Reinforced Learning Based Algorithm: Reducing Accidents and Increasing Road Safety

by

Syed Ishmum Ahnaf
21301347
Md Zahin Abrar Badruddoza
21301377
Salauddin Mahmood Rafid
21301174

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

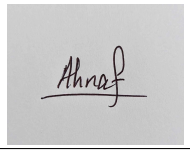
Department of Computer Science and Engineering
Brac University
January 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Syed Ishmum Ahnaf
21301347

Md Zahin Abrar Badruddoza
21301377

Salauddin Mahmood Rafid
21301174

Approval

The thesis/project titled "Reinforced Learning Based Algorithm: Reducing Accidents and Increasing Road Safety" submitted by

1. Syed Ishmum Ahnaf (21301347)
2. Zahin Abrar Badruddoza (21301377)
3. Salauddin Mahmood Rafid (21301174)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 31, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Safety on the roads is of utmost importance for both autonomous and human-driven vehicles. State-of-the-art accident anticipation methods, essentially based on supervised learning, show promise but are bounded by their dependence on large labeled datasets and their inability to generalize straightforwardly to new driving environments. This research extends the existing framework of DRIVE, a deep reinforcement learning model that predicts accidents from dashcam videos by mimicking human visual attention. DRIVE proposed a new approach that involves dynamically learned adaptive policies through integrated visual attention and accident prediction. Through this paper, we will be integrating the DRIVE framework tightly with TD3, refining the reward mechanisms of DRIVE in the process; mainly, the dense anticipation rewards and sparse fixation rewards. We would like to explore how such enhancement can further result in early and accurate accident prediction together with robust visual explanations for its decisions while making the computation less hardware intensive. Preliminary insights could also provide the fact that an optimized DRIVE might bring a sea of change in the accuracy and timeliness of accident predictions that will go a long way in ensuring much safer and more reliable autonomous driving systems. This work underlines the imperative of continuous innovation in advanced technologies to check reckless driving and improve road safety globally.

Keywords: Reinforcement Learning (RL), Autonomous Vehicles (AV), Accident Prevention, Sparse Fixation reward, Soft Actor-Critic (SAC) Algorithm, Twin-Delayed Deep Deterministic (TD3) Algorithm, Dense Anticipation Reward

Table of Contents

Declaration i

Approval ii

Abstract iii

Table of Contents iv

1 Introduction 2
- 1.1 Introduction . . . 2
  - 1.1.1 Reliance on Large Labeled Datasets . . . 2
  - 1.1.2 Lack of Explainability . . . 2
  - 1.1.3 Poor Generalization Ability . . . 2
- 1.2 Proposed Approach . . . 3
  - 1.2.1 Explicitly Simulating Human Visual Attention . . . 3
  - 1.2.2 The Importance of Dense Anticipation Reward . . . 3
  - 1.2.3 Leveraging a Sparse Fixation Reward . . . 3
  - 1.2.4 Implementing an Improved DRL Training Algorithm . . . 3

2 Research Problem and Objective 5
- 2.1 Research Problem: A Need for Explainable and Adaptive Accident Anticipation . . . 5
  - 2.1.1 The Need for an Early and Accurate Prediction Model . . . 5
  - 2.1.2 The Lack of Visual Explainability in Existing Models . . . 5
- 2.2 Research Objectives: Towards Safer AVs through Explainable Accident Anticipation . . . 6
  - 2.2.1 Develop a Visually Explainable Accident Anticipation Model . 6
  - 2.2.2 Improve the Performance of Accident Anticipation Models . . 7

3 Literature Review and Work Plan 8
- 3.1 Literature Review . . . 8
  - 3.1.1 Joining the dots between visual attention and accident anticipation . . . 8
  - 3.1.2 Supervised Learning Methods . . . 8
  - 3.1.3 Strengths and Weaknesses of Supervised Learning . . . 9
  - 3.1.4 Using Reinforcement Learning for Visual Attention Modeling . 9
  - 3.1.5 This Research: Building Upon and Contributing to the State-of-the-Art . . . 10

4 Methodology **12**
4.1 Methodology . . . 12
4.1.1 Framework Overview . . . 12
4.1.2 Problem Setup . . . 13
4.2 A guide into building the environment . . . 14
4.3 Representing the state . . . 15
4.4 The functionality of the Stochastic Multi-Task Agent . . . 16
4.5 Reward Function . . . 17
4.5.1 Dense Anticipation Reward . . . 17
4.5.2 Sparse Fixation Reward . . . 18
4.6 Changes Implemented in Model Training . . . 19
4.6.1 Training Setup and Workflow . . . 19
4.6.2 Critic and Actor Updates . . . 20
4.6.3 Integrating TD3 with DRIVE's Architecture . . . 20
4.6.4 Hyperparameters and Stabilization . . . 21
4.6.5 Advantages of TD3 for DRIVE . . . 21
4.6.6 Challenges and Mitigations . . . 21
4.6.7 Summary of Our Deep Reinforcement Learning Contribution . 22

5 Experiment **23**
5.1 Experiments . . . 23
5.1.1 Evaluation Metrics . . . 23
5.1.2 Implementation . . . 24
5.1.3 Results . . . 24
5.1.4 Graph Analysis . . . 25
5.1.5 Graphs . . . 26

6 Conclusion **29**
6.1 Conclusion . . . 29

Bibliography **32**

Chapter 1

Introduction

1.1 Introduction

A critical aspect that would enable autonomous vehicles to be successfully integrated into the fabric of society involves being able to autonomously and safely navigate the roads. The most crucial ability in this regard is being proactive with potential traffic accidents. To proactively anticipate danger, human drivers instinctively rely on a combination of experiences, visual cues, and an understanding of the dynamics of traffic to become familiar with difficulties and challenges. The existing accident anticipation models are mainly developed following the supervised learning paradigm and show proficiency in this task, yet they suffer from several limitations:

1.1.1 Reliance on Large Labeled Datasets

Supervised learning models require massive amounts of accurately annotated data for training. This reliance presents a practical obstacle as collecting and labeling such data is often expensive, time-consuming, and may not cover the wide range of real world driving scenarios.

1.1.2 Lack of Explainability

Although deep learning models are good at learning very complex patterns, they often end up as black boxes in terms of offering any kind of explanation as to why a particular prediction is made. Ultimately, it is a complex model to comprehend because of the non transparent process through which it makes decisions. This, therefore, makes the whole model opaque and prohibits trust, which can be a significant hindrance, especially in the debugging phase to improve performance for safety-critical applications like autonomous driving.

1.1.3 Poor Generalization Ability

Supervised learning models tend to work well only in the particular settings in which they are trained. However, for many new or even different settings, they fail to generalize to new, unseen data. Such inflexibility would represent a significant bottleneck in practical driving, in which AVs must adapt at every moment to cope

with changes in the operation environment, unexpected events, or even different driving behaviors.
The following paper addresses such limitations by exploring a fundamentally different approach to accident anticipation regarding the human visual system. In essence, we delve into the research question: *Where do drivers look when anticipating possible future accidents?* We accomplish this by understanding where drivers look to infer potential risks and develop more intelligent and intuitive models of accident anticipation for AVs. Such learning makes it possible to develop models that not only make more accurate predictions of accidents but also predict them at an earlier stage so that an explanation can be made for the prediction to build trust and transparency concerning the model.

1.2 Proposed Approach

The proposed approach, the DRL-based model for Deep Reinforced accident anticipation with Visual Explanation (DRIVE) as first brought up in [17], overcomes those above limitations in the following ways:

1.2.1 Explicitly Simulating Human Visual Attention

DRIVE contains a mechanism that simulates human visual attention, which allows it to focus dynamically on the relevant areas and features within video frames, much in the same way a human driver would. These innovations not only make the model more robust in feature extraction in predicting accidents but also give the visualized regions to which the model attends — a form of visual explanation for its decisions.

1.2.2 The Importance of Dense Anticipation Reward

DRIVE uses a new reward function that promotes early and accurate prediction by heavily rewarding correct predictions at each time step, with the reward magnitude increasing with prediction times. This, in turn, drives the model to predict crashes as early as possible through the prediction while preserving accuracy.

1.2.3 Leveraging a Sparse Fixation Reward

To overcome the challenge of limited fixation data, DRIVE uses a sparse reward scheme that only gives feedback to predictions made at frames where there is ground-truth fixation data. Given the visual attention mechanism, sparse rewards enable the training from a small number of annotations for fixations.

