Università di Genova

DEPARTMENT OF NAVAL, ELECTRICAL, ELECTRONIC AND TELECOMMUNICATIONS ENGINEERING

Master's Thesis in Electronic Engineering

VLM-Based System Architecture for Real-Time Recognition and Characterization of Driving Scenarios

Architettura di Sistema Basata su VLM per il Riconoscimento e la Caratterizzazione in Tempo Reale di Scenari di Guida

Simone Incerti

July 17, 2026

Supervisors: Prof. Francesco Bellotti, Prof. Riccardo Berta

Co-supervisors: Dr. Matteo Fresta, Dr. Alessandro Pighetti

Ringraziamenti

Desidero esprimere la mia più sincera gratitudine ai miei relatori, il Prof. Francesco Bellotti e il Prof. Riccardo Berta, per avermi guidato e supportato durante tutto il percorso che ha portato alla stesura di questa tesi. I loro preziosi consigli, la loro competenza e la disponibilità dimostrata sono stati fondamentali per il raggiungimento di questo traguardo.

Un ringraziamento speciale va ai miei correlatori, il Dr. Matteo Fresta e il Dr. Alessandro Pighetti, per l'aiuto costante, la pazienza e l'entusiasmo che mi hanno trasmesso lavorando a questo progetto. Insieme a loro, vorrei ringraziare di cuore tutti i ragazzi dell'Elios Lab: con voi ho condiviso non solo un ambiente di lavoro stimolante e proficuo, ma anche momenti di confronto e leggerezza che porterò sempre tra i miei ricordi più belli.

Il grazie più grande e affettuoso va alla mia famiglia. Ai miei genitori, per aver creduto sempre in me, per i sacrifici fatti per permettermi di studiare e per il loro amore e sostegno incondizionato in ogni singola fase della mia vita. Se oggi sono qui, è soprattutto grazie a voi.

Infine, un pensiero speciale ai miei amici. A chi mi è stato vicino in questi anni di università, a chi ha condiviso con me le ansie pre-esame, le gioie dei traguardi raggiunti e le serate indimenticabili. Senza il vostro supporto, questo percorso non sarebbe stato lo stesso.

Grazie di cuore a tutti.

Alla mia famiglia e ai miei amici.

Sommario

La guida autonoma richiede sia una comprensione semantica della scena di guida sia la stima precisa di grandezze fisiche di sicurezza, quali la velocità del veicolo (ego-vehicle speed), il tempo di interdistanza (time headway) e il tempo di collisione (time to collision). Queste due capacità sono state tradizionalmente affrontate da modelli separati e specifici per ogni singolo compito: pipeline di percezione per la stima metrica e modelli linguistici per la descrizione della scena. Questa tesi analizza se un singolo Vision-Language Model (VLM) possa svolgere entrambi i ruoli simultaneamente attraverso l'apprendimento multi-task, senza sacrificare la qualità in nessuno dei due compiti.

Il sistema proposto è costruito attorno a Qwen3-VL-2B-Instruct, un VLM da 2 miliardi di parametri utilizzato come backbone condiviso per due compiti eterogenei. Il primo compito è la regressione fisica: partendo da una sequenza di tre frame acquisiti da una dashcam, il modello stima la velocità del veicolo, il Time Headway (THW) e il Time To Collision (TTC) tramite una semplice testa Multi-Layer Perceptron (MLP) collegata all'ultimo stato nascosto (hidden state) del backbone. Il secondo compito è la descrizione della scena: lo stesso backbone genera una frase in linguaggio naturale che descrive l'azione del veicolo e la sua motivazione, seguendo lo stile di annotazione del benchmark BDD-X.

Vengono confrontati due paradigmi di addestramento: un approccio di baseline in cui il backbone è congelato (frozen) e viene addestrata solo la testa

di regressione, e un approccio completo end-to-end in cui il backbone viene fine-tuned utilizzando la Quantized Low-Rank Adaptation (QLoRA). Una masked regression loss personalizzata gestisce la disponibilità condizionale di THW e TTC, che sono definiti fisicamente solo quando è presente un veicolo che precede. L'addestramento segue una strategia di proportional random interleaving che ottimizza congiuntamente entrambi i compiti in ogni epoca, prevenendo la dominanza di un task sull'altro.

Vengono utilizzati due dataset. Il primo è un dataset telemetrico Advanced Driver Assistance Systems (ADAS) personalizzato, costruito a partire da riprese video grezze di dashcam sincronizzate con lo stato del veicolo e i log radar provenienti da una piattaforma compatibile con OpenPilot. Il secondo è BDD-X, un benchmark pubblicamente disponibile di video di guida annotati con relative spiegazioni in linguaggio naturale.

Viene sviluppata e validata su sequenze video mai viste (unseen) una pipeline di inferenza in tempo reale, caratterizzata da un filtro semantico anti-flicker basato su sentence embedding, che stabilizza le descrizioni della scena generate durante passi di inferenza consecutivi.

I risultati sperimentali confermano che un VLM compatto può svolgere efficacemente entrambi i ruoli percettivi. L'approccio end-to-end con QLoRA supera in modo sostanziale la baseline con backbone frozen, ottenendo un Mean Absolute Error (MAE) sulla velocità del veicolo di 5.32 km/h (un miglioramento del 47.6 %) e dimostrando eccellenti capacità di ragionamento semantico nel compito di descrizione della scena con un BERTScore F1 di 0.9124.

Abstract

Autonomous driving requires both a semantic understanding of the driving scene and the precise estimation of physical safety quantities such as ego-vehicle speed, time headway, and time to collision. These two capabilities have traditionally been addressed by separate, task-specific models: perception pipelines for metric estimation and language models for scene description. This thesis investigates whether a single Vision-Language Model (VLM) can serve both roles simultaneously through multi-task learning, without sacrificing quality on either task.

The proposed system is built around Qwen3-VL-2B-Instruct, a 2-billion-parameter VLM used as a shared backbone for two heterogeneous tasks. The first task is physical regression: from a sequence of three dashcam frames, the model estimates ego speed, Time Headway (THW), and Time To Collision (TTC) via a simple Multi-Layer Perceptron (MLP) head attached to the backbone's last hidden state. The second task is scene description: the same backbone generates a natural language sentence describing the vehicle's action and its reason, following the annotation style of the BDD-X benchmark.

Two training paradigms are compared: a baseline approach in which the backbone is frozen and only the regression head is trained, and a full end-to-end approach in which the backbone is fine-tuned using Quantized Low-Rank Adaptation (QLoRA). A custom masked regression loss handles the conditional availability of THW and TTC, which are only physically defined when a lead vehicle is present. Training follows a proportional random interleav-

ing strategy that jointly optimizes both tasks in each epoch, preventing task dominance.

Two datasets are used. The first is a custom Advanced Driver Assistance Systems (ADAS) telemetry dataset built from raw dashcam footage synchronized with vehicle state and radar logs from an OpenPilot-compatible platform. The second is BDD-X, a publicly available benchmark of annotated driving videos with paired natural language explanations.

A real-time inference pipeline is developed and validated on unseen video sequences, featuring a semantic anti-flicker filter based on sentence embeddings that stabilizes the generated scene descriptions over consecutive inference steps.

The experimental results confirm that a compact VLM can effectively serve both perceptual roles. The end-to-end QLoRA approach substantially outperforms the frozen backbone baseline, achieving an ego speed Mean Absolute Error (MAE) of 5.32 km/h (a 47.6 % improvement) and demonstrating strong semantic reasoning on the scene description task with a BERTScore F1 of 0.9124.

Contents

List of Acronyms

NF4	NormalFloat4	21
PEFT	Parameter-Efficient Fine-Tuning	18
PET	Post-Encroachment Time	14
QLoRA	Quantized Low-Rank Adaptation	v
SFT	Supervised Fine-Tuning	29
SoC	System-on-Chip	52
THW	Time Headway	v
TTC	Time To Collision	v
ViT	Vision Transformer	7
VLM	Vision-Language Model	v
VRAM	Video Random Access Memory	30

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of autonomous and semi-autonomous vehicles represents one of the most demanding challenges in modern engineering [1]. Unlike traditional software systems, a self-driving vehicle must operate reliably across an open-ended range of environments, weather conditions, traffic densities, and edge cases, making decisions in real time with consequences that are immediately physical. Meeting this challenge requires simultaneously solving problems at very different levels of abstraction: from the low-level estimation of inter-vehicle distances and collision risk margins, to the high-level comprehension of scene semantics such as traffic rules, the intentions of other road users, and the overall narrative of what is happening on the road.

Until recently, these two levels of understanding were treated as entirely separate engineering problems. Safety-critical quantities such as THW and TTC were estimated by dedicated ADAS modules relying on radar or lidar returns [2], sensor modalities that provide accurate range measurements but carry no semantic information about the scene. Scene understanding, on the other hand, was the domain of visual recognition pipelines built on Convolutional Neural Networks (CNNs) [3; 4]: systems that could detect and classify objects but could not reason about them in natural language or provide

human-readable explanations of their behaviour.

The landscape shifted substantially with the rise of large-scale VLMs [5; 6]. These models, trained on hundreds of millions of image–text pairs, learn visual representations that are not only rich in semantic information but also deeply intertwined with linguistic concepts. A VLM does not merely classify what is in an image; it can describe it, answer questions about it, and reason about it in the same way a human observer would. This raises a natural question: if VLMs encode such comprehensive visual knowledge, do their internal representations also carry information about physical, quantitative properties of the scene, the kind of information that ADAS systems currently extract from radar? And if so, could a single VLM backbone serve both roles simultaneously, acting as a unified perceptual substrate for both semantic understanding and physical metric estimation?

Answering this question is the central motivation of this thesis.

1.1 Context of the Study

This thesis was developed in the context of the growing intersection between Large Language Models (LLMs) and the autonomous driving domain. Several research groups have demonstrated that large VLMs can be prompted or fine-tuned to perform driving-related reasoning tasks [7]: generating explanations of vehicle behaviour [8], answering questions about traffic scenarios [9], and even predicting high-level control commands from dashcam images [10; 11]. However, these works have largely treated VLMs as semantic engines, focusing on the language output and leaving physical state estimation (TTC, THW, ego speed) to separate, non-language modules.

This thesis takes a different stance: it treats the VLM backbone as a *shared*

feature extractor for both a physical regression task and a language generation task, and trains both branches jointly through Multi-Task Learning (MTL) [12; 13]. The backbone used is Qwen3-VL-2B-Instruct, a compact 2-billion-parameter open-source VLM that natively supports multi-image inputs within a single conversation turn, a property that is essential for processing the short video sequences (three frames spanning one second) used in this work.

Two datasets are employed to train the two branches of the system. The first is a custom ADAS telemetry dataset assembled from raw driving logs recorded on an OpenPilot-compatible platform, in which front-facing dashcam video is synchronized with vehicle state logs and radar measurements. The second is BDD-X [14], a publicly available benchmark that pairs driving videos with human-written natural language explanations of vehicle actions and their justifications.

A further practical constraint shapes the design of the system: the entire training pipeline must run on a single consumer-grade Graphics Processing Unit (GPU). This rules out full fine-tuning of the 2B-parameter backbone and motivates the use of QLoRA [15], which reduces the memory footprint of fine-tuning by a factor of approximately four through 4-bit quantization of the frozen backbone weights.

