

BOSTON UNIVERSITY
GRADUATE SCHOOL OF ARTS AND SCIENCES

Thesis

**EFFICIENT VISION AND LANGUAGE MODELS FOR
AUTONOMOUS SYSTEMS**

by

SANDESH BHARADWAJ

B.Tech. + M.Tech., IIITDM Kancheepuram, 2020

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

2024

Approved by

First Reader

DocuSigned by:
Eshed Ohn-Bar
96E579E900394CB...

Eshed Ohn-Bar, PhD
Assistant Professor of Electrical & Computer Engineering

Second Reader

DocuSigned by:
Bryan Plummer
4A8FE7742448460...

Bryan A. Plummer, PhD
Assistant Professor of Computer Science

Third Reader

DocuSigned by:
Sabrina Neuman
08BB5FB130F9411...

Sabrina Neuman, PhD
Assistant Professor of Computer Science

Acknowledgments

First, I want to extend my deepest gratitude to my advisor, Prof. Eshed Ohn-Bar, for his constant support, valuable insights, and guidance throughout the process of writing this thesis. His expertise, patience, and encouragement have played a crucial role in shaping this work and my growth as a researcher.

I am also grateful to Prof. Eshed and Boston University for providing the necessary computational resources and platforms that were essential for carrying out the research presented in this thesis. Their assistance has been instrumental in bringing this project to fruition.

I want to thank Jimuyang Zhang, Zhongkai Shangguan, Kathakoli Sengupta, and my colleagues at the Human-to-Everything (H2X) Lab for their contributions, insightful ideas, and stimulating discussions, which have greatly enriched this work. Their collaboration and encouragement have been pivotal in overcoming challenges and expanding the scope of exploration.

Lastly, I want to thank my friends and family for their unwavering love, support, and understanding throughout this remarkable journey. Their belief in me and encouragement have been the cornerstone of my perseverance and accomplishments.

This thesis would not have been possible without the collective support and encouragement from all these individuals and institutions. Thank you for being part of this journey.

Sandesh Bharadwaj

Department of Computer Science

EFFICIENT VISION AND LANGUAGE MODELS FOR AUTONOMOUS SYSTEMS

SANDESH BHARADWAJ

ABSTRACT

The transition of vision-based systems, including autonomous vehicles, from controlled lab environments to real-world deployment poses significant challenges due to constraints such as limited data availability and computational resources. Current approaches often rely on sending data to remote cloud servers for processing, leading to energy inefficiencies. To address these challenges, this thesis proposes novel methodologies for achieving reliable and efficient inference in dynamic scenarios.

Firstly, large-scale vision-based models are evaluated within the CARLA autonomous driving simulator. We introduce a 'switch' policy to offload inference between local and cloud models, trained using reinforcement learning. We evaluate the effectiveness of this policy using a newly introduced evaluation metric: Ecological Navigation Score (ENS). This metric considers route deviations, collisions, and energy consumption, critical factors for assessing driving agent effectiveness.

Secondly, we present a novel dataset to assess the performance of language models (LLMs) and vision language models (VLMs) in autonomous driving tasks. The dataset comprises questions from learner's license examinations, covering various question formats and including both text-based and visual-text pairs. We conduct evaluations on small-scale LLMs using this dataset to analyze their performance, aiming to achieve a passing score of 80% or higher on the driving test.

Through this comprehensive approach, the thesis aims to address the pressing need for robust and efficient vision-based systems in dynamic real-world environments, contributing to advancements in autonomous driving research and technology.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Organization	3
2	Related Work	5
2.1	Imitation Learning	5
2.2	Task Offloading and Energy-Efficient Models	6
2.3	Decision-making in Dynamic Environments	7
2.4	Large Language Models	7
3	Task Offloading through Reinforcement Learning	9
3.1	Problem Formulation	9
3.2	Learning Driving Policies for Local and Cloud Settings	11
3.3	Learning Routing Policy	12
3.4	Experiments	13
3.5	Evaluation Metric	14
3.6	Results	16
4	Passing the Driving Test	19
4.1	Dataset	20
4.2	Large Language Models	23
4.3	Experiments	25
4.3.1	Training Protocol	25
4.3.2	Evaluation Metrics	26

4.3.3	Results	27
5	Conclusions	31
5.1	Contributions	31
	References	33
	Curriculum Vitae	36

List of Tables

3.1	Generalized Ablation We present our comparative analysis between individual imitation-learning models (Local and Cloud only), baseline model (Deep Sequential RL) and our proposed routing model for autonomous driving in CARLA. We also show the impact of our novel metric on determining the energy footprint of driving agents. ENS is Ecological Navigation Score (%), SR is Success Rate (%), RC is Route Completion (%), Infract. is Infraction Rate (/m), Energy is measured in Joules per meter (J/m), and FPS is Frames Per Second.	16
4.1	Baseline evaluations for concise answers (w/o chain-of-thought reasoning). We evaluate baseline models for concise answering (correct answer option and answer only).	27
4.2	Baseline evaluations for chain-of-thought reasoning. We evaluate baseline models for chain-of-thought reasoning on our test set (Explanation of correct answer followed by correct answer option and answer).	27
4.3	Evaluations for fine-tuned concise answering. We evaluate fine-tuned models w/o chain-of-thought reasoning on our test set (correct answer option and answer only).	28

4.4	Evaluations for fine-tuned chain-of-thought reasoning. We evaluate fine-tuned models w/ chain-of-thought reasoning on our test set (Explanation of correct answer followed by correct answer option and answer).	28
4.5	Evaluations for concise answering with context. We evaluate baseline models w/o chain-of-thought reasoning and RAG-based context on our test set (correct answer option and answer only).	29
4.6	Evaluations for chain-of-thought reasoning with context. We evaluate baseline models w/ chain-of-thought reasoning and RAG-based context on our test set (Explanation of correct answer followed by correct answer option and answer).	29
4.7	Evaluations for fine-tuned concise answering with context. We evaluate fine-tuned models w/ RAG-based context on our test set (correct answer option and answer only).	30
4.8	Evaluations for fine-tuned chain-of-thought reasoning with context. We evaluate fine-tuned models w/ chain-of-thought reasoning and RAG-based context on our test set (Explanation of correct answer followed by correct answer option and answer).	30

List of Figures

3.1	Overview of the training process. We train a PPO-based switching policy that takes action input predicted by a pre-trained light model with low-resolution input image input deployable in local devices and decide whether to implement that or query the computation-heavy cloud with image embeddings for more accurate action compromising the extra energy consumption involved with cloud computing.	11
3.2	Reward Progression During Training. We evaluate the rewards for our routing policies throughout the training process against the baseline model. We demonstrate high sample efficiency compared to prior methods, particularly when the routing module is given a history input, i.e., prior actions and their source (cloud or local decision). Despite common instabilities in training reinforcement learning models, Router is shown to achieve significantly higher rewards early in the training process.	17
4.1	Categorization based on images and text-only. We provide 2 main categories of questions based on images, TextQA and VisualQA	20
4.2	Categorization based on Question Type. We provide 3 main categories of questions based on type.	21
4.3	Categorization based on Difficulty. We provide 3 main categories of questions based on difficulty, Easy , Medium , and Hard	22

4.4	Categorization of Driving Manuals based on file formats. We provide driving manuals in both PDF and text formats.	23
-----	---	----

List of Abbreviations

BC	Behavioral Cloning
BLEU	BiLingual Evaluation Understudy
ENS	Ecological Navigation Score
IL	Imitation Learning
LLM	Large Language Model
MLP	Multi-Layer Perceptron
PPO	Proximal Policy Optimization
QA	Question-Answer
RAG	Retrieval-Augmented Generation
RL	Reinforcement Learning
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SOTA	State-of-the-art
VLM	Vision Language Model

Chapter 1

Introduction

1.1 Motivation

We are currently undergoing a transformative societal phase as vision-based systems, such as mobile robots, personalized devices, and autonomous vehicles, transition from their controlled lab environments into the real world. However, various constraints impede their performance in deployed environments. For instance, limited data availability leads to accuracy constraints, while computational and energy limitations restrict the capabilities of high-cost systems. Presently, these systems may only support lightweight neural network models with restricted runtime and accuracy, often depleting onboard batteries quickly. As sensor resolutions and complexity increase alongside model sizes, existing efficiency bottlenecks may only worsen over time. Moreover, larger models don't necessarily guarantee improved accuracy, as they demand more data and experience diminishing returns in accuracy improvement. Consequently, there's a growing imperative to devise better models and methodologies for learning from available sensor data.

