UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards Safe, Human-Centered Autonomous Driving:
Real-World Artificial Intelligence for Enhanced Situation Awareness and Transition Control

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Ross Greer

Committee in charge:

Professor Mohan Trivedi, Chair
Professor Manmohan Chandraker
Professor Thomas Marcotte
Professor Bhaskar Rao

2024

The Dissertation of Ross Greer is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

III Learning from Trajectories: Novelty in Motion 79

IV Safely Handling the Unexpected: Driver Control Transitions 116

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Chapter 3, in part, is a reprint of the material as it appears in Ross Greer and Mohan M. Trivedi. "Towards Explainable, Safe Autonomous Driving with Language Embeddings for Novelty Identification and Active Learning: Framework and Experimental Analysis with Real-World Data Sets." arXiv preprint arXiv:2402.07320. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in Greer, Ross, Bjørk Antoniussen, Mathias V. Andersen, Andreas Møgelmose, and Mohan M. Trivedi. "The Why, When, and How to Use Active Learning in Large-Data-Driven 3D Object Detection for Safe Autonomous Driving: An Empirical Exploration." arXiv preprint arXiv:2401.16634 (2024). The dissertation author was the primary investigator and author of this paper.

Chapter 5, in part, is a reprint of the material as it may appear in Robotics and Automation Letters, 2024: Greer, Ross, Bjørk Antoniussen, Andreas Møgelmose, and Mohan M. Trivedi. "Language-Driven Active Learning for Diverse Open-Set 3D Object Detection." The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, is a reprint of the material as it appears in Greer, Ross, Nachiket Deo, and Mohan M. Trivedi. "Trajectory prediction in autonomous driving with a lane heading auxiliary loss." IEEE Robotics and Automation Letters 6 (3), 4907-4914 (2021). The dissertation author was the primary investigator and author of this paper.

Chapter 7, in part, in part, has been submitted for publication of the material as it may appear in Robotics and Automation Letters, 2024: Ross Greer and Mohan M. Trivedi. "Perception Without Vision for Trajectory Prediction: Ego Vehicle Dynamics as Scene Representation for Efficient Active Learning in Autonomous Driving." The dissertation author was the primary investigator and author of this paper.

Chapter 8, in part, is a reprint of the material as it appears in Rangesh, Akshay, Nachiket Deo, Ross Greer, Pujitha Gunaratne, and Mohan M. Trivedi. "Autonomous Vehicles that Alert Humans to Take-Over Controls: Modeling with Real-World Data." arXiv preprint arXiv:2104.11489

(2022). The dissertation author was the primary investigator and author of this paper.

Chapter 9, in part, is a reprint of the material as it appears in Greer, Ross, Jason Isa, Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. "On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2022). The dissertation author was the primary investigator and author of this paper.

Chapter 10, in part, is a reprint of the material as it appears in Greer, Ross, Akshay Gopalkrishnan, Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. "Salient Sign Detection in Safe Autonomous Driving: AI Which Reasons Over Full Visual Context." Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV) (2023). The dissertation author was the primary investigator and author of this paper.

Chapter 11, in part, is a reprint of the material as it appears in Greer, Ross, Akshay Gopalkrishnan, Jacob Landgren, Lulua Rakla, Anish Gopalan, and Mohan M. Trivedi. "Robust Traffic Light Detection Using Salience-Sensitive Loss: Computational Framework and Evaluations." Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV) (2023). The dissertation author was the primary investigator and author of this paper.

Appendix A, in part, is a reprint of the material as it appears in Ross Greer and Mohan M. Trivedi. "Ensemble Learning for Fusion of Multiview Vision with Occlusion and Missing Information: Framework and Evaluations with Real-World Data and Applications in Driver Hand Activity Recognition." arXiv preprint arXiv:2301.12592 (2023). The dissertation author was the primary investigator and author of this paper.

I would like to acknowledge the support of Qualcomm, Toyota Collaborative Safety Research Center, especially Pujitha Gunaratne, and Amazon Web Services Machine Learning Solutions Laboratory, for their support of my research through the Qualcomm Innovation Fellowship and project funding. I would like to acknowledge the support of my colleague Kuan-Lin Chen and professors Bhaskar Rao and Mohan Trivedi towards the Qualcomm Innovation Fellowship. From AWS MLSL, I would particularly like to thank Dr. Suchitra Sathyanarayana, Ninad Kulkarni, and Jeremy Feltracco of AWS Machine Learning Solutions Laboratory for sharing AWS Sagemaker GroundTruth. I would also like to acknowledge the support of project funding from the University of California Office of the President Innovative Learning Technology Initiative.

I would like to acknowledge Walter Zimmer, Christian Creß, Huu Tung Nguyen, and Alois Knoll for the excellent maintenance of the TUMTraf dataset, and the National Advanced Driving Simulator (NADS) Team at the University of Iowa for the collection of the CDS dataset. The contributions of these colleagues which made the quantitative and qualitative analysis of certain research chapters possible.

I would like to acknowledge and thank my Ph.D. committee, Professors Trivedi, Rao, Chandraker, and Marcotte, for their support and feedback throughout my studies, and Professors Touri and Pashler for their support during my Preliminary and Qualifying exams. I would also like to acknowledge and thank Professor Shlomo Dubnov, for his support, mentorship, and the many great research and teaching opportunities he has extended to me during my Ph.D.

I would like to thank Dr. Thomas Marcotte, Alejandra Vidrio, Jake Rattigan, and the staff at the Center for Medicinal Cannabis Research for their kind and continuous support in our collaborative studies in interdisciplinary driver safety monitoring.

Finally, I would like to express my gratitude and appreciation for my LISA colleagues, especially Nachiket Deo, Akshay Rangesh, Larry Ly, Akshay Gopalkrishnan, Lulua Rakla, Mathias Viborg Andersen, Bjørk Antoniussen, Jason Isa, Maitrayee Keskar, Bowen Zhang, Kaouther Messaoud, Sumega Mandadi, Afnan Alofi, Andreas Møgelmose, Eddie Huang, Anish Gopalan, Ethan Lee, Brendan Morris, and the many LISA alumni I have had the pleasure of meeting and working with. In this group, I would like to express my highest thanks to my advisor and mentor, Professor Mohan Trivedi, for his constant support and guidance during my graduate studies. Taking his autonomous driving course as a Masters student inspired me to pursue research (and eventually this Ph.D.), and his support and cheering-on in my research and more have made my years spent at UC San Diego the best experience. I feel so fortunate and appreciative to have had the pleasure of being Professor Trivedi's student, and I aspire to be as knowledgeable, caring, generous, and engaged as Professor Trivedi in my own mentoring.

In addition to my research colleagues, I would like to express my gratitude for the support, care, and kindness of my family (especially my parents, Doug and Sabine Greer) and the many, many friends that have been part of my academic journey. I am so grateful to all of the people that have encouraged my learning. Your positive influence (and perhaps on occasion your annotations of training data, questions during presentations, or participation in demos) is surely reflected in the completion of this dissertation. Thank you for all that you do for me.

Thank you all again, and here's to a future with safe, accessible transportation for all!

VITA

2015	Bachelor of Science, Engineering Physics, University of California Berkeley
2015	Bachelor of Science, Electrical Engineering and Computer Science, University of California Berkeley
2015	Bachelor of Arts, Music, University of California Berkeley
2018	Master of Science, Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California San Diego
2024	Doctor of Philosophy, Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California San Diego

ABSTRACT OF THE DISSERTATION

Towards Safe, Human-Centered Autonomous Driving:
Real-World Artificial Intelligence for Enhanced Situation Awareness and Transition Control

by

Ross Greer

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2024

Professor Mohan Trivedi, Chair

Autonomous driving systems involved in perception and planning require large volumes of carefully annotated data for learning and validation. These same systems also must be aware of failure cases so that they can safely request and initiate control transitions to human drivers or remote operators. In this dissertation, I present novelty detection as a unifying solution to both of these problems. Through novelty detection, active learning algorithms can reduce annotation costs by intelligently selecting informative data, which I demonstrate on tasks of 3D object detection and vehicle trajectory prediction. Similarly, novelty detection acts as a requisite step for safely handling hazardous scenarios. Lastly, I present the concept of salience as a property of road objects which expresses their criticality to control decisions, discussing the relevance of this property in developing machine learning systems which have stronger learning and validation over safety-critical scene elements for autonomous driving and can adapt to novelty found in the open world.

Part I

Finding Meaningful Representations of Real-World Driving Scenes

Chapter 1

Prologue: Novelty, The Long Tail, and Data-Driven Autonomy in the Real World

Autonomous driving presents an open avenue for studying questions of how machines might think about, interact with, and learn from the observed environment. But for all the ways we can compare the algorithms of these intelligent robotic systems to humans, their momentum carries dangerous physical consequences, meaning these robots cannot be only a playground for developing machine intelligence, but rather a place for critical, safety-forward study.

As the story of machine intelligence continues unfolding, the power of data has become exceedingly apparent. But, not just any data: “good" data, data which is free of noise and full of all of the variance that we want our intelligent systems to learn. And, the touted solution to any shortcoming of an intelligent system? More (good) data! As a result, larger and larger datasets have been released to address an increasing number of autonomous driving subtasks, so much so that a variety of review papers and repositories exist just to keep track of the wealth of data available [5–7]. Further emblematic of the power of data, foundation models have now entered center stage, with large-language and vision models trained on enormous online repositories showing remarkable generative abilities and so-called “zero-shot learning", by which tasks can be solved without any data collection[1].

[1] In reality, this is a bit of a mischaracterization; the data has been collected previously, in such extreme magnitude that these models show capabilities reflecting large-scale general knowledge of nearly everything one might find on an internet search.

While it may seem that a natural solution to the challenge of autonomous driving would be to continuously collect data (evidenced by the deployment of fleets which repeatedly sweep through cities, collecting videos, maps, and point clouds), the story of data does not stop at collection. Equally important is the meaning ascribed to the data via human annotation. For safety-driven development, this data annotation allows not only for supervised learning techniques, but also for validation of methods in their ability to detect and perceive at levels considered acceptable to humans who will be in and around these machines. However, this annotation takes significant human labor [8]. Mathematician Hanno Gottschak walks through one such illustration of the magnitude of this problem: how many driving scene frames must be annotated to make the statistical claim that the autonomous vehicle is “safer" than a human driver? To this end, he estimates the lower-bound number of hours collectively driven by humans before an accident typically occurs [9]. Considering only errors in perception, a basic sub-task for obstacle avoidance, Gottschalk poses that we must label at least the number of frames that exceed the number of frames a human driver would see statistically prior to an accident, and show perfect (safety-critical[2]) perception on at least this many frames. Using approximations of bounding box and image segmentation labor costs (a subject I elaborate on in Chapter 4), Gottschalk estimates a cost of 1.16 to 51,800 Euros for such safety-validating data annotation (and, since publication, minimum wage has risen). Translating to USD at current exchange rates, we find these values to range from $1.24 trillion to $55,407 trillion. For comparison, at the time of writing, the market valuation[3] of the top companies with autonomous driving ventures are:

1. NVIDIA at $2.19 trillion,
2. Alphabet at $2.14 trillion ($30 billion of which is associated with Waymo),
3. Tesla at $5.36 trillion,

[2] I will introduce this idea of safety-critical perception in later chapters on salience.

[3] https://companiesmarketcap.com/autonomous-driving/largest-autonomous-driving-companies-by-market-cap/

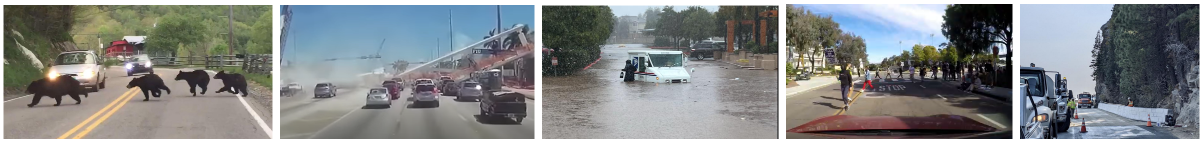


Figure 1.1. There will always be new, creative ways that humans and nature throw chaos into the environment we navigate. Construction, protests, climate events, weather conditions, animal migration, accidents, and natural disaster will continue to create long-tail events for human and autonomous drivers, and systems must be able to respond to these events safely.

4. Mercedes-Benz at $85.15 billion, and
5. Mobileye at $23.54 billion.

So, while data itself can now be widely and massively collected, the labor necessary to enable both supervised learning and validation must be used efficiently if these data-driven systems are to be made feasible, accessible, reproducible, and verifiable. This presents a first problem: for autonomous driving, available human labor limits our utilization of collected data.

But even if such a massive volume of data is annotated, and the case is made that a vehicle is a “statistically safer driver than humans", this does not actually move us towards *having* safer systems; it only makes the case that the state-of-the-art perception is safer than human performance. Safety requires more than perception; systems must look beyond perception to tasks of planning and control.

The Long-Tail Problem

Real-world autonomous driving is characterized as a “long-tail problem". This refers to the long-tailed distribution of events which lie away from typical driving, and is especially a problem when it comes to safety for autonomous driving, as it is not the familiar which poses a risk, but rather encounters with unexpected or novel situations. This phenomenon has also been called the *curse of rarity*, again referring to the difficulty of gathering samples of events that are most likely to cause system safety failures [10]. Further confounding the issue, collected data itself can be noisy, redundant, and uninformative.

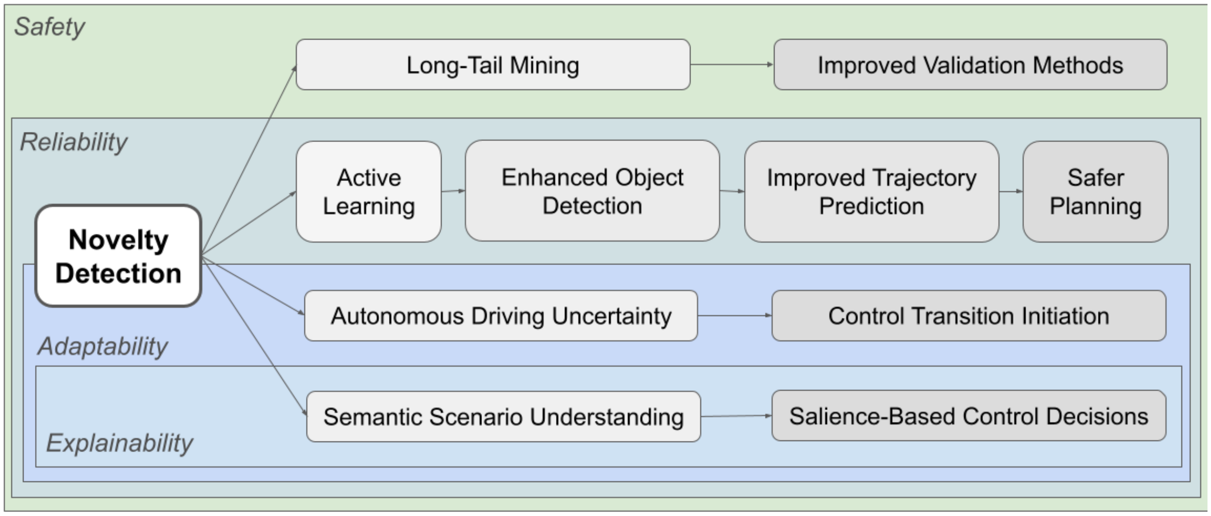

Safety
Long-Tail Mining
Improved Validation Methods
Reliability
Active Learning
Enhanced Object Detection
Improved Trajectory Prediction
Safer Planning
Novelty Detection
Autonomous Driving Uncertainty
Control Transition Initiation
Adaptability
Semantic Scenario Understanding
Salience-Based Control Decisions
Explainability

Figure 1.2. Novelty detection enables a variety of capabilities in the learning, control, and validation of autonomous driving systems, making possible improved safety, reliability, adaptability, and explainability. This dissertation introduces the relationship between these connected system algorithms, representations of the driving environment, and real-world safety outcomes.

This creates an interesting situation for data-driven development; on one hand, we have an annotation bottleneck due to an overabundance of data, and on the other hand, we must continue data collection to continue addressing long-tail events. But, these problems can both be addressed in the same manner: data *curation* can allow for systems to intelligently identify instances of criticality which are most important to annotate for resource efficiency, robust learning, and safety validation, and this is the story I tell in this dissertation.

To be able to detect and respond to surprising occurrences is not a task to be solved once by collecting and annotating enough data. It requires new systems of adaptive learning which can recognize, understand, and respond to changes in the environment. My research addresses these challenges through novelty detection, providing a framework (Figure 1.2) for active learning, safe control transitions, and development of systems which are highly adaptable to the ever-changing and chaotic real world.

This dissertation is organized as follows:

- In Part I, I will present algorithms for detecting novelty in visual datasets using vision-language representations, and in trajectory datasets using state and sequence clustering.

- In Part II, I will present the utility of such novelty detection as a means of active learning in 3D object detection.
- In Part III, I will continue presenting the utility of novelty detection, but now for trajectory prediction, and the relationship of trajectory prediction as an intermediate task between object detection and path planning for safe autonomous driving.
- In Part IV, I will describe algorithms for modeling the transition between autonomous and manual control, highlighting the need for novelty detection in these frameworks.
- In the concluding Part V, I will introduce the idea of road object salience, and how, paired with novelty detection, a framework can support autonomous driving in the presence of long-tail events.

Chapter 2

Terminology and Approaches to Identify Tail Cases

The idea of a “normal" driving scenario is an abstractly defined concept which varies with respect to geographic, individual, and social influences; as humans, we are usually able to notice when something is abnormal, but there are many dimensions by which this abnormality can be characterized, leading to a lack of agreed terminology [11]. Related words include edge cases, corner cases, extreme cases, boundary cases, outliers, anomalies, novelty, unexpected events, unusual events, and long-tail events. In this section, I present definitions adopted in prior literature to facilitate a common understanding toward the contents of this dissertation, and include clarifying commentary on the relationship between these concepts.

- Outliers are situations which deviate from “normal" driving [12]. The term is used to describe information with respect to the distribution of driving scenarios rather than the system’s ability to handle such scenarios. When these are anticipated and addressed for system handling, they are often called *edge cases*. The terminology of *extreme cases* and *boundary cases* typically fall within this *edge case* concept.

- Situations which deviate from “normal" driving and are unexpected or unplanned for system handling are often called *corner cases*. This definition assumes some agreed-upon understanding of what constitutes “normal" driving. Robust detection of corner cases is necessary for reliable and safe autonomous driving perception [11]. Heidecker et al. organize corner cases into three broad layers: content, temporal, and sensor. In the content layer, edge cases can exist at the scene, object, or domain level. In the temporal layer, the playout of a scenario can be a corner case. In the sensor layer, anomalous cases at the pixel or point-cloud level are detected. Koopman et al. provide a different definition, built around the idea of a corner, referring to corner cases as those which arise from the combination of several normal situations coinciding simultaneously in a rare combination [13]. Also aligned with these definitions, Bolte et al. define a corner case as a case with a non-predictable, relevant object in a relevant location [14]. While the term *anomaly* describes a deviation from normality, for autonomous driving, Heidecker et al. clarify that this deviation is manifested in non-conforming behavior or patterns, making the term nearly synonymous to *corner case* [11, 14–16].

- The term *novelty* is used to describe previously unseen instances [12, 17]. This can arise from changes to the distribution underlying a process, or sampling from previously unvisited area of a distribution. We note that corner cases are also essentially novel in this regard, because they are unexpected and unaccounted for in the system. The definition of novelty is especially useful for data-driven learning systems, as it can be used to refer to instances which are “unseen" in training data.

To summarize the above: an unusual occurrence can be atypical with respect to the distribution of driving scenarios found in the real-world, or atypical with respect to what an autonomous driving system has been designed to handle. In the first case, when these are accounted for in system design, they are referred to as “edge cases", while all cases unaccounted for (whether typical or not) can be handled by the umbrella term “corner case". The long-tail problem in autonomous driving assumes a coupling between scenario typicality and danger, pairing normal driving scenarios with safety, and atypical scenarios with increased risk.

In this dissertation, I use novelty as an all-emcompassing term to describe data instances which are yet-to-be-sampled but within the realm of normalcy (shown as discontinuities in

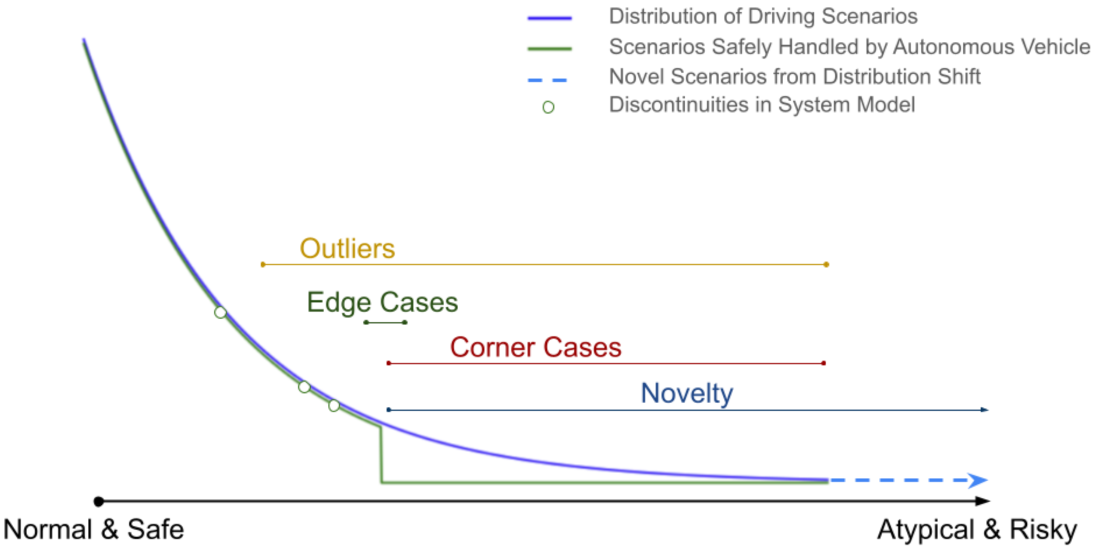

Distribution of Driving Scenarios
Scenarios Safely Handled by Autonomous Vehicle
Novel Scenarios from Distribution Shift
Discontinuities in System Model
Outliers
Edge Cases
Corner Cases
Novelty
Normal & Safe
Atypical & Risky

Figure 2.1. A visual interpretation of the relationship between adopted terminology used to describe unusual driving events and autonomous systems. In my dissertation, I use *novelty* to refer to events that are unhandled by the autonomous system as well as events which are completely unexpected within the distribution of existing driving events, reflecting that the world (and sensors and systems) may present scenarios which are completely novel compared to that which has been observed or existent prior.

Figure 2.1), as well as those which are yet-to-be-sampled within the outlying driving scenario distribution, and then also those which exist beyond the range of expected data, introduced under distribution shifts. In any case, this term “novelty" allows us to represent the idea of something which is not yet part of training data or safe-handling system knowledge. We also make the important note that these words can be used to describe occurrences in the scene or scenario, but also occurrences (or failures) of sensors and hardware.

Prior research in autonomous driving corner case detection have used a variety of methods with a variety of data modalities, such as:

- **Images**: CNN for overexposure detection [18], direct planar hypothesis testing [19] and FCN [20] for small-object detection, learned embedding density estimation for uncertainty in semantic segmentation [21], unexpected scene object detection by synthetic image reconstruction [22],

- **Video**: image prediction of future frames for anomaly detection [14],
- **LiDAR**: CNN on point clouds for open-set instance segmentation [23] and object detection [24], and
- **Radar**: CNN features for open-set radar waveform classification [25].

The utility of curating or mining novelty is multifaceted: (1) Efficiency within data pipelines is improved when data collection fleets can be selectively deployed, data analysis and storage can be filtered to particular instances, and human annotation labor can be reduced; (2) Model performance can be increased by curating data in instances of uncertainty, a topic I address in Chapter 4; and (3) Novel cases, once curated and annotated, can be presented to models during validation to assess safety robustness. Heidecker et al. reinforces this value, stating that "[corner cases] provide both the appropriate training, and the crucial test data to successfully develop and validate robust perception methods" [11], and again by Hanselmann et al., who state “Due to the high consequences of failure, [autonomous driving systems] have to satisfy extraordinarily high standards of robustness in the face of unseen and safety-critical scenarios. However, real-world data collection and validation for these situations is dangerous and lacks the necessary scalability." [26]

With a case made for the value of curation of novelty in driving scenarios, in the next chapter, I introduce a method for this curation using language-based representations of visual data, and in following chapters, I continue the search for novelty through representations of perception model uncertainty and trajectory diversity.

Chapter 3

Towards Explainable, Safe Autonomous Driving with Language Embeddings for Novelty Identification and Active Learning: Framework and Experimental Analysis with Real-World Data Sets

3.1 Introduction: Novelty in Autonomous Driving

Unique failure cases of autonomous vehicles frequently make current news headlines, sometimes for their absurdity, other times for their tragedy; together, such news highlights that there are many situations autonomous vehicles are unable to navigate [27], and sometimes with grave consequence.

We can imagine, as human drivers, certain situations which are unexpected and require careful decision-making; driving into a patch of intense and sudden fog, interacting with a construction worker guiding a detour around an active site, pulling over safely when an ambulance needs to pass or police officer needs our attention, airport construction changing the contour of the usual dropoff and pickup zones, etc. In these cases, for an autonomous system trained to adhere to lane flow and avoid obstacles may be missing the higher-level reasoning abilities required of a human driver, and may, rightfully, provide a human takeover request [2,3,28]. But, how does the system recognize when such a control takeover is necessary, especially when a metric like time-to-collision oversimplifies the problem of safety for complex scenes?

In this case, it becomes important for the system to have an onboard method of *novelty* detection, recognizing when an unfamiliar or uncertain scene is presented.

The benefit of novelty detection does not stop at takeover requests; novelty detection serves a dual purpose in active learning. Active learning systems seek to select training data from a large, unlabeled pool to make machine learning more data-efficient. These methods are broadly classified by their acquisition functions into those which select data based on uncertainty and those that select data based on novelty.

Why research methods of sampling based on novelty instead of uncertainty? Uncertainty-based methods deal with the task-specific confidence of models in localizing or classifying objects or task-related instances. On the other hand, the novelty method proposed in this paper handles input at the scene level, observing the field of view agnostic of the number of objects proposed by a specified task learner. This provides the dual purpose of novelty detection to initiate takeover responses for safety, rather than a measure of effectiveness of an object detector.

Further, even as exemplified in the second paragraph of this article, we are able to express our scene understanding (in particular, describing novel features) through the modality of language, as illustrated in Figure 3.1. We propose that a language-based representation of a scene is a useful representation for novelty detection and, by extension, active learning.

In this research, we present an experiment by which we organize a large autonomous driving dataset into sets united by presence of notable features, and use clusters of language descriptor embeddings to identify scenes as novel. Having a language-based means of assessing scene complexity or novelty may be useful not only for handling model regime changes (autonomous modes for different settings) [29], human takeover requests (remote or in-cabin), and active learning methods for data collection, curation, and annotation, but also for doing the above in a way which may be explainable through decoding of language embeddings. We demonstrate this explainability by presenting an algorithm for generating text descriptions of what sets novel-identified scenes apart from their surrounding pool, leveraging large language

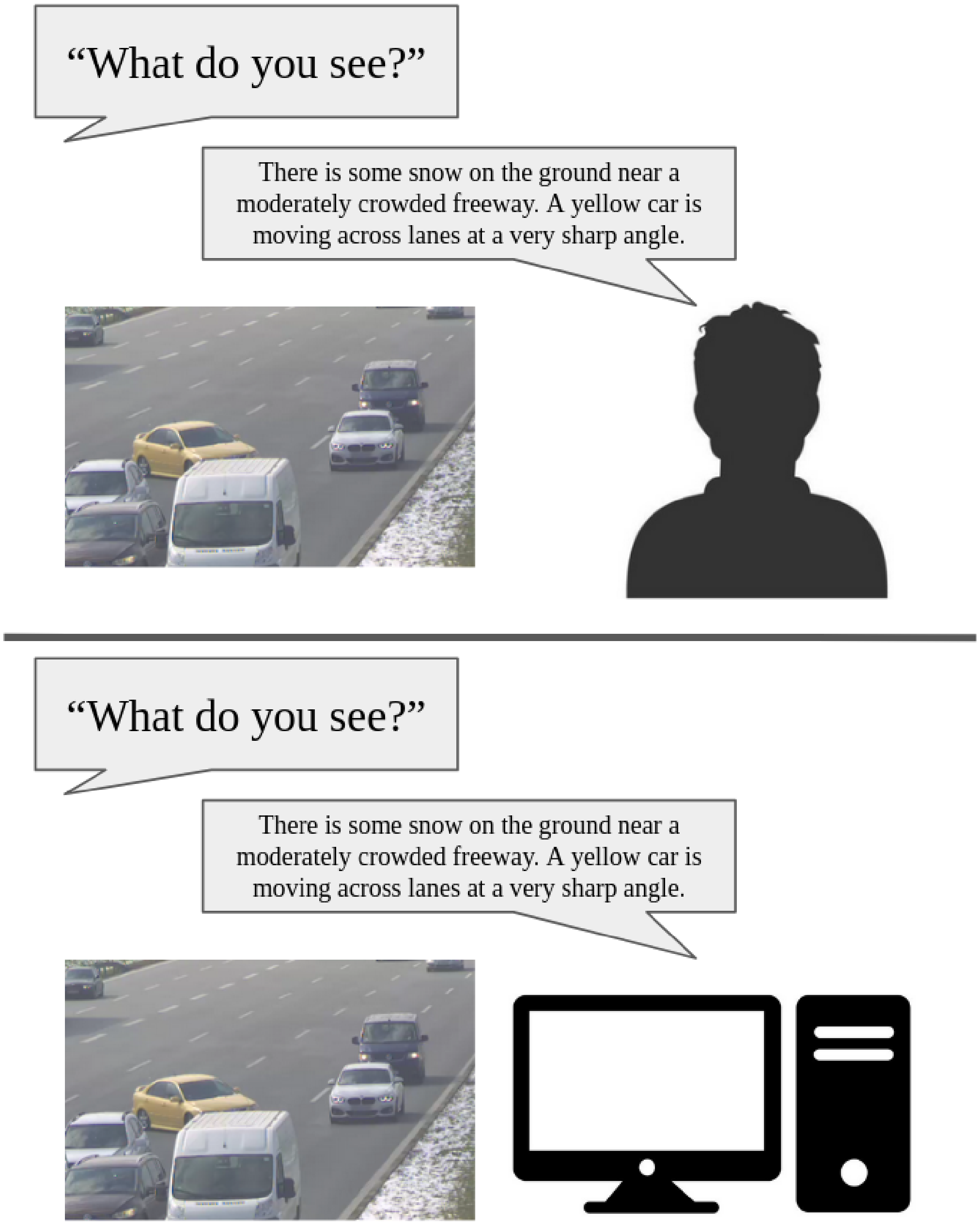

“What do you see?”
There is some snow on the ground near a moderately crowded freeway. A yellow car is moving across lanes at a very sharp angle.
“What do you see?”
There is some snow on the ground near a moderately crowded freeway. A yellow car is moving across lanes at a very sharp angle.

Figure 3.1. Natural language serves as a form of feature extraction, whereby data can be represented by meaningful description immediately understandable to a human reader. Such representations can also be generated by machines using vision-language models, and we present algorithms by which such representations (in both final and intermediate forms) can serve tasks of novelty identification in autonomous driving, useful towards anomaly detection and active learning tasks.

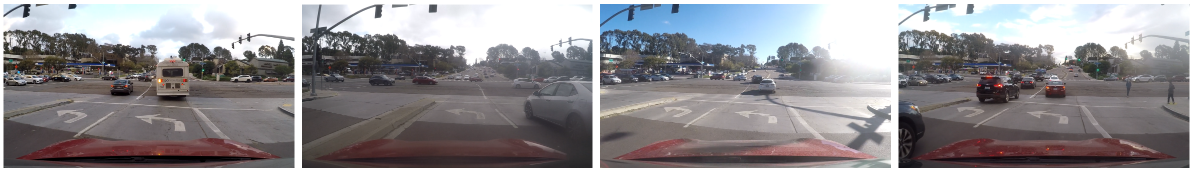


Figure 3.2. There are many important tasks to solve for the autonomous vehicle in this scene: detection of obstacles and external agents, prediction of agent trajectories for safe planning, and interpretation of traffic control elements for control decisions. For a limited data budget, at what point does it become more beneficial for a learning model to bring in new scenes instead of variants of old scenes? Does the information gain of data in new scenes exceed the information gain of variants of old scenes across all tasks?

and language-vision models in the process and providing qualitative results on the autonomous driving dataset.

3.2 Novelty as Active Learning

Here we adopt the definition of Cohn et al., where active learning is any form of learning in which the learning program has some control over the inputs on which it trains [1]. In their research, they qualify that “selective sampling is active learning"; they propose a method by which all samplings is done from the so-called *region of uncertainty*. In Figure 3.3, we adapt their original framing of query sampling to the larger, multi-task problem of safe autonomous driving. In the original framing, one of the largest problems the authors point out is that as a class model becomes more complex, it becomes difficult to compute an accurate approximation of the region of uncertainty. In this research, we propose language-embedding novelty as a suitable analogy to uncertainty for these purposes, avoiding the active learning collapse to random sampling.

In general, active learning acquisition functions can be separated into categories of model-dependent uncertainty measurement, in which a function quantifies uncertainty based on some task-dependent measurement from a model, and novelty measurement, in which data is sampled independent of the model on some other property or properties. One downside of using a model-dependent uncertainty-related acquisition function is that different tasks may select different data to be included in the task training pool. In the cases where the active learning

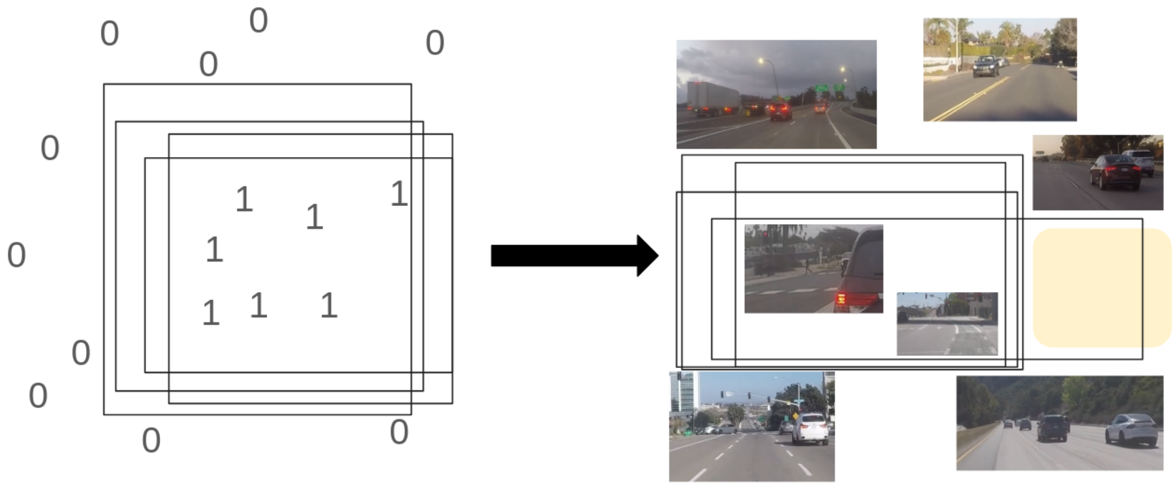


Figure 3.3. In [1], Cohn et al. use an abstract setting like the figure shown on left to suggest that there are many possible models (black rectangles) which could be used to classify the points, but that this model performance does not necessarily indicate a complete and accurate learning of the appropriate concept. By sampling in the spaces where the model may be uncertain, a stronger refining of the model boundary can occur, leading to improved generalizability. On right, we abstractly show how this manner of thinking might be applied to similar active learning for autonomous driving. In the center, we have scenes which contain pedestrians, as opposed to scenes without outside. A region shaded in yellow indicates a hypothetical region where the model could benefit from sampling, to narrow its hypothesis of what separates pedestrian scenes from others. However, the general problem of safe autonomy is much more complex, where multiple tasks (such as object detection, tracking, and localization) must all be met with high performance, and a point sampled as uncertain toward one task may be redundant to another. Further, the high-dimensional nature of the data does not reduce to such an easily-separable space. In this research, we propose that language-based embeddings of scene images are a useful reduction for identification of novel qualities, on the premise that sampling novelty may be useful towards multi-task model improvement.

method may be driving large-scale data collection, curation, and annotation, it is better that the expensively acquired data be strongly beneficial to many required tasks [30]. While acquiring data based on a novelty heuristic may not guarantee optimality for a particular task, its task independence may be useful in serving a variety of models simultaneously. As another benefit toward a novelty-based method of active sampling, it has been shown that under low data budgets, sampling typical examples gives the greatest performance gains, but beyond a certain budget (which would reasonably be expected of a safety-centric autonomous driving system), learning gains actually come from the sampling of *atypical* examples [31].

There are a variety of strategies toward identifying novel samples in the data pool for inclusion in the training set; a prototypical approach may include handcrafting a descriptor of each sample, and using some unsupervised method, such as clustering or overfitting single-sample learners, paired with a thresholding function, to identify what is most dissimilar to what is already in the training set. We show an example of such a method in Figure 3.4, where a feature vector of each sample image is mapped to some latent space, and included in the training pool if satisfactorily distinct from existing training points. In the methods presented in this paper, rather than using a handcrafted feature descriptor for each sample, we propose using a pre-traininied language-based feature descriptor, as such models are effective toward captioning (i.e. describing and explaining) visual input. Such a method assumes that details which differ between samples are distinct enough that they may be described and distinguished verbally from their image representations.

3.2.1 What makes autonomous driving imbalance different than other class imbalance problems?

Much of machine learning research treats the class imbalance problem as an issue of having feature-represented and labelled samples to classify, with some classes appearing more often than others [32]. In our domain, the problem is at a different level of abstraction. Each driving scene is unique with its own high-dimensional fingerprint, and there does not exist a

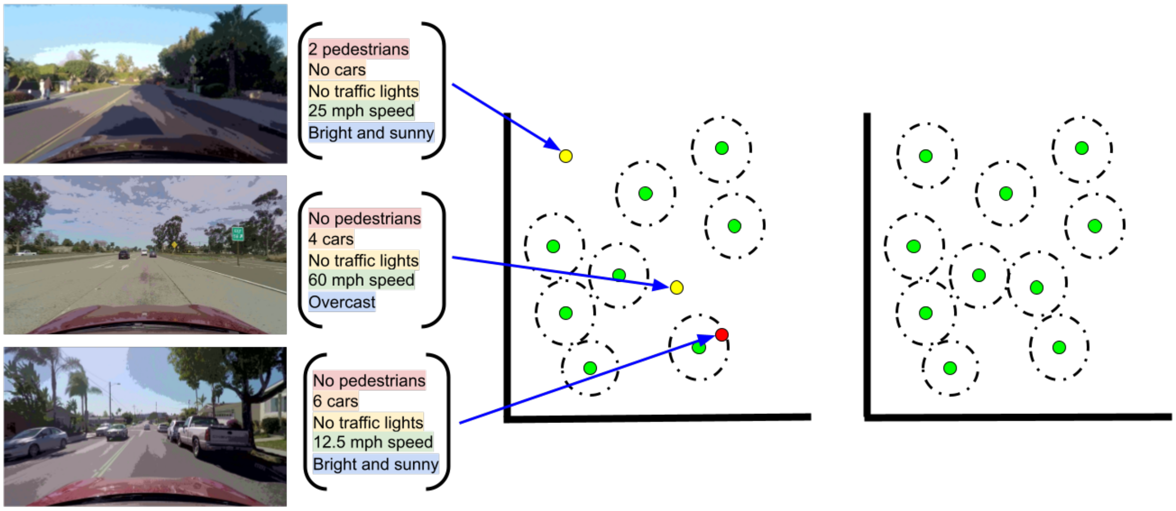


Figure 3.4. If we view deep learning (and machine learning in general) as a process by which parameters algorithmically extract useful features from data (by means of converting data from its original structure to a structure of abstract, lower-dimensional, intelligent meaning), then we can consider each data point to be projected into a variety of spaces of varying dimension throughout the forward process. For a model to be successfully fit to its task (i.e. not overfit nor underfit), at some point, the data must reach a meaningful, useful projected representation. An example projection is depicted in the two graphs on right. Presumably, each point carries with it some “coverage" of the latent space, shown with a black radius, such that similar points not found in the training set would receive similar prediction by the model. When we add new data to train a model, such as the candidates shown in yellow and red in the middle graph, we would like to be efficient, adding only data which improves the model’s coverage of the problem latent space. The driving question of this research is: what descriptors or features make a useful representation, such that an algorithm can quickly identify points which are less useful (such as the point shown in red)? Do these descriptors come from high-level abstract meaning, as we show on the left with human-understandable features like number of pedestrians, speed, and weather? Or, should these descriptors emerge from an embedded, learned feature directed from the raw sensor input and the model’s own transformations of this input, trading explainability for optimality? How can these descriptors be leveraged towards active learning, and what implications do these choices make towards curating and annotating such datasets?

standard and fixed taxonomy by which we sort driving encounters. Even in driver monitoring alone, the problem is considered *open-set* due to its real-world placement and the natural ability of drivers to be creative, independent agents, who may make decisions to hold an object, maneuver through a trajectory, or drive to a location that has never been observed before [33]. In the words of Calumby et al., “[L]ow-level visual features are usually not able to properly describe the rich semantic intent of a query nor the high-level concepts found in the images of a collection (the well-known semantic gap)." [34]

There are a multitude of approaches that can be taken to resolve this *over-representation problem*, but there also exists a necessary relationship between the solution and the intended task’s data-driven method. For example, a technique as simple as filtering to limit records from a particular GPS coordinate may be helpful to ensure a geographical spread, which might be helpful for mapping traffic signs and lane systems, but such an approach does not help for a task around estimating traffic flow or predicting driver lane change behavior, where scene factors like traffic density and speed play a greater role than geographic location.

Methods which reallocate learning priority to samples to turn a distribution from unbalanced to uniform are at a non-start, because there is not such a distribution framework to draw from (abstractly) from these enormous, high-dimensional datasets. Low-level descriptive scene features such as lighting and ego position can be readily extracted from the raw data, but many notable features which make a driving scene ‘novel’ exist as high-level descriptors, such as driving maneuvers [35]; presence, location, and count of surrounding pedestrians and vehicles; and irregular road events [36]. Thus, we propose the development of such a taxonomy as a valid intermediate step, such that the wealth of research in low-level data imbalance methods can be applied and explored. This would enable the use of standard methods such as class-balancing oversampling and undersampling, and weighted loss functions which associate higher loss values with data derived from safety-critical or under-represented scenes. A natural question for this domain is, should such a taxonomy be explicitly defined in explainable form, or can a latent, self-organized representation of all driving scenes be learned that creates an informative sampling space? We propose here that the latent embeddings which encode language suffice to form this organized space, building from the assumption that there are observable patterns in the data that we can use towards our decision, and that the words that we use to describe a scene may help point towards features we have not seen before. A collateral benefit of such a representation is the ability to explain data inclusion through language itself.

3.2.2 Data Imbalance from Scene Redundancy

To motivate this style of learning, consider the scene shown in Figure 3.2; the data collecting vehicle repeatedly visits the same intersection. At some point, the vehicle will have observed a great variety (perhaps a near-exhaustive variety) of scene agent configurations, vehicle types, and visibility conditions at this location. Once the location ceases to be novel, is the vehicle's time (and data capture) better spent in another location to improve its driving abilities?

Data sampling methods are commonly used to overcome data imbalance, such as random under-sampling (to remove majority cases from training data), and random over-sampling (having under-represented classes appear more frequently during training). In principle, standard data augmentation serves this same purpose, on the basis that the collected data is has sufficient examples of prototypical data but under-represents the variance of the complete population of data along some parameter which is being augmented for (e.g. lighting, translation, reflection). Naturally, augmentation methods can be applied to minority-class data to build a stronger representation within a training dataset, but this relies on sufficient examples of the minority-class's principal patterns. By sampling for novelty, our method may introduce new instances of minority-class data by providing only data which can be described or captioned in a way unlike what is already in the training set.

Because autonomous driving data is heavily multimodal, polling multiple modes for uncertainty is complex; selecting data which supports learning is not only task-dependent, but even modality dependent, which makes the task of guiding data collection for improved learning outcomes even more difficult when certain sensors have disagreement on what regions of a map or types of encounters carry the most uncertainty within their respective data modality.

3.2.3 Solutions in Active Learning

Active learning is the process by which a learning system interactively selects which data points should be added from the unlabeled data pool to the labeled training set, assisted by the intervention of a human expert providing associated annotations [37]. If this process is done with no information about the model, we refer to this as *data curation*. In the data cycle, such a step naturally exists between *collection* and *annotation*.

For the purpose of active learning, low-level descriptive scene features such as lighting and ego position can be readily extracted from the raw data, but many notable features which make a driving scene 'novel' exist as high-level descriptors, such as driving maneuvers [35]; presence, location, and count of surrounding pedestrians and vehicles; and irregular road events [36]. Accordingly, in this research we investigate feature definition, extraction, and effectiveness for active learning algorithms.

How does active learning relate to these problems? We can view active learning as a method of intelligent oversampling. In this frame, the range of knowledge which the model has learned serves as a training “majority", while knowledge the model has yet to learn serves a training “minority". In the process of determining which samples to draw from the available (unlabeled) data pool, we intend to oversample from those which are underrepresented in the training data.