1.2 Objectives and Contributions

The primary objective of this thesis is to design, train, and evaluate a unified VLM-based system capable of simultaneously estimating physical ADAS safety metrics and generating natural language descriptions of driving scenes from short multi-frame video sequences.

The specific contributions of this work are:

1. The design of a unified multi-task architecture and a custom telemetry dataset, combining a dedicated OpenPilot data extraction pipeline with a joint Qwen3-VL-2B backbone. This single backbone is augmented with a lightweight `ADAS_MLP` regression head to predict ego speed and radar-derived safety metrics (THW and TTC), while simultaneously serving a language generation branch to produce natural language scene descriptions following the BDD-X annotation style.

2. The implementation of an advanced training methodology and paradigm evaluation, featuring a custom masked regression loss designed to handle the conditional accessibility of THW and TTC based on lead vehicle presence. This is coupled with a proportional random interleaving strategy to balance multi-task batches and prevent gradient dominance, alongside a comprehensive evaluation comparing a computationally lightweight frozen-backbone baseline against a full end-to-end QLoRA fine-tuning scheme.

3. The development of a real-time inference pipeline with semantic anti-flicker filtering that applies the trained model to unseen dashcam footage, overlaying both predicted ADAS metrics and a live scene description onto each frame. A cosine similarity filter based on sentence embeddings is integrated to prevent the displayed description from flickering when consecutive model outputs are semantically equivalent.

1.3 Overview of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 - Related Work reviews the technical background relevant to this work. It covers the evolution of deep learning for autonomous driving

perception, the architecture and training of modern VLMs (with particular attention to Qwen3-VL), the definition and estimation of ADAS safety metrics (TTC and THW), the principles and challenges of MTL, parameter-efficient fine-tuning methods (Low-Rank Adaptation (LoRA) and QLoRA), and the prior literature on explainable autonomous driving and the BDD-X benchmark.

Chapter 3 - System's Architecture describes the proposed multi-task system in detail. It presents the shared backbone, the `ADAS_MLP` regression head and its masked regression loss, the autoregressive generation branch, and the two experimental training paradigms.

Chapter 4 - Methodology details the dataset construction pipelines for both the ADAS telemetry dataset and the BDD-X sequences, the multi-task training configuration including the proportional random interleaving strategy, optimizer and learning rate schedule, and the real-time inference pipeline with the semantic anti-flicker filter.

Chapter 5 - Experimental Results presents the quantitative evaluation of both approaches on the held-out test sets, including MAE metrics for the ADAS regression task and standard natural language generation metrics (BLEU-4, METEOR, CIDEr, BERTScore) for the scene description task. Training curves and qualitative examples are also discussed.

Chapter 6 - Conclusions summarizes the main findings of this thesis, confirming that a compact VLM can effectively serve both perceptual roles. It highlights the superior performance of the end-to-end approach, acknowledges limitations regarding dataset scale and monocular depth ambiguity, and outlines future research directions such as learned temporal aggregation and unified single-pass architectures.

Chapter 2

Related Work

This chapter reviews the technical foundations underlying the proposed system. Section 2.1 introduces the role of deep learning in autonomous driving perception. Section 2.2 covers the architecture and training paradigms of modern VLMs. Section 2.3 defines the ADAS safety metrics estimated in this thesis. Section 2.4 reviews MTL theory and relevant prior work. Section 2.5 describes parameter-efficient fine-tuning methods, with a focus on LoRA and QLoRA. Section 2.6 summarizes the literature on textual explainability for autonomous driving.

2.1 Deep Learning for Autonomous Driving Perception

Early autonomous driving systems relied on hand-crafted features and rule-based decision logic. Perception stacks of the 1990s and early 2000s combined classical computer vision techniques, such as edge detection, optical flow, and Kalman filtering, with symbolic reasoning modules that encoded traffic rules explicitly [16]. These systems were brittle: hand-crafted features generalised poorly to the immense visual variability of real-world roads, and every new

environmental condition (rain, fog, unusual lighting, occlusion) typically required additional engineering effort. The advent of deep CNNs [17] shifted the paradigm toward *learned* visual representations. Landmark results such as AlexNet [3] demonstrated that features learned end-to-end from raw pixels dramatically outperformed engineered alternatives on visual recognition benchmarks, catalysing a rapid transition in the automotive perception stack [18].

This transition unfolded across several perceptual sub-problems. Object detection frameworks evolved from two-stage region-proposal networks to single-shot detectors capable of running in real time on embedded hardware [19], enabling the reliable localisation of vehicles, pedestrians, and cyclists from monocular camera feeds. In parallel, semantic and instance segmentation networks provided pixel-level scene understanding [20; 21], supporting drivable-area estimation and free-space detection. Depth and motion estimation networks, whether monocular, stereo, or LiDAR-based, complemented these detectors by recovering the three-dimensional structure of the scene. Modern systems typically decompose the perception problem into a pipeline of specialized modules (object detection, semantic segmentation, depth estimation), each trained independently on task-specific datasets and loss functions. While this modular design facilitates engineering, debugging, and incremental deployment, it limits the potential for mutual improvement across tasks: errors made by an early module (e.g., a missed detection) propagate downstream without the possibility of correction, and there is no shared representation that could allow one task to benefit from the supervisory signal of another.

The introduction of the Transformer architecture [22] to vision through the Vision Transformer (ViT) [23] marked a turning point. By treating an image as a sequence of patches and applying global self-attention, ViT mod-

els capture long-range spatial dependencies that CNNs, with their inherently local receptive fields, struggle to model without very deep stacks of layers. A subsequent line of work sought to combine the inductive biases of convolutions with the global reasoning of attention: the Swin Transformer [24], for instance, introduced a hierarchical architecture with shifted windows, computing self-attention within local windows that are shifted between successive layers, thereby achieving linear computational complexity with respect to image size while retaining cross-window connectivity. This hierarchical design made Transformer-based backbones practical for dense prediction tasks such as detection [25] and segmentation, which are core building blocks of the automotive perception stack.

In the autonomous driving domain specifically, Transformer-based models are increasingly applied to multi-sensor fusion [4], trajectory prediction [26], and end-to-end driving [27; 28], demonstrating the advantage of processing full scene context within a unified representation. UniAD [4], in particular, is emblematic of a broader shift toward *planning-oriented* perception, in which detection, tracking, mapping, and motion forecasting are unified within a single query-based Transformer architecture that is optimized end-to-end for the final planning objective, rather than optimizing each perceptual sub-task in isolation. This trend toward unified, jointly-optimized architectures is directly relevant to the present thesis, which similarly seeks to unify two traditionally separate perceptual objectives, physical metric regression and semantic scene description, within a single shared backbone.

2.2 Vision-Language Models

2.2.1 From Contrastive Pre-Training to Generative Architectures

The foundational work on large-scale vision-language alignment is CLIP [5], which trained a dual-encoder model on 400 million image-text pairs using a contrastive objective: the model learns to project images and their corresponding captions into a shared embedding space such that matched pairs are pulled together while mismatched pairs are pushed apart. This simple yet remarkably scalable objective produced visual representations that transferred well to a wide range of downstream tasks in a zero-shot manner, without any task-specific fine-tuning. However, CLIP-style dual encoders produce fixed-size embeddings suitable for retrieval and classification, but are not designed to generate open-ended text.

Subsequent work moved toward *generative* architectures capable of producing open-ended text conditioned on visual inputs. BLIP [29] proposed a unified framework that jointly pre-trains on understanding-based objectives (image-text contrastive and matching losses) and generation-based objectives (image-conditioned language modelling), and introduced a bootstrapping mechanism to filter noisy web-scraped captions and synthesize higher-quality ones, improving data efficiency. BLIP-2 [30] extended this idea by introducing a lightweight Querying Transformer (Q-Former) that bridges a frozen image encoder and a frozen LLM, dramatically reducing the number of trainable parameters required to align vision and language representations. LLaVA [6] took a complementary, even simpler approach, demonstrating that a single linear projection layer mapping visual features directly into the LLM's token embedding space

is sufficient for strong visual instruction following, provided the model is fine-tuned on a high-quality corpus of instruction-response pairs generated with the assistance of a stronger teacher model. Underlying all of these generative VLMs is a pre-trained LLM decoder; the LLaMA family of models [31], an open and efficient collection of foundation language models trained on publicly available corpora, has been a particularly influential choice of backbone for the open-source VLM community, since it offers competitive performance to closed, proprietary models while remaining inspectable and fine-tunable.

2.2.2 Multi-Image Understanding and the Qwen3-VL Architecture

Early generative VLMs processed a single image per query, which is sufficient for tasks such as image captioning or visual question answering but is fundamentally limited when temporal reasoning over a sequence of observations is required, as is the case for driving scene understanding. Extending VLMs to accept multiple images within a single context is therefore essential for temporal understanding. Recent models like InternVL [32] and Qwen2-VL [33] introduced dynamic visual token budgeting, allowing an arbitrary number of images (or, equivalently, video frames) to be passed as separate context entries within one conversation turn. Rather than resizing all images to a fixed resolution, as was common practice in earlier VLMs, each image is tiled dynamically based on its native aspect ratio, preserving fine-grained spatial details without imposing a fixed, potentially wasteful token overhead on small or low-detail images. This mechanism also allows the model to allocate more visual tokens to higher-resolution or more information-dense frames, which is particularly valuable for driving scenes where small, distant objects (e.g., traffic signs, brake lights) can carry disproportionate semantic importance.

The backbone used in this thesis, Qwen3-VL-2B-Instruct [34], builds on these principles. Its architecture integrates a Vision Transformer encoder that processes each input image independently. The resulting embeddings are mapped into the LLM's token space via a projection module, allowing the causal decoder to autoregressively generate output tokens conditioned on interleaved sequences of visual and text tokens. Its instruction-tuned nature makes it highly adept at following natural language prompts about driving scenes. Beyond raw instruction-following, the model's multi-image mechanism is a key architectural enabler for this thesis: since the same visual token stream is available both to the autoregressive language-modelling head and to any auxiliary head attached to the model's hidden states, a single forward pass can, in principle, support heterogeneous downstream objectives without duplicating the visual encoder. This property is exploited directly in Chapter 3, where the last hidden state of the shared backbone is routed simultaneously to a regression head and to the standard language-modelling head.

It is also worth situating Qwen3-VL-2B-Instruct with respect to model scale. Contemporary VLM families are typically released in a range of sizes, from a few hundred million to tens of billions of parameters, reflecting a trade-off between representational capacity and deployment cost. The 2-billion-parameter configuration selected for this work sits close to the smallest end of this spectrum among models that retain competitive instruction-following and multi-image capabilities, a deliberate choice motivated by the single-consumer-GPU training and real-time inference constraints discussed in Section 2.5 and Chapter 4.

2.3 ADAS and Safety-Critical Metrics

2.3.1 Overview of ADAS and Sensor-Based Estimation

ADAS encompass technologies designed to enhance vehicle safety and situational awareness, such as Adaptive Cruise Control (ACC), Automatic Emergency Braking (AEB), Forward Collision Warning (FCW), and Lane Keeping Assist (LKA). These systems rely on a combination of sensors (radar, lidar, and cameras) to monitor the environment and intervene when safety thresholds are exceeded. Radar sensors, in particular, are the dominant modality for longitudinal safety functions such as ACC and AEB, since they directly measure range and range-rate via the Doppler effect with high temporal resolution and robustness to adverse weather and lighting conditions. Lidar sensors provide dense, accurate three-dimensional point clouds but remain comparatively expensive and, at the time of writing, less common in mass-market vehicles than radar-camera combinations.

