

Estimating Object Importance and Modeling Drivers' Situational Awareness for Intelligent Driving

Pranay Gupta

CMU-RI-TR-24-52

08-01-2024



The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:

Professor David Held, *co-chair*
Professor Henny Admoni, *co-chair*
Professor Andrea Bajcsy,
Ravi Pandya

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.*

To my mother, and father.

Abstract

The ability to identify important objects in a complex and dynamic driving environment helps assistive driving systems decide when to alert drivers. These assistance systems also require a model of the drivers’ situational awareness (SA) (what aspects of the scene they are already aware of) to avoid unnecessary alerts. This thesis builds towards such intelligent driving assistance systems, we first address the problem of estimating the importance of objects in a driving scenario. Next, we propose a model for estimating a driver’s awareness of objects in the scene.

We tackle object importance estimation in a data-driven fashion and introduce HOIST - Human-annotated Object Importance in Simulated Traffic. HOIST contains driving scenarios with human-annotated importance labels for vehicles and pedestrians. We additionally propose a novel approach that relies on counterfactual reasoning to estimate an object’s importance. We generate counterfactual scenarios by modifying the motion of objects and ascribe importance based on how the modifications affect the ego vehicle’s driving. Strong performance on importance estimation, and an explainable nature make it ideal for driving assistance.

Next, we tackle driver’s situational awareness (SA) estimation. Situational Awareness is an internal human cognitive state, hence acquiring ground-truth awareness labels is challenging. We first propose a novel interactive labeling protocol that effectively captures dense, continuous SA labels and use it to collect an object-level SA dataset in a VR driving simulator. We formulate object-level driver SA prediction problem as a semantic segmentation problem. This formulation allows all objects in a scene at a timestep to be processed simultaneously, leveraging global scene context and local gaze-object relationships together. Our experiments show that this formulation leads to improved performance over common sense baselines and prior art on the SA prediction task.

Acknowledgments

To begin with I would like to express my gratitude towards my advisors, Prof. David Held and Prof. Henny Admoni for closely guiding me on throughout this journey. Their advice not only helped me become a better researcher but also helped me develop a strong moral and ethical character. I would also like to thank Abhijat Biswas for the enlightening discussions.

I would like to specially thank my labmates both from HARP lab and from the RPAD lab for being my companions throughout my time as a MSR student. I would also like to thank my friends Anish Madan, Dvij Kalaria, Pushkal Katara, Poorvi Hebbbar, Sarthak Bhagat, Shagun Uppal and Shreya Sharma who made my life in Pittsburgh extremely fun, providing the much needed balance. Study sessions with them helped me balance my courses and research. They have had a huge impact on cultural tastes, and I have been extremely fortunate to have them as my friends with whom I have shared some wonderful memories over the years.

Finally I would like to thank my family for their never-ending support and motivation. Their trust in me has allowed me work on my research without any worries and their encouragement in times of distress and failure has helped me get back up on my feet.

This thesis would not have been possible without the support of many people, and I would like to dedicate this work to all of them, and I hope to have their support in my future endeavours as well.

Funding

I gratefully acknowledge the financial support from multiple funding sources, which made this research possible. Specifically, I would like to acknowledge the support from an NSF award (IIS-1943072) and a Google Research Award.

Contents

1	Introduction	1
2	Object Importance Estimation using Counterfactual Reasoning for Intelligent Driving	5
2.1	Introduction	5
2.2	Related Works	8
2.3	HOIST: Human-annotated Object Importance in Simulated Traffic	9
2.3.1	Dataset Curation	10
2.3.2	Human Annotations	10
2.3.3	Dataset Statistics	11
2.4	Estimating Object Importance	11
2.4.1	Vehicle Importance Estimation	12
2.4.2	Pedestrian Importance Estimation	15
2.4.3	Implementation Details	16
2.5	Experiments	17
2.5.1	Comparison with Baselines	17
2.5.2	Ablations	19
2.5.3	Importance Prediction for Vehicles and Pedestrians	21
2.5.4	Failure Cases	22
3	Modeling Drivers' Situational Awareness from Eye Gaze for Driving Assistance	25
3.1	Introduction	25
3.2	Situational Awareness Data Collection	28
3.3	Modeling Driver SA	30
3.4	Evaluation & Discussion	32
4	Conclusion and Future Work	37
	Bibliography	39

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

2.1	We perform counterfactual reasoning to identify important vehicles in a driving scenario. We modify the motion of vehicles, and ascribe importance based on how the modification affects the ego vehicle's driving.	6
2.2	The removal score is the L2 distance between the corresponding waypoints of the true trajectory (shown in a) and the predicted trajectory after removing Car 0 (shown in b).	12
2.3	The velocity perturbation score assesses potential collision, when the non-ego vehicle undergoes a velocity perturbation. A collision occurs when the corresponding waypoints of the ego vehicle and the non-ego vehicle trajectories are within a threshold. The collisions occur at the k^{th} waypoint. The trajectories of the ego vehicle and the non-ego vehicle are shown in orange and green, respectively.	13
2.4	Qualitative comparison of our approach with the baselines. The objects are marked as important based on the optimal threshold for the F1 Score for each approach.	18
2.5	Failure cases of our method.	23
3.1	We collect drivers' object-level situational awareness (SA) data via a novel interactive protocol in a VR driving simulator. We use the generated data to train a driver SA predictor from visual scene context and driver eye gaze. Casting this as a semantic segmentation problem allows our model to use global scene context and local gaze-object relationships together, processing the whole scene at once regardless of the number of objects present.	26
3.2	Example sequence of right hand turn with object-level driver responses. The top row shows the scene from the driver view and the bottom row shows the same scene via a birds-eye view. Labels are shown as colored arrows above the respective traffic object. Labels correspond to buttons on the steering wheel (right). Blue corresponds to vehicle labels and red to pedestrian labels.	29

3.3	Object-wise SA prediction algorithm. A history of raw driver gaze is filtered to exclude saccades and then transformed to 2D pixels in the current camera position. These are used to create a gaze history map which is input together with an object segmentation of the scene (or optionally, RGB). The Feature Pyramid Network (FPN) then produces a 3 class segmentation (unaware, aware, background). During training, loss is ignored for objects which entered into the driver awareness prior to the gaze history window.	30
3.4	Qualitative results for our model and baselines. Each row represents an independent driving scene. The RGB image shows the most recent 10s of gaze overlaid as red dots.	35

List of Tables

2.1	Our approach outperforms the baseline approaches with a considerable margin on all metrics.	19
2.2	Ablation experiments demonstrate the impact of the various components and design choices of our method.	20
2.3	Comparison of our approach with the baseline approaches for <i>vehicle</i> importance estimation. Our approach outperforms the baseline approaches across all 3 metrics	21
2.4	Comparison of the baseline approaches for <i>pedestrian</i> importance estimation. $\mathcal{IS}$ refers to our approach for <i>vehicle</i> importance estimation (Sec. 2.4.1, Eqn. 2.3). Our approach outperforms all other approaches across all 3 metrics.	22
3.1	Prediction performance of models and baselines on the SA prediction task. Our model outperforms the non-trivial baselines on all 3 metrics and scales better as objects in the scene increase. [inf. cpmlx. = inference time complexity with N objects, Acc. = Accuracy, Prec. = Precision]	33

Chapter 1

Introduction

Driver inattention contributes to a significant number of traffic accidents. For example, the U.S. National Highway Traffic Safety Administration (NHTSA) reports that in 2022, approximately 3,308 people lost their lives in motor vehicle crashes involving distracted drivers in the United States alone [37]. Advanced Driver Assistance Systems (ADAS) are designed to assist drivers in safe driving. ADAS operating at the lowest-levels of driving automation can alert drivers in safety-critical scenarios using proximity/range sensors, surround cameras etc.

Recent times have seen a surge of commercial vehicles with ADAS operating at Level 2-3 of driving automation. At these levels, the vehicle takes most of the driving decisions and the human driver here plays a more supervisory role in driving the vehicle, only intervening when necessary. However, prior research [12, 22] has shown that drivers performing a supervisory role are more prone to distractions. A distracted driver could thus prove disastrous in case they need to take over control of the wheel since they might not be aware of a crucial vehicle or pedestrian in the environment. Therefore, there is a pressing need to have systems that can identify important objects in the scene, which can be used to alert drivers when necessary.

Further, naively alerting the driver to all important objects in the scene can be counter-intuitive, due to effects like alert-fatigue. Prior work [60] has also shown that the presence of multiple important objects can cause a drop in the driver's Situational Awareness. Monitoring the driver's awareness of objects can help reduce obtrusive and redundant alerts pertaining to objects that the driver is already aware of. Thus,

modeling a driver’s situational awareness is critical for effectively alerting the driver to important objects in the scene.

This thesis, aims to address these two problems. We first study object importance estimation in a driving scenario. Next, we aim to model the driver’s situational awareness, to estimate the driver’s awareness status of objects in the scene.

Driving assistance systems, which monitor the driver to filter unnecessary alerts are not new. Fletcher *et al.* [20] propose a system that detects road signs, estimates the driver’s awareness and provides feedback accordingly. Road signs are detected by locating circular shapes in the input-image stream. Driver awareness is estimated based on the deviation between the direction of the sign and the gaze direction. However, they do not cull scene objects based on importance. Moreover, they also assume that the driver gains awareness of the object whenever the deviation between the raw gaze direction and the sign direction is within the tolerance limits. Bar *et al.* [4] first identify all traffic objects that were missed by the driver, using the gaze fixations of the driver. They further classify the missed objects into important and unimportant. The object importance is estimated based on the gaze patterns from previous drives with comparable traffic context. Similar to Bar *et al.*, Wu *et al.* [54] also relies on eye-gaze fixations to estimate awareness.

Building towards a system similar to the one proposed in prior literature, this thesis first addresses the problem of identifying important objects in a driving scenario. The goal is to identify all objects that the ego vehicle needs to be aware of for safe decision making. Our first contribution in this direction is a novel dataset called **HOIST**: Human-annotated Object Importance in Simulated Traffic. HOIST contains driving scenarios with human labeled important objects. Additionally, it contains data from all the sensor configurations offered by CARLA [14]. Next, we propose a novel approach that performs counterfactual reasoning to estimate the importance of objects in the scene. Our approach perturbs the motion of objects and compares how it changes the ego vehicle’s behaviour, predicted using an autonomous driving agent to ascribe importance. In addition to it’s strong performance on HOIST, its explainable nature also makes it ideal for driving assistance.

Next, we propose a novel method for modeling a driver’s situational awareness. The goal is to estimate a driver’s awareness of all objects in the scene. Our first contribution comes in the form of a novel protocol for measuring the driver’s ground-

truth awareness of objects in the scene. This protocol enables us to curate a large scale driving dataset, with the driver’s eye gaze and explicitly labeled ground-truth object awareness. Using this dataset we train a model that relies on scene information and the driver’s eye-gaze to predict object level awareness.