3.3 Related Research

3.3.1 Diversity and Novelty

To clarify between related active learning sampling concepts, [38] categorizes data by informativeness (have the most uncertainty as viewed by a particular model), diversity (minimal redundancy between like-data, e.g. maximizing angle between representation for angular metrics), and representativeness (measure of similarity of one unlabeled data point to the rest of the unlabeled pool). As an example, Calumby et al. [34] re-rank images for retrieval by text queries by seeking to increase diversity of returned sets using visual and textual descriptors so that the system can better learn relevant retrieval from human feedback. In this research, we explore the related concept of novelty, which we may conceptualize as a neighbor to representativeness; where representativeness assesses an unlabeled datum's ability to represent others in the unlabeled pool, our novelty assesses an unlabeled datum's ability to different than the labeled set. Liang et al. [39] even show that active learning with sampling based on spatial and temporal diversity (i.e. drawing samples from non-overlapping locations and times) show improvements in 3D object detection on the NuScenes dataset. Elhafsi et al. [40] show that language models can be effective in finding significant semantic anomalies in simulated autonomous driving and robotic manipulation.

Novelty is useful not only in efficient learning paradigms, but also in direct safety applications. For example, the measurement of Bayesian surprise (or KL divergence between an expected and observed distribution) has been used to detect novelty in the form of unexpected obstacles for autonomous driving of a warehouse robot [41]. The ability of an autonomous system to recognize novel or unfamiliar settings also allows such systems to request human intervention or guidance, especially important for safety [42, 43]. Currently, graph-based methods comprise the state of the art in autonomous driving, and the heterogeneity of data sensors and corresponding methods, as well as the formalization of sufficient ontologies to capture the nuances and complexity of real-world scenarios, make this an important open safety challenge [11, 44].

3.3.2 Explainability

The integration of interpretability/explainability and active learning has been considered in prior research; for example, Mahapatra et al. [45] use interpretability salience maps from training a model for classifying lung disease from chest x-ray images, and actively selecting samples classified to the highest level of ‘informativeness’ from these maps. Language has been shown to be a promising medium of explainability in autonomous driving, for tasks such as scenario interpretation [46], decision-making [47], and intention prediction [48], even allowing for passenger queries to these systems.

3.3.3 Efficient Learning

Learning from non-task-specific features is a characteristic of self-supervised learning; as an example, Saeed et al. [49] show the ability of a model to learn semantic representations of accelerometer data in an unsupervised way through transformation recognition networks, leveraging the invariance (or, known alterations) of signals through certain transformations, then using this learning for human activity recognition. Rather than transferring the learned patterns directly from the non-task-specific pretraining, in our presented research, we instead utilize these representations of data directly as a means of active selection of informative samples. These methods share in common a benefit toward multi-task learning.

Li and Guo, discussing model uncertainty-based active learning [50], state, with our added emphasis:

> These works however merely evaluate the informativeness of instances with most uncertainty measures, which assume an instance with higher classification uncertainty is more critical to label. Although the most uncertainty measures are effective on selecting informative instances in many scenarios, **they only capture the relationship of the candidate instance with the current classification model and fail to take the data distribution information contained in the unlabeled data into account**. This may lead to selecting non-useful instances to label. For example, an outlier can be most uncertain to classify, but useless to label. This suggests representativeness of the candidate instance in addition to the classification uncertainty should be considered in developing an active learning strategy.

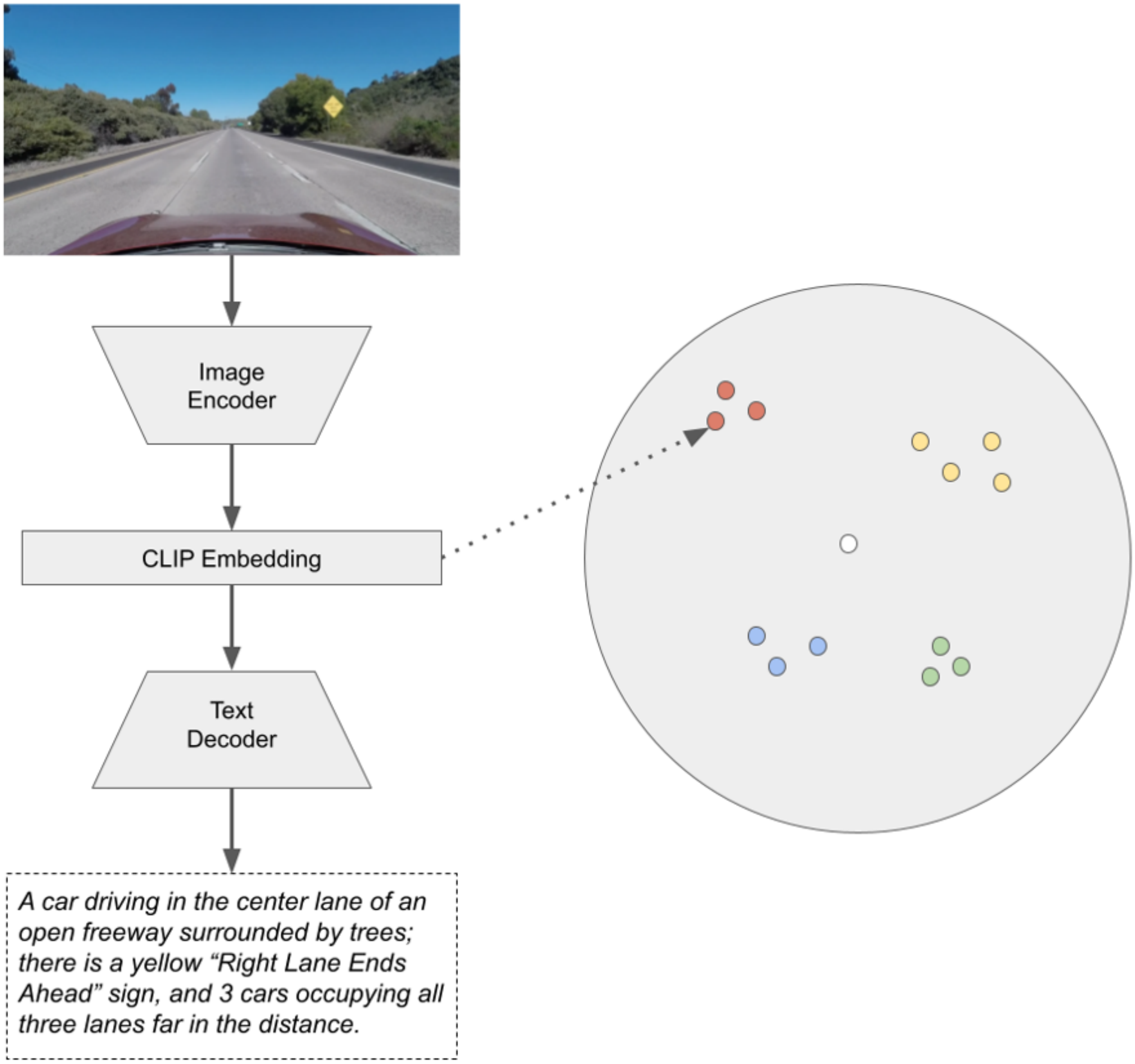
Because there are so many models which must operate successfully over the same data for safe autonomous driving (e.g. lane detection [51], 3D object detection [52], sign and light recognition [53–55], multi-object tracking [56], path planning [57–59], trajectory prediction [60–66], intention prediction [35, 67, 68]), having data which supports all models is necessary, but impractical when the sampling method depends on any one task or model. By leveraging language-based descriptors of the data itself, we do not sample using model uncertainty, but rather from the representativeness of a data point in relation to all other data points.

CLIP (Contrastive Language–Image Pre-training) [69] is a multi-modal neural network architecture trained on a wide variety of images and associated language description. Its pretraining allows it to adapt to a variety of zero-shot learning tasks [70], with a multitude of applications in image search and retrieval. It typically uses two Transformer backbones; one which acts as an image encoder and another which acts as a text encoder, projecting the features to a shared vector space. Images are handled by splitting into non-overlapping patches, linearly embedded, and concatenated with positional encodings. During training, contrastive loss is used to maximize the similarity (dot product) between encodings of image-text pairs:

$$\frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}, \tag{3.1}$$

where **a** is the image encoding and **b** is the text encoding.

The pre-trained representations in the CLIP model have been shown to be effective at a variety of zero-shot learning tasks, such as 3D object detection, classification, and segmentation, by combining textual features with standard point clouds and depth maps prior to performing the detection, classification, or segmentation tasks [71]. Impressively, embodied AI agents which use CLIP information can even autonomously navigate to objects that were not used as targets during training [72]. By using learned language embeddings, such a system acts as a multi-label learner (i.e. data may have more than one class label, which a model should be able to assign simultaneously), which have been effective for active multitask learning in prior research [73,74].



A car driving in the center lane of an open freeway surrounded by trees; there is a yellow "Right Lane Ends Ahead" sign, and 3 cars occupying all three lanes far in the distance.

Figure 3.5. An overview of the method presented in this chapter. Scene images from a pool of driving scenarios are input to a Contrastive Language-Image Pretrained image encoder. The resulting embedding *could* be used in a text decoder for image captioning, but instead, we perform clustering over the resulting embedding vectors from a large pool of samples, as shown at right. Images whose representation appears independent of the identified clusters, such as the one in white at the center of the representation space, are considered to be novel. The experiments shared in this research describe whether or not the novelty identified by this method aligns with the concepts of novelty reflected in the organization of the datasets.

Algorithm 1: Image Encoding and Clustering

```
Data: Set of images $\mathscr{I}$
Result: Novelty set $\mathscr{N}$
Step 1: Encode all images into vectors using CLIP model
for each image $I$ in $\mathscr{I}$ do
    $v_I \leftarrow$ CLIP_encode($I$);
Step 2: Cluster vectors using hierarchical clustering with threshold $t$
$\mathscr{C} \leftarrow$ Hierarchical_Clustering($\{v_I\}$, $t$);
Step 3: Add unclustered vectors to novelty set $\mathscr{N}$
for each vector $v$ in $\{v_I\}$ do
    Add $v$ to $\mathscr{N}$ if $v$ is not in any cluster;
```

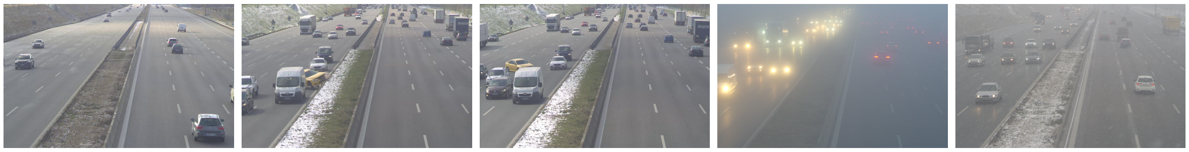


Figure 3.6. Images from each set of the TUMTraf dataset. From left to right, the figure shows normal traffic, accident, pre-accident, dense fog, and snow scenes.

3.4 Algorithm for Novelty Identification by Clustering over CLIP Embeddings

We present our algorithm for novelty identification in Algorithm 1, with illustration in Figure 3.5. This algorithm is used to create a set of novel scenes from a group of scene images. While the presented algorithm utilizes the pre-trained CLIP encoder and hierarchical clustering, the same procedure can be applied for alternative descriptor vectors and clustering algorithms.

3.5 Experimental Evaluation

3.5.1 Datasets

LAVA

For this experiment, we sample scenes from the LAVA dataset [75]. We define 12 sets of data, each containing 500 images:

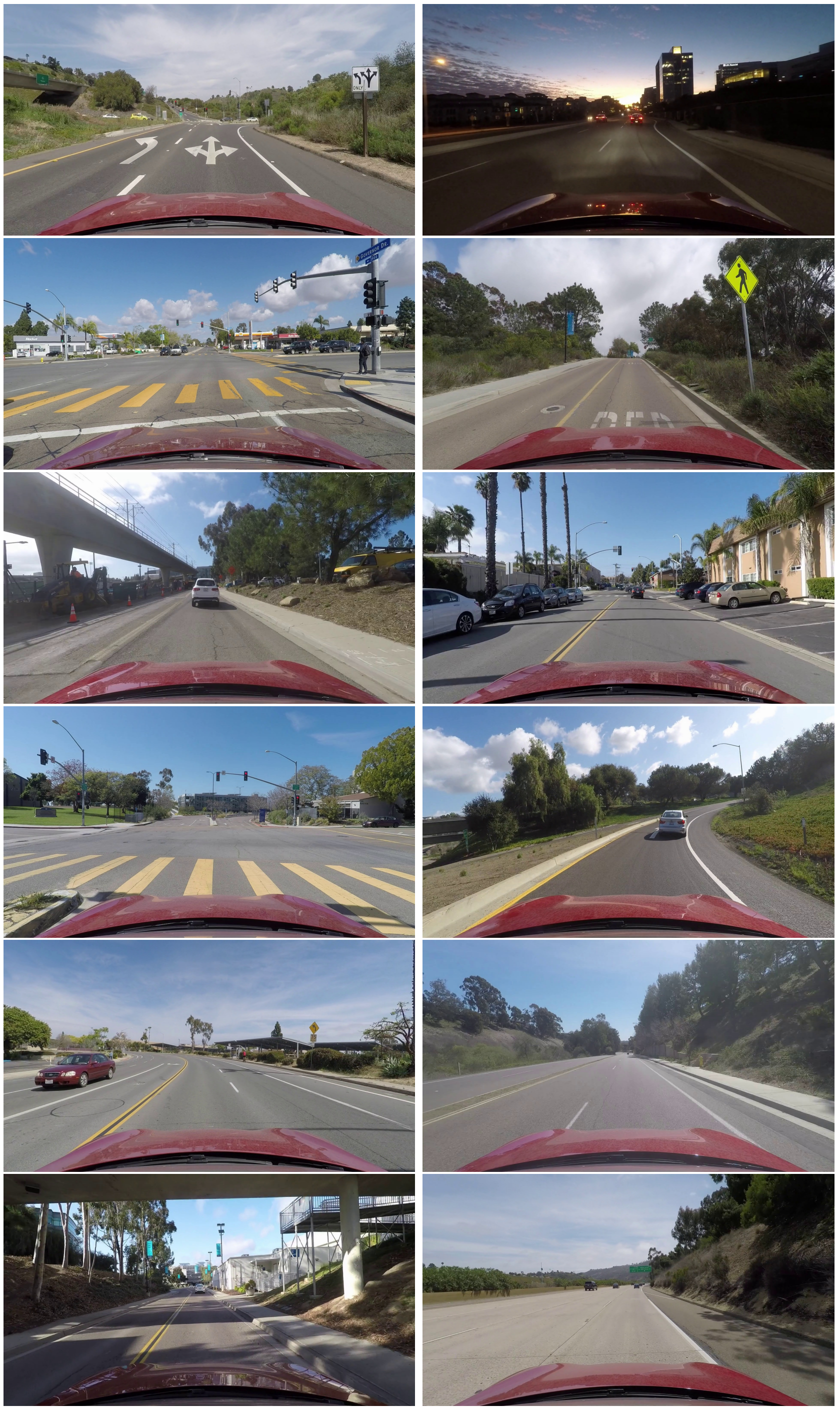


Figure 3.7. Images from each set and opposite set are shown next to each other. Some features are easier to spot-the-difference than others. In order from top to bottom, the figure shows day-night, with/without pedestrians, with/without construction, with/without traffic lights, with/without traffic signs, and on/off college campus.

1. Scenes with street signs,
2. Scenes without street signs,
3. Scenes with active construction signs and/or workers,
4. Scenes without active construction signs and/or workers,
5. Scenes captured around a college campus,
6. Scenes captured away from a college campus,
7. Scenes captured during daytime,
8. Scenes captured at night,
9. Scenes with traffic lights,
10. Scenes without traffic lights,
11. Scenes with pedestrians, and
12. Scenes without pedestrians.

Representative images from the sets are shown in Figure 3.7.

We note that these sets vary in level of abstraction; some contain specific objects, and others contain a higher-level idea not necessarily exemplified by the presence of a particular object. Further, some sets are defined on the presence of an object, while others are defined on the absence of those objects.

Each of the sets above has a clear antithesis set. Using this property, we create twelve *near homogeneous* sets, where each set contains its original 500 images, plus one image randomly sampled from its antithesis set. This additional image, within the near homogeneous set, is guaranteed to be novel on the feature which defines the set.

TUM Traffic

While the LAVA dataset is taken from a vehicle-mounted camera, we perform another set of experiments from the infrastructure-mounted cameras of the TUM Traffic (TUMTraf) dataset [76] [77], which observes freeway activity along the A9 autobahn in Germany. This dataset also includes a rare traffic accident event. We isolate the following subsets of data:

1. Scenes in normal traffic (175 images),
2. Scenes in dense fog (358 images),
3. Scenes in snowy conditions (375 images),
4. Scenes just before a traffic accident, and
5. Scenes just after a traffic accident.

In the case of “Scenes just before a traffic accident", we include images where it would be evident to an omniscient observer that something so anomalous is happening that an accident is surely to occur in the near future, illustrated in Figure 3.6. For the before-and-after accident scenes, since there is only one accident occurence, we only form two sets from this data: all normal + one pre-accident, and all normal + one accident. For the other two novelties (snow and fog), we form all normal + one snow, all normal + one fog, and their opposites all snow + one normal and all fog + one normal. Examples of images from each scene type are shown in Figure 3.6.

3.5.2 Implementation Details

For CLIP encoding of images [74], we utilize the Vision Transformer (ViT) backbone [78], with the “large" model size and image patches of size 14x14 pixels (in general, smaller patch sizes require more total model parameters, but may lead to better performance). We use an embedded vector size of 512.

Table 3.1. LAVA Experiment Results

Set Category	Set Size with Novel Element
Without Traffic Signs	3
Without Construction	2
Around College Campus	2
Away from College Campus	1
Daytime	2
Nighttime	3
Traffic Lights	4
Without Traffic Lights	2
Without Pedestrians	3

Table 3.2. TUMTraf Experiment Results

Set Category	Set Size with Novel Element
Normal (One Accident)	1
Normal (One Pre-Accident)	1
Normal (One Snow)	1
Normal (One Fog)	1
Snow	1
Fog	1

We compute the cosine distance

$$\arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{||\mathbf{a}||\cdot||\mathbf{b}||}\right) \tag{3.2}$$

between each pair of vectors for clustering, and apply the hierarchical clustering algorithm [79, 80]. We use the average distance of all points in a cluster in re-assigning cluster distances when constructing the dendrogram (i.e. unweighted pair group method with arithmetic mean). A threshold τ is applied to estimate the flat clusters, such that the cophenetic distance between any pair within one of the flat clusters is no greater than τ. We explore values of τ between 0.22 and 0.75 empirically, and optimize for each trial for this experiment, selecting values between 0.35 and 0.65 depending on the experimental set.

3.6 Zero-Shot Novelty Classification Results

Results of our LAVA experiments are provided in Table 3.1 and results of our TUMTraf experiments are provided in Table 3.2. In the data pool for each set category, one element belongs to the opposite set. The column at right describes the size of the algorithmically-determined “novel set" which contains this one unique element (as well as any true set elements classified as “novel"). In the ideal case, only one element (i.e. the novel element) would remain unclustered at the end of the algorithm, and in the worst case, 500 elements would be unclustered (i.e. the algorithm considers all elements unique). Our values are promising; on the LAVA dataset, novel set sizes range from 1 to 88, with an average size of 14 (approximately 3% of the available data pool). On the TUMTraf dataset, ***all*** novel set sizes are 1! This indicates that the algorithm is able to isolate, based on our set construction criteria, the unique element of the set without making false-positive novelty identifications.

Further, we observe that in general, the algorithm is more successful at identifying the *presence*, rather than the *absence*, of its defining property. This is naturally reflected in language; when humans describe a scene in natural language, we describe what the scene contains, not the long list of everything *not* found in the scene. Notable examples, reflected in the Challenge Cases in Table 3.3, include difficulty in identifying that one sample was missing traffic signs (novel set size of 35, as opposed to 3 when finding the one that *did* have a traffic sign), pedestrians (novel set size of 88, as opposed to 3 when finding the novel set that *did* have a pedestrian), and construction (novel set size of 17, as opposed to 2 when finding the novel set that *does* feature construction). We note that the identification of the scene without pedestrians was made especially hard by the inclusion of 3 nearly identical images that did feature pedestrians (same neighborhood, in the distance) as illustrated in Figure 3.8; considering the similarity of the target image to the other three, the fact that these four were *not* clustered at the point when the target image was labeled ‘novel’ is great. We also note that the construction category may be difficult by the fact that many elements that define a construction site (cones, signs, and people wearing

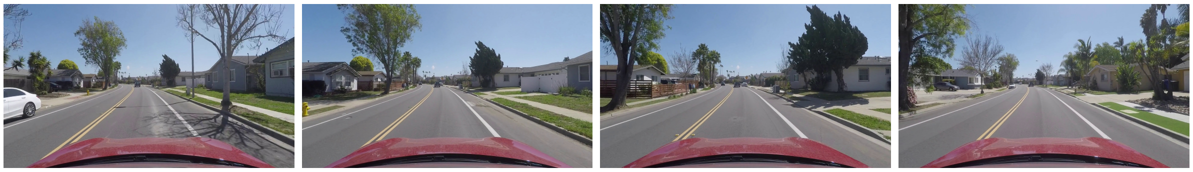


Figure 3.8. An example of a challenge case; the dataset contains four images which are nearly identical (among the 500 total), and the pedestrians when present are far in the distance, making the scene in the same neighborhood (albeit with no pedestrians) different to discern as unique.

Table 3.3. Results on Challenge Cases

Set Category	Set Size with Novel Element
Pedestrians	88
Traffic Signs	35
Construction	11

orange) may also be found in non-construction scenes, making it more of a challenge to identify the construction scene as particularly unique, since it is the combination of all these elements that creates this uniqueness. Further, chance “novelty" also appears in some of these datasets, such as a rare nighttime scene occurrence in an otherwise mostly-daytime set.

As an unexpected but exciting result, we also found that in some cases, the additional samples “mistakenly" marked as novel were in fact novel for a different reason: the camera became occluded due to rain, fog, light saturation, or motion blur. We show some of these interesting novelty detections in Figure 3.9, which, for purposes of novelty detection, we would consider to be unexpected successes of the algorithm.

3.7 Machine Explainability

Rather than trusting the machine to identify novelty correctly using language embeddings, we add one further layer of explainability to our experiment: we ask the machine to state what makes the selected ‘novel’ image different from all other clusters. Consider all observable features of the scene, we would like to find:

$$F_{novel} \setminus (F_1 \cup F_2 \cup \ldots \cup F_N), \tag{3.3}$$

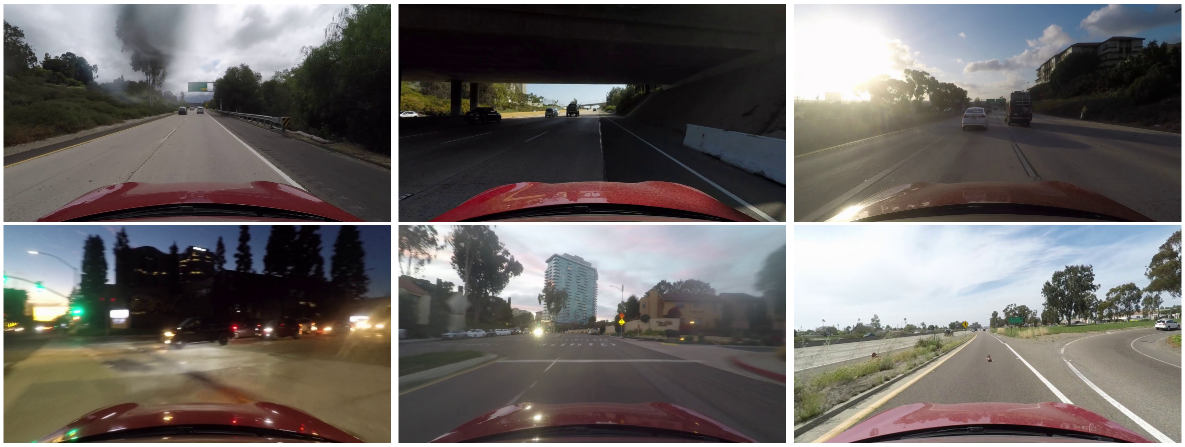


Figure 3.9. Certain images were not novel along the intended set quality, but were nonetheless novel to their set. Especially promising is that some of these novel captures reflect a failure or occlusion of the sensor, rather than a novel scene element, suggesting that these embedded representations may also be useful in providing information about the sensor state. The above images include cases of condensation blur, passing under a bridge, light saturation, motion blur, and even surprising debris in the vehicle's path.

where F_{novel} is the set of observable features in the novel scene, and F_i indicates the set of observable features from scene i, from the total pool of N scenes, excluding the novel scene.

The Large Language and Vision Assisnt (LLaVA) is an end-to-end trained large multimodal model that connects a vision encoder and LLM for general-purpose visual and language understanding [81]. This multimodal model forms the basis for the decoding of our images from their visual embedding to a language form. We use the Mistral 7-billion parameter LLM [82] as our text embedding backbone[1]. After generating text associated with images, we use the GPT-3.5 LLM model from OpenAI to connect information between images, prompting the system to identify what features from the "novel" image distinguish it from the other images in its pool.

Encoding all observable features of an image to a textual description provides our first loss of information (essentially the opposite action of the adage "A picture is worth a thousand words"). Referring to our text-described features as T, we now update our goal as:

$$T_{novel} \setminus (T_1 \cup T_2 \cup \ldots \cup T_N), \tag{3.4}$$

[1] The algorithms we present can be used with even stronger backbones for systems with more computational power.

where T_{novel} is the set of text-described features in the novel scene, and T_i indicates the set of text-described features from scene i, from the total pool of N scenes, excluding the novel scene.

We now reach an interesting limit, illustrated in Figure 3.10. The more images we compare to, the more of our (language-limited) information we may exclude from the possible description of novelty. However, we still need to compare to enough images so that only the novel features are left in the description. Fortunately, to mitigate this tradeoff, we can leverage the clustering that has already been performed on the image embeddings; we assume that each cluster is united on some feature(s), and that by selecting an element from each cluster, we may effectively sample for that feature, thereby eliminating that feature as a possible novelty of the novel image.

With this, we update our goal once more as:

$$T_{novel} \setminus (T_{c1} \cup T_{c2} \cup \ldots \cup T_{cn}), \tag{3.5}$$

where T_{novel} is still the set of text-described features in the novel scene, and T_{ci} indicates the set of text-described features from one image of cluster i, from the total pool of $n < N$ clusters, excluding the novel scene.

Algorithm 2: Generating Explanation of Scene Novelty

```
Data: Novel scene image, clustered scene images, language-vision model, LLM
Result: String description explaining what is novel in the input scene relative to the
        other scenes
Generate a detailed description of the novel scene using the language-vision model;
foreach cluster of scenes do
    Sample one scene from the cluster;
    Generate a detailed description of the scene using the language-vision model;
Prompt the LLM to identify what makes the novel image description different from
  all other images;
Return description explaining novelty.
```

This procedure is summarized in Algorithm 2. We also provide discussion of further

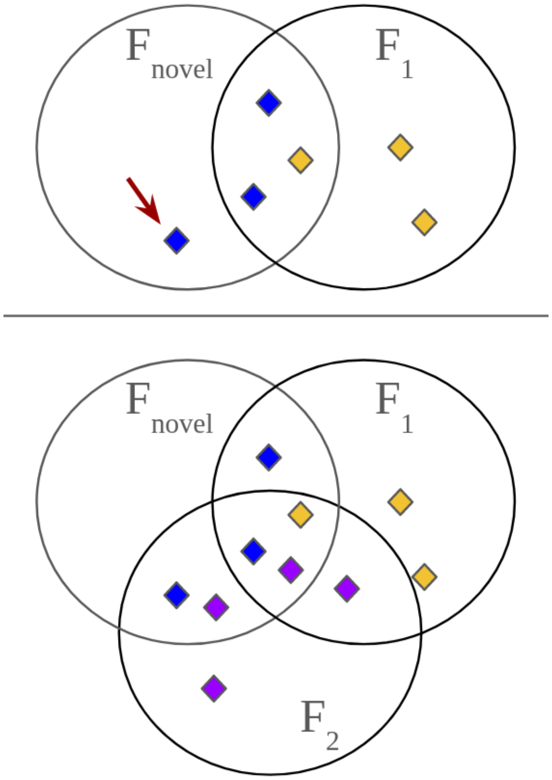


Figure 3.10. In attempting to identify the features which make one scene novel from the rest, there is a tradeoff induced by the reduction of images to a text space. Each scene's observable feature set is represented by a circle. Only a discrete number of those features may also be represented by generated language descriptions, indicated as colored diamonds associated with each feature set. In the top scenario, we see that by accounting for commonalities, it is possible to identify a remaining language-describable feature available to explain the novelty of the novel scene (identified by the red arrow). However, in the bottom scenario, by introducing another scene into the comparison, we have eliminated all language-describable features. In the ideal scenario, we have an infinitely-strong vocabulary to fully describe the set of all observable features, making this a non-issue, but to overcome the challenges still present in state-of-the-art vision-language models, we present a sampling algorithm to allow for explainable results of scene novelty.

enhancements in the Future Work section, using repeated sampling for more robust descriptions of novel elements.

We utilize this algorithm to generate an explanation for what makes each of the novel set elements novel, and reach the qualitative descriptions presented as captions next to each image in Figures 3.11 and 3.12. In addition to identifying the novelty that we constructed into the sets, we also provide examples where the algorithm identifies other sources of novelty, shown qualitatively in Figure 3.13.

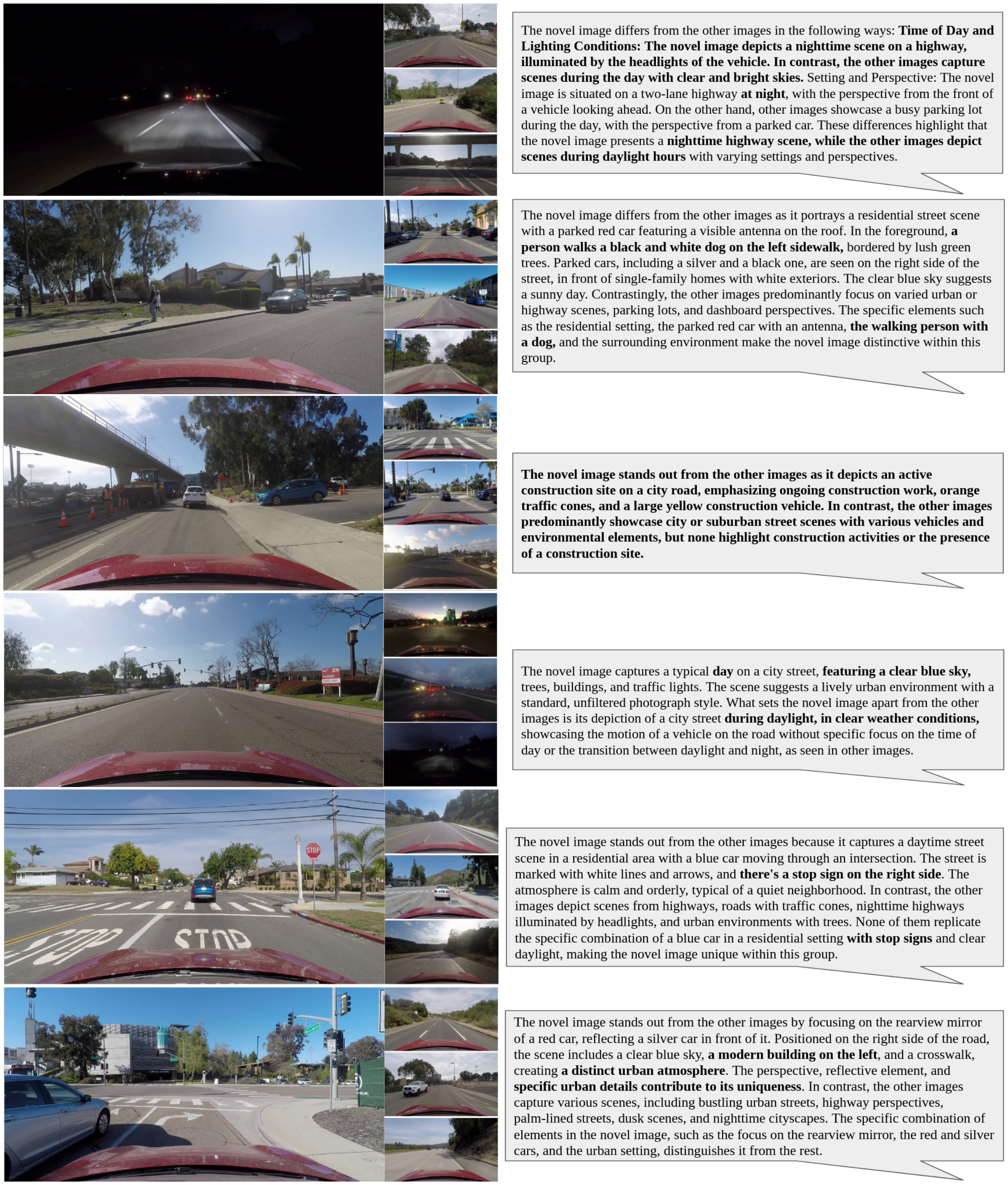
3.8 Concluding Remarks

The real world is an open set; there will always be new elements, and things that appear in unexpected ways. We cannot create a discrete class system which accurately accounts for (and describes) the variety of what we might encounter while driving; yet, we can identify when we are encountering something new, and we can find ways to describe our encounter with natural language. For these reasons, the use of language-driven embeddings as a means of novelty detection provide great promise toward continued development in safe takeovers, data curation, active learning, and explainability.

3.8.1 Future Research

In this research, we show that language embeddings are sufficient for identifying novelty in a collection of datasets. As a next step towards understanding the role of this novelty in active learning, future work should apply this novelty measure as a means of selection for elements to add to the training pool for a large autonomous driving dataset, preferably training on multiple tasks with the same pool, as a means of measuring improvement in multi-task active learning [30].

In generating explanations of novelty, we recommend use of the evolving state-of-the-art as the modular language-vision model and LLM within our algorithmic framework. Further, as the field of visual question answering (VQA) and image difference description continues



The novel image differs from the other images in the following ways: **Time of Day and Lighting Conditions: The novel image depicts a nighttime scene on a highway, illuminated by the headlights of the vehicle. In contrast, the other images capture scenes during the day with clear and bright skies.** Setting and Perspective: The novel image is situated on a two-lane highway **at night**, with the perspective from the front of a vehicle looking ahead. On the other hand, other images showcase a busy parking lot during the day, with the perspective from a parked car. These differences highlight that the novel image presents a **nighttime highway scene, while the other images depict scenes during daylight hours** with varying settings and perspectives.

The novel image differs from the other images as it portrays a residential street scene with a parked red car featuring a visible antenna on the roof. In the foreground, **a person walks a black and white dog on the left sidewalk,** bordered by lush green trees. Parked cars, including a silver and a black one, are seen on the right side of the street, in front of single-family homes with white exteriors. The clear blue sky suggests a sunny day. Contrastingly, the other images predominantly focus on varied urban or highway scenes, parking lots, and dashboard perspectives. The specific elements such as the residential setting, the parked red car with an antenna, **the walking person with a dog,** and the surrounding environment make the novel image distinctive within this group.

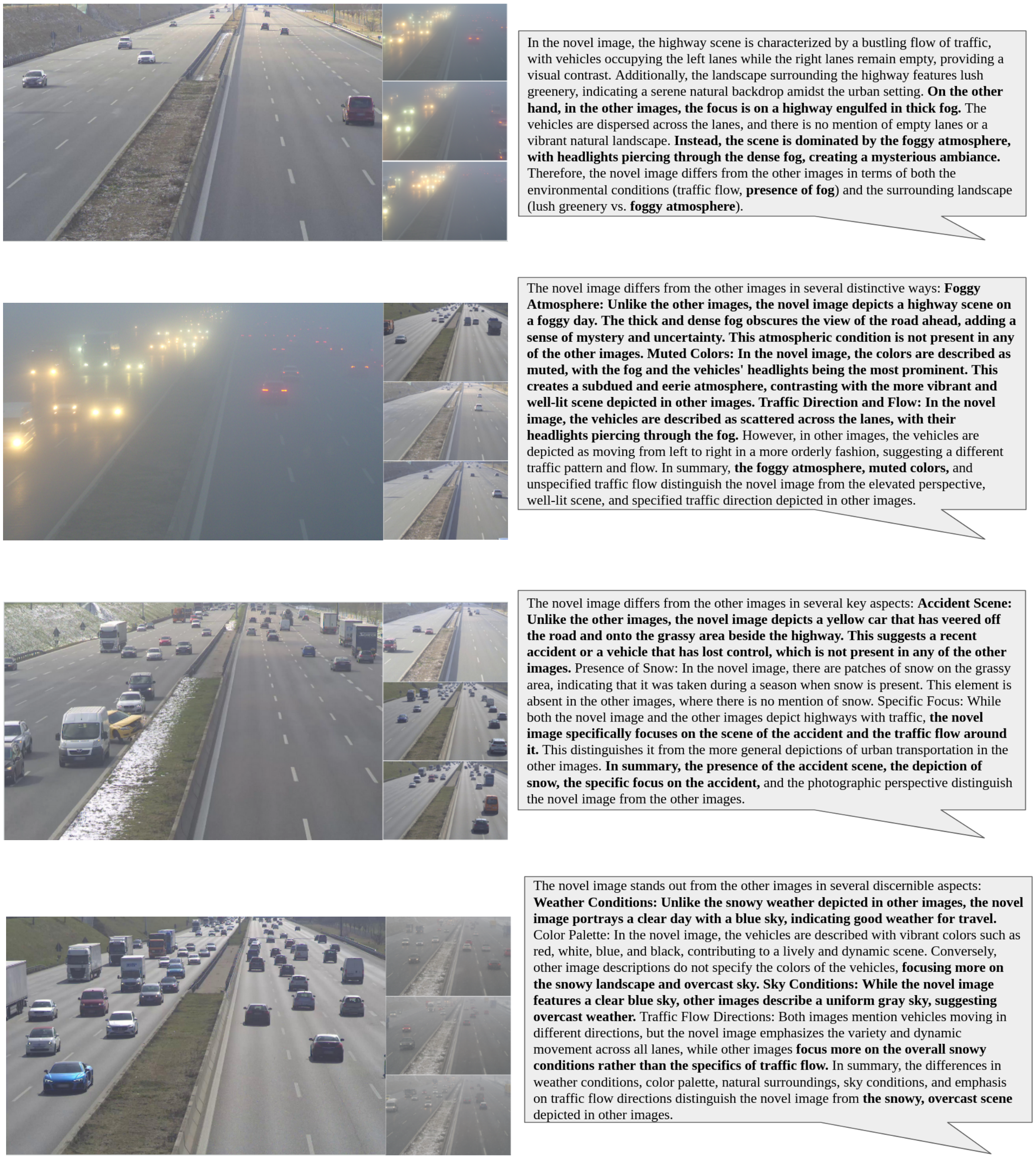
The novel image stands out from the other images as it depicts an active construction site on a city road, emphasizing ongoing construction work, orange traffic cones, and a large yellow construction vehicle. In contrast, the other images predominantly showcase city or suburban street scenes with various vehicles and environmental elements, but none highlight construction activities or the presence of a construction site.

The novel image captures a typical **day** on a city street, **featuring a clear blue sky,** trees, buildings, and traffic lights. The scene suggests a lively urban environment with a standard, unfiltered photograph style. What sets the novel image apart from the other images is its depiction of a city street **during daylight, in clear weather conditions,** showcasing the motion of a vehicle on the road without specific focus on the time of day or the transition between daylight and night, as seen in other images.

The novel image stands out from the other images because it captures a daytime street scene in a residential area with a blue car moving through an intersection. The street is marked with white lines and arrows, and **there's a stop sign on the right side**. The atmosphere is calm and orderly, typical of a quiet neighborhood. In contrast, the other images depict scenes from highways, roads with traffic cones, nighttime highways illuminated by headlights, and urban environments with trees. None of them replicate the specific combination of a blue car in a residential setting **with stop signs** and clear daylight, making the novel image unique within this group.

The novel image stands out from the other images by focusing on the rearview mirror of a red car, reflecting a silver car in front of it. Positioned on the right side of the road, the scene includes a clear blue sky, **a modern building on the left**, and a crosswalk, creating **a distinct urban atmosphere**. The perspective, reflective element, and **specific urban details contribute to its uniqueness**. In contrast, the other images capture various scenes, including bustling urban streets, highway perspectives, palm-lined streets, dusk scenes, and nighttime cityscapes. The specific combination of elements in the novel image, such as the focus on the rearview mirror, the red and silver cars, and the urban setting, distinguishes it from the rest.

Figure 3.11. LAVA images correctly identified as novel within the construction of the experimental sets. The novel image is shown enlarged, with three examples from the characteristically “normal" pool shown next to each novel image. Next to each image is the explanation of novelty generated by Algorithm 2 for the image. We add emphasis for phrases which describe the specific feature we used in constructing the set (e.g. discussion of nighttime scenery for the night image imposed on the daytime set). We note that the unique urban architecture referred to in the bottom image is a reflection of the “college campus" data pool.



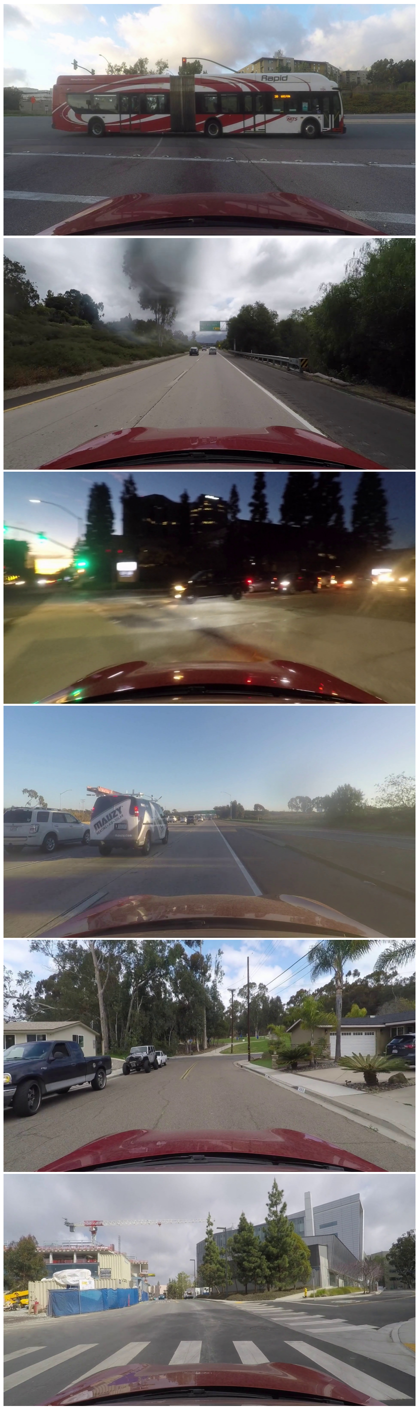
In the novel image, the highway scene is characterized by a bustling flow of traffic, with vehicles occupying the left lanes while the right lanes remain empty, providing a visual contrast. Additionally, the landscape surrounding the highway features lush greenery, indicating a serene natural backdrop amidst the urban setting. **On the other hand, in the other images, the focus is on a highway engulfed in thick fog.** The vehicles are dispersed across the lanes, and there is no mention of empty lanes or a vibrant natural landscape. **Instead, the scene is dominated by the foggy atmosphere, with headlights piercing through the dense fog, creating a mysterious ambiance.** Therefore, the novel image differs from the other images in terms of both the environmental conditions (traffic flow, **presence of fog**) and the surrounding landscape (lush greenery vs. **foggy atmosphere**).

The novel image differs from the other images in several distinctive ways: **Foggy Atmosphere: Unlike the other images, the novel image depicts a highway scene on a foggy day. The thick and dense fog obscures the view of the road ahead, adding a sense of mystery and uncertainty. This atmospheric condition is not present in any of the other images. Muted Colors: In the novel image, the colors are described as muted, with the fog and the vehicles' headlights being the most prominent. This creates a subdued and eerie atmosphere, contrasting with the more vibrant and well-lit scene depicted in other images. Traffic Direction and Flow: In the novel image, the vehicles are described as scattered across the lanes, with their headlights piercing through the fog.** However, in other images, the vehicles are depicted as moving from left to right in a more orderly fashion, suggesting a different traffic pattern and flow. In summary, **the foggy atmosphere, muted colors,** and unspecified traffic flow distinguish the novel image from the elevated perspective, well-lit scene, and specified traffic direction depicted in other images.

The novel image differs from the other images in several key aspects: **Accident Scene: Unlike the other images, the novel image depicts a yellow car that has veered off the road and onto the grassy area beside the highway. This suggests a recent accident or a vehicle that has lost control, which is not present in any of the other images.** Presence of Snow: In the novel image, there are patches of snow on the grassy area, indicating that it was taken during a season when snow is present. This element is absent in the other images, where there is no mention of snow. Specific Focus: While both the novel image and the other images depict highways with traffic, **the novel image specifically focuses on the scene of the accident and the traffic flow around it.** This distinguishes it from the more general depictions of urban transportation in the other images. **In summary, the presence of the accident scene, the depiction of snow, the specific focus on the accident,** and the photographic perspective distinguish the novel image from the other images.

The novel image stands out from the other images in several discernible aspects: **Weather Conditions: Unlike the snowy weather depicted in other images, the novel image portrays a clear day with a blue sky, indicating good weather for travel.** Color Palette: In the novel image, the vehicles are described with vibrant colors such as red, white, blue, and black, contributing to a lively and dynamic scene. Conversely, other image descriptions do not specify the colors of the vehicles, **focusing more on the snowy landscape and overcast sky. Sky Conditions: While the novel image features a clear blue sky, other images describe a uniform gray sky, suggesting overcast weather.** Traffic Flow Directions: Both images mention vehicles moving in different directions, but the novel image emphasizes the variety and dynamic movement across all lanes, while other images **focus more on the overall snowy conditions rather than the specifics of traffic flow.** In summary, the differences in weather conditions, color palette, natural surroundings, sky conditions, and emphasis on traffic flow directions distinguish the novel image from **the snowy, overcast scene** depicted in other images.

Figure 3.12. TUMTraf images correctly identified as novel within the intent of our experimental design. The novel image is shown enlarged, with three examples from the characteristically “normal" pool shown next to each novel image. Next to each image set is the explanation of novelty generated by Algorithm 2 for the image. We add emphasis for phrases which describe the specific feature we used in constructing the set.