In commercial systems, safety-critical distances are estimated primarily from radar or lidar returns, which provide direct range and range-rate measurements without requiring any geometric assumption about the scene. Camera-based estimation of the same quantities exists [35] and offers the appeal of a single, low-cost sensor that also supports semantic tasks such as sign and lane detection. However, purely visual estimation of metric distance requires solving an ill-posed monocular depth estimation problem, since a single camera view does not by itself constrain the absolute scale of the scene; distance must instead be inferred indirectly from cues such as object size priors, vanishing-point geometry, ground-plane assumptions, and perspective foreshortening. Stereo camera rigs partially alleviate this ambiguity by recovering depth through tri-

angulation, at the cost of additional hardware, calibration effort, and computational load. This thesis aims to train a VLM to predict telemetry-derived metrics directly from visual input by learning correlated visual cues like inter-vehicle gaps and perspective, effectively asking a monocular, camera-only system to approximate outputs that are conventionally the domain of dedicated radar hardware.

2.3.2 Key Safety Metrics: Time Headway and Time To Collision

Two metrics are of particular interest from a safety perspective. THW is defined as the temporal gap between the front bumper of the following vehicle and the rear bumper of the leading vehicle, assuming constant speed:

$$\mathrm{THW} = \frac{d_{\mathrm{rel}}}{v_{\mathrm{ego}}} \tag{2.1}$$

where d_{rel} is the distance to the lead vehicle in metres and v_{ego} is the ego speed in m/s. Values below 1 s are generally considered unsafe [2], and many ACC systems allow the driver to select a target headway setting (commonly between 1 and 2.5 s) that is then actively maintained by the control loop. Unlike TTC, THW does not depend on the relative speed between the two vehicles and is therefore defined at all times a lead vehicle is present, regardless of whether the gap is opening or closing, making it a useful steady-state proximity indicator in addition to its role as an instantaneous safety margin.

TTC is a more direct indicator of imminent collision risk, estimating how long until an impact occurs if velocities remain constant:

$$\mathrm{TTC} = \frac{d_{\mathrm{rel}}}{|v_{\mathrm{rel}}|} \tag{2.2}$$

where $v_{\text{rel}} = v_{\text{ego}} - v_{\text{lead}}$ is the closing speed. TTC is only defined when the gap is actively closing ($v_{\text{rel}} > 0$), making it highly informative about active collision scenarios [36]. Because TTC diverges to infinity as the closing speed approaches zero, most practical implementations, including the one adopted in this thesis (Section 4.1), impose an upper bound on the reported value, treating scenarios with a very slow closing speed as equivalently low-risk rather than reporting numerically unstable values.

Beyond THW and TTC, the ADAS literature has proposed a number of related surrogate safety measures, such as Post-Encroachment Time (PET), which measures the temporal gap between two road users occupying the same spatial point at different times, and the Deceleration Rate to Avoid a Crash (DRAC), which quantifies the braking effort required by the following vehicle to avoid a collision given the current kinematic state. These metrics are widely used in traffic-safety and infrastructure-design research but are less directly tied to a per-frame, per-vehicle telemetry signal than THW and TTC, which is why this thesis focuses on the latter two as regression targets: both admit a closed-form definition from instantaneous ego speed and radar-derived relative distance and speed, matching the synchronized telemetry available in the custom ADAS dataset described in Section 4.1.

2.4 Multi-Task Learning

2.4.1 Principles and Optimization Challenges

MTL refers to the simultaneous training of a model on multiple related tasks to improve generalization by exploiting shared structure [12]. The intuition behind MTL is that related tasks share underlying regularities in the input distribution; by forcing a model to find representations that are simultaneously

useful for several tasks, the model is biased toward more general, less task-overfit features, which can act as an implicit form of regularization, particularly when one of the tasks has comparatively little labelled data. This last property is especially relevant to the present work, where the scene-description task is supervised by a relatively small annotated corpus (Section 4.1), and may therefore benefit from the representational pressure exerted by the much larger ADAS regression task.

Ruder [13] provides a comprehensive taxonomy of MTL techniques in deep neural networks, distinguishing broadly between *hard* and *soft* parameter sharing. In hard parameter sharing, the hidden layers of the network are shared across all tasks, with only a small number of task-specific output layers kept separate; this is the historically dominant approach and provides the strongest regularization effect, since the risk of overfitting on a task-specific representation is reduced roughly in proportion to the number of tasks being learned jointly. In soft parameter sharing, by contrast, each task retains its own model and set of parameters, and similarity between the tasks' parameters is instead encouraged through a regularization term (for instance, penalizing the distance between corresponding weight matrices across task-specific networks). The system proposed in this thesis follows the *hard parameter sharing* paradigm: a single VLM backbone is shared across all tasks, assuming a common low-level visual representation, with only the lightweight ADAS regression head and the standard language-modelling head kept task-specific, as illustrated in Figure 2.1.

A well-known challenge in this domain is *task interference*, where conflicting gradient directions from different tasks harm overall performance. This phenomenon can manifest in at least two distinct ways: *negative transfer*, where the joint model performs worse on one or more tasks than an equiva-

lent single-task model would, and *gradient dominance*, where one task's loss produces gradients of much larger magnitude than another's, causing the optimizer to effectively ignore the smaller-magnitude task during early training. Strategies to address this include manual loss weighting [37], in which fixed or dynamically-normalized scalar coefficients are applied to each task's loss term to equalize their contribution to the gradient; uncertainty-based weighting [38], which treats the relative task weights as learnable parameters derived from each task's homoscedastic uncertainty, removing the need for manual tuning; and dynamic gradient projection techniques like PCGrad [39], which directly detect conflicting gradient vectors between task pairs (identified via a negative cosine similarity) and project away the conflicting component before applying the combined update. This thesis adopts manual loss weighting, setting a fixed weight on the ADAS regression task to provide stable training without the computational overhead of dynamic weighting; the specific value and its motivation are discussed in Section 4.2.

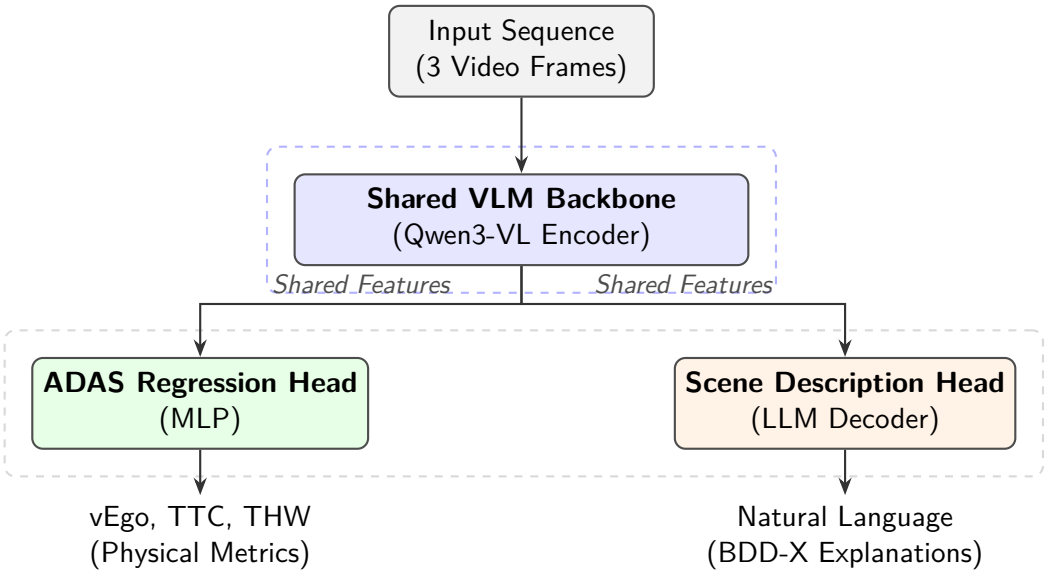


Figure 2.1: The Hard Parameter Sharing paradigm. A shared visual backbone extracts features that are routed to task-specific heads for metric regression and scene description.

2.4.2 Multi-Task Learning in Autonomous Driving

Multi-task learning is a staple in automotive perception. Early architectures like MultiNet [40] demonstrated simultaneous road detection, vehicle detection, and drivable-area segmentation within a single encoder-decoder network, showing that a shared convolutional feature extractor could support several dense-prediction heads at a fraction of the inference cost of running separate single-task networks. Modern equivalents integrate detection, tracking, mapping, motion forecasting, occupancy prediction, and planning within a single query-based architecture [4], extending the hard-parameter-sharing principle to a substantially larger and more heterogeneous set of tasks than the segmentation-only settings of earlier work. However, the combination of physical metric regression and natural language generation within a single VLM is comparatively less explored. Recent works [10; 41] focus primarily on high-level

reasoning, such as predicting discrete or textual driving decisions and control commands directly from a language-model head, but do not typically attach an auxiliary regression branch to recover continuous, physically interpretable quantities from the same shared representation. Explicitly estimating physical telemetry from VLM representations via a lightweight regression head, jointly trained alongside a natural-language generation objective, remains a novel contribution of this work.

2.5 Parameter-Efficient Fine-Tuning

Fine-tuning a pre-trained model by updating all its parameters achieves the best adaptation to downstream tasks, but its cost scales linearly with the parameter count. For a 2B-parameter model, float32 gradients and optimizer states require upwards of 24 GB of GPU memory, making full fine-tuning infeasible on consumer hardware. This memory bottleneck arises because standard optimizers such as Adam maintain two additional moment estimates per parameter, on top of the parameter itself and its gradient; in mixed or full precision training, this results in a total memory footprint of roughly 12–16 bytes per parameter once model weights, gradients, and optimizer states are all accounted for, before even considering the activation memory required for backpropagation through a multi-billion-parameter Transformer.

Parameter-Efficient Fine-Tuning (PEFT) methods address this bottleneck by updating only a small subset of parameters, or a small number of additional parameters, while keeping the bulk of the pre-trained weights frozen. Several families of PEFT methods have been proposed in the literature, differing in where and how the trainable parameters are introduced. *Adapter modules* insert small trainable bottleneck layers between the existing layers of a frozen

Transformer, learning a residual transformation of the frozen layer's output. *Prompt-tuning* and *prefix-tuning* methods instead prepend a small number of trainable continuous vectors to the input sequence (or to the keys and values of every attention layer), leaving all of the original model weights, including the embedding layer, entirely untouched. LoRA, described in detail below, instead reparameterizes the weight *update* itself as a low-rank matrix product, applied directly to the existing weight matrices rather than as a separate module. Among these families, LoRA has become particularly popular for large generative models because, unlike adapters, it introduces no additional inference latency once the low-rank update is merged back into the original weight matrix, and unlike prompt-tuning, it does not consume any of the model's limited context length.