1. Introduction

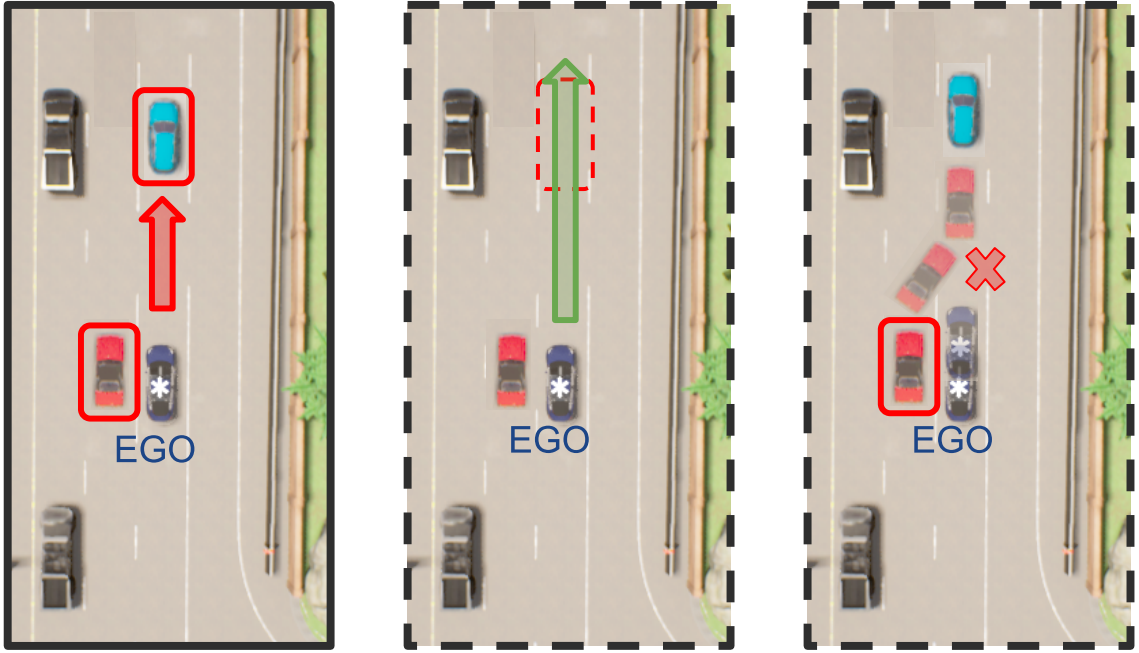
Chapter 2

Object Importance Estimation using Counterfactual Reasoning for Intelligent Driving

2.1 Introduction

Assistive and autonomous driving agents need to reason about objects within the driving scene to make safe driving decisions. However, not all objects are equally important. The ability to identify important objects would allow autonomous driving systems to perform efficient path planning and resource management by reducing the efforts spent on tracking and modeling. Moreover, prior work has also shown that importance estimation can help reduce false positive alerts in assistive driving systems that detect driver inattention and preemptively warn the driver (*e.g.*, detecting a jaywalking pedestrian and highlighting them on a HUD) [54]. Recent work has also shown improvement in autonomous driving agents' performance when trained to de-prioritize unimportant objects [6]. Thus, in this work, we focus on the task of important object identification in the context of driving.

In prior works [3, 18, 38, 39, 56] researchers have tried to identify salient regions in a driving scene by modeling driver attention through eye-gaze fixations. However, in the context of driving, the correlation between gaze and attention is questionable [28] (see



(a) Red boxes indicate vehicles important for making safe driving decisions by the ego vehicle.

(b) The blue car is important since its removal would cause the ego vehicle to speed up.

(c) The red vehicle is important since its changing lanes would risk a collision with the ego vehicle.

Figure 2.1: We perform counterfactual reasoning to identify important vehicles in a driving scenario. We modify the motion of vehicles, and ascribe importance based on how the modification affects the ego vehicle’s driving.

Sec. 2) as the driver may not always look at important objects due to distractions [52], inexperience [11], or paying attention via peripheral vision [5].

Prior works [21, 34, 58] also try to identify the importance of objects within the scene directly. However, their definition of “important objects” is limited to objects that influence the driving behavior of the ego vehicle. We extend this definition of “important objects” to include all the objects that a driver should be aware of, in order to drive the ego vehicle safely (Sec. 2.3.2) *i.e.*, all objects necessary to maintain sufficient situational awareness [17]. This includes objects that both currently influence and *can potentially influence* the driving decisions of the ego vehicle. This definition allows us to be proactive rather than reactive for identifying hazardous objects. This is ideal for alert based assistive driving systems [54], where its necessary to be ahead of objects and prepare the driver in case the objects act

abnormally.

Moreover, the object-importance datasets proposed in prior works [21, 58] are unimodal (only ego-view RGB videos) and do not benefit from multiple sensor modalities (multi-view RGB cameras, LiDAR) for object detection and importance estimation; further, these datasets are not publicly accessible. To address these challenges, we introduce HOIST: Human-annotated Object Importance in Simulated Traffic. HOIST consists of driving scenarios from representative routes of the Longest6 benchmark [10] with human-annotated object importance. While prior works [21, 34, 58] obtain annotations using ego-view driving videos, we collect annotations using bird’s-eye-view RGB videos of the driving scene, as the bird’s-eye-view offers more context regarding the driving scenario as compared to the front-view video. In addition to the RGB videos, HOIST also contains data from all sensor modalities and configurations supported by CARLA [14]

Our new dataset enables our second contribution: a novel approach for object importance estimation. We propose to determine an object’s importance by answering counterfactual questions about those objects. Specifically, we modify the motion of objects in the scene and generate counterfactual scenarios. We gauge the change in the ego vehicle’s driving under the counterfactual (estimated using an autonomous driving model [9]) to estimate each object’s importance.

We evaluate our approach on HOIST for object importance estimation. The results highlight the effectiveness of our approach in identifying important objects. We also perform extensive ablation studies, to validate our design choices and analyze the significance of the different components of our approach. The main contributions of our work are:

- HOIST, a novel dataset containing simulated driving scenarios with human-labeled important objects, which influence or can potentially influence the ego vehicle’s driving. In addition to RGB videos, HOIST contains data from all the sensors configurations offered by CARLA [14].
- A novel approach that employs counterfactual reasoning to estimate object importance in driving scenarios. We modify the motion of objects to generate counterfactual scenes and gauge the change in the ego vehicle’s behavior to ascribe importance.

- Benchmarking and ablation experiments on HOIST that substantiate the effectiveness of our approach in detecting important objects in a driving scene. Our code and dataset will be available on our project page¹.

This chapter is adapted from our work [24] published in IEEE Robotics and Automation Letters (RAL), 2024.

2.2 Related Works

Object Importance Estimation in on-road driving: The task of identifying important objects in the context of driving has garnered increasing attention from researchers. Prior approaches [21, 30, 58] learn to predict object importance using end-to-end models, leveraging the ego vehicle’s goal [21], interactions between objects [58], or the ego vehicle’s predicted behavior [34] to identify important objects. Since manual annotation is time intensive, Li *et al.* also performs semi-supervised learning by generating ranking based pseudo-labels for the unlabeled data. Instead of learning end-to-end models for estimating object importance, our approach relies on counterfactual reasoning.

Another line of works [30, 31], performs causal inference to identify risky objects. Similar to our approach, they verify whether the removal of objects produces a change in the ego vehicle’s driving decision. While these works only consider removal, we perturb other dynamic attributes of the objects as well. A third category of works attributes importance to objects using attention weights obtained from attention-based models trained for a different task, such as trajectory forecasting [32, 33, 48], or end-to-end driving [1, 40].

Driver’s Attention Prediction Methods: While driving, humans can selectively attend to relevant objects in the scene [55, 57]. Hence, prior works [3, 39, 56] tried to identify salient regions within a dynamic traffic scene by studying the driver’s attention. Here, the driver’s eye gaze serves as a proxy for the driver’s attention. For example, Palazzi *et al.* [39] proposes to use a multi-stream 3D convolutional network that leverages RGB frames, optical flow, and segmentation maps to predict the driver attention heatmap.

¹<https://vehicle-importance.github.io>

Bae *et al.* [3] models eye-gaze fixations as a sequence of states and actions and proposes maximum entropy deep inverse reinforcement learning [61] to predict maximally rewarding fixation locations. In our work, we directly estimate the importance of objects within the scene, rather than detecting the driver’s attention heatmaps.

Driver’s Attention Prediction Datasets: Typically, methods for driver attention prediction leverage large-scale driver attention datasets comprising ego-centric driving videos with the corresponding driver gaze [56]. DReyeVE [39] was the first large-scale driver attention dataset recorded using eye-tracking glasses in a car. Recent driver attention datasets [3, 18] have focused on accident scenarios. Pal *et al.* [38] argues that driver gaze is insufficient in describing everything a driver should attend to, and proposes to use ground truth gaze map combined with segmentation masks of all the objects within the scene. Unlike these datasets, HOIST consists of data from a host of sensor observations (RGB cameras, LIDAR, GPS) along with explicitly annotated object importance, rather than using the driver gaze as a proxy for importance.

2.3 HOIST: Human-annotated Object Importance in Simulated Traffic

We consider the problem of object importance estimation in the context of driving. We define “important objects” as the objects that the driver should watch out for in order to make safe driving decisions. Prior object importance estimation datasets [21, 34] limit important objects to only those which currently influence the ego vehicle’s driving decisions. Moreover, they only contain single-view RGB videos and are not publicly accessible. Transcending these challenges, we curate **HOIST**: Human-annotated Object Importance in Simulated Traffic. HOIST is a new dataset of driving scenarios with human annotations of object importance. In addition to RGB videos, HOIST also provides access to data from a host of different types of sensors, including multi-view cameras and LIDAR. These different sensors will allow users to use this dataset to evaluate different methods of object importance estimation, as discussed in Sec. 2.4.

2.3.1 Dataset Curation

We obtain the driving videos for HOIST from the Longest6 benchmark [10], which consists of routes from the CARLA [14] driving simulator. The Longest6 benchmark was introduced as a substitute for the official CARLA evaluation routes to overcome the compute restrictions set by CARLA. It consists of the six longest routes from each of the six towns in CARLA, resulting in a total of 36 routes. It preserves the most challenging aspects of the CARLA evaluation routes in terms of traffic density, weather conditions, and adversarial traffic scenarios. For our dataset, we select six representative routes (one from each town) of the Longest6 benchmark. We drive the ego vehicle through the six selected routes using a rule-based-expert [10], recording simulation data along each route.

We extract the bird’s-eye-view videos from the simulation recording of each route and split them into smaller 13 second videos. Unfortunately, the routes in the Longest6 benchmark majorly involve driving in a straight lane. Driving in straight lanes leads to trivial cases for object importance estimation as the objects closely leading and following the ego vehicle are generally deemed important. To ensure that we include more diverse driving scenarios, we limit the number of videos with straight lanes and include videos where the ego vehicle is undergoing complex driving maneuvers like crossing intersections, turning, changing lanes, or stopping for pedestrians. We end up with a total of 409 videos.