The novel image depicts a bustling city street scene with **a large red and white bus marked with the word "Rapid."** The road is marked with white lines, and various buildings of different heights surround the scene. There's a red car on the right, and the sky is clear with fluffy white clouds. **The main difference in the novel image compared to the other images is the prominent presence of the "Rapid" bus in the city street scene, while the other images lack a similar distinctive bus or vehicle, especially one with explicit markings like "Rapid."**

The novel image differs from the collective group of other images in the following ways: Weather Conditions: **In the novel image, the sky is overcast with dark clouds, suggesting an impending storm or rainy weather. On the other hand, the collective group of other images showcases clear and sunny skies, indicating different weather conditions.** Highway Focus: The novel image emphasizes a multi-lane highway with a solid white line dividing the lanes, while the collective group of other images focuses on city streets filled with various vehicles and regulated by traffic lights. These differences highlight that the main image provides a distinct perspective of a rural or suburban highway **with overcast weather, setting it apart from the urban scenes depicted in the other images with clear and sunny skies.**

The novel image captures a vibrant nighttime scene on a city street, **emphasizing the motion of cars with bright headlights,** the silhouette of trees, and illuminated buildings. The atmosphere is characterized by the hustle and bustle of urban nightlife. What distinguishes the novel image from the other images is its focus on a nighttime city setting **with cars in motion,** contrasting with other images, which also depict a nighttime scene on a highway but with an emphasis on the car's dashboard and the sense of travel. Other images portray a scene at dusk with a parked red car, emphasizing a serene atmosphere illuminated by the setting sun, providing a stark contrast to the urban activity in the novel image.

The novel image stands out from the other images because it captures a scene on a highway with a distinct composition. The foreground features a vehicle, likely a car, with a red and white hood, positioned in the lower right corner, indicating possible motion or a shot from another vehicle. **A white van with a "MAX" logo drives in the same direction in the middle ground.** The background displays a clear sky and a line of distant trees. In contrast, the other images predominantly showcase various road scenes, often focusing on highways or city streets. The unique composition, **including the specific placement of the red and white hooded car and the presence of the "MAX" van,** distinguishes the novel image from the rest of the images in this group.

The novel image stands out from the other images as it captures a dynamic residential street scene. It features a red car in motion with a visible rearview mirror, alongside **a stationary blue pickup truck**. The scene is characterized by a partly cloudy sky, **palm trees, houses,** and a warm climate. In contrast, the other images showcase various scenes such as city streets, parking lots, highways, and suburban settings. None of them precisely replicate the elements found in the novel image, such as the combination of a moving red car, **a parked blue pickup truck, palm trees, and the residential surroundings.** These distinctive features make the novel image unique within this group.

The novel image stands out from the other images due to its unique elements. It features a red car in motion, captured from the car's rear window perspective, with a blurred background suggesting movement. **The street is marked with white lines indicating a crosswalk, and there's a modern building with a flat roof and a large glass facade on the right side. Additionally, a construction site with a visible crane adds an urban development context.** The sky is overcast, contributing to the overall urban atmosphere. In contrast, the other images showcase various scenes, including urban streets, highways, parking lots, and suburban settings. While some images capture moving cars or parked vehicles, none replicate the specific combination of a red car in motion, **crosswalk markings, a modern building, and a construction site. These distinct features make the novel image unique within this group.**

Figure 3.13. Images identified as novel, but based on some feature outside the defining feature used in set construction. These images are still be novel relative to their pool, just not along the dimension in which the set was originally constructed. Next to each image is the explanation of novelty generated by Algorithm 2 for the image. We add emphasis for phrases which describe features which are most likely novel within the larger pool, illustrating the algorithms effectiveness.

growing, we recommend applying such techniques to image data, avoiding the bottleneck of language in describing differences. As an intermediate step toward robustness, statistical passes of the description generating algorithm may be useful; by resampling a variety of images in each cluster and generating difference descriptions, the LLM could effectively take a consensus among multiple candidate descriptions.

Continuing towards safety, if novelty is identified at the scene level, there remaining open questions in mediating between the severity of the situation outside the vehicle, the readiness of the driver in the vehicle, and the ability of the vehicle to autonomously navigate the scenario. How does an autonomous system evaluate uncertainty in its ability to safely handle a novel scene? Are detection, segmentation, prediction, and planning metrics sufficient, or must we rate the novelty of a scene we encounter, and at what time horizons should a vehicle perform these assessments?

While we may never be able to gather enough data to account for all possible long-tail cases, with the methods presented in this research, we may be able to at least identify when we are encountering a long-tail event, and make safer choices in our use and training of machine autonomy at these important moments.

Acknowledgements

Chapter 3, in part, is a reprint of the material as it appears in Ross Greer and Mohan M. Trivedi. "Towards Explainable, Safe Autonomous Driving with Language Embeddings for Novelty Identification and Active Learning: Framework and Experimental Analysis with Real-World Data Sets." arXiv preprint arXiv:2402.07320. The dissertation author was the primary investigator and author of this paper.

Part II

Chasing the Long Tail: Algorithms for Active and Continual Learning

Chapter 4

The Why, When, and How to Use Active Learning in Large-Data-Driven 3D Object Detection for Safe Autonomous Driving: An Empirical Exploration

4.1 Introduction

Many autonomous driving tasks rely on supervised learning, and task performance under such methods is heavily dependent on accurate, high-volume data annotation. The conventional approach for most autonomous driving tasks, such as 3D object detection [83–87], is to ask humans to label (or supervise the labeling of) all data collected in driving, then train learning machines using the labeled data.

However, such annotation often requires meticulous treatment and expensive labor from expert human annotators [88]. When the volume of the data grows faster than the available human resources, annotating data becomes a challenging bottleneck to better-performing models. This is especially the case for autonomous driving, where the data itself can be collected quickly and diversely from fleets or even a single vehicle [89]. In fact, a German study in autonomous vehicle data estimated the annotation cost to produce direct statistical evidence of reliable AI-perception ranges in the scale of 1.16 trillion to 51,800 trillion Euro – 14,800 times Germany's gross domestic product! [8, 90] In this research, we explore and evaluate an entropy-based querying

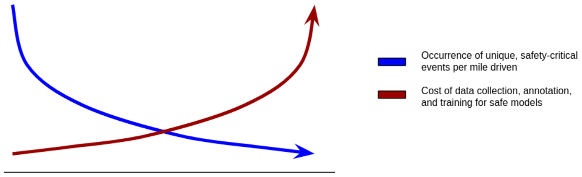


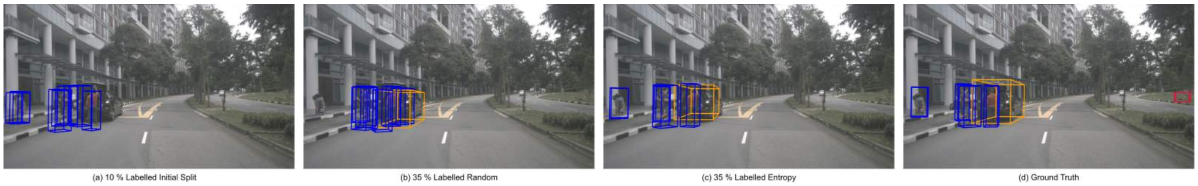
Figure 4.1. Novel safety-critical events occur with low probability while driving, making data collection of such events an enormous cost, especially since the number of instances required to teach a high-dimensional model scales exponentially with the number of data dimensions. While the left region of this curve may represent scenarios encountered in normal driving, as we progress to the right, we would expect to find not only unexpected driving environments and interactions, but also those of near-miss accidents and catastrophic failures. Collecting real-world data on dangerous accidents (and, at that, sufficient instances of this data to build models via supervised learning in high-dimensional vector spaces) is an extremely challenging task. The blue curve carries the moniker of “long-tail events".

active learning solution to this annotation bottleneck with consideration to the multimodal, multitask, and safety-critical nature of intelligent vehicle learning systems.

4.1.1 Redundancy and Data Imbalance

As a motivating example, consider a fleet which seeks to gather data in a particular region. By the nature of our roadway system, over time, vehicles will likely encounter the same roads in the same conditions and same context multiple times (e.g. a 5 o’clock rush hour traffic jam on southbound I5 near Exit 26B). For this reason, many data points collected for autonomous driving may be redundant or similar between capture sessions.

Why is this redundancy, or *data imbalance*, a problem to begin with? When nearly-identical, highly-repeated samples are used to train a model (and distinct samples are significantly less present), the data imbalance can cause the model to overfit parameters to be sensitive to the minor deviations in the over-represented data instead of solving the intended problem – an issue addressed with active learning [1] [91]. Additionally, in a well-designed model trained on a sufficiently diverse dataset, the model learns a latent space which interpolates between encoded samples, allowing the model to generalize to noisy data in the wild [92]. While collecting large amounts of data is important, there comes a point when further data collection of similar samples


(a) 10 % Labelled Initial Split

(b) 35 % Labelled Random

(c) 35 % Labelled Entropy

(d) Ground Truth

Figure 4.2. The amount of carefully annotated data available during training is closely tied to the success of the learned model. This is an image from the nuScenes dataset, whose camera and LiDAR measurements are used as input to the BEVFusion 3D Detection model discussed in this paper. When the model is trained with 10% of the available training data, we can see a high rate of false positive detections throughout the scene, and failure to note even the obvious-but-partially-occluded vehicle. As we increase the training data to 35% of the available pool, under random sampling, the false positive detections remain confounding, but the pedestrians on the sidewalk are missed altogether, and there is a general difficulty to capture the precise position, size, and orientation of these objects. On the other hand, when using the entropy querying active learning method detailed in this paper, under the same data budget, the pedestrian on the sidewalk is found and the false positive detections are significantly reduced relative to the ground truth. The ground truth, depicted on right, shows the ideal detection, which requires the careful selection of additional data points to further boost trained model performance without incurring expensive demand for extensive data annotation. In this research, we present methods for intelligently querying the available data pool for new training samples using active learning.

becomes redundant as the learning of the latent space sufficiently covers the real-world pattern for similar samples. This is especially the case when it comes to safety for autonomous driving, as it is not the familiar which poses a risk, but rather encounters with unexpected or novel situations, so-called “long-tail" (infrequent) driving events. At a practical level, because ML systems optimize over a loss function summed over each training sample, in cases of severe class imbalance, catering to the “majority" serves to place the learner in a comfortable local minimum of loss. Further, when it comes to safe autonomy, these non-majority cases are often the most significant from a safety standpoint. This challenge is shared with biomedical research, earning the name *curse of rarity*, referring to the difficulty of gathering samples of events that are most likely to cause system safety failures [10]. This is also referred to as the “long-tail problem".

Data sampling methods are commonly used to overcome data imbalance, such as random under-sampling (to remove majority cases from training data), and random over-sampling (having under-represented classes appear more frequently during training). In principle, standard data augmentation serves this same purpose, but on the basis that the collected data under-represents the variance of the complete population of data. Naturally, augmentation methods can be applied to minority-class data to build a stronger representation within a training dataset. However, here we seek solutions which add more to a model's knowledge than crafted re-use of existing training data, such that a system can continually learn from new examples, finding “useful novelty" through examining the entropy of considered data [93].

4.1.2 Dealing with High-Dimensional Data

In addition to data imbalance, data for intelligent vehicle tasks tends to be high- dimensional. For example, a typical testbed may be collecting data along dimensions of time, arrays of pixels from 2D spatial cameras, sweeps of 3D spatial lidar measurements, and a variety of additional sensors such as GPS, INS, and CAN.

By learning an expansive low-to-high-level feature set, this scale and variety of information has proven to be helpful towards a variety of tasks such as lane detection [51], vehicle and VRU detection and tracking [94], traffic sign and light classification [53, 55, 95], trajectory prediction [64, 96], vehicle landmark identification [97], driving maneuver and driver style classification [98]; such tasks are important not only towards autonomous driving, but also towards effectiveness of ADAS systems [99]. While this data provides a wealth of information to learn from, the infamous “curse of dimensionality" puts systems at risk of improperly fitting models to complex data (requiring exponential amount of increased data with each new dimension introduced). Further, even annotating this data at a high-quality, frame-by-frame, pixel-by-pixel, voxel-by-voxel level is a monumental task, near impossible to complete exhaustively given resource constraints and costs in human annotation, discussed further in later sections.

In essence, much of machine learning involves reducing the dimensionality of data from its high-dimensional observed form to a task-useful form. Sometimes we do this before the data enters the learning mechanism (e.g. pre-processing the data by selecting features to learn from), sometimes we do this inside the learning mechanism (e.g. an early bottleneck layer in a neural

Table 4.1. Percentage of nuScenes 3D object dataset possible to be annotated by 40 hours of work, calculated from rate estimates in recent research.

Method	% of nuScenes objects annotated in 40 Hours (est.)
3D-BAT [100]	6.86%
Lee et al. [101]	2.78%
Without assistance average [102]	0.09%
With assistance average [102]	0.34%

Table 4.2. Percentage of nuScenes 3D object dataset annotated by 40 hours of work per week, with the number of weeks shown in the left column, calculated using the average rate of Table 4.1. We use this dataset size as the allowed size of our training sets to evaluate the effectiveness of active learning approaches.

Weeks of Annotation	% of nuScenes dataset in Training Pool
1	2.52%
2	5.04%
3	7.56%
4	10.08%
5	12.60%
6	15.12%
7	17.64%
8	20.16%
9	22.68%
10	25.20%

network, which learns lower-dimensional encodings of feature combinations). Sometimes we do this explicitly (e.g. extract particular features, such as one color channel for a task like brake light extraction [97]), often termed *selecting*, other times letting the system learn the features (e.g. neural network which outputs a low-dimensional vector for system inference [103]), often termed *mapping*.

In addition to implications toward the theoretical limits of a systems ability to learn, high-dimensional data also contributes to a lack of explainability in systems, and complicates the process of safety regulation on a practical level. Techniques in intelligent data selection and feature extraction help to resolve these challenges, but as information is discarded, a tradeoff is induced between system performance and system explainability. Pes et al. [91] categorize three types of feature selection methods:

- Filter methods, which remove data according to some non-learned criteria,
- Wrapper methods, which essentially search over different feature subsets to optimize performance, and
- Embedded methods, which, critically, make use of learning algorithm internal information in the process of searching for optimal features. For example, while a wrapper method might make use of system accuracy over a test set to select a best feature set, an embedded method may examine the uncertainty values of logits during classification to drive its selection criteria.

As expected, filter methods bear the least computational cost, but show the most constrained performance (albeit, sometimes this constrained performance may be sufficient towards a task). In this research, we explore an embedded method, accepting increased computational complexity to enhance model performance.

Table 4.3. Comparison of previous research in Active Learning for 3D Object Detection in Autonomous Driving Datasets

Active Learning Methods	Datasets	Modalities	Insights
Entropy, Monte Carlo dropout, ensemble learning	KITTI	Camera, LiDAR	Can save up to 60% of labeling efforts for same performance [104]
Class Entropy and Spatial Uncertainty	Private	LiDAR	Importance of both classification and spatial uncertainty [105]
Kernel coding rate	KITTI, Waymo	LiDAR	44% box-level annotation costs savings without compromising performance [106]
Sensor consistency-based selection score, LiDAR guidance as semi-supervision for monocular detection	KITTI, Waymo	Camera	17% savings in labeling costs, top performance in BEV monocular object detection official benchmarks with 2.02 AP gain [30, 107]
3D consistency of bounding box predictions in both semi-supervised and active learning	KITTI	LiDAR	Improves from baseline by more than 60% with only 1500 annotated frames [108]
Consensus score variation ratio, sequential region-of-interest matching	KITTI	Camera, LiDAR	Saves 35% of labeling efforts [109]
Bi-domain active learning, diversity-based sampling	KITTI	LiDAR	Gains on cross-domain settings; retraining Waymo-trained model on just 5% of KITTI data outperforms 100% KITTI-trained model [110]
Uncertainty sampling	Astyx	Radar, Camera, LiDAR	Semi-automatic labeling for efficient dataset creation [111]
Augmentation, dropout, insertion, deletion	KITTI	LiDAR	Practical method for fast annotation [112]
Semi-supervised co-training on prediction disagreement	KITTI, Waymo	Camera	Semi-supervised co-training clearly boosts detection accuracy in regimes where the training size is just 5-10% of the pool [113]
Ego-pose distance-based sampling	Navya3DSeg	LiDAR	Heuristic-free method; outperform random sampling [114]
Bayesian surprise (KL divergence)	AGV Anomaly Dataset	LiDAR	Effective in warehouse environment anomaly detection; may be applied as AL to identify novel data [41]
Uncertainty sampling	Private	LiDAR, Camera	Effective for identifying road damage [115]
Spatial and temporal diversity-based sampling	NuScenes	LiDAR	Annotation costs vary between scenes; diversity methods are effective and allow warm start [39]
Classification Entropy Querying	**NuScenes**	**LiDAR, Camera**	**Outperforms random sampling, reduces intra-class performance difference, learning of minority classes (this research)**

4.1.3 Using Active Learning

Active learning is the process by which a learning system interactively selects which data points should be added from the unlabeled data pool to the labeled training set, assisted by the intervention of a human expert providing associated annotations. Within this framework, in the case of classification tasks, we consider the *information gain* of a new datum to be a measure of the decrease in entropy when that datum is added to the training set.

This problem is therefore twofold: (1) for model cost and performance, a large set of these non-informative data points increases the time and decreases effectiveness of the training process and model tuning, and (2) for annotation cost, in situations where a data corpus has high levels of redundancies, annotating all collected data may waste a lot of human resources on non-informative samples.

4.2 Related Research

Cohn et al. engage in a particular style of active learning as concept learning via queries, by which the learner requests from an oracle a label for a particular sample [1]. In particular, their work examines the effectiveness of such methods in improving generalization behavior. One of the goals in active learning is to label a small subset of collected unlabeled data so as to maintain or achieve better performance given the cost of labeling or requesting human oracle. Conventional query strategies usually evaluate informativeness based on handcrafted functions or heuristic selection methods, such as query-by-committee [116], uncertainty sampling [117, 118], region of uncertainty and version space [1], and expected model change [119]. Empirical studies [120, 121] have shown that the best strategy or informativeness measure may be application specific. Moreover, the effectiveness of such heuristic methods is limited and varies across different datasets.

Due to the variability in datasets, models, and query selection methods, it is difficult to form a noticeable consensus for the state of the art in active learning. Accordingly, through this paper, we show the clear utility of one such method towards the detection safety goals of autonomous driving systems. Early works in the literature applying active learning in autonomous driving tasks mostly utilized handcrafted features such as Haar wavelets and the histogram of oriented gradients on SVMs or Adaboost [122–124]. As deep learning became a dominant approach in computer vision [125], more works have resorted to DNNs as models in active learning to further boost performance. In [126], four active learning methods (sum of entropy, maximum entropy, average entropy, and Monte Carlo dropout) are applied to the Apollo Synthetic dataset and Waymo Open dataset on 2D object detection and instance segmentation tasks, using R-CNNs appropriate for each task, and finding that active learners beat baselines in these autonomous driving tasks, and that summation-entropy learners tend to bring forward samples with the most instances, which seem to have the strongest effect on learning. While these insights are valuable, in this research, we focus on the task of 3D object detection, reflecting the need for vehicles to recognize an object's relative position for purposes of safe planning; accordingly, our discussion of related works will continue with active learning towards this task. We highlight relevant literature towards effective detection and efficient annotation of such datasets in Table 4.3, and discuss particular methods in the following paragraphs.

In [104], Feng et al. use active learning to find the fewest number of labeled training samples to improve the performance of 3D object detection by convolutional neural networks (CNNs) trained on LiDAR point clouds, using Monte Carlo Dropout and Deep Ensembles to measure entropy in predictive labels and mutual information between model weights and class predictions. Moses et al. [105] coin a "*LOCAL*" acquisition function, utilizing both classification and localization-based uncertainty and summing values across all objects in a sample as inclusion criteria. They adapt the exclusive Basic Sequential Algorithmic Scheme (BSAS) clustering scheme for per-object matching to allow for entropy calculation, and use variance of spatial estimation as measure of spatial uncertainty. However, their training and evaluation is carried out on a limited 41 LiDAR point clouds of data from a private, government-owned airborne-collected dataset, and they point out the important difference in scale compared to autonomous driving datasets such as KITTI [127], Waymo, and nuScenes. Luo et al. [106] show that maximizing the kernel coding rate as criteria for data selection can strongly outperform most generic (task-agnostic) active learning methods, and marginally improves over task-specific active learning methods for 3D detection, at lower running time than near performers. Hekimoglu et al. [128] use active learning on a monocular-input for the 3D detection task, quantifying uncertainty using (1) the variance of predicted Gaussian localizations, and (2) the variance in predicted position when an image undergoes a variety of intensity and sharpness transforms to form a query-by-committee, and perform experiments using a fixed training size, showing that the combinations of data augmentation query-by-committee and heatmap uncertainty lead to clear improvement over random sampling. Hekimoglu et al. are later the first to use a teacher-student paradigm for active learning data selection and semi-supervised training, this time combining LiDAR measurement with monocular images to form this teacher-student relation, and setting a new state of the art for “monocular" (since the LiDAR is technically used without label) 3D object detection on KITTI [107]. Hwang et al. [108] exploit the ability to localize 3D objects under flips, rotations, and scalings so that unlabeled data can be used to train the model to be consistent in assessing object locations, using this value as both an additional training term and uncertainty measurement towards active learning. These papers are all united on the theme that active learning leads to higher 3D detection performance at lower data budgets, shown in a general sense on a limited number of object classes.

From our search, Liang and et al. [39] provide the only prior investigation of active learning on the nuScenes dataset. While we study uncertainty-based active learning in this research, Liang et al. study diversity-based active learning, finding that spatial and temporal diversity of samples are effective strategies. They importantly highlight the differences of annotation costs being variable between scenes, due to the varying number of objects that may appear in each; accordingly, they define the annotation budget by a combined scene-object formulation. They also hypothesize that these entropy-based methods may introduce redundant samples in a scene, since having a high-entropy class at any one pooling round would likely identify all members of that class to be high entropy, when a smaller representative amount would suffice for learning. Further, their diversity-based active learning approach allows for a “warm start" to their base training pool, as the diversity criteria can be established without a trained model. Under their annotation budget, the entropy-based method appears to underperform compared to random sampling (and, this makes sense given that a scene’s entropy is formed by the sum of the detected object entropies). However, we do recommend that entire scenes be annotated at once (even if highly crowded), due to the difficult task of the model to identify all objects within any annotated scene; state-of-the-art models are not trained to look for single objects in a field of many, but rather to identify all instances simultaneously, and the task of identifying instances within a crowd warrants appropriate data. Accordingly, we show that at the scene-sampling level budget, entropy-driven active learning actually does exceed a random baseline.

We point out key differences between our research and the research of [104], [106], [128], [107], [108], and [39]:

- We experiment over the nuScenes [129], while other works experiment on KITTI [104,106–108, 128] and Waymo [106, 107]; by experimenting on an additional strongly-established dataset, we further enhance their case for the benefits of entropy-driven querying and active learning in autonomous driving.

- Accordingly, while the KITTI and Waymo-based approaches [104, 106–108, 128] divide objects into five or less classes (for example, small vehicle, human, truck, tram, and miscellaneous), we divide objects into 10 classes[1], better capturing the distribution of minority classes and the effects of active learning on less-represented data.

- Some of the above prior works do not include orientation [39, 104] or classification [128] of objects in their detection. These attributes are important for the purposes of understanding

[1] Pedestrian, Bicycle, Car, Bus, Construction Vehicle, Motorcycle, Barrier, Traffic Cone

possible direction-of-travel and behavioral patterns for an object [130]. We include and evaluate these predictions in our network output.

- [104] uses ground-truth and pre-trained image 2D detectors in their 3D detection pipeline, while [39, 106, 108] utilize LiDAR only and [128] utilizes monocular camera only. By contrast, we *train* our image-based 2D detector as part of a two-stage (image + LiDAR) network; thus, active learning decisions influence the complete network performance.

We create an active learning framework for autonomous driving to jointly minimize redundant, expensive annotation while avoiding the risk introduced by domain adaptations and overfitting. Such an approach allows autonomous vehicles to efficiently learn new knowledge for unseen environments under constrained resources.

4.2.1 How long does it take to annotate 3D bounding boxes?

3D object detection is a very relevant and important task to autonomous driving because unlike 2D object detection, the object's position and orientation in space is inferred. However, the task of drawing 3D bounding boxes to train models for such tasks can be more time consuming than 2D annotation. In this section, we highlight just how expensive this data can be to make a case for active learning as a cost-reducing measure so these systems can be developed safely at scale.

To assist in this annotation task, tools such as Zimmer et al.'s 3D-BAT [100] have been developed for semi-automatic labelling. In the 3D-BAT test case, they find that the most efficient expert human annotator is able to use the system to annotate approximately 57 objects per minute, and the average among users is approximately 40 objects per minute. However, IoU with ground truth is very low for these fast annotations, with the best annotator reaching only around 20%. Lee et al. design a system where annotators provide object anchor clicks to generate instance segmentation results in 3D, reporting 3.7 seconds per bounding box [101]. To motivate their auto-labelling system MAP-Gen, Liu et al. report statistics that an experienced annotator takes

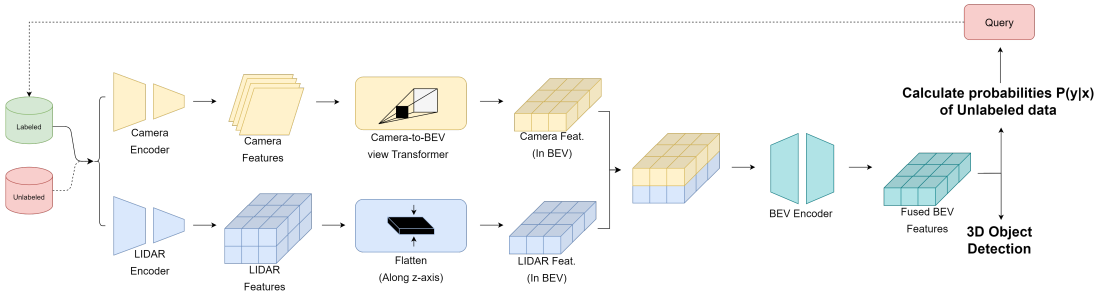


Figure 4.3. An illustration of Active Learning setup with the BEVFusion model.

Table 4.4. Class Frequencies in NuScenes, ordered most to least present.

Class	Frequency (%)
Car	42.30
Pedestrian	19.05
Barrier	13.04
Traffic Cone	8.40
Truck	7.59
Trailer	2.13
Bus	1.4
Construction Vehicle	1.26
Motorcycle	1.08
Bicycle	1.02

around 114 seconds per 3D bounding box, and those using a 3D object detector assistant around 30 seconds [102]. While auto-labelling may eventually be a viable solution toward massive data annotation, here we emphasize the importance of expert annotators in the loop for the purpose of human-validated safety in such a risky domain.

NuScenes contains 1.4M camera images and 390k LIDAR sweeps of driving data, originally labeled by expert annotators from an annotation partner. 1.4M objects are labelled with a 3D bounding box, semantic category (among 23 classes), and additional attributes. In Table 4.1, we form estimates of the portion of nuScenes dataset that annotators utilizing above-described methods could annotate in 40 hours, again noting that the quality of annotation for some of these methods is substandard.

Though this paper demonstrates the utility of active learning towards the task of 3D object detection, we would like to stress that this paper is not about improved 3D object detection, but rather about systematically selecting data in a way that improves model learning under limited resources. There are many additional tasks in autonomous driving beyond 3D object detection; for example, Motional has accompanying semantic visual and LiDAR segmentation tasks, which are even more time-intensive during annotation (for example, Schmidt estimates up to 90 minutes to fully segment an autonomous vehicle domain image [131]). The benefits demonstrated on our sample task are applicable towards other tasks; active learning is used to increase efficient utility of data towards improving any task model, especially in the cases of multi-task active learning frameworks [30, 132].

4.3 Data Methods

Because the rate of newly collected data is faster than the rate of annotation, prioritizing data for learning new knowledge is expected to boost performance in a more optimal rate per datum. Therefore, we formulate the autonomous driving tasks as pool-based active learning problems [133]. We assume that large collections of unlabeled data are collected continuously in the pool and associate queries for the accurate annotation by expert human annotators with some costs. To minimize the total cost while maximizing the autonomous driving performance, our proposed algorithms only request humans to annotate data points when they are novel to the existing dataset and influential to the current model. The other data points are assigned with the label generated by the current model or have their annotation delayed. For evaluation, the model is trained with a few steps in each cycle based on the union of the requested labels and a subset of assigned labels of data points.

4.3.1 Active Learning for NuScenes

NuScenes comprises 1000 scenes. In order to maintain complete control over the scenes within the dataset, we will be making slight adjustments to the fundamental database setup. These modifications are necessary to accommodate the presence of unlabeled data and the computations associated with active learning queries. The specific adjustments will depend on the selected method. This alteration is a crucial step in the process of sampling underrepresented data from the current labeled pool.

Towards reproducability of our methods, throughout the training and testing of the chosen model we will use the *trainval* split of the dataset, which containes 850 scenes. We will split this into labeled, unlabeled and validation subsets, where the validation set will contain 150 scenes used to evaluate and test the model. We will discard the provided *test* subset for our experiments, as the labels are not provided by the creators.

The remainder of the scenes in *trainval* will initially be part of the unlabeled subset and iteratively be sampled approximately 5% at the time into the labeled set. This process will proceed until models have been trained on the labeled subset containing up to 50% of the original scenes present in the *trainval* dataset.

4.3.2 Baseline: Random Sampling

We create a baseline budget using the average of the statistics surveyed in Table 4.1, or 2.52% of the nuScenes dataset annoted with a 40-person-hour labelling budget. We create 10 iterative batches of such labels, representing in a figurative sense the amount that one (very dedicated) annotator might label over 10 weeks, shown in Table 4.2.

For each baseline trial, we randomly sample a percentage of scenes described in Table 4.2 and train the model to N epochs. We will start with 10.08% scenes and add 5.04% for every round representing a start with 4 weeks worth of work and an increase of 2 weeks worth of work for every sampling round.

4.3.3 Active Learning Method: Entropy Querying

We aim to investigate the implications of utilizing a commonly employed uncertainty measure for sampling from an unlabeled data pool [104], [128], [107], [108].

While certain methods, like "least confidence" and "smallest margin," derive their acquisition function based on individual or paired confidence values across all semantic classes, our

Table 4.5. Performance across standard 3D object detection metrics at different training dataset sizes, training by Random Sampling and Entropy Querying.

Round	Pool	mAP ↑		mATE ↓		mASE ↓		mAOE ↓		mAVE ↓		mAAE ↓		NDS ↑	
		Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy
1	10%	0.3095	**0.3106**	0.4665	**0.4588**	**0.3494**	0.3669	1.108	**1.030**	**1.236**	1.420	0.3794	**0.3187**	0.3353	**0.3409**
2	15%	0.3419	**0.3639**	0.4392	**0.4144**	0.3397	**0.3386**	0.9418	**0.8909**	**1.223**	1.347	0.3095	**0.3074**	**0.3679**	0.3868
3	20%	0.380	**0.4041**	0.4041	**0.3994**	0.3503	**0.3270**	0.8296	**0.8131**	1.317	**1.060**	0.3017	**0.2955**	0.4014	**0.4185**
4	25%	**0.4236**	0.4217	0.3921	**0.3786**	**0.3136**	0.3319	0.7685	**0.6780**	**0.8695**	0.9803	**0.277**	0.2942	**0.4497**	0.4446
5	30%	0.4494	**0.4557**	0.3713	**0.3552**	**0.3112**	0.3169	0.6989	**0.6563**	0.7764	**0.7106**	0.2485	**0.2287**	0.4841	**0.5011**
6	35%	0.4474	**0.4676**	**0.3498**	0.3679	0.3168	**0.3066**	0.6569	**0.6152**	0.8830	**0.6354**	0.2941	**0.2324**	0.4736	**0.5181**
SOA	100.00%	0.750		-		-		-		-		-		0.761	

specific focus lies on the "entropy querying" method. This method takes into account a model's uncertainty across all conceivable classes. Our objective is to uncover potential enhancements that the entropy query method could bring about, given that the informativeness measure is determined by comparing a sample's probability of belonging to a class across all possible classes. [134]

This process starts by conducting inference on the unlabeled subset and strategically selecting samples found to be the most informative. The criterion for informativeness is determined by the entropy scores associated with each sample. These scores are calculated, generally, using the formula expressed in Equation 4.1.

$$\Phi_x = \sum_y P(y|x)\log_2 P(y|x) \tag{4.1}$$

In the equation, Φ_x represents the entropy score for a given sample x. The calculation involves the summation over all possible class labels y, where $P(y|x)$ represents the probability of class y given the input x. The resulting entropy score serves as a quantitative measure of uncertainty, guiding the selection of samples for active learning.

By adopting the entropy sampling approach, we aim to enhance our understanding of its impact on the selection process within the context of 3D datasets. The utilization of entropy scores provides a nuanced perspective on uncertainty, enabling the selection of samples that contribute most significantly to the model's learning process.

4.3.4 BEVFusion Model for 3D Object Detection

For the purpose of designing and experimenting on data selection and learning schemes, in this paper we consider the fundamental driving task of 3D object detection. This is an essential task for obstacle avoidance and path planning.

More specifically, we consider the recent BEVFusion approach to 3D object detection [135]. At the time of writing, this method holds third place in the NuScenes tracking challenge and seventh place in the detection challenge, with newer variants of the BEVFusion architecture populating additional high rankings. While there are many techniques to find a unified representation of image and LiDAR data, LiDAR-to-Camera projection methods introduce geometric distortions, and Camera-to-LiDAR projections struggle with semantic-orientation tasks. BEVFusion is meant to create a unified representation which maintains both geometric structure and semantic density.

The Swin-Transformer [136] is used as the image backbone, while VoxelNet [137] is used as the LIDAR backbone. To create the bird's-eye-view (BEV) features for images, first a Feature Pyramid Network (FPN) [138] is applied to fuse the multi-scale camera features. This produces a feature map 1/8 of the original size. After this, images are downsampled to 256x704 and the LiDAR point clouds are voxelized to 0.075m to get the BEV features needed for object detection. These two modalities are fused using a convolution-based BEV encoder to prevent local misalignment between LiDAR-BEV features and camera-BEV features under depth estimation uncertainty from the camera mode. The full architecture with active learning can be seen in 4.3.

4.3.5 Explanation of nuScenes Metrics

We summarize here some common metrics in 3D object detection for conceptual description, and direct the reader to the nuScenes documentation for implementation thresholds and class-specific details:

- Mean Average Precision (mAP): for the nuScenes dataset, AP is computed by taking the 2D center distance on the ground plane, filtering predictions beyond a certain threshold, and integrating the recall-precision curves for values over 0.1. These values are averaged over match thresholds of 0.5, 1, 2, 4 meters, and then averaged across classes.

- Average Translation Error (ATE): Euclidean center distance in 2D in meters.

- Average Scale Error (ASE): $1 - IOU$ after aligning centers and orientation.

- Average Orientation Error (AOE): Smallest yaw angle difference between prediction and ground-truth in radians.

- Average Velocity Error (AVE): Absolute velocity error in m/s.

- Average Attribute Error (AAE): Calculated as $1 - acc$, where acc is the attribute classification accuracy.

These metrics are all positive (or zero) valued, and translation and velocity errors can grow unbounded. For metrics presented in this paper, we take a mean over all classes when presenting general statistics in Table 4.5, and also examine per-class performance to observe the effects of active learning on minority classes in further analysis.

4.4 Experimental Evaluation

Experiments are conducted to test if entropy sampling performs better than random sampling. The initial dataset contains approximately 10% of the original dataset, we add approximately 5% of data for each subsequent round of training.

A single round involves training one model with six epochs on the current labeled training set. Following this training phase, the checkpoint file from the last round is employed to perform inference on the unlabeled dataset pool. Thereafter, the employed active learning method will be used on the obtained results. This process identifies the samples to be included in the labeled

Table 4.6. Merged table with samples from random and entropy sampling.

Data [%]	10		15		20		25		30		35	
	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy	Random	Entropy
Car	31,940	32,488	42,308	42,942	56,415	53,760	71,209	64,451	88,131	74,933	108,562	82,911
Pedestrian	20,356	24,448	30,636	31,994	40,901	39,679	46,442	48,129	54,062	58,708	61,281	62,752
Barrier	7,915	15,224	24,166	20,335	28,904	22,117	34,338	28,117	38,903	34,791	44,906	38,639
Truck	7,972	6,128	11,467	10,184	14,354	15,555	18,267	19,871	21,503	22,796	25,908	25,926
Traffic Cone	3,767	6,165	10,283	8,921	12,539	10,225	15,628	13,028	18,584	15,179	20,584	18,007
Trailer	2,562	1,635	2,779	2,977	3,801	5,750	5,580	7,658	6,448	8,237	7,486	9,591
Bus	1,698	1,574	2,172	2,447	2,729	3,112	3,774	3,808	4,496	4,556	5,475	5,084
Construction Vehicle	1,262	1,401	2,138	2,253	2,877	2,903	3,678	3,634	4,595	4,366	5,145	4,752
Bicycle	762	954	1,468	1,427	2,090	1,750	2,378	2,042	2,659	2,508	2,967	2,917
Motorcycle	1,539	802	1,016	1,364	1,400	1,749	1,852	2,255	2,489	2,721	2,875	3,095

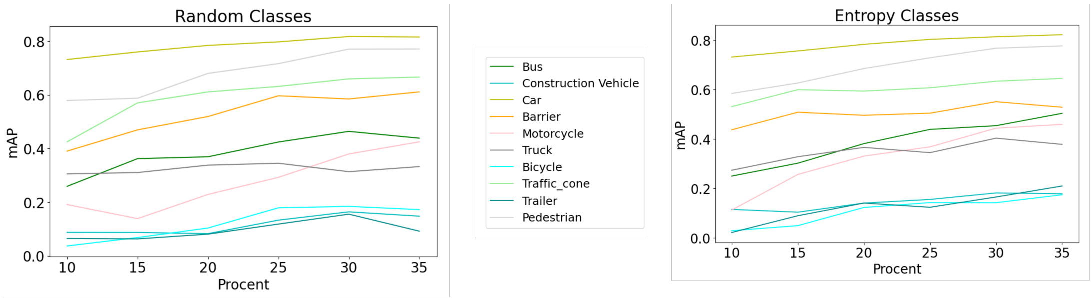


Figure 4.4. Overview of per-class results, with Random Sampling on left and Entropy Querying on right. While the ordering of classes remains intact and nearly identical to the frequency of appearance of respective classes in the dataset, under entropy sampling, the margin between best and worst performing classes decreases.

training dataset for the subsequent round. Each experiment will involve six rounds, as seen in Table 4.5. We note that in general, the more training data sampled, the stronger the model learns to generalize to real-world test data.

The Active Learning strategy dominates on 26 of the 35 checkpoints and metrics in Table 4.5. A sampling of qualitative examples are provided in Figure 4.7.

Table 4.4 describes the class frequencies of appearance in the nuScenes dataset. We collapse the pedestrian class to contain adults, children, construction workers, those using personal mobility devices, wheelchairs, or strollers, and wearing construction or police uniforms. From Figure 4.4, we observe that the ordering of classes by highest-to-lowest mAP approximately matches the ordering of class appearance in Table 4.4 (car, pedestrian, traffic cone, barrier, truck, bus, motorcycle, construction vehicle, trailer, bicycle). While this ordering is preserved by active learning, we notice that the gap between the lowest mAP and greatest mAP is smaller under

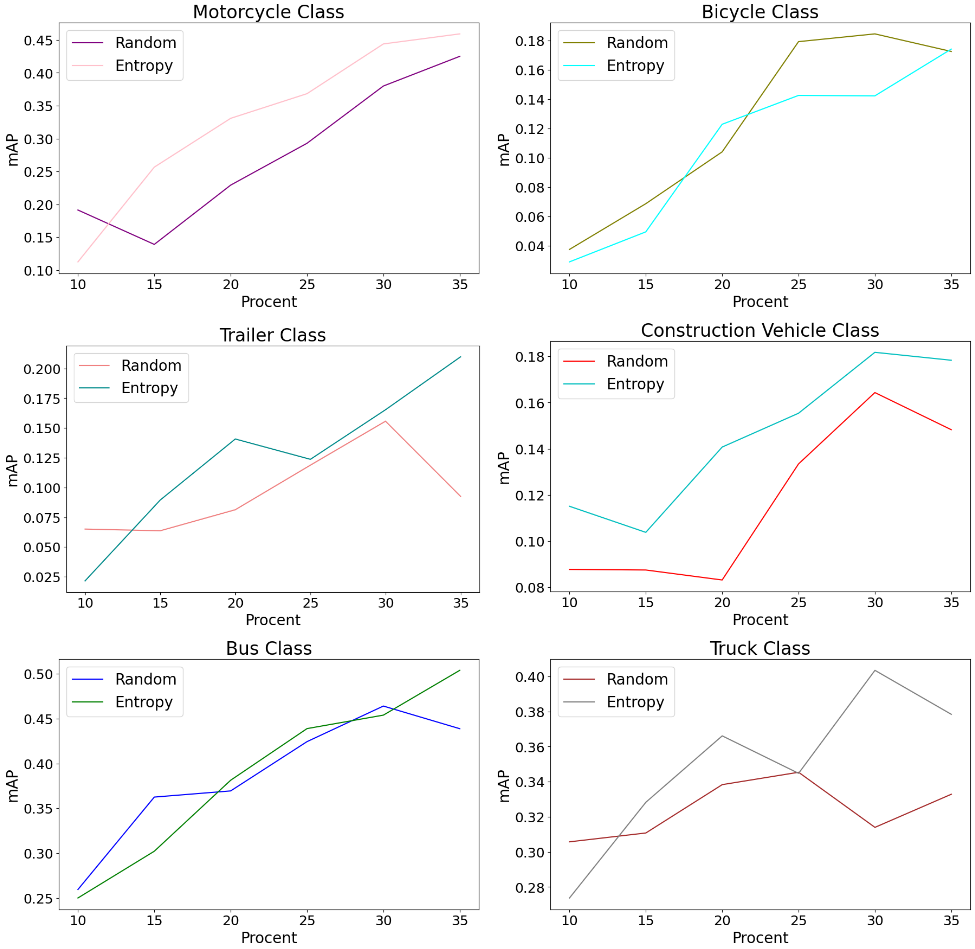


Figure 4.5. Class-separated analysis of mAP performance between random sampling and entropy query active learning. Entropy query active learning shows a tendency to outperform random sampling on mAP, shown on the six minority classes in these graphs.

active learning, and progressively tightens as more data is added to the pool. Class-specific comparisons are illustrated in Figure 4.5. In general, entropy-driven active learning shows improved precision over random selection on all classes, especially beyond early data pool sizes. The margin of performance varies by class.

We make a few observations over these class performances. Most of the worst performing classes (trailer, construction vehicle, bicycle, motorcycle) perform better under entropy sampling than in random sampling. The trailer class performed the worst in random sampling and a little better in entropy sampling, and when looking at Table 4.6, it can be observed that entropy sampling focuses on querying trailer data for every round. The Construction Vehicle class is another class which did not do well in either entropy or random sampling, however, we again see in the table that entropy sampling still outperforms random sampling by a small margin in all rounds, even though the random sampling method draws more examples of this class beyond 30%, suggesting that the active learning algorithm was not finding better “informative" samples beyond this point (corroborated by random sampling’s greater sampling amount still not besting the performance of entropy querying). As a more classic case, in the motorcycle class, for the initial round the mAP result for this class is comparable to the lowest accuracies observed in other classes. But, under entropy querying, there is a rapid growth in the amount of samples present for this class and as a result the mAP performance consistently increases as the training pool grows.

To what extent does entropy querying resolve uncertainty by corrective sampling of minority classes? As shown in Figure 4.6, for each class in each graph, the entropy-driven method tends to pull the distribution to the right toward underrepresented classes as the training pool size increases. We observe the margin between methods for the majority class (car) being widened as the active learning method samples larger pool sizes, with this difference being distributed among the minority classes. The non-normalized data values are presented in Table 4.6.