2.5.1 Low-Rank Adaptation (LoRA and QLoRA)

LoRA [42] addresses this bottleneck by parameterizing the weight update to a matrix $W \in \mathbb{R}^{d\times k}$ as the product of two low-rank matrices:

$$W' = W_0 + \Delta W = W_0 + \frac{\alpha}{r} BA, \qquad B \in \mathbb{R}^{d\times r},\ A \in \mathbb{R}^{r\times k},\ r \ll \min(d,k) \quad (2.3)$$

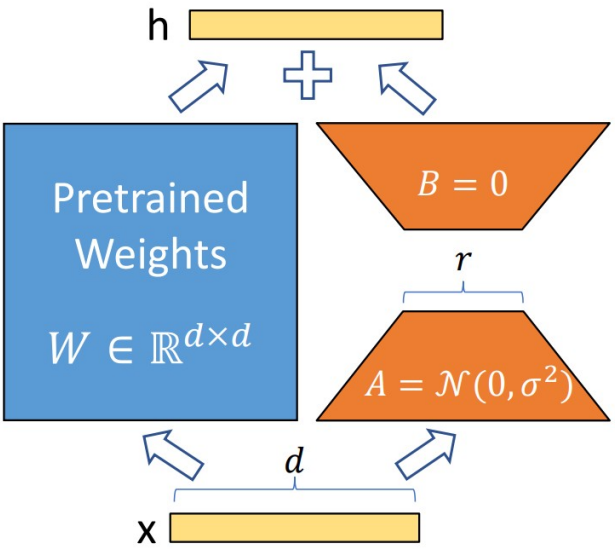


Figure 2.2: The Low-Rank Adaptation (LoRA) mechanism. The frozen pre-trained weights W_0 are updated via two trainable low-rank matrices, A and B.

where W_0 is the frozen pre-trained weight and r is the low rank. Only B and A are updated, reducing trainable parameters by a factor of over 100. Matrix A is typically initialized from a random Gaussian distribution while B is initialized to zero, ensuring that $\Delta W = 0$ at the start of training and that fine-tuning begins from the exact pre-trained behaviour of the model. The scaling factor α/r controls the effective magnitude of the update and is usually treated as a hyperparameter alongside the rank r itself; because the update is a simple additive term, at inference time BA can optionally be merged directly into W_0, so that a LoRA-adapted model carries no additional latency or memory overhead compared to a fully fine-tuned model of the same architecture. This property distinguishes LoRA from adapter-based methods, which introduce extra layers, and from prompt-tuning methods, which consume part of the model's context window.

QLoRA [15] extends this by quantizing the frozen backbone to 4-bit preci-

sion (NormalFloat4 (NF4)). Matrix multiplications are dequantized on-the-fly to 16-bit (bfloat16) before computation, allowing the LoRA adapters to operate at full precision. This dual approach dramatically reduces memory footprint, enabling the fine-tuning of multi-billion parameter models on single GPUs. Beyond the NF4 quantization scheme itself, QLoRA introduces two further optimizations that are relevant to the training configuration adopted in this thesis (Section 4.2): *double quantization*, which further compresses the quantization constants themselves to reduce memory overhead by an additional fraction of a bit per parameter, and *paged optimizers*, which use NVIDIA's unified memory feature to automatically offload optimizer states to Central Processing Unit (CPU) memory during transient GPU memory spikes, preventing out-of-memory failures when processing long sequences or large batches on a single consumer-grade GPU.

2.6 Explainability in Autonomous Driving

2.6.1 Textual Explanations and the BDD-X Dataset

End-to-end neural networks for autonomous driving are often opaque, producing control outputs without exposing intermediate reasoning. Textual explanations offer a natural form of post-hoc interpretability, satisfying safety and regulatory requirements [43]. Compared to alternative explainability techniques, such as saliency maps or attention visualizations, which highlight *where* in the input a model is focusing but not *why* a particular decision was made, natural language explanations have the advantage of being directly interpretable by non-expert stakeholders, including passengers, regulators, and accident investigators, without requiring specialized knowledge of the underlying model architecture.

The BDD-X dataset [14] serves as a primary benchmark for this, providing dense temporal annotations for driving videos. Human annotators marked time intervals for specific actions (e.g., braking, lane changes) and provided justifications, such as *"The car is slowing down because there is a stop sign."* BDD-X itself is built on top of the larger BDD100K dataset [44], a diverse driving video collection spanning multiple cities, weather conditions, and times of day, originally introduced to support a range of heterogeneous perception tasks including detection, segmentation, and lane marking; BDD-X augments a subset of these videos with the paired action-justification annotations described above. In this thesis, BDD-X provides the supervision signal for the model's scene description branch, and its relatively constrained size and vocabulary, discussed further in Section 4.1, is an important factor in interpreting the scene-description results reported in Chapter 5.

2.6.2 Related VLM-Based Approaches for Driving

Several recent works apply large VLMs to driving scenes. DriveGPT4 [8] prompts GPT-4V for explanations, while LMDrive [11] and DriveLLM [9] focus on end-to-end driving and control prediction. DriveGPT4 in particular distils explanations and control signals from a proprietary, closed large multimodal model into a smaller open-source student model, illustrating one alternative strategy to the direct supervised fine-tuning approach adopted in this thesis. LMDrive instead couples a language-instruction-following module with a closed-loop driving policy, targeting instruction-conditioned navigation rather than the joint estimation of continuous safety metrics and natural-language descriptions. Compared to these, this thesis distinguishes itself by utilizing a compact, open-source backbone amenable to single-GPU training, and explicitly combining text generation with the regression of physically interpretable

safety metrics (TTC, THW) through a custom masked loss function, an objective that, to the best of the author's knowledge, has not been directly addressed by the works surveyed above.

Chapter 3

System's Architecture

3.1 Overview

The proposed system is built around a single VLM backbone shared across two tasks. Rather than designing separate pipelines for scene understanding and physical metric estimation, a unified architecture was developed in which a single model processes multi-frame visual input and produces outputs on both branches simultaneously. This design choice is motivated by the hypothesis that the rich visual representations learned by large-scale VLMs encode information relevant to both semantic reasoning and physical state estimation, and that multi-task training on both objectives can lead to mutually beneficial feature learning.

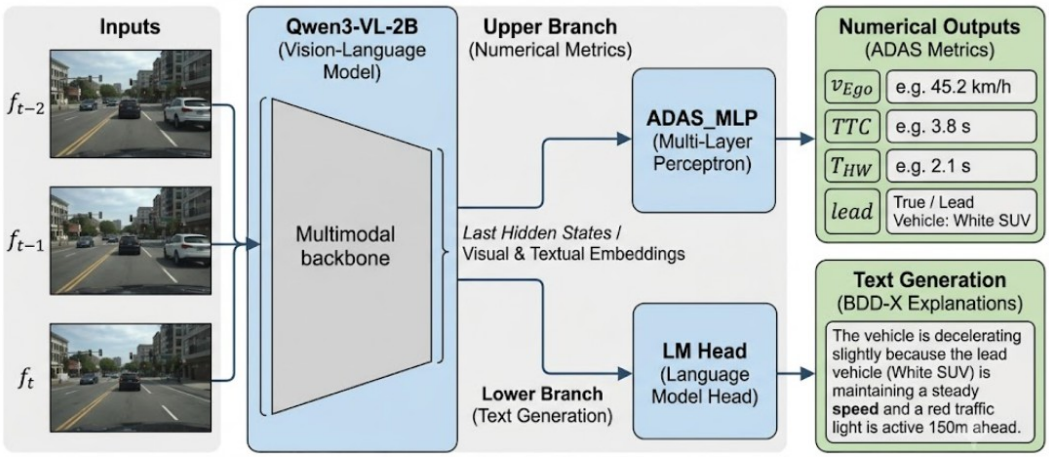


Figure 3.1: High-level system diagram illustrating the multimodal architecture.

3.2 Backbone: Qwen3-VL-2B-Instruct

The backbone of the system is Qwen3-VL-2B-Instruct [34], a 2-billion parameter vision-language model from the Qwen3 family. The model uses a hybrid architecture combining a ViT image encoder with a causal language model decoder. It natively supports multi-image inputs within a single conversation turn, which makes it well-suited for processing sequential frame inputs without requiring modifications to the input pipeline.

Input images are processed through a dynamic resolution mechanism: each image is tiled into non-overlapping patches and passed through the visual encoder, with the number of patches (and hence the visual token count) depending on the image resolution. To balance computational cost and visual fidelity during training, minimum and maximum pixel counts were enforced:

$$P_{\min} = 128 \times 28, \quad P_{\max} = 320 \times 28 \tag{3.1}$$

where 28 is the ViT patch size in pixels.

Each 3-frame sequence is passed to the model as a user message containing three image blocks followed by a task-specific text prompt. The model's processor applies the official chat template to construct the full input token sequence.

3.3 Task 1: Physical Regression via MLP Head

3.3.1 Architecture

For the ADAS regression task, the final hidden state of the language model backbone was used as a compact representation of the visual scene. Specifically, the hidden state corresponding to the last non-padding token in the sequence was extracted:

$$\mathbf{h} = \mathrm{LM}(x)[-1, :] \in \mathbb{R}^d \tag{3.2}$$

where d is the hidden dimension of the language model, determined by `model.config.text_config.hidden_size`.

This representation was then passed through a lightweight MLP regression head, termed `ADAS_MLP`. Rather than simply listing its sequential layers, it is instructive to outline its design rationale. The head is structured as a three-layer funnel architecture that progressively compresses the high-dimensional visual-semantic features extracted by the language model down to a compact physical state representation.

Specifically, the input vector of dimension d is first projected into an intermediate 512-dimensional space, followed by a second reduction to 128 dimensions, and finally mapped to the 4-dimensional output target. To ensure training stability across diverse feature scales and to facilitate smooth gradient flow, Layer Normalization and ReLU activation functions are applied after the

first two linear transformations. Furthermore, a dropout layer (with a probability of $p = 0.3$) is introduced early in the network pipeline. This acts as a crucial regularizer to prevent overfitting on the specific telemetry distribution of the training set, encouraging the network to learn robust, distributed representations of the physical scene.

The head produces a 4-dimensional output vector: $[\hat{v}_{\text{ego}}, \hat{t}_{\text{TTC}}, \hat{t}_{\text{THW}}, \hat{l}]$, where the first three are continuous regression outputs and $\hat{l}$ is a raw logit representing lead vehicle presence.

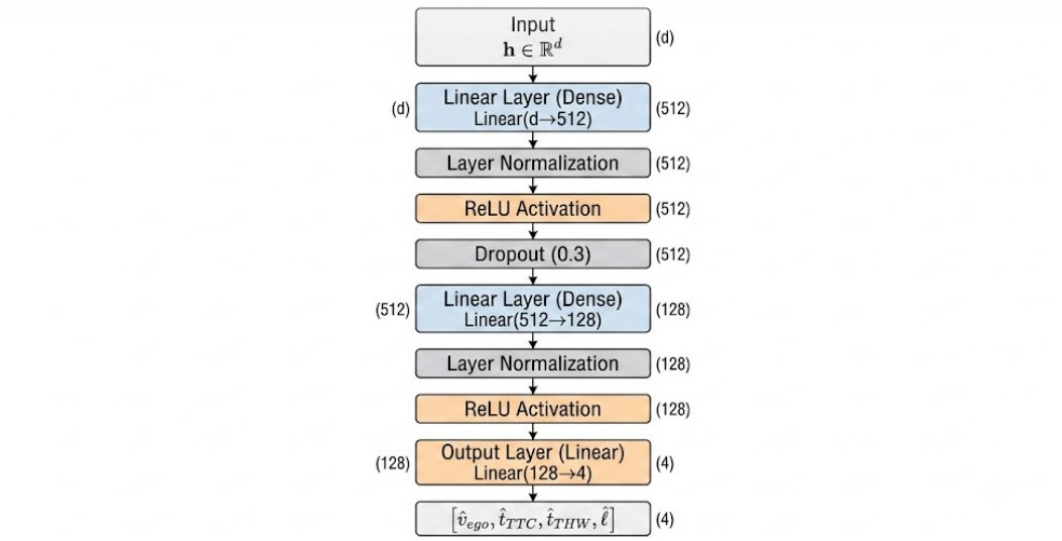


Figure 3.2: ADAS_MLP architecture diagram

3.3.2 Target Normalization

Prior to training, the three continuous targets (`vEgo`, TTC, THW) were standardized using the mean and standard deviation computed on the training split:

$$\tilde{y} = \frac{y - \mu_{\text{train}}}{\sigma_{\text{train}} + \epsilon} \tag{3.3}$$

The normalization statistics were saved alongside the MLP checkpoint so that predictions can be correctly de-normalized at inference time.