2.3.2 Human Annotations

We hire annotators on Prolific² to collect annotations for the videos obtained in the previous step (2.3.1). The annotators were first shown the bird’s-eye-view RGB videos and were then asked to draw bounding boxes around important objects in the last frame of the video. In our instructions, we defined important objects as the ones the annotators would watch out for in order to drive the ego vehicle safely if they were to take control of the ego vehicle from the point at which the video ended. Our instructions allow annotators to select multiple important objects in the driving scenario. We want to ensure that both objects that are important right now

²<https://www.prolific.co/>

as well as other objects that could be important in the near future are included in the dataset. Important object identification systems designed using our dataset would be useful in proactively alerting human drivers to be aware of all important objects. We instructed the annotators to focus on only two types of objects: vehicles and pedestrians.³

Our annotators reported that they are more than 18 years of age and have a valid US state-issued driving license. Each annotator did three practice rounds, two at the beginning and one at the end. These practice rounds were straightforward and were used to exclude annotators whose annotations demonstrated a lack of comprehension of the task. We hired 188 annotators and each annotated 15 videos. To mitigate the bias due to the driving habits of individual annotators, each video was annotated by at least 5 different annotators. The fleiss' kappa [19] inter-rater reliability between the annotators for the entire dataset was 0.558. However, to remove the outliers from the annotations we consider an object important in the ground truth data if it is marked important by at least θ_1 annotators. Objects marked important by fewer than θ_2 annotators are considered unimportant. We set $\theta_1 = 3$ and $\theta_2 = 2$. We end up filtering 10.68 % objects. The inter-rater reliability after filtering the outlier annotation is 0.657, showing substantial agreement according to [29].

2.3.3 Dataset Statistics

The HOIST dataset consists of a total of 409 videos with 642 unique human-annotated important objects. Out of these 409 videos, 8.8% of the videos have 0 important objects, 44.9% have 1 object, 32.2% have 2 important objects, and 13.9% have more than 2 important objects. In this work, we only consider vehicles and pedestrians. Out of the 642 total important objects, 597 are vehicles and 45 are pedestrians.

2.4 Estimating Object Importance

In this section, we describe our novel method for estimating the importance of objects for making safe control decisions for the ego vehicle. Because the movement patterns

³Details about the interface and the prompts used to obtain the annotations are available on our project page : <https://vehicle-importance.github.io>

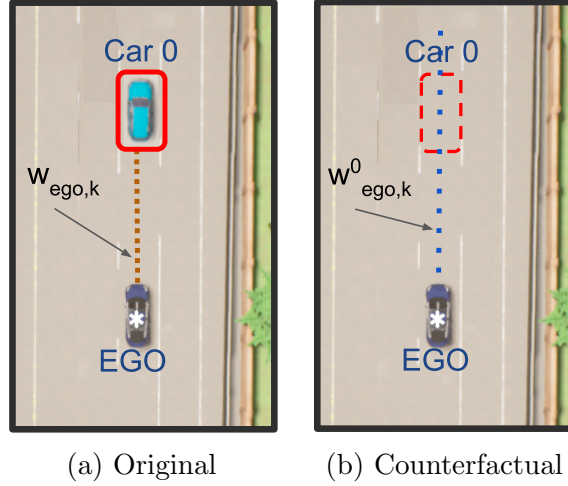


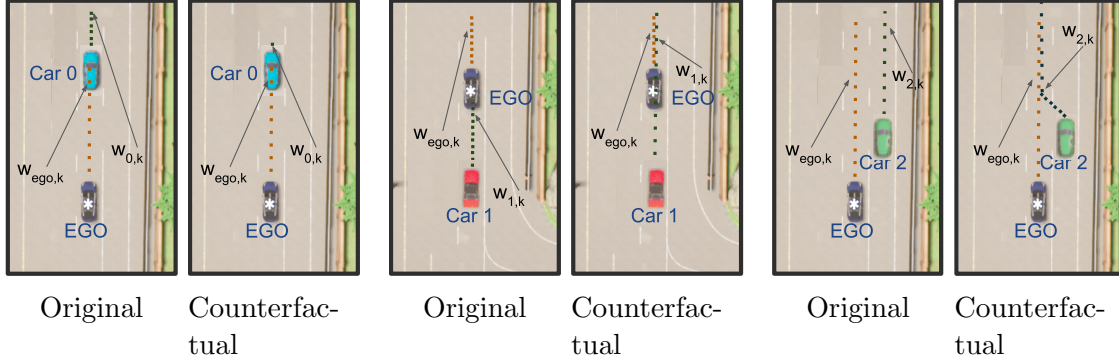
Figure 2.2: The removal score is the L2 distance between the corresponding waypoints of the true trajectory (shown in a) and the predicted trajectory after removing Car 0 (shown in b).

of vehicles and pedestrians are different from each other, we propose to use different approaches to estimate their importance. For estimating the importance of vehicles, we perform counterfactual reasoning about the vehicles (Sec 2.4.1). For pedestrians, we find their proximity to the ego vehicle to be a very strong cue for importance estimation (Sec 2.4.2). We explain both of these approaches in detail in the following sections.

2.4.1 Vehicle Importance Estimation

For estimating vehicle importance, we perform counterfactual reasoning. We generate counterfactual scenarios by modifying the motion of the non-ego vehicles; we ascribe importance based on how the modifications affect the ego vehicle’s driving. We modify the ego vehicle’s sensor observations to simulate modifications in the motion of the non-ego vehicles. We then compare the changes in the ego vehicle’s trajectory under the counterfactual. If the ego vehicle’s trajectory changes significantly under the counterfactual, the vehicle is deemed important (details below). We also perform collision estimation under the counterfactuals to ascribe importance.

We leverage autonomous driving systems [9] to predict the ego vehicle’s trajectory. Typical autonomous driving systems [9, 10, 42, 43, 53] perceive a scene through various sensor modalities (e.g. multiview RGB cameras and LIDAR) and predict the



(a) **Hard-Stop** perturbation. (b) **Speed-Up** perturbation. (c) **Lane-Change** perturbation.

Figure 2.3: The velocity perturbation score assesses potential collision, when the non-ego vehicle undergoes a velocity perturbation. A collision occurs when the corresponding waypoints of the ego vehicle and the non-ego vehicle trajectories are within a threshold. The collisions occur at the k^{th} waypoint. The trajectories of the ego vehicle and the non-ego vehicle are shown in orange and green, respectively.

future trajectory for the ego vehicle. A trajectory is represented by a set of waypoints. We query such a system with the true and the counterfactual sensor observations and compare changes in the predicted waypoints. We predict importance using two counterfactual scenarios:

Removal: To check if a vehicle affects the ego vehicle’s driving decisions, we verify whether removing that vehicle from the scene would cause the ego vehicle to change its trajectory. We remove vehicles by modifying the sensor observations input to the autonomous driving system (See Section 2.4.3 for details). Consider the example shown in Fig. 2.2. The importance of an object is calculated as the L2 distance between the waypoints predicted using the true sensor observations (Fig. 2.2a) and the counterfactual observations (Fig. 2.2b). We call this the Removal Score ($\mathcal{RS}$). Let $w_{ego,k}$ be the k th predicted waypoint of the ego vehicle with the true sensor observations, and let $\hat{w}_{ego,k}^i$ be the k th predicted waypoint of the ego vehicle under the counterfactual observation generated by removing the i th vehicle. The Removal Score $\mathcal{RS}_i$ for the i th vehicle is

$$\mathcal{RS}_i = \sum_{k \in \{1, 2, \dots, K\}} \|w_{ego,k} - \hat{w}_{ego,k}^i\|^2 \quad (2.1)$$

where K is the total number of predicted waypoints.

Velocity Perturbations: The Removal Score identifies vehicles that affect the ego vehicle’s trajectory in the current state. However, a vehicle can become a safety hazard if it suddenly changes its speed or direction. Thus, we check whether changing the velocity of a particular vehicle would make it susceptible to collisions with the ego vehicle (see Fig. 2.3). We simulate velocity changes by perturbing the trajectories of the vehicles. Similar to the ego vehicle, we obtain the trajectories for non-ego vehicles using an autonomous driving agent. Recent methods for autonomous driving predict the trajectories of non-ego vehicles in the scene either by using deep models [9] or by projecting it using a constant velocity model [42], where the constant velocity is calculated using a moving average over the past timesteps.

We consider three possible velocity perturbations. To identify vehicles that would become safety critical if they suddenly stopped, we generate a counterfactual scenario in which the vehicle makes a **hard stop**. To simulate hard stopping, we collapse all the waypoints of the vehicle into the first waypoint (See Fig. 2.3a). Next, we identify the vehicles that would become safety critical if they suddenly **sped up**. For this, we generate a counterfactual scenario where we increase the vehicle’s velocity. We simulate this by increasing the distance between consecutive waypoints of the vehicle’s trajectory (See Fig. 2.3b). Finally, to identify vehicles that would become safety critical if they suddenly **changed lanes**, we simulate lane change maneuvers for each object. We perturb the trajectories of the vehicles such that they first perform a 45-degree lane shift into an adjacent lane (left or right) and then follow a straight line (See Fig. 2.3c).

Ego Vehicle Perturbations: In addition to the non-ego vehicles in the scene, the ego vehicle might also have to deviate from its predicted trajectory under emergencies. Therefore, as a defensive driving measure, we also perturb the ego vehicle’s trajectory according to the three velocity perturbations. Simultaneously perturbing the ego vehicle with the non-ego vehicles also allows us to simulate combinations of perturbations (*e.g.*, the ego vehicle changes its lane while a non-ego vehicle in the adjacent lane speeds up.)

We consider a vehicle important due to velocity-based perturbations if it is predicted to collide with the ego vehicle. A collision is predicted if a waypoint from any of the non-ego vehicle’s trajectories (predicted or perturbed) is within the safety

threshold distance of a waypoint from the ego vehicle’s trajectories (predicted or perturbed). Moreover, a predicted collision that happens sooner is considered more important than a predicted collision that happens later. Hence, we score vehicle importance based on how soon they are predicted to collide with the ego vehicle. Thus, the Velocity Perturbation Score ($\mathcal{VS}_i$) for the i th object is given by

$$\begin{aligned} d_{i,k} &= ||w_{i,k} - w_{ego,k}||^2 \\ k_i^* &= \underset{k \in [0, \dots, K-1]}{\operatorname{argmin}} d_{i,k} \\ \mathcal{VS}_i &= \begin{cases} -k_i^* & \text{if } d_{i,k_i^*} < \tau \\ -K & \text{otherwise} \end{cases} \end{aligned} \quad (2.2)$$

Here, $w_{ego,k}$ is the k th waypoint of the predicted (or perturbed) ego vehicle’s trajectory. Similarly, $w_{i,k}$ is the k th waypoint of the i th non ego vehicle’s trajectory. K is the total number of waypoints; k_i^* is the index of the waypoint at which the ego vehicle’s trajectory is predicted to be closest to the i th vehicle. If this distance is less than the safety threshold τ , then we assign to the vehicle a score of $-k_i^*$, i.e. a closer index of collision corresponds to a higher score.