First, to address the need for reliable, accurate inference, the currently prevailing approach for processing with powerful models involves sending data e.g., an image, to remote cloud servers for computational offloading, ensuring reliable and accurate inference. While effective in providing high-quality predictions, the transmission and cloud processing overheads incur substantial energy costs. Therefore, it is unsuitable for dynamic real-world agents like autonomous robots, which must continuously an-

ticipate and respond to dynamic settings. The energy consumption associated with offloading decision-making to the cloud is a major factor in building efficient autonomous systems. Thus, we seek to address a fundamental research question: How to achieve robust vision-based systems that can be flexibly optimized for both safety and efficiency in dynamic scenarios?

Large language models (LLMs) and vision language models (VLMs) have recently emerged as leading candidates for adapting to out-of-domain data across various disciplines, including vision and robotics. With their conversation, perception, and action capabilities, they demonstrate fundamental qualities suitable for providing sophisticated language feedback to supervise driving agents. Additionally, their ability to generalize and excel in domains beyond their training data, facilitated by techniques like fine-tuning, prompt engineering, and retrieval-augmented generation, suggests promising prospects for their integration with autonomous systems. However, due to their susceptibility to hallucinate and demonstrate bias, it is important to rigorously evaluate these language models and ensure that they possess the necessary common sense required to understand and provide context equivalent to, if not surpassing, human drivers. This leads to our second research question: How do we effectively assess the efficacy of LMs in the context of autonomous driving?

This thesis aims to address these goals through the following approach; First, we assess large-scale vision-based models meant for deployment on the cloud within the CARLA autonomous driving simulator. We also evaluate a control policy ('switch') that offloads inference to local or cloud devices. Additionally, we introduce a new evaluation metric, namely the **E**cological **N**avigation **S**core (*ENS*). This metric takes into account the consequences of route deviations and energy consumption on the overall effectiveness of a driving agent. Second, we present a novel dataset designed to assess the performance of LLMs and VLMs within the realm of autonomous driving. This

dataset comprises questions typically encountered in learner’s license examinations in the United States. It encompasses a variety of question formats, primarily multiple-choice questions with a single correct answer, ranging from straightforward true-false assertions to more complex reasoning-based queries with four choices. Additionally, the dataset includes both text-based question-answer pairs and visual-text question-answer pairs. Third, we conduct evaluations on small-scale LLMs using this dataset and analyze their performance. Our objective is to determine whether these models can achieve a passing score of 80% or higher on the driving test.

1.2 Organization

The organization of this document is as follows.

Chapter 2 serves as the foundation of our research, offering an exploration of related work. It provides essential background information, discussing three primary topics: imitation learning, task offloading and energy-efficient models, decision-making in dynamic environments, and LLMs. This chapter plays a pivotal role in shaping our approach by presenting an overview of the current state-of-the-art.

Chapter 3 introduces task offloading for autonomous driving and our proposed evaluation metric, called Ecological Navigation Score, tailored specifically for CARLA. We discuss the metrics currently used in CARLA and their limitations, laying the foundation for our proposed metrics. Then, we explain the formulation of the metric, laying out the groundwork behind the idea. To show the effectiveness of this metric, we evaluate imitation learning-based models and our ‘switch’ for autonomous driving.

In Chapter 4, we delve into the capabilities of LLMs in the context of autonomous driving. We introduce a novel dataset featuring questions akin to those found in learner’s license examinations. We provide diverse question types to mimic the varying difficulty levels encountered in such tests. By evaluating multiple small-scale

LLMs on this dataset using various strategies, we showcase the efficacy of our evaluation methodology as a crucial preprocessing step for integrating language into vision-based autonomous driving approaches.

The final Chapter, Chapter 5, serves as a culmination of our research, summarizing our contributions and findings. We reiterate the importance of our work and its potential impact on the field of autonomous driving. Additionally, we present future directions and avenues for continued progress, providing a roadmap for the advancement of this field.

This structured approach ensures a thorough exploration of our research, from its foundational concepts to practical implementation and analysis, ultimately contributing to the body of knowledge in the domain of autonomous systems.

Chapter 2

Related Work

2.1 Imitation Learning

Imitation learning has emerged as a pivotal paradigm in the pursuit of developing intelligent, adaptable driving agents. It offers a mechanism for agents to acquire driving skills by observing and mimicking the behavior of an expert driver. Each expert trajectory contains a sequence of states and actions, and all "state-action" pairs are extracted to construct datasets. In IL, the model leverages the constructed dataset to learn the latent relationship between the state (feature) and action (labels). Therefore, the specific objective of IL is to appraise the best mapping between state and action, so that the agent achieves the expert trajectories as much as possible. Imitation learning in autonomous driving is primarily done through behavior cloning (Torabi et al., 2018).

Recent advances in IL for autonomous driving trace back to ALVINN (Pomerleau, 1988), a pioneering neural network trained to imitate the driving behavior of an ego vehicle. Since then, more elaborate IL-based approaches for autonomous driving have surfaced (Chen et al., 2015; Chen and Krähenbühl, 2022). Despite these advancements, large-scale deployment of these IL-based approaches in real-world settings remains elusive, best by several learning-related challenges, including shortcut learning and overfitting on spurious correlations (Jaeger et al., 2023; De Haan et al., 2019).

2.2 Task Offloading and Energy-Efficient Models

The integration of cloud computing and edge devices combines the advantages of both, which has attracted increasing attention in recent years. Leveraging the cloud’s capacity to employ advanced hardware and deploy large models allows quick execution of computationally expensive tasks with accurate results. However, the promptness of these results can be compromised by transmission speeds, while edge devices can enable real-time processing and reduced latency by accessing data close to the source. However, the hardware constraints imposed on edge devices prevent them from using larger models, leading to less accurate results. Existing research (Kag et al., 2022; Ding and Lin, 2020; Penmetcha and Min, 2021) focuses on hybrid approaches for efficient selection of queries from the cloud, but does not take into account the effects of computational delay, making them difficult to adapt for real-time decision-making.

The promise of energy-efficient models has been substantially developed through various research fields, focusing on optimizing computational efficiency and resource utilization, which can reduce carbon emissions and improve energy utilization. One aspect of the field of energy-efficient models emphasizes hardware design (Karras et al., 2020). However, hardware development is often time-consuming and expensive. Another aspect provides insight into the efficacy of model pruning and quantization to minimize model size, thus helping save energy consumption (Yang et al., 2017; Zhu and Rosendo, 2022).

Additionally, the design of energy-efficient architectures tailored specifically for autonomous driving applications has also been a focal point of research. Low-power and low-complexity architectures such as MobileNet and EfficientNet (Koonce, 2021; Sandler et al., 2018; Tan and Le, 2019) have been developed to strike a balance between computational efficiency and accuracy, along with optimization for deployment on edge devices. This makes them suitable for real-time inference in autonomous

driving.

2.3 Decision-making in Dynamic Environments

In recent years, significant progress has been made in enabling robots to learn and adapt to their surroundings. Imitation learning (Hussein et al., 2017) has become a standard approach, proving to be a valuable tool for training models to navigate and make informed decisions in dynamic environments. However, reinforcement learning (RL) (Moerland et al., 2023; Nagabandi et al., 2018) has also been applied to optimize decision-making processes, allowing robots to navigate efficiently while avoiding obstacles and optimizing path planning. The wide-scale deployment of real-time vision-based models on autonomous vehicles is primarily hindered by computational and hardware constraints, where large and accurate models cannot be deployed on local robots. While cloud computing has advanced rapidly and solved the problem of hardware limitations to a certain extent, the carbon emissions generated due to reliance on large data centers cannot be ignored (Sudhakar et al., 2022).

2.4 Large Language Models

In today’s technological landscape, LLMs are drawing attention with their superior contextual understanding and in-context learning capabilities (Achiam et al., 2023; Touvron et al., 2023). Their enriched common sense knowledge has facilitated significant advancements in many downstream tasks, and could provide a transparent application of the autonomous driving decision-making process, significantly enhancing system reliability and performance.

While numerous methodologies have been proposed in recent literature for advancing autonomous driving, achieving SOTA performance on established benchmarks (Caesar et al., 2020; Geiger et al., 2012), they predominantly focus on utilizing

large LLMs ($> 7\text{B}$ params.). Deploying such models requires substantial computational resources, which is not feasible for real-world autonomous driving applications. Furthermore, these LLMs are fine-tuned on extensive driving datasets to bridge the domain gap, entailing significant computational and resource overhead.