1.2.4 Implementing an Improved DRL Training Algorithm

DRIVE's training currently implements an improved algorithm as discussed in [20], the Soft Actor-Critic (SAC) algorithm that is well-known to be stable and efficient in handling continuous action spaces. Advanced techniques, such as automatic entropy tuning, make the training process more time-efficient, with exploration and stable convergence characteristics.

With these elements combined, DRIVE reliably provides a much more robust, explainable, and adaptable accident anticipation system that can contribute to the development of much safer and more reliable AVs than currently exist.

Chapter 2

Research Problem and Objective

2.1 Research Problem: A Need for Explainable and Adaptive Accident Anticipation

This research tackles two vital problems in order to overcome the shortcomings of the traffic accident prediction techniques that are currently in use:

2.1.1 The Need for an Early and Accurate Prediction Model

Rather than just predicting an accident might occur, a successful accident anticipation system must predict about the accident before a minimum time where the driver/rider will get time to make necessary decisions and actions to save him/her from the accident.

2.1.2 The Lack of Visual Explainability in Existing Models

Adoption and trust are severely hampered by the use of black-box deep learning models in accident anticipation systems. Comprehending the reasoning behind a model's prediction is particularly important for safety-sensitive applications such as autonomous driving. The model's decision-making process becomes opaque which makes it much harder to debug, assess, and enhance its performance.
Current methods which primarily rely on supervised learning are inadequate for meeting these two crucial requirements:

1. **Supervised Learning Models Require a Lot of Labeled Data:**

   As discussed earlier, SL models cannot be trained for accident prediction without a lot of annotated data which can be difficult and expensive to obtain. Furthermore, the model can't learn from its own actions or adjust to new scenarios if it only uses labeled data.

2. **Supervised Learning Models Lack Dynamic Interaction with the Environment:**

   Since pre-defined datasets are usually used to train SL models, this limits their capacity to pick up on and adjust to the dynamic nature of real-world traffic situations. They cannot actively explore different strategies, learn from their mistakes, or adjust their behavior based on real-time feedback.

Our research tackles these limitations by introducing two key innovations:

1. **Utilizing Reinforcement Learning to Dynamically Balance Exploration and Exploitation:** DRL enables DRIVE to learn optimal behavior through interaction with a simulated driving environment. The agent receives rewards for making accurate and early predictions and penalties for inaccurate predictions or collisions, learning to maximize its cumulative reward over time. This trial-and-error learning approach allows DRIVE to actively explore different strategies, learn from its experiences, and adapt to varying driving conditions.

2. **Explicitly Modeling Driver Visual Attention:** DRIVE incorporates a mechanism that simulates how human drivers allocate their attention when anticipating potential accidents. By focusing on the salient features and regions within video frames, the model can learn to extract more relevant information for accident prediction. Moreover, by visualizing the attended regions, we can gain insights into the model's decision-making process providing a form of visual explanation for its predictions. This transparency enhances trust and facilitates debugging and improvement.

This integration of visual attention with a DRL framework allows DRIVE to overcome the limitations of SL-based approaches, offering a more robust, adaptable, and explainable solution for traffic accident anticipation.

2.2 Research Objectives: Towards Safer AVs through Explainable Accident Anticipation

The primary goal of this research is to work on the current accident prevention systems to create a model that has the ability to predict accidents more accurately. Such a model could then be implemented in autonomous vehicles which would lead to such vehicles being safer and more dependable. To do so, the following two goals are considered:

2.2.1 Develop a Visually Explainable Accident Anticipation Model

Our objective is to build a model that not only perfectly recognizes accidents but also provides a visual description for its prediction. This explainability will increase transparency and confidence which is essential for AV uptake and public acceptance.

2.2.2 Improve the Performance of Accident Anticipation Models

To further improve the safety and dependability of AVs, we want to create a model that outperforms current techniques in terms of accuracy and early warning while also requiring lower computational power.
These goals translate into the following research questions:

- Can a DRL-based model with its ability to dynamically interact with the environment and learn adaptive policies significantly improve accident prediction accuracy and earliness compared to SL-based models?
- Can visual attention, by highlighting the features and regions considered crucial by the model, effectively serve as an explanation mechanism for its predictions, enhancing trust and transparency?

We will combine a number of established evaluation metrics to determine the research's performance objectively:

Area Under the ROC Curve (AUC) AUC is a commonly used metric which evaluates how well a model is performing on a classification problem. The model results are plotted on a graph with True Positive Rate on the y-axis and True Negative Rate on the x-axis. The area under the curve drawn on this graph is known as the AUC. A higher AUC indicates a better performance.

Mean Time-to-Accident (mTTA) mTTA quantifies the model's earliness in anticipating accidents. It measures the average time difference between when the model first predicts an accident with sufficient confidence and when the accident actually occurs. A higher mTTA indicates earlier and more timely anticipation.

Saliency Evaluation Metrics Saliency evaluation metrics will be used to determine how effectively the visual attention model captures driver attention. The following metrics are mostly used:

1. **Similarity (SIM)**: Checks how much the predicted attention map overlaps with the ground truth attention map which is based on data from human eye movements.
2. **Linear Correlation Coefficients (CC)**: Evaluates the linear relationship between the predicted attention map and the ground truth, indicating how closely they match.
3. **Kullback-Leibler Divergence (KLD)**: Measures how much the predicted attention map and the ground truth attention map differ from an information-theoretic perspective. A lower KLD indicates better alignment between the model's attention allocation and human attention.

Through a detailed analysis of our model using these metrics, we aim to present a full evaluation of its performance, comprehend its advantages and disadvantages, and emphasize its potential to improve the dependability and safety of autonomous driving systems.

Chapter 3

Literature Review and Work Plan

3.1 Literature Review

3.1.1 Joining the dots between visual attention and accident anticipation

Predicting traffic accidents is a challenge because there are many sides that need to be considered first to understand a situation properly. Most importantly, as discussed in [21], it is crucial to find a proper trade off between accuracy and early prediction. It is, also, imperative for a system to understand driving behaviors, traffic dynamics, and the ability to identify subtle visual indicators which can be used to predict accidents more accurately. Researchers have investigated numerous strategies to tackle this issue, however, deep learning is currently one of the most used solutions.

3.1.2 Supervised Learning Methods

Supervised learning has been the dominant paradigm for accident anticipation, with several studies proposing different model architectures and training strategies. Some notable examples include:

DSA-RNN: Chan et al. introduced the the DSA-RNN model. [5] uses object detection-based spatio-temporal features and dynamic soft-attention methods to provide an input to a VGG16 model[7]. The next step is to enter these extracted features into an RNN for a sequential prediction of the accident scores.

AdaLEA: Suzuki et al. proposed AdaLEA.[12] It is a novel loss function designed to improve the earliness of accident anticipation while still providing a high rate of accurate prediction. By adaptively weighting the importance of different time steps during training, AdaLEA encourages the model to make accurate predictions earlier in the video sequence.

GCN : Recognizing the importance of capturing relationships between different objects and events in video frames, Bao et al.[16] investigated the application of GCNs for accident anticipation. GCNs excel at modeling relational data and can learn to represent complex interactions between entities within a scene, potentially leading to more informed and accurate predictions.

Bayesian Deep Learning : In addressing the inherent uncertainty in predicting accidents, Bao et al.[16] investigated Bayesian deep learning methods. These meth-

ods allowed a systematic approach to quantify uncertainty in model predictions. As a result, more reliable predictions and estimations could be made which would lead to better decision making, even in uncertain situations.

3.1.3 Strengths and Weaknesses of Supervised Learning

Supervised learning models have been widely used by researchers because of highly accurate results, however, there are also several limitations to such a model:
Strengths of Supervised Learning:

- **High Prediction Accuracy:** Supervised Learning models are highly accurate for labeled datasets and systems, allowing such models to be used for specific scenarios they are trained on.

Weaknesses of Supervised Learning:

- **Extensive Labeled Data Requirement:** According to a research by [19], Supervised Learning models obtain their high accuracy due to the extensive set of labeled datasets provided to train the model, however, collecting such a large dataset is time-consuming and costly which has also been discussed in [15], especially for complex tasks such as accident anticipating which involve a wide range of driving scenarios and visual cues.

- **Lack of Explainability:** Even though supervised learning models are largely used by researchers worldwide, their black-box nature of deep learning used in such models make it difficult to understand the reasoning behind judgements[18]. This makes it harder to debug the model and therefore improve the accuracy.

- **Not having Complete Dynamic Interaction with the Environment:** Supervised Learning models behave with a very high degree of accuracy, however, this high accuracy mostly works best with static datasets. However, in the case of accident predictions or similar situations, it is important for the model to be able to adapt to new situations and learn from its actions.

