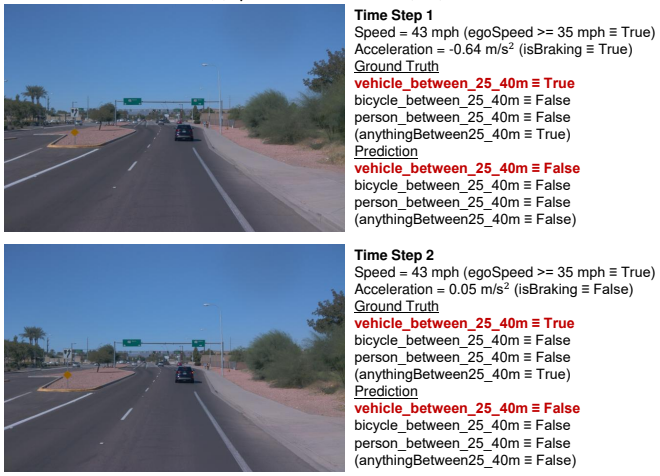
Figure 5a shows an example of a true violation (TP) of φ_2, where there is a vehicle between 25m and 40m (black SUV), the ego speed is 44 mph and the ego vehicle does not decelerate. In contrast, Figure 5b shows an example of a missed violation (FN), where there is a vehicle within 25m and 40m, yet the VLM did not detect that there is one. In this case, the ground truth suggests that the car is between 25m and 40m, while the VLM predicts that the car is between 40m and 60m, which is not correct. This is a subtle miss-classification from the VLM given that the car is close to the range limit, around 40 meters.

VI. Conclusion

In this paper we investigated the use of VLMs for monitoring safety properties in ADSs. We described the DriST dataset, which was used for evaluating state-of-the-art VLMs for capturing spatial triplets. Although out-of-the-box VLMs perform poorly on the task, our DriST dataset can be used for model fine-tuning, significantly improving performance. We also showcase the use of monitors to check safety properties on real images, by leveraging the best-performing VLM with promising results.

Future work includes two lines of work. First, we will explore fine-tuning smaller and more cost-effective VLMs, such as MobileVLM [49], to reduce inference costs, and thus facilitate deployment as part of monitors in an on-line setting. Second, we will extend our triplets to encode more general spatial relationships and express richer set of properties, and improve the logic to accommodate the imprecisions of sensing and VLM interpretation.

(a) φ_2 True Violation (TP).



(b) φ_2 Missed Violation (FN).

Fig. 5: φ_2 True and Missed Violation examples.

References

[1] A. Marshall, "Uber video shows the kind of crash self-driving cars are made to avoid," Mar 2018, accessed on 02.07.2024. [Online]. Available: https://www.wired.com/story/uber-self-driving-crash-video-arizona/

[2] N. Board, "Collision between vehicle controlled by developmental automated driving system and pedestrian. nat. transpot. saf. board, washington, dc," USA, Tech. Rep. HAR-19-03, 2019. URL https://www.ntsb. gov/investigations . . . , Tech. Rep., 2019.

[3] B. Templeton, "Tesla in taiwan crashes directly into overturned truck, ignores pedestrian, with autopilot on," *Forbes*, Jun 2020, https://www.forbes.com/sites/bradtempleton/2020/06/02/tesla-in-taiwan...

[4] N. E. Boudette and N. Chokshi, "U.s. will investigate tesla's autopilot system over crashes with emergency vehicles," *New York Times*, Aug 2021, accessed on 02.07.2024. [Online]. Available: https://www.nytimes.com/2021/08/16/business/tesla-autopilot-nhtsa.html

[5] R. Bellan, "A waymo self-driving car killed a dog in 'unavoidable' accident," Jun 2023, accessed on 02.07.2024. [Online]. Available: https://techcrunch.com/2023/06/06/a-waymo-self-driving-car-killed-a-dog-in-unavoidable-accident/

[6] H. Araujo, M. R. Mousavi, and M. Varshosaz, "Testing, validation, and verification of robotic and autonomous systems: A systematic review," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, mar 2023. [Online]. Available: https://doi.org/10.1145/3542945

[7] N. Mehdipour, M. Althoff, R. D. Tebbens, and C. Belta, "Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges," *Automatica*, vol. 152, p. 110692, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109822005568

[8] F. Toledo, T. Woodlief, S. Elbaum, and M. B. Dwyer, "Specifying and monitoring safe driving properties with scene graphs," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 577–15 584.

[9] K. Viswanadha, F. Indaheng, J. Wong, E. Kim, E. Kalvan, Y. Pant, D. J. Fremont, and S. A. Seshia, "Addressing the IEEE AV test challenge with scenic and verifai," in *2021 IEEE International Conference on Artificial Intelligence Testing, AITest 2021, Oxford, United Kingdom, August 23-26, 2021*. IEEE, 2021, pp. 136–142. [Online]. Available: https://doi.org/10.1109/AITEST52744.2021.00034

[10] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence*, 2013, pp. 854–860.

[11] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2304.00685

[12] 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 *CVPR*, 2020.