Recently, tiny LLMs (Chung et al., 2022) have shown potential in terms of providing decent performance with emergent abilities achieved at a significantly smaller scale compared to their large-scale LLM counterparts. With fewer parameters, tiny LLMs offer significant computational efficiency in terms of fast pre-training and inference with reduced memory and storage requirements.

Chapter 3

Task Offloading through Reinforcement Learning

Our objective is to learn a policy that can determine the dynamic allocation of local and cloud resources while optimizing energy efficiency and real-time performance. In this chapter, we discuss the main components of our approach. First, we formulate the autonomous driving task in the context of reaching a specific goal (Sec. 3.1). Second, we employ imitation learning to train two driving policies tailored for local and cloud settings respectively (Sec. 3.2). Third, we discuss how we train our *routing policy* via Proximal Policy Optimization (PPO) (Schulman et al., 2017) (Sec. 3.3). We also introduce our novel metric (ENS) that incorporates energy efficiency and navigation performance. Finally, we discuss the performance of our models and the use of ENS for determining real-world deployment.

3.1 Problem Formulation

We formulate our autonomous driving task as a real-time decision-making problem from a set of observations $o = \{I, p\} \in O$, comprising a front-view camera image $I \in \mathbb{R}^{W \times H \times 3}$, the next waypoint $p \in \mathbb{R}^2$, towards a set of actions $a = \{d, v\} \in A$, where $d \in [-3.14, 3.14]$ represents the orientation and v is the speed.

Our approach consists of three policies: a local and a cloud driving policy π_θ^l and π_ϕ^c with weights $[\theta, \phi]$ that map the observations to actions, producing a_t^l and a_t^c , respectively. Additionally, we introduce a routing policy $\pi_\omega^r(o|\pi_\theta^l, \pi_\phi^c)$ parameter-

Algorithm 1 Routing Policy Training

Input: Image I , next waypoint p , local policy π_θ^l , cloud policy π_ϕ^c
Initialize: Number of iterations T , history \mathcal{H} , routing policy π_ω^r , reply buffer \mathcal{S}
 Collect on policy samples:
for $t = 1$ **to** T **do**
 Obtain local action \mathbf{a}_t^l and embeddings \mathbf{e}_t using local policy $\pi_\theta^l(\mathbf{I}_t, \mathbf{p}_t)$
 Append $(\mathbf{a}_t^l, 0)$ to \mathcal{H}_t and remove the first value
 if $\pi_\omega^r(\mathcal{H}_t, \mathbf{e}_t) = 0$ **then** $\mathbf{a}_t = \mathbf{a}_t^l$
 else
 Send \mathbf{e}_t to cloud, $\mathbf{a}_t = \pi_\phi^c(\mathbf{I}_t, \mathbf{p}_t)$
 Update last value of \mathcal{H}_t to $(\mathbf{a}_t, 1)$
 end if
 Compute instant reward r_t
 if Arrived destination **then** break
 end if
 Update replay buffer $\mathcal{S} = \mathcal{S} \cup \{\mathbf{I}_t, \mathbf{p}_t, \mathcal{H}_t, r_t\}$
 Update routing policy parameters with PPO
end for

ized by ω , to dynamically determine the optimal utilization of local resources versus offloading to the cloud server.

The information transmission between the local and the cloud server potentially introduces delays, resulting in a noisy action that lacks real-time accuracy. Therefore, we introduce a Gaussian noise in the action of our CARLA-based driving environment to simulate real-world conditions, and the final action taken by the driving agent is the original output from the cloud plus the random noise associated with the delay.

As our primary objective with the routing policy is to enhance energy efficiency while preserving task performance, our task will be offloaded to the cloud for processing only when local computation proves inefficient for achieving optimal task performance. Despite the inherent advantages of cloud processing over local computing in terms of speed and accuracy, the communication bottlenecks compel us to prioritize local processes to conserve energy and minimize latency, except when only the cloud meets performance under a specific state e.g., difficult scenarios such as

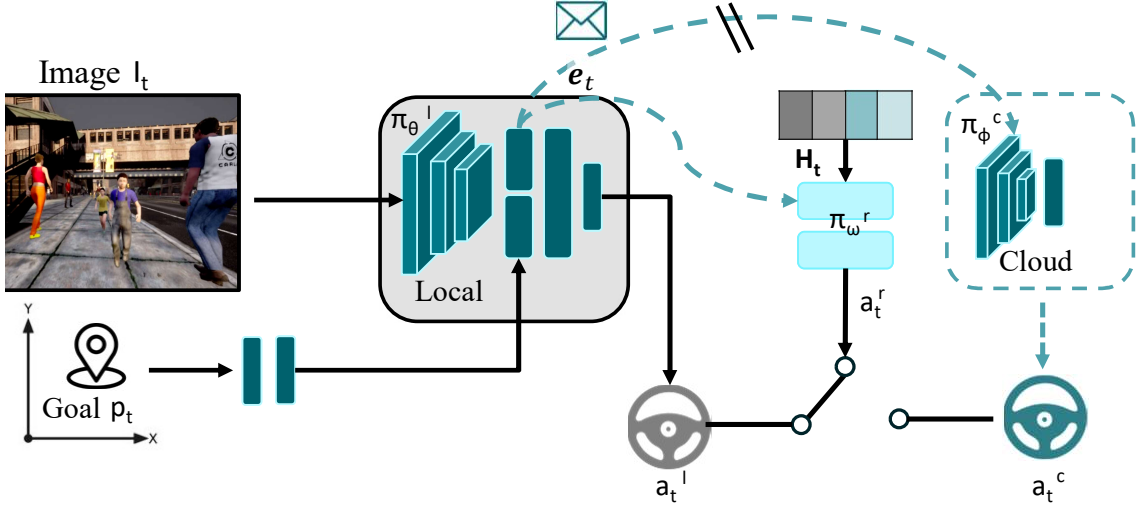


Figure 3.1: Overview of the training process. We train a PPO-based switching policy that takes action input predicted by a pre-trained light model with low-resolution input image input deployable in local devices and decide whether to implement that or query the computation-heavy cloud with image embeddings for more accurate action compromising the extra energy consumption involved with cloud computing.

driving in bad weather, low lighting conditions, and crowded scenarios.

3.2 Learning Driving Policies for Local and Cloud Settings

We follow the standard imitation learning approach to train our driving policies in an offline manner. Specifically, we first collect a dataset $\mathcal{D} = \{\mathbf{I}_i, \mathbf{p}_i, \mathbf{a}_i\}_{i=1}^N$ on diverse and complex routes and weathers using CARLA (Dosovitskiy et al., 2017) to simulate real-world scenarios to provide the basis for our driving policies.

As shown in Fig. 3.1, both the local and cloud driving policies comprise (i) a visual semantics feature extractor module for obtaining embeddings from input images, (ii) multilayer perceptron to extract the feature associated with the robot’s imminent goal point, and (iii) a goal-conditional module that takes concatenation of image embeddings and goal features to predict robot actions, encompassing both direction

d and speed v . To enable the sharing of common features, we first train a robust cloud policy with a large neural network (Schneider et al., 2017). Subsequently, we freeze the parameters of the first few layers of the pre-trained cloud policy to serve as a shared feature extractor for the local neural network (Koonce, 2021; Sandler et al., 2018). Additional fully connected layers are then added to the extracted features to form the local policy. Therefore, the local policy is significantly smaller compared to the cloud policy. Only the parameters of the fully connected layer in the local policy are optimized, leading to reduced computational consumption and improved efficiency in both training and inference time.

The learning objective for both local and cloud policies is achieved by minimizing the \mathcal{L}_1 distance:

$$\text{minimize } \mathbb{E}_{(\mathbf{I}, \mathbf{p}, \mathbf{a}) \sim D} [\mathcal{L}_1(\mathbf{a}, \pi(\mathbf{I}, \mathbf{p}))] \quad (3.1)$$

where π represents driving policy $\{\pi_{\theta}^l, \pi_{\phi}^c\}$.