3.1.4 Using Reinforcement Learning for Visual Attention Modeling

Reinforcement Learning offers a good alternative to supervised learning. The functioning of Reinforcement Learning is more dynamic and adaptive than Supervised Learning which results in the system becoming more adept at a wide range of situations as has been seen in [21]. Reinforcement Learning allows agents to learn optimal behavior by direct interaction with the environment. This is done by allowing the agent to try out different actions and set rewards or penalties based on desirable and undesirable behavior. This process allows the agent to progressively learn to maximize the cumulative reward obtained and, in turn, achieve their objectives. Applying Reinforced Learning to visual attention modeling, in the context of driving has yielded positive results.

Several key trends in RL-based visual attention modeling have emerged:
Actor-Critic Algorithms: Actor-critic algorithms, such as Asynchronous Advantage Actor-Critic (A3C) as observed in [9] and Soft Actor-Critic (SAC) as demonstrated in [11] have become increasingly popular for learning complex control policies in the case of continuous action spaces. Comparing these algorithms to traditional Q-learning methods, demonstrates a more effective balance between exploration and exploitation and offers better sample efficiency as well. As a result, such algorithms are much more suitable for challenging domains like autonomous driving.
Visual Fixation Prediction: For an accident-prevention system to be more efficient, it is important to predict where a driver will look next which is known as visual fixation. This offers valuable insights into the attention allocation and decision making processing of a human driver. Reinforced Learning based algorithms have shown success in visual fixation predictions, demonstrating their ability to predict and model human attention [8].
Imitation Learning and Inverse Reinforcement Learning: Two methods to highlight for Reinforcement Learning methods are Imitation Learning and Inverse Reinforcement Learning. Imitation learning works by leveraging expert demonstrations. Recordings of skilled human drivers in different driving conditions are used for initializing policies and improving sample efficiency. This is considered as a promising approach since Imitation Learning allows the model to directly learn a policy from expert demonstrations and apply it. On the other hand, Inverse Reinforcement Learning allows the model to view an expert's action from a demonstration and then infer the reasoning behind the action, as a result, setting up the reward function for the model.

3.1.5 This Research: Building Upon and Contributing to the State-of-the-Art

This research builds upon existing literature by integrating visual attention and accident anticipation into a unified RL framework. This novel approach addresses the limitations of SL-based methods and offers several key advantages:
Enhanced Earliness and Accuracy of Anticipation: By explicitly modeling the driver's visual attention process, DRIVE can learn to focus on the most relevant visual cues for predicting accidents. This focus on informative features has the potential to lead to more accurate and timely predictions, enabling earlier intervention and accident prevention.
Improved Explainability and Trust: Incorporating visual Attention not only improves the predictive performance but also has a way to explain the decisions made by DRIVE. We can visualize these attended regions on video frames so as to better understand why the model is focusing on a particular part of the data, improving interpretability and trust. This type of visual explainability is essential when it comes to convincing the public that they can trust autonomous driving systems.[22]
Dynamic Interaction with the Environment: With this, DRIVE[17] operates at the intersection of the driving environment using the reinforcement learning (RL) framework, learning from the consequences of its actions and updating its course of action for new situations. This interaction and adaptation are necessary to handle the erratic and always changing situations from the real-world traffic. DRIVE can

then gradually increase its capabilities, by holistically learning new driving scenarios as they arise and unforeseen challenges to improve over time.

This research contributes to the field by:

1. **Modifying Novel Reward Functions**: The dense anticipation reward and sparse fixation reward are specifically designed to encourage the model to learn an optimal policy for early and accurate accident prediction while effectively learning from limited fixation data. These rewards guide the agent toward the desired behavior and enhance the efficiency of training. The reward functions are modified further to reduce the impact of false negatives.

2. **Implementing an Improved DRL Training Algorithm**: The proposed training algorithm adapts the SAC algorithm [11] to accommodate the multi-task agent (accident score prediction and fixation prediction). It also incorporates advanced techniques like temperature clipping and state reconstruction to improve the stability and efficiency of learning.

3. **Developing a Visually Explainable Model**: DRIVE's integration of visual attention provides a unique and insightful way to understand its predictions. By visualizing the attended regions, we can make the model's decision-making process transparent, enhancing trust and facilitating debugging and improvement.

By bridging the gap between visual attention and accident anticipation through a unified RL framework, this research paves the way for developing safer, more reliable, and trustworthy autonomous driving systems.

Chapter 4

Methodology

4.1 Methodology

4.1.1 Framework Overview

One of the foundations of this research was the DRIVE model, and to ensure consistent input data quality and enhance the robustness of our model, we adopted the framework as described in [17]. The usage of a similar data pre-processing method is useful here as it has already been shown effective in normalizing and standardizing datasets in prior studies. This technique was inspired by the approach outlined in [17], where it has demonstrated significant improvements in model performance for similar datasets.
Following the method in [17], we first take a dashcam video as input and pass it along to a stochastic multi-task agent. The task of this agent is to recurrently output the accident score, $\hat{a}$, and the point of next fixation, $\hat{f}$, at each time step based on an observation state from the environment. The environment, from which an observation state is obtained, is built by considering two separate attention mechanisms, namely bottom-up and top-down attentions.

- **Bottom-up attention**: Emphasizes objects or regions in the video frames based on sensory input.
- **Top-down attention**: Focuses on regions considered risky with respect to accident prediction, based on the driver's anticipated fixation point .

The top-down attention [3] focuses on the things a driver usually looks at or is searching for actively, whereas the bottom-up attention [3] highlights objects that pop up and grab immediate attention. These two attention mechanisms are applied to dashcam video frames to construct the environment. The stochastic multi-task agent consists of a shared state auto-encoder with two parallel prediction branches. The accident score and fixation predictions are guided by the rewards r_A and r_F, encouraging earliness, correctness, and attentiveness. While making a prediction, the model simultaneously does two tasks—observing the driving environment through visual attention mechanisms to allocate risky regions and predicting the probability of a future accident.

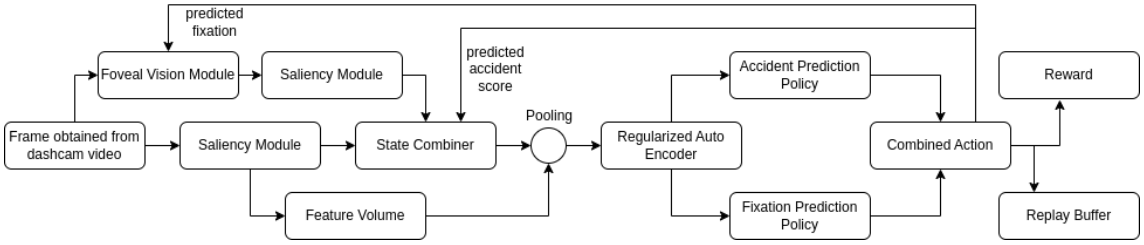


Figure 4.1: Framework for our model inspired by the DRIVE model

4.1.2 Problem Setup

Similar to DRIVE, we follow the task setting according to existing literature [5], [6]. As discussed previously, the trained stochastic multi-task agent predicts whether an accident will take place. An accident anticipation model predicts accidents at the frame level by computing an accident score, a_s, on an individual frame from the video. This indicates the of an accident that might take place in the future. This frame-level accident score can be used to evaluate the performance of the model. To make an evaluation on the earliness of the model however, Time-to-Accident (TTA) is calculated as:

$$TTA = \max(0, t_a - t) \tag{4.1}$$

where t_a is the actual beginning time of the accident, and t is is the first point in time when the frame-level prediction is greater than a pre-defined threshold value. This is an extremely important factor for the evaluation as in any accident the closer a person approaches actual accident time, the more visual cues are available, but for a successful accident prediction model the agent needs to be reliable while also being as early as possible meaning figuring out an accident might take place with less visual cues which makes it stand out. A larger time-to-accident means that the model is able to anticipate the traffic accident. While earliness is extremely important, so are correctness and even attentiveness since we are working with fixation points so to evaluate those, binary classification and saliency evaluation metrics are also used.