Combining the scores: The final importance score $\mathcal{IS}_i$ for a vehicle i is given by the maximum of the two scores:

$$\mathcal{IS}_i = \max(\mathcal{RS}_i, \mathcal{VS}_i) \quad (2.3)$$

Here, $\mathcal{RS}_i$ is the removal score for the i th vehicle, and $\mathcal{VS}_i$ is the velocity perturbation score for the i th vehicle. Both scores are normalized between 0 and 1 using their maximum and minimum values across the dataset.

2.4.2 Pedestrian Importance Estimation

In the HOIST dataset, the movement patterns of pedestrians are significantly different from vehicles. Vehicle movement is confined to the roads, and they typically adhere to traffic regulations. Pedestrians have more freedom in their movement. They sometimes violate traffic regulations and can even dart out onto the roads. Hence, the mechanisms for estimating importance of vehicles do not necessarily apply to

pedestrians. Since pedestrian movement is less constrained, their distance to the ego vehicle is a strong indication for importance. Hence, we ascribe importance to pedestrians based on their L2-distance from the ego vehicle. The importance score ($\mathcal{PS}_p$) for a pedestrian p is given by:

$$\mathcal{PS}_p = -||x_p - x_{ego}||^2 \quad (2.4)$$

Here, x_p and x_{ego} are the locations of the pedestrian and the ego vehicle, respectively.

2.4.3 Implementation Details

We use LAV [9] to predict the ego vehicle trajectory in our experiments. LAV uses LIDAR point clouds as input for predicting the trajectories of the vehicles. Thus, the counterfactual observations for the Removal Score ($\mathcal{RS}$) are created by removing the points within the 3D bounding box of the concerned objects. We obtain ground-truth bounding boxes from the simulator, although a 3D bounding box detection module [50] can also be used. The average time taken to generate counterfactual point clouds (by removing 8 vehicles on avg) for one frame is 0.068 seconds.

We query LAV with the counterfactual point clouds to generate the predicted trajectories for the ego and non-ego vehicles. We quash the counterfactual point clouds in a single batch, and predict all the trajectories in a single forward pass which takes 0.127 seconds per frame on average (typically 8 counterfactual point clouds). Overall our method takes 0.194 seconds per frame on average. We test our method on a single NVIDIA RTX 3070 GPU. In addition to the ego vehicle trajectory, LAV can also predict the trajectories of the non-ego vehicles in the scene. However, LAV ignores the objects behind the ego vehicle. We use a constant velocity model for predicting the trajectories of the vehicles ignored by LAV. We average the speed and direction of the vehicles over 5 previous timesteps. Each trajectory predicted by LAV consists of K waypoints; we set $K = 20$.

2.5 Experiments

We evaluate our approach for important object identification against three baselines using the HOIST dataset. Human annotations are used as ground truth. As mentioned in 2.3.1, an object marked by at least θ_1 annotators are considered important and objects marked important by fewer than θ_2 annotators are considered unimportant. Objects marked important by fewer than θ_1 but no less than θ_2 annotators are ignored, i.e., approaches are not penalized for identifying them incorrectly. For our experiments, we set $\theta_1 = 3$ and $\theta_2 = 2$.

2.5.1 Comparison with Baselines

We compare our method with the following baselines for estimating object importance: **Everything Important:** All objects within the scene are considered equally important. We do so to highlight the need for the important object identification task.

Inverse Distance: Objects are deemed important based on their proximity to the ego vehicle. The importance score is the negative L2 distance between the object and the ego vehicle.

PlanT [40]: PlanT is a transformer-based planner for autonomous driving. However, PlanT also outputs relevance scores for vehicles in the form of learned attention weights. We use PlanT’s predicted relevance score as the importance score for the objects in our dataset. PlanT model was trained on the longest6 benchmark by imitation learning using a rule based expert driver. HOIST contains data from a subset of the longest6 benchmark, and the ego vehicle in HOIST is also driven using the same rule based expert. Therefore, our evaluations are “in-domain” for PlanT’s training distribution.

Table 2.1 quantitatively compares our approach with the baseline approaches discussed above. We report the Average Precision (AP), the Optimal Threshold F1 Score (OT-F1), and the Optimal threshold Accuracy (OT-Acc) values. The optimal threshold F1 score and accuracy are the highest F1 score and accuracy values across different thresholds.

The results in Table 2.1 show that our method outperforms all the baseline approaches across the three metrics. Compared to our approach, the “Everything

2. Object Importance Estimation using Counterfactual Reasoning for Intelligent Driving

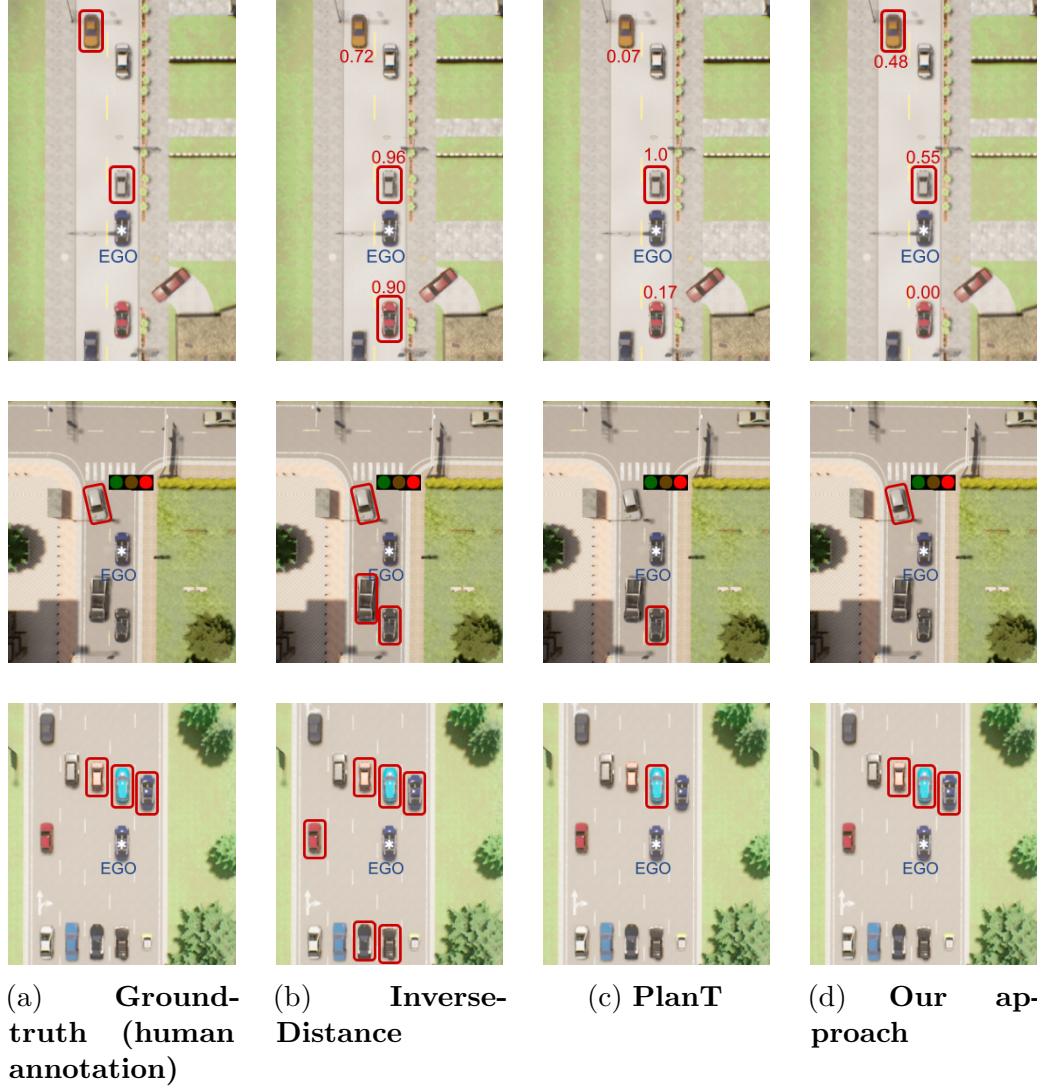


Figure 2.4: Qualitative comparison of our approach with the baselines. The objects are marked as important based on the optimal threshold for the F1 Score for each approach.

Approach	AP $\uparrow$	OT-F1 score $\uparrow$	OT-Accuracy $\uparrow$
Everything Important	0.159	0.277	90.5 %
PlanT	0.572	0.594	92.7 %
Inverse Distance	0.630	0.599	93.1 %
Ours	0.710	0.708	94.8 %

Table 2.1: Our approach outperforms the baseline approaches with a considerable margin on all metrics.

Important” approach performs significantly worse (0.551 lower AP, 0.431 lower OT-F1 score, and 4.3% worse OT-Accuracy). This shows that attending to every object in the scene is not necessary in order to make safe driving decisions for the ego vehicle.

Our approach also outperforms PlanT on all the metrics (0.138 lower AP, 0.114 lower OT-F1 score, and 2.1% worse OT-Accuracy). PlanT considers a very short horizon when reasoning about the state of the ego vehicle and other objects. It only predicts four waypoints into the future for the ego vehicles and reasons about the states of the other objects only for the next timestep. Due to this, PlanT usually assigns a high relevance score to only a few objects nearest to the ego vehicle. Further, it does not consider perturbations in the trajectories of other objects, which can cause it to miss some objects that would be important under perturbations.

Our approach also outperforms the “Inverse Distance” baseline (0.080 lower AP, 0.109 lower OT-F1 score, and 1.7% lower OT-Accuracy). This demonstrates that proximity does not always indicate importance. Importance can be better estimated by reasoning about the trajectories of the non-ego vehicles and their influence on the ego vehicle’s driving. The qualitative examples in Fig. 2.4 shed more light on the reasons behind the superior performance of our approach.

2.5.2 Ablations

We perform extensive ablation experiments to highlight the importance of the various components of our approach and justify our design choices. We first ablate over the two counterfactual scenarios, i.e., the removal score ($\mathcal{RS}$) and the velocity perturbation scores ($\mathcal{VS}$). The results on the top half of Table 2.2 show that using these scores alone perform worse on all three metrics compared to our method. This demonstrates the complementary nature of the two scores and the benefit of combining them, as in

Approach	AP $\uparrow$	OT-F1 score $\uparrow$	OT-Accuracy $\uparrow$
Object Removal Score	0.466	0.444	91.7 %
Velocity Perturbations	0.621	0.670	94.2 %
Ours	0.710	0.708	94.8 %
Ours w/o Hard Stop	0.652	0.628	93.9 %
Ours w/o Speed Up	0.701	0.695	94.5 %
Ours w/o Lane Change	0.565	0.512	93.1 %
Ours w/o Ego vehicle Perturbations	0.635	0.600	93.7 %
Ours w/o Index based Weighting	0.435	0.601	90.5 %

Table 2.2: Ablation experiments demonstrate the impact of the various components and design choices of our method.

our method.