3.3.3 Masked Regression Loss

A key challenge in the ADAS task is that TTC and THW are only physically defined when a lead vehicle is present. Naively training on these values when set to -1.0 would introduce spurious gradients. To address this, a Masked Regression Loss was designed:

$$\mathcal{L}_{\text{ADAS}} = 2.0 \cdot \mathcal{L}_{v_{\text{ego}}} + 1.0 \cdot \mathcal{L}_{\text{lead}} + 0.5 \cdot \mathcal{L}^*_{\text{TTC}} + 0.5 \cdot \mathcal{L}^*_{\text{THW}} \tag{3.4}$$

where:

- $\mathcal{L}_{v_{\text{ego}}} = \text{MAE}(\hat{v}_{\text{ego}}, v_{\text{ego}})$ is an unmasked L1 regression loss on ego speed, which is always defined.
- $\mathcal{L}_{\text{lead}} = \text{BCEWithLogits}(\hat{l}, m)$ is a binary cross-entropy classification loss on lead vehicle presence, where $m \in \{0, 1\}$ is the ground-truth mask.
- $\mathcal{L}^*_{\text{TTC}}$ and $\mathcal{L}^*_{\text{THW}}$ are masked L1 losses, computed only over the subset of samples in the batch where a lead vehicle is present ($m = 1$):

$$\mathcal{L}^*_{\text{TTC}} = \frac{\sum_i m_i \cdot |\hat{t}_i - t_i|}{\sum_i m_i} \tag{3.5}$$

If no lead vehicle is present in any sample in a batch, these terms are set to zero. This formulation ensures that the model is never penalized for TTC/THW predictions on frames where these quantities are undefined. The ego speed loss was assigned a weight of 2.0, reflecting its fundamental importance to safety and the fact that it is observable in every frame regardless of traffic conditions; a detailed rationale is provided in Section 4.2.1.

3.4 Task 2: Scene Description via Autoregressive Generation

For the BDD-X task, the model was trained end-to-end for conditional text generation. Each training sample consists of a full conversation: the user's 3-frame visual input and text prompt, followed by the ground-truth scene description as the assistant's response. The standard causal language modeling loss (cross-entropy over the target token sequence) was used:

$$\mathcal{L}_{\text{BDD}} = -\sum_{t} \log P_\theta(y_t \mid y_{<t}, x) \tag{3.6}$$

where x is the multi-modal input and y is the tokenized ground-truth description. Labels were set to be identical to the input IDs, following the standard Supervised Fine-Tuning (SFT) convention. Within the HuggingFace `Trainer` framework, tokens corresponding to padding or the user prompt are assigned a label of -100, which is ignored by the cross-entropy loss, ensuring that only the assistant's response tokens contribute to the gradient signal.

3.5 Model Training Paradigms

Two distinct training paradigms were investigated to evaluate the inherent capabilities of the pre-trained VLM versus the benefits of task-specific adaptation.

The first approach serves as a computationally inexpensive baseline, where the Qwen3-VL-2B backbone is utilized strictly as a static feature extractor. In this configuration, all backbone parameters remain frozen, and only the `ADAS_MLP` regression head is updated. This isolates how much information relevant to physical metric estimation is already encoded in the pre-trained

visual representations without requiring any domain-specific fine-tuning.

The second, primary approach performs joint end-to-end training of both the backbone and the regression head, simultaneously optimizing for scene description and physical regression. Given the 2-billion parameter scale of the backbone, full fine-tuning is computationally prohibitive on a single consumer GPU. Therefore, QLoRA [15] was employed. The backbone was loaded in 4-bit NF4 quantization using `bitsandbytes`, with double quantization enabled and computations performed in `bfloat16`. To adapt the network, low-rank matrices (rank $r = 16$, scaling factor $\alpha = 32$, dropout $p = 0.05$) were injected into all linear layers of the backbone:

$$W' = W + \frac{\alpha}{r} BA, \quad B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k} \tag{3.7}$$

During this multi-task training phase, only these LoRA parameters and the `ADAS_MLP` head were updated, keeping the quantized backbone weights strictly frozen. Additionally, gradient checkpointing was enabled to further reduce Video Random Access Memory (VRAM) usage.

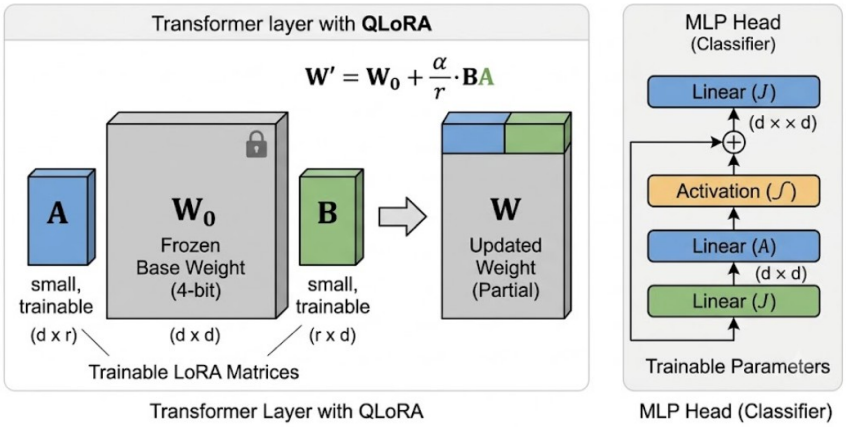


Figure 3.3: Overview of the QLoRA fine-tuning paradigm. The pre-trained backbone is frozen and quantized to 4-bit precision, while lightweight LoRA adapters and the ADAS regression head are trained simultaneously.

Chapter 4

Methodology

4.1 Dataset Construction

This work relies on two complementary datasets, each designed to support one branch of the multi-task learning objective. The first, derived from real-world driving telemetry logs, provides ground-truth physical measurements of the ego vehicle and its immediate lead vehicle. The second is a publicly available annotated driving benchmark providing paired video sequences and natural language descriptions of driving events. Together, these datasets enable a single model to be trained for both physical regression and semantic scene understanding.

4.1.1 ADAS Telemetry Dataset

The ADAS dataset was built from a collection of raw driving segments recorded on an OpenPilot-compatible platform. Each segment consists of three files stored together in the same directory: a front-facing dashcam video (`takeover.mp4`), a vehicle state log (`carState.csv`) containing timestamped ego speed (`vEgo`) measurements, and a radar log (`radarState.csv`) containing timestamped measurements of the lead vehicle's presence (`leadOne.status`), rela-

tive distance (`leadOne.dRel`), and relative speed (`leadOne.vRel`).

From this raw radar and vehicle state data, two safety-critical metrics were dynamically derived as target variables. THW is defined as the time it would take for the ego vehicle to reach the current position of the lead vehicle while maintaining its current speed:

$$\mathrm{THW} = \frac{d_{\mathrm{rel}}}{v_{\mathrm{ego}}} \tag{4.1}$$

This quantity is only meaningful when a lead vehicle is present and the ego vehicle is moving. THW values were set to -1.0 otherwise, acting as a sentinel for the masked loss computation described in Section 3.3.3.

TTC estimates how long until a collision would occur given the current closing speed between the two vehicles:

$$\mathrm{TTC} = \frac{d_{\mathrm{rel}}}{|v_{\mathrm{rel}}|} \tag{4.2}$$

TTC is only defined when the lead vehicle is present and the relative speed is negative (i.e., the gap is closing). It was likewise set to -1.0 otherwise. Both TTC and THW were capped at maximum values of 10.0 s and 5.0 s respectively. These caps correspond approximately to the 95th percentile of the distributions in the training split (see Figure 4.2), beyond which the metrics carry little safety-relevant information and represent outliers in lightly trafficked road segments.

To properly extract the training sequences, the raw driving videos, which were recorded at varying native frame rates (typically 20–30 fps), required temporal alignment. A logical subsampling scheme at 10 Hz was employed to ensure temporal consistency across all segments: frame indices in the original video were mapped to their equivalent positions in a virtual 10 Hz timeline,

avoiding the need to re-encode any video. Each training sample consists of a sequence of 3 frames spanning a 1-second temporal window. Frames are spaced at equal intervals of 0.5 seconds, corresponding to virtual indices $[0, 5, 10]$ at 10 Hz. A sliding window approach with a stride of 0.5 seconds (5 frames at 10 Hz) was used to densely extract overlapping sequences from the full video duration. Ground-truth telemetry labels were matched to each sequence by aligning the timestamp of the last frame in the window to the nearest telemetry sample.

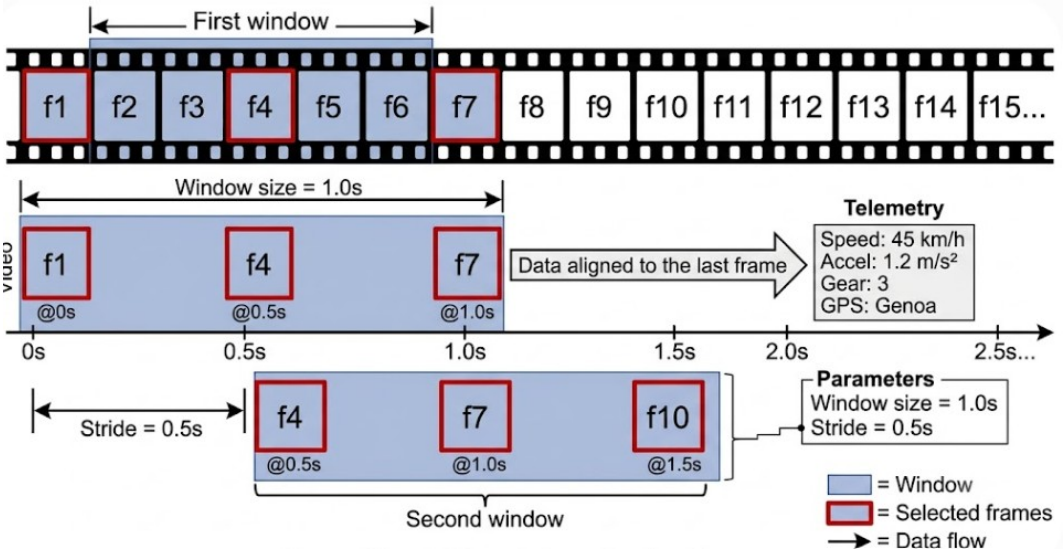


Figure 4.1: Diagram illustrating the sliding window extraction procedure

Table 4.1: ADAS telemetry dataset statistics. Values are computed across the full dataset and the training split.