The inspiration of DRIVE arises from the natural decision-making process in human drivers where a person observes and, based on that observation, anticipates the next action. So, to make the model act in a similar way, accident anticipation and fixation prediction are considered as a unified Markov Decision Process (MDP). Formally, as provided in the literature of [17], a tuple containing six things- $\{S, A, P, R, L, \gamma\}$, are chosen to represent a discounted MDP. The following provides the description for each of the six chosen attributes from the tuple:

- L: Finite horizon or video length
- S, A: State and action spaces
- R: Reward function
- P: State transition model
- γ: Discount factor, between 0 and 1

In this paper, the action a_t in the action space where the t represents different timestamps consist of the accident score, a_s, and the next fixation point, f. The next fixation point can be obtained using the formulae defined as:

$$f = \left[x^{t+1}, y^{t+1}\right]^T \tag{4.2}$$

which is defined in the image domain such that:

$$a_t = \left[a^t, x^{t+1}, y^{t+1}\right]^T \tag{4.3}$$

The state s_t is shared with two kinds of actions - state representation and action policy which will be introduced in the next section. P, which was defined as the state transition model, is achieved by the fixation prediction mechanism which will be shown later through equation 3 and the environment observation module which will be obtained using equations 1 and 2.

4.2 A guide into building the environment

As mentioned previously, the framework in [17] uses top-down and bottom-up attention mechanisms to create the environment. While this was just a rough summary of the process, this section will look into the details as well as explain how the environment is kept relatively dynamic for the model to help make predictions. To understand how traffic is modeled using the above mentioned mechanisms first we need to identify a symbol for the observation at a current time step, I_t. This time step can be used for the bottom-up attention by using a saliency prediction module G according to $S_{bu} = G(I_t)$. Here, G is instantiated by recent deep convolutional neural networks (CNN) as seen in [7], [14] so that the saliency module can use the feature extraction. In this module, since there is no update to the saliency module by the actions of the agent, and the bottom-up mechanism depends on the saliency module, therefore S_{bu} is only determined by the appearance as is in the video frames. To simulate top-down attention, on the other hand, DRIVE [17] proposed an auxiliary task to predict the fixation point which will dynamically guide the visual attention allocation to the risky region at each time step. However, there is one extra step before passing the environment to the saliency module and that is, the environment at each time step is first fed into a foveal vision module then it is forwarded to the saliency module. The foveal method used has been explored in [2]. The advantage of using a foveal method is that it increases the focus on objects of more importance and since most of the visual cues in an environment are ignored, this method is very helpful in improving the state. The result from the foveal vision module is passed on to the saliency module making the whole formula look like: $S_{td} = G(F(I_t, p_t))$ where G is the CNN method discussed previously. While S_{bu} was mostly static and focused on the frames only, S_{td} is dynamic in nature since it is dependent on the action of fixation prediction. As a result, the agent interacts with the attention-based observation environment dynamically.
To follow the probability nature, both S_{bu} and S_{td} have been normalized in a range of 0 to 1. As mentioned previously, the bottom-up attention highlights the objects that stand out most in an environment whereas the top-down attention is more focused on the risky regions. This is because, as explained previously, in the case of top-down attention the foveal vision module filters out irrelevant visual data and only focuses on fixated areas which indicate a high risk for a future accident.

To combine the two attention mechanisms in [17], DRIVE proposed a novel dynamic attention fusion (DAF) to create a weighted sum of S_{bu} and S_{td} represented by the following equation:

$$S^t = (1 - p^t) * S_{bu} + p^t * S_{td} \tag{4.4}$$

Where p^t is defined as $p^t = \min(m, a^t)$. Here a^t represents the probability of an accident taking place and m is a pre-defined hyper-parameter between 0 and 1. The use of m is extremely important here as it clips a^t meaning if a^t becomes greater than our predefined value of m then instead of using the value of a^t the value of m is chosen. What this does is that the model gets the ability to use the learned top-down attention as m controls the maximum percentage of how much S_{td} is used ($p^t < m$) as described in [17]. For all values of a^t that are less than the pre-defined value of m, the probability p^t is chosen equal to the value of a^t which makes the value a^t and $1 - a^t$ serve as weighing factors in equation 1. The specialty of DAF is that the more probable an accident is the greater the value of a^t making a^t closer to 1 at any current time step, which makes the predicted top-down attention more confident to be used at the next time step.
The equation described in the previous paragraph is most beneficial as it allows us to make the attention fusion dynamic instead of static. As both p^t and S_{td} are dependent on the actions from the stochastic multi-task agent therefore the DAF method dynamically fuses visual attention by considering not just the environment but also the previous decision made by the agent. Instead, if a value would have been manually fixed then the model would only be static by nature as it would no longer be able to change according to past actions or predictions made. According to past research in [17] Dynamic Attention Fusion (DAF) is seen to show significantly better results than Static Attention Fusion (SAF). This also helps with the fact that since the equation uses the last step's prediction to decide, it could be used to visually explain which region is risky.

4.3 Representing the state

The paper which proposed the DRIVE framework in [17] focused on state representation using the feature volume where $V^t \in \mathbb{R}^{C*H*W}$ from CNN-based saliency model G for state representation. However, since DRIVE implemented SAC which is extremely hardware intensive, so to save the GPU memory usage while maintaining the representation capability of CNN features, DRIVE aggregates the feature maps of the volume and does further L2-normalization by global max pooling (fGMP) and global average pooling (fGAP). Afterwards the normalized features are concatenated as the observation state representation as given in the equation below:

$$s_t^i = \text{cat}\left(\tilde{f}_{GMP}\left(S^t \odot V_i^t\right), \tilde{f}_{GAP}\left(S^t \odot V_i^t\right)\right) \tag{4.5}$$

Where $\cdot$ is the element-wise product on the i-th channel of the feature volume and cat is the concatenation along the channel dimension. While TD3 is less hardware intensive than SAC, it is still better to represent the state in this manner for faster computation.

4.4 The functionality of the Stochastic Multi-Task Agent

As previously mentioned in past sections, the agent follows on two actions simultaneously - accident prediction and next fixation point, so to do so the state must be shared with the two tasks. This task sharing brings two advantages for the model:

- As the two tasks use the same state at any specific point in time, using a similar frame allows the model to better understand the correlation between a prediction made by the agent and the fixation point related to the state.
- As the state is shared between both the tasks the workload between the two tasks needed to communicate decreases significantly making the model work more efficiently, especially when the input state is of a higher dimension.

It is important to understand that the entire basis of the agent's work is dependent on the quality of the input state. To tackle this, a common method used usually is the inclusion of auxiliary observation reconstruction task with prediction task of the agent which basically means filling the missing data or state. Based on this idea, DRIVE [17] proposed Regularized Auto-Encoder (RAE), which encodes the state into a more compact and low dimensional latent representation. The framework uses an encoder to encode the state and a decoder to reconstruct the original observation state. The encoding can be simply represented as:

$$\mathbf{z}_t = \mathcal{E}(s_t) \tag{4.6}$$

where $\mathcal{E}$ is the encoder part of RAE.
One issue that most prediction systems face is that they stop exploring once they make a good enough prediction, however in most cases this is not the most optimum solution. To ensure this does not happen and to increase exploration of the environment, the policies for the agent are designed to be stochastic in state-of-the-art DRL algorithms [9]–[11]. DRIVE framework also focuses on a similar approach here as the actions which are associated with accident score and the next fixation point are both drawn from a Gaussian distribution which increases exploration. As mentioned in [17], the framework mentions that the shared latent embedding z_t is used to predict the mean and variance of each action dimension for the two tasks by two parallel policy networks. During training, the agent picks an action, a_t, by sampling from the Gaussian distribution based on the guidance of the policy network parameters which is denoted as ϕ. Both the policy networks are built using a simple layer of neurons with a ReLU activation function, which helps to process information. Additionally, to handle the timing and order of actions (like how one action affects the next), we use a special type of neural network called an LSTM [1]. This is placed after the final layers of the policy networks and helps the agent remember patterns over time, similar to other recent DRL models from [4], [13]. In the testing phase the action output is produced using the predicted mean of the accident score policy ϕ_A and the fixation policy ϕ_F which are concatenated together to give the following equation:

$$\hat{a}_t = \text{cat}\left(\phi_A\left(\mathcal{E}(s_t)\right), \phi_F\left(\mathcal{E}(s_t)\right)\right) \tag{4.7}$$

It is important to highlight that DRIVE [17] specifies that the framework does not directly predict the top-down attention map, but instead predicts the next point of fixation as one of the actions, a_t. The underlying reasoning for this is that using attention map prediction would lead to the generation of a higher dimension action space which would not be efficient to be learned by the agent.

4.5 Reward Function

Based on the state and action, the agent needs a scalar reward r from the environment to help make progressive learning. Since the framework focuses on two functionalities - accident prediction score and next fixation prediction, the agent requires two separate reward functions first. However, it is important to note that accident predictions can change frequently but fixation points do not change as often. So, the DRIVE [17] framework suggests the use of a dense anticipation reward r_A and a sparse fixation reward r_F. Using the values of r_A and r_F the total reward r can be calculated using $r = r_A + r_F$.