3.3 Learning Routing Policy

To effectively balance cloud and local computing resources while achieving good performance at the same time, we propose a *routing policy* that seamlessly switches between local and cloud. We follow the standard PPO training process as shown in Algorithm 1 and the formulation of our routing policy is introduced below:

State Space: We denoted the current state as $\mathbf{s}_t = \{\mathbf{e}_t, \mathcal{H}_t\}$, where \mathbf{e}_t represents the image embeddings and goal features extracted from the shared feature extractor employed by both local and cloud policies, as discussed in Sec. 3.2, and $\mathcal{H}_t = \{(\mathbf{a}_i, \mathbf{1}_i)\}_{t-k}^t$ is a sequence of the last k history actions, where \mathbf{a}_i represents previous actions and $\mathbf{1}_i$ is an indicator function indicating whether the action is obtained locally or from the cloud.

Action Space: The action produced by our router policy is a binary discrete value, indicating whether to accept the local policy π_{θ}^l or transmit the embeddings \mathbf{e}_t to the cloud policy π_{ϕ}^c .

Reward Function: We design our reward function encompassing five key components: geodesic reward r_{geo} , speed reward r_{speed} , energy disadvantage bonus r_{energy} , extreme action clip r_{action} , and collision penalty $r_{collision}$. We use the current standard approaches that linearly combine the rewards (Zhuang et al., 2023; Booth et al., 2023; Knox et al., 2023) and shape the reward function as:

$$r_{standard} = \alpha_g r_{geo} + \alpha_s r_{speed} + \alpha_e r_{energy} + \alpha_c r_{collision} \quad (3.2)$$

where $\alpha_g, \alpha_s, \alpha_e, \alpha_c$, are tuned scalar hyperparameters (we perform careful tuning of these with a grid search).

3.4 Experiments

Data Collection and Training: To learn the local and cloud driving policies through imitation learning, we collect driving data from our CARLA environment. We spawn 100 vehicles around our ego vehicle’s route and employ a heuristic policy for data collection: we first design several fixed paths with waypoints, and the driving agent follows the waypoints from the start position to the destination, halting when other vehicles block its path, allowing them to pass, and proceeds along the route if the path is clear. The routes are created with distinct start and end points across CARLA’s Town 10 map, ensuring diverse data collection.

Local and Cloud Policies: As discussed in Sec. 3.2, our local and cloud policies utilize a common feature extractor, which processes a 480×480 image and a 2D imminent position. Specifically, we employ the first few layers of RegNet (Schneider

et al., 2017) to extract the image feature, and two fully connected (FC) layers to capture the position features. For the local policy, the image feature is flattened and concatenated with the position feature. Subsequently, another FC layer is appended to produce local actions. In the case of the cloud policy, both the image feature and position feature are transmitted from local to the cloud. The image feature is fed to the remaining layers of RegNet (Schneider et al., 2017), followed by concatenation with the position feature. An MLP is built on top of the combined features to predict the cloud action.

Routing Policy: The proximal policy optimization (PPO) (Schulman et al., 2017) based router policy is employed in conjunction with a multi-layer perceptron (MLP) policy network. The policy network consisted of two hidden layers, each having 16 units and the value network is structured with two hidden layers with 256 units each.

Training Protocol: We use AdamW (Loshchilov and Hutter, 2017) optimizer and train 200 epochs at a learning rate of 0.0001 with our driving dataset for both local and cloud policies. We train the routing policy for 1,000 episodes, with each episode comprising at most 1,500 steps. The discount factor γ is set to 0.99. Episodes are subject to truncation if the agent encounters a collision with other objects, completes the designated number of steps, or reaches the predefined destination. Additionally, episodes are terminated if the agent deviates beyond a threshold of 30 meters away from the pre-defined route.

3.5 Evaluation Metric

Ecological Navigation Score: As our primary task is to optimize energy consumption while ensuring driving performance, we introduce *Ecological Navigation Score*(*ENS*) to balance between driving performance and ecological considerations.

We define the ENS as

$$\text{ENS} = P_E \cdot RC \cdot P_I^{IC} \cdot P_{RD} \quad (3.3)$$

where RC is route completion, P_I represents the infraction penalty for collisions, IC represents the number of robot collisions per meter, P_{RD} represents the penalty for route deviation, and P_E is the penalty term for energy consumption,

$$P_E = 1 - \frac{\text{Energy}}{N_E} \quad (3.4)$$

where N_E is a normalization factor.

Following the CARLA leaderboard settings, P_I is set to 0.5, and P_{RD} is defined as

$$P_{RD} = \begin{cases} 0.8, & \text{if } RD > \epsilon_{RD} \\ 1.0, & \text{otherwise} \end{cases} \quad (3.5)$$

where $\epsilon_{RD} = 1.5\text{m}$ is the route deviation threshold.

Run-Time: In autonomous driving, real-time decision-making is crucial. Driving agents need to be able to respond instantly to environmental changes and quickly adjust path planning to adapt to new situations. To ensure a real-time response, we also report run-time Frames Per Second (FPS), encompassing both model processing time and communication time between a local device and the cloud server.

Table 3.1: Generalized Ablation We present our comparative analysis between individual imitation-learning models (Local and Cloud only), baseline model (Deep Sequential RL) and our proposed routing model for autonomous driving in CARLA. We also show the impact of our novel metric on determining the energy footprint of driving agents. ENS is Ecological Navigation Score (%), SR is Success Rate (%), RC is Route Completion (%), Infract. is Infraction Rate (/m), Energy is measured in Joules per meter (J/m), and FPS is Frames Per Second.

Method	ENS \uparrow	SR \uparrow	RC \uparrow	Infract. \downarrow	Energy \downarrow	FPS \uparrow
Local-only	53.48	0.00	59.66	0.024	3.47	65.40
Cloud-only	0.00	76.66	83.24	0.009	25.00	7.10
Deep Sequential RL 2019	46.21	48.46	37.68	0.031	9.16	57.53
Router	51.04	54.21	67.14	0.022	7.71	42.31
Router w/ History	59.72	60.00	72.78	0.02	5.25	35.06

3.6 Results

We use the CARLA simulator (version 0.9.13) (Dosovitskiy et al., 2017) for evaluation. All algorithms are tested over five different routes for 30 episodes in CARLA’s Town 10 map under diverse weather conditions (including hard rain, sunny, wet, and sunset weather) to simulate real-world scenarios. We analyze our Router’s performance compared to our imitation-learning models and a baseline reinforcement learning-based approach (Wang et al., 2019) and show the impact of the metric for evaluating the deployability of such approaches for real-world driving.

As illustrated in Tab. 3.1, the ENS metric drops to 0.00% when assessing high energy-consuming cloud model exclusively, suggesting that conducting all computations in the cloud is ecologically inappropriate. We observe that our router with history buffer performs well, achieving an ENS score of 59.72%, and a route completion score of 72.78%. The routing policy also achieves the lowest infraction rate of 0.02/km, while maintaining a minimal energy consumption.

We also observe that the ENS does not significantly improve between the routing policy and the local-only models, with only 6 percentage points of improvement. The local policy is significantly smaller compared to the cloud policy, which allows it to