Property	**Value**
Total sequences extracted	60,284
Sequences with lead vehicle	26,258 (43.6 %)
Sequences without lead vehicle	34,026 (56.4 %)
Training split (80 %)	48,227
Validation split (10 %)	6,028
Test split (10 %)	6,029
Mean vEgo (train)	50.0 km/h
Mean TTC (train, lead present)	8.69 s
Mean THW (train, lead present)	2.73 s

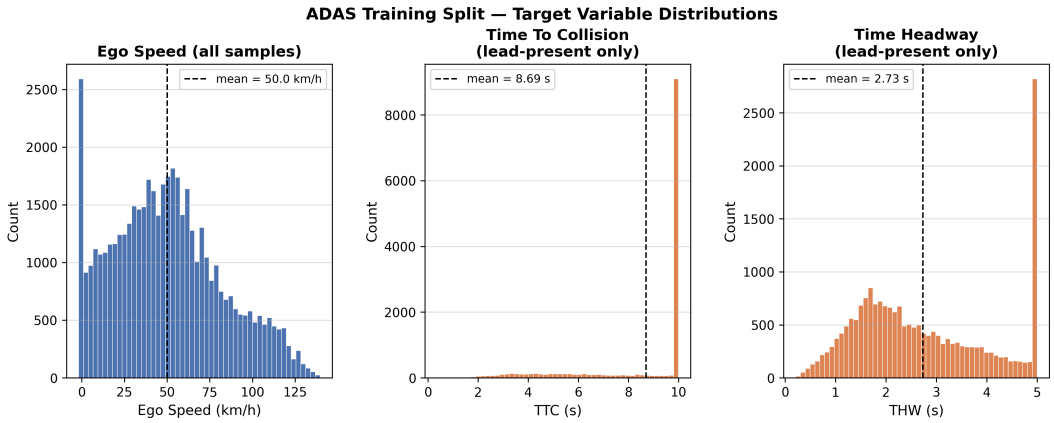


Figure 4.2: Distributions of target variables (Ego Speed, TTC, and THW) within the ADAS training split.

4.1.2 BDD-X Natural Language Dataset

The BDD-X dataset [14] extends the large-scale BDD100K driving dataset with natural language annotations. Each video in the dataset is accompanied by a structured annotation file containing up to 15 intervals, each with a start timestamp, a textual description of the vehicle action (e.g., ”the car is slowing down”), and a justification of that action (e.g., ”because there is a stop sign ahead”).

To ensure a uniform input format across both tasks, video frames for BDD-X were extracted using the exact same 10 Hz logical subsampling scheme and 3-frame, 1-second window representation used for the ADAS dataset. For each annotated interval, extraction began at the interval's specified start time, and frames were resized to a standard height of 720 pixels to reduce disk usage while preserving the aspect ratio.

The ground-truth text label for each sequence was constructed by concatenating the action and justification fields ("{action} {justification}"). This produces a concise, self-contained sentence describing both what the vehicle is doing and why. It should be noted that this yields a relatively small dataset for sequence-to-sequence language generation (1,996 total sequences); the implications of this constrained supervision scale for overall model performance are discussed in Chapter 6.

Table 4.2: BDD-X dataset statistics. Text metrics are computed on the training split.

Property	**Value**
Total sequences extracted	1,996
Training split (80 %)	1,596
Validation split (10 %)	200
Test split (10 %)	200
Mean words per description	13.9
Max words in a description	51
Vocabulary size (approx.)	631

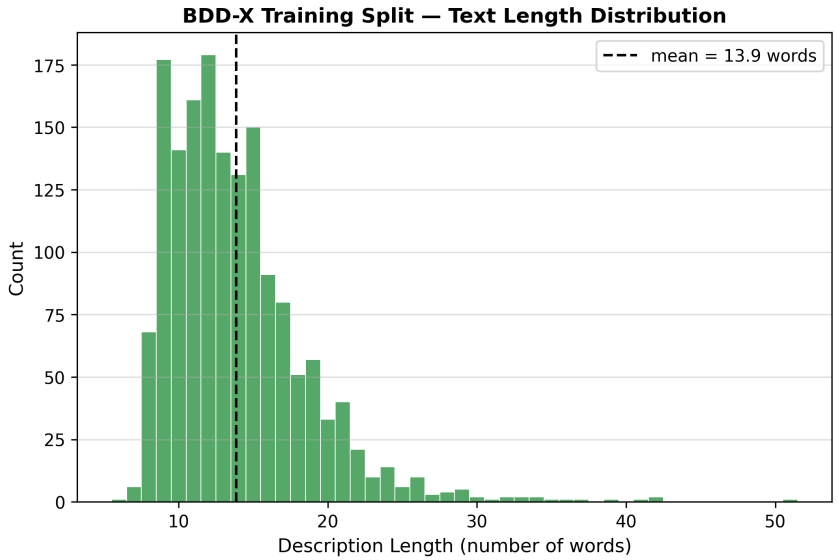


Figure 4.3: Distribution of sequence descriptions length (word count) within the BDD-X training split.

4.2 Training Configuration

4.2.1 Multi-Task Training Strategy

A central challenge in multi-task learning is determining how and when to mix gradients from different tasks. A naive approach, alternating between tasks epoch-by-epoch, can cause task interference if one task dominates early training. Instead, a proportional random interleaving strategy was used: for each epoch, the full set of ADAS batches and BDD-X batches was merged into a single list, shuffled randomly, and iterated over in that order. This ensures that both tasks are encountered with roughly equal frequency throughout every epoch, while preventing any systematic ordering bias.

The combined training loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BDD}} + w_{\text{ADAS}} \cdot \mathcal{L}_{\text{ADAS}} \tag{4.3}$$

where $w_{\text{ADAS}} = 2.0$ was chosen to compensate for the fact that the BDD-X loss is averaged over a variable-length token sequence (which can dilute the gradient signal), while the ADAS loss is computed over a fixed 4-dimensional output.

4.2.2 Optimization and Early Stopping

All trainable parameters, including both the LoRA adapters and the `ADAS_MLP` head, were jointly optimized using AdamW with a learning rate of 2×10^{-4} and a cosine learning rate schedule with a 10% warmup phase. The total number of optimizer steps was computed as:

$$N_{\text{steps}} = \frac{(N_{\text{ADAS}} + N_{\text{BDD}}) \cdot E}{k} \tag{4.4}$$

where N_{ADAS} and N_{BDD} are the number of batches in each training loader, $k = 2$ is the gradient accumulation factor, and $E = 60$ is the maximum number of training epochs. Gradient accumulation over $k = 2$ steps was utilized to effectively double the batch size without increasing VRAM requirements beyond what a single-GPU setup can support.

To prevent overfitting, the training process was monitored continuously using a combined validation loss:

$$\mathcal{L}_{\text{val}} = \mathcal{L}_{\text{val,BDD}} + w_{\text{ADAS}} \cdot \mathcal{L}_{\text{val,ADAS}} \tag{4.5}$$

An early stopping mechanism with a patience of 5 epochs was implemented; training was halted if the combined validation metric failed to improve for 5 consecutive epochs. The checkpoint corresponding to the lowest combined validation loss was then preserved as the definitive `best_model`.

4.2.3 Hyperparameters Summary

Table 4.3 outlines the complete set of architectural, optimization, and data-processing hyperparameters applied during the multi-task training phase. The values selected reflect a deliberate balance between hardware constraints and convergence requirements. Specifically, 4-bit NF4 quantization and LoRA adapters ($r = 16$) were necessary to fit the 2-billion parameter backbone within the VRAM limits of a consumer GPU, while gradient accumulation and a learning rate of 2×10^{-4} ensured a stable optimization trajectory across both the physical regression and text generation tasks.

Table 4.3: Summary of training hyperparameters

Hyperparameter	**Value**
Backbone	Qwen3-VL-2B-Instruct
Quantization	4-bit NF4 + double quant
LoRA rank r	16
LoRA alpha α	32
LoRA dropout	0.05
LoRA target modules	all-linear
Batch size	4
Gradient accumulation steps	2
Max epochs	60
Learning rate	2×10^{-4}
Optimizer	AdamW
LR schedule	Cosine + warmup (10%)
ADAS loss weight	2.0
Early stopping patience	5
Min pixels	128×28
Max pixels	320×28
Sequence length cap	2048 tokens

4.3 Inference System

4.3.1 Real-Time Video Processing Pipeline

To validate the trained system on unseen video sequences, a real-time inference pipeline was developed. The pipeline accepts a raw driving video as input and produces an annotated output video with a HUD overlaying both the model's predictions and the available ground-truth telemetry.

For each video frame, a sliding buffer of the 3 most recent keyframes (sampled at the same temporal spacing used during training) is maintained. Every 0.5 s, the current buffer is passed to the model for inference. On the testing hardware (NVIDIA GeForce RTX 4080), each dual-pass inference step requires an average of approximately 1050 ms (ranging from ∼640 ms to ∼1840 ms depending on background OS activity). While this indicates a computational overhead that slightly exceeds the strict 0.5 s real-time threshold, the system remains highly performant for offline analysis and provides a strong foundation for future single-pass optimization.

Two sequential forward passes are performed per inference step:

1. **BDD-X pass**: The frame buffer is processed with the scene description prompt. The model generates up to 30 new tokens using greedy decoding.

2. **ADAS pass**: The same frame buffer is processed with the ADAS prompt. The last hidden state is extracted and passed through the `ADAS_MLP` head. The resulting predictions are de-normalized using the training set statistics and converted to human-readable units (km/h for speed, seconds for TTC/THW).

Lead vehicle presence is determined by thresholding the sigmoid of the fourth MLP output at 0.6. If no lead vehicle is predicted, TTC and THW are

displayed as ”N/A”.

Figure 4.4: Screenshots of the real-time HUD under various driving scenarios.

4.3.2 Anti-Flicker Semantic Filter

A known failure mode of autoregressive VLMs in video contexts is text flickering: the generated description changes abruptly between consecutive inference steps even when the visual scene has not meaningfully changed, producing an unstable overlay that is distracting in a real-time context.

To address this, a semantic similarity filter was applied using a lightweight sentence embedding model (`all-MiniLM-L6-v2` from `sentence-transformers`). After each new generation, the cosine similarity between the new prediction and the currently displayed text was computed:

$$s = \frac{\mathbf{e}_{\text{new}} \cdot \mathbf{e}_{\text{curr}}}{|\mathbf{e}_{\text{new}}||\mathbf{e}_{\text{curr}}|} \tag{4.6}$$

The displayed text was updated only if $s < \tau$, where $\tau = 0.50$. This value was selected empirically by visual inspection of a held-out validation video; a formal sensitivity analysis is left as future work. The threshold allows substantial semantic changes to pass while successfully suppressing synonymous or paraphrased reformulations of the same scene description.

Figure 4.5 illustrates the operating principle of the filter under two different conditions. In Scenario A, the model paraphrases the previous output; the sentence embeddings retain high cosine similarity ($s \geq 0.50$) and the display update is suppressed to prevent flickering. In Scenario B, a genuine scene change occurs (e.g., stopping for a red light); the embeddings diverge ($s < 0.50$) and the HUD is subsequently updated with the new text.