4.5.1 Dense Anticipation Reward

As mentioned previously, the accident prediction is made more frequently hence the name "dense". The reward is provided for all time steps by considering both the earliness as well as the correctness of the prediction. To make a comparison, a threshold value, a_0, is predefined and then using an indicator function, the accident prediction score is compared with the threshold. The result obtained from the indicator function is then passed on to a XNOR function with a binary value y where $y \in \{0, 1\}$. Here, $y = 1$ is used to represent that an accident does take place whereas $y = 0$ indicates no accident takes place. The result from the XNOR however is not directly considered to be the reward for the accident anticipation, instead it is weighted using a factor, w_t, obtained from the time step it was for making earliness a crucial factor. Finally, the entire process can be summarized by the following equation:

$$r_A^t = w^t \cdot \text{XNOR}\left[\mathbb{I}\left[a^t > a_0\right], y\right] \tag{4.8}$$

Figure 1 shows the truth table for XNOR.

Table 4.1: XNOR (Equivalence) Truth Table

A	B	$A \leftrightarrow$ B
0	0	1
0	1	0
1	0	0
1	1	1

The truth table indicates that the result of the XNOR comes out as 1 when both input values are similar - y and $a^t > a_0$. This means that XNOR assigns one as the reward to a true prediction - true positive and true negative, and assigns zero to false predictions. While DRIVE [17] highlights that false predictions are really bad in

the context of an accident anticipation system but it was specifically ignored during implementation as the solution to this would be to manually design the weights. As mentioned previously, the use of the weighting factor, w_t, in Equation 4 was used to encourage earliness of the agent's prediction. The design of the weighing factor is normalized because it ensures that both r_A and r_F can be represented and balanced using the same scale:

$$w_t = \frac{1}{e^{t_a - 1}} \left(e^{\max(0, t_a - t)} - 1 \right) \tag{4.9}$$

Where t is the current time step and t_a is the beginning time of a future accident. An exponential is used to reduce the value from 1 to 0 as the time nears accident occurring time. This ensures that the earlier the agent makes the prediction, the more reward it can obtain.

4.5.1.1 Modification to the original reward

To reduce the number of false negative, we introduce a penalty system for the model which is used whenever the model makes a false negative. As false negatives are much more detrimental than any other case in an accident prediction system, we penalize the model heavily in cases of false negatives or false positives. To do so, we use the following approach, where we first identify if an accident will actually take place or not. For accident cases, we compute the penalty using (1-a^t) where a^t is the accident prediction score. This provides a higher penalty for a lower prediction to the model. We convert the predicted accident probability (a^t) to a binary class using a predefined threshold(a_0). Using this threshold, the model can decide whether the prediction was a positive or negative forecast. We then compute an XNOR based operation between the binary predicted class and the actual ground truth by first using logical XOR and then following it using a logical NOT where a value of 1 means that the prediction matched the original ground truth. Finally, we multiply the penalty with a predefined value to scale the value and then subtract it from the original reward to obtain the new penalized reward. This combination ensures that while the model is rewarded for correct and early predictions, it is heavily penalized for false negatives.

4.5.2 Sparse Fixation Reward

As mentioned previously, it is not possible to reward the fixation policy in a similar manner to anticipation as to confirm, the predicted fixation has to be compared with the ground truth fixation, which are dispersed and not as much available. To account for this, DRIVE [17] utilizes a sparse rewarding mechanism that can be represented using the following equation:

$$r_t^F = \mathbb{I}[t > t_a] \exp\left(-\frac{\|\hat{p}^t - p^t\|^2}{\eta} \right) \tag{4.10}$$

As the fixation reward is to be provided after an accident occurs, an indicator function is multiplied to the exponential. What this does is, if t is less than t_a, then the result of the indicator function is zero and the reward comes out as zero. Inside the exponential, $\hat{p}^t$ is used to represent the prediction fixation point while p^t is the ground truth fixation point. Both $\hat{p}^t$ and p^t are 2D coordinates obtained from the

video frame space. $\hat{p}^t$ and p^t are subtracted, and then squared on to find the distance between the prediction and actual fixation point, and then lastly normalized using a hyper-parameter η to ensure it has the same value as r_F and r_A. The use of a negative inside the exponential indicates that the closer the distance between the predicted fixation and the actual fixation, the bigger the reward obtained by the agent.

4.6 Changes Implemented in Model Training

The DRIVE framework in [17] utilizes the Soft Actor-Critic (SAC) model [17] but extends it to accommodate the accident anticipation task. The intention of using SAC is that SAC increases the model's ability to explore compared to traditional actor-critic reinforcement learning (RL) through policy entropy maximization. The primary aim of SAC is to optimize the following objective:

$$\max_{phi} \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\phi}} \left[r(s_t, a_t) + \alpha H(\pi_\varphi(\cdot|s_t)) \right] \tag{4.11}$$

However, since SAC requires extensive hardware usage, we replace it with the deterministic policy optimization of TD3 (Twin Delayed Deep Deterministic Policy Gradient). One issue of TD3, however, is the existence of overestimation, but we tackle this using three key improvements: twin Q-networks, delayed policy updates, and target policy smoothing. The goal is to stabilize training and improve the robustness of the accident anticipation and fixation prediction policies.

Key Components for TD3

- **Twin Q-Networks**: The implementation of two separate Q-networks (Q_1 and Q_2) assists in reducing overestimation bias.
- **Delayed Policy Updates**: To stabilize learning, the actor is updated less frequently than the critics.
- **Target Policy Smoothing**: Gaussian noise is added to the target actions to increase exploration and prevent overfitting.

These components are adapted to DRIVE's existing multi-task reinforcement learning (RL) setup, where the agent predicts both the accident score $\hat{a}^t$ as well as the fixation point $\hat{p}^t$.

4.6.1 Training Setup and Workflow

It has been established that the agent first interacts with the environment by constructing a state using bottom-up and top-down visual attention features. However, since we are using TD3, which is deterministic by nature, the actor network π_ϕ predicts a joint action $\mathbf{a}_t = (\hat{a}^i, \hat{p}^i)$, combining the probability of an accident occurring, $\hat{a}^i$, and the coordinates of the next fixation, $\hat{p}^i$. As we are no longer using a similar approach for exploration as done in [17], we add Gaussian noise to our action. During training, Gaussian noise $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma_{\text{explore}}), -c, c)$ is added to the action,

ensuring the agent explores diverse accident-relevant regions while following a valid action range. We retain the replay buffer from [17], and the total result obtained is stored alongside the transition (s_t, a_t, r_t, s_{t+1}) in a replay buffer $\mathcal{D}$. The advantage of this buffer is that it enables off-policy learning by decoupling data collection from policy updates.

4.6.2 Critic and Actor Updates

During critic updates, instead of using all past transitions, a batch of transitions is sampled from the replay buffer, and the twin Q-networks (Q_1 and Q_2) are trained to minimize the clipped Bellman error. The twin Q-networks are trained by minimizing the clipped Bellman error to reduce overestimation bias by choosing the minimum Q-value obtained from the two networks as a target, y_t, during the critic update. To further stabilize the target action, we use a technique known as target policy smoothing, where we add clipped Gaussian noise represented by $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma_{\text{target}}), -c, c)$. The critic's loss is given by the equation:

$$J(\theta_i) = \frac{1}{N} \sum_{j=1}^{N} \left(Q_{\theta_i}(s_t^j, a_t^j) - y_t^j \right)^2, \quad i \in \{1, 2\} \tag{4.12}$$

where

$$y_t^j = r_t^j + \gamma(1 - d) \cdot \min(\overline{Q}_{\theta_1}, \overline{Q}_{\theta_2}) \tag{4.13}$$

includes a discount factor $\gamma = 0.99$ and terminal state indicator d.
The actor network is updated less frequently compared to the critic network to stabilize training. Initially, we set $K = 2$ steps for the actor network updates, meaning that the actor is updated after two iterations of the training loop. The primary objective of the actor network is to maximize the expected Q-value using Q_1 by the following equation:

$$J(\phi) = \frac{1}{N} \sum_{j=1}^{N} Q_{\theta_1}(s_t^j, \pi_\phi(s_t^j)) \tag{4.14}$$

Finally, we use Polyak averaging ($\tau = 0.005$) to softly update the target actor and critic networks by gradually blending their parameters with the corresponding online networks. This improves stability as this smooth parameter transfer promotes stable learning by reducing abrupt changes in the target values.