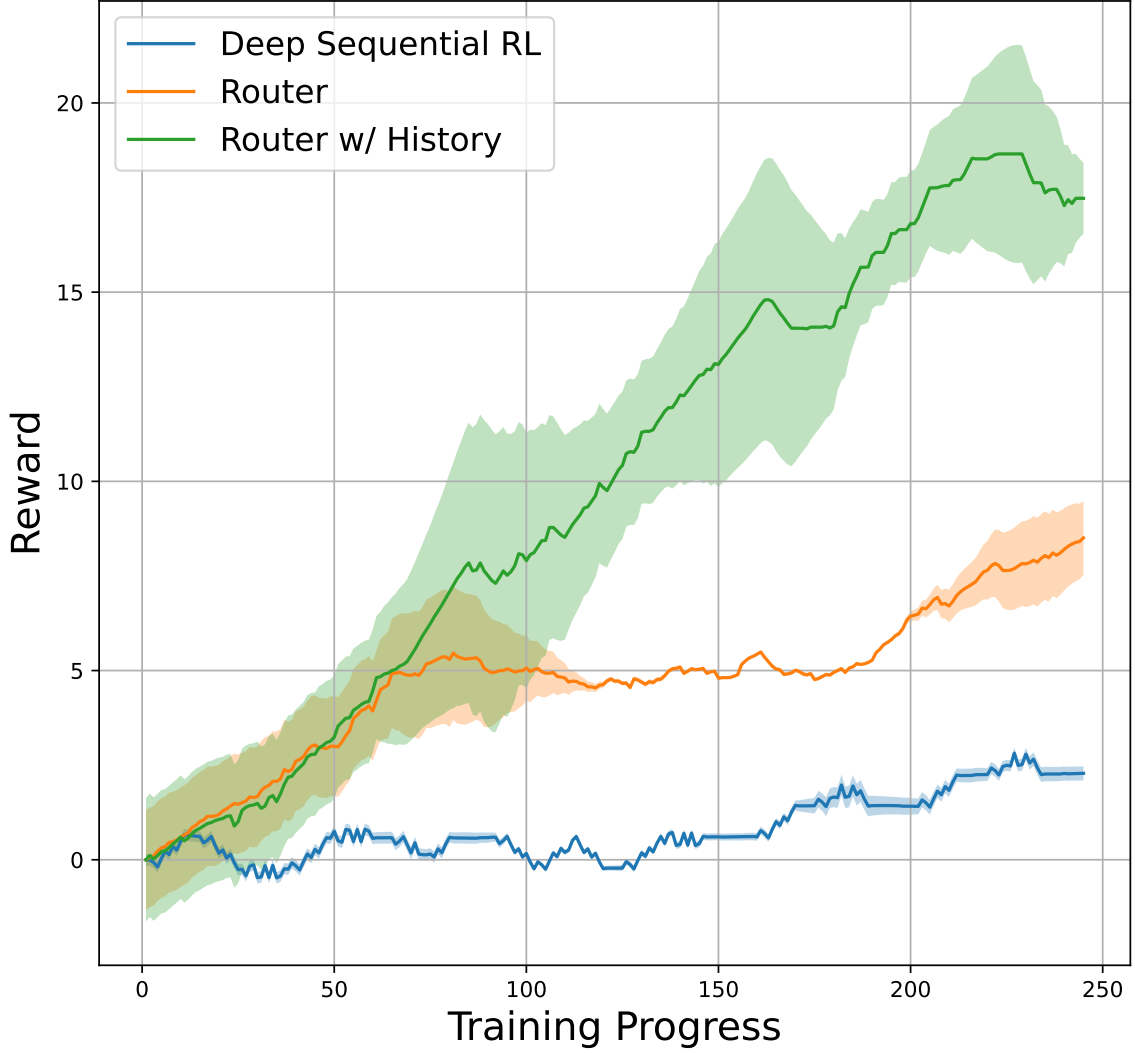


Figure 3.2: Reward Progression During Training. We evaluate the rewards for our routing policies throughout the training process against the baseline model. We demonstrate high sample efficiency compared to prior methods, particularly when the routing module is given a history input, i.e., prior actions and their source (cloud or local decision). Despite common instabilities in training reinforcement learning models, Router is shown to achieve significantly higher rewards early in the training process.

achieve significant run-time performance and lowest energy consumption. Improving the local policy would lead to a significant reduction in energy consumption by reducing the need to query the cloud for inference. We use this as motivation to explore

the impact of language in autonomous driving, which will be discussed in Chapter 4.

Chapter 4

Passing the Driving Test

As demonstrated in Chapter 3, while task-offloading through reinforcement learning is a potential solution for efficient real-world deployment of driving agents, the driving agents must be improved to ensure minimal energy consumption without compromising on accuracy. While current research integrates language models with vision-based architectures, these approaches do not efficiently showcase the level of understanding that such language models have about common driving rules that humans generally understand and follow. Before integrating language with current SOTA driving agents, it is important to understand the generative capabilities of LLMs, particularly their propensity for hallucinations and bias. Therefore, it is important to determine the efficacy of incorporating language into vision-based driving agents.

In this chapter, we present the main components of our approach. First, we present a novel dataset comprising single-response questions and answers sourced from the internet, commonly encountered in learner’s license examinations. The dataset comprises both visual and text-only QA pairs, with a large variety of questions available. We also provide driving handbooks for additional contextual information. Second, we evaluate several tiny LLMs and benchmark them on our dataset using different strategies. We discuss the performance of these models and their potential for analyzing driving scenarios.


TextQA	VisualQA
<p>Question: The likelihood of an accident increases if a driver is under the influence of</p> <p>Options:</p> <ul style="list-style-type: none"> A. a cup of coffee. B. softly playing music. C. a cup of tea. D. alcohol, illegal drugs, or prescription medications. <p>Correct Option: D</p> <p>Explanation: Alcohol, some illegal drugs, some prescription medications, and some over-the-counter medications can cause drowsiness, sedation, or impaired coordination, which may impair your ability to drive. If you are unsure of the effects of a particular medication, check the warning label or ask your pharmacist or doctor.</p>	<p>Question: You drive up to an intersection and you see this sign. What should you do??</p> <p>Image:</p>  <p>Options:</p> <ul style="list-style-type: none"> A. Come to a complete stop and then proceed. B. Slow down and only proceed if the intersection is clear. C. Come to a complete stop and yield to traffic before proceeding. D. Find another route; you cannot proceed through here. <p>Correct Option: C</p> <p>Explanation: At a stop sign, you must stop before the stop line, crosswalk, or intersection, whichever you arrive at first. Then yield to pedestrians and other vehicles, and proceed only when the intersection is clear. [Regulatory Signs, Traffic Signs, Section 4: Traffic Controls, Florida Driver License Handbook]</p>

Figure 4-1: Categorization based on images and text-only. We provide 2 main categories of questions based on images, **TextQA** and **VisualQA**.

4.1 Dataset

To provide a good benchmark for determining the efficacy of LLMs in driving scenarios, we present a novel dataset for driving benchmarking. This dataset consists of 38,452 QA pairs, encompassing two distinct types of questions: TextQA, which consists solely of text-based question-answer pairs, and VisualQA, featuring image-question-answer triplets. Questions are split in a 11:9 ratio between these two subsets. Within the TextQA subset, questions are further categorized into True-False and Multiple-Choice pairs. We present questions and answer choices separately to enable flexibility in prompt usage across different LLMs.

Complete the Sentence	Find the Correct/Incorrect Option	Fill in the Blank
<p>Question: The likelihood of an accident increases if a driver is under the influence of</p> <p>Options:</p> <p>A. a cup of coffee. B. softly playing music. C. a cup of tea. D. alcohol, illegal drugs, or prescription medications.</p> <p>Correct Option: D</p> <p>Explanation: Alcohol, some illegal drugs, some prescription medications, and some over-the-counter medications can cause drowsiness, sedation, or impaired coordination, which may impair your ability to drive. If you are unsure of the effects of a particular medication, check the warning label or ask your pharmacist or doctor.</p>	<p>Question: Which of the following is considered to be a safe driving practice?</p> <p>Options:</p> <p>A. Aggressive driving B. Tailgating C. Distracted driving D. Defensive driving</p> <p>Correct Option: D</p> <p>Explanation: The National Safety Council defines defensive driving as "driving to save lives, time, and money, in spite of the conditions around you and the actions of others." Defensive driving is about anticipating potentially dangerous situations in advance, including driving conditions and mistakes made by others, and planning how to deal with those situations. You owe it to yourself to become a defensive driver.</p>	<p>Question: The Alaska DMV has the authority to _____ a driver's license regardless of any court's verdict.</p> <p>Options:</p> <p>A. suspend B. revoke C. revoke, suspend, or cancel D. cancel</p> <p>Correct Option: C</p> <p>Explanation: The Alaska DMV has the authority to revoke, suspend, or cancel your driving privileges. These can be administrative penalties rather than criminal penalties. That is, the DMV can take such actions even before a court reaches a verdict on the charges against you. Even if you are acquitted of the charges in court, you will still have to pay a reinstatement fee to the DMV to have your driving privileges reinstated. [Suspensions and Revocations, State of Alaska Driver Manual]</p>

Figure 4-2: Categorization based on Question Type. We provide 3 main categories of questions based on type.

Questions can be categorized into 3 distinct categories based on their style:

1. Fill in the Blank.
2. Find the Correct/Incorrect Option.
3. Complete the Sentence.

Additionally, questions can be categorized based on their level of difficulty as Easy, Medium, or Hard.