Next, we ablate over the three velocity perturbations. The bottom half of Table 2.2 shows that removing any of the three velocity perturbations worsens performance. Removing the “lane change” perturbation results in an especially large drop in performance (0.145 lower AP). This might be because the “lane change” perturbation assigns a high importance score to objects moving in lanes adjacent to the ego vehicle.

Further, we try to understand the benefits of perturbing the ego vehicle trajectory. The performance deteriorates (drop of 0.075 in AP) if we do not perform velocity perturbations for the ego vehicle. Finally, we perform an ablation experiment to assess the significance of index based weighting in the velocity perturbation score. We score vehicle importance due to the Velocity Perturbations based on how soon they are predicted to collide with the ego vehicle (Eq. 2.2). Without index based weighting (treating all vehicles identified as important due to velocity perturbations with equal importance), the performance drops drastically across all metrics (AP drops by 0.357, OT-F1 score drops by 0.107, and OT-Accuracy drops by 4.3%). Removing index based weighting results in a higher false positive rate as vehicles that influence the ego vehicle’s decisions immediately are considered equally important as those which affect the ego vehicle’s decisions later.

Approach (Only Vehicles)	AP $\uparrow$	OT-F1 score $\uparrow$	OT-Accuracy $\uparrow$
Everything Important	0.151	0.265	91.0 %
PlanT	0.577	0.598	93.0 %
Inverse Distance	0.614	0.593	93.2 %
<i>IS</i> (Eq 2.3, Ours - Vehicles)	0.688	0.697	94.8 %

Table 2.3: Comparison of our approach with the baseline approaches for *vehicle* importance estimation. Our approach outperforms the baseline approaches across all 3 metrics

2.5.3 Importance Prediction for Vehicles and Pedestrians

In a dynamic traffic scene, vehicles and pedestrians have different movement characteristics, and hence we use different approaches for estimating the importance of vehicles and pedestrians (see Sec 2.4.2). To understand the performance of our approach on vehicles versus pedestrians, we also analyzed each of these categories separately.

Table 2.3 compares our approach for vehicles (Sec 2.4.1) with the baseline approaches (Sec 2.5.1). We observe similar trends as for the full dataset (vehicles & pedestrians). Our method outperforms the other baselines on all metrics.

Table 2.4 compares our approach for pedestrians only (Sec. 2.4.2) against the baseline approaches (Sec. 2.5.1). As described in Sec. 2.4.2, we use inverse distance for pedestrian importance estimation. Using inverse distance outperforms all the other approaches across the three metrics. Its near-perfect AP (0.996) shows that pedestrians are deemed important based on their distance from the ego vehicle. PlanT only reasons over a short horizon (Sec 2.5.1), hence it frequently identifies pedestrians closer to the ego vehicle as important, resulting in a good performance. We also experiment with using our proposed approach for vehicles (*IS*, Eqn. 2.3) to estimate pedestrian importance; this approach also does not perform as well as using inverse distance, and hence we use inverse distance for estimating pedestrian importance in our method. While *IS* is able to identify pedestrians as important when they venture out on the road due to the removal score ($\mathcal{RS}$), it fails to identify the pedestrians on the sidewalks as important.

Approach (Only Pedestrians)	AP $\uparrow$	OT-F1 score $\uparrow$	OT-Accuracy $\uparrow$
Everything Important	0.600	0.749	60.0 %
PlanT	0.974	0.926	90.0 %
<i>IS</i> (Eq. 2.3)	0.963	0.960	95.0 %
Inverse Dist. (Ours - Ped)	0.996	0.978	97.5 %

Table 2.4: Comparison of the baseline approaches for *pedestrian* importance estimation. *IS* refers to our approach for *vehicle* importance estimation (Sec. 2.4.1, Eqn. 2.3). Our approach outperforms all other approaches across all 3 metrics.

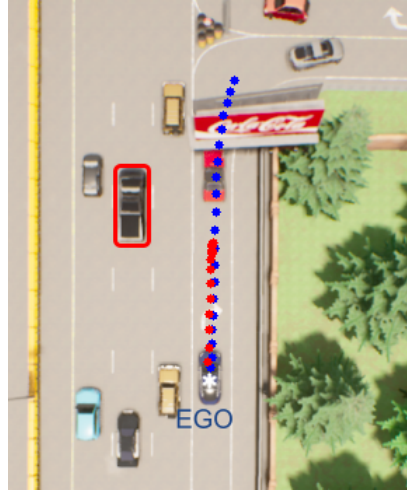
2.5.4 Failure Cases

Occasionally, our approach failed to correctly identify important objects. We analyzed the reasons for these failures.

Errors in trajectory prediction: Our method leverages an autonomous driving agent [9] to predict trajectories of the vehicles in the scene. Hence, Our method is prone to errors made by the autonomous driving agent [9] in trajectory prediction. Fig. 2.5a shows a driving scenario under which an incorrect removal score is predicted for vehicles due to an error in the ego vehicle’s trajectory prediction. Since the calculation of the Removal Score ($\mathcal{RS}$) involves probing an autonomous driving agent with true and counterfactual driving scenarios, analysing the error cases that arise due to an incorrect removal score can help gain insights about the driving scenarios where autonomous driving agents fail.

Vehicles in far-away lanes: As a defensive driving measure, We perturb the ego vehicle trajectory along with the other vehicles in the scene. While perturbing the ego vehicle’s trajectory improves overall performance (Table 2.2), sometimes this approach marks vehicles with remote chances of collision as important. For example, in Fig. 2.5b, the white vehicle is identified as important, even though the chances of it colliding with the ego vehicle are slim.

Rear Vehicles: We devised the speed-up perturbation to identify rear vehicles as important. However, In some cases even after speeding up the perturbed trajectory of the rear vehicle does not result in a collision with the ego vehicle. In those cases, our method fails to identify rear vehicles as important. These failures usually occur



(a) Removing the black truck (red box) incorrectly results in a predicted slow down for the ego vehicle (blue trajectory to red trajectory).



(b) The white car (red box) is incorrectly identified as important. The perturbed trajectories of the ego vehicle and non-ego vehicle are shown in orange and dark green, respectively.

Figure 2.5: Failure cases of our method.

when the autopilot controlling the rear vehicles is moving at a very slow speed or is maintaining a large gap from the ego vehicle.

2. Object Importance Estimation using Counterfactual Reasoning for Intelligent Driving

Chapter 3

Modeling Drivers’ Situational Awareness from Eye Gaze for Driving Assistance

3.1 Introduction

Future Driving Assistance Systems (DAS) might be designed to warn users about important [24] objects in their environment. However, you are not very likely to adopt such a system if it naively alerts you about every important object on the road. This leads to unnecessary and redundant alerts. Repeated exposure to such alerts can lead to — a well documented phenomenon known as “alert fatigue” [2]. To address this gap, we tackle the real-time object-level modeling of drivers’ Situational Awareness (SA) [26], specifically the set of traffic objects (vehicles, pedestrians, and two-wheelers) in the world that the driver is aware of at any given time.

Drivers’ eye gaze is closely linked to their situational awareness [23, 49, 60]. Hence a popular idea has been to approximate attention using raw eye-gaze fixations. However, this is not accurate, since we regularly use our peripheral vision and memory to build and maintain situational awareness while driving [23]. Additionally, drivers can ostensibly “gaze” at objects without gaining situational awareness, due to effects like inattention blindness or saccading over objects without fixating on them [51].

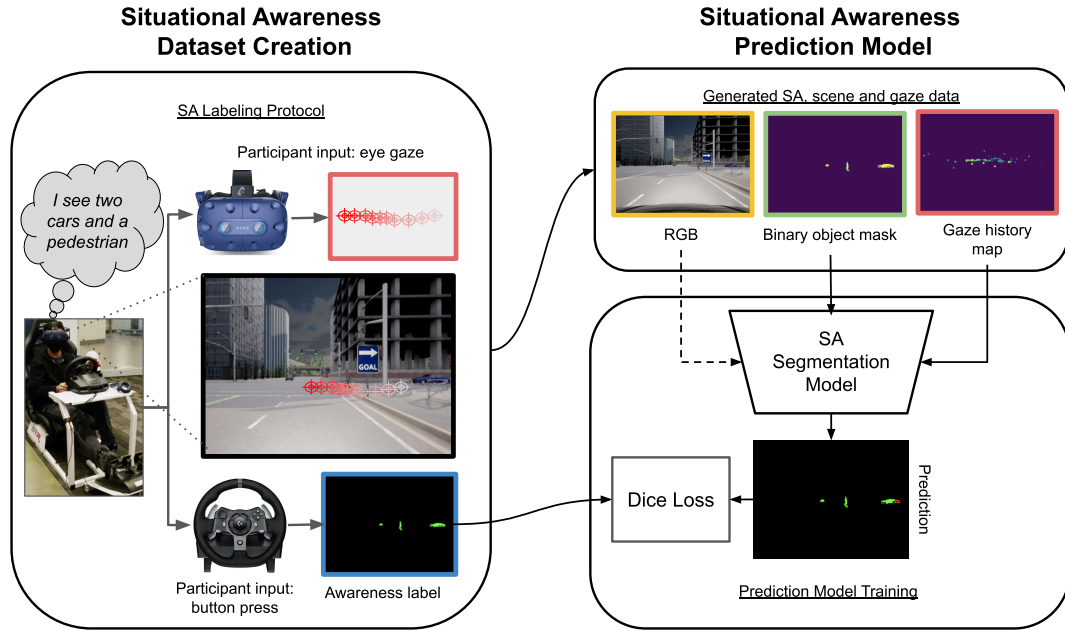


Figure 3.1: We collect drivers' object-level situational awareness (SA) data via a novel interactive protocol in a VR driving simulator. We use the generated data to train a driver SA predictor from visual scene context and driver eye gaze. Casting this as a semantic segmentation problem allows our model to use global scene context and local gaze-object relationships together, processing the whole scene at once regardless of the number of objects present.

Thus, we aim to learn a supervised model for predicting a driver's situational awareness from their eye gaze and the scene context. However, training such a model requires a driving dataset with explicitly labeled drivers' object-level situation awareness. This dataset should be a collection of sequences of driving events comprising the scene context, the driver eye gaze history over the scene, and labels of the drivers' situational awareness over each traffic object.

To facilitate learning and the downstream assistance tasks, the labels need to have certain characteristics:

1. Labels should be continuous, i.e., they should explicitly denote the start of the drivers' awareness over each object. This is important since the transition of driver awareness is crucial for determining when it is appropriate to alert the driver to the object.
2. Labels should be dense, i.e. we need an awareness label for every object that

enters the field of view of the driver.

3. Labels should be collected in a way that does not affect the normal gaze behavior of the driver to avoid distribution shift at test-time.