4.6.3 Integrating TD3 with DRIVE's Architecture

To ensure consistency is maintained, we integrate the TD3 framework tightly with DRIVE's attention-driven design. As mentioned previously in Section **??**, the state is derived from fusing bottom-up and top-down attention maps, which is then compressed into a latent representation state using the Regularized Auto-Encoder (RAE). This latent vector is shared across both policy branches, ensuring that both the accident score and fixation prediction tasks are optimized together. As the twin Q-networks need to handle the combined action space of accident scores and fixation points, the deterministic actor outputs a joint action vector, $\mathbf{a}_t$, where $\mathbf{a}_t = (\hat{a}^i, \hat{p}^i)$ and $\hat{p}^i$ is used to guide the top-down attention mechanism in a dynamic manner

($S_{\text{ta}^i} = G(F(I^i, \hat{p}^i))$). As the update in attention comes directly from previous predictions and encounters made by the model, the system can better refine where to look based on past guesses—similar to a human driver constantly trying to anticipate what might happen next. This process repeats for each time step, continuously adjusting attention and action together.

4.6.4 Hyperparameters and Stabilization

In our implementation of the TD3 framework for the DRIVE system, we carefully selected several key hyperparameters to ensure a balance between exploration and stability. Specifically, the exploration noise was set to $\sigma_{\text{explore}} = 0.1$ and clipped within the range of $[-0.3, 0.3]$, effectively balancing the need for sufficient exploration while maintaining system stability. For target noise, we utilized $\sigma_{\text{target}} = 0.2$, clipped to $[-0.5, 0.5]$, which helps prevent divergent target actions that could destabilize the learning process. We used two different learning rates for the critics and the actor, training critics at a rate of 3×10^{-4} and the actor at a slower rate of 3×10^{-5}. This slower learning rate for the actor helps avoid abrupt shifts in policy, contributing to a smoother and more reliable learning process. Additionally, we apply batch normalization to the Regularized Auto-Encoder (RAE) encoder, which helps to mitigate gradient instability during backpropagation, further enhancing the overall stability of the training process.

4.6.5 Advantages of TD3 for DRIVE

The TD3 algorithm offers several advantages specific to the multi-task environment of DRIVE. The existence of twin Q-networks and the use of delayed policy updates effectively address the critical limitations observed with Soft Actor-Critic (SAC) in this setting. By utilizing twin critics and taking the minimum Q-value, TD3 reduces overestimation bias, which is particularly important for handling sparse fixation rewards (r_F^i). Furthermore, TD3 employs a deterministic policy instead of SAC's stochastic actions, significantly simplifying deployment in real-time systems where consistent and reliable fixation predictions are essential. Moreover, TD3's exploration strategy, which relies on action noise, is computationally more efficient than SAC's entropy regularization. This efficiency makes TD3 well-suited for training on large-scale dashcam datasets, enhancing sample efficiency without compromising performance.

4.6.6 Challenges and Mitigations

Despite the enhanced stability provided by TD3, certain challenges remain within the DRIVE framework. One significant issue is noise sensitivity, where excessive exploration noise can destabilize fixation predictions. To mitigate this, we implemented aggressive clipping of the exploration noise ($c = 0.3$) and employed annealing of σ_{explore} over time, ensuring that the noise remains within manageable limits throughout training. Another challenge is task interference, arising from the joint optimization of accident scores and fixation points, which can lead to conflicting gradients. To address this, we applied gradient masking to isolate task-specific updates during backpropagation. This prevents interference and ensures that each task is optimized effectively without adversely affecting the other.

4.6.7 Summary of Our Deep Reinforcement Learning Contribution

By integrating TD3 into DRIVE, we ensure that the model retains its original strength—visually explainable attention—while achieving greater robustness and sample efficiency. Since we need to handle sparse rewards and multi-task action spaces, the use of a deterministic policy and twin critics is specifically advantageous, making it more aligned with the original goal of making DRIVE a reliable and real-time accident anticipation system.

Chapter 5

Experiment

5.1 Experiments

This section describes the tests carried out to assess the efficacy of the deep reinforcement learning (DRL) -based DRIVE model for anticipating traffic accidents and predicting visual attention. A sizable traffic dataset, DADA-2000, which includes dashcam videos of different driving situations, were used for the tests.

5.1.1 Evaluation Metrics

A variety of criteria were employed, with an emphasis on accident anticipation and visual attention prediction, to assess the DRIVE model's performance. These metrics ensure the model's performance is evaluated in terms of both prediction accuracy and decision-making explanation.

5.1.1.1 Accident Anticipation Metrics

Area Under the Receiver Operating Characteristic Curve (AUC): AUC measures the model's ability to distinguish between accident and non-accident scenarios. Better performance is shown by a greater AUC, indicating a higher true positive rate (correctly predicting accidents) compared to false positives.
Time-to-Accident (TTA): TTA measures the timeliness of the model's accident predictions. It represents the time between the actual start of an accident and the model's first accurate forecast. A larger TTA value indicates earlier predictions, which is crucial for implementing preventive measures in real-world applications like autonomous driving systems.

5.1.1.2 Visual Attention Metrics (For DADA-2000 Dataset)

The predicted attention maps were compared with ground truth fixation data using several metrics to evaluate the DRIVE model's predictive capacity for human visual attention:

- **Similarity (SIM):** Measures spatial similarity between predicted and ground truth attention maps.
- **Linear Correlation Coefficient (CC):** Evaluates the linear relationship between the predicted and actual attention maps.

- **Kullback-Leibler Divergence (KLD):** Quantifies the difference between the ground truth fixation map distribution and the predicted attention map distribution.

5.1.2 Implementation

The DRIVE model was implemented using PyTorch. The saliency module used a VGG-16 architecture pre-trained with fixation data from DADA-2000. Videos were scaled and zero-padded to a resolution of 480x640.

5.1.3 Results

Metric	**SAC** ([17])	**TD3** (Our Model)
mTTA@0.5 (Mean Time to Anticipation)	3.9857 seconds	4.1861 seconds
AP (Average Precision)	0.5029	0.6084
Precision@0.5	0.4902	0.4928
Recall@0.5	0.8278	0.9794
v-AUC (Visual Area Under Curve)	0.66566	0.60833
f-AUC (Visual Area Under Curve)	0.54744	0.84689
Fixation MSE (Mean Squared Error)	136.6031	119.6025

Table 5.1: Accident Anticipation Results for SAC and TD3 using DADA-2000 Dataset

5.1.3.1 Results Analysis

The results of our experiment comparing SAC and TD3 within the DRIVE framework using the DADA-2000 dataset highlight key trade-offs in model performance.

In terms of **earliness**, the Mean Time to Anticipation (mTTA@0.5) for our model (4.1861 seconds) is slightly higher than SAC (3.9857 seconds), indicating a minor delay in accident anticipation. However, in terms of **correctness**, TD3 achieves higher Average Precision (AP = 0.5029) and Precision@0.5 (0.4902) compared to TD3 (AP = 0.6084, Precision@0.5 = 0.4928). This suggests that TD3 provides more confident accident predictions or it can be said that it is neck to neck with SAC. Despite this, TD3 outperforms SAC in **recall** (0.8601 vs. 0.9794), showing that TD3 captures more positive accident cases but at the cost of increased false positives.

For **attention-based performance**, SAC achieves a marginally better v-AUC (0.66566 vs. 0.60833) but TD3 achieves higher f-AUC (0.54744 vs. 0.84689), implying that TD3 attention mechanisms are on par with human fixation patterns. However, the difference is minimal, indicating that TD3 can still provide a reliable visual attention model.

In terms of **fixation accuracy**, the Fixation Mean Squared Error (Fixation MSE) for TD3 (119.6025) is slightly lower than SAC (136.6031), suggesting that TD3 can produce slightly more stable fixation predictions. The observed differences in fixation MSE can be attributed to the deterministic nature of TD3, which lacks the entropy-driven exploration of SAC.

Overall, while SAC demonstrates superior precision and fixation accuracy, TD3 provides stronger recall and a competitive fixation performance. The higher recall of TD3 makes it more effective for accident anticipation in scenarios where capturing all potential risks is more critical than minimizing false positives. These results validate the efficacy of TD3 as a stable alternative to SAC, particularly in computationally constrained environments where deterministic policy optimization offers advantages in sample efficiency and hardware compatibility.

5.1.4 Graph Analysis

The graphs from Section 5.1.5 demonstrate the difference in performance between our model and the predecessor model using SAC. We use a number of different comparisons to track the performance of our model, as seen in the graphs. Through these graphs, we gain valuable insight into the effectiveness of TD3 in this research. This section provides a brief explanation of the analysis and comparison of trends observed in key metrics such as loss functions, policy performance, and fixation prediction accuracy.