Easy	Medium	Hard
<p>Question: The likelihood of an accident increases if a driver is under the influence of</p> <p>Options:</p> <p>A. a cup of coffee. B. softly playing music. C. a cup of tea. D. alcohol, illegal drugs, or prescription medications.</p> <p>Correct Option: D</p> <p>Explanation: Alcohol, some illegal drugs, some prescription medications, and some over-the-counter medications can cause drowsiness, sedation, or impaired coordination, which may impair your ability to drive. If you are unsure of the effects of a particular medication, check the warning label or ask your pharmacist or doctor.</p>	<p>Question: You are driving on a divided highway. You see a school bus stopped with flashing red lights on the opposite side of the highway. Do you need to stop here?</p> <p>Options:</p> <p>A. No. B. No, but you must slow to 25 mph or less. C. Yes, you must stop until the bus has switched off its red signals, retracted its stop arm, and started moving. D. Yes, you must stop and yield to any pedestrians on the road before you proceed.</p> <p>Correct Option: A</p> <p>Explanation: You must stop for a stopped school bus whose red lights are flashing, no matter which direction it is traveling in. Children are probably unfamiliar with the rules of the road, so they may do something unexpected.</p>	<p>Question: You are driving and a situation in the back seat requires your attention. You should</p> <p>Options:</p> <p>A. pull over to the side of the road and park your vehicle before dealing with the situation. B. move the rear-view mirror so you can see the back seat. C. slow down and deal with the situation while you continue to drive. D. turn around and deal with the situation while glancing ahead from time to time.</p> <p>Correct Option: A</p> <p>Explanation: Do not take your eyes off the road to turn around to deal with the needs of passengers, children or pets. If you must give attention to passengers or animals, pull over to the side of the road and park your vehicle.</p>

Figure 4.3: Categorization based on Difficulty. We provide 3 main categories of questions based on difficulty, **Easy**, **Medium**, and **Hard**.

We also provide supplementary explanations for the correct answer choice for all multiple-choice question-answer pairs and image-question-answer triplets. Supplementary explanations for True-False pairs are not provided since they are self-explanatory. Additionally, we provide driver’s manuals from all 50 states and the District of Columbia in the United States. The handbooks are provided in both document (PDF) and parsed text formats, allowing for use as required.

Driving Manual (PDF)	Driving Manual (Text)
<p>WHO MUST HAVE AN ALASKA DRIVER'S LICENSE? Every person who operates a motor vehicle on Alaska streets, highways, or other public property must have a valid Alaska driver's license or permit. The few exceptions are listed below.</p> <p>WHO IS EXEMPT?</p> <ol style="list-style-type: none"> 1. A non-resident who is at least 16 years of age and has in their possession a valid driver's license issued by another state or country. However, an Alaska driver's license must be obtained by the end of a 90-day period after entry into the state. 2. A member of the armed forces of the United States, and their spouse who is over the age of 18, who has a valid driver's license issued by another state, and who maintains permanent residence in that state. A member's dependents are not exempt. 3. A person when driving farm equipment that is only temporarily driven or moved on a highway. 4. An employee of the United States Government while operating a United States Government vehicle on official business. 5. A commercial driver who is domiciled in another state. <p>LICENSES AND PERMITS Alaska has seven classes of driver's licenses and two types of permits. Classes A, B, and C are licenses used for operating commercial motor vehicles. A separate manual is published for persons interested in obtaining a commercial driver's license. Class D is the license used for operating passenger vehicles. Motorcycles and motor scooters with engine displacements of less than 50cc can also be operated with a class D license. To take the test for your Class D license online, please visit ak.knowtodrive.com.</p>	<p>WHO MUST HAVE AN ALASKA DRIVER'S LICENSE? Every person who operates a motor vehicle on Alaska streets, highways, or other public property must have a valid Alaska driver's license or permit. The few exceptions are listed below. WHO IS EXEMPT? 1. A non-resident who is at least 16 years of age and has in their possession a valid driver's license issued by another state or country. However, an Alaska driver's license must be obtained by the end of a 90-day period after entry into the state. 2. A member of the armed forces of the United States, and their spouse who is over the age of 18, who has a valid driver's license issued by another state, and who maintains permanent residence in that state. A member's dependents are not exempt. 3. A person when driving farm equipment that is only temporarily driven or moved on a highway. 4. An employee of the United States Government while operating a United States Government vehicle on official business. 5. A commercial driver who is domiciled in another state. LICENSES AND PERMITS Alaska has seven classes of driver's licenses and two types of permits. Classes A, B, and C are licenses used for operating commercial motor vehicles. A separate manual is published for persons interested in obtaining a commercial driver's license. Class D is the license used for operating passenger vehicles. Motorcycles and motor scooters with engine displacements of less than 50cc can also be operated with a class D license. To take the test for your Class D license online, please visit ak.knowtodrive.com.</p>

Figure 4-4: Categorization of Driving Manuals based on file formats. We provide driving manuals in both PDF and text formats.

We create two test sets for TextQA and VisualQA, consisting of 200 questions each, sampled across all difficulty levels. This allows us to comprehensively test LLMs and VLMs and benchmark their potential for driving. Qualitative examples are provided in Figs. 4-1, 4-2, 4-3, and 4-4.

4.2 Large Language Models

To evaluate the efficacy of LLMs, we propose benchmarking their performance on the novel dataset proposed in Sec. 4.1. To improve the performance of driving agents deployed on the edge, we only consider tiny LLMs ($\leq 7B$ params.) for our experiments. Where available, we use the chat (instruction-finetuned) versions of the models as they are better suited for following question-answer prompts than the base versions.

We use the following models:

1. TinyLlama (Zhang et al., 2024)
 - (a) TinyLlama-1.1B-Chat-v1.0 (1.1B params.)
2. Gemma (Team et al., 2024)
 - (a) gemma-2b-it (2B params.)
 - (b) gemma-7b-it (7B params.)
3. Phi (Li et al., 2023)
 - (a) phi-1.5 (1.5B params.)
 - (b) phi-2 (2B params.)

We set a minimum threshold score required to 'pass' the driving test i.e. 80%, which is the standard score required for human drivers to pass the learner's license test in the US. As we only consider LLMs for our experiments, we use the TextQA subset of the dataset to finetune these models and consequently use the TextQA test set for evaluations. We run the following set of experiments on the selected models:

1. Baseline w/ and w/o Chain-Of-Thought Reasoning
2. Fine-tuning on TextQA w/ and w/o Chain-Of-Thought Reasoning
3. Use of Retrieval-Augmented Generation w/ and w/o Chain-Of-Thought Reasoning

Baseline w/ and w/o Chain-Of-Thought Reasoning: We use the models for evaluations, without fine-tuning on our dataset. We prompt the models to only provide us with the correct answer option and answer, without further text generation or explanation e.g., "A. Driving up the hill."

Additionally, we run experiments prompting the models to provide step-by-step reasoning (chain-of-thought reasoning) followed by the correct answer e.g., "You must stop before the stop line or intersection, yield to pedestrians and other vehicles, and proceed only when the intersection is clear. Therefore, the correct answer is C. Come to a complete stop and yield to traffic before proceeding."

Finetuning on TextQA w/ and w/o Chain-Of-Thought Reasoning: We finetune the models on the TextQA subset and evaluate on the TextQA test set. As in the baseline evaluations, we prompt the models w/ and w/o chain-of-thought reasoning and report the performance.

Testing with Retrieval-Augmented Generation: We provide contextual information from the driving handbooks as additional context for each question to the LLM and prompt the models for an answer. We run experiments for prompting w/ and w/o chain-of-thought reasoning, using both baseline and fine-tuned models to understand the implications of context on the answers. We also run experiments to use multiple matches from driving handbooks to evaluate the effectiveness of large context blocks on answering.

4.3 Experiments

4.3.1 Training Protocol

The TextQA subset contains 17,456 questions, all of which are used for finetuning. For each LLM, we follow the prompt template used during training for our evaluations, adding conditions for providing correct answer options with and without explanations to ensure strict adherence to the requirements. Additionally, we implement QLoRA finetuning (Detrmers et al., 2024) and fine-tune each model for 3 epochs.

4.3.2 Evaluation Metrics

For our question-answering task, we use statistical scoring metrics such as BLEU (BiLingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation). We additionally consider the Phrase Match metric, which is a useful metric for multiple-choice question-answering since it provides a positive match if the reference answer is found in the predicted text.

BLEU is a metric for comparing a candidate translation to one or more reference translations. However, it can be used to evaluate text generated for other tasks. A score of 0.08-0.1 is usually considered to be a good score, however, the underlying nature of the metric means that a score of 1 can never be achieved. It is an easy metric to compute and has a high correlation with human judgment of prediction quality, making it useful for question-answering.