Most current SA labeling efforts are ineffective as they collect data either intermittently or sparsely [13, 15, 16, 36, 47].

The most popular prior work Situation Awareness Global Assessment Technique (SAGAT) [27, 60] pauses the simulated driving session, and measures SA through post-hoc queries to the driver about traffic objects present in the scene. These collected labels are intermittent — they are valid for the moment when the simulation was paused and are sparse, as the driver is only probed about objects within certain parts of the scene.

In this work, we first introduce a novel SA labeling protocol (Sec. 3.2) that produces continuous and dense object-level SA labels. As a part of our protocol, drivers indicate their awareness of all objects in their field-of-view, by pressing directional buttons on the steering wheel controller (Fig. 3.1). Using our protocol, we collect a dataset of 80 episodes. In each episode, drivers are instructed to drive to an in-world goal inside a VR driving simulator [44] while following the SA labeling protocol. We record their driving actions, eye gaze, and SA labeling button presses along with the simulator state.

We use the aforementioned dataset to train a SA estimation model that takes in the scene context and a history of the driver’s eye gaze to predict a drivers’ object-level SA status given the (Sec. 3.3). We cast this problem as a semantic segmentation problem and show that it performs better than a common-sense gaze-intersection baseline and prior work that uses handcrafted features [60].

In summary, our contributions (Fig. 3.1) are the following:

- **SA Labeling Protocol:** an interactive protocol for obtaining continuous and dense SA labels for on-road agents in a driving scene, without disrupting the driving task
- **SA Data Collection:** a driving dataset with continuous object-level SA labels, traffic object states, and driver eye gaze collected using our protocol in a VR driving simulator with 20 drivers
- **SA Prediction Model:** a learned gaze-based driver situational awareness model which predicts SA over the scene on an object-level basis

This chapter is adapted from our work [7] published in the 7th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions.

3.2 Situational Awareness Data Collection

We collected our driver object-level SA dataset in a VR driving simulator (DReyeVR [44]). Drivers were asked to drive safely following a series of directional goal signs (see RGB image in Fig. 3.1) along scripted routes. The drives were instructed to simultaneously follow the SA labeling protocol to record object-level SA labels.

Situational Awareness Labeling Protocol: Under our proposed SA labeling protocol, drivers are instructed to push a button on their steering wheel as soon as they perceive a vehicle, pedestrian, or two-wheeler (collectively, traffic objects). For each new traffic object they perceive, they are instructed to press one of four buttons to indicate their awareness (see Fig. 3.2). The button to be pressed is determined by the relative position of the target object to the ego-vehicle. For instance, if there is an object in front of the vehicle, the forward button should be pressed. The steering wheel used has two sets of four buttons; the set of buttons on the left is used for vehicles and the right one is used for 2-wheelers+pedestrians. An example sequence of traffic objects and their corresponding button presses is shown in Fig. 3.2.

The awareness labels are generated by associating button clicks with target objects. The direction is used to associate button presses with target objects. An object in a scene is considered ‘unaware’ until it is associated with a button press, after which it’s status is flipped to ‘aware’. More details about how the awareness labels are generated can be found in the supplementary material.

Route & traffic design: Each route consists of a predefined source, destination, and path. Each route also contains in-world navigational goal signs to direct the drivers along the path. Routes were designed to have an average drive time of about 4 minutes. Each route was driven by a maximum of 8 drivers and a minimum of 4 drivers and there were a total of 15 unique routes. Participants were pre-assigned routes so each route would be seen equally but some chose to terminate early due to VR-induced nausea, causing an imbalance in the final number of routes.

At least one safety critical scenario such as a jaywalking pedestrian was included in each route. We did so to ensure that driver gaze before and during safety critical

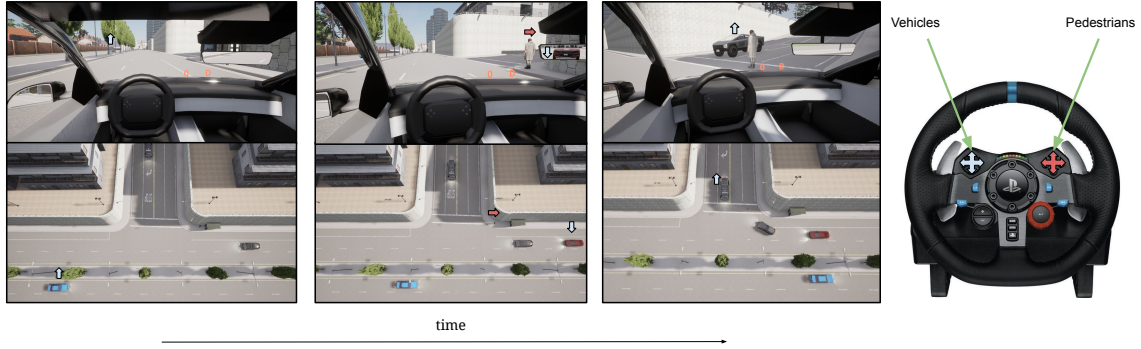


Figure 3.2: Example sequence of right hand turn with object-level driver responses. The top row shows the scene from the driver view and the bottom row shows the same scene via a birds-eye view. Labels are shown as colored arrows above the respective traffic object. Labels correspond to buttons on the steering wheel (right). Blue corresponds to vehicle labels and red to pedestrian labels.

scenarios was also represented in the dataset. More details on the scenarios can be found in the supplementary material. The traffic along each route was randomly generated. However, multiple objects appearing in the scene from any single direction could lead to ambiguities in associating objects with button clicks. Hence, we limit the number of new objects of each type appearing simultaneously at intersections in each direction to one. Note that having different sets of buttons for vehicles and pedestrians(+two wheelers) allows us to disambiguate between object types appearing in the same direction.

Data collection details: We ran our SA protocol with 20 participants, each with 1+ year of holding a valid US or international driver’s license. Each participant was given a set of scripted instructions and were first given time to interact and familiarize themselves with the interface and the simulator. Once they were comfortable with driving in the simulator, they were introduced to the secondary labeling task and asked to perform it while completing a trial route. Participants saw a maximum 5 non-trial routes each, but some participants did not complete all 5 routes due to the onset of discomfort from VR cybersickness. We collected a total of 80 routes worth of data which resulted in about 340 minutes of recorded driving time. The data collection was approved by the university’s IRB. Some additional details about the data collection are provided in the supplementary material.

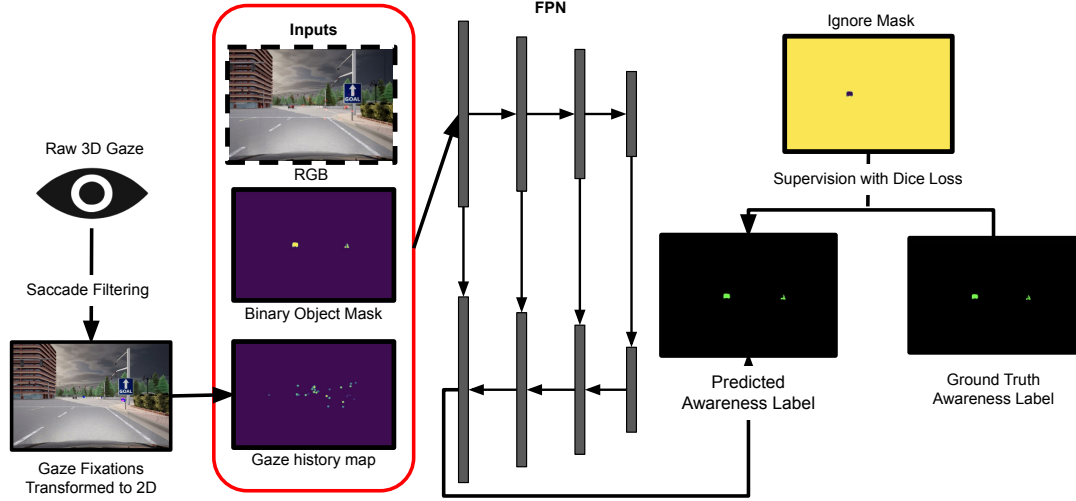


Figure 3.3: Object-wise SA prediction algorithm. A history of raw driver gaze is filtered to exclude saccades and then transformed to 2D pixels in the current camera position. These are used to create a gaze history map which is input together with an object segmentation of the scene (or optionally, RGB). The Feature Pyramid Network (FPN) then produces a 3 class segmentation (unaware, aware, background). During training, loss is ignored for objects which entered into the driver awareness prior to the gaze history window.

3.3 Modeling Driver SA

In modeling driver situational awareness, our goal is to predict a driver’s awareness status over all dynamic traffic objects in the scene at a given time using scene information and the driver’s gaze. Specifically, for any given traffic object obj , we would like to produce a prediction of the binary awareness status A_{obj} where $A_{obj} \in \{aware, unaware\}$.

Problem formulation: We cast the problem of driver SA modeling as a segmentation problem, where the input is a visual representation of the scene in front of the user and the user’s gaze, and the output is a prediction of the objects in the scene that the driver is aware of.

The model takes the scene context and the gaze history of the driver and outputs a segmentation map with 3 classes: *aware*, *unaware*, & *background*. Object-level awareness labels are then derived from the output segmentation by taking the mode

class of the pixels corresponding to each object while ignoring the background class, giving us A_{obj} for each object that is visible in O_t .

Gaze representation: We represent the gaze as a history map, which is obtained from a sequence of 3D gaze over a historical window of length W seconds. Each gaze sample is a 3D ray G_i pointing in the direction of the driver’s gaze, which we project into the camera coordinates to convert to a 2D pixel location. The 3D point on this ray we project into 2D is the first point of intersection with the world. We transform the gaze into 2D pixel coordinates $g_i = M_t G_i \quad \forall \quad i \in \{1, 2, \dots, N_g\}$, where M_t represents a transform from world coordinates to the coordinates of the camera used at timestep t . Gaze pixel locations are represented as a fixed size dot (see “Gaze history map” in Figure 3.3). Additionally, since drivers cannot gain new awareness during gaze saccades (see saccadic suppression, Ch 2. [25]), we perform gaze event detection using the I-BMM classifier [46] and exclude saccades from the gaze map.

Ignore Mask We also use an additional “ignore mask” to zero out losses from traffic objects that entered the user’s awareness more than W seconds ago. Consider a vehicle that entered the user’s awareness 15 s prior to the current timestep. If we use a history window $W = 10$ s, the driver gaze correlated with awareness of that vehicle is no longer represented, though the vehicle is still labeled as *aware*. If we penalize the network during training for mis-classifying that object, we are penalizing a prediction for which the network has incomplete information.

Visual scene representation: The visual scene representation uses a binary object mask to represent the scene; We choose to use a fixed size (600×800) image representation from a viewpoint in front of the ego-vehicle to control the scope of experiments. The binary object mask was obtained directly from CARLA, but could be replaced by any off-the-shelf vehicle/pedestrian segmentation algorithm.