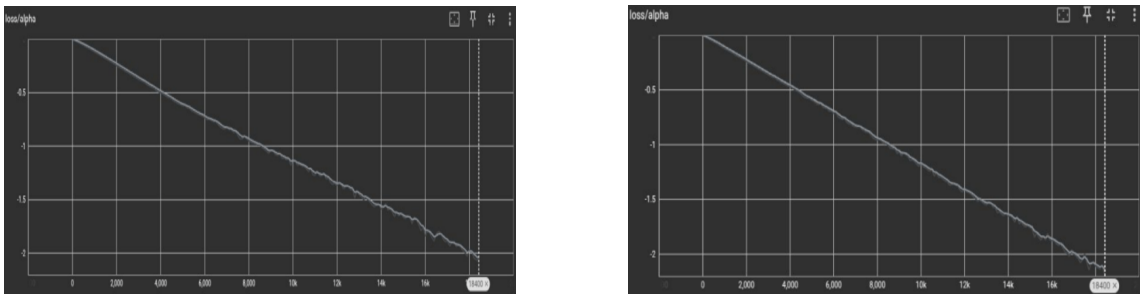
5.1.4.1 Comparison between the Trends in Loss Function

Using the loss functions, we can obtain a good measure of the model's learning progress and ability. Among the loss functions, we have displayed alpha, autoencoder, critic, and policy networks:
Alpha Loss: Both graphs display a decreasing alpha loss, which indicates that both models achieve a proper balance of exploration-exploitation at each step. However, the TD3 model ends up with a smaller value of the alpha parameter.
Autoencoder Loss: TD3 displays a steeper descent compared to SAC, highlighting its efficiency in encoding visual attention-based representations.
Critic Loss: The critic loss for SAC is very noisy and unstable. While our TD3 model also displays some instability, its loss descends from the start, indicating that the TD3 critic is learning to minimize its loss.
Accident Policy Loss: Both losses are not smooth and fluctuate during training, showing the difficulty of training both models. However, TD3 is slightly more stable than SAC.
Actor Policy Loss: The actor loss decreases from the start for SAC, ending up with a high negative value, whereas the loss for TD3 is mostly increasing from the beginning to an almost zero value. An increasing value would generally be bad, but since TD3 is a deterministic model, an initially increasing trend is beneficial as it means the model is focusing on exploration.
Fixation Policy Loss: Both models display a very noisy loss throughout the training loop, requiring further evaluation to ensure the model can better predict the next fixation point.

5.1.4.2 Trends in Total Policy Graphs for TD3

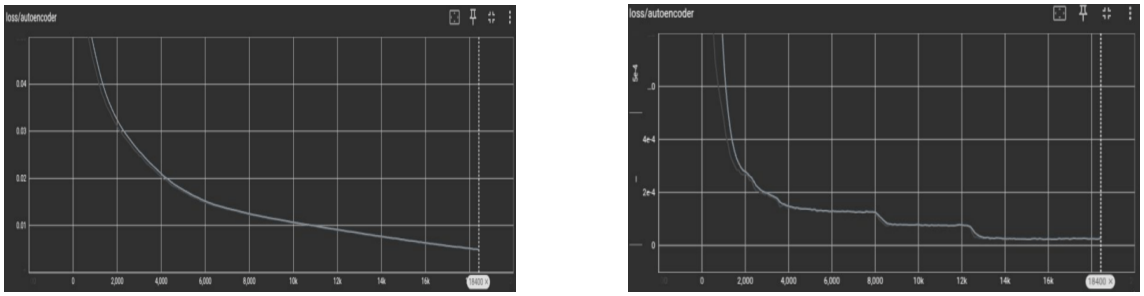
Total Accident Policy Loss: The loss converges towards zero with some minor instability, indicating that the model is using patterns to minimize the loss.
Total Fixation Policy Loss: The graph shows an increasing trend in policy loss. This suggests that the TD3 model may be struggling to learn effectively in this task, possibly due to high variance in policy updates or inadequate exploration.