ROUGE is a set of metrics used for evaluating automatic summarization and machine translation. ROUGE is case insensitive and computes several types of metrics over different n-gram scores. We use the following rouge types for our task:

1. ROUGE-1: unigram (1-gram) based scoring
2. ROUGE-2: bigram (2-gram) based scoring
3. ROUGE-L: Longest common subsequence-based scoring
4. ROUGE-L-SUM: Splits texts using "\n" and calculates average ROUGE-L for all sentences in text.

ROUGE-1 scores between 0.4 and 0.5 are considered good. For ROUGE-2: scores of 0.2 to 0.4 are considered good. ROUGE-L and ROUGE-L-SUM scores are good around 0.25-0.4 and low below 0.2.

Phrase Match is an absolute metric that matches predictions to references and outputs a score based on the number of matching predictions found. Since it directly

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.050	0.138	0.058	0.126	0.127	0.105
gemma-2b-it	0.149	0.289	0.203	0.282	0.281	0.250
gemma-7b-it	0.085	0.132	0.076	0.126	0.128	0.115
phi-1.5	0.125	0.226	0.156	0.217	0.220	0.330
phi-2	0.186	0.285	0.218	0.284	0.282	0.260

Table 4.1: Baseline evaluations for concise answers (w/o chain-of-thought reasoning). We evaluate baseline models for concise answering (correct answer option and answer only).

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.060	0.152	0.076	0.141	0.142	0.135
gemma-2b-it	0.092	0.177	0.135	0.198	0.199	0.250
gemma-7b-it	0.045	0.111	0.05	0.101	0.102	0.085
phi-1.5	0.133	0.24	0.17	0.233	0.235	0.350
phi-2	0.15	0.263	0.2	0.256	0.26	0.445

Table 4.2: Baseline evaluations for chain-of-thought reasoning. We evaluate baseline models for chain-of-thought reasoning on our test set (Explanation of correct answer followed by correct answer option and answer).

relates to the correctness of the answer, it is a useful metric for evaluating the correctness of fixed answers. In the context of learner’s license examinations, a score of 0.8 or higher is good.

4.3.3 Results

Baseline Evaluations: In our initial experiments, we observe that the baseline models exhibit subpar performance across all metrics in providing concise answers, despite using the instruction fine-tuned versions of the models (Table 4.1). Microsoft’s Phi models emerged as the top performers, despite not being instruction fine-tuned. They achieve commendable rouge1, rougeL, and phrase match metrics, showcasing their strength as language-generation models.

Adding chain-of-thought prompting into the experiments improves performance for most of the models. Notably, Phi-2 gets the highest phrase match score of 0.445

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.118	0.258	0.178	0.250	0.250	0.210
gemma-2b-it	0.24	0.428	0.377	0.423	0.426	0.590
phi-2	0.181	0.359	0.293	0.351	0.354	0.555

Table 4.3: Evaluations for fine-tuned concise answering. We evaluate fine-tuned models w/o chain-of-thought reasoning on our test set (correct answer option and answer only).

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.082	0.192	0.114	0.183	0.184	0.280
gemma-2b-it	0.078	0.178	0.102	0.168	0.169	0.195
phi-2	0.100	0.220	0.138	0.199	0.199	0.330

Table 4.4: Evaluations for fine-tuned chain-of-thought reasoning. We evaluate fine-tuned models w/ chain-of-thought reasoning on our test set (Explanation of correct answer followed by correct answer option and answer).

on the test set (Table 4.2), without fine-tuning or providing additional context during inference.

For the following experiments, we consider only the $\leq 2\text{B}$ parameter models i.e. TinyLlama, Gemma-2B and Phi-2.

Finetuning on TextQA: Finetuning on the TextQA subset improves performance across several metrics, even on the least performant models such as TinyLlama (Table 4.3). We notice a significant improvement in phrase match scores, with Gemma-2B having the highest score of 0.590. ROUGE and BLEU scores also improve for all models.

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.064	0.161	0.088	0.152	0.154	0.180
gemma-2b-it	0.113	0.227	0.138	0.22	0.221	0.200
phi-2	0.169	0.297	0.232	0.292	0.293	0.540

Table 4.5: Evaluations for concise answering with context. We evaluate baseline models w/o chain-of-thought reasoning and RAG-based context on our test set (correct answer option and answer only).

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.043	0.137	0.056	0.125	0.125	0.110
gemma-2b-it	0.091	0.210	0.141	0.202	0.205	0.270
phi-2	0.177	0.305	0.244	0.298	0.301	0.585

Table 4.6: Evaluations for chain-of-thought reasoning with context. We evaluate baseline models w/ chain-of-thought reasoning and RAG-based context on our test set (Explanation of correct answer followed by correct answer option and answer).

However, we observe that fine-tuned models with chain-of-thought reasoning do not carry over the same trend (Table 4.4, with Phi-2 reporting lower performance compared to its baseline scores. This implies that fine-tuning with chain-of-thought reasoning requires a longer training period and different hyperparameters to adapt to the new task. However, the TinyLlama model appears to learn better in the limited training period, achieving a high phrase match score of 0.28.

Testing with Retrieval-Augmented Generation: We observe the highest scores in these experiments, with baseline models utilizing RAG-based context w/ and w/o chain-of-thought reasoning having a significant boost across metrics (Table 4.5, 4.6), compared to baseline experiments. Phi-2 performs consistently above the other LLMs, reporting a phrase match of 0.585 when using context in conjunction with chain-of-thought reasoning.

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.145	0.224	0.144	0.219	0.219	0.130
gemma-2b-it	0.217	0.406	0.352	0.398	0.400	0.545
phi-2	0.222	0.428	0.366	0.425	0.424	0.575

Table 4.7: Evaluations for fine-tuned concise answering with context. We evaluate fine-tuned models w/ RAG-based context on our test set (correct answer option and answer only).

Model	BLEU	ROUGE				Phrase Match
		rouge1	rouge2	rougeL	rougeLsum	
tinylama-1.1b-chat-v1.0	0.089	0.205	0.124	0.196	0.198	0.315
gemma-2b-it	0.062	0.168	0.080	0.153	0.153	0.155
phi-2	0.096	0.215	0.134	0.198	0.196	0.325

Table 4.8: Evaluations for fine-tuned chain-of-thought reasoning with context. We evaluate fine-tuned models w/ chain-of-thought reasoning and RAG-based context on our test set (Explanation of correct answer followed by correct answer option and answer).

Additionally, using fine-tuned models in these experiments also improves the BLEU and ROUGE scores, as observed in Table 4.7. As observed in the previous experiments, chain-of-thought reasoning needs to be improved upon, these fine-tuned models do not perform better in any metric (Table 4.8).

Chapter 5

Conclusions

5.1 Contributions

In this work, we first design a PPO-based routing policy that learns to switch seamlessly between local and cloud policies depending on the situation of the observation and task objective. We propose Ecological Navigation Score, an evaluation metric in CARLA that takes into consideration route deviations and energy consumption to provide a better metric for scoring driving agents. Second, we present a novel dataset designed to assess the performance of LLMs and VLMs for autonomous driving. Finally, we evaluate several tiny LLMs on this dataset and benchmark their performance.

Firstly, our study presents an offloading policy that can offload inference for real-time decision-making without compromising on performance. We provide empirical evidence that demonstrates the impact of the new metric and show that autonomous driving agents can be improved by using this metric as a benchmark for real-world readiness. This work is under review for publication at the European Conference on Computer Vision (ECCV) 2024.

Secondly, our investigation focuses on the subtle aspects of LLMs in the context of autonomous driving. We demonstrate that despite employing various strategies, LLMs exhibit poor generalization to driving tasks, as evidenced by their subpar performance on our novel dataset. Even with optimal fine-tuning and the utilization of retrieval-augmented generation for deriving context from driving handbooks, LLMs

struggle to achieve the 80% threshold required to pass the learner’s license examination.