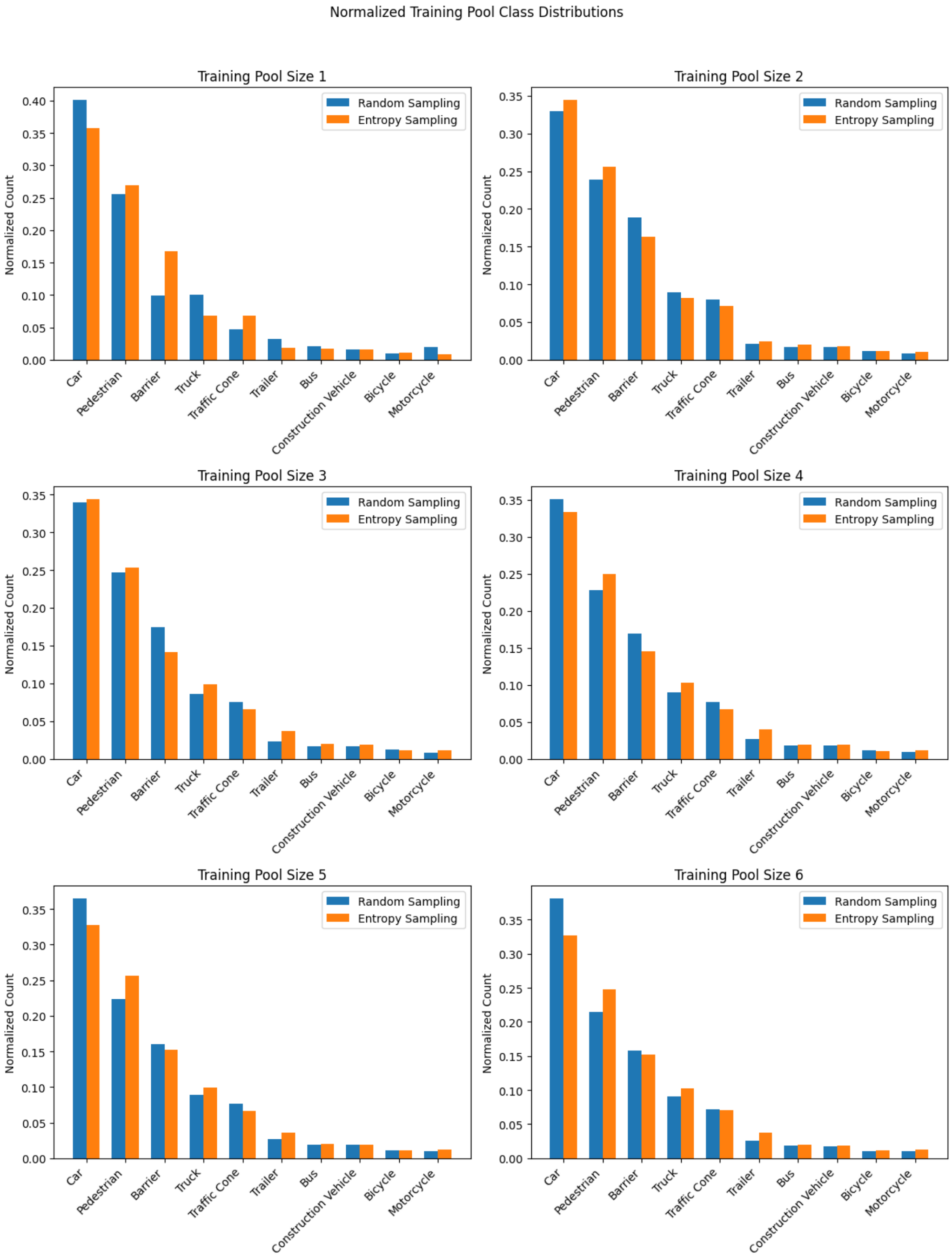
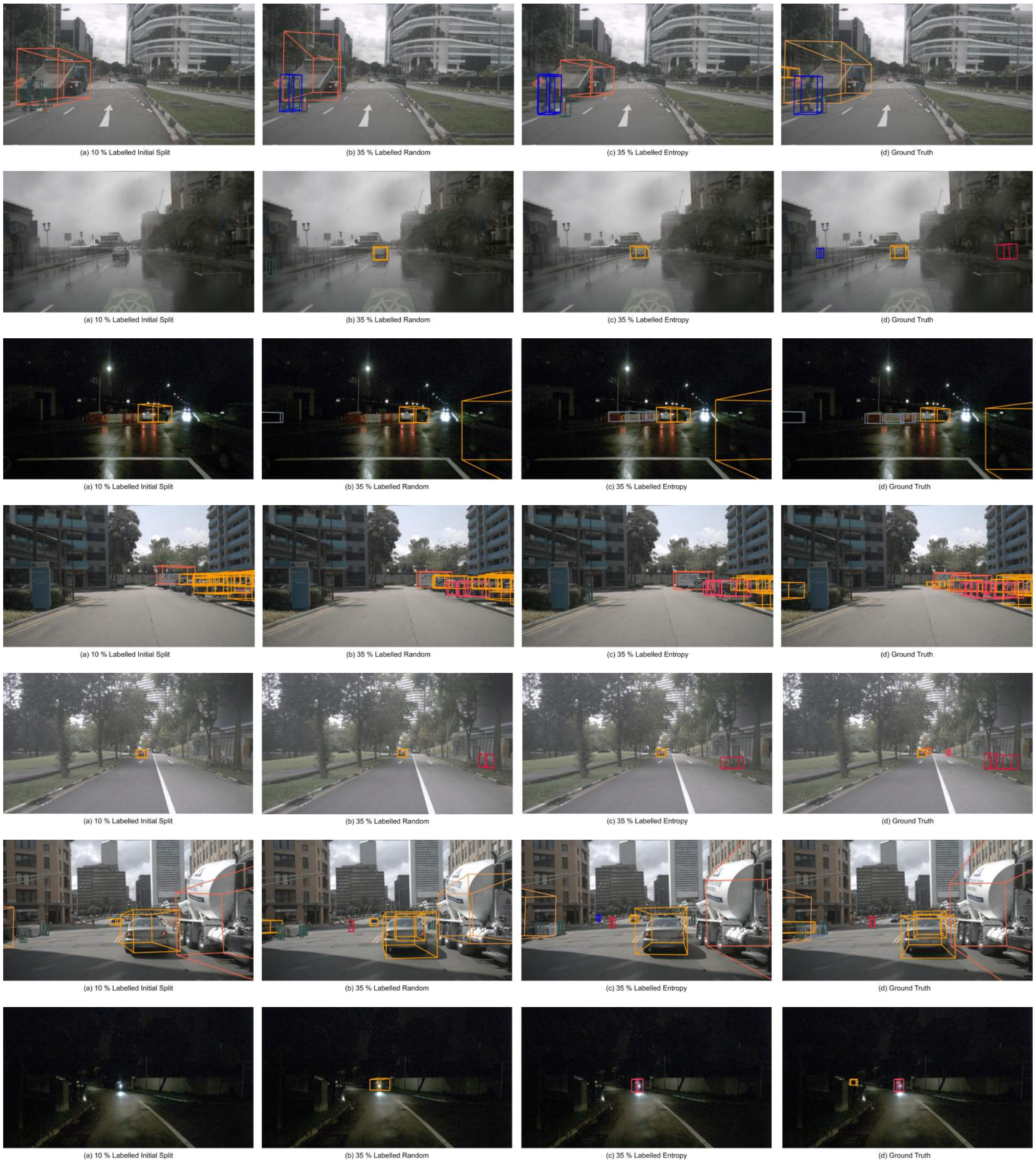


Figure 4.6. Distribution of samples among classes for each method, varying with training pool size. Entropy sampling methods tend to reduce the addition of majority class (car) samples to the training pool, opting instead to distribute this budget towards the remaining classes. Note that the columns for the respective methods are normalized, as sample sizes will not necessarily sum to the same value since sampling is performed at the scene level, and different scenes may have different numbers of objects.



(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

(a) 10 % Labelled Initial Split
(b) 35 % Labelled Random
(c) 35 % Labelled Entropy
(d) Ground Truth

Figure 4.7. Qualitative examples from nuScenes, comparing the initial training on a randomly selected 10% of nuScenes, followed by random sampling to 35% of nuScenes versus 35% selected using entropy queries, and finally the ground truth annotations. Different color boxes refer to different object classes. A few notable observations under entropy-driven AL: in the first and second row, we see a better handling of orientation-error; in the third row (night), the barrier class is more readily detected; in the fifth row, the presence shape of the bicycles are better inferred; in the sixth row, the truck class and size is correctly inferred; finally, in the last row, the nearby bicycle is detected and correctly classified, where it is missed altogether or mistaken to be a car via random sampling.

4.5 Concluding Remarks

Based on the observed results, it is evident that the integration of the entropy querying method with the Birds-Eye-View Fusion model constitutes a favorable combination, demonstrating the effectiveness of active learning.

One limitation of this analysis is the robustness of results, containing a single method with six iterations of training, each comprising six epochs. To address this limitation, it is recommended that future testing and analysis involve a more extensive approach, where several runs would be conducted for each method. Taking an average of these runs would yield more reliable and comprehensive results. Additionally, the decision to increase the number of epochs from six to ten in future experiments is motivated by the anticipation that a more distinct pattern which more closely matches the fine-tuned state of the art performance of such models may emerge with extended training. Specifically, this adjustment also aligns with the amount of epochs used in the BEVFusion paper, facilitating better direct comparisons with their outcomes.

4.5.1 Future Research in Active Learning

Future research unexplored in active learning in this field includes the learning of query policies directly from autonomous driving tasks and data, instead of relying on handcrafted policies. This could be done using deep reinforcement learning approaches to learn the query policy in the active learning framework. Because the query selection has been shown as a decision process, reinforcement learning can be applied to learn the query [139]. Query strategies learned by reinforcement learning have been shown to outperform the heuristic selection methods such as uncertainty sampling and random sampling in a natural language processing task [139]. However, to the best of our knowledge, such data-driven query strategies have not been explored in autonomous driving. This is especially important considering the necessity of such systems to efficiently adapt to new environments [29, 140].

In conclusion, the findings in this research give an affirmation that entropy querying effectively samples the most informative instances from classes with lower accuracies and limited available data, showcasing its utility in the active learning framework, encouraging the adoption of active learning approaches to simultaneously reduce annotation costs and increase data efficiency in learned models for autonomous driving tasks.

Acknowledgements

Chapter 4, in part, is a reprint of the material as it appears in Greer, Ross, Bjørk Antoniussen, Mathias V. Andersen, Andreas Møgelmose, and Mohan M. Trivedi. "The Why, When, and How to Use Active Learning in Large-Data-Driven 3D Object Detection for Safe Autonomous Driving: An Empirical Exploration." arXiv preprint arXiv:2401.16634 (2024). The dissertation author was the primary investigator and author of this paper.

Chapter 5

Language-Driven Active Learning for Diverse Open-Set 3D Object Detection

5.1 Introduction

Object detection is critical for safe autonomous driving. Data-driven approaches currently provide the best performance in detecting and localizing objects in the 3D driving scene. Detection models perform best on objects which are most represented in driving datasets. This creates challenges when some objects are less represented (minority classes), or unrepresented within the annotation scheme (“novel" objects [141], relevant for “open-set" learning [142]), and becomes especially important when minority objects are most salient to driving decisions [53–55, 143]. Further, from a pragmatic standpoint, the collection, curation, and annotation of such datasets can be extremely expensive [75, 144], motivating the use of heuristics and algorithms which limit annotation efforts while maximizing model learning, illustrated in Figure 5.1.

5.2 Related Research

Active learning methods are driven by a query function which selects relevant data from an unlabeled pool to be annotated and joined to the training set. These methods broadly divide into two classes: uncertainty-based and diversity-based methods [145]. In uncertainty-based methods, data is selected by the query function’s assessment of how confusing the datum is *to*

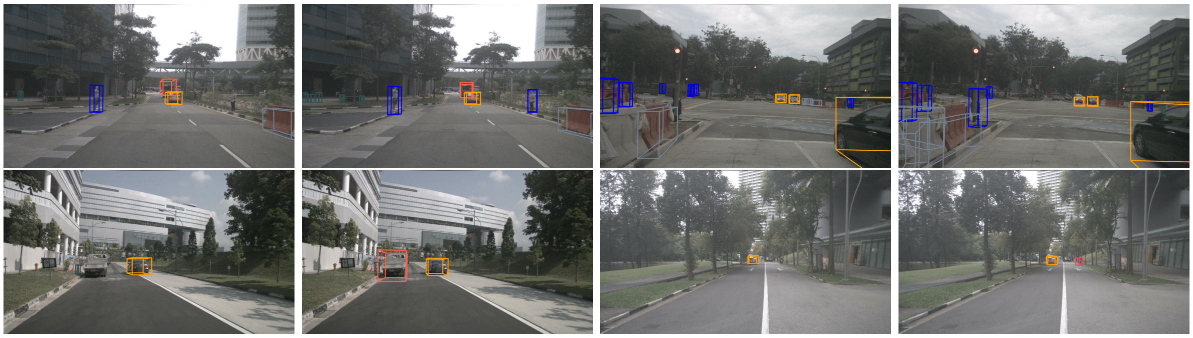


Figure 5.1. Choosing the most informative data can impact object detection model performance. Images in the left column are the results of a model trained on 50% of nuScenes data, selected at random. Images in the right column are the results on the same images of a model trained on 50% of nuScenes data, but selected using our VisLED active learning query strategy. In the top two rows, we see cases where challenging pedestrians are missed on the left image (preparing to cross on the right side of the road, and standing behind the crossing pole, respectively), but correctly detected on the right. Similarly, in the bottom two rows, the under-represented classes of motorcycle and truck are more readily detected using our active learning strategy.

the existing model. On the other hand, in diversity-based methods, data is selected by being distinct from existing training data by some measure, and this can be done without consideration of the learning model.

5.2.1 The Role of Uncertainty and Diversity-Based Methods in Closed and Open Set Learning

In closed-set learning, it is assumed that a system should classify or learn about a fixed set of target classes. By contrast, in open-set learning, the system assumes that it may encounter novel data which belongs to a class unrepresented by its current target set. Naturally, this brings up many research challenges in recognizing this novelty when it appears, determining when to define a new set construct, and integrating new constructs into the learning mechanism.

Here, we suggest that diversity-based methods are particularly well-suited for these open-set learning tasks. Because uncertainty-based methods select relative to their existing world model, there is an inductive bias imposed in relating new data to existing patterns. On the other hand, in diversity-based methods, data is compared only to other data, analogous to unsupervised learning. This does create a tradeoff: closed-set learning excels under uncertainty-

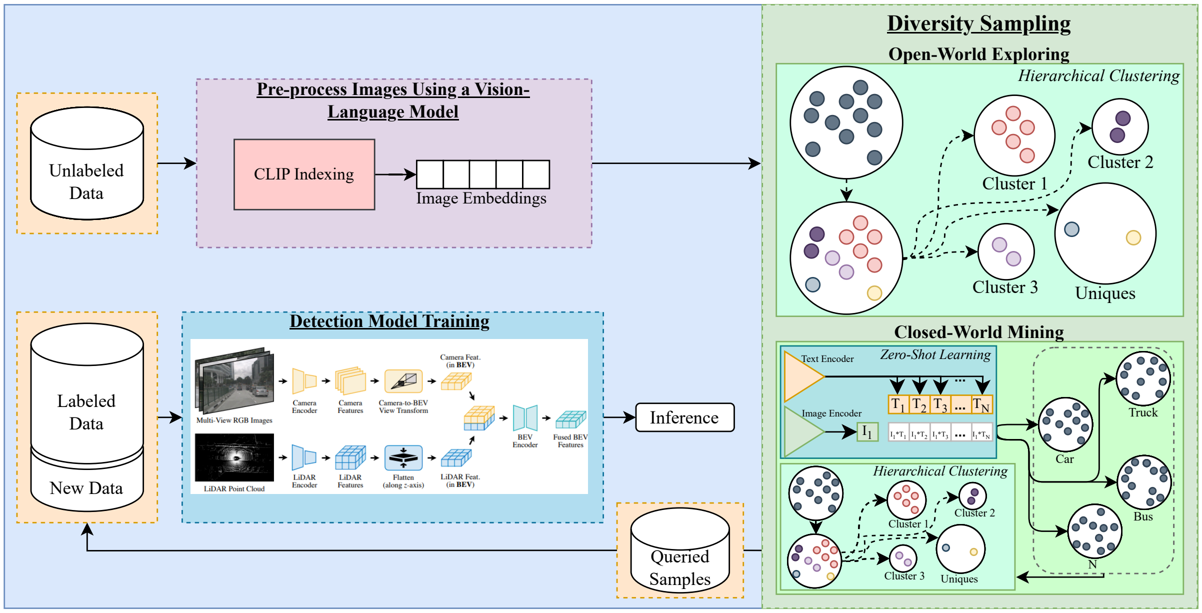


Figure 5.2. VisLED System Overview. For both Open-World Exploring and Closed-World Mining, the system begins with the processing of the unlabeled data pool into vision-language embedding representations. In Open-World Exploring, these embeddings are clustered and used as the basis for a query. In Closed-World Mining, the embeddings are first used in zero-shot learning to classify scenes based on object appearance, and then further clustered per-class, offering a chance to sample from particular classes which are known to be minority in the labeled training set.

driven sampling, since these methods are optimized for the current world model and target set, but cannot treat the world as “open" as diversity-driven sampling. But, critically, we show in this research that diversity-based active learning still provides a strong benefit to the learning system (even if not “optimal" to the particular model and set definition), *and* is suitable for open-set data selection.

5.2.2 Learning from Vision-Language Representations

Prior research has shown that vision-language representations such as embeddings from contrastive language-image pretraining (CLIP) [69] can be used to identify novelty of an image relative to a set (and, as a bonus, can be decoded into a verbal explanation of novelty) [146]. In our research, we utilize this representation and corresponding ability to select novel images as a proxy for the amount of useful, previously-unexplored information within a complete multimodal driving scene, allowing for an active learning query to select diverse samples based on vision-language encodings of scene images.

5.3 Algorithm

Here, we present our algorithm named Vision-Language Embedding Diversity Querying (VisLED-Querying), which can be viewed in Figure 5.2. The algorithm can be used in two different settings:

1. Open-World Exploring: this method imposes no particular class expectations on the data. It is suitable for cases when the model seeks to include information which is most novel relative to data it has seen previously.

2. Closed-World Mining: this method utilizes a zero-shot learning [69] step to sort data between a fixed set of classes before evaluating for novelty, filtering any points estimated to not belong to one of the closed-set classes. This method is suitable for mining new and different instances of existing classes, but may also filter out the most difficult or unusual instances even from known classes if the zero-shot method fails to recognize the object.

Algorithm 3: Open-World Exploring VisLED-Querying

```
Input: Unlabeled pool of egocentric driving scene images
Output: Updated training set
Embed each egocentric driving scene image from the unlabeled pool using CLIP;
Use hierarchical clustering to separate the embeddings;
Sample new data points from the unclustered set for addition to the training set;
```

In the closed-world mining setting, when employing CLIP's [147] zero-shot learning technique for classification, the algorithm examines each sample image to identify objects which are predicted to belong to one or more of the model's predefined classes. Each sample is assigned to a single class, in this case taken as the argmax class over all classes considered using the zero-shot learning method. We note that, in our experiments, this method predominantly identifies

Algorithm 4: Closed-World Mining VisLED-Querying

```
Input: Unlabeled pool of egocentric driving scene images
Output: Updated training set
Embed each egocentric driving scene image from the unlabeled pool using CLIP;
Encode each class label using a text encoding;
Applying zero-shot learning by maximizing the product of the embeddings, sort the
  embedded images by class;
For each class, apply hierarchical clustering;
Sample new data points from the unclustered set associated with the desired class,
  and add to the training set;
```

one class with high accuracy. In instances where other classes may also be identified, their confidence scores are typically low enough to risk false positives, rendering them inadequate for threshold-based classification; therefore, we use a single-class assignment for simplicity and accuracy. We do note that, as an algorithm variant, it is reasonable to distribute scene images to multiple classes if respective confidence values for the additional classes are sufficiently high.

Once the samples for each class have been identified, embeddings will be generated separately for each class, followed by hierarchical clustering. Subsequently, a number of samples will be selected from each class, with a focus on sampling from clusters with minimal data representation. Initially, the algorithm will prioritize unique samples (clusters with only one sample present), matching them with corresponding scene names until the desired number of unique scenes is achieved in the training set. Upon inclusion of all scene-names from unique samples, the algorithm will proceed to clusters containing pairs of images, and so on, until the required number of scenes have been sampled for the training set.

In the open-world exploring setting, this same procedure is followed beginning with sampling embeddings from unique singleton clusters, without any pre-classification step to prioritize drawing from particular classes.

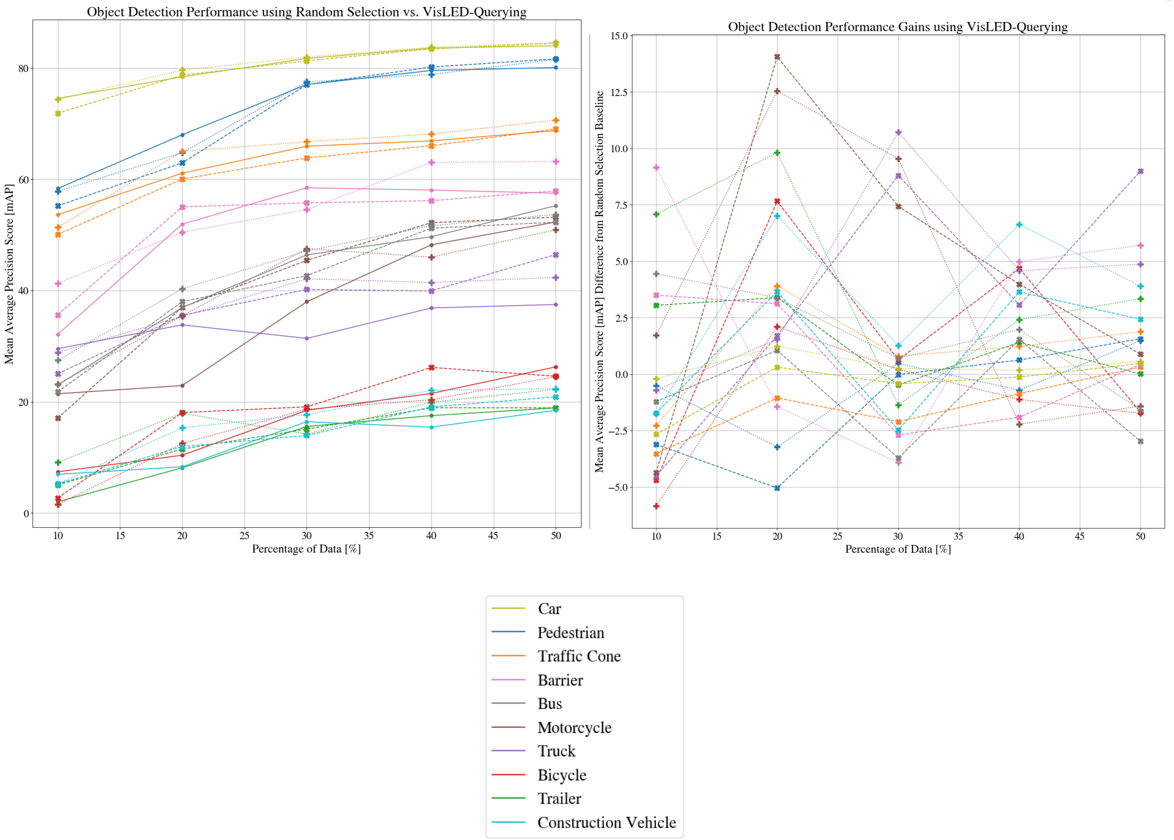


Figure 5.3. BEVFusion models are trained using three different data selections: Random (dot markers and solid line), VisLED-Closed-World (x-markers and dashed line), and VisLED-Open-World (+-markers and dotted line). The top graph illustrates detection performance, while the bottom graph illustrates performance difference relative to the random-selection baseline. Performance is averaged over 5 complete data selection + training runs of each model at each training pool size.

5.4 Experimental Evaluation

5.4.1 Dataset

We use the nuScenes object detection dataset [148] for our experiments. nuScenes contains 1.4M camera images and 400k LIDAR sweeps of driving data, originally labeled by expert annotators from an annotation partner. 1.4M objects are labeled with a 3D bounding box, semantic category (among 23 classes), and additional attributes. nuScenes comprises 1000 scenes. In order to maintain complete control over the scenes within the dataset, we modify the fundamental database setup slightly, using the method introduced in [32, 37] to accommodate active learning queries. We use the *trainval* split of the dataset for public reproducibility.

5.4.2 3D Object Detection Model

We explore the BEVFusion approach to 3D object detection [85], which has demonstrated notable performance, ranking third in the nuScenes tracking challenge and seventh in the detection challenge, and the top performing method which has been made publicly reproducible. While various methods exist to integrate image and LiDAR data into a unified representation, LiDAR-to-Camera projection methods often introduce geometric distortions, and Camera-to-LiDAR projections face challenges in semantic-orientation tasks. BEVFusion addresses these issues by creating a unified representation that preserves both geometric structure and semantic density.

In our implementation, we utilize the Swin-Transformer [136] as the image backbone and VoxelNet [137] as the LiDAR backbone. To generate bird's-eye-view (BEV) features for images, we employ a Feature Pyramid Network (FPN) [138] to fuse multi-scale camera features, resulting in a feature map one-eighth of the original size. Subsequently, images are down-sampled to 256x704 pixels, and LiDAR point clouds are voxelized to 0.075 meters to obtain the BEV features necessary for object detection. These modalities are integrated using a convolution-based BEV encoder to mitigate local misalignment between LiDAR-BEV and camera-BEV features, particularly in scenarios of depth estimation uncertainty from the camera mode. We provide a comprehensive overview of the architecture, including its integration with VisLED-Querying, in Figure 5.2.

5.4.3 Experiments and Results

We train the BEVFusion model in increasing training set sizes of 10% increments, using four different acquisition modes: (1) Random Sampling, (2) Entropy-Querying, (3) VisLED-Querying with Closed-Set Mining setting, and (4) VisLED-Querying with Open-World Exploring setting. We repeat each VisLED method four times at each data pool size, taking the average performance from four trials.

As hypothesized, active learning strategies outperform the random baseline, and the entropy-querying method is dominant due to its nature of optimizing uncertainty with respect to the model, as opposed to VisLED's model-agnostic sampling. Yet, as illustrated in Table 5.1, VisLED still stays consistently ahead of random sampling, and offers a 1% gain over random sampling mAP at 50% of the data pool, *all without requiring* ***any*** *model training or inference*. Interestingly, the open-world exploration setting tends to marginally outperform the closed-world mining setting at nearly all data pool sizes for both metrics, suggesting that the novelty represented in the language embeddings is sufficient for identification of informative samples, even without inducing any bias from categorizing samples beforehand. On the other hand, it is also possible that the uncertainty in classifying the objects being mined for in fact makes these objects *less* likely to be found in the closed-world mining setting, again encouraging the use of the open-world exploring setting in any case.

Per-class performance is illustrated in Figure 5.3. As expected, class performance correlates with class representation in the nuScenes dataset. Observing the differences in VisLED-selected detection performance over the random selection baseline in the bottom graph of Figure 5.3 reveals some interesting patterns; at 10% data, 4 of the 10 classes perform above random. At 20% data, this increases to 7 of 10 classes above random, and by significantly higher margins of benefit than the opposing margins of detriment when underperforming. The same “benefits-outweigh-costs" pattern repeats at the other data levels. The particular spike in performance around 20% data may also have an interesting explanation, which relates to performance on the two least-represented classes, illustrated in Figure 5.4. These classes, motorcycle and bicycle, represent 1.08 and 1.02% of the nuScenes objects, respectively. When VisLED-CW is used to sample uniformly from each class, it would actually *run out* of motorcycle and bicycle samples around 20% training data, because at each 10% data increment, 1% of nuScenes data should be coming from each of the 10 classes. This means that after two training rounds, the data from the particular class should be exhausted, which explains why we see the greatest margin in performance over random happening at this level - and, this is a strong gain, around 10% mAP for bicycle and 5% mAP for motorcycle. This further explains the asymptotic behavior we see as the data volume approaches 50%; there is less prototypical data for these classes available for the detector to learn at this point. For similar reason, we see a consistent boost in the performance on the truck class (illustrated in Figure 5.6); this class has 7.59% representation in nuScenes, making it an excellent candidate for uniform gain throughout all training rounds, and reaching almost its entire representative dataset by the 50% data round. This balance of data proportionality and sampling may explain the consistent gains, even as high as 20% mAP improvement over baseline at intermediate rounds.

Besides the issue of dataset representation, we can also examine performance on classes which may be generally difficult to learn. Looking at the two lowest-performing classes on baseline (trailer and construction vehicle, representing 2.13 and 1.26% of nuScenes respectively), Figure 5.5 shows that these classes indeed benefit from VisLED sampling - in fact, at 20, 40, and 50% training data, both closed-world and open-world methods dominate the random sampling selection method.

Table 5.1. This table shows the mean average precision (mAP) and nuScenes detection score (NDS) metrics for the random sampling, and VisLED-querying (Closed-World Mining and Open-World Exploring) in every round. It also shows the mAP and NDS scores for the full training split when trained using one GPU.

Labeled Pool		mAP					NDS				
Rounds	%	Random	VisLED (CWM)		VisLED (OWE)		Random	VisLED (CWM)		VisLED (OWE)	
			Mean	STD	Mean	STD		Mean	STD	Mean	STD
1	10%	30.95	28.94	0.37	**32.14**	0.76	33.53	32.59	0.33	**34.85**	0.71
2	20%	38.00	40.61	0.94	**41.70**	0.95	40.14	41.34	0.56	**42.44**	0.96
3	30%	44.94	45.28	0.93	**46.94**	0.25	48.41	48.82	0.86	**50.84**	1.16
4	40%	47.73	49.26	0.53	**49.59**	0.66	53.10	**53.64**	0.32	52.99	0.59
5	50%	49.90	50.98	0.13	**51.74**	1.08	55.64	56.40	0.40	**56.61**	1.09
	100%	52.88					58.73				

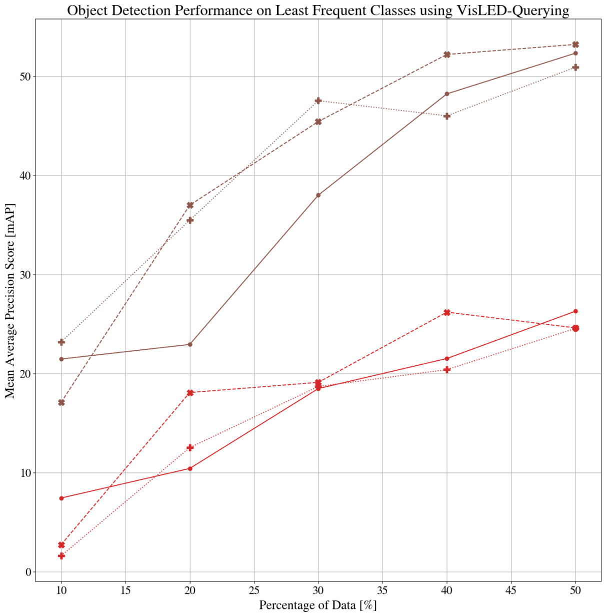


Figure 5.4. The bicycle and motorcycle classes are least represented in the nuScenes dataset, which causes these classes to appear infrequently during training when selecting data with random sampling. By using VisLED to sample, more bicycle and motorcycle instances are drawn, leading to a performance gain at early data increments. This gain levels off as the training pool aggregates all bicycle and motorcycle samples.

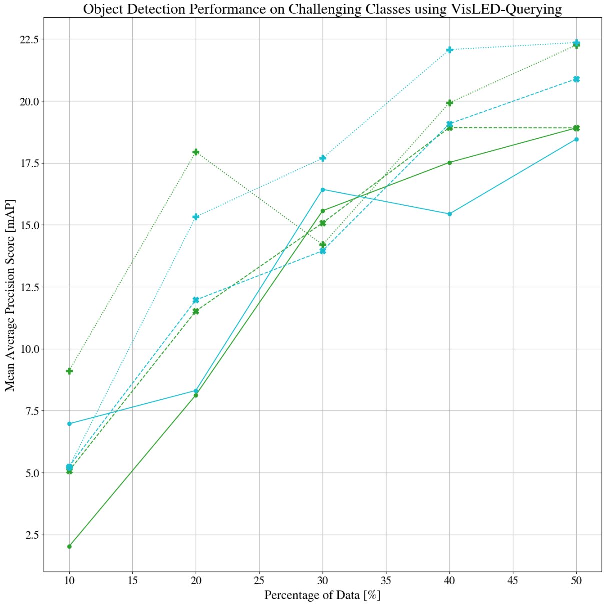


Figure 5.5. From class performance, the trailer and construction vehicle classes are most challenging to learn. When VisLED querying is used, informative samples from these classes are pulled into the training pool, giving stronger detection performance than random sampling at nearly all data volumes.

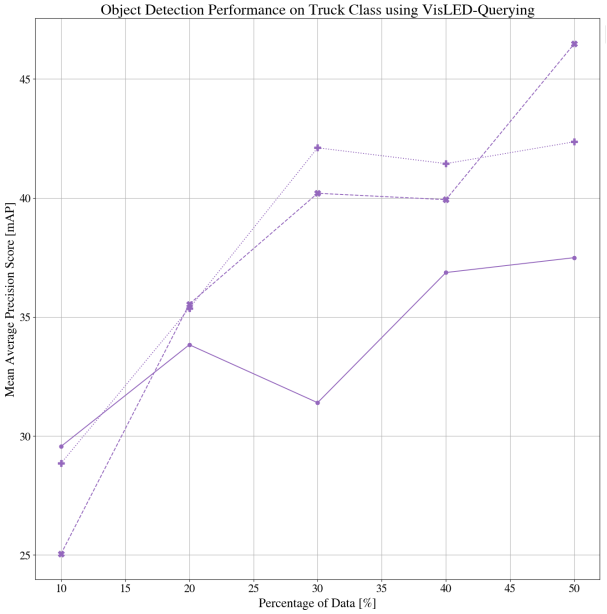


Figure 5.6. Detection performance on the truck class provides a clear illustration of the benefits of VisLED querying. Of special interest is the fact that the truck class would be nearly completely sampled around 70% nuScenes training pool size, using the uniform sampling scheme of closed-world-VisLED; in other words, once all instances of a particular class are sampled, the benefit will begin to level off.

5.5 Discussion and Conclusion

Our presented learning method, VisLED-Querying, samples without any information about the model. This enables VisLED to select novel, informative data points, to the extent that novelty is visibly identifiable, for *any* model. The benefit this offers is that a data point may need to be annotated only once, and can then be used in a variety of models for additional autonomous driving tasks instead of sampling and possibly forming an entirely different set for annotation. While these gains may be marginal in the current data setting ($<$ 1000 scenes), at scale, these performance gains may translate to serious reductions in annotation costs and safety-critical detection failures. Further, VisLED offers one key possibility that is otherwise limited on uncertainty-driven approaches: VisLED will recommend unique samples without any prior assumptions on class taxonomy, making it especially suited to open-set learning, where new classes may be introduced at any time. This capability, when paired with methods of self- or semi-supervised learning for object detection by fusing LiDAR and camera [107], may prove especially beneficial in identifying and learning from novel encounters. In future research, we plan to experiment on the effectiveness of VisLED in multi-task learning settings [30], experiments on other benchmark datasets [77], and experiments in open-set and continual learning. Further, experiments will also examine the benefits of VisLED querying over safety-critical underrepresented classes in driving scenes, such as pedestrians using a stroller (0.09% of nuScenes objects), mobility aid (0.03%), or wheelchair (0.04%), or emergency vehicles such as an ambulance (0.00004%). In these cases, the ability to use zero-shot learning methods or even the general, open-world VisLED querying approach, may lead to training data which is more balanced and effective at capturing data which sits on the long tail of driving scenarios, making for safer perception and planning.

Acknowledgements

Chapter 5, in part, is a reprint of the material as it may appear in Robotics and Automation Letters, 2024: Greer, Ross, Bjørk Antoniussen, Andreas Møgelmose, and Mohan M. Trivedi. "Language-Driven Active Learning for Diverse Open-Set 3D Object Detection." The dissertation author was the primary investigator and author of this paper.

Part III

Learning from Trajectories: Novelty in Motion

It is important to collect and annotate copious volumes of data, as described in the previous chapters, for robust perception of driving scenes. But, this perception is not necessarily the end task of greatest importance. Safe driving comes down to a reduced task of determining acceleration and steering inputs, but to be able to plan in a complex environment requires perception of the driving scene.

Further, the driving environment is dynamic, containing agents which move capriciously, and because of this, it is important for planners to have an idea of where these other agents may be moving. This is accounted for in the task of trajectory prediction, which I present in this chapter.

When the driving scene itself becomes input to *learn* the paths to predict or plan, having high volumes of correctly-annotated driving scenes again becomes bottleneck to performance. In this chapter, I again show the utility of active learning, this time in the context of trajectory prediction, and again for the purpose of increasing data efficiency and saving annotation cost.

Chapter 6

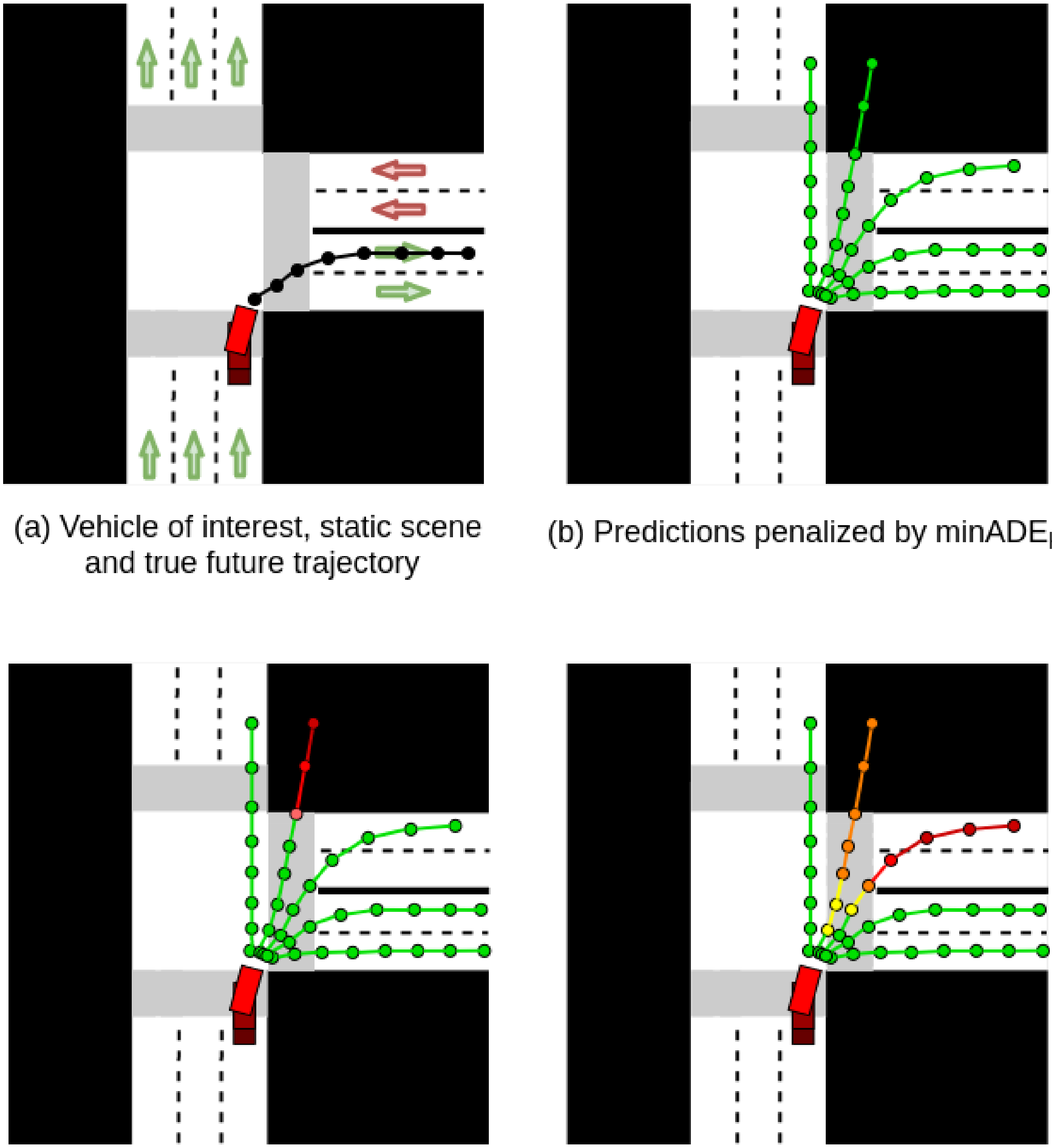
Trajectory Prediction in Autonomous Driving with a Lane Heading Auxiliary Loss

6.1 Introduction

To safely navigate complex city traffic, autonomous vehicles need the ability to predict the future trajectories of surrounding vehicles. There is inherent uncertainty in predicting future trajectories, making it a challenging task. In particular, the distribution of future trajectories is multimodal. At a given instant in a traffic scene, a driver could have one of several plausible goals, with multiple paths to each goal.

Recent work has addressed multimodality in trajectory prediction by learning models that output multiple trajectories conditioned on the past motion of agents and the static scene around them. Common approaches include learning mixture models [60, 61, 149–154], sampling latent variable models [65, 155–167], or sampling stochastic policies trained using inverse reinforcement learning [168–171]. However, defining appropriate evaluation metrics for models that output multiple trajectories still remains an open challenge.

The most commonly used evaluation metric for multimodal trajectory prediction is the minimum average displacement error over K trajectories (minADE$_K$). This has the advantage of not penalizing diverse, but plausible trajectories output by models. A limitation of minADE$_K$ is



(a) Vehicle of interest, static scene and true future trajectory

(b) Predictions penalized by minADE$_K$

(c) Predictions penalized by off-road rate and off-road distance

(d) Predictions penalized by proposed off-yaw metric and loss

Figure 6.1. Motivating example: A vehicle of interest approaching an intersection (top-left). The commonly used minADE$_K$ metric fails to penalize a diverse set of poor trajectories (top-right). The off-road rate and off-road distance metrics partially address this (bottom-left), but fail to penalize trajectories that violate lane direction. Our proposed off-yaw metric and corresponding YawLoss seek to address this (bottom-right). Severity of imposed penalty is illustrated by color, with green minimal and red maximal.

that it fails to penalize models that output a diverse set of trajectories of poor quality (Fig 6.1b). This has been addressed in prior work by additionally reporting sample quality metrics. Of particular interest are the off-road rate and off-road distance metrics [172, 173] which penalize predictions that fall outside the drivable area in a scene, visualized in Fig 6.1c. However, there's more structure to vehicle motion: vehicles typically follow the direction ascribed to lanes. A naive formulation of the off-road rate or off-road distance metrics fails to penalize trajectories wrongly predicted in the direction of oncoming traffic.

In this work, we define a new metric for sample quality of predicted trajectories termed the *off-yaw rate*. The off-yaw rate measures the adherence of predicted trajectories to lane direction, and penalizes predictions that violate lane direction (Fig 6.1d). Moreover, we show that the off-yaw rate can be used as a differentiable loss function termed *YawLoss*, which can serve as an auxiliary training loss for multimodal trajectory prediction models. Our formulation of the YawLoss can be applied for training both mixture models as well as latent variable models for trajectory prediction, and leads to predicted trajectories that better conform to the lane direction, while also achieving lower minADE$_K$ values. We report results on the publicly available NuScenes prediction benchmark by incorporating the YawLoss for training two vehicle trajectory prediction models that represent the state of the art, namely MTP proposed by Cui *et al.* [149] and Multipath proposed by Chai *et al.* [150].

6.2 Related Research

6.2.1 Multimodal trajectory prediction

A large body of recent literature has addressed the problem of human and vehicle trajectory prediction. For comprehensive surveys we refer the reader to [174, 175]. Here, we discuss models that output multimodal predictions. A common approach for multimodal trajectory prediction is to learn mixture models. Each mixture component represents a mode of the trajectory distribution. Models typically output mean trajectories for each mode and standard deviations, along with a categorical probability distribution over modes. Early work associated modes of the trajectory distribution with pre-defined maneuvers or intents [60, 61]. The need for pre-defined maneuvers was alleviated by the multiple trajectory prediction (MTP) loss proposed by Cui *et al.* [149]. The MTP loss has a cross-entropy component for learning the categorical probability distribution over modes, and a regression component that only penalizes the mode that is closest to the ground truth.

This formulation has since been used by subsequent works [151–154]. More recently Chai *et al.* [150] extended this idea to learn deviations from anchor trajectories as modes of the trajectory distribution, rather than mean trajectories themselves, and Phan-Minh *et al.* [176] proposed to discard regression outputs altogether while just assigning probabilities to a discrete trajectory set. Another common approach for multimodal trajectory forecasting is learning latent variable models. Conditioned on input context such as past trajectories and static scene, latent variable models map samples from a simple latent distribution to trajectory samples. Prior works have used generative adversarial networks [155–160], conditional variational autoencoders [65, 161–163], and more recently normalizing flow based models [164–167]. Finally, some approaches output multimodal predictions by sampling stochastic policies learned using inverse reinforcement learning [168–171].

While our proposed off-yaw metric and YawLoss can be used in conjunction with any approach that involves regression outputs, here we report results using the MTP and Multipath models as baselines. Both models aim to predict the most likely trajectory of a vehicle from a set of trajectories output by a neural network and their respective probabilities. In the MTP network, a rasterized map containing an overhead view of the surrounding roadyway and vehicles is passed through a CNN backbone, then flattened and concatenated with the ego vehicle's state vector (velocity, acceleration, and heading rate change). This combined vector is then passed through a series of fully-connected layers, ending with an output of M modes comprised of $2H+1$ values each, representing the H (x,y)-values per trajectory plus an associated probability. Similarly, the Multipath model takes the same rasterized map as input, but utilizes a crop around the ego vehicle in between convolutional layers to better feed relevant mid-level features forward in the network. However, the Multipath approach makes use of pre-computed anchors, taken to be the K-mean clusters (or alternative cluster methodology) of the training set trajectories. The network will output M modes comprised of $5H+1$ values each, representing the offset from the anchor in the x and y directions, the three parameters used to define the covariance matrix for the prediction, and the associated mode probability.

6.2.2 Sample quality metrics and auxiliary loss functions for trajectory prediction

As described in section 6.1, the commonly used minADE$_K$ metric for trajectory prediction is a good measure for sample diversity, but can be a poor measure of sample quality or precision. There is inherent tension between sample diversity and sample quality or precision [164]. Several works have thus employed metrics in addition to minADE$_K$ for measuring sample quality of trajectories. Rhinehart *et al.* [164] define a symmetric KL divergence metric with a component that measures sample diversity, and a component that measures sample precision and also use both metrics as loss functions for training. Some works [163, 165, 177] report collision rates for trajectories predicted for multiple actors in the scene, penalizing falsely predicted collisions. Cui *et al.* [178] report kinematic feasibility of predicted vehicle trajectories. Casas *et al.* [153] report lane infractions via traffic light or lane divider violations in predicted trajectories, as well as performance metrics for a downstream planner relying on these predictions. They also use prior knowledge of reachable lanes and the route of the autonomous vehicle to define a reward function for training the trajectory prediction model via the REINFORCE algorithm. Finally, closely related to our work, a large number of approaches use the off-road rate and off-road distance metrics [151, 152, 171–173, 179] for evaluating predicted trajectories. These metrics compute the proportion of predicted points that lie outside the drivable region of the road and the nearest distance of predicted points to the drivable region respectively. Niedoba *et al.* [172], Boulton *et al.* [173] and Messaoud *et al.* [152] also use the off-road rate as a loss function for training trajectory prediction model. Our off-yaw rate and YawLoss improve upon the off-road rate by explicitly reasoning about the direction of motion of lanes and penalizing predicted trajectories that violate it.