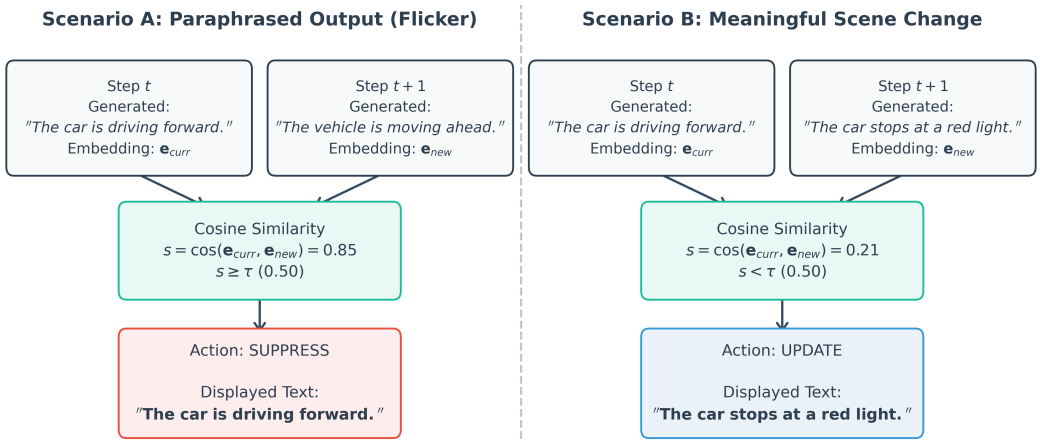


Figure 4.5: Illustration of the semantic anti-flicker filter in action.

Chapter 5

Experimental Results

5.1 ADAS Regression Results

The performance of the physical metric estimation branch was evaluated using MAE for the continuous variables and accuracy for the binary lead vehicle detection. To ensure fair evaluation, the MAE for TTC and THW was computed exclusively on the subset of test samples where a lead vehicle was actually present.

Table 5.1 compares the two training paradigms: Approach 1 (frozen VLM backbone with a trained MLP head for 1000 epochs) and Approach 2 (end-to-end multi-task fine-tuning via QLoRA). It should be noted that the two approaches are not perfectly controlled experiments: Approach 1 was trained for 1,000 epochs on the ADAS task alone, while Approach 2 underwent multi-task QLoRA training for approximately 25 epochs before early stopping. The observed improvement in Approach 2 therefore reflects the combined effect of backbone adaptation and joint semantic supervision, whose individual contributions are left for future ablation studies.

Table 5.1: ADAS test set results comparing the frozen backbone baseline (Approach 1) against the end-to-end fine-tuned multimodal architecture (Approach 2).

Method	**Det. Acc.**	**v_{ego} MAE**	**TTC MAE**	**THW MAE**
Approach 1 (Frozen)	86.9 %	10.15 km/h	3.62 s	0.68 s
Approach 2 (QLoRA)	92.0 %	5.32 km/h	2.97 s	0.35 s

Further analysis of the ADAS regression task under Approach 2 is provided in Figure 5.1, which displays scatter plots comparing ground-truth and predicted values on a subsample of the test set. The dashed line represents the ideal perfect prediction ($y = x$), while points are colored based on local density (Kernel Density Estimation (KDE)).

The model achieves a remarkably strong correlation ($R^2 = 0.946$) on Ego Speed, with predictions tightly clustered along the diagonal across the entire velocity range (from 0 to over 130 km/h). This indicates that the VLM is highly adept at extracting ego-motion cues, likely leveraging implicit optical flow and perspective shifts across the 3-frame input sequence.

In contrast to the purely ego-centric speed estimation, the spatial metrics reveal the inherent challenges of monocular depth perception. THW demonstrates a solid predictive capability ($R^2 = 0.830$, MAE = 0.38 s), maintaining a generally linear relationship with the ground truth. However, the variance increases visibly at higher headway values, and there is a noticeable accumulation of predictions at the 5.0 s upper cap, representing scenarios where the lead vehicle is small and distant.

Notably, TTC exhibits a severe drop in performance with a negative R^2 of -1.587, indicating that the model's predictions are more dispersed than a constant mean predictor. The scatter plot shows a wide spread of points and heavy clustering at the 10.0 s maximum cap. This reflects the fundamental difficulty of estimating this specific safety metric from monocular vision alone.

While THW depends only on absolute distance and ego speed (which the network predicts very well), TTC depends on the relative closing speed. Effectively, this requires the network to estimate the time derivative of an unscaled monocular depth, compounding the error and making accurate frame-by-frame TTC regression highly unstable without explicit geometric constraints or radar fusion.

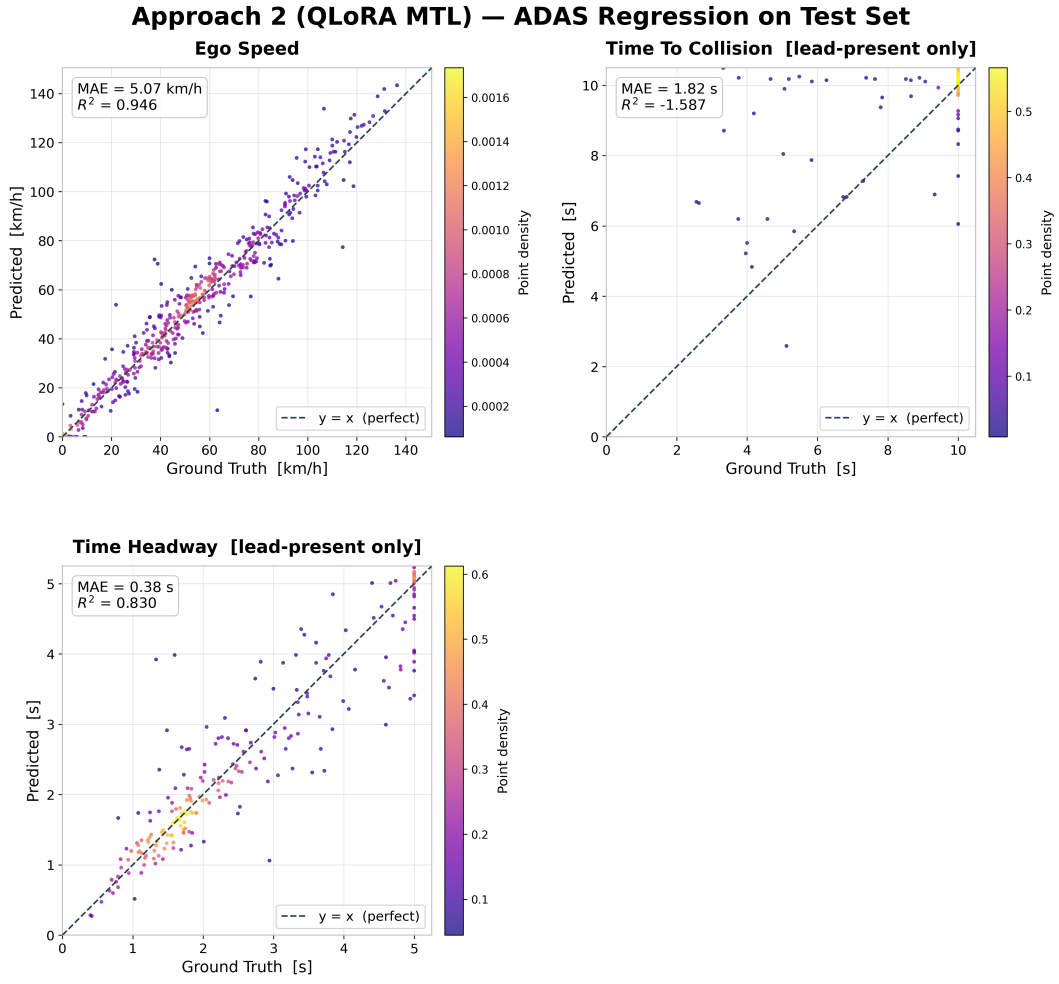


Figure 5.1: Scatter plots comparing ground-truth and predicted ADAS metrics.

5.2 Scene Description Results

The model's natural language generation performance was evaluated on the held-out BDD-X test set. Traditional n-gram matching metrics (BLEU, ROUGE-L) and modern semantic similarity metrics (BERTScore, ST-Sim) were computed to assess both lexical and semantic fidelity.

Table 5.2: Scene description generation performance on the BDD-X test set.

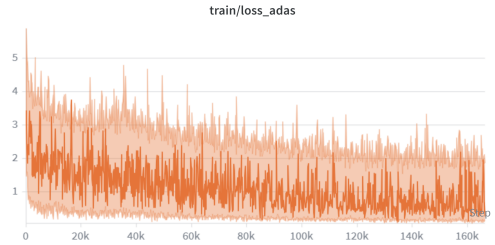
Method	**BLEU**	**ROUGE-L**	**BERTScore**	**ST-Sim**
Approach 2 (QLoRA)	0.0982	0.3821	0.9124	0.5792

The disparity between the high BERTScore (0.9124) and the lower n-gram metrics (BLEU-4: 0.0982) suggests that the model generates semantically coherent descriptions using its own vocabulary, rather than strictly memorising the BDD-X annotation style. However, to contextualise the high BERTScore, a naive baseline constantly predicting a generic phrase (e.g., "The car is driving forward") yields a BERTScore of 0.9160. This remarkably high baseline score highlights a known limitation of using contextual embeddings in highly constrained domains: generic but syntactically valid sentences can achieve semantic similarities comparable to the fine-tuned model (0.9124) simply by matching high-frequency driving vocabulary. Therefore, the model's numerical performance must strictly be interpreted alongside the qualitative outputs to verify true causal reasoning.

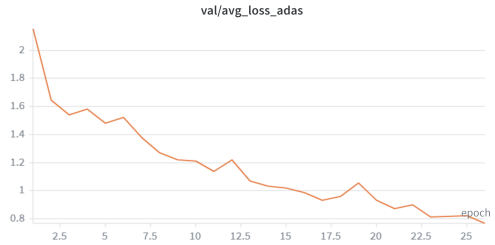
5.3 Training Curves

Figure 5.2 displays the Weights & Biases training logs. The left column shows the step-wise training metrics, demonstrating the learning rate warm-up and cosine decay alongside the task-specific training losses. The right column

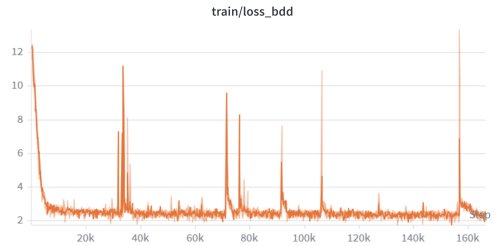
presents the epoch-wise validation metrics, illustrating the steady convergence of the model on unseen data. The absence of overfitting on the ADAS validation loss and the monotonically decreasing BDD-X validation loss confirm that the proportional interleaving strategy successfully prevents task dominance throughout training.



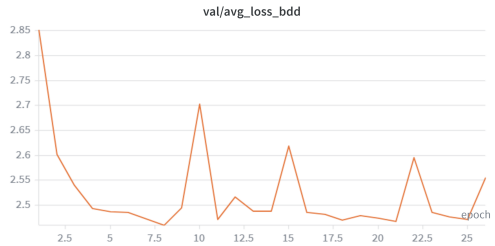
(a) Training Loss (ADAS)



(b) Validation Loss (ADAS)

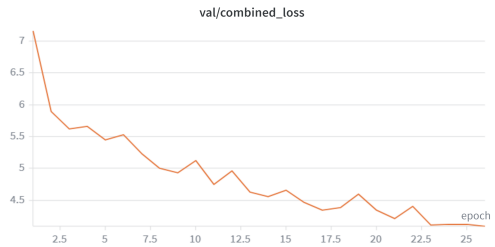


(c) Training Loss (BDD-X)



(d) Validation Loss (BDD-X)

(e) Learning Rate Schedule



(f) Validation Combined Loss

Figure 5.2: Training and validation logs.