We realize that incorporating chain-of-thought reasoning into LLMs requires an enhanced training protocol to learn step-by-step rationale, particularly if it cannot do so. A promising initial approach would be to simultaneously fine-tune the RAG embedding module and the LLM. This dual refinement process would allow for detailed inference during tasks such as the learner’s examination. Elevating the LLM’s capacity for reasoning represents a significant stride towards achieving human-level performance on factual tasks.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Booth, S., Knox, W. B., Shah, J., Niekum, S., Stone, P., and Allievi, A. (2023). The perils of trial-and-error reward design: misdesign through overfitting and invalid task specifications. In *AAAI*.
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631.
- Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pages 2722–2730.
- Chen, D. and Krähenbühl, P. (2022). Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- De Haan, P., Jayaraman, D., and Levine, S. (2019). Causal confusion in imitation learning. *Advances in neural information processing systems*, 32.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- Ding, S. and Lin, D. (2020). Dynamic task allocation for cost-efficient edge cloud computing. In *SCC*.
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

- Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE.
- Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.
- Jaeger, B., Chitta, K., and Geiger, A. (2023). Hidden biases of end-to-end driving models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8240–8249.
- Kag, A., Fedorov, I., Gangrade, A., Whatmough, P., and Saligrama, V. (2022). Efficient edge inference by selective query. In *ICLR*.
- Karras, K., Pallis, E., Mastorakis, G., Nikoloudakis, Y., Batalla, J. M., Mavromoustakis, C. X., and Markakis, E. (2020). A hardware acceleration platform for ai-based inference at the edge. *Circuits, Systems, and Signal Processing*, 39(2):1059–1070.
- Knox, W. B., Allievi, A., Banzhaf, H., Schmitt, F., and Stone, P. (2023). Reward (mis) design for autonomous driving. *J. Artif. Intell.*
- Koonce, B. (2021). Convolutional neural networks with swift for tensorflow. *Berkeley (CA): Apress*, pages 63–72.
- Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., and Lee, Y. T. (2023). Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.
- Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Moerland, T. M., Broekens, J., Plaat, A., Jonker, C. M., et al. (2023). Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118.
- Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. (2018). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*.
- Penmetcha, M. and Min, B.-C. (2021). A deep reinforcement learning-based dynamic computational offloading method for cloud robotics. *Access*.
- Pomerleau, D. A. (1988). Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*.
- Schneider, N., Piewak, F., Stiller, C., and Franke, U. (2017). Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Sudhakar, S., Sze, V., and Karaman, S. (2022). Data centers on wheels: Emissions from computing onboard autonomous vehicles. *IEEE Micro*, 43(1):29–39.
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.
- Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. (2024). Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Torabi, F., Warnell, G., and Stone, P. (2018). Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Wang, J., Hu, J., Min, G., Zhan, W., Ni, Q., and Georgalas, N. (2019). Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning. *IEEE Communications Magazine*, 57(5):64–69.
- Yang, T.-J., Chen, Y.-H., and Sze, V. (2017). Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5687–5695.
- Zhang, P., Zeng, G., Wang, T., and Lu, W. (2024). Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.
- Zhu, W. and Rosendo, A. (2022). Psto: Learning energy-efficient locomotion for quadruped robots. *Machines*, 10(3):185.
- Zhuang, Z., Fu, Z., Wang, J., Atkeson, C., Schwertfeger, S., Finn, C., and Zhao, H. (2023). Robot parkour learning. In *CoRL*.

CURRICULUM VITAE

Sandesh Bharadwaj

sandeshb@bu.edu

EDUCATION

Boston University (BU) - Boston, MA Sep 2022 - Present
M.S. in Computer Science - Advisor: Eshed Ohn-Bar
Area of Interests: Vision & Language, Autonomous Driving

IIITDM Kancheepuram, India Jul 2015 - Jun 2020
Bachelor of Technology in Electronics and Communication Engineering
Master of Technology in Signal Processing and Communication System Design

WORK EXPERIENCE

Human-to-Everything (H2X) Lab, Boston University Mar 2023 - Present
Graduate Research Assistant

- Unified Local-Cloud Decision-Making via Residual Reinforcement Learning. (Under Review, ECCV 2024)
- Research on tiny vision-language models for autonomous driving.
- Devised efficient algorithms for cloud-edge collaboration, outperforming existing algorithms by 17%.
- Created a custom environment for crowd navigation in CARLA simulator, combining OpenAI Gym and Stable Baselines 3 for training reinforcement learning algorithms.
- Developed visual servoing in multi-robot simulations with RGB and depth sensors.
- Compared reinforcement learning algorithms (DDQN, PPO) for person-following and integrated human feedback for improving performance.

Teaching Assistant Jan 2023 - Present

- Teaching assistant for ENG EC 444 - Smart and Connected Systems, an introductory course to cyber-physical and IoT systems.
- Topics: Microcontroller architecture, I/O interfaces, real-time OS programming and control, wireless personal area networks (WPANs), IP gateways, mobile cloud computing, reliability, security, and privacy.

- Responsible for assisting the Instructor with course logistics, conducting classes, and grading.

Ottometric Inc., Boston, MA

Jun 2023 - Aug 2023

Software Engineering Intern

- Created a fork of *submodlib* library to utilize weighted features for data summarization.
- Devised data summarization algorithms based on submodular optimization for the Ottoviz platform.
- Crafted custom features to separate important images as subsets (1%) from large datasets for efficient training of deep learning models.

Boston University, Boston, MA

Jan 2023 - Apr 2023

Independent Researcher

- Researched music source separation with Iddo Drori (BU/MIT) and Nikhil Singh (MIT Media Lab).
- Developed modified architecture using noise-invariant loss and band-splitting.

Ignitarium Technology Solutions, Bangalore, India

Jan 2022 - Jul 2022

Engineer, AI and Cloud

- Wrote optimized production code to make neural networks compatible with imAligne SDK of Untether AI.
- Revamped legacy code and reduced size of codebase by 40%.
- Led team, and provided critical support during project phases (Python, CUDA, C++, Bash).

Synopsys India, Bangalore, India

Jan 2021 - Jan 2022

Graduate Engineer Trainee

- Optimized large-scale data extraction and processing leveraging Apache Spark RDD and GraphX APIs.
- Fine-tuned NLP models on custom logs with 92% accuracy, deployed to alert users about runtime issues.
- Redesigned outdated applications for log analysis, streamlining them into a unified production system that offers enhanced user experience and accessibility for current users.
- Built log analysis pipeline with Apache Airflow and Logstash to isolate unique events, track and notify users if intervention is required (Python, Apache Airflow, Logstash, Elasticsearch, Apache Spark, Bash).

Robert Bosch GmbH, Bangalore, India

Jan 2020 - Jun 2020

Software Engineering Intern

- Adapted open-set classification for identifying rare/unknown objects in vehicular datasets.
- Merged traditional classification models with meta-learning and few-shot algorithms, evaluated on subset of Open Images dataset. 56% accuracy on unknown image classification in testing.
- Worked on a time-critical project to enable predictive diagnosis of automobile parts, identify degradation, and suggest preemptive actions (Python, Keras/TensorFlow 2.1, PyTorch, OpenCV).

CDAC, Kolkata, India

May 2019 - Oct 2019

Research Intern

- Led and implemented modified approach to person re-identification based on dynamic gait.
- Achieved best accuracy of 91.13% on CASIA-B Gait Dataset (Python, Pandas, Numpy, Scikit-learn).

Thermo Fisher Scientific Pvt. Ltd. Hyderabad, India

May 2018 - Jul 2018

Summer Intern

- Developed proof-of-concept for upgrading an existing product, providing nearly $2\times$ processing power while cutting manufacturing and development costs by almost 70%.
- Developed ARM micro-controller, designed GUI for user-friendly product interaction (C++, Qt5, Yocto).

PUBLICATIONS

- K. Sengupta, **S. Bharadwaj**, Z. Shangguan, S. Arora, E. Ohn-Bar, R. Mancuso, “Unified Local-Cloud Decision-Making via Residual Reinforcement Learning.” (Under Review), European Conference on Computer Vision (ECCV) 2024
- **S. Bharadwaj**, K. Chanda, “Person Re-Identification by Analyzing Dynamic Variations in Gait Sequences”, Emerging Technologies for Computing, Communication and Smart Cities 2020

SKILLS

- **Languages:** Python, C/C++, MATLAB, Java, Scala
- **Tools & Frameworks:** OpenCV, Huggingface Transformers, Lightning, Scikit-Learn, Pandas, Selenium, Soundfile. PyTorch, Keras/TensorFlow, OpenStack, OpenShift, CUDA, Robot Operating System (ROS), Apache Spark, Apache AirFlow
- **Datastores:** MySQL, Logstash, ElasticSearch
- **DevOps:** AWS, GCP, Docker, Git, Perforce
- **OS:** Ubuntu, Linux Yocto, Arch Linux, Windows, CentOS
- **Simulators:** CARLA, OpenAI Gym