6.3 Off-Yaw Rate as a Metric

6.3.1 Off-Yaw Rate

By accepted legal and social convention, when driving in a lane, the vehicle must move in the direction of the lane heading as to not interfere with other traffic. The off-yaw rate is a measure of a trajectory's ability to orient in the direction of the nearest lane.

Define a vehicle's initial position on trajectory τ as $(x_0^\tau, y_0^\tau) = (0,0)$, and its initial orientation in the local frame as $\theta = 0$ aligned with the standard y-axis. Given a trajectory of points $\tau = \{(x_0^\tau, y_0^\tau), (x_1^\tau, y_1^\tau), ..., (x_n^\tau, y_n^\tau)\}$, where points 1 through n correspond to predicted future points, we can estimate the vehicle heading relative to its initial orientation with the following procedure. First, we assume the trajectory sample rate relative to map scale is sufficiently high that we can accept a straight-line approximation between consecutive points. Let $(\hat{x}_i^\tau, \hat{y}_i^\tau)$ be the midpoint of two consecutive points $(x_i^\tau, y_i^\tau), (x_{i+1}^\tau, y_{i+1}^\tau)$, defined by the function:

$$(\hat{x}, \hat{y})(x_1, y_1, x_2, y_2) = (\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}) \tag{6.1}$$

The angle between the same two consecutive trajectory points surrounding $(\hat{x}_i^\tau, \hat{y}_i^\tau)$ is found using

$$\theta(x_1, y_1, x_2, y_2) = \arctan(\frac{x_2 - x_1}{y_2 - y_1}). \tag{6.2}$$

This angle $\theta(x_i^\tau, y_i^\tau, x_{i+1}^\tau, y_{i+1}^\tau)$ is then paired with the midpoint $(\hat{x}_i^\tau, \hat{y}_i^\tau)$, illustrated in Fig. 6.2. From a series of n estimated trajectory points, we create a series of n midpoints and associated headings relative to the initial orientation, which can be converted directly from the local frame to the global frame using the ego vehicle's rotation matrix. We refer to the i-th

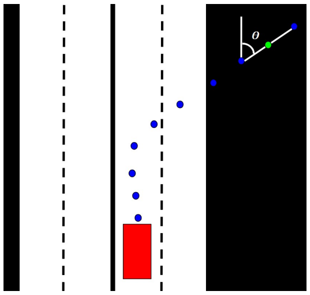


Figure 6.2. The predicted trajectory of the ego vehicle (red) is shown in blue. The green circle represents a midpoint i between two points of the trajectory. The angle θ_i, in the local frame, is assigned to midpoint i.

heading of a trajectory in the local frame as $\theta_{\tau,i}$, and the same heading in the global frame as $\theta^G_{\tau,i}$.

The angular difference between a trajectory midpoint heading in the global frame, θ and the heading of the nearest lane, $\theta_{NL}(x,y)$ can be calculated as follows:

$$\delta(x,y,\theta) = min(\theta - \theta_{NL}(x,y), \theta_{NL}(x,y) - \theta). \tag{6.3}$$

A successful measure of off-yaw driving should increase for any portion of the trajectory τ which deviates from the lane orientation. Further, greater angular differences should be assigned greater values than smaller angular differences. The off-yaw measure of an n-point trajectory is:

$$Y(\tau) = \sum_{i=1}^{n} \delta(\hat{x}_i^\tau, \hat{y}_i^\tau, \theta^G_{\tau,i}). \tag{6.4}$$

Extending over all m predicted modes, we reach the per-sample average off-yaw expression:

$$Y = \sum_{\tau=1}^{m} Y(\tau) \tag{6.5}$$

6.3.2 Lane Change Approximations

There is a small margin of expected angular error, ε, for minor adjustments to the vehicle heading in order to stay within the lane. In addition to lane-correcting error ε, a second exception to the assumption of lane-aligned driving occurs when a driver changes lanes, during which their vehicle may orient at an angle no more than (and typically much less than) 90° to perform the lane change maneuver, with a 90° lane change occurring only when traffic is at a stop. Typical lane changes occur at angles relative to the flow of traffic and vehicle dynamics such as turning radius and velocity. Since a trajectory should not be considered off-yaw during a legal lane change, nor during small-angle lane corrections, we therefore constrain the measure function to only penalize angular differences which exceed a threshold, α. The modified angular difference, $\hat{\delta}_i$, has the following formula:

$$\delta^{\alpha}(x,y,\theta) = \begin{cases} 0 & \delta(x,y,\theta) \leq \alpha \\ \delta(x,y,\theta) & \delta(x,y,\theta) > \alpha \end{cases} \tag{6.6}$$

For our experiments, we selected a threshold of 45°.

6.3.3 Off-Yaw in Intersections

When a vehicle passes through an intersection, the vehicle must cross over lanes which flow in discordant directions (look no further than the existence of stoplights as proof). At these moments, the nearest lane point to the vehicle may belong to a lane which flows in opposite direction, even though it is perfectly reasonable for the vehicle to be in this position. To account for these situations, the measure should not penalize deviation from the heading of the closest lane for midpoints which lie in an intersection. Thus, the measure is modified to drop values which occur in an intersection:

$$Y^{\alpha}(\tau) = \frac{1}{n}\sum_{i=1}^{n} I(x_i^{\tau}, y_i^{\tau})\delta^{\alpha}(x_i^{\tau}, y_i^{\tau}), \tag{6.7}$$

where

$$I(x_i^\tau, y_i^\tau) = \begin{cases} 0 & (x_i^\tau, y_i^\tau) \text{ in intersection} \\ 1 & \text{otherwise} \end{cases} \tag{6.8}$$

.

Summing the values computed for all m predicted modes, we reach the modified per-sample off-yaw measure expression:

$$\bar{Y}^\alpha(T) = \sum_{\tau=1}^{m} Y^\alpha(\tau) \tag{6.9}$$

The off-yaw rate for a set of samples and their predicted trajectory sets is the average fraction of trajectories which contain off-yaw events, defined in the following equation:

$$R_{\text{off-yaw}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{m}\sum_{\tau=1}^{m} Y^\alpha(\tau) \tag{6.10}$$

6.4 YawLoss

6.4.1 Off-Yaw Metric as a Loss Function

In this section, we show that the Off-Yaw Metric in Eq. (6.9) is differentiable, and is therefore suitable as an auxiliary loss function which penalizes vehicle trajectories that move against the flow of traffic, which we name *YawLoss*.

We begin with (6.9) and differentiate with respect to network output set of trajectories $T = \{\tau_1, \tau_2, ..., \tau_m\}$. For brevity, we abbreviate $x_i^\tau, y_i^\tau, x_{i+1}^\tau, y_{i+1}^\tau$ as $\mathbf{x}_i^\tau$.

$$\nabla \bar{Y}^{\alpha}(T) = \frac{1}{m} \sum_{\tau=1}^{m} \nabla Y^{\alpha}(\tau) \tag{6.11}$$

$$= \frac{1}{mn} \sum_{\tau=1}^{m} \sum_{i=1}^{n} \nabla I(x_i^{\tau}, y_i^{\tau}) \delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\mathbf{x}_i^{\tau})) \tag{6.12}$$

Since the sum of differentiable functions is differentiable, we continue our analysis with the sum term:

$$g(\mathbf{x}_i^{\tau}) = \nabla I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}) \delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\mathbf{x}_i^{\tau})) \tag{6.13}$$

Computing the gradient, first for x_i^{τ}, we find:

$$\begin{aligned} \frac{\partial g}{\partial x_i^{\tau}} = & \frac{\partial I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})}{\partial x_i^{\tau}} \delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\mathbf{x}_i^{\tau})) \\ & + I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}) \frac{\partial \delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\mathbf{x}_i^{\tau}))}{\partial x_i^{\tau}} \end{aligned} \tag{6.14}$$

Because the value of the function I in the expression

$$\frac{\partial I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})}{\partial x_i^{\tau}} \tag{6.15}$$

can only take on values of 0 or 1, the gradient function is simply 0 when the vehicle remains on-road or off-road, and the positive or negative reciprocal of the displacement of x_i^{τ} otherwise; in any case, defined for all input.

The function δ^{α} of

$$\frac{\partial \delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\mathbf{x}_i^{\tau}))}{\partial x_i^{\tau}} \tag{6.16}$$

will always give a value in the range [0, 360), so the rate change relative to any distance that the x_i^{τ} coordinate is displaced will be defined for all input. The same cases can be extended to the remaining three variables of differentiation (y_i, x_{i+1}, y_{i+1}), thus making the function $\bar{Y}^{\alpha}(T)$ differentiable and therefore a suitable loss function.

Ultimately, this auxiliary loss function encourages trajectories to stay near lanes whose headings they align with, and to adjust their own headings to more closely match that of the nearest lane. For each midpoint between points in a trajectory, the loss function's value increases as the yaw associated with the midpoint turns further from the yaw of the nearest lane, reaching a maximum when this difference is 180°, and a minimum at 0° or within the provided tolerance threshold.

Because a map-based trajectory should (in regular cases) not predict movement against the flow of traffic, the loss function is appropriate to apply to all trajectories in multimodal models such as MTP and Multipath. This is a unique quality, as other loss functions may be used only for the most-likely mode per sample, to prevent changing a model's prediction for non-relevant trajectories. For example, a trained MTP model may produce a spread of trajectories covering many possible actions as an intersection (left turn, straight, right turn, U-turn, etc.), but during training, MTP loss would rightfully make adjustments to only its left-turn modes when examining a left-turn sample. By contrast, YawLoss enforces a real-world constraint which must apply across all trajectories (that is, a car must not turn into oncoming traffic), and is therefore applicable to every mode simultaneously.

6.5 Experimental Analysis and Evaluations

6.5.1 Dataset

To train and evaluate our model, we use the public nuScenes dataset [180], containing real-world inner-city drives conducted in Boston and Singapore, where each sample includes a raster of the surrounding map, vehicle state information (velocity, acceleration, heading), and target trajectory. Ego vehicle information is encoded with a color index (in this case, red) while surround vehicles are provided a different color (yellow); darker shade renderings of the respective vehicle are used to indicate vehicle location at past time samples, as a means of illustrating prior motion from a single image. Our data is divided using the official benchmark split for the nuScenes prediction challenge; in total, we used 29889 instances in the train set, 7905 instances in the validation set, and 8397 instances in the test set.

6.5.2 Network Architecture and Implementation Details

As introduced in Section II, we perform experiments using both the MTP network defined in [149] and the MultiPath network defined in [150]. For our experiments, we use a network output of 15 modes with 12 predicted points per mode (representing 6 seconds of travel) for MTP, and 12 predicted offsets per anchor for MultiPath. We use a base CNN of ResNet-50 [181]. In accordance with the expected input to ResNet with ImageNet dataset pretraining, we normalize our rasterized map images in RGB space prior to training. We use the classification and regression loss functions as defined in [149], with an additive term for the lane heading auxiliary loss (YawLoss) defined in this work, with a scaling hyperparameter of 1.

With earlier described rasterized map physical dimensions of 50 meters x 50 meters, using a scale of 0.1 meters per pixel, we assume the lane and trajectory to be approximately straight (i.e. of single uniform heading) on the pixel scale. Each scene map contains information on lane placement and heading, drivable area, and surrounding vehicles and pedestrians. Vehicle state is provided as a three-dimensional input. We use a batch size of 16 and Adam optimizer [182], implemented using PyTorch [183].

6.5.3 Reducing Network Training Time & Memory Requirements with Secondary Maps

Calculating this loss per-sample can be computationally expensive. For every predicted mode of each sample instance, it is required to find the $L2$-nearest lane point to each midpoint on the predicted trajectory, with predictions changing on every iteration.

This computational hurdle can be lowered through preprocessing; for each instance map, which in our case extends 10 meters behind the vehicle, 40 meters ahead, 25 meters left, and 25 meters right, we generate a secondary orientation map, covering a larger area to account for trajectories which leave the original map. This secondary map extends 20 meters behind the vehicle, 80 meters ahead, 50 meters left, and 50 meters right. On this map, each pixel location is assigned a value which equals the orientation of the nearest lane point.

These secondary maps are generated and saved for each data sample prior to training. Each grid location on the map represents a heading from the continuous range [0, 360) degrees in the global frame. To represent each grid location as a 64-bit floating point value can quickly become storage intensive for a large set of 500x500 maps. However, only a coarse precision of the angle is required for this problem; we would never consider a driver to be going the ‘wrong way’ if their heading was off by just a few degrees. For this reason, a representation with precision only to the scale of degrees is appropriate for this problem. With this in mind, we can create a data-efficient representation which encodes each heading as an 8-bit grayscale integer pixel value in the range [1, 255], with the value of 0 reserved for map locations corresponding to intersections. Headings are mapped from range [0, 360) degree values to [1, 255] grayscale values as follows:

$$\theta_{map} = 1 + \lfloor \frac{254}{360}\theta \rceil . \tag{6.17}$$

Using the above function, we assign to each point on the secondary map the mapped value of the heading of the $L2$-nearest lane, illustrated in Fig. 6.3. During training, when inversely mapping from grayscale integer to degrees, there is a loss of precision that occurs as the 360 degrees are mapped to 254 values. In this sense, each ‘bin’ of the data representation actually represents a span of approximately 1.417°, a reasonable precision for this task.

6.5.4 Baselines and Metrics

Results are shown in comparison to the following baselines:

- Constant Velocity, Yaw: The predicted trajectory is a continuation of the vehicle’s current velocity and heading.

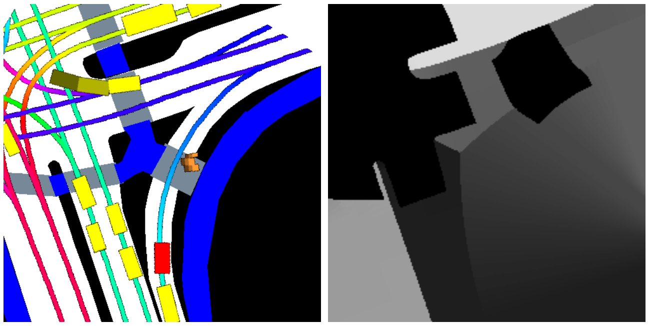


Figure 6.3. Left: The rasterized bird's-eye-view RGB input map for a sample. Right: The secondary map for the same sample, where each pixel maps to the approximate heading of the nearest lane, or zero if in an intersection. Each pixel's shade of gray represents the orientation of the nearest lane to the pixel. Areas of intersection (i.e. multiple lanes converging or crossing) are given a value of 0 in the grayscale map to represent the ambiguity between the nearest lane and the driver's intended lane in such situations.

Table 6.1. Results of comparative analysis of different models on the nuScenes dataset, over a prediction horizon of 6-seconds. Variants of MultiPath and MTP are grouped for comparison on nine selected metrics. In general, models using YawLoss (this research) improve over the baseline on most metrics.

	$MinADE_1 \downarrow$	$MinADE_5 \downarrow$	$MinADE_{10} \downarrow$	$MinFDE_1 \downarrow$	$MinFDE_5 \downarrow$	$MinFDE_{10} \downarrow$	$MissRate_{5,2} \downarrow$	$MissRate_{10,2} \downarrow$	$Off-RoadRate \downarrow$
Constant Velocity, Yaw	4.61	4.61	4.61	11.21	11.21	11.21	0.91	0.91	0.14
Physics Oracle	3.69	3.69	3.69	9.06	9.06	9.06	0.88	0.88	0.12
MultiPath	4.06	**1.63**	**1.50**	9.34	3.36	3.00	**0.75**	**0.74**	0.40
MultiPath with YawLoss	**3.95**	**1.63**	**1.50**	**9.08**	**3.33**	**2.95**	**0.75**	**0.74**	**0.38**
MTP	4.59	2.44	**1.57**	10.75	5.37	3.16	0.70	**0.55**	0.11
MTP with Off Road Loss	4.51	**2.16**	1.60	10.44	**4.73**	3.23	0.72	0.58	0.13
MTP with YawLoss	**4.16**	2.23	**1.57**	**9.65**	4.85	**3.14**	**0.69**	0.56	**0.10**

- Physics Oracle: As introduced in [176], the proposed trajectory is selected as the best trajectory from four dynamics models: constant velocity and yaw, constant velocity and yaw rate, constant acceleration and yaw, and constant acceleration and yaw rate. Note that this method is not used to make predictions, but rather provides a reference benchmark to four simple physical models, to illustrate improvement from models which account for more complex maneuvers.

- MultiPath: The predicted trajectories are the output of the MultiPath model, as described in [150].

- MTP: The predicted trajectories are the output of the original MTP model, as described in [149].

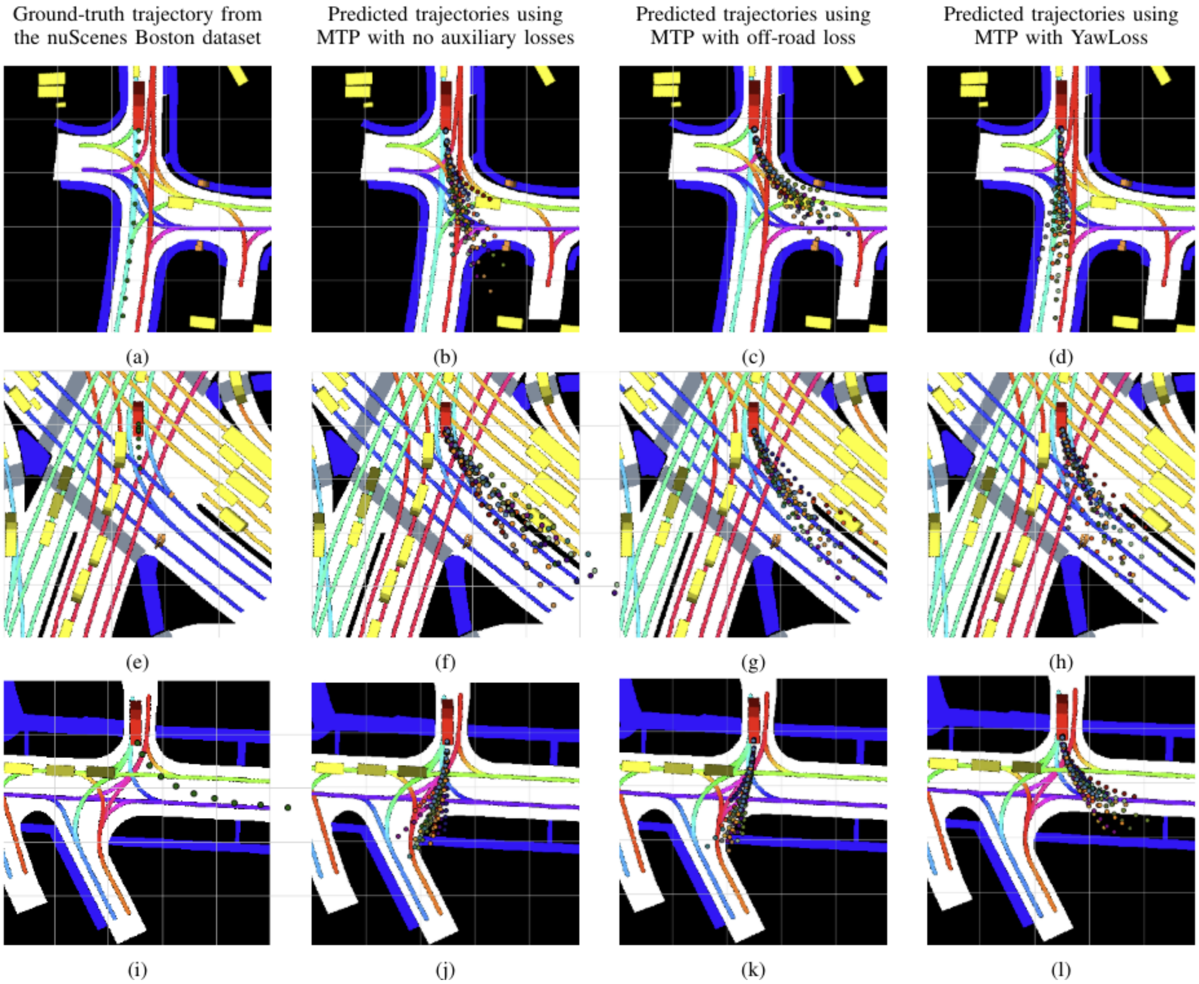

Ground-truth trajectory from the nuScenes Boston dataset
Predicted trajectories using MTP with no auxiliary losses
Predicted trajectories using MTP with off-road loss
Predicted trajectories using MTP with YawLoss
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)

Figure 6.4. Three examples of improved trajectory prediction using YawLoss. Each row represents a naturalistic Boston driving scenario from the nuScenes dataset. The first column contains the ground-truth trajectory, and the second column contains predictions by the standard MTP model. In the third column, the model is extended with off-road loss. While all three off-road loss examples show trajectories closer to a drivable area, trajectories in the third column are incorrectly pushed into oncoming traffic. By contrast, examples trained with YawLoss (fourth column) show trajectories restored to the drivable area into lanes with the correct heading.

Table 6.2. Results of comparative analysis of off-yaw rate between two versions of the nuScenes dataset, over a prediction horizon of 6-seconds. The first version is the full validation set, and the second version excludes trajectories whose ground truth contains points within an intersection or off the rasterized map.

Off-Yaw Rates [rad] ↓:	All Scenarios	No Intersections
MultiPath	0.375	0.280
MultiPath with YawLoss	**0.367**	**0.276**
MTP	**0.114**	0.110
MTP with YawLoss	0.124	**0.097**

Reported metrics include minimum average displacement error ($MinADE_k$), minimum final displacement error ($MinFDE_k$), miss rate at 2 meters ($MissRate_{k,2}$), off-road rate, and off-yaw rate (the new metric as defined in this paper, measuring the amount of positive angular difference of predicted trajectories from the nearest lane yaw, averaged over all agents. $MinADE_k$, $MinFDE_k$, and $MissRate_{k,2}$ are taken over the k most probable trajectories, for $k = 1, 5,$ and 10. While $k = 1$ is generally helpful to evaluate precision of trajectory prediction, in cases when the most probable trajectory is incorrect, the metric value for trajectories comprised of modes which take the average of multiple paths (e.g. going straight when deciding between a left and a right turn) will outperform an incorrect turn for $k = 1$. Thus, including higher k values evaluates whether the model has developed diversity of modes. In all cases, optimal values minimize these displacement errors.

6.5.5 Quantitative Results

We compare our extension of the MTP and MultiPath models to the various baselines in Table 6.1. Our MTP model outperforms or matches the non-extended MTP model on 8 of the 9 reported metrics, the exception being a .01 increase of Miss Rate at 2 m for $k = 10$. In contrast, the MTP model with off-road loss outperforms the baseline on just 4 of the 9 reported metrics. Our MultiPath model outperforms or matches the non-extended MultiPath model on all 9 reported metrics. These improvements suggest that using YawLoss to extend the models

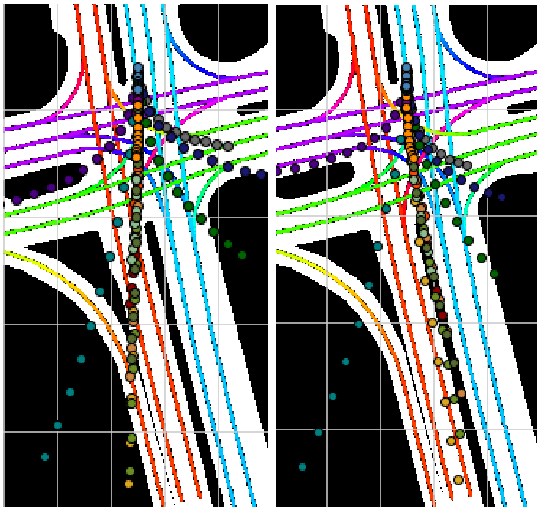


Figure 6.5. Predicted trajectories using MultiPath with (left) and without (right) our auxiliary YawLoss, illustrating the influence of intersection and off-map points on the calculation of the Off-Yaw Metric. The trajectories in the left image have 31 more intersection points (which contribute no penalty to the metric), so the left trajectories have a much lower off-yaw rate (0.26 difference) despite being less aligned to their lanes.

created trajectories which have points more closely aligned to the ground truth trajectories and better maintain paths on drivable regions. Additionally, the predicted final location of the vehicle is more close to the known destination.

Qualitative illustrations comparing the effects of Off-Road Loss and YawLoss on an MTP base model are shown in Fig. 6.4. As the scenes demonstrate, while off-road loss is effective at bringing trajectories closer to the drivable area, YawLoss is more effective at bringing trajectories closer to the drivable area with the correct heading.

It is interesting to note that off-yaw rates are similar regardless of auxiliary loss, and in fact sometimes slightly higher when using YawLoss. By Equation 6.8, a non-linearity is introduced for points within an intersection or outside the map region, where the additive rate term is dropped to 0 instantaneously. Thus, it is possible that trajectories at higher velocity (i.e. more likely to leave the drivable region) and trajectories comprising intersection points, even if further from the ground truth, may receive a lower off-yaw measure than an correct trajectory which leaves the intersection or stays within the map. An illustration of this behavior is shown in

Fig. 6.5, with a qualitative comparison of the complete dataset with and without intersection and off-map points provided in Table 6.2. For this dataset, the off-yaw rate rises when using MTP with YawLoss, while it expectedly decreases when we only consider samples that do not contain this sudden non-linearity. Thus, as a comparative tool, YawLoss is most useful when comparing samples with the same number of non-intersection, on-map points.

6.6 Concluding Remarks

In this paper we presented an auxiliary loss function which may be used to augment the performance of existing models for vehicle trajectory prediction in urban environments. This lane heading loss function leverages the expectation that vehicles follow the direction ascribed to roadway lanes at all times, with exception for corrective maneuvers, lane changes, and intersection crossings. This loss function applies to all predicted modes, since no mode should predict driving opposite the lane direction. Experiments showed that extending the benchmark MTP model with the lane heading auxiliary loss outperforms the model's original classification and regression losses.

A possible extension of this work would be the application of the lane heading auxiliary loss to other existing deep learning models, in tandem with other auxiliary losses such as off-road loss. Another possibility for future investigation is the tuning of the angular difference threshold and weighting using agent dynamics and scene context. Finally, in our future work, we intend to design a methodology for quantifying nearest lane heading within an intersection or outside of the drivable area to reduce the effect of this non-linearity on training and metric reporting.

As stated by Daily et al. [184], “Self-driving and highly automated vehicles must navigate smoothly and avoid obstacles, while accurately understanding the highly complex semantic interpretation of scene and dynamic activities." While convolutional neural networks and other data-driven approaches may be effective at repeating known patterns, there is a lost element of explainability which is crucial towards public safety and adoption. By encoding familiar driving expectations through the introduced off-yaw rate metric and YawLoss, we initiate a step towards autonomous vehicle computational models which can both learn and explain.

Acknowledgements

Chapter 6, in part, is a reprint of the material as it appears in Greer, Ross, Nachiket Deo, and Mohan M. Trivedi. "Trajectory prediction in autonomous driving with a lane heading auxiliary loss." IEEE Robotics and Automation Letters 6 (3), 4907-4914 (2021). The dissertation author was the primary investigator and author of this paper.

Chapter 7

Perception Without Vision for Trajectory Prediction: Ego Vehicle Dynamics as Scene Representation for Efficient Active Learning in Autonomous Driving

7.1 Introduction

The accurate prediction of the trajectories of agents in the observed environment is paramount to the safe path planning of autonomous systems. Whether the agents are observed from infrastructure, the ego vehicle, or some combination of modalities, forecasting where other vehicles and pedestrians helps intelligent systems (human and machine alike) to make their own control decisions.

Machine learning has provided a means for trajectory prediction of traffic agents using rasterized bird's-eye-view maps, contextual scene information, and social dynamics [63, 149]. Road infrastructure [64, 185], agent occupancy [60], and navigation goals [171] largely determine where and how a vehicle will move through the environment. However, collection and especially annotation of data for such systems can be costly. Methods in trajectory prediction and planning rely on the ability to perceive road infrastructure and agents; for example, in methods which use a bird's-eye-view map of the scene to predict a trajectory, the data must include accurate annotations of the position of scene agents, lane markings, and intersections. While the trajectory itself

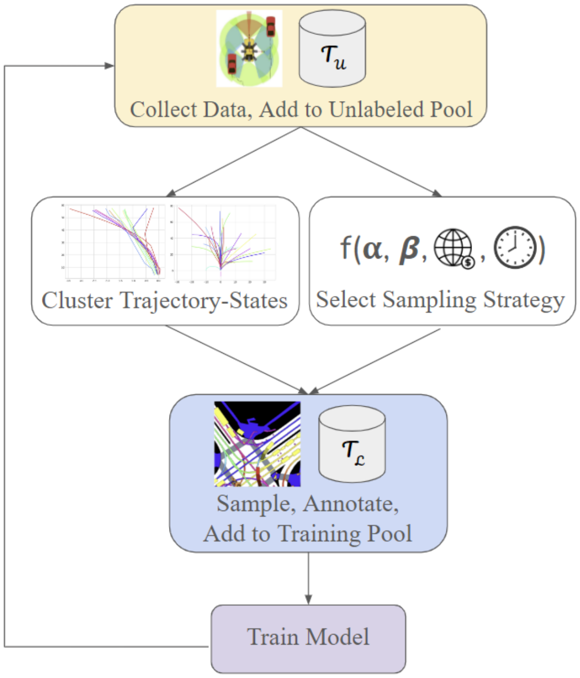


Figure 7.1. In supervised learning for tasks such as trajectory prediction, data is collected (yellow), annotated and added to a training pool (blue), and then a model is trained (purple). When more data is collected than can be afforded by an annotation or computational budget, intelligent sampling using active learning (white) may provide solutions which maintain model performance at reduced data cost. We contribute algorithms for clustering of trajectory-states and sampling strategies which are model-agnostic, providing a benefit of active learning based only on the current training data and without requiring computation of uncertainty from the partially-trained model.

can be quickly collected from onboard positioning sensors, the annotation of the surrounding scene which informs the driving decision-making is a costly effort [32, 100]. In this research, we consider the utility of trajectory data as the basis of acquisition functions for the purposes of active, semi-, or self-supervised learning [37, 114, 186]; in other words, how might information on a vehicle's positioning help us to curate data for efficient machine learning using minimal annotation budgets?

To help illustrate this idea, consider a situation where you are a passenger in a vehicle driving with modern advanced driver assistance functionality, and perhaps you are tired and decide to close your eyes. You may experience a variety of kinematic cues even without vision; you may feel the car come to a stop, and after a few moments (or perhaps a bit longer), you feel the car turn to the left, then continue smoothly. Even though you have no vision of the environment, there are many pieces of information which you can already gleam from these dynamics alone. First, you came to a stop - this does not happen without a reason. Perhaps you approached a stop sign, a red traffic light, or a person crossing the road. You then waited for a bit (presumably, enough to come to a complete stop and wait until safe to proceed, or the light turns green, or the person finishes crossing). Then, you made a left turn, meaning that you were likely at some kind of intersection, and depending on your wait, possibly with other agents. In any of the above cases, from the trajectory alone, you would be able to reasonably infer that you are not cruising on the freeway - and with enough examples like this, you may be able to recognize patterns in the dynamics which relate to the outside scene, all without observing the outside scene!

In this way, we propose that trajectory information shares mutual information with the visual observation of a scene, and that we can use this trajectory information in an unsupervised manner to inform our data curation process for autonomous driving machine learning tasks, to promote diversity in our data. Having data which covers the input space as thoroughly as possible is critical to robust learning [187].

Towards the continued development of such techniques, this research presents methods of curating and integrating further training data for such systems, such that systems can efficiently learn new behavioral patterns and adapt to changes in the open set of real-world driving scenarios. Our contributions are as follows:

1. Demonstration of the ability of trajectory and vehicle dynamic state information to be clustered for the purposes of learning acquisition functions, and algorithms for such acquisition in active and continual learning settings,

2. Presentation of sampling techniques related to (a) breadth and depth of data clusters and (b) introduction of novelty,

3. Discussion of the relevant data features toward an example task of trajectory prediction,

with relation to the cost of annotation and benefit to learning systems, and discussion of extension to related tasks of object detection and path planning,

4. Empirical analysis of the learning phase transition with respect to novel data, and

5. Empirical analysis of the effectiveness of novelty-sensitive sampling in an active learning experiment, illustrating the potential of the system to continually learn from intelligently-selected new data.

7.2 Related Research

Machine learning relies on transforming data into separable representations, and to do so effectively, requires training data which approximately covers the variance of data expected to be encountered in deployment in the real-world. To this end, active learning is a method by which data is incrementally annotated and added to a training pool for a machine learning system, selected in a strategic manner for efficiency over an annotation budget. Broadly, these methods are divided into uncertainty-driven methods, which take into account a model's level of confidence in its prediction of an unlabeled datum, and diversity-driven methods, which take into account the relationship of a datum to all other data [188, 189]. Active learning has been useful in supporting a variety of autonomous driving tasks such as vehicle detection, recognition, and tracking [122–124].

Hacohen et al. define, derive, and empirically support the existence of an active learning “phase transition" in model performance with respect to data *typicality* [31]. The term *typicality* is used to describe points in a high-density region of the input space, analogous but opposite to the meaning of *diversity* for such tasks, and without regard for model *certainty*. Hacohen et al. show that on low budgets, sampling typical data is most beneficial, while on high budgets, sampling least typical data is most beneficial. They evaluate their hypothesis on three image classification tasks (CIFAR-10, CIFAR-100, and ImageNet-100). Important to this research, they also remind readers that their work is especially relevant for applications which require “an expert tagger whose time is expensive", and autonomous driving certainly falls into this category, where companies frequently outsource data annotation to teams of taggers [148], whose expertise and attention directly influence safety outcomes of algorithms trained on this annotated data [75, 190]. Their discussion of the importance is not just related to efficiency; when data is within the “low budget" regime, general active learning methods fail to surpass random sampling! This is referred to as the *cold start problem* [191], and may be a consequence of early models being without the critical mass of data to form accurate measurements of its “uncertainty" of unlabeled points.

This provides a few implications relevant for tasks in autonomous driving and, more generally, robotics:

1. It is important to identify at what data volume this phase transition occurs. Without awareness of the phase transition, because of the cold start problem, one cannot identify whether to employ active learning, and even then which active learning method to employ.
2. Once the phase transition is identified, selecting the right learning strategy will depend on defining a notion of “typicality" (or, in dual, “novelty") which is pertinent to the domain, task, and data at hand.

In our experiments, we present evidence for this phase transition within data systems for an example task of trajectory prediction, and provide a measurement of typicality useful for clustering such data in the domain.

7.3 Novelty-Sensitive Active Learning Algorithm using Trajectories and Dynamic States

Trajectory and dynamic information [192, 193] is particularly low-cost to collect and annotate. Assuming a well-calibrated GPS and IMU system, the vehicle is localized and trajectories can be reconstructed in a 2D overhead projection, along with state variables such as velocity, accleration, and heading. This requires virtually no annotation, as opposed to 2D or

3D objects in a scene, which require meticulous annotation. In the next sections, we describe ways that the low-cost information can be leveraged to curate only particular, learning-efficient scenes (which can then be expensively annotated) for an overall reduction in data budget while maintaining performance.

7.3.1 Sampling Iteration Parameterization

We begin with an assumption that it is possible to identify novel data using unsupervised techniques, which we detail in the following sections. We adopt a clustering approach, where any data sufficiently distant from centers of clusters with members in the training pool are considered novel. From this, we consider two parameters which define our sampling mechanism: α, representing the proportion of novel data which should be sampled (where $1 - \alpha$ is the proportion of training-pool-similar data to be sampled), and β, the proportion of each cluster allowed to be sampled (in other words, how many instances of a novel concept can be added in a sampling iteration). For a fixed annotation budget, α and β can be tuned to manage the breadth of novel clusters sampled and the depth with which a novel cluster is sampled.

7.3.2 Acquisition Function using Trajectory and State Similarity Clustering

For the purposes of prediction, one formulation of an autonomous driving trajectory involves the combination of 2D ground plane world coordinates that the vehicle will occupy for n seconds sampled at rate r, as well as any initial state variables s that describe the agent at the beginning of the prediction period. In the case of nuScenes, for example, data is sampled at 2 Hz for 6 seconds, and s is comprised of the vehicle velocity v, accleration a, and heading change rate h.

We define a measure of similarity between two vehicle trajectory-states V_i and V_j as

$$d(V_i,V_j) = \sum_{n=1}^{n=12} \sqrt{(x_{i,n}-x_{j,n})^2+(y_{i,n}-y_{j,n})^2} \\ + k_a|a_i-a_j| + k_v|v_i-v_j| + k_h|h_i-h_j| \quad (7.1)$$

We use k as a scaling parameter to weigh different aspects of the vehicle state in relation to the position error. In our experiments, we use $k_h = 1$ since heading change rates tend to range from -0.5 to 0.5, $k_v = 1/40$, with velocity ranging from 0 to 20, and $k_a = 1/20$, with acceleration ranging from -5 to 5. These values can be further changed to reflect the importance of different parts of the trajectory-state for clustering applications to identify different types of trajectory corner cases [194].

We apply the hierarchical clustering algorithm [79, 80], using the average distance of all points in a cluster in re-assigning cluster distances when constructing the dendrogram (i.e. unweighted pair group method with arithmetic mean). A threshold τ is applied to estimate the flat clusters, such that the cophenetic distance between any pair within one of the flat clusters is no greater than τ. We use $\tau = 10$ in our experiments. Results of clustering are illustrated in Figure 7.2, showing just 12 of the 3,267 clusters formed in application of this algorithm to our experimental dataset, sampled at random. We also randomly sample a subset of 20 of the “novel" trajectories (i.e. those which did not belong to a cluster), shown in Figure 7.3 and illustrating that our trajectory-state distance measurement is effective in grouping like-trajectories and separating unique trajectories.

7.3.3 Active Learning Algorithm

With the clusters available, it now becomes possible to use this unsupervised information within an active learning algorithm to enhance the learner’s ability to intelligently acquire new data for annotation. This method does not require any measure of uncertainty from the learning model, but does require awareness of the currently-annotated training pool, as the membership of

Algorithm 5: Novelty-Sensitive Active Learning Round

```
Require : $\alpha$, $\beta$, initial training pool $T_l$, unlabeled pool $T_u$, and data budget $B$
novel samples $\leftarrow \emptyset$ ;
familiar samples $\leftarrow \emptyset$ ;
Cluster($trajectorystates\ v_i \in T_l \cup T_u$) ;
while $\|$novel samples$\| < \alpha \times B$ do
    $C_n \leftarrow c_i \| \forall v_i \in c_i, v_i \notin T_l$;
    $V_n \leftarrow v_i \| \forall c_i, v_i \notin c_i$;
    $S_n \leftarrow$ RandomSelect($s_i \in C_n \cup V_n$) ;
    if $S_n \in C_n$ then
        for $i = 1$ to $\beta \times \|S_n\|$ do
            novel samples += RandomSelect($v_i \in S_n$) ;
    else
        novel samples$+ = S_n$ ;
Annotate(novel samples);
$T_l$ += novel samples;
while $\|$familiar samples$\| < (1 - \alpha) \times B$ do
    $C_t \leftarrow c_i \| \forall v_i \in c_i, \exists v_i \in T_l$;
    for $i = 1$ to $\beta \times \|C_t\|$ do
        familiar samples$+ =$ RandomSelect($s_i \in C_t$);
Annotate(familiar samples);
$T_l + =$ familiar samples ;
```

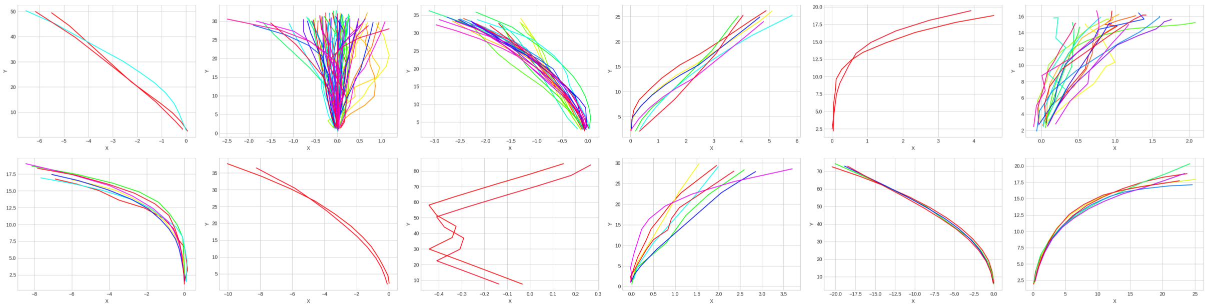


Figure 7.2. We randomly select 12 clusters, formed using our distance measurement over trajectory-states (which include trajectory coordinates and vehicle dynamics). Comparing across the selected clusters, clear patterns emerge even over the 2D coordinates alone (visualized), showing the effectiveness of grouping like-trajectories.

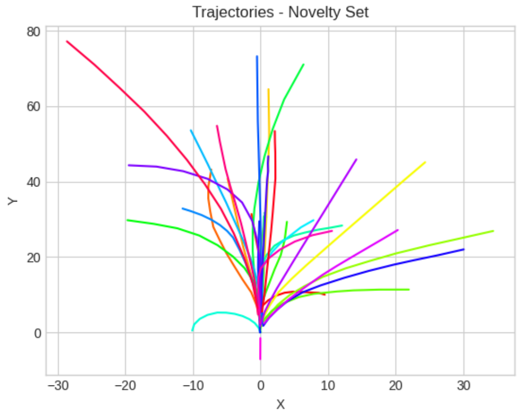


Figure 7.3. Many trajectory-states remain unclustered due to sufficient distance from all nearest trajectory-state clusters. We randomly sample just 20 of these unmatched trajectory-states, visualizing the 2D path coordinates and illustrating the diversity of behaviors found to be unique within the dataset.

annotated trajectories within a cluster may disqualify that cluster from being acquired as ‘novel’.

We present our algorithm in Algorithm 5. In summary, two parameters are used to define the included breadth (amount of ‘novel’ samples added) and depth (how much of a cluster to be added) of the acquisition and annotation of new samples to the training pool. Up to these limits, clusters which are unrepresented in the training data, or singleton unclustered unique instances, can be drawn and added to the training pool, until the data budget is filled. In the case that no further sampling of the desired type is possible (e.g. there are no unvisited clusters or unique samples remaining), samples are drawn at random from the entire unlabeled pool.

7.4 Evaluation: Real-World Data and Experimental Design

7.4.1 Datasets

We perform our experiments on the nuScenes dataset, using a subset of the public “train" training split for training and another for validation, and the public “train_val" subset for testing. The nuScenes dataset contains 850 driving scenes for training and 150 for evaluation, divided into instances with 2 seconds of past history to be used in predicting 6 seconds into the future. Information on ego and surround vehicle state are available, as well as map structure (including lanes and intersections).

7.4.2 Experimental Design

We sweep through α and β parameters in 20% increments, beginning at $\alpha = 0$ (no novel data) and $\beta = .2$ (maximum number of samples from a given cluster is 20% of the cluster size). We repeat this sweep on 5 training volumes: 10% of the dataset through 50% of the dataset, in 10% increments. Results are illustrated in Figures 7.4 and 7.5 and summarized in Table 7.1.

We use the Prediction via Graph-based Policy (PGP) model [195] as the trajectory prediction model for training in the active learning framework. PGP learns discrete policies, exploring lane graph goals and waypoints with consideration for both lateral variability (lanekeeping, turning) and longitudinal variability (acceleration). PGP is one of the top models in trajectory prediction at the time of this writing, with top-3 performance on minimum average displacement and miss rate metrics of the nuScenes leaderboard, but regardless, the methods described in this paper are applicable to any machine learning system for trajectory prediction.

7.5 Analysis of Results

We present the results of our experiment in Table 7.1. This table presents performance of the best-performing active learning strategy in comparison to a random baseline on two common trajectory prediction metrics, the minimum average displacement error over the five

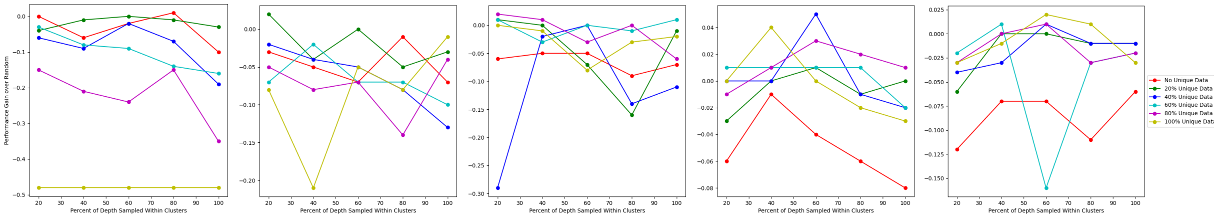


Figure 7.4. These five graphs represent the minimum average displacement error metric ($mADE_5$) performance of various parameterizations of the active learning strategy over a random baseline, considering the 5 most likely trajectory predictions from the model. Positive numbers indicate improvement over random. From left to right, each graph has a different training pool size, with the amount of data in the training pool increases from 10% to 50% of nuScenes (in 10% increments). The y-axis represents improvement over random, while the x-axis represents the allowable “depth" into a cluster that the algorithm samples. Each color line represents a different proportion of unique (novel, diverse) data, versus resampling data which is similar (typical) to data which already exists in the training pool. The point that we seek to highlight is the change in position of the yellow line (all novel data) and the red line (all typical data). We see that as the annotation budget or training pool size increases, these two trends effectively switch roles in over- (or under-) performing relative to the random baseline. This pattern matches the findings of Guy et al. in image classification tasks, providing evidence for the presence of the active learning phase transition within the trajectory prediction task - and, within the bounds of the nuScenes dataset size. Sampling typical data helps in overcoming a cold start, while novel data should be sampled in higher proportion as the training pool grows.