5.4 Qualitative Analysis

To better understand the generation capabilities and the typical failure modes of the end-to-end fine-tuned model (Approach 2), a qualitative analysis was conducted on the BDD-X test set. Predictions were automatically stratified based on their semantic similarity (ST-Sim) to the ground truth to select unbiased, representative examples.

Figure 5.3 illustrates three driving scenarios categorized by their ST-Sim scores. In the Good case (ST-Sim ≥ 0.70), the model demonstrates near-perfect reasoning, correctly identifying both the stopped vehicle and the red light. In the Mediocre case ($0.40 \leq$ ST-Sim < 0.70), the model struggles with complex lighting (nighttime/reflections), hallucinating a red light stop instead of recognizing a right turn. Finally, the Failure case (ST-Sim < 0.40) shows a severe misinterpretation in a cluttered daytime scene, predicting forward movement instead of yielding to construction workers and merging traffic.

In summary, the qualitative analysis reveals that the model's failure modes are predominantly perceptual (such as lighting and severe clutter) rather than reasoning-based. In well-lit, unambiguous scenes, the model produces descriptions that are both factually correct and causally grounded, suggesting that the multi-task training has successfully transferred semantic reasoning capabilities to driving scenarios.

Figure 5.3: Qualitative results on the BDD-X test set, categorized by ST-Sim scores.

Chapter 6

Conclusions

This thesis investigated whether a single VLM can simultaneously serve as a physical metric estimator and a natural language scene describer for autonomous driving, without sacrificing performance on either objective. The proposed system, built around Qwen3-VL-2B-Instruct as a shared backbone, was trained jointly on a custom ADAS telemetry dataset and the BDD-X benchmark [14] through a proportional random interleaving strategy, and evaluated under two distinct training paradigms: a frozen-backbone baseline in which only the ADAS MLP regression head was optimised (Approach 1), and a full end-to-end approach in which both the backbone adapters, via QLoRA [15], and the regression head were updated simultaneously (Approach 2).

The experimental results demonstrate that the answer to the central research question is affirmative: a compact 2-billion-parameter VLM backbone can encode representations that are simultaneously useful for semantic scene understanding and physical state estimation. As reported in Table 5.1, the end-to-end QLoRA approach substantially outperforms the frozen backbone baseline across all ADAS metrics, reducing ego speed MAE by 47.6% (from 10.15 to 5.32 km/h), THW MAE by 48.5% (from 0.68 to 0.35 s), and TTC MAE by 18.0% (from 3.62 to 2.97 s), while improving lead vehicle detection

accuracy from 86.9% to 92.0%.

The strong performance of Approach 1 itself is a noteworthy finding. Achieving a lead vehicle detection accuracy of 86.9% and an ego speed MAE of 10.15 km/h without any task-specific backbone adaptation confirms that the visual representations encoded by a large-scale pre-trained VLM already carry substantial information about physical scene properties. This supports the hypothesis that VLMs trained on internet-scale image–text data implicitly learn visual cues, such as inter-vehicle spacing, perspective foreshortening, and apparent road texture, that correlate with safety-critical quantities even in the absence of explicit metric supervision.

The physical regression branch achieves its strongest results on ego speed ($R^2 = 0.946$, MAE = 5.07 km/h), as visible in Figure 5.1. This is consistent with the observation that speed-related cues such as optical flow, motion blur, and the rate of change of perspective are well-represented in the three-frame dashcam sequence. TTC and THW, being fundamentally depth-dependent quantities, exhibit considerably higher estimation variance ($R^2_{\text{TTC}} = -1.587$, $R^2_{\text{THW}} = 0.830$). The negative R^2 for TTC reflects the inherent ill-posedness of inferring an absolute metric distance and its time derivative from monocular images alone, where scale is not recoverable without additional geometric constraints. This result should be interpreted not as a failure of the model, but as a confirmation of the fundamental difficulty of the task: even dedicated monocular TTC estimation networks struggle to outperform simple baselines on unconstrained datasets [35].

On the scene description task, the fine-tuned model achieves a BERTScore F1 of 0.9124 and an ST-Sim cosine similarity of 0.5792 on the BDD-X test set (Table 5.2), indicating strong semantic fidelity to ground-truth annotations even when exact lexical overlap is limited (BLEU-4: 0.0982, ROUGE-L: 0.3821).

The disparity between high semantic scores and low n-gram scores is consistent with a model that has learned to describe driving scenes in its own vocabulary rather than replicating the specific phrasing of the BDD-X annotations. Qualitative analysis in Figure 5.3 reveals that the model generalises well to clear, well-lit daytime scenarios, correctly identifying both the vehicle action and its visual cause, but struggles with complex lighting conditions, cluttered urban scenes, and rare manoeuvres that are underrepresented in the limited training set of 1,596 annotated sequences.

The real-time inference pipeline, validated on unseen dashcam sequences, successfully overlays both predicted ADAS metrics and a live scene description onto the video stream at a 0.5 s update rate. The semantic anti-flicker filter based on sentence-embedding cosine similarity (Section 4.3.2) proves effective in stabilising the generated descriptions across consecutive inference steps, suppressing synonymous reformulations while remaining responsive to genuine scene transitions, as illustrated in Figure 4.5.

The training curves in Figure 5.2 confirm that the proportional random interleaving strategy successfully prevents task dominance: both the ADAS and BDD-X validation losses converge steadily without one task degrading the other, and early stopping is triggered at approximately 25 epochs with no sign of overfitting on the ADAS branch.

6.1 Limitations

Several limitations of the present work should be acknowledged, primarily concerning the scale of the dataset used for language generation. Specifically, the BDD-X dataset provides only 1,596 training sequences for the scene description branch. This constitutes a relatively small corpus for fine-tuning a generative

language model, which likely contributes both to the low n-gram overlap scores and to the hallucination failures observed in the qualitative analysis. Consequently, the model's tendency to default to "the light is red" in ambiguous scenes (Figure 5.3, mediocre case) is a typical symptom of a network that has not been exposed to sufficient visual and linguistic diversity during supervision.

6.2 Future Work

Several research directions are identified for future investigation, focusing on improving the system's robustness and facilitating its deployment in real-world automotive hardware. The most immediate priority is to increase the scale and diversity of both training corpora. Incorporating driving logs collected across multiple geographic regions, seasonal conditions, and illumination levels would significantly improve the robustness and generalisation capabilities of the ADAS regression branch. Concurrently, for the scene description task, augmenting the BDD-X supervision with additional annotated driving explanation datasets, such as NuScenes-QA or Rank2Tell, would provide broader coverage of traffic scenarios and a richer causal vocabulary. Expanding the data in this manner would directly address the hallucination failures identified in the qualitative analysis, equipping the model with the necessary diversity to handle complex urban environments and rare manoeuvres more reliably.

Beyond dataset improvements, another critical step toward the practical adoption of this architecture is its transition to embedded edge hardware, specifically targeting Field-Programmable Gate Arrays (FPGAs) or dedicated automotive System-on-Chips (SoCs). While the current pipeline operates effectively in real-time on a consumer-grade GPU, commercial autonomous driving applications demand strict deterministic latency, high energy efficiency, and

functional safety compliance. Porting the VLM and the dual-task inference pipeline to an FPGA would allow for highly parallelised, hardware-level acceleration of the Transformer backbone. Therefore, subsequent research should explore hardware-aware quantization techniques and custom logic designs to execute the network directly at the edge, thereby minimising power consumption and thermal footprint while maintaining the real-time throughput necessary for safety-critical driving operations.

Concluding Remarks

This thesis contributes a concrete demonstration that the rich visual representations encoded by modern VLMs extend beyond semantic scene understanding to capture quantitative physical properties of driving environments. The proposed multi-task architecture, trained entirely on a single consumer-grade GPU through QLoRA, achieves performance competitive with dedicated task-specific models on ego speed estimation, provides semantically coherent natural language descriptions of vehicle actions and their causes, and runs in a real-time loop on unseen dashcam footage. These results support the view that compact, general-purpose VLMs are a viable foundation for unified perception systems in autonomous driving, and motivate further investigation into their capacity to serve as shared encoders across the full breadth of driving intelligence tasks: from low-level metric estimation to high-level behavioural explanation.

References

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Suzuki, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020. 1

[2] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident analysis & prevention*, vol. 35, no. 3, pp. 427–433, 2003. 1, 13

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. 1, 7

[4] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023. 1, 8, 17

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, pp. 8748–8763, PMLR, 2021. 2, 9

[6] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024. 2, 9

[7] L. Wen, X. Yang, D. Fu, X. Wang, P. Cai, X. Li, T. Ma, Y. Li, L. Xu, D. Shang, *et al.*, "On the road with gpt-4v(ision): Early explorations of visual-language model on autonomous driving," *arXiv preprint arXiv:2311.05332*, 2023. 2

[8] Z. Xu, Y. Zhang, E. Xie, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *IEEE Robotics and Automation Letters*, 2024. 2, 22

[9] C. Sima, K. Ruan, W. Wang, Y. Chen, *et al.*, "Drivelm: Driving with graph visual question answering," *arXiv preprint arXiv:2312.14150*, 2023. 2, 22

[10] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with large language models," *arXiv preprint arXiv:2310.01415*, 2023. 2, 17

[11] H. Shao, Y. Wang, G. Letian, H. Li, and Y. Qiao, "Lmdrive: Closed-loop end-to-end driving with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15120–15129, 2024. 2, 22

[12] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997. 3, 14

[13] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017. 3, 15

[14] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 563–578, 2018. 3, 22, 34, 49

[15] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 3, 20, 30, 49

[16] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006. 6

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 7

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

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. 7

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015. 7

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic

urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016. 7

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 7

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. 7

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021. 8

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020. 8

[26] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17863–17873, 2023. 8

[27] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17222–17231, 2022. 8

[28] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal,

L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016. 8

[29] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning (ICML)*, pp. 12888–12900, PMLR, 2022. 9

[30] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning (ICML)*, pp. 19730–19742, PMLR, 2023. 9

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023. 10

[32] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2024. 10

[33] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, *et al.*, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2408.12966*, 2024. 10

[34] Q. Team, "Qwen3-vl technical report," *arXiv preprint*, 2025. 11, 25

[35] X. Song, Y. Wang, D. Li, Y. Chen, H. Lin, *et al.*, "Monocular depth estimation for autonomous driving: A comprehensive review," *arXiv preprint arXiv:2103.12091*, 2021. 12, 50

[36] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001. 14

[37] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International conference on machine learning*, pp. 794–803, PMLR, 2018. 16

[38] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018. 16

[39] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020. 16

[40] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1013–1020, IEEE, 2018. 17

[41] X. Ding *et al.*, "Hilm-d: Towards high-resolution understanding in multimodal large language models for autonomous driving," *arXiv preprint arXiv:2309.05186*, 2023. 17

[42] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021. 19

[43] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52138–52160, 2018. 21

[44] F. Yu, H. Chen, X. Wang, W. Xian, Y. Ying, F. Hou, L. Huang, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2636–2645, 2020. 22