Model and training details: We used a Feature Pyramid Network [35] segmentation model with a MobileNetV2 [41] backbone (pre-trained on ImageNet). The backbone was chosen for its low number of parameters ($2M$) and runtime efficiency. While our dataset contained a similar number of aware to unaware objects, unaware objects usually were further from the ego-vehicle and occupied much smaller portions of the input images. We calculated the ratio of the unaware pixels to aware pixels in the label masks as about 1:20 and used an unaware class weight of 20 (background weight= 10^{-5}). We trained with the Dice loss due to its ability to handle class

imbalanced data [59].

3.4 Evaluation & Discussion

Baselines: We compare our method to three baselines: the majority class, a common-sense gaze intersection baseline, and a prior art baseline using handcrafted features. The “**majority class**” baseline simply predicts the majority class in the test set (“unaware”: 53% share). The “**gaze intersection**” baseline performs a simple check: if the driver’s gaze is within the segmentation mask of a traffic object (vehicle, pedestrian, or 2-wheeler) in the past T seconds, it assigns the *aware* label to it (others assigned *unaware*). We use $T = 10$, matching the other baselines.

The prior art baseline (“**handcrafted features**”) is an SVM model that takes several handcrafted features as input and produces a binary label output [60]. We re-implemented their model based on the paper description (authors’ code or data were not publicly available). The original work lists 5 sets of features, computed across a 10s analysis window (similar to the gaze history window in our method): *Gaze point-based*, *Human visual sensory dependent*, *Object spatial-based*, *Object property-based*, and *Human short-term memory-based*. We implemented the first 3 of these feature sets and the object type feature (vehicle vs pedestrian) from the “Object property-based” set. Most of the “Object property-based” features were excluded since they were difficult to compute and required privileged scene information (*e.g.* one feature required the state of the corresponding traffic light for every traffic object in scene; another was manually annotated). Human short-term memory-based features were also excluded since they were difficult to compute and did not contribute much ($< 1\%$ point) to overall performance in the original evaluation [60]. The original SVM was trained on 1078 training samples. Since neither the trained model nor code were available, we trained our implementation of the SVM on a subset of our training data. We trained the SVM on 10 episodes in our train set, which is about $3\times$ the training data used in the original work. SVM implementations generally cannot handle very large datasets since the entire dataset is loaded into memory during training and mini-batch SVM training is non-trivial. To train the SVM, we used a machine with 128GB RAM but could only use 15% of the training set.

Experimental settings: Our dataset contains 80 episodes of which we used

Model	inf. cmplx.	Acc. (↑)	Prec. (↑)	Recall (↑)	Model Ablation	Acc.	Prec.	Recall
Majority class	1	52.99%	0.53	1	No ignore mask	71.07%	0.79	0.62
Gaze intersection	1	46.87%	0.41	0.54	Raw gaze	73.69%	0.84	0.61
Handcrafted features [60]	N	65.47%	0.66	0.69	Gaze heatmap	76.13%	0.85	0.65
Ours	1	79.21%	0.83	0.77	No gaze fading	77.22%	0.85	0.69
					Gaze 20s hist.	74.05%	0.83	0.60
					Gaze 5s hist.	78.62%	0.87	0.70
					RGB	59.92%	0.83	0.30
					Ours (Full)	79.21%	0.83	0.77

(a) Performance of our model & baselines

(b) Ablations for our model

Table 3.1: Prediction performance of models and baselines on the SA prediction task. Our model outperforms the non-trivial baselines on all 3 metrics and scales better as objects in the scene increase. [inf. cmplx. = inference time complexity with N objects, Acc. = Accuracy, Prec. = Precision]

64 (80%) for training. 10% of the training episodes were used as the validation set. The test set was a separately held out set of 16 episodes. It was partitioned so that participants were disjoint between the train and test set. This is important since we want to test the generalization to new users; it would be impractical to put every new driver through the SA protocol when deploying such a system.

We use 3 metrics to evaluate and compare methods: object-level accuracy, precision, and recall. For precision and recall, the positive class is the “unaware” class. This is because downstream applications such as driver assistance systems which alert the driver will care about how well our system can predict which traffic objects the driver is not aware of. “Precision” is thus a measure of how often our prediction of an object being unaware is correct — errors are “aware” objects classified as “unaware.” This type of error can lead to alert fatigue for an end-user [2]. “Recall,” on the other hand, indicates how many of the “unaware” objects in the dataset were correctly predicted — these are objects that the driver wasn’t aware of but our system predicted that they were.

Results & Discussion Our quantitative evaluation results can be found in Table 3.1. The naive gaze-intersection baseline, as expected, performs the worst,

confirming that it is not enough to simply count which objects were “gazed-at”. The prior art handcrafted features baseline performs better but significantly worse than our method. In terms of runtime, the prior art baseline has 2 expensive parts: computing features per object and doing SVM inference (this can be batched across objects). On average each part takes 5 ms, resulting in a total average runtime of $(5N + 5)\text{ms}$ on an AMD 5955WX CPU (for N objects in scene). In contrast, our network takes 11ms total for a forward pass (on a 4090 GPU) and does not scale with the number of scene objects. Some qualitative comparisons of model outputs can be seen in Fig. 3.4.

Our ablations (Table 3.1, right) show the performance impact of several design choices described in Sec. 3.3. In terms of gaze representation, the ignore mask (used to avoid penalizing mispredictions of awareness transitions outside the gaze history window) was the most important during training — responsible for an 8% accuracy drop when removed. Using saccade filtered gaze instead of raw gaze was the next most important. We also investigated the use of gaze heatmaps as the gaze representation similar to previous work [8, 39], in which each gaze point is represented by an isometric 2D Gaussian that could accumulate in weight at fixations; this performed about 3% worse than using fixed sized dots. This is similar to the issue of representing corrective clicks in an interactive segmentation task, where a similar result has been found [45]. The results indicate that the use of gaze fading was only responsible for about 2% of the model’s performance. This suggests that the presence and location of a gaze point within the gaze history window contains most of the information about awareness rather than the exact temporal order of the gaze. Finally, using an RGB image as input resulted in 20% worse accuracy with the same model size (except the initial layer), as the model now has to simultaneously perform segmentation and SA modeling.

Limitations: Our proposed SA labeling protocol is mainly limited by the fact that some traffic configurations can lead to ambiguity in assigning a button — whenever there is more than one new object of the same type (vehicle or pedestrian) from the same cardinal direction relative to the driver. We created an interface for manual annotation to resolve ambiguities post-hoc. The biggest limitation of our model is its static, memoryless nature. Since SA is inherently a temporal signal, improvements can probably be achieved by performing temporal modeling. Currently, our method treats

3. Modeling Drivers' Situational Awareness from Eye Gaze for Driving Assistance

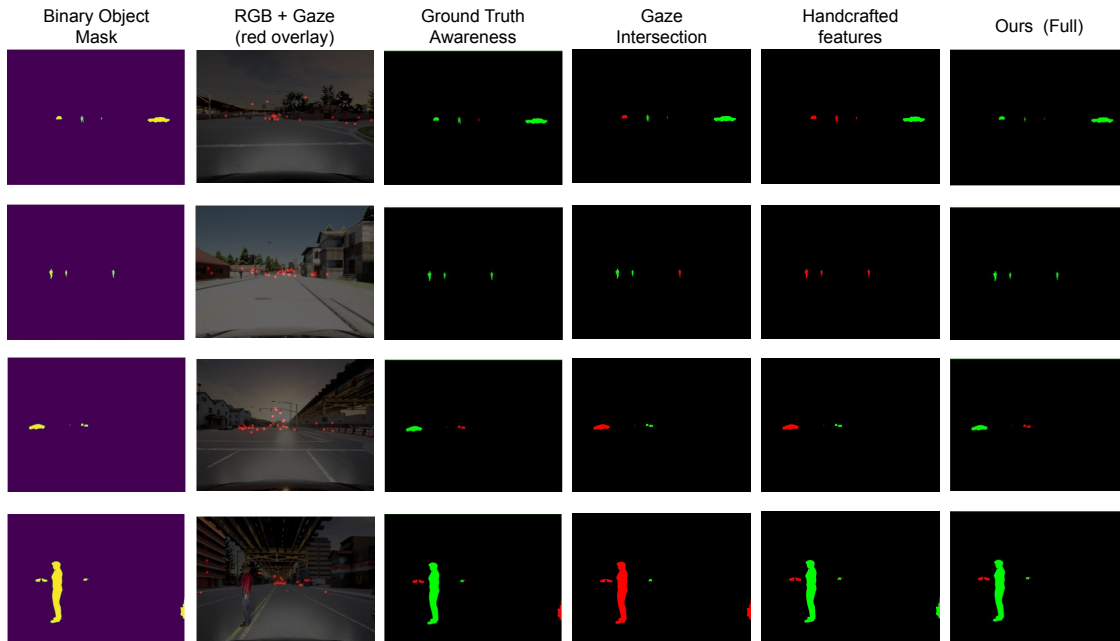


Figure 3.4: Qualitative results for our model and baselines. Each row represents an independent driving scene. The RGB image shows the most recent 10s of gaze overlaid as red dots.

each timestep as independent and would require an external module to implement memory.

3. Modeling Drivers' Situational Awareness from Eye Gaze for Driving Assistance

Chapter 4

Conclusion and Future Work

We focus on the task of important object identification in the context of driving. We define important objects as the ones a driver would watch out for in order to drive the ego vehicle safely. This definition of importance is better suited for the development of situational awareness support systems than prior works which only consider vehicles currently influencing the ego-vehicle’s trajectory. We put forth a new public dataset named HOIST, which contains human-annotated importance labels for vehicles and pedestrians in simulated driving scenarios. Unlike prior works, HOIST consists of multi-modal sensor data used by state-of-the-art autonomous driving agents. We further contribute a novel approach that relies on counterfactual reasoning to identify object importance within dynamic traffic scenes. Our approach outperforms various baselines for the task of important object identification on HOIST. In this work, we have shown the effectiveness our approach in identifying important objects in the HOIST dataset. In the future, we wish to extend our approach to provide safety alerts to drivers in real-time.

We proposed a new interactive protocol to record human drivers’ object-level situational awareness that produces continuous and dense awareness labels. We use it to record a SA dataset with 20 drivers in a VR driving simulator. Additionally, we use this dataset to train a driver object-level SA model by casting it as a semantic segmentation problem. Our model outperforms baselines and prior work while scaling better to arbitrary numbers of objects in the scene. In the future, we plan to use our driver SA model in the inner loop of a driver assistance system that provides

4. Conclusion and Future Work

intelligent alerts or interventions in safety-critical situations and evaluate this in a simulator-based user study. We also commit to releasing our code and data publicly upon acceptance in the hope that it will facilitate more work in the domain.