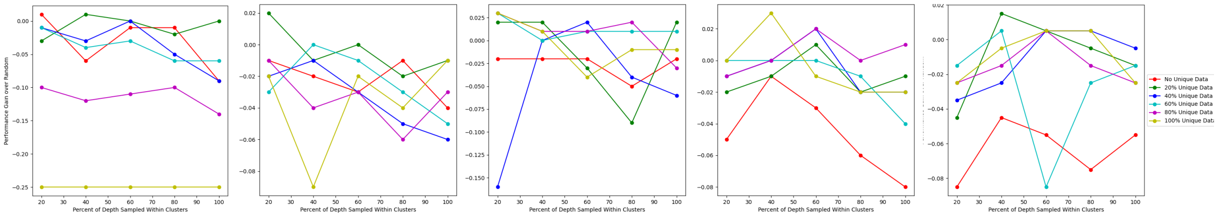


Figure 7.5. These five graphs represent the minimum average displacement error metric ($mADE_{10}$) performance of various parameterizations of the active learning strategy over a random baseline, considering the 10 most likely trajectory predictions from the model. Positive numbers indicate improvement over random. From left to right, each graph has a different training pool size, with the amount of data in the training pool increases from 10% to 50% of nuScenes (in 10% increments). The y-axis represents improvement over random, while the x-axis represents the allowable “depth" into a cluster that the algorithm samples. Each color line represents a different proportion of unique (novel, diverse) data, versus resampling data which is similar (typical) to data which already exists in the training pool. We observe the same pattern as noted in the graphs of $mADE_5$, in the transposition of performance of the strategy which samples novel data and the strategy which samples typical data.

Table 7.1. Performance of the best-performing active learning strategy in comparison to a random baseline on two common trajectory prediction metrics, taken at five data pool sizes.

Labeled Pool	mADE5			mADE10		
%	Random	Active	α,β	Random	Active	α,β
10%	1.59	**1.58** (−0.01)	(0%, 20%)	1.18	**1.17** (−0.01)	(0%, 60%)
20%	1.44	**1.42** (−0.02)	(20%, 20%)	1.08	**1.06** (−0.02)	(20%, 20%)
30%	1.37	**1.35** (−0.02)	(80%, 20%)	1.04	**1.01** (−0.03)	(60%, 20%)
40%	1.35	**1.30** (−0.05)	(40%, 60%)	1.00	**0.97** (−0.03)	(100%, 40%)
50%	1.31	**1.29** (−0.02)	(100%, 40%)	0.97	**0.96** (−0.01)	(20%, 40%)
SoA 100% [195]	**1.30**			**1.00**		

highest-probability trajectories (*minADE*$_5$), and the same error over the ten highest-probability trajectories (*minADE*$_{10}$). We measure these values at five different data pool sizes, up to 50% of the complete nuScenes training set. We also compare to the performance reported by [195] in the original PGP paper. Our diversity-driven active learning methods show consistent performance gains over random sampling, surpassing or equivalent at all data pool sizes, and even reaching (or surpassing) performance on the full dataset at just a fraction of the training pool size.

These methods do rely on selection of α,β parameters; in our table, we have the experimental luxury of providing the optimal values, but in practice, this would require some assessment of whether a model for a particular task has passed the point of inflection for active learning “phase"; that is, whether or not it is more beneficial to sample *typical* data or *novel* data. The trend in our table, and in the associated figures, is still apparent: it is beneficial to sample typicality at the beginning, to address the “cold-start problem", and as the data budget increases, begin introducing more-and-more novelty. We see at the 20% budget, we accept 20% novelty (one increment up from the initial 0%), and in the higher budget sizes of 30-50%, we begin finding the higher novelty α values to be optimal, making the case for some form of “novelty scheduling" to be integrated into learning systems as a means of active learning.

Qualitative results are depicted in Figures 7.6-7.9, with examples of model results from the ten percent data pool, fifty percent random data pool, and fifty percent active learning pool. Specific cases are discussed in the figure captions, and one pattern that emerges between examples is the pace of the model’s learning of lane-conforming behavior and multimodality.

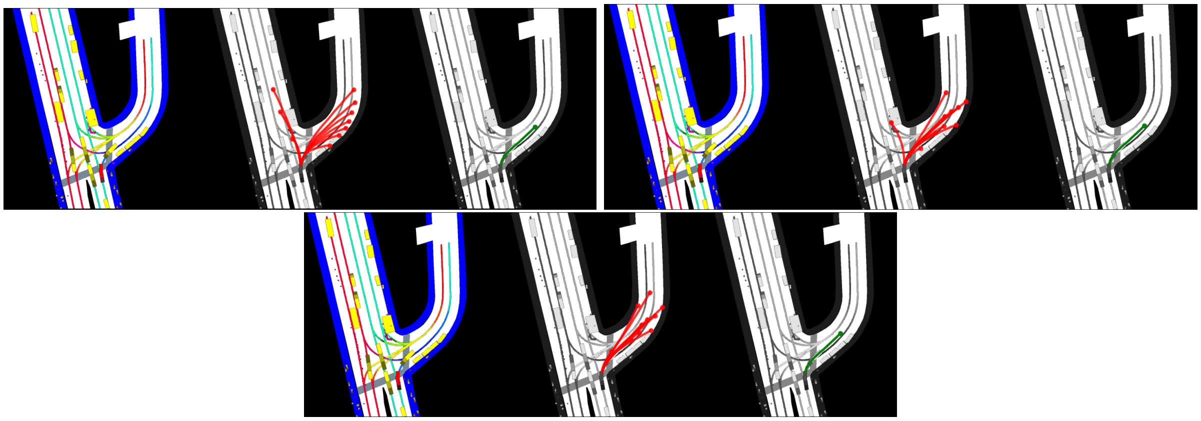


Figure 7.6. From left to right: map input, predicted trajectories, and ground truth. From top to bottom, results from models trained on: 10% training data, 50% training data randomly selected, and 50% training data selected using our active learning algorithm. The 10% data model shows a large spread of possible trajectories, with little scene conformity. Though the scene conformity improves at 50% data, with active learning, the trajectories adapt even better to the lane contours of the scene.

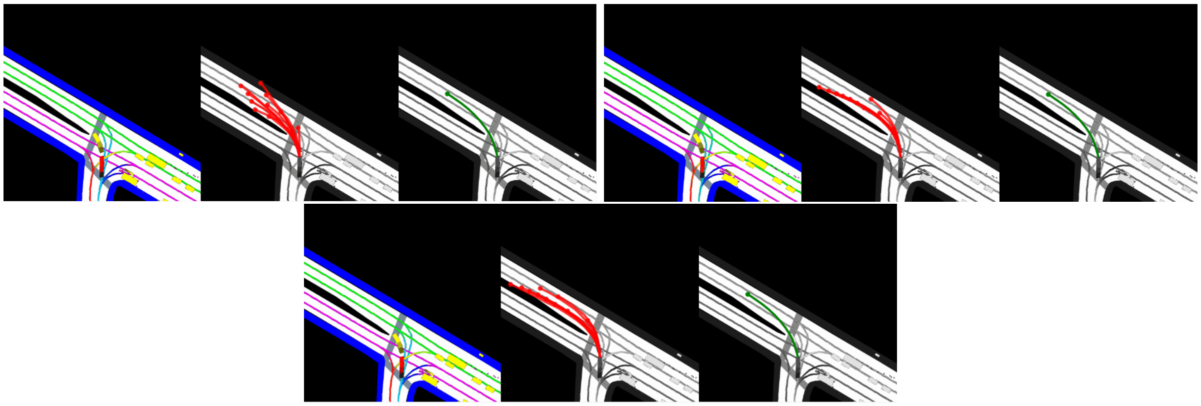


Figure 7.7. As in the previous example, though the scene conformity improves at 50% data, with active learning, the trajectories snap more closely to the lanes in the freeway, while maintaining the multimodal options appropriate for the driver's choice in the scene.

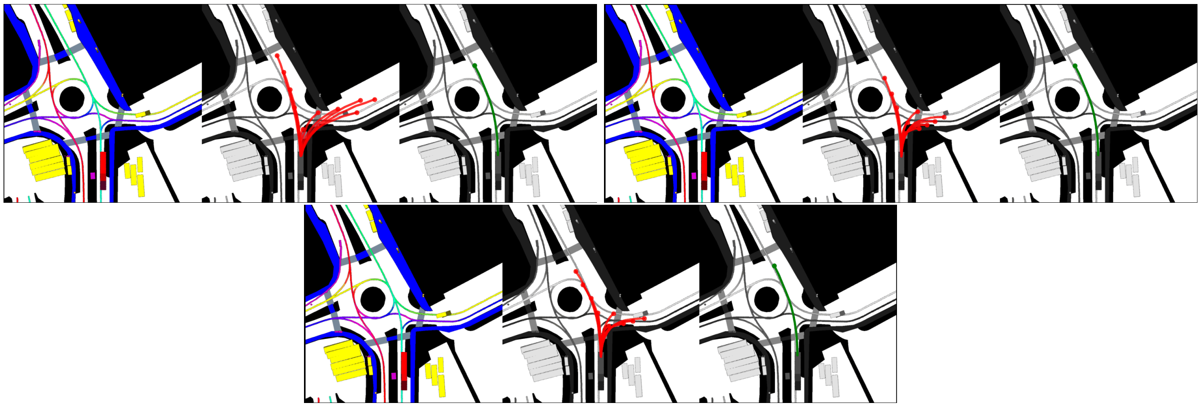


Figure 7.8. At the roundabout, many trajectories proposed by the 10% training data model are non-compliant, and while the 50% randomly-selected training data model shows a better conformity to the two possible modes, the 'right turn' mode still has four very distinct (incorrect) variants. At 50% active-learning-selected training data, these four variants collapse to one (scene-appropriate) turn.

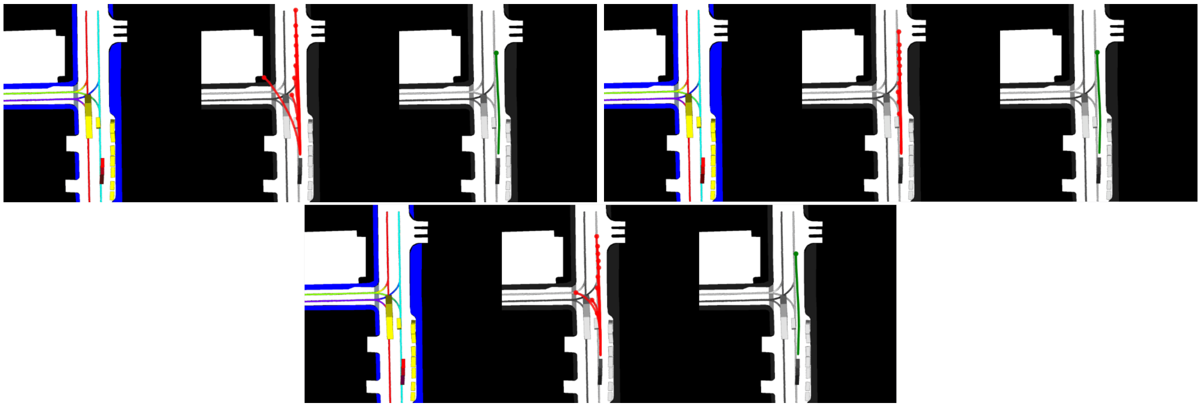


Figure 7.9. Though the vehicle continues straight in this example, a good trajectory prediction model should maintain an awareness of the possibility of a left turn, as the future (in this case) is truly unpredictable since the driver has agency to elect to take the turn. The 10% training data model has no real understanding of the map, and only produces some kinematically-possible modes which are inappropriate to the scene. The 50% randomly selected training data model loses the mode for the left turn, while the active learning training method maintains both the mode *and* strong lane adherence in all predictions.

In the ten percent data volume, the model begins with a wide spread of modal coverage, but rarely conforming to any of the scene input. Rather, the model very loosely approximates a variety of kinematically-feasible spreads, regardless of map state. At fifty percent data randomly selected, the model begins to converge toward lane configurations, but notably deviates from the lane centerlines at a greater rate than the fifty percent data efficiently sampled using our active learning algorithm. Further, as shown in Figure 7.9, the active learning approach seems to show the same level of conformity even in its multi-trajectory predictions in the case of multiple possible futures.

7.6 Concluding Remarks

In this research, we present a method by which information about a vehicle's trajectory and dynamic state, collectively referred to as a trajectory-state, can be clustered. We show the utility of these clusters as the drivers of a selection criteria in an active learning framework. However, this is not the only way to cluster such data, nor is this the only possible data which can be used in the clustering process; future research can iterate on these methods to further drive development of learning systems which select data at low cost to human annotators and intelligently guide data curation at scale. While we provide a selection based on “novelty" or “uniqueness" in this research, other measures, such as salience [53–55] or even language-based queries [196], may also be highly informative to efficient and safe model learning [197]. Beyond learning itself, such novelty-mining is also important in selection of data for system validation [198].

Further, in this set of experiments, we apply the trajectory-state-informed active learning toward the task of trajectory prediction, but the utility should be further explored in additional autonomous driving tasks [30]. We make an argument at the beginning of the paper that one can infer much about the outside scene from the trajectory alone. While the outside scene is subsequently annotated and used for the trajectory prediction task we annotate, it would be interesting to see how well the trajectory informs active learning for the other relevant task of object detection, reinforcing the mutual information between the visual sensing of a scene and an agent’s response trajectory to a scene (i.e. “perception without vision").

In closing, we repeat that the proposed clustering and active learning algorithms are methods by which large-scale data systems can be more efficient without sacrificing performance on safety-critical predictive tasks. Data-driven methods show significant promise towards robust safety, but handling the long-tail nature of high-risk driving events requires intelligent approaches to collecting, curating, and annotating this valuable data.

Acknowledgements

Chapter 7, in part, in part, has been submitted for publication of the material as it may appear in Robotics and Automation Letters, 2024: Ross Greer and Mohan M. Trivedi. "Perception Without Vision for Trajectory Prediction: Ego Vehicle Dynamics as Scene Representation for Efficient Active Learning in Autonomous Driving." The dissertation author was the primary investigator and author of this paper.

Part IV

Safely Handling the Unexpected: Driver Control Transitions

For Level 3 autonomous driving, the driver may be called upon to take control of the vehicle at any time. For a safe control transition, the outside scene as well as the driver's readiness must contribute factors to control decisions (alternation between autonomous and manual modes) [3].

As presented in earlier chapters, recognizing novelty or hazards is an important part of the “looking outside" piece of these systems. Even if not detected from on-board sensors, this capability can still assist in safe control transitions; for example, detecting novelty from infrastructure can provide a basis for a signal to be broadcast to a vehicle, alerting the driver that there are upcoming conditions requiring a takeover.

Following this, the research presented in this chapter investigates methods by which the system can maintain an awareness of the driver's state, including their observable readiness [28, 199] and estimated takeover time [2]. In the appendix, I also include discussion of hand activity occupation [103] [28].

These projects, in leveraging complementary modes of observation, contain research across areas of machine learning such as ensemble learning, cross-attention mechanisms and shared data features between modalities, and temporal and spatial attention in video sequences. Further, the demands of such systems to run in real-time and with minimum sensing hardware require investigation of efficient neural architectures and ablative analysis of available information streams. The implications of these systems extend beyond autonomous driving control transitions, providing new opportunities in driver observation and control decisions related to fatigue or substance-related impairment [200].

Chapter 8

Take-over Time Prediction for Autonomous Driving in the Real-World: Robust Models, Data Augmentation, and Evaluation

8.1 Introduction

Motivations for studying driver behavior in highly automated vehicles can be found aplenty in human factors studies (e.g. [201], [202], [203], [204], [205], [206]). It is widely regarded that as soon as the level of cognitive stimulation falls below a person's own comfortable “set point", the person will seek out alternate/additional sources of information, leading to distraction (e.g. [207], [208], [209], [210]). This makes the intermediate levels of automation (as per NHTSA [211] or SAE [212]) very dangerous, causing problems such as inattention, trust, skill atrophy, complacency, etc. [213]. The authors in [213] postulate that rising levels of automation will lead to declining levels of awareness. They also state that most problems are expected to arise in systems that take the driver out of the loop, yet these are the very systems that drivers want, because they free the driver to do something else of interest. Elsewhere, the authors in [214] emphasize the *irony of automation*, whereby “the more advanced a control system is, the more crucial may be the contribution of the human operator". They also acknowledge that decades of research has shown that humans are not particularly good at tasks that require vigilance and sustained attention over long periods of time [215].

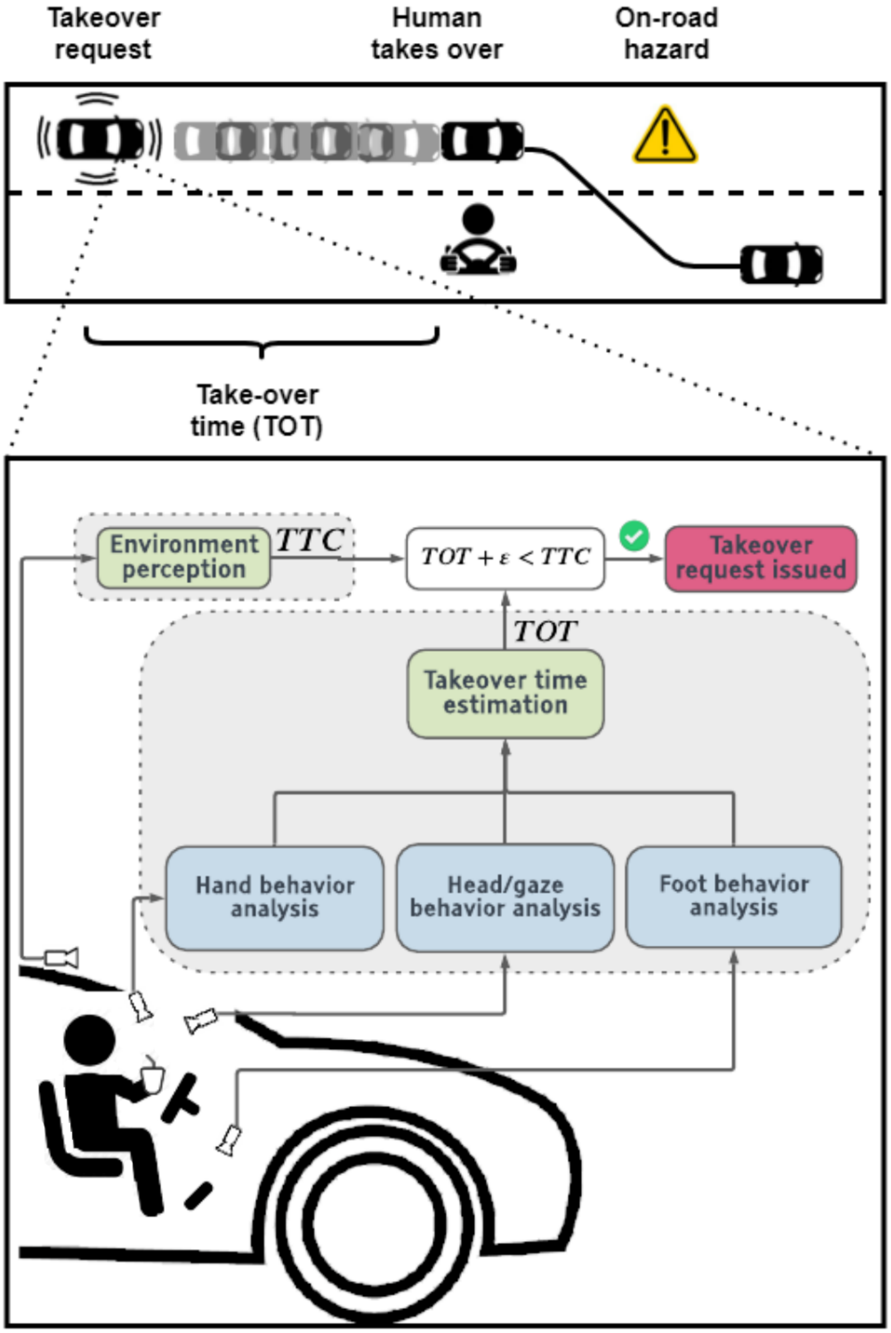


Figure 8.1. Role of take-over time (TOT) prediction: We propose a model for predicting TOT during control transitions based on driver behavior. The proposed model can be used in conjunction with time-to-collision estimation to determine whether to issue a take-over request and transfer control to the human, or to deploy active safety measures for collision avoidance.

All above points seem to suggest that a drop in attention is inherent in human behavior. Thus, it is not a matter of *if*, but *when* the driver will resort to non-ideal behavior. This makes the safe and smooth handling of control transitions, which entail the transfer of vehicle controls from the autonomous agent to the human driver and vice versa, extremely important and timely. Consider the scenario illustrated in Fig. 8.1, indicating the transition of control from an autonomous agent to the human driver to be a function of the driver state. We propose that a system that takes the state of the driver into account can decide between handing over control if the driver is ready, versus coming to a safe and smooth halt if not. Driver state can also dictate how and when a takeover alert must be supplied to ensure an uneventful transition of control.

In this paper, we focus on transitions from the autonomous agent to the human driver. In particular, we consider scenarios where limits of the autonomous system are reached. For example, an unforeseen on-road hazard may be detected that needs to be evaded. The conditions for L3 autonomy may be coming to an end with the vehicle leaving a geofenced area, or a traffic jam assist system encountering dissipation of the traffic jam. Such scenarios require timely human intervention within a predictable time window. In describing such control transitions, we make use of the take-over time (TOT) metric, defined as the interval of time between a take-over request (TOR) being issued and the assuming of human control. The take-over request could be an auditory/visual/tactile cue used to indicate to the driver that their intervention is immediately needed. Due to the complexity of human attention, we define the assumption of control as the completion of the following three behaviors:

1. **Hands-on-wheel:** hand(s) return to the vehicle's steering control.

2. **Foot-on-pedal:** foot returns (from floorboard or hovering) to make contact with any driving pedal.

3. **Eyes-on-road:** gaze is directed forward, toward the active driving scene.

We work with the assumption that these three cues occurring simultaneously are *necessary* to consider the driver both attentive to the scene and in control of the vehicle. We do note that the three cues may not be *sufficient* to consider the driver attentive and in control. This would additionally depend on factors such as the driver's situational awareness and the corrective/stabilizing maneuver performed post TOR. We limit the scope of this work to predicting the time taken for the above three cues, as a first step towards analysis of control transitions using real-world autonomous driving data. Analysis of situational awareness and corrective maneuvers will be addressed in future work.

As depicted in Fig. 8.1, the transition of control from an autonomous agent to the human driver should be a function of both the surrounding scene and the state of the driver. The surrounding scene can be concisely expressed using a metric such as time-to-collision (TTC), whereas the state of the driver can be captured by the predicted TOT. Combined, this forms a criterion for safe control transitions:

$$TOT + \varepsilon < TTC, \tag{8.1}$$

where ε is a marginal allowance that represents the time it takes for the human driver to gain situational awareness and perform a corrective maneuver. A system that takes the state of the driver into account can decide between handing over control if the driver is ready, versus coming to a safe and smooth halt if not. While there are many approaches to accurately estimate TTC, TOT prediction (especially in the real world) remains relatively unexplored. In this paper, we present a long short-term memory (LSTM) model for predicting TOT based on driver behavior prior to the TOR. We train and evaluate our model using a real world dataset of control transitions captured using a commercially available conditionally autonomous vehicle. This work is an extension of our prior work [3], with three new contributions:

1. **TOT prediction with limited real-world data:** Capturing real-world takeover events in autonomous vehicles is expensive and time-consuming. Thus generating a large enough dataset for training machine learning models can be a challenge. To address this, we propose a data-augmentation scheme to increase the number of training samples by an

order of magnitude. Additionally we use transfer learning, and pre-train our TOT prediction models to estimate the driver's observable take-over readiness index (ORI) [199].

2. **Multimodal TOT prediction:** There is inherent uncertainty in predicting the future. The driver could perform multiple plausible sequences of actions after the issued TOR. To model this, we extend the model proposed in [3] to output a multimodal distribution over TOT.

3. **Extensive evaluation:** We present a more extensive set of ablation experiments, particularly focused on the above two contributions. We also present additional qualitative analysis of TOT estimates beyond [3].

8.2 Related Research

8.2.1 Vision based driver behavior analysis

A large body of literature has addressed driver behavior analysis using in-cabin vision sensors. The most commonly addressed task is driver gaze estimation [216–227], since the driver's gaze closely relates to their attention to driving and non-driving tasks. Early works relied on head pose estimation [216, 217, 219, 228] or a combination of head and eye features [218, 220–222, 229] for estimating the driver's gaze. More recent work [223–227] uses convolutional neural networks (CNNs) to directly map regions around the driver's eyes to gaze zones. In this work, we use the CNN model proposed by Vora *et al.* [224] driver gaze analysis.

Driver hand and foot activity has also been the subject of prior work, being useful cues to gauge the driver's motor readiness. Several approaches have been proposed for detection, tracking and gesture analysis of the driver's hands [230–237] using in vehicle cameras and depth sensors. Recently proposed CNN models [238, 239] accurately localize the driver's hands in image co-ordinates and in 3-D respectively, and further classify hand-activity and held objects. We build upon the model proposed by Yuen *et al.* [238] in this work, for driver hand analysis. Relatively few works have addressed the driver's foot activity [240–242]. However, we believe this is a significant cue for TOT estimation, especially since we estimate the foot-on-pedal time after the TOR. We use the model proposed by Rangesh *et al.* [243] for driver foot activity analysis.

There has also been significant research that builds upon cues from driver gaze, hand and foot analysis for making higher level inferences such as driver activity recognition [33, 244–250], driver intent or behavior prediction [251–257] and driver distraction detection [258–263]. Of particular interest is recent work [199], where the authors map driver gaze, hand and foot activity to the driver's observable take-over readiness index (ORI) obtained via subjective ratings assigned by multiple human observers. We use ORI estimation as a transfer learning task for pre-training our TOT prediction model.

8.2.2 Take-over time analysis in autonomous driving

Take-over time in partial and conditionally autonomous vehicles has been the subject of several recent studies [264–274]. The primary focus of these studies has been to analyze the effect of various human and environmental factors on take-over time and quality. The independent variables analyzed for their effect on TOT are as follows:

TOT budget (or time to collision): This corresponds to the time window between the TOR and the imminent collision or system boundary. Gold *et al.* [264] compare TOT and take-over quality for two different TOT budgets of 5s and 7s. They report longer TOTs for the 7s budget but better take-over quality. Mok *et al.* [265] report a similar finding while comparing TOT budgets of 2s, 5s, and 8s, with the 2s case corresponding to significantly worse take-over quality and collision rates.

Traffic density: Radlmayr *et al.* [266] and Gold *et al.* [267] analyze the effect of traffic density on TOT and take-over quality, with both studies reporting longer TOTs and worse take-over quality in situations involving high traffic density.

Driver age: Korber *et al.* [268] and Clark and Feng [269] analyze the effect of driver age on

TOT by comparing a group of young drivers with a group of old drivers. Korber *et al.* [268] report similar TOTs, but different modus operandi – older drivers brake harder and more often leading to higher TTC. Clark and Feng [269] report lower TOTs for the young group for a TOT budget of 4.5s, and lower TOTs for the old group for a 7.5s TOT budget.

TOR modality: Petermeijer *et al.* [270] and Huang *et al.* [271] compare different modalities for issuing the TOR. Auditory and tactile TORs are considered in [270] while auditory, tactile and visual TORs and their combinations are considered in [271]. Both studies report the lowest TOTs for multimodal TORs. Dogan *et al.* [272] analyze the effect of providing the driver anticipatory information about the vehicle and traffic state prior to the TOR, but report similar TOTs with and without the anticipatory information.

Non-driving-related tasks (NDRTs): Several prior works [266, 272–274] have consistently reported worse take-over times or take-over quality when the driver is engaged in a NDRT prior to the take-over, whether the NDRT places visual, cognitive or motor-control based demand on the driver. In this paper, we thus primarily focus on the effect of driver behavior and NDRTs on TOT. In particular, we map the observed NDRTs to feature descriptors of driver gaze, hand and foot activity using vision based models for driver behavior analysis and predict TOT based on these feature descriptors.

8.2.3 Take-over time prediction for autonomous driving

While the studies described in the previous section analyze take-over times under various experimental conditions, closest to our work are recently proposed machine learning models [275–280] that *predict* TOT prior to the control transition.

Braunagel *et al.* [275] and Du *et al.* [278] propose binary classifiers that output whether or not the driver is ready to take-over. Gaze activity, NDRT label and a label for situation complexity are used as input features in [275], while gaze activity, heart rate variability, galvanic skin response, traffic density and TOT budget are used as inputs in [278]. Pakdamanian *et al.* [280] propose a three class classifier over TOT intervals based on driver gaze activity, heart rate variability, galvanic skin response, NDRT label and vehicle signals. Lotz and Weissenberger [276] compare various classifiers over 4 TOT intervals trained using features capturing driver's head orientation and gaze activity, along with TTC. Hwang *et al.* [279] propose a regression model based on hidden Markov models that outputs TOT based on vehicle signals prior to the TOR. Finally, Berghofer *et al.* [277] propose a regression model for TOT prediction based on driver gaze activity and driver characteristics such as age, gender, sleepiness, attitude towards highly automated driving and previous experiences with automated driving.

Our work differs from previously proposed TOT prediction models on two counts. First, we use fine-grained descriptors of driver gaze, hand and foot activity obtained purely using non-intrusive vision sensors as inputs to our TOT prediction model. Second, we train and evaluate our models using a large *real-world* dataset of take-overs captured in a conditionally autonomous vehicle. Prior work on TOT prediction has been limited to the simulator setting [275, 276, 278–280]. Berghofer *et al.* [277] do use a real world dataset. However, they use a 'Wizard of Oz' setting where a safety driver with access to vehicle controls plays the role of the autonomous vehicle.

8.3 Datasets & Labels

8.3.1 Controlled Data Study (CDS)

To capture a diverse set of real-world take-overs, we conduct a large-scale study under controlled conditions. More specifically, we enlist a representative population of 89 subjects to drive a Tesla Model S testbed mounted with three driver-facing cameras that capture the gaze, hand, and foot activity of the driver. In this controlled data study (CDS), we required each subject to drive the testbed for approximately an hour in a pre-determined section of the roadway, under controlled traffic conditions. During the drive, each test subject is asked to undertake a variety of distracting secondary activities while the autopilot is engaged, following which an auditory take-over request (TOR) is issued at random intervals. This initiates the control transition during which the driver is instructed to take control of the vehicle and resume the drive. Each such transition corresponds to one take-over event, and our CDS produces 1,375 take-over events in total.

8.3.2 Annotation

Automated video segmentation: Each driving session is first segmented into 30 second windows surrounding known take-over events, consisting of 20 seconds prior to the take-over request (TOR) and 10 seconds after the take-over event.

Event annotations: For each 30 second clip corresponding to a take-over event, we manually annotate the three times after the take-over request corresponding to when the driver's eyes are on the road, hands are on the wheel, and foot is on the pedal. We also label the secondary activity being performed by the driver during each take-over event, assigning one of 8 possible activity labels: (1) No secondary activity, (2) talking to co-passenger, (3) eyes closed, (4) texting, (5) phone call, (6) using infotainment unit, (7) counting change, (8) reading a book or magazine. The take-over events are distributed between secondary activities as shown in Table 8.1.

Figure 8.2 shows the average times corresponding to eyes on road, hands on wheel and foot on pedal for each of the 8 secondary activities. It also shows the overall take-over time, which is the maximum of the three markers for each event. We note that texting, phone-calls, counting change and reading correspond to longer average take-over times, as compared to talking to the co-passenger or using the infotainment unit, which can be reasonably expected. Counter to intuition, the 'eyes closed behind the wheel' activity has low take-over times. This is mainly because the drivers are merely 'acting' to be asleep, since actual sleep could not have been achieved given the duration and nature of each trial. We also note that the 'hands on wheel' event seems to take much longer on average, as compared to eyes on road or foot on pedal. This reinforces the need for driver hand analysis, which is also a key predictor of the driver's observable readiness index (see next section). Finally, we note that for the more distracting secondary activities (reading, texting, phone calls, counting change), even the foot on pedal times are longer compared to the other secondary activities, although the secondary activities do not involve the driver's feet. Thus, there seems to be a delay corresponding to the driver shifting attention from secondary activity to the primary activity of driving.

8.3.3 Data Augmentation

Takeover time data is very limited and expensive to capture and label. This is illustrated by the size of the CDS dataset (1,375 unique takeover events). This introduces challenges during training neural networks for TOT prediction, as these models typically require tens of thousands of training samples. Care must also be taken to avoid overfitting, as this is more prevalent in the limited data regime. To address these issues, we propose a new data augmentation scheme to increase the number of samples in the dataset by an order of magnitude.

Figure 8.3 illustrates our data augmentation scheme. We term each take-over event in the CDS dataset a *raw sample*. Each raw sample has annotated timestamps corresponding to the take-over request (t_{tor}), as well as the time taken by the driver to get their eyes on the road (t_{eyes}), hands on the wheel (t_{hands}) and foot on the pedals (t_{foot}) after the TOR as shown in Figure 8.3. We wish to learn a model that maps a 2 second window of driver activity prior to the TOR to the take-over times, $\{t_{eyes}, t_{hands}, t_{foot}\}$.

The raw samples alone are insufficient to train a machine learning model from scratch. We thus mine *augmented training samples* from each takeover event as shown in Figure 8.3. An augmented training sample is characterized by an augmented TOR at time t_{off} after the actual t_{tor}. We use a 2 second window of driver activity before the augmented TOR as the input to the model while the corresponding takeover times are given by $\{t_{eyes} - t_{off}, t_{hands} - t_{off}, t_{foot} - t_{off}\}$. If the driver's hands, eyes, or foot are already in position at t_{tor} + t_{off}, the corresponding takeover time is set to 0.

An augmented training sample maps the driver's state at an intermediate timestamp during the takeover event, to their reaction times from that timestamp. While this doesn't correspond to an actual TOR, it still serves as useful data for training our TOT prediction model as we show in Section 8.5. Intuitively, the driver can be expected to be less and less distracted by a non-driving activity as t_{off} is increased, leading to shorter takeover times. Thus the augmented samples provide additional instances where the driver is increasingly prepared to takeover control from the vehicle.

We capture data at a frame rate of 30 Hz. Thus, t_{off} can be varied from 0 to the maximum of $\{t_{eyes}, t_{hands}, t_{foot}\}$ using increments of 1/30 seconds to yield multiple augmented samples per takeover event. The augmentation scheme is only applied to the training split. The validation and test splits are left untouched for accurate evaluation.

Table 8.1. Control Transition Secondary Activity Frequency.

Secondary Activity	Number of samples	Percent
No secondary activity	308	23.0%
Texting	262	19.6%
Infotainment unit	262	19.6%
Talking to passenger	182	13.6%
Reading book or magazine	100	7.5%
Counting coins	97	7.2%
Eyes closed/Looking at lap	85	6.4%
Phone call	42	3.1%

8.4 Models & Algorithms for Predicting Takeover Times

It is important to preserve both the diverse and sequential nature of all features related to driver behavior while designing a holistic take-over time (TOT) prediction framework. High level tasks such as TOT prediction are influenced by low level driver behaviors, both in the short and medium to long term. Figure 8.4 provides an overview of our proposed approach for estimating TOT. Our approach consists of two major components. The first component is a set of convolutional neural networks (CNNs) for extracting frame-wise descriptors of driver gaze, hand and foot activity from the raw camera feed. We describe these is greater detail in section 8.4.1. The second component is an LSTM model for estimating TOT based on a sequence of frame-wise features over a pre-defined time window. We describe the different variants of our

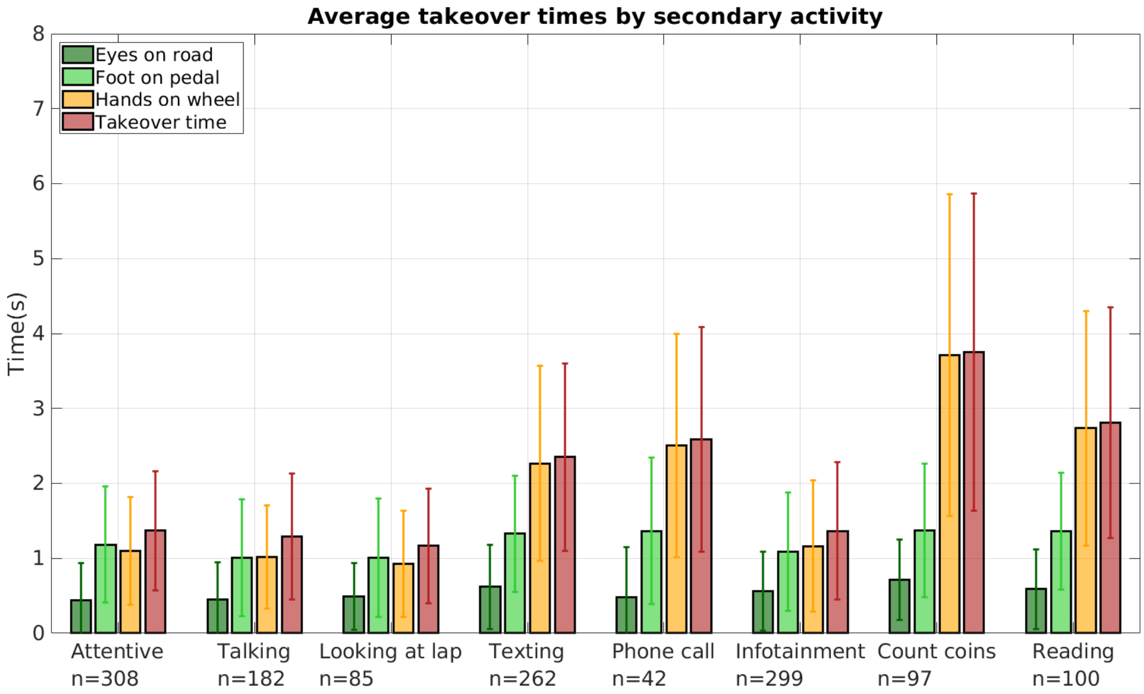


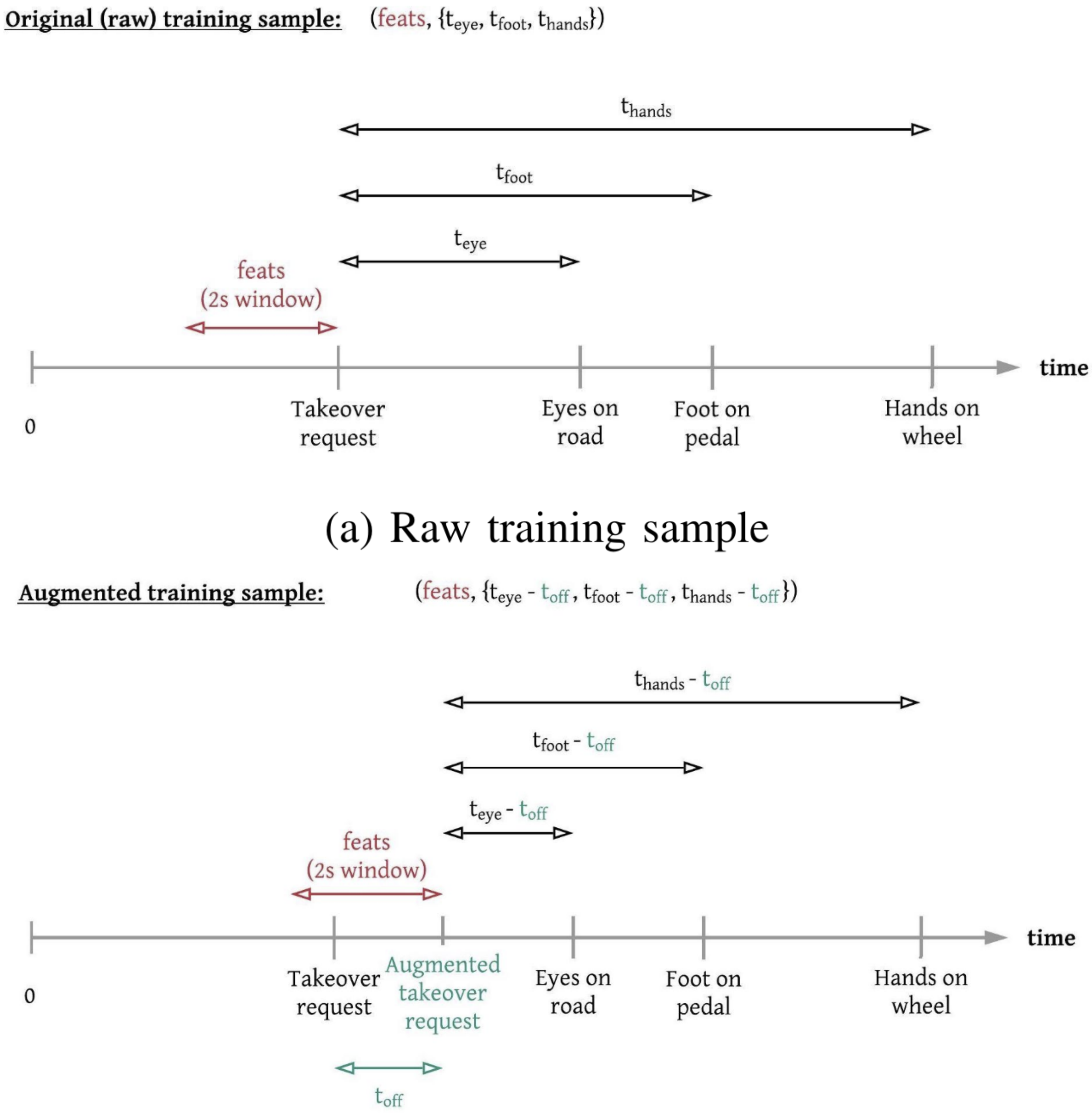
Figure 8.2. Take-over time statistics from the CDS: We plot the mean values (with error bars) of the different take-over related event timings for each secondary activity.

LSTM based models in section 8.4.2.

8.4.1 Frame-wise feature extraction

Gaze activity: We use the model proposed by Vora *et al*. [223] for driver gaze analysis. The inputs to the model are frames from the face camera. We use a face detector [281] for localizing the driver's eyes. A cropped bounding box around the driver's eyes is passed through a CNN, which outputs the driver's gaze zone. We consider 8 gaze zones: {forward, left mirror, lap, speedometer, infotainment unit, rear-view mirror, right mirror, over the shoulder}. The CNN outputs frame-wise probabilities for each gaze zone. We use this 8 dimensional vector to represent driver's gaze features.

Hand activity: We use the model proposed by Yuen and Trivedi [282] for driver hand analysis. The model localizes the elbow and wrist joints of the driver using part affinity fields [283]. A cropped bounding box around the driver's wrist is passed through a CNN to output probabilities



(a) Raw training sample

(b) Augmented training sample

Figure 8.3. Illustration of TOT dataset augmentation scheme.

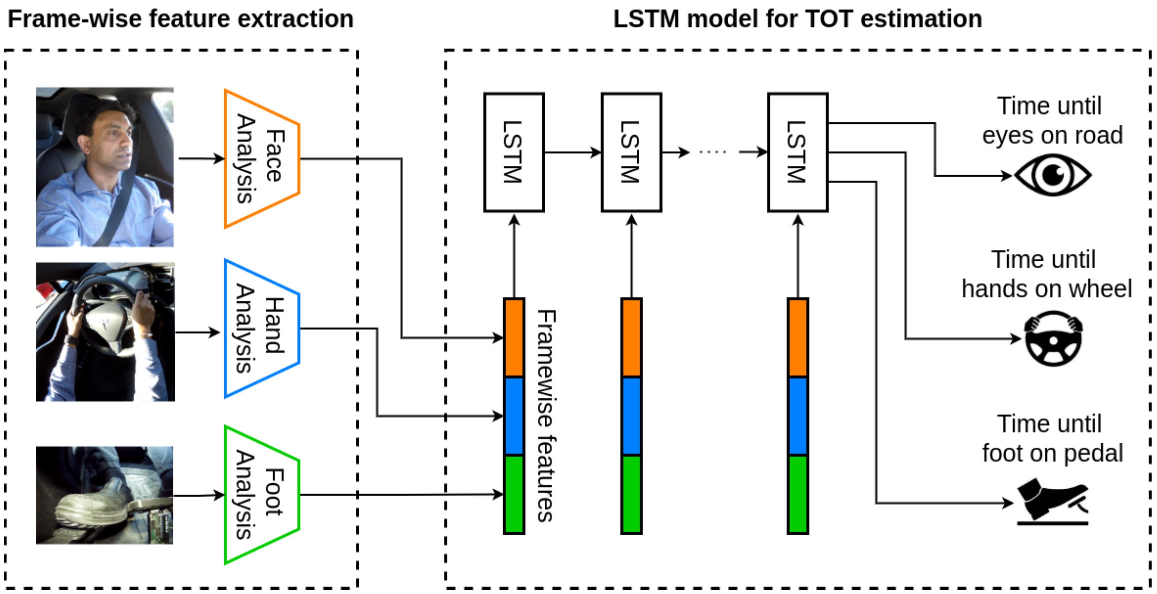


Figure 8.4. Overview of the proposed approach: We extract frame-wise descriptors of driver gaze, hand and foot activity. We propose an LSTM model for predicting TOT based on a sequence of the extracted features over a 2 second window.

corresponding to 6 hand activities for each hand: {on lap, in air, hovering over steering wheel, on steering wheel, on cupholder, interacting with infotainment unit}. We extend the model to additionally output hand-held object probabilities. We consider 7 object categories: {no-object, phone, tablet, food, beverage, book, other}. By running the models on images from a stereo camera pair, we also obtain 3-d coordinates for the driver's wrist locations and the steering wheel using triangulation. We then calculate the distance of each hand (wrist) of the driver to the steering wheel in 3-d. The hand activity probabilities, hand object probabilities and 3-d distance to steering wheel together form the hand activity features for each frame.

Foot activity: We use the model proposed by Rangesh and Trivedi [243] for driver foot analysis. Each frame from the foot camera feed is passed through a CNN to output probabilities over 5 foot activity classes: {away from pedal, on brake, on gas, hovering over brake, hovering over gas}. These probabilities represent the foot activity features for each frame.