5.1.5 Graphs



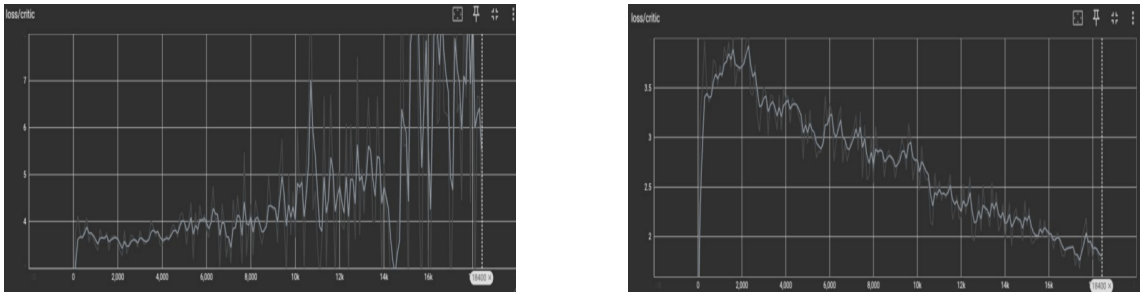
(a) SAC (b) TD3

Figure 5.1: Comparison of loss/alpha



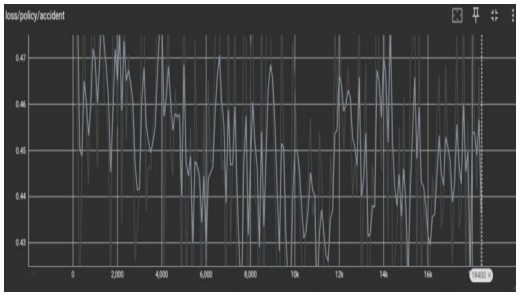
(a) SAC (b) TD3

Figure 5.2: Comparison of loss/autoencoder

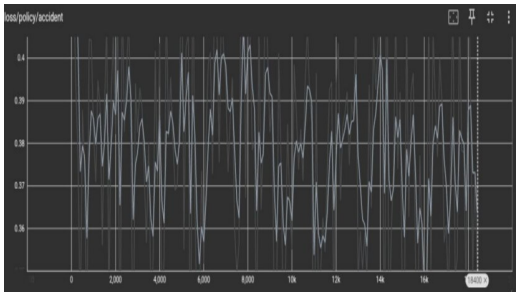


(a) SAC (b) TD3

Figure 5.3: Comparison of loss/critic

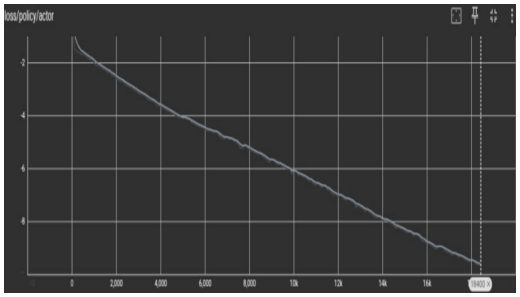


(a) SAC

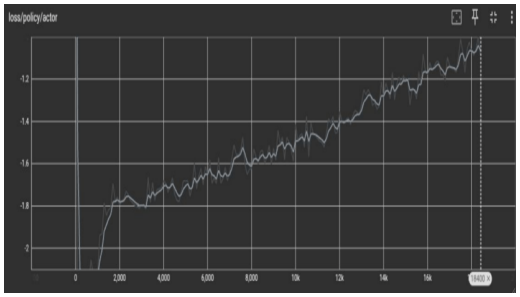


(b) TD3

Figure 5.4: Comparison of loss/policy/accident

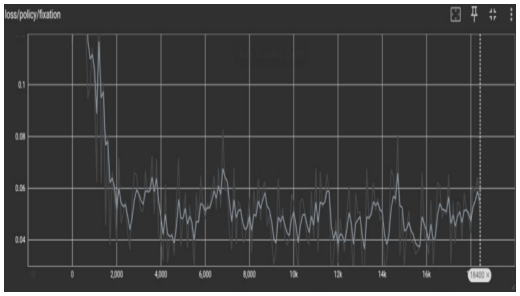


(a) SAC

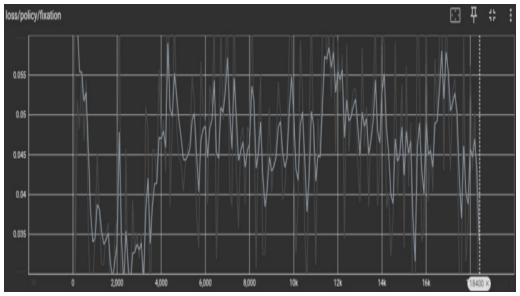


(b) TD3

Figure 5.5: Comparison of loss/policy/actor



(a) SAC



(b) TD3

Figure 5.6: Comparison of loss/policy/fixation

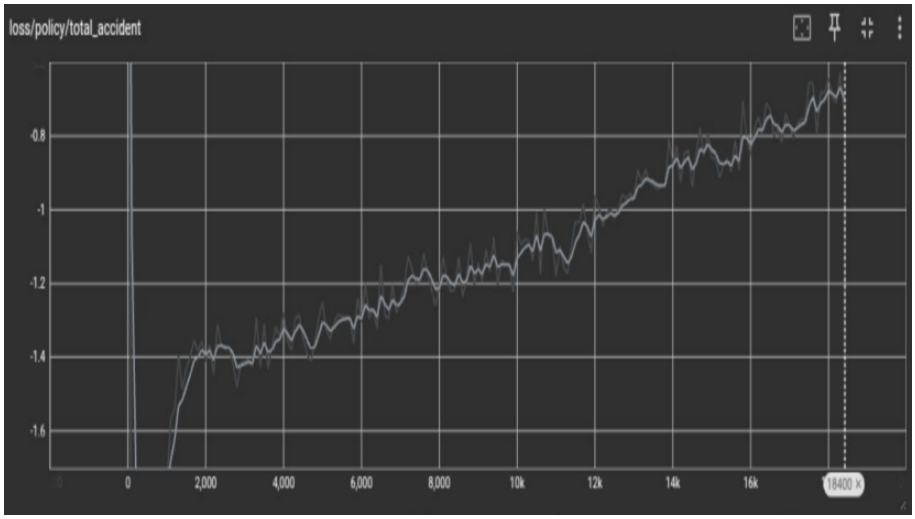


Figure 5.7: loss/policy/total_accident for TD3

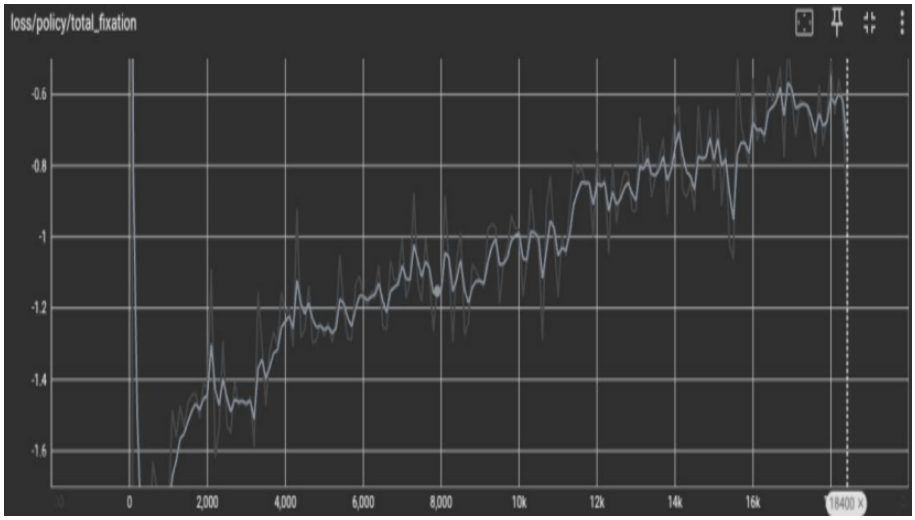


Figure 5.8: loss/policy/total_fixation for TD3

Chapter 6

Conclusion

6.1 Conclusion

The proposed model works toward an essential step that is immensely needed: safer and more dependable autonomous vehicles. The main strength of the model is in overcoming the limitations of existing supervised learning approaches by jointly addressing visual attention and accident prediction in a unified framework. The latter explains the prediction of driving error in the model. This state-of-the-art model deals with datasets based on real-world traffic accidents, thus indicating the potential of the model to aid in improving the safety and reliability of autonomous systems for driving. Moreover, visual attention affords an explanatory mechanism for the model's decisions, increases transparency, and engenders trust in it. Explainability is at the core of the drive for public acceptance and adoption of AVs; through visualization, it will be afforded a deep understanding of the model's reasoning for debugging and improvement.
Simultaneous acknowledgment of potential challenges and risks is instrumental in ensuring the system's strength and generalizability in real-world driving scenarios. Future studies should focus on:

- **Annotating Datasets:** Expanding and diversifying the size of datasets will gain importance in training DRIVE with a broader scope of driving conditions, types of accidents, or driver behaviors. It will generalize to real life and get enhanced, thus reducing bias. Future work could also focus on using datasets from countries where driving environment is more unpredictable such as countries in South-East Asia to better verify the model's ability to anticipate.

- **Other DRL Algorithms:** The SAC and TD3 algorithm are quite promising, but different algorithms within the DRL family can potentially have better performance and stability in addition to increased sample efficiency. Even more research on holistic and multi-task-centric algorithms, coupled with a focus on safe-critical applications, is needed.

- **More Sensor Data:** As of now, the model makes use of information from the dashcam video only. More sensory information around the event, including LiDAR, radar, and vehicle dynamics data, would restructure the model's

perception concerning prospective accidents. Multi-modal approaches are inherently more robust and, hence, bound to provide better results in terms of prediction reliability.

- **Develop More Sophisticated Visual Attention Models:** While current models of visual attention have been demonstrated to suffice, they may not capture all the nuances and complexities in allocating human attention. Future studies could explore the potential of transformer-based models, which are known for their ability to model long-range dependencies and handle more intricate relationships in sequential data. Additionally, incorporating more factors such as drive experience, emotional state, and cognitive load could result in more realistic and accurate simulations that mimic human attention more effectively.

- **Inherent Visual Attention Biases:** Great care has to be taken so that there are no intrinsic biases in the visual attention model that restrict the same from portraying what would otherwise become a complete array of human attention patterns. Addressing this potential bias will require a complete understanding of human visual perception, coupled with more deliberate selection and training for the attention model.

Overall, this research lays a fundamental foundation for realizing a future in which autonomous vehicles will get into the mainstream transportation systems with improved safety, efficiency, and accessibility to all users. Building stronger, explainable, adaptable systems in AI systems can get us closer to that vision and set the way for a future of safer and more sustainable mobility.

Bibliography

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] W. S. Geisler and J. S. Perry, "Real-time foveated multiresolution system for low-bandwidth video communication," in *Human Vision and Electronic Imaging III*, vol. 3299, 1998, pp. 294–305.

[3] C. E. Connor, H. E. Egeth, and S. Yantis, "Visual attention: Bottom-up versus top-down," *Current Biology*, vol. 14, no. 19, R850–R852, 2004.

[4] V. Mnih, N. Heess, A. Graves, *et al.*, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[5] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2016.

[6] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2016.

[7] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," in *International Conference on Pattern Recognition (ICPR)*, 2016.

[8] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao, "Learning to predict sequences of human visual fixations," *IEEE Transactions on Neural Networks and Learning Systems (IEEE TNNLS)*, vol. 27, no. 6, pp. 1241–1252, 2016.

[9] V. Mnih, A. Badia, M. Mirza, *et al.*, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*, 2018.

[12] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 11, pp. 2693–2708, 2018.

[14] K. Min and J. J. Corso, "Tased-net: Temporally aggregating spatial encoder-decoder network for video saliency detection," in *ICCV*, 2019.

[15] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data - ai integration perspective," *IEEE Access*, vol. 7, 160
bibrangessep 187–160
bibrangessep 204, 2019.

[16] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2682–2690.

[17] W. Bao, Q. Yu, and Y. Kong, "Drive: Deep reinforced accident anticipation with visual explanation," *arXiv preprint arXiv:2107.10189*, 2021.

[18] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.

[19] Z. Emam, A. Kondrich, S. Harrison, *et al.*, "On the state of data in computer vision: Human annotations remain indispensable for developing deep learning models.," *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 3399–3408, 2021.

[20] J. Lyu, X. Ma, J. Yan, and X. Li, "Efficient continuous control with double actors and regularized critics," *arXiv preprint arXiv:2106.03050*, 2021.

[21] I. Cho, P. K. Rajendran, T. Kim, and D. Har, "Reinforcement learning for predicting traffic accidents," *arXiv preprint arXiv:2212.04677*, 2022.

[22] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *arXiv preprint arXiv:2303.08246*, 2023.