Bibliography

- [1] Ekrem Aksoy, Ahmet Yazıcı, and Mahmut Kasap. See, attend and brake: An attention-based saliency map prediction model for end-to-end driving. *arXiv preprint arXiv:2002.11020*, 2020. [2.2](#)
- [2] Jessica S Ancker, Alison Edwards, Sarah Nosal, Diane Hauser, Elizabeth Mauer, Rainu Kaushal, and With the HITEC Investigators. Effects of workload, work complexity, and repeated alerts on alert fatigue in a clinical decision support system. *BMC medical informatics and decision making*, 17:1–9, 2017. [3.1](#), [3.4](#)
- [3] Sonia Bae, Erfan Pakdamanian, Inki Kim, Lu Feng, Vicente Ordonez, and Laura Barnes. Medirl: Predicting the visual attention of drivers via maximum entropy deep inverse reinforcement learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13178–13188, 2021. [2.1](#), [2.2](#)
- [4] Tobias Bär, Denys Linke, Dennis Nienhüser, and J Marius Zöllner. Seen and missed traffic objects: A traffic object-specific awareness estimation. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 31–36. IEEE, 2013. [1](#)
- [5] Abhijat Biswas and Henny Admoni. Characterizing drivers’ peripheral vision via the functional field of view for intelligent driving assistance. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. IEEE, 2023. [2.1](#)
- [6] Abhijat Biswas, Badal Arun Pardhi, Caleb Chuck, Jarrett Holtz, Scott Niekum, Henny Admoni, and Alessandro Allievi. Mitigating causal confusion in driving agents via gaze supervision. In *Aligning Robot Representations with Humans workshop @ Conference on Robot Learning*, 2022. [2.1](#)
- [7] Abhijat Biswas, Pranay Gupta, David Held, and Henny Admoni. An interactive protocol to measure a driver’s situational awareness. In *7th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions*, 2024. [3.1](#)
- [8] Abhijat Biswas, Badal Arun Pardhi, Caleb Chuck, Jarrett Holtz, Scott Niekum, Henny Admoni, and Alessandro Allievi. Gaze supervision for mitigating causal confusion in driving agents. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2159–2161, 2024. [3.4](#)

- [9] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022. [2.1](#), [2.4.1](#), [2.4.1](#), [2.4.3](#), [2.5.4](#)
- [10] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [2.1](#), [2.3.1](#), [2.4.1](#)
- [11] David Crundall, Peter Chapman, Steven Trawley, Lyn Collins, Editha Van Loon, Ben Andrews, and Geoffrey Underwood. Some hazards are more attractive than others: Drivers of varying experience respond differently to different types of hazard. *Accident Analysis & Prevention*, 45:600–609, 2012. [2.1](#)
- [12] Joost CF De Winter, Riender Happee, Marieke H Martens, and Neville A Stanton. Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence. *Transportation research part F: traffic psychology and behaviour*, 27:196–217, 2014. [1](#)
- [13] Joost CF de Winter, Yke Bauke Eisma, CDD Cabrall, Peter A Hancock, and Neville A Stanton. Situation awareness based on eye movements in relation to the task environment. *Cognition, Technology & Work*, 21(1):99–111, 2019. [3.1](#)
- [14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. [1](#), [2.1](#), [2.3.1](#)
- [15] Francis T Durso, Carla A Hackworth, Todd R Truitt, Jerry Crutchfield, Danko Nikolic, and Carol A Manning. Situation awareness as a predictor of performance for en route air traffic controllers. *Air Traffic Control Quarterly*, 6(1):1–20, 1998. [3.1](#)
- [16] Mica R Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society annual meeting*, volume 32, pages 97–101. Sage Publications Sage CA: Los Angeles, CA, 1988. [3.1](#)
- [17] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1):32–64, 1995. [2.1](#)
- [18] Jianwu Fang, Dingxin Yan, Jiahuan Qiao, Jianru Xue, He Wang, and Sen Li. Dada-2000: Can driving accident be predicted by driver attentionf analyzed by a benchmark. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4303–4309. IEEE, 2019. [2.1](#), [2.2](#)
- [19] Joseph L Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973. [2.3.2](#)

- [20] Luke Fletcher, Gareth Loy, Nick Barnes, and Alexander Zelinsky. Correlating driver gaze with the road scene for driver assistance systems. *Robotics and Autonomous Systems*, 52(1):71–84, 2005. [1](#)
- [21] Mingfei Gao, Ashish Tawari, and Sujitha Martin. Goal-oriented object importance estimation in on-road driving videos. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5509–5515. IEEE, 2019. [2.1](#), [2.2](#), [2.3](#)
- [22] John Gaspar and Cher Carney. The effect of partial automation on driver attention: A naturalistic driving study. *Human factors*, 61(8):1261–1276, 2019. [1](#)
- [23] Leo Gugerty et al. Situation awareness in driving. *Handbook for driving simulation in engineering, medicine and psychology*, 1:265–272, 2011. [3.1](#)
- [24] Pranay Gupta, Abhijat Biswas, Henny Admoni, and David Held. Object importance estimation using counterfactual reasoning for intelligent driving. *IEEE Robotics and Automation Letters*, 2024. [2.1](#), [3.1](#)
- [25] DE Irwin. Eye movements and perception. 2001. [3.3](#)
- [26] Robert Kaufman, David Kirsh, and Nadir Weibel. Developing situational awareness for joint action with autonomous vehicles. *arXiv preprint arXiv:2404.11800*, 2024. [3.1](#)
- [27] Hyungil Kim, Sujitha Martin, Ashish Tawari, Teruhisa Misu, and Joseph L Gabbard. Toward real-time estimation of driver situation awareness: An eye-tracking approach based on moving objects of interest. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1035–1041. IEEE, 2020. [3.1](#)
- [28] Iuliia Kotseruba and John K Tsotsos. Behavioral research and practical models of drivers’ attention. *arXiv preprint arXiv:2104.05677*, 2021. [2.1](#)
- [29] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977. [2.3.2](#)
- [30] Chengxi Li, Stanley H Chan, and Yi-Ting Chen. Who make drivers stop? towards driver-centric risk assessment: Risk object identification via causal inference. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10711–10718. IEEE, 2020. [2.2](#)
- [31] Chengxi Li, Stanley H Chan, and Yi-Ting Chen. Droid: Driver-centric risk object identification. *IEEE transactions on pattern analysis and machine intelligence*, 2023. [2.2](#)
- [32] Jiachen Li, Hengbo Ma, Zhihao Zhang, Jinning Li, and Masayoshi Tomizuka. Spatio-temporal graph dual-attention network for multi-agent prediction and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10556–10569, 2021. [2.2](#)
- [33] Jiachen Li, Fan Yang, Hengbo Ma, Srikanth Malla, Masayoshi Tomizuka, and

- Chiho Choi. Rain: Reinforced hybrid attention inference network for motion forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16096–16106, 2021. 2.2
- [34] Jiachen Li, Haiming Gang, Hengbo Ma, Masayoshi Tomizuka, and Chiho Choi. Important object identification with semi-supervised learning for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2913–2919. IEEE, 2022. 2.1, 2.2, 2.3
- [35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3.3
- [36] Nikolas Martelaro, David Sirkin, and Wendy Ju. Daze: a real-time situation awareness measurement tool for driving. In *Adjunct Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 158–163, 2015. 3.1
- [37] National Highway Traffic Safety Administration. Distracted Driving, n.d. URL <https://www.nhtsa.gov/risky-driving/distracted-driving>. 1
- [38] Anwesan Pal, Sayan Mondal, and Henrik I Christensen. ” looking at the right stuff”-guided semantic-gaze for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11883–11892, 2020. 2.1, 2.2
- [39] Andrea Palazzi, Davide Abati, Francesco Solera, Rita Cucchiara, et al. Predicting the driver’s focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1720–1733, 2018. 2.1, 2.2, 3.4
- [40] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In *Conference on Robotic Learning (CoRL)*, 2022. 2.2, 2.5.1
- [41] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3.3
- [42] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. *arXiv preprint arXiv:2207.14024*, 2022. 2.4.1, 2.4.1
- [43] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

- Recognition*, pages 13723–13733, 2023. [2.4.1](#)
- [44] Gustavo Silvera, Abhijat Biswas, and Henny Admoni. Dreyevr: Democratizing virtual reality driving simulation for behavioural & interaction research. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 639–643, 2022. [3.1](#), [3.2](#)
 - [45] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022. [3.4](#)
 - [46] Enkelejda Tafaj, Gjergji Kasneci, Wolfgang Rosenstiel, and Martin Bogdan. Bayesian online clustering of eye movement data. In *Proceedings of the symposium on eye tracking research and applications*, pages 285–288, 2012. [3.3](#)
 - [47] Richard M Taylor. Situational awareness rating technique (sart): The development of a tool for aircrew systems design. In *Situational awareness*, pages 111–128. Routledge, 2017. [3.1](#)
 - [48] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. [2.2](#)
 - [49] Trent Victor, Marco Dozza, Jonas Bärghman, Christian-Nils Boda, Johan Engström, Carol Flannagan, John D Lee, and Gustav Markkula. Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk. Technical report, 2015. [3.1](#)
 - [50] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. [2.4.3](#)
 - [51] Yuwei Wang, Yimin Wu, Cheng Chen, Bohan Wu, Shu Ma, Duming Wang, Hongting Li, and Zhen Yang. Inattentional blindness in augmented reality head-up display-assisted driving. *International Journal of Human-Computer Interaction*, 38(9):837–850, 2022. [3.1](#)
 - [52] Cale B White and Jeff K Caird. The blind date: The effects of change blindness, passenger conversation and gender on looked-but-failed-to-see (lbfts) errors. *Accident Analysis & Prevention*, 42(6):1822–1830, 2010. [2.1](#)
 - [53] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022. [2.4.1](#)
 - [54] Tong Wu, Enna Sachdeva, Kumar Akash, Xingwei Wu, Teruhisa Misu, and

- Jorge Ortiz. Toward an adaptive situational awareness support system for urban driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1073–1080. IEEE, 2022. [1](#), [2.1](#), [2.1](#)
- [55] Ye Xia. *Driver eye movements and the application in autonomous driving*. University of California, Berkeley, 2019. [2.2](#)
- [56] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pages 658–674. Springer, 2019. [2.1](#), [2.2](#)
- [57] Ye Xia, Jinkyu Kim, John Canny, Karl Zipser, Teresa Canas-Bajo, and David Whitney. Periphery-fovea multi-resolution driving model guided by human attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1767–1775, 2020. [2.2](#)
- [58] Zehua Zhang, Ashish Tawari, Sujitha Martin, and David Crandall. Interaction graphs for object importance estimation in on-road driving videos. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8920–8927. IEEE, 2020. [2.1](#), [2.2](#)
- [59] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 851–860. IEEE, 2020. [3.3](#)
- [60] Haibei Zhu, Teruhisa Misu, Sujitha Martin, Xingwei Wu, and Kumar Akash. Improving driver situation awareness prediction using human visual sensory and memory mechanism. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6210–6216. IEEE, 2021. [1](#), [3.1](#), [3.1](#), [3.4](#), [??](#)
- [61] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008. [2.2](#)