8.4.2 LSTM models for take-over time prediction

Baseline LSTM: This is the simplest (baseline) version of all TOT models. The input features are first transformed using a fully-connected (FC) layer of size 16 (plus non-linearity), which is then fed to an LSTM with a hidden state of size 32 at each timestep. The LSTM layer receives the transformed input features at each timestep and updates its internal representation known as the hidden state. In all our experiments, we choose a 2 second window of features as input to our models. After 2 seconds worth of inputs and updates, the hidden state of the LSTM after the latest timestep is passed through an output transformation (FC layer plus non-linearity) to predict the three times of interest.

We apply a simple $L1$ loss to train this network. Let o_e, o_f, and o_h be the outputs produced by the model. Assuming t_e, t_f, and t_h are the target eyes on road time, foot on pedal time, and hands on wheel time respectively, the total loss is:

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}|t_e^i - o_e^i| + \frac{1}{N}\sum_{i=1}^{N}|t_f^i - o_f^i| + \frac{1}{N}\sum_{i=1}^{N}|t_h^i - o_h^i|. \tag{8.2}$$

The entire model is trained using an Adam optimizer with a learning rate of 0.001 for 10 epochs.

Independent LSTMs: Figure 8.5 shows the independent LSTM model architecture. This model is the same as the baseline LSTM model, except for one major difference: each target output time has its own independent LSTM. The reasoning behind this is to accommodate different hidden state update rates for different driver behaviors, for example – eyes on road behavior is generally faster (short term) than hands on wheel behavior (mid/long term). Having multiple independent LSTMs allows each one to update at different rates, thereby capturing short/mid/long term behaviours separately.

Although each branch has its own LSTM cell, the input and output transformations are still shared between the three LSTMs as the feature inputs to the three branches are the same.

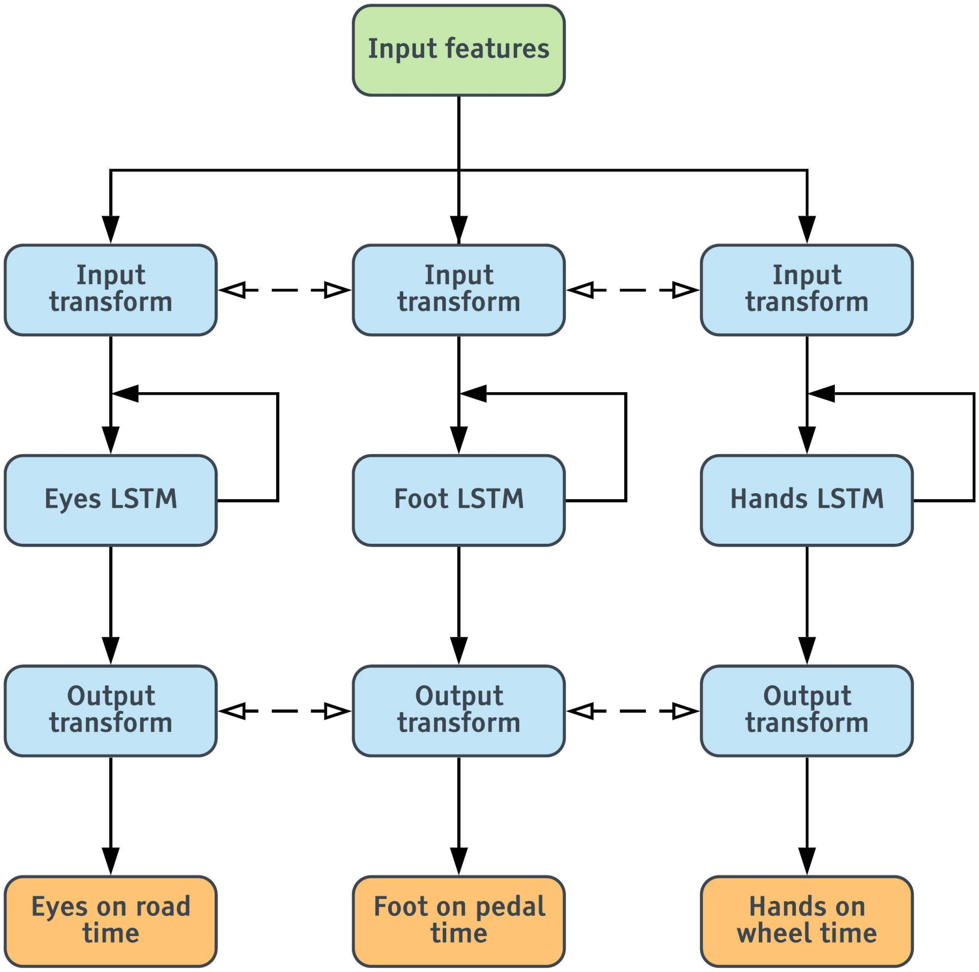


Figure 8.5. Independent LSTMs model architecture.

This tends to reduce overfitting based on our experiments.

We use the identical loss (eq. 8.2) and optimizer settings as the baseline LSTM for training the independent LSTMs model.

LSTM with Multi-modal Outputs: This model is largely based on the baseline LSTM with one addition: *multi-modal outputs*. Instead of just producing one output for each of the three targets, we output $K(=3)$ outputs per target and their associated probabilities. We do this to model the inherent multi-modality and subjectiveness of takeover times. For example, given similar history of behavior, one driver may respond faster in taking control of the vehicle than another. Producing multiple probable outputs (and their probabilities) could possibly address this ambiguity and provide more usable information to any downstream controller.

Unlike the previous models, this model is trained using a minimum of K loss, where $L1$ losses are only applied to the output modes closest to the ground truth target. Additionally, the output probabilities are refined using cross-entropy. Let $o_e(k)$, $o_f(k)$, $o_h(k)$ and $q(k)$ denote the k^{th} set of outputs and corresponding probability produced by the model. Assuming t_e, t_f, and t_h are the target eyes on road time, foot on pedal time, and hands on wheel time respectively, the total loss is:

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N} \min_k \left(|t_e^i - o_e^i(k)| + |t_f^i - o_f^i(k)| + |t_h^i - o_h^i(k)|\right) \\ - \lambda \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} p^i(k)\log(q^i(k)), \quad (8.3)$$

where $p^i(k)$ is a one-hot categorical probability distribution given by the indicator function,

$$p^i(k) = \mathbb{1}\Bigg(\arg\min_l \left(|t_e^i - o_e^i(l)| + |t_f^i - o_f^i(l)| + |t_h^i - o_h^i(l)|\right) = k\Bigg), \quad (8.4)$$

and λ is a coefficient used for relatively weighting the L1 and cross-entropy losses.

Table 8.2. Estimation errors for different times of interest on the CDS validation set.

Model type (s)	Overall MAE (s)	Eyes on road MAE (s)	Foot on pedal MAE (s)	Hands on wheel MAE (s)	Takeover time MAE (s)
Constant prediction (Max over train set stats)	3.9271	2.4540	2.9880	6.3392	6.1969
LSTM[1]	0.5104	0.3353	0.5029	0.7126	0.8098
ID LSTMs[2]	**0.5073**	**0.3266**	**0.4841**	**0.7113**	**0.7912**
LSTM + MM[3]	0.5589	0.3582	0.5262	0.7921	0.8908
ID LSTMs + MM	0.5319	0.3415	0.5019	0.7524	0.8441
LSTM + MM (best of K)	0.3921	0.2393	0.4204	0.5167	0.6265
ID LSTMs + MM (best of K)	0.3911	0.2344	0.3875	0.5513	0.6586

[1] baseline LSTM model [2] Independent LSTMs [3] Multi-modal outputs (with $K = 3$ modes)

As before, the entire model is trained using an Adam optimizer with a learning rate of 0.001 for 10 epochs. We use $\lambda = 1$ for simplicity.

Independent LSTMs with Multi-modal Outputs: The final proposed model uses a combination of independent LSTMs and multi-modal outputs described before. One difference to the original independent LSTMs model is that we now concatenate the hidden states of all three LSTMs and transform them together to produce the target outputs. This is done because probabilities are assigned to the joint of all three target times, and thus need to be operated on together.

We use the identical loss (equation 8.3) and optimizer settings as the LSTM with multi-modal outputs for training the independent LSTMs with multi-modal outputs.

8.5 Experiments & Evaluation

Comparison of LSTM models for TOT prediction: First, we conduct an experiment to assess the effects of different model architectures. All proposed models (from Section 8.4.2) were trained on CDS train set with augmented data, and then evaluated on the validation set. We use individual and overall mean absolute errors (MAEs) as metrics for comparison. Table 8.2 contains results from this experiment. In addition to the LSTM models, as a sanity check, we include a simple baseline that always predicts a constant value for all take-over time markers, corresponding to the maximum value for each marker from the train set.

From these results, we note that all LSTM models considerably outperform the constant

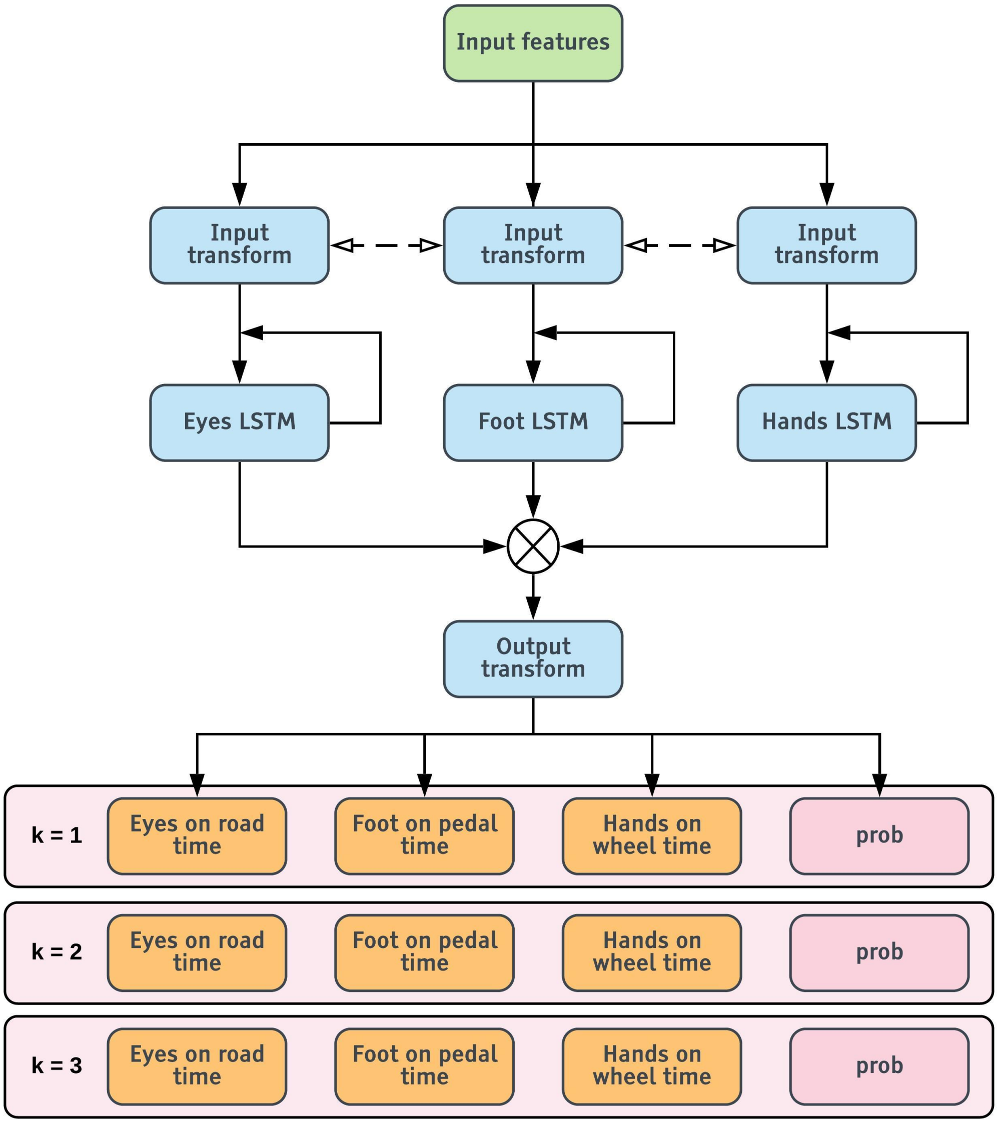


Figure 8.6. Independent LSTMs with multi-modal outputs model architecture.

Table 8.3. Estimation errors for different times of interest on the CDS validation set when trained on a variety of datasets.

Training dataset (s)	Overall MAE (s)	Eyes on road MAE (s)	Foot on pedal MAE (s)	Hands on wheel MAE (s)	Takeover time MAE (s)
CDS (R)[1]	0.5799	0.3676	0.5435	0.8285	0.8576
CDS (A)[2]	**0.5073**	0.3266	0.4841	0.7113	0.7912
ORI[3]→ CDS (A)	0.5184	**0.3246**	0.5182	**0.7054**	**0.7729**

[1] raw dataset [2] augmented dataset [3] ORI estimation dataset

value baseline showing that there is a learnable signal in the data and the usefulness of using a machine learning model. We observe that the independent LSTMs model consistently outperforms other models. At first glance, the multi-modal models tend to perform worse than the ones without multi-modal outputs. To further analyze the source of these errors, we provide the *best-of-K* MAEs for these models in Table 8.2. The *best-of-K* MAEs simply mean that instead of choosing the most probable set of predictions for error calculation, we use the set that produces the least error i.e. assume perfect classification. The *best-of-K* numbers are vastly superior to the ones without multi-modal outputs. This indicates that in most cases, at least one of $K(=3)$ sets of predictions is highly accurate. However, accurate probability assignment for these K modes (i.e. classification) remains error-prone. Nevertheless, we believe that having multiple probable outputs instead of one less accurate one could be beneficial for downstream controllers.

Effect of data augmentation and transfer learning: Next, we conduct experiments to assess the effects of our data augmentation and transfer learning schemes. To isolate these effects, we use the same ID LSTMs model for all experiments. We compare the following training schemes:

- **CDS (R):** First, as a baseline, we train a model purely using the raw CDS data without augmentation.
- **CDS (A):** Next, we train a model using the augmented training dataset using the augmentation scheme described in section 8.3.3. The raw dataset contained 1,375 samples, which we augment to 47,461 datapoints.

Table 8.4. Estimation errors for different times of interest on the CDS validation set for a variety of feature combinations.

Features					Overall	Eyes on road	Foot on pedal	Hands on wheel	Takeover time
Foot	Gaze	Hands Activities	Hands Distances	Held Objects	MAE (s)	MAE (s)	MAE (s)	MAE (s)	MAE (s)
✓					0.5735	0.3587	0.5018	0.8599	0.8856
	✓				0.5811	0.3332	0.5690	0.8411	0.8837
		✓			0.5560	0.3729	0.5384	0.7565	0.9012
		✓	✓		0.5420	0.3783	0.5109	0.7369	0.8315
		✓		✓	0.5217	0.3702	0.4973	0.7177	0.8621
		✓	✓	✓	0.5182	0.3747	0.4857	0.7141	0.7983
	✓	✓		✓	0.5202	0.3244	0.5220	0.7163	0.7920
	✓	✓	✓	✓	0.5213	0.3299	0.5124	0.7215	0.7921
✓	✓	✓	✓		0.5384	**0.3222**	0.5059	0.7870	0.8475
✓	✓	✓		✓	0.5088	0.3277	0.5074	0.7144	0.7918
✓	✓	✓	✓	✓	**0.5073**	0.3266	**0.4841**	**0.7113**	**0.7912**

- **ORI → CDS (A):** Finally, we consider a model pre-trained to estimate the observable take-over readiness index (ORI) proposed in [199]. The ground truth ORI values are obtained via subjective ratings assigned by multiple human observers rating how ready a driver is to take-over control from the vehicle based on the past two seconds of video feed from the driver facing cameras. The ratings are normalized and averaged to account for rater bias as described in [199].

Results from these experiments are presented in Table 8.3. As before, we use individual and overall mean absolute errors (MAEs) as metrics for comparison.

From Table 8.3, we notice that training on the augmented dataset (as proposed in Section 8.3.3) consistently and considerably improves performance as compared to the raw dataset. We believe that doing so prevents overfitting, provides regularization, smooths the outputs of model, and adds new training samples that would be cumbersome or impossible to capture.

Finally, we observe that training the model for observable readiness index (ORI) estimation [199], followed by transfer learning on TOT prediction improves some metrics. This highlights the commonality between the two tasks - features from learning one task can improve performance in the other.

Effect of hand, gaze and foot activity features: Finally, we conduct an experiment to assess the relative importance of different input features and their combinations. To isolate effects from features, we train the same ID LSTMs model with different input feature combinations. We

Table 8.5. Estimation errors for different models on the takeover time test set.

Model type (s)	Overall MAE (s)	Eyes on road MAE (s)	Foot on pedal MAE (s)	Hands on wheel MAE (s)	Takeover time MAE (s)
Constant prediction (Max over train set stats)	4.0835	2.6790	3.1540	6.4175	6.2073
LSTM[1]	0.5242	**0.2365**	0.5007	0.8710	0.9457
ID LSTMs[2]	**0.5208**	0.2497	**0.4650**	**0.8055**	**0.9144**
LSTM + MM[3]	0.5339	0.2635	0.5265	0.8117	0.9307
ID LSTMs + MM	0.5526	0.2665	0.5180	0.8734	0.9418
ID LSTMs (75%[4])	0.5348	0.2557	0.5013	0.8474	0.9779
ID LSTMs (90%[5])	0.5282	0.2514	0.4851	0.8482	0.9424

[1] baseline LSTM model [2] Independent LSTMs [3] Multi-modal outputs (with $K = 3$ modes)
[4] 75% of the dataset used for training [5] 90% of the dataset used for training

use individual and overall mean absolute errors (MAEs) as metrics for comparison. Table 8.4 contains results from this experiment.

We notice that hand features are the most important, followed by foot and gaze features respectively. This might be because gaze dynamics are relatively predictable during takeovers as the first thing drivers tend to do is look at the road to assess the situation, leading to less variance in eyes-on-road behavior. Next, we notice that adding more informative hand feature like 3D distances to the steering wheel and hand-object information improves the performance further. Hand-objects in particular seem to vastly improve the performance in general. This makes sense as hand-objects are the strongest cue related the secondary activities of drivers. Adding stereo hand features improves the results, but not by much. Adding foot features also tends to reduce the errors considerably, illustrating the importance of having a foot camera.

In conclusion, one could get close to peak performance by utilizing 3 cameras - 1 foot, 1 hand, and 1 face camera respectively. Hand features are most informative, followed by foot and gaze features respectively.

Quantitative results on test set: In this section, we present quantitative error metrics on the held out test set, separate from the validation set, for all proposed models in Table 8.5. As before, we see that ID LSTMs is the best performing model. We also notice that hands-on-wheel MAEs are usually the largest owing to large variance in hand behaviors, and large absolute values associated with hands-on-wheel time.

We also show results for ID LSTMs when trained on 75% and 90% of available training data. This helps us gauge the expected improvement in performance as more training data is added. Based on the numbers presented in Table 8.5, we can expect meager improvements as more data is added. This indicates a case of diminishing returns.

8.6 Concluding Remarks

This paper presented one of the largest real-world studies on takeover time prediction and control transitions in general. We introduced a dataset of take-over events captured via controlled driving studies in a commercially available partially autonomous vehicle, with a large pool of test subjects performing a variety of secondary activities prior to the control transition. We proposed a machine learning model for take-over time prediction based on driver gaze, hand and foot activity prior to the issue of take-over requests. We also proposed a data augmentation and transfer learning scheme for best utilizing the limited number of take-over events in our dataset. Our experiments show that our model can reliably predict takeover times for various secondary activities being performed by the drivers. In particular, we showed the usefulness of analyzing driver hand, foot and gaze activity prior to issuing the take-over request. We also showed the utility of our transfer learning and data augmentation schemes for best utilizing limited training data with control transitions. We believe that this study outlines the sensors, datasets, methods and models that can benefit the intermediate stages of automation by accurately assessing driver behavior, and predicting takeover times - both of which can be used to smoothly transfer control between human and automation.

Acknowledgements

Chapter 8, in part, is a reprint of the material as it appears in Rangesh, Akshay, Nachiket Deo, Ross Greer, Pujitha Gunaratne, and Mohan M. Trivedi. "Autonomous Vehicles that Alert Humans to Take-Over Controls: Modeling with Real-World Data." arXiv preprint arXiv:2104.11489

(2022). The dissertation author was the primary investigator and author of this paper.

Part V

Salience: Towards Safe Planning in Complex Scenes

Previous research in the field of visual attention and machine learning has considered where human drivers may look when driving a vehicle, showing that this behavior can be modeled [284], and that this so-called visual salience behavior can be used to estimate a driver's scene awareness [285] and predict driving maneuvers [253].

Presented in this area of my dissertation, I query objects for a related property which I refer to as *road object salience* [54]; rather than considering an object salient by its tendency to attract the visual attention of a human driver, I instead consider an object to be salient by its importance for safety-critical decision-making by the ego agent. The nuance in this framing is that even unnoticed objects to a human driver may be very impactful toward algorithmic planning.

In the first chapter of this part, I explore whether this salience property can be learned using image features, positional encodings, maneuver information, and scene properties [54], motivated by the impact this property may have towards machine learning models that affect driving decisions. To this point, in the following chapters, I show that awareness of salience can be used to define metrics and loss functions which both enhance performance of object detectors and also provide a more sensitive assessment of a model's performance in consideration of critical scene elements [286] [55].

Another reason to explore salience is the possibility that determining an object's salience toward control decisions will allow for more informed trajectory prediction and association between scene elements in the interest of forming relationship graphs between agents and infrastructure. Detecting objects in the scene surrounding an intelligent vehicle is well-known as a primary task in vehicle perception. Benchmark datasets have tasked AI practitioners to accurately detect every traffic sign and traffic light visible to a vehicle's outside-facing camera, while datasets such as the Waymo Open Dataset and NuScenes have provided further challenge in locating objects not only within the camera frame (2D object detection), but placing them within the real driving world (3D object detection). Though there are many research works which compete for top performance on 2D object detection benchmarks, my research addresses the question, “to what end?". Detecting objects is crucial for autonomous safety, but simply detecting the objects does not make the vehicle inherently safer. It is only when the detected object becomes used as a feature towards downstream tasks, particularly in the category of *planning*, much beyond obstacle avoidance. In relation to trajectory prediction, trajectory planning [287] is the task of identifying driving waypoints toward a larger-scale driving target, as opposed to observing and predicting the path that some external (and non-controllable) agent may be taking. While a kinematically-informed model may have some ability to produce output for either task, it is clear to any driver that some understanding of scene objects also plays a vital role in our choice of driving path. Navigating safely through a scene with no regard for signage or traffic lights, treating planning like some Frogger-like collision avoidance game, would satisfy current canon metrics for “safety", namely staying on the drivable area and avoiding collisions with any agents (vehicle and VRU). However, this disturbs our notions of trustworthiness, explainability, and comfort of the AI planning system, and these principles are of high importance for users, automakers, and regulators – necessary stakeholders in bringing autonomy to real roads. This is the same challenge which burdens end-to-end driving systems, which are missing the degree of human abstraction that makes their plans explicitly relatable to their human occupants; that is, it is important for the vehicle not to behave only safely, but also in a way that it is *expected* to behave. *Salience* bridges the gap between perception and planning; once an object is detected, its salience (here defined as relevance or importance to decision-making) can be taken into account in determining whether the object’s meaning should be factored into the vehicle’s plans. Further, salience may act as a means of association. Beyond detecting objects such as vehicles, lanes, and lights, it is important that a planner understands the relationships between these objects (e.g. Car A occupies Lane 2, which is being dictated by Lights X and Y) [288]. If a model emulating a human driver’s perception can identify a lane and light as salient to their vehicle, this may be used to create this association, providing a graph representation which connects scene components in a way that support decision processes.

Chapter 9

On Salience-Sensitive Sign Classification in Autonomous Vehicle Path Planning: Experimental Explorations with a Novel Dataset

9.1 Introduction

Autonomous vehicles need to share the road with multiple decision making agents, each with their own goals and different directions of motion. Traffic signs play an important role in regulating the motion of all agents on the road. They are easy to notice and provide safety critical information in an intuitive and succinct manner to drivers and pedestrians. Traffic sign detection and recognition has thus received significant attention in recent research on autonomous driving and advanced driver assistance systems (ADAS). Several datasets have been released, with bounding boxes ([289] [290] [291] [292] [293] [294] [295] [296]), pixel level masks ([292]), as well as fine-grained category labels for traffic signs ([289] [290] [291] [292] [295] [296]). These in turn have allowed researchers to leverage modern CNN based detectors and classifiers for traffic sign detection and recognition ([297] [298] [299]).

While detection and recognition of traffic signs are important tasks, they aren't sufficient to inform an autonomous vehicle how to operate in a traffic scene. Crucially, an autonomous vehicle needs the ability to determine whether a traffic sign is *salient* or applicable to its planned path. This is a challenging task due to several factors:

- **Scene complexity**: City streets are complex environments. Consider the montage shown in Figure 9.1; in addition to being a visual maze on its own (between lane flows, non-perpendicular intersections, and train tracks) the scene contains excessive sign information, where the controller must know which signs are meant to inform its own lane and not follow the signs intended for others.

- **Asymmetric importance of scene elements**: While there is information available in every pixel visible to a vehicle, autonomous or manually-driven, certain portions of a given scene are more important in path planning. As a motivating example, being aware of a speed limit sign directed at cross traffic certainly informs a driver of expected behavior of other vehicles in the scene, but is less relevant to the driver's plans than an imminent stop sign, as illustrated in Figure 9.2.

- **Extraneous traffic signs**: In other cases, the sign which is easiest to detect may not necessarily be instructive for the ego vehicle. Consider Figures 9.3 and 9.4, where the closest sign on the right, while in a typically informative location, actually provides information to a different lane than the ego vehicle, and following such instructions may prove dangerous and unexpected to surrounding drivers.

- **Non-local context cues**: The context cues to determine whether a traffic sign applies to the ego-vehicle can often be non-local to the traffic sign itself. These non-local cues could ego-vehicle's lane, its planned route, and in some cases (such as yield signs) even the locations of surrounding agents.

This paper represents a first step towards traffic sign salience recognition. We define a sign to be *salient* if the visible sign provides an instruction intended for the ego vehicle location before the next decision point, independent of the actions of other agents and instructions provided to other lanes. To facilitate further research on traffic sign salience recognition, we

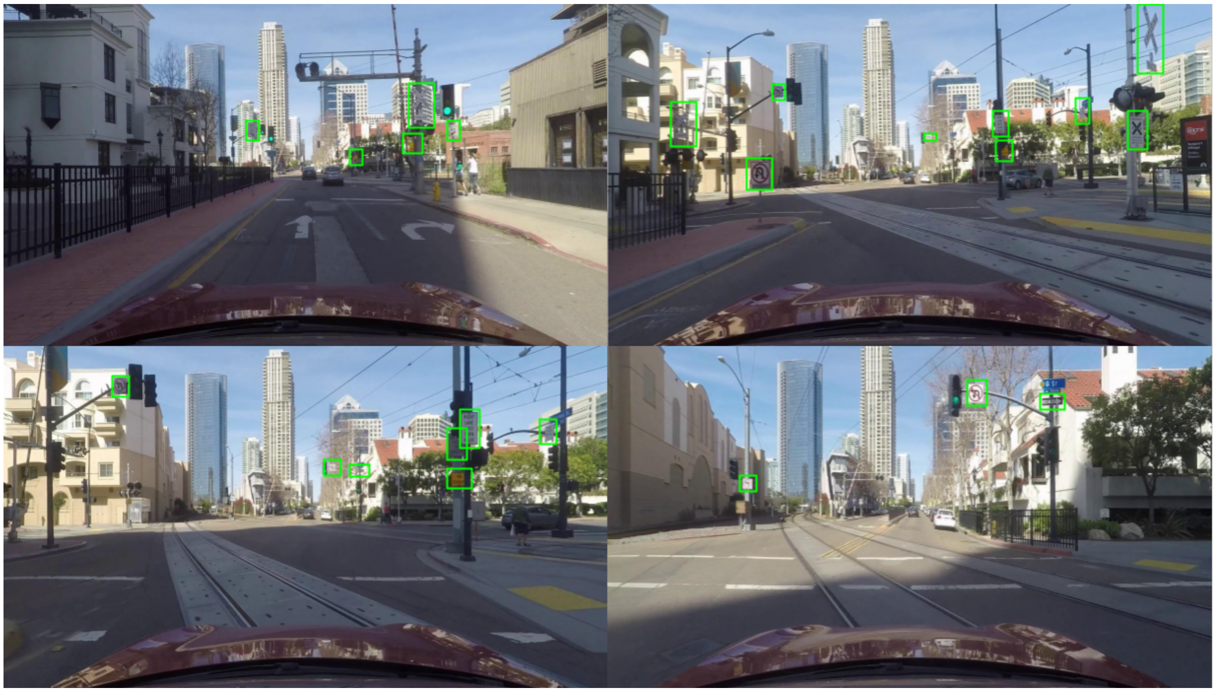


Figure 9.1. In this montage, a vehicle drives through a complex scene containing a heavy amount of signs of varying salience to the intended path. There are signs in the field of view which instruct the lanes to the left and right of the ego lane, as well as the cross-traffic at the intersection. While informative about the possible paths of other agents, these signs do not provide direction to the vehicle in proposing its own path through the intersection.

present the LAVA traffic sign dataset with traffic sign bounding boxes, fine-grained traffic sign category labels, as well as binary labels indicating traffic sign salience. Additionally we provide auxiliary meta-data for each scene including roadway type, and the next planned maneuver for the ego-vehicle. Finally, we present analysis on traffic sign salience recognition using a CNN based classifier that takes into account the appearance of traffic signs, their locations in a given scene, the roadway type (e.g. highway, intersection, on-ramp, school-zone etc.), and the planned route of the ego-vehicle.

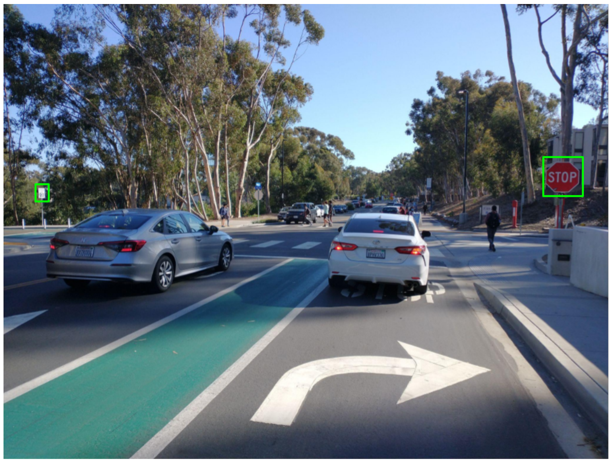


Figure 9.2. In a model designed to detect and classify signs for safe autonomous driving, being aware of the stop sign is much more important than the cross-traffic speed limit. A model should be able to weigh missed detections accordingly, as made possible with the sign salience property. By the proposed definition, the stop sign is salient while the speed limit sign is not.

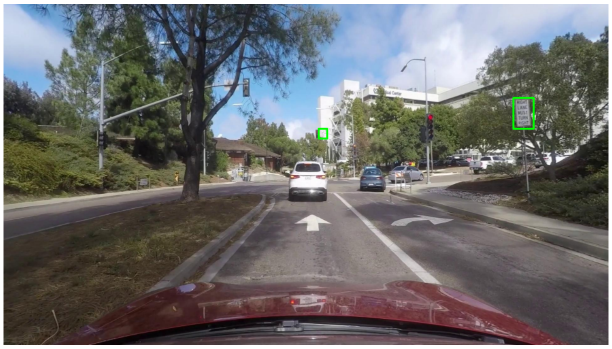


Figure 9.3. In this scenario, two possible sign detections are made, but while the detection on the right is easier, it provides no value to the ego vehicle in understanding allowable maneuvers in the upcoming intersection. Detecting this sign is less critical, but existing traffic sign datasets do not contain features with this information. Additionally, were the autonomous vehicle to mistakenly associate this “Must Turn Right" sign as salient to its lane, it would make an illegal maneuver by following its instruction. Only the white regulatory sign located across the intersection is salient.

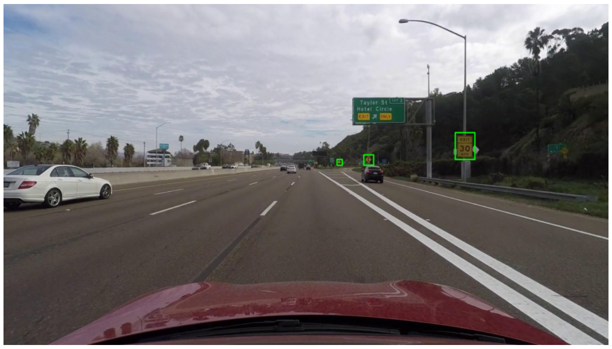


Figure 9.4. While most speed limit signs apply to all lanes in the direction of travel, this exit speed limit applies only to the lane to the right of the ego vehicle. An autonomous vehicle must have the ability to determine whether this sign is salient to its lane, and select its speed accordingly. The detected signs in this scene are not salient.

9.2 Related Research

9.2.1 Sign Detection

The task of monocular sign detection is well-established and well-addressed in the field, with prime evidence in the nearly-perfect precision-recall curves associated with the German Traffic Sign Detection Benchmark public results [289]. Recent approaches of significant performance across multiple datasets include:

- A cascaded R-CNN with multiscale attention [297], with data augmentation to balance class prevalence of commonly-missed small signs. This method is designed to address false detections due to illumination variation and bad weather.
- A sparse R-CNN with residual connections in the ResNest backbone and a self-attention mechanism [298], designed to be robust to foggy, frosty, and snowy images.

In our work, we assume prior knowledge of the detected sign location, as would be given using any of the above methods.

Table 9.1. Comparison of traffic sign datasets. All datasets contain at least images, class labels, and information about the image location and size of the bounding box for the region of interest, in addition to any unique features listed. The LAVA dataset (bottom row) is notable for its inclusion of 10 second video context, and importantly is the only dataset which contains a binary label indicating sign salience.

Dataset	Number of Images	Country	Features
German Traffic Sign Detection Benchmark [289]	50,000+	Germany	
Mapillary [290]	100,000	World	
LISA Traffic Sign Dataset [291]	7,855	USA	occlusion, on-side-road
Tsinghua-Tencent 100K [292]	100,000	China	pixel-level mask
CURE-TSD [293]	1.7M	Belgium	
LiU Traffic Signs Dataset [294]	3,488	Sweden	occlusion
Chinese Traffic Sign Database [295]	6,164	China	
Russian Traffic Sign Images Dataset [296]	104,358	Russia	
LISA Amazon-MLSL Vehicle Attributes Dataset [75]	14,112	USA	10s video context, occlusion, **salience**

9.2.2 Sign Classification

While sign classification in itself is not necessary in our model of sign importance, the problem has been addressed to high levels of accuracy in [299], the top performer on the German Traffic Sign Recognition Benchmark, using a CNN with three spatial transformers. Our work predicts sign salience using standard convolutional filter features extracted from the cropped sign image, but ongoing SOA approaches to sign recognition can provide improved backbone features to salience classification, since a sign's appearance can certainly affect its relevance (e.g. a stop sign detected while the ego vehicle is on a freeway is likely meant for off-freeway traffic and is therefore irrelevant). Further, sign classification can be combined with sign salience for downstream control, such that the control module can understand first which signs are important, and second what expected behaviors those important signs are indicating.

9.2.3 Traffic Sign Datasets

The traffic sign datasets listed in Table 9.1 facilitate research in the above tasks of traffic sign detection and classification. In this table, we highlight the relative size of these datasets, as well as any unique annotated features beyond the traffic sign class and bounding box image coordinates.

9.2.4 Traffic Signs in Planning and Control

Most current consumer-market vehicles offer little path planning for traffic regulations apart from maintaining lane, maintaining reasonable speed, and avoiding collision, hence advisory restricting the use of these features to freeways-only. Autonomous vehicles are expected to come with minimum safety guarantees, but the verification and explainability of such systems is difficult to address with common end-to-end learning approaches. Integration of algorithms which address traffic regulations can provide the deterministic and explainable action qualities important to public acceptance and safety. As explained by Fulton et al. [300], “Autonomous systems that rely on formally constrained RL for safety must correctly map from sensory inputs into the state space in which safety specifications are stated. I.e., the system must correctly couple visual inputs to symbolic states." A recent approach to address this explainability, Cultrera et al. [301] use end-to-end visual attention model which allows identification of what parts of the image the model has deemed most important. The specific importance of regulatory scene understanding has proven useful in trajectory prediction by Greer et al. [64], using a weighted lane-heading loss to ascribe importance to lane-following. Learned attention to the static scene has been demonstrated effective by Messaoud et al. [302], showing that end-to-end approaches to trajectory prediction which take in only agent motion are missing valuable information from the regulations of the static scene.

As an example of a recent model which acknowledges the importance of sign-adherence capabilities, [303] use an end-to-end learning approach to control using navigational commands, but note a shortcoming: “Traffic rules such as traffic lights, and stop signs are ignored in the dataset, therefore, our trained model will not be able to follow traffic lights or stop at stop signs." Some regulations can be addressed by algorithms (rule-based or learned) which are tailored to specific signs. Alves et al. [304] explore planning under traffic sign regulations by modelling and implementing three Road Junction rules involving UK stop and give-way signs.

While this work is the first to ascribe and predict sign salience, Guo et al. [305] create a dataset and learning architecture to promote descriptive understanding of signs beyond common detection and recognition. Their model is intended to output a semantic, verbal description which connects the texts and symbols on a sign. In contrast, our method does not seek to understand the semantic meaning of the sign, but rather whether the sign is important to the attention of the vehicle. These features are clearly informative to one another, but while their output is intended to influence navigational decisions, our output is better fit for loss-weighting schemes in safety-critical detections and recognitions.

9.2.5 Road Object Salience

Identifying salient objects has been explored in connection with driver behavior analysis; knowing where a driver is looking can be a predictor of scene salience, but alternatively, knowing salient objects prior to observing gaze can inform an intelligent vehicle of possible gaps in the driver's attention. Dua et al. [306] create the DGAZE dataset to map driver gaze in scene images, connecting gaze to driver focus and attention, topics extensively studied in connection to safe, highly-automated driving in the recent survey by Kotseruba and Tsotsos [307]. Lateef et al. [308] use a GAN to predict important objects in driving scenes, with data from existing driving datasets labeled using a salience mechanism which weights object classes from the semantic segmentation of the scene, building a Visual Attention Driving Database. Su et al. [309] show that salience is a property which can transfer from non-driving-related tasks to driving tasks, learning attention on CityScapes from standard salient object detection (SOD) datasets. Pal et al. [310] show that combining static scene information with driver gaze information in their SAGE-Net can propose important regions of attention.

Li et al. [311] define the task of risk object identification, under the hypothesis that objects influencing drivers' behavior are risky. Though their work is intended to cover a more general scope of objects than traffic signs, signs that we determine to be salient do hold a similar property; that is, were the sign not present, it is possible that the driver's behavior would change. However, there are cases where the sign is intended to create an awareness of surrounding scene elements, in which case the sign would still be salient by our definition, but not necessarily a risk object. Zhang et al. [312] agree to the importance of salience analysis, stating “A vehicle driving along the road is surrounded by many objects, but only a small subset of them influence the driver’s decisions and actions. Learning to estimate the importance of each object on the driver’s real-time decisionmaking may help better understand human driving behavior and lead to more reliable autonomous driving systems." Their work builds this estimation using interaction graphs which allow for the importance of scene elements to change depending on interactions observed between other scene elements (without involvement of the ego-vehicle). Our work is completely driver-centric; that is, we consider here signs which address the ego vehicle independent of the actions of other scene agents.

9.3 LAVA Dataset for Salient-Sensitive Traffic Signs

The LISA Amazon-MLSL Vehicle Attributes (LAVA) Dataset [313] includes a collection of traffic signs bounded and labeled in images taken from a front-facing camera, including 10 second video clips for full scene and trajectory context, accompanied by INS data. The data has been collected from the greater San Diego area, curated in a manner which includes a diversity of road types, traffic conditions, weather, and lighting. The traffic signs are categorized as stop, yield, do not enter, wrong way, school zone, railroad, red and white regulatory, white regulatory, construction and maintenance, warning, no turn, one way, no turn on red, do not pass, speed limit, guide, service and recreation, and undefined. The frequency of the sign types are described in Table 9.2. Signs are given a tag if electric (0.18%) or occluded (11.86%), and each sign is assigned an *is_salient* property with respect to the position of the ego vehicle (66.42%). For experimentation, we divide the 14,112 samples into 11,289 training instance, 1,411 validation instances, and 1,412 test instances, ensuring no scene is divided between sets.

Table 9.2. Sign type frequencies in the LAVA dataset. The data is non-uniformly distributed, reflecting an approximate real-world distribution of signs within the region of collection.

Sign Type	Frequency
Stop	725
Yield	72
Do Not Enter	134
Wrong Way	51
School Zone	172
Railroad	7
Red & White Regulatory	710
White Regulatory	3,048
Construction & Maintenance	773
Warning	2,364
No Turn	419
No Turn on Red	224
One Way	109
Do Not Pass	9
Speed Limit	563
Guide	249
Service & Recreation	2
Undefined	833

9.3.1 Automatic Road Type Classification

Reducing video and image data to mid-level semantic drive information has been shown to be important in understanding naturalistic drive data [314]. Similarly, we posit that information such as road scene and drive maneuver may contain important contextual information related to sign salience. Beyond traffic sign information explained above, we further classify each image scenes found in the LAVA dataset into 12 possible classes: highway, city street, residential, roundabout, intersection, construction zone, tunnel, freeway entrance, freeway exit, and unknown. This classification is performed automatically as follows:

- *Road Type by Global Coordinates*: LAVA sensor data includes latitude and longitude coordinates associated with each frame. Using Nominatim's reverse geocoding [315], we find the name and OpenStreetMap [OSM] ID of the current street. OSM provides categories of motorway, primary, secondary, tertiary, trunk, residential, roundabout, and pedestrian. We map primary, secondary, tertiary, and trunk to city street, motorway to highway, residential to residential, roundabout to roundabout, and pedestrian to parking lot. This excludes the classes of intersection, construction zone, tunnel, school zone, freeway entrance, and freeway exit.

- *Road Type by Object Detection*: We use CenterNet [316] trained on NuScenes [148] for detecting traffic signs, lights and traffic cones in the scene. If an image is detected to contain two or more traffic cones, it is classified as construction zone. Similarly, if one or more stop sign is detected, or three or more traffic lights are detected, the image is classified as an intersection.

- *Road Type by Class Change*: Using a contextual 10 second clip, if the frame class begins as highway and transitions to city street, the images of the clip are re-labeled as freeway entrance. Similarly, in the reverse case, the images are re-labeled as freeway exit.

The frequency with which signs are found on a particular road type are summarized in

Table 9.3. Road type frequencies per sign in the LAVA dataset.

Road Type	Frequency
Highway	1,788
City Street	8,285
Residential	1,243
Roundabout	17
Intersection	972
Construction Zone	300
Freeway Entrance	228
Freeway Exit	207
Unknown	1,072

Table 9.4. Maneuver frequencies per sign in the LAVA dataset.

Maneuver	Frequency
Forward	8,593
Stop	4,535
Turn Left After Stopping	18
Turn Right After Stopping	41
Turn Left	476
Turn Right	449

Table 9.3.

9.3.2 Maneuver Classification

For each frame in the LAVA dataset, we analyze the following 10 seconds of vehicle speed and yaw data for rule-based classification of the intended driving maneuver as Forward, Stop, Turn Left After Stopping, Turn Right After Stopping, Turn Left, and Turn Right. The frequency of maneuvers are described in Table 9.4.

9.4 Sign Salience Prediction

9.4.1 On-Right Classifier Baseline

While a random classifier would give an expected accuracy of 50%, we consider a reasonable trivial classifier which is better-grounded in traffic sign priors. This classifier assigns salience to signs which are located on or to the right of center, and non-salience to signs which are located left of center. This is consistent with typical drive-on-the-right traffic flow in the US, and should be adjusted for countries which drive on the left when comparing across datasets.

9.4.2 ResNet50 Model

We begin from the hypothesis that visual information can be used as a preliminary indicator of sign salience. The convolutional model uses the ResNet50 [317] convolutional architecture typically found in sign recognition. It takes a cropped sign region as input, and outputs a binary label for salience. We use an Adam optimizer with an initial learning rate of 10^{-3} and a batch size of 64.

9.4.3 Road Type Augmentation

To improve performance beyond the ResNet50 model, we consider the effects of road type on expected sign salience. Certain road types are less likely to see particular relevant signs; for example, a stop sign is unlikely to appear on a freeway, and a 65 MPH speed limit is unlikely to appear in a school zone. For this reason, if the features of such a sign are found in the convolutional layers, it is likely that the detected sign belongs to a different road or lane than that of the ego vehicle (perhaps past an off-ramp or overpass).

To test this hypothesis, we augment the model by appending a one-hot encoded vector representing the perceived road type to the flattened convolutional output prior to the fully-connected layer. We then add a ReLU activation, followed by another fully-connected layer, another ReLU, and a final fully-connected layer before the softmax binary output activation.

Until the last binary activation, we maintain 2,048 nodes at each fully-connected layer.

9.4.4 Image Coordinate Augmentation

Another feature which may improve model performance is the information contained in the pixel size and location of the detected bounding box within the scene image. In general, salient signs are found to the right of center and above the ego vehicle, as illustrated in Figure 9.5, and there is a relationship between the size of the sign and its location which can provide information about the 3D depth of the sign. This depth information provides further context to the model about where the sign may be located relative to the ego vehicle.

We augment the model by appending the top-left coordinate (x, y) of the bounding box (normalized to the image width and height), the bounding box width w, and bounding box height h to the flattened convolutional output prior to the fully-connected layer. Similar to the above road type augmentation, we then add a ReLU activation, followed by another fully-connected layer, another ReLU, and a final fully-connected layer before the softmax binary output activation. Until the last binary activation, we maintain 2,048 nodes at each fully-connected layer.