[13] P. S. et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] A. R. et al., "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 8748–8763. [Online]. Available: http://proceedings.mlr.press/v139/radford21a.html

[16] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html

[17] L. B. et al., "Paligemma: A versatile 3b vlm for transfer," 2024. [Online]. Available: https://arxiv.org/abs/2407.07726

[18] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, A. Wang, R. Fergus, Y. LeCun, and S. Xie, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," 2024. [Online]. Available: https://arxiv.org/abs/2406.16860

[19] OpenAI, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[20] G. Team, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[21] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "FLAVA: A foundational language and vision alignment model," *CoRR*, vol. abs/2112.04482, 2021. [Online]. Available: https://arxiv.org/abs/2112.04482

[22] J. A. et al., "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html

[23] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *CoRR*, vol. abs/2102.12092, 2021. [Online]. Available: https://arxiv.org/abs/2102.12092

[24] K. Ranasinghe, S. N. Shukla, O. Poursaeed, M. S. Ryoo, and T.-Y. Lin, "Learning to localize objects improves spatial reasoning in visual-llms," 2024. [Online]. Available: https://arxiv.org/abs/2404.07449

[25] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," 2024. [Online]. Available: https://arxiv.org/abs/2401.12168

[26] J. H. Cho, B. Ivanovic, Y. Cao, E. Schmerling, Y. Wang, X. Weng, B. Li, Y. You, P. Krähenbühl, Y. Wang, and M. Pavone, "Language-image models with 3d understanding," *CoRR*, vol. abs/2405.03685, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.03685

[27] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. G. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[28] T. He, L. Gao, J. Song, and Y. Li, "Towards open-vocabulary scene graph generation with prompt-based finetuning," ser. Lecture Notes in Computer Science, S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., vol. 13688. Springer, 2022, pp. 56–73. [Online]. Available: https://doi.org/10.1007/978-3-031-19815-1_4

[29] Y. Zhang, Y. Pan, T. Yao, R. Huang, T. Mei, and C. W. Chen, "Learning to generate language-supervised and open-vocabulary scene graph using pre-trained visual-semantic space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 2915–2924. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.00285

[30] R. Li, S. Zhang, D. Lin, K. Chen, and X. He, "From pixels to graphs: Open-vocabulary scene graph generation with vision-language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, 2024, pp. 28 076–28 086. [Online]. Available: https://doi.org/10.1109/CVPR52733.2024.02652

[31] Q. M. Rahman, P. Corke, and F. Dayoub, "Run-time monitoring of machine learning for robotic perception: A survey of emerging trends," *IEEE Access*, vol. 9, pp. 20 067–20 075, 2021.

[32] A. Balakrishnan, J. Deshmukh, B. Hoxha, T. Yamaguchi, and G. Fainekos, "Percemon: Online monitoring for perception systems," in *Runtime Verification*, L. Feng and D. Fisman, Eds. Cham: Springer International Publishing, 2021, pp. 297–308.

[33] J. Anderson, G. Fainekos, B. Hoxha, H. Okamoto, and D. V. Prokhorov, "Pattern matching for perception streams," in *Runtime Verification - 23rd International Conference, RV 2023, Thessaloniki, Greece, October 3-6, 2023, Proceedings*, ser. Lecture Notes in Computer Science, P. Katsaros and L. Nenzi, Eds., vol. 14245. Springer, 2023, pp. 251–270. [Online]. Available: https://doi.org/10.1007/978-3-031-44267-4_13

[34] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. D. Nesnas, and M. Pavone, "Semantic anomaly detection with large language models," *Auton. Robots*, vol. 47, no. 8, pp. 1035–1055, 2023. [Online]. Available: https://doi.org/10.1007/s10514-023-10132-6

[35] V. D. of Motor Vehicles, "Virginia driver's manual." [Online]. Available: https://transactions-t.dmv.virginia.gov/webdoc/pdf/dmv39.pdf

[36] A. V. Malawade, S. Yu, B. Hsu, H. Kaeley, A. Karra, and M. A. A. Faruque, "roadscene2vec: A tool for extracting and embedding road scene-graphs," *Knowl. Based Syst.*, vol. 242, p. 108245, 2022. [Online]. Available: https://doi.org/10.1016/j.knosys.2022.108245

[37] H. T. et al., "Llama: Open and efficient foundation language models," *CoRR*, vol. abs/2302.13971, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.13971

[38] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2023.

[39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," 2023. [Online]. Available: https://arxiv.org/abs/2303.15343

[40] A. D. et. al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[41] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[42] V. Pham, C. Pham, and T. Dang, "Road damage detection and classification with detectron2 and faster R-CNN," *CoRR*, vol. abs/2010.15021, 2020. [Online]. Available: https://arxiv.org/abs/2010.15021

[43] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[44] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[45] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 901–16 911.

[46] P. with Code, "Object detection on coco leaderboard," 2024, accessed: 2024-11-14. [Online]. Available: https://paperswithcode.com/sota/object-detection-on-coco

[47] J. Lorenz, R. Schön, K. Ludwig, and R. Lienhart, "A review and efficient implementation of scene graph generation metrics," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*. IEEE, 2024, pp. 2567–2575. [Online]. Available: https://doi.org/10.1109/CVPRW63382.2024.00263

[48] S. Zhu, G. Pu, and M. Y. Vardi, "First-order vs. second-order encodings for ltlf-to-automata translation," in *Theory and Applications of Models of Computation*, T. Gopal and J. Watada, Eds. Cham: Springer International Publishing, 2019, pp. 684–705.

[49] X. Chu, L. Qiao, X. Lin, S. Xu, Y. Yang, Y. Hu, F. Wei, X. Zhang, B. Zhang, X. Wei *et al.*, "Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices," *arXiv preprint arXiv:2312.16886*, 2023.