We further note that there is a relationship between expected sign location and road type, as illustrated by the heatmaps in Figure 9.6. This motivates experiments with a combined model, in which both road type and image coordinate features augment the convolutional output before the fully-connected layers.

9.4.5 Maneuver Augmentation

A vehicle's intended motion contains information about which signs will be relevant. For example, if a vehicle is planning a right turn, it will likely be in a right lane, and a sign which reads “Right Lane Must Turn Right" would be salient. Models which generate control for autonomous vehicles are of course unaware of the future trajectory, but it is reasonable that such model uses a series of planned maneuvers to navigate toward its goal. Accordingly, though the LAVA dataset does contain specific trajectory information, we use the coarse maneuver

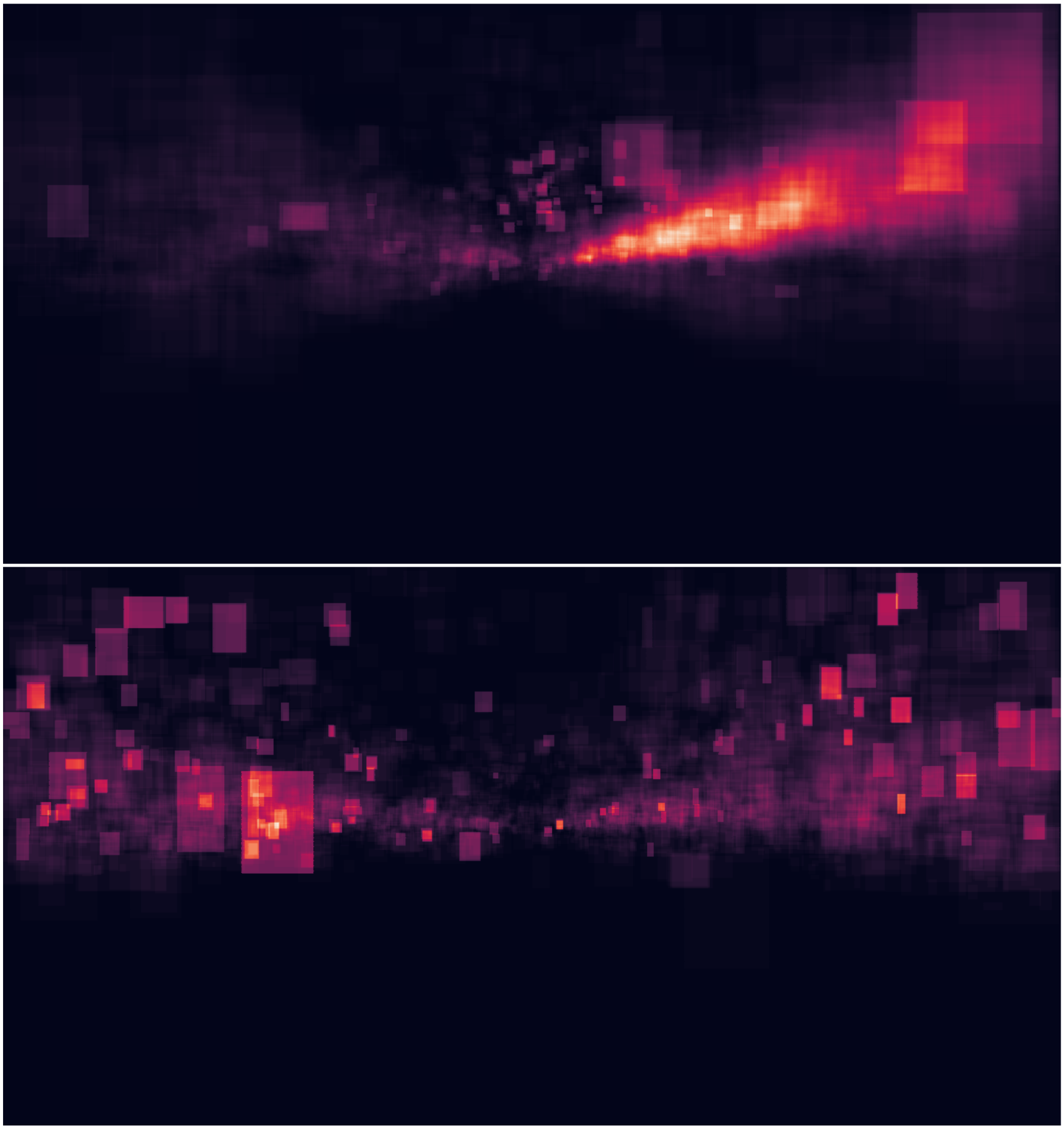


Figure 9.5. Heatmaps illustrating frequency with which a pixel is occupied by (a) a salient sign or (b) a non-salient sign.

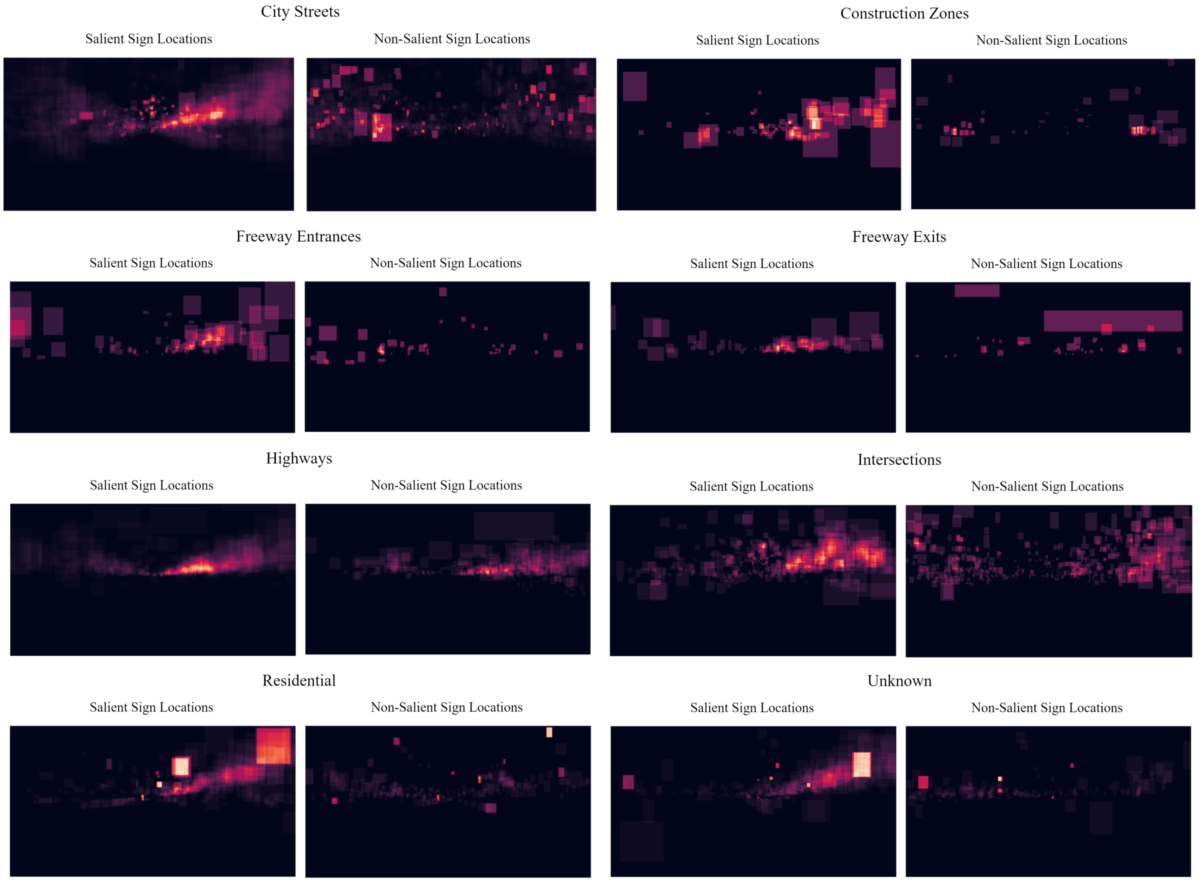


Figure 9.6. Heatmaps illustrating frequency with which a pixel is occupied by a salient sign (left) or non-salient sign (right) for city streets (a), construction zones (b), freeway entrances (c), freeway exits (d), highways (e), intersections (f), residential (g), and unknown (h).

Model	Accuracy
On-Right Baseline	0.6650
ResNet50	0.7422
Maneuver Augmentation	**0.7599**
Road Type Augmentation	0.7153
Coordinate Augmentation	0.7231
Coordinate & Road Type	0.7188
Coordinate & Maneuver	0.7252
Coordinate & Road Type & Maneuver	0.7358

Table 9.5. Classification accuracy of the sign salience models.

classification instead, as this is a reasonable substitute for the vehicle's intended path without assuming a particular trajectory.

As in the previous augmentations, we integrate this classification into a one-hot encoded vector, appended to the feature set just before the fully-connected layers. We perform experiments in combining image, maneuver, road type, and image coordinate features, summarized in Table 9.5. Convolutional methods and augmentations outperform the trivial classifier, achieving approximately 10% improvement. Image coordinates (which indicate the location and distance of the sign relative to the ego vehicle position) do not appear to enhance beyond the ResNet50 baseline, though we expect that as the dataset grows, the performance of these models will improve as more examples of possible sign positions are used in training. From results on this dataset, augmentation with the vehicle's maneuver information shows the strongest results in determining sign salience.

9.5 Conclusion & Future Research

In this work, we presented

- an analytical dimension of sign salience to weigh importance of particular traffic signs in path planning
- a traffic sign dataset which contains information on this property, with ability to infer road

type and maneuver intent, and

- analysis of models for prediction of sign salience.

The property of sign salience is intended for use in downstream path planning, where it could strategically penalize missed sign detections, sign classifications, and control decisions in salience-aware models. Extensions of the work include the conversion of salience from a binary to scalar property, and methods of determining scalar salience using subjective labelling between drivers. Sign salience is of further importance to driver-assistance systems which seek to understand human readiness and attention [318], and systems with augment a driver's scene awareness to the full surround [56]. The LAVA dataset continues to grow, with an expected volume for future work which is four times the size available to this study.

Acknowledgements

Chapter 9, in part, is a reprint of the material as it appears in Greer, Ross, Jason Isa, Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. "On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2022). The dissertation author was the primary investigator and author of this paper.

Chapter 10

Salient Sign Detection in Safe Autonomous Driving: AI Which Reasons Over Full Visual Context

10.1 Introduction

Detecting and recognizing traffic signs is an important module for an autonomous vehicle to observe and interact with its surroundings in a safe manner. The Safety of the Intended Functionality (SOTIF) process [319] examines highly automated systems for possible hazards and triggering events for unintended behaviors; in this framework, failure to detect a sign crucial to driving performance would be considered a triggering event, independent of the hazardous events, based on system limitations. Accordingly, detection systems are continuously improved to push the safe limits of their operation. Until recently, standard object detectors operated by proposing regions of interest or considering a standard set of anchors or window centers within an image, and classifying the contents of the found region. These approaches are typically limited by the span of the convolutional filters which drive them; these filters operate on local windows, or with a pre-determined span and spacing. While the reach of the convolutional filters can be tuned to spread and cover the entire image, doing so creates massive computational costs or creates gaps in coverage. As a solution, the popular transformer model has been proposed as a means of reasoning over the entire image and bringing forward features relevant to the region of interest. To minimize computational costs, this approach has been further refined to include a stage of learning (via a limited number of deformable attention heads) where an image should be sampled to extract meaningful relational features to the region of interest. This approach is known as the Deformable Detection Transformer, introduced in technical detail in the following section.

While advances in detection may improve sign recall, we pose one more consideration to be addressed in driving scenes: many signs simultaneously compete for the attention of a human driver or autonomous driving system. While the ideal intelligent vehicle detection module will have perfect precision and recall of all signs in the field of view, environmental noise and underrepresented examples make it possible that detectors continue to make mistakes. However, on the assumption that error is unavoidable, there are some errors preferred over others. For example, it is less critical that a vehicle passing by a freeway exit sees the sign corresponding to the speed limit of an off-freeway side street, or that a vehicle in the right lane preparing to make a right-hand turn sees the lane guidance for the left lane to navigate the intersection. We ascribe this quality of pertinence and attention-worthiness to the word salience, as introduced in [54]. We define the term as follows, with clarification on edge cases further described in Methods section:

Salience: A sign is salient if it has the potential to directly influence the next immediate decision to be made by the ego vehicle if no other vehicles were present on the road. Additionally, for signs directing traffic by lane, only signs pertaining to the lane the ego vehicle is in can be classified as salient. In the case of multiple sequential intersections or highway exits visible in the same frame, only signs pertaining to the next immediate intersection or exit could be labeled as salient.

Recent research in sign salience has shown that factors such as sign location, sign appearance, road type, and planned vehicle maneuver can be used to classify signs by salience [54]. Here, we propose a benefit of sign data with salience annotations: salience-aware training methods can be used to improve training of sign detection systems. We make three contributions: (1)

creation of the large, salience-annotated LAVA Salient Signs Dataset, (2) definition of Salience-Sensitive Focal Loss, and (3) experimental evaluation of the impact of Salience-Sensitive Focal Loss while training detection transformer models.

10.2 Related Research

Traffic Sign Detection and Classification Models Traffic sign detection has been well addressed by the field such that near perfect sign detection can be achieved on public sign datasets like the German Traffic Sign Detection Benchmark [320] and similar benchmarks. Detecting traffic signs requires cameras monitoring traffic scenes, which allow us to extract frames from videos and build traffic sign annotation datasets. Trivedi et al. [321] proposed that the best way to capture this traffic surveillance is through a multicamera surveillance approach known as distributed interactive video arrays (DIVA). DIVA helps address issues single view cameras have like handling occlusion and having many overlapping views to obtain 3D information. Such a multicamera system can facilitate easier traffic sign detection by addressing the issues mentioned. Some examples of high performance of traffic sign detection on public traffic sign datasets include: Using a separate traffic sign detector model and then a sign recognition model [322]. The traffic sign detection model learns the color of the sign and then the shape, and the sign recognition model works best with an ensemble of CNNs. A fully convolutional network to guide traffic sign proposals and then a CNN for sign classification [323]. The FCN learns the rough regions of where the traffic signs are present and the CNN identifies the traffic signs and removes false positives with non-max suppression. A Pyramid Transformer that uses atrous convolutions and a RCNN as a backbone [324]. This approach improves the network's ability to detect traffic signs of various sizes. Using transfer learning with state-of-the-art object detection models on the German Traffic Sign Detection Benchmark dataset [325]. Faster R-CNN Inception Resnet V2 achieves the best mean average precision while R-FCN Resnet 101 has the best tradeoff between accuracy and execution time.

Transformers have begun to outperform other deep learning techniques like CNNs since they can reason over full image context, or learn where to look to extract more features from an image. The detection transformer DETR [326] is a transformer that allows to learn such global image context and achieves state-of-the-art performance on the COCO object detection dataset. DETR is an end-to-end object detection module that treats object detection as a direct set prediction problem and removes the need for any hand-designed components used by other object detection models. A main weakness of DETR is that it has low performance on detecting small objects. Deformable DETR [327] builds on DETR, reducing the computational complexities and also improving performance on detecting small objects. Deformable DETR uses a different attention module that focuses on a subset of sampling points to perform object detection. This method shows theoretical promise in situations where novel, unusual, or newly emergent signs may appear [328], as the signs can be detected not only on the contents of a box which anchors and tries to recognize the sign's face pattern, but also through inferring on learned generic, face-independent contextual features from training. In this work, we apply this state-of-the-art object detection module to the application of traffic sign detection. In addition, we show that we can steer Deformable DETR to improve performance on salient signs via a novel loss function.

10.3 Traffic Sign Datasets

There are various traffic sign datasets that allow for researchers to develop traffic sign detection and classifications dataset. A comparison of the size and features of many traffic sign datasets can be seen in [54]. For this paper, we extend the LISA Amazon-MLSL Vehicle Attributes Dataset (LAVA) [75] to create the LAVA Salient Signs (LAVA SS) Dataset, the only dataset which includes the salience property we are interested in utilizing. The datasets and their important properties are listed in the table below:

Table 10.1. Comparison of Traffic Sign Datasets. The LAVA Salient Signs (LAVA SS) Dataset is used for our research and is the only dataset in this table to include the salience property for traffic signs.

Dataset	Number of Images	Important Features
LISA Traffic Sign Dataset [291]	7,855	occlusion, on-side road
LISA Amazon-MLSL Vehicle Attributes Dataset [75]	14,112	10s video context, occlusion, salience
LAVA Salient Signs Dataset	31,191	10s video context, occlusion, validated salience

10.4 Traffic Object Salience Research

Learning to focus on salient vehicle objects and construct vehicle visual attention mechanisms has been studied by various researchers, with many using different definitions of what it means to be a "salient" traffic object. We categorize two main types of object saliency from related research: instructive and attentive salience. Attentive salience relates to what objects and directions drivers tend to look at even if these objects may not be what a driver should look at. For this definition of salience, it is often important to monitor the driver's eye gaze to estimate where they are looking at. Tawari and Trivedi [329] use driver pose dynamic information to determine the likelihood of a driver gaze zone. This approach tracks facial landmarks like eye corners, nose tip, and nose corners to determine head pose and use the pose to predict the gaze estimation. They found using head pose dynamic features over time increased performance versus using static features like current head pose angles. Robust attentive salience systems must be invariant to different subjects, scales, and perspectives. Vora et al. [223] address this gaze generalization issue using a convolutional neural network to predict driver gaze direction. To improve generalization, they collected a large naturalistic dataset that used ten different subjects and was tested on three unseen subjects. Dua et al. [306] create the first large-scale driver gaze mapping dataset DGAZE, allowing to study attentive salience and where drivers tend to look at for different road and traffic conditions. This dataset contains data from a lab setting of road and driver camera views. Pal et al. [310] learn attentive salience by developing a model named SAGE-Net that uses attention mechanisms to learn how to predict an autonomous vehicle's focus of attention. SAGE-NET uses driver gaze and other important properties like the distance to objects and ego vehicle speed to determine object saliency. Tawari et al. [284] represent gaze behavior for a sequence of image frames by constructing a saliency map using a fully convolutional RNN. The saliency map uses three kinds of pixels: salient (positive) pixels, non-salient (negative) pixels, and neutral pixels. Other than gaze estimation, other important factors like predicting driver maneuvers and braking intent are important to understand attentive salience. Ohn-Bar et al. [330] use a multi-camera head pose estimation model to predict overtaking and braking intent and maneuvers. This system emphasizes real-time performance, which is critical for any attentive salience model in order to timely observe the driver state and react if they are distracted.

In contrast to attentive salience, instructive salience aims to emphasize important objects which an ego vehicle should observe and respond to; these objects should influence the car's future decisions. Our work focuses on instructive saliency and highlights what traffic signs the car needs to be aware of to safely operate. Instructive salience models are often more costly since labeling important objects requires understanding how various road objects and signs influence a driver's decisions and vehicle navigation, so a cognitively-demanding and maneuver-aware manual process is required to annotate such data. To overcome these challenges, Bertasius et al. [331] use an unsupervised learning approach to learn how to detect important objects in first-person images without any instructive salience labels, skipping the costly manual annotation process. The unsupervised network uses a segmentation network to propose possible important objects, and this output is fed into a recognition agent which uses these proposals and other spatial features to predict the important objects. Greer et al. [54] utilize a supervised learning process to classify salient road signs, which can be applied for efficient dataset annotation in future road sign data collection after initial training. Lateef et al. [308] use a conditional GAN to predict what a driver should be looking at in a traffic scene, which parallels this instructive salience definition. For constructing ground truths, they use semantic labels (annotations of traffic objects in images) from various autonomous driving datasets and use various saliency detection algorithms that select which object annotations are the most important. Zhang et al. [312] use interaction graphs to perform object importance estimation in driver scenes. The interaction graph updates features of each object node through interactions with graph convolutions. This task learns to model instructive saliency, as Zhang et al. note that their object importance definition relates to how objects can help with the driver's real-time decision making and improve safe autonomous driving systems.

10.5 Methods

Data Collection The LISA Amazon-MLSL Vehicle Attributes (LAVA) dataset contains labeled bounding boxes of traffic signs taken from a front-facing camera of a vehicle. This dataset was collected from the greater San Diego Area and contains a variety of road types, lighting, road types, and traffic conditions. The traffic signs are categorized as stop, yield, do not enter, wrong way, school zone, railroad, red and white regulatory, white regulatory, construction and maintenance, warning, no turn, one way, no turn on red, do not pass, speed limit, guide, service and recreation, and undefined. Along with the traffic sign categorization, we carefully labeled the sign salience property for all the sign annotations. A sign salience validation process was also performed in which the salience property for a sign annotation was checked again for consistency with the above definition of salience; the resulting dataset is referred to as the LAVA Salient Signs (LAVA SS) Dataset. This data collection process ensured that the salience property was properly labeled and the curated dataset had accurate ground truths. In the process of annotation, the provided definition of salience was used as a standard for annotation, with select frequent ambiguous cases handled according to the additional criteria below: Guides (signs which indicate the street name, often green) at intersections and freeways were labeled as salient, as long as such signs were the closest such guide in the scene. That is, in the case of multiple sequential intersections, only the guides of the nearest intersection to the car would be labeled as salient. Salient guides should be visible to the vehicle and indicate a possible street the car could turn onto or an exit the car could take. We note that guides tend to have the highest annotator ambiguity, as the class "guide" contains instances of street-level guides as well as freeway-level guides, which may be interpreted differently by different annotators. Likewise, parking guides are a highly missed ground-truth annotation. For this reason, certain applications may benefit from computing precision disregarding guides and parking signs, especially since such signs are less safety critical. Instructions pertaining to HOV or Carpool Lanes are marked as salient when the vehicle is moving in the direction of traffic, regardless of lane. Such a sign may indicate a lane available to the intelligent vehicle for optimized traffic flow, or a lane which the vehicle is required to leave if requirements are not met. A “No Parking” sign is salient only if the ego vehicle is in a lane which has immediate access to the restricted parking location (e.g. far right lane). An example of this rule is shown in Figure 1. While most signs which are facing backwards are marked as non-salient (since they provide instruction to an oncoming lane), in some cases, a yellow reflective warning sign is placed on the back of the sign. In these cases, we mark this as a salient warning sign if the sign is adjacent to the ego vehicle’s lane. The vehicle should be aware of such signs to avoid collision with the sign or median. This rule is exemplified in Figure 2. Signs which indicate a fine for littering or carpool violations are regarded as non-salient, since an intelligent vehicle should not be littering under any circumstance, nor motivated by the cost of breaking a traffic ordinance.

Figure 1. Because this No Parking sign is located in the lane closest to the ego vehicle, it is considered salient. Were the ego vehicle in the left lane of a two lane road, this would be annotated as non-salient.

Figure 2. This sign is facing away from the ego vehicle, but because the reflector is placed to warn the vehicle of its presence, it is annotated as salient.

Example sign annotations from the LAVA Salient Signs dataset are shown in Figure 3. The LAVA Salient Signs dataset contains 31,992 sign annotations with 20,377 annotations being salient and 11,615 annotations being non-salient. The sign type frequencies for the LAVA Salient Signs Dataset are defined in Figure 4. Because the data was collected and annotated using a selection method which promotes maximal coverage of driving area (including diversity of driving environment, conditions, and road types), the non-uniform distribution of signs may reflects the real-world distribution of salient and non-salient signs as well as the real-world distribution of sign categories during naturalistic driving.

Figure 3. Example Sign Annotations from the LAVA Salient Signs Dataset. A green bounding box indicates a salient sign and a red bounding box indicates a non-salient sign. As shown in the figure, the salient annotations often mean that the sign relates to the current lane or intersection the driver is in and provides meaningful information that affects the driver's future actions. On the other hand, the non-salient signs are often in a different lane, intersection, or face the wrong way, so these signs don't offer any important information. A vehicle's intended maneuver is important in classifying sign salience, so temporal dynamics should be considered when annotating and utilizing salience data, as explained in [54].

Figure 4: Sign Type Frequencies in the LAVA Salient Signs Dataset. The blue columns are for salient signs and the orange for non-salient signs. The White Regulatory, Construction & Maintenance, and Warning Signs were the most common sign types. This distribution of signs may be dependent on location, as all of our data was collected in the greater San Diego area.

10.6 Sign Detection with Deformable DETR

We use Deformable DETR, introduced in the Related Works section, to detect signs in the images. This detection method forms our performance baseline, described by Figures 2 and 3. We split the LAVA Salient Signs Dataset into 25,591 training instances, 3,200 validation instances, and 3,201 test instances. The model is trained for 15 epochs, retaining the model which reports the strongest precision (with a "hit" at 0.5 intersection-over-union, and 100 maximum detections per image). We use a ResNet50 backbone [317], 300 attention heads, a learning rate of 0.0002, a batch size of 2, and employ gradient clipping and learning rate decay.

10.7 Prioritizing Salient Signs with Salient-Sensitive Loss

The bounding box regression module of each Deformable DETR detection head is a 3-layer feed-forward neural network. Each detection head also has one linear projection for classification of the estimated bounding box into categories of foreground (object) or background (no object). This classification is trained using a sigmoid focal loss [332], an extension of standard categorical cross-entropy which down-weights easy examples to focus training on hard negatives. The equation for focal loss is

$$\mathrm{FL}(p+t) = -\mathrm{FL}(1-p+t)\log(p_t), \tag{10.1}$$

where FL is a hyperparameter to balance the focal loss among other loss functions, is a focusing parameter to control the influence of hard negatives, and pt is the predicted probability associated with the ground truth class.

As explained in the introduction, the goal of our detection model is to prioritize successful detection on signs which are salient, ideally placing any model error on non-salient signs. To achieve this, we weigh the focal loss heavily for salient signs according to the function

$$\mathrm{FL}(d, p_t) = -\mathrm{FL}w_{\mathrm{SS}}(d)(1-p_t)log(p_t) \tag{10.2}$$

where $w_{ss}(d) = ss$ if the ground truth sign nearest detection d is salient, and $w_{ss}(d) = 1$ otherwise. In our case, we use a hyperparameter $ss = 4$. We name the function $FL(d, p_t)$ salience-sensitive focal loss.

10.8 Results

The performance of the Deformable DETR model on the LAVA Salient Signs Dataset with and without salience-sensitive focal loss is provided in Figures 5-7. These figures display interpolations between precision-recall pairs generated with a detection thresholding of 0 to 1 in increments of 0.1 (with an early stop at thresholds where no positive detections are made). We note that thresholds should be tuned for precision and recall according to intended application; a representative descriptor of performance is given by the precision-recall curves. Results suggest that not only does training with salience-sensitive focal loss distribute error to non-salient signs instead of salient signs, but that the method actually improves overall performance of the model under otherwise equal training.

Figure 5. Deformable DETR shows uniformly better performance in recalling salient signs when using Salience-Sensitive Focal Loss. Additionally, as a general measure of performance, the area under the precision-recall curve is greater when using Salience-Sensitive Loss.

Figure 6. Deformable DETR additionally shows better performance in recalling all signs (both salient and non-salient) when using Salience-Sensitive Focal Loss. A possible reason for this improvement is that signs which are salient tend to be localized to particular image regions which amass both sign types, whereas some locations of non-salient signs would very rarely have a salient sign appear. This may help guide the transformer as it learns which regions of the image to attend.

Figure 7. How well does the Salience-Sensitive Loss bring out performance on salient signs? In this graph, we show the difference in performance between salient sign recall and all sign recall (in other words, how much better is the model at recalling salient signs than the aggregate collection of signs). Deformable DETR does generally perform better on salient signs than all signs together, but with exception as precision increases (in fact, negative at its greatest precision). On the other hand, Deformable DETR with salience-sensitive focal loss maintains improved performance on salient signs, and at greater margin than the baseline model.

10.9 Concluding Remarks

Detection transformers make use of full-image context in a selective manner, and this property makes them an excellent candidate for tasks which often require human drivers to make evaluations over which portion of their visual field to attend to. We illustrated the performance of the recent (and computationally tractable) Deformable DETR model on sign detection for a large dataset even under limited computational budget, providing a baseline for model performance on the dataset. Preliminary results are provided under reduced training time to illustrate the potential of detection-transformer-based methods and to provide a clear demonstration of the impact of modified loss functions on model performance compared to a baseline. Under elongated training regimens and increased dataset sizes, sign detection modules would reasonably be expected to perform to the standards of comparable benchmark models and datasets as described in related research.

We expand this analysis, noting that road objects carry an implicit importance and relevance to the ego vehicle. By including this property, salience, in the training regimen, we show that the sign detector can be further improved, both in general performance and especially in recall of signs which are most important to the safe operation of the autonomous vehicle. Gains in sign detection performance afforded via modification of the training loss function, especially in recalling salient signs, are directly related to the safety of the vehicle in navigating a scene and responding appropriately and safely to surrounding agents.

Acknowledgements

Chapter 10, in part, is a reprint of the material as it appears in Greer, Ross, Akshay Gopalkrishnan, Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. "Salient Sign Detection in Safe Autonomous Driving: AI Which Reasons Over Full Visual Context." Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV) (2023). The dissertation author was the primary investigator and author of this paper.

Chapter 11

Robust Traffic Light Detection Using Salience-Sensitive Loss: Computational Framework and Evaluations

11.1 Introduction

11.1.1 Overview

Accurate detection and recognition of traffic lights is vital for an autonomous vehicle to observe and interact with its surroundings in a safe manner, communicating information relevant to predicting an agent's trajectory [171] [63] [64] [130] or enabling ADAS features [89]. In general, standard object detectors for traffic lights, signs, or pedestrians operated by proposing regions of interest, which involves considering a standard set of anchors or window centers within an image, and classifying the contents of the found region [333] [334]. However, approaches involving regions of interest are generally limited by the computational costs and gaps in coverage that come with creating convolutional filters that cover an entire image. As an alternative, the transformer model has been proposed to cover an entire image and select only features that are relevant to the region of interest. Transformers allow for more parallelization than older approaches and thus, reduce training times. To minimize the problem of computational costs that the regions of interest approach has, the transformer approach has been further refined to include a stage of learning (via a limited number of deformable attention heads) where an image should be sampled to extract meaningful relational features to the region of interest. As defined in [55], this approach is known as the Deformable Detection Transformer, described in technical detail in the following section.

While this transformer-based approach may improve traffic-light detection, consider a driving scene with numerous traffic lights that are simultaneously presented to an autonomous driving system, containing some lights that are relevant to the vehicle and some of that are not. Ideally, a detector will be able to precisely recall the statuses of all traffic lights in an image and make complex driving-related decisions accordingly. However, factors such as environmental noise and underrepresented driving situations may lead to errors in detectors, but some errors may be preferred over others. For instance, it is less critical that a vehicle making a right turn while in the right lane sees the status of the traffic light corresponding to the left lane, or that a vehicle continuing straight along an intersection sees the status of a traffic light perpendicular to the car's current trajectory. We ascribe this quality of pertinence and attention-worthiness to the word salience, as introduced in the context of traffic signs in [54], adopting an equivalent definition in the following section. We illustrate the use of our salience-sensitive computational framework in Figure 1.

11.1.2 Salience

As defined in [54], a traffic light is salient if it directly influences the next immediate decision to be made by the ego vehicle if no other vehicles were present on the road. For instance, consider if a car were to be driving straight through an intersection. We would classify the vehicle-facing straight traffic light as salient, as its status will influence whether the vehicle will stop or proceed. However, we would not classify the protected traffic lights for left and right turns as salient, as the status of these lights is irrelevant to the vehicle's decision. In general, after taking into account factors such as the ego vehicle's current lane and next vehicle maneuver, we would selectively classify traffic lights as salient. Similar to the definition in [54] involving traffic signs, in the case of multiple sequential intersections or traffic lights that are present in the same

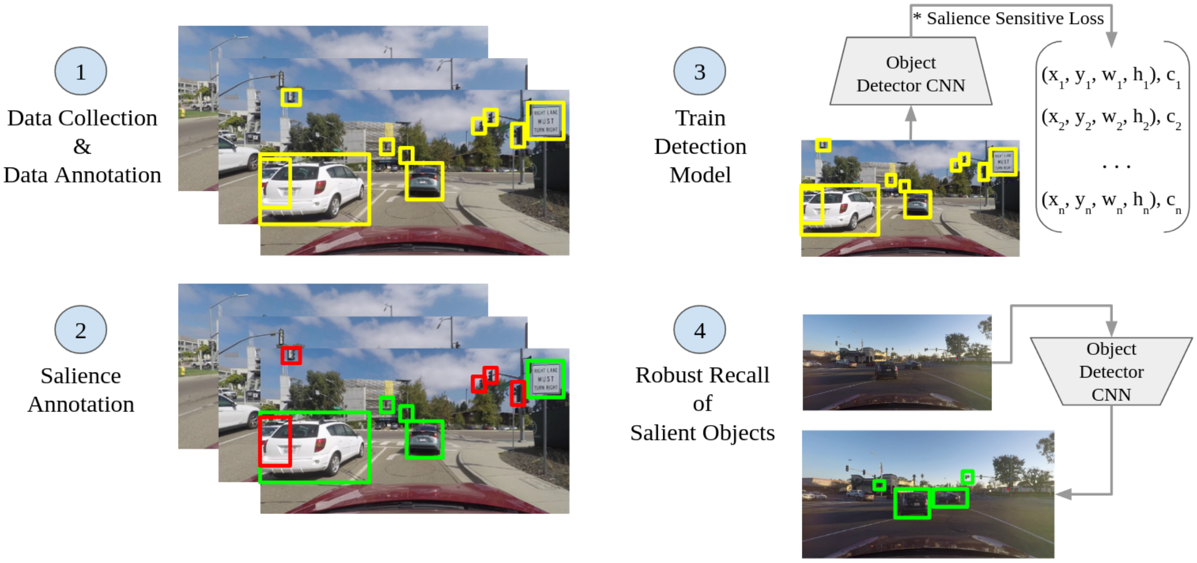


Figure 11.1. The general computational framework for leveraging object salience in deep learning tasks. First, data is collected and annotated per task specifications. The data is provided an additional annotation, salience, which can be provided by either an expert annotator or computer model. The salience property is utilized during the training process, leading to a robust recall of salient objects during deployment inference.

frame of data, only the traffic lights that are relevant to the vehicle's immediate next action would be important to detect, and thus, would be classified as salient. For example, if the ego-vehicle is in a left-turn lane, then we would classify the traffic light signaling the left-turn as salient, while the non left-turn traffic lights in the same intersection would be considered non-salient. Figure 11.2 shows qualititative examples of salient and non-salient traffic light annotations from our collected dataset.

We propose that salience-aware training methods can be used to improve the training of traffic light detection systems. We make two contributions: (1) creation of the first salience-annotated LAVA Salient Lights Dataset, and (2) using the concept of Salience-Sensitive Focus Loss defined in [55], evaluation of the impact of Salience-Sensitive Focal Loss while training detection transformer models.

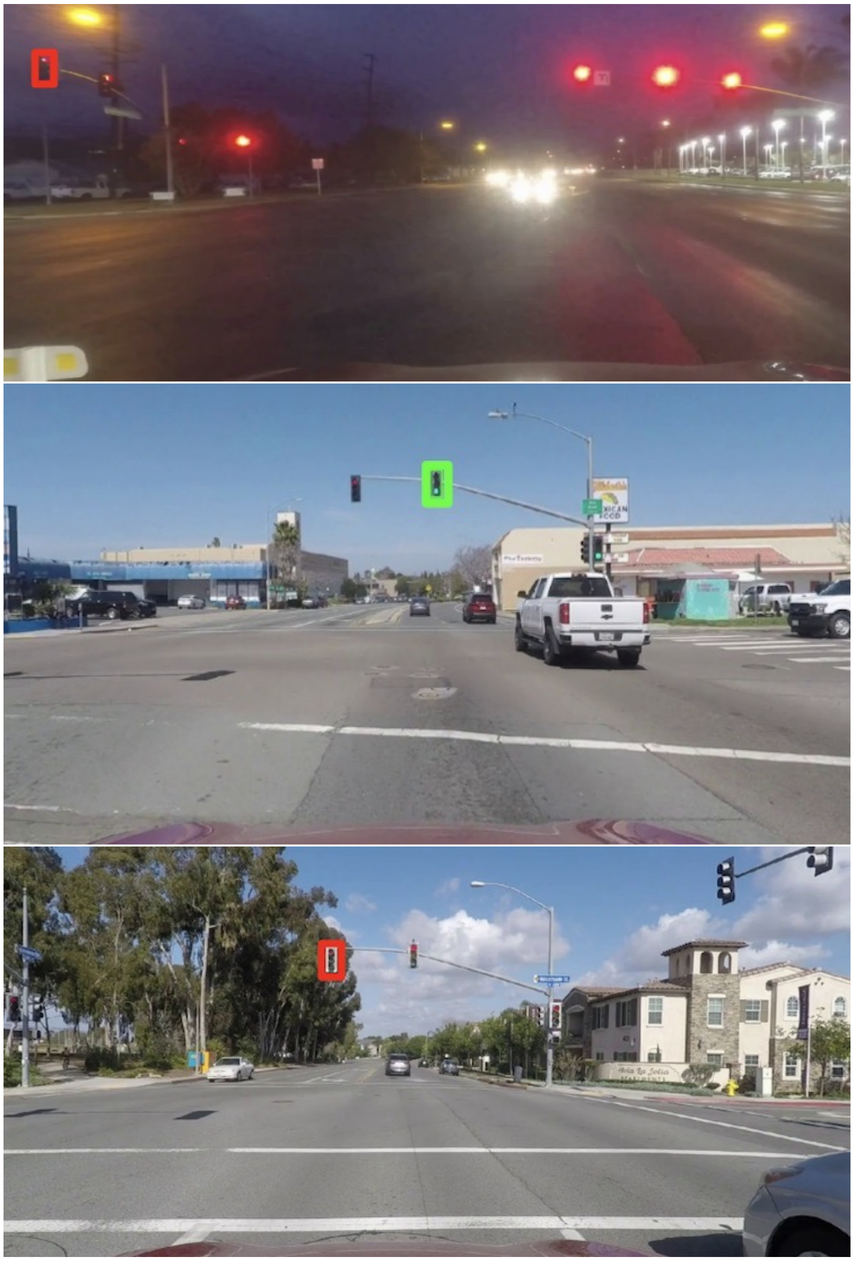


Figure 11.2. Annotated examples of the *salience* property for selected traffic light scenarios. A red box indicates a non-salient example while a green box indicates a salient example. Since the traffic light in the top image is in a different intersection than the driver, the light is not considered to be salient. In the middle image, the traffic light is in the same intersection of the driver and indicates the direction the driver is going in so it is considered to be salient. Finally, the bottom image is not salient since the driver is not in a left turn lane and thus the left turn traffic light is not relevant to the driver.

11.2 Related Research

11.2.1 Traffic Light Detection and Classification Models

There are various issues and challenges related to traffic light detection.

1. Illumination: Images containing traffic lights may contain different illumination due to various environmental factors [335]. Traffic lights also have variability in their own lighting, as sometimes the traffic lights themselves may be off or in different cases (green, yellow, red).

2. In driving scenarios such as intersections, traffic lights may appear in a variety of orientations, requiring a robust model that can recognize a traffic light in all different positions.

To specifically address illumination problems for traffic light detection, Shi et al. [336] proposed a model that is robust to different illumination conditions. The first step of the model uses an adaptive background suppression algorithm to highlight detected traffic lights, and the second step, the recognition module, verifies each candidate regions and classifies the traffic lights. Instead of using hand-designed features or algorithms, deep learning approaches with an abundance of data will allow a model like A CNN do learn filters and features that help identify traffic lights. Behrendt et al. [337] use a YOLO architecture with the classification network removed to perform traffic light detection. They emphasize that this network is optimized for automated driving, as the network is efficient enough to make real-time predictions. Image segmentation can be also be used to detect the locations of traffic lights. Weber et al. [338] devise the network DeepTLR, a single deep CNN that outputs a probability map that represents a traffic light being present in a certain region. With this region-wise classification, a regression module is used to predict a set of bounding boxes for detected traffic lights. Ennahhal et a. [339] evaluate traffic light detection performance on various state-of-the-art object detection models. They find that Faster R-CNN gives the best mean average precision for this task.

Table 11.1. Information on two traffic light datasets which are annotated with the salience feature. The LAVA Salient Lights is the first to include traffic lights from the United States.

Dataset	Num. Of Images	Country	Important Features
DriveU Traffic Light Dataset (DLTD)	232,039	Germany	color, directionality, num. of lamps, orientation, occlusion, “relevance"
LAVA Salient Lights Dataset	30,566	United States	color, directionality, occlusion, “salience"

11.2.2 Traffic Light Datasets

Traffic light datasets are invaluable in the realm of autonomous vehicles for building robust traffic light detection and classification systems. There are various traffic light datasets, many of which define various features and categories for traffic lights. The typical categories these labels touch on are: on/off, color, or go/warning/stop. Examples of such public datasets are the Bosch Small Traffic Light Database [337] and the VIVA challenge dataset [340]. While these datasets have been historically useful for the purposes of traffic light detection and classification, for the purpose of this paper, we want to consider datasets that incorporate the added annotation category of "salience".

Using this filter, we identified the DriveU Traffic Light Dataset (DLTD) [341]. This dataset includes 232,039 annotations from 11 different cities in Germany, with each annotation having tags to describe: color, directionality, number of lamps, orientation, occlusion level, and any other visual abnormality. In addition to these tags, the DLTD dataset also includes a special feature property tag they name "relevance", which they define to correspond to traffic lights that transport the information relevant to the planned route of the vehicle. This definition for “relevance" is equivalent to our definition for “salience". Our proposed dataset, the LAVA Salient Lights Dataset, shares this same key attribute of relevance/salience, and for the first time applies salience to US traffic lights.

11.2.3 Traffic Object Salience Research

The idea of saliency with regard to vehicle objects and visual attention mechanisms has been studied by many researchers. However, the definition of what constitutes a "salient" traffic object is not universally agreed upon. From the literature, we have identified two main categories of object saliency: attentive salience and instructive salience. Attentive salience corresponds to objects or regions drivers tend to look at, despite whether or not they are important to the driver's trajectory and/or decision-making. For attentive salience, a common approach is to monitor the driver's gaze and estimate what objects or regions they are looking at [330]. Tawari and Trivedi [217] take such an approach, where driver pose dynamic information was used to estimate a driver gaze zone. This system detects and tracks points of interest on the face such as eye corners, nose tips, and nose corners to estimate the head pose and thus predict the driver's gaze. They found that this approach increased performance over using static features such as head pose angles. For an attentive salience system to be robust, it must be invariant to different scales, perspectives, and subjects. To reach this gaze generalization, Vora et al. [223] utilize a convolutional neural network trained on ten different subjects to estimate the gaze direction. Also instrumental to the development of attentive salience, Dua et al. [306] construct the first large-scale driver gaze mapping dataset, DGAZE, which allows for further analysis of driver gaze in different road and traffic conditions. The SAGE-Net model developed by Pal et al. [310] learns attentive salience using attention mechanisms to predict an autonomous vehicle's focus of attention. This model utilizes driver gaze alongside other important metrics such as the distance to objects and the ego-vehicle's speed to evaluate object saliency. Tawari et al. [284], on the other hand, represent driver gaze behavior for a sequence of frames by building a saliency map via a fully convolutional RNN. This map uses three pixel classes: salient pixels, non-salient pixels, and neutral pixels. Attentive salience is also important for predicting driver maneuvers and braking intent. Ohn-Bar et al. [330] utilize a head pose prediction model to predict overtaking and braking intent. The system they outline has a strong emphasis on real-time performance, which is extremely important for attentive salience models.

Instructive salience, on the other hand, aims to prioritize traffic objects that are critical to the ego-vehicle's future trajectory. This version of salience emphasizes objects/regions that a driver should be observing in order to maneuver the vehicle properly, and is the version of salience that our work focuses on in regards to traffic lights. Instructive salience models are typically more costly than attentive models due to the manual labeling of important objects with respect to the ego-vehicle. This type of labeling is more cognitively demanding as the annotator must take into consideration both the ego-vehicles current position and future trajectory. To avoid this manual step, Bertasius et al. [331] utilize an unsupervised learning approach for detecting important objects without any saliency labels. This unsupervised model uses a segmentation network that proposes possible important objects, which are then fed to a recognition agent that incorporates spatial features to predict the important objects. Instead, using a supervised learning process, Greer et al. [54] classify the saliency of road signs, which can be utilized for efficient dataset annotation. This led to the LAVA Salient Signs Dataset [55], a dataset of 31,191 labeled traffic signs with a validated salience property. Lateef et al. [308] utilize a conditional GAN that predicts the important things a driver should be looking at within a traffic scene, matching the definition we use for instructive salience. For this model, they incorporate semantic labels from existing datasets and use saliency detection algorithms to predict which traffic objects are most important. Zhang et al. [312] also create a model for instructive saliency using interaction graphs that estimate object importance in driving scenes. They note that their object important definition corresponds to the objects that help with the driver's real-time decision-making, falling in line with our definition for instructive salience.

11.3 Methods

11.3.1 Data Collection

The LISA Amazon-MLSL Vehicle Attributes (LAVA) dataset contains labeled bounding boxes of traffic lights taken from a front-facing camera of a vehicle. This dataset was collected from the greater San Diego Area and contains a variety of road types, lighting, and traffic conditions. The traffic lights can be categorized with a status of on, off, or undefined, with a color of red, yellow, green, or undefined (traffic light is not on), true/false attributes for whether the light is directional or not, and with an occlusion level of non-occluded, partial, or major. Additionally, we added an additional light salience property for all traffic light annotations that were performed as either true or false. Due to the potentially ambiguous nature of saliency with regard to any given traffic snapshot, this additional property brings added difficulty with respect to accurate data annotation. As there is some level of subjectivity to a definition for saliency which takes into account a driver's attention or perceived relevance of road objects, it is important that any provided definition is appropriately constrained and that annotations are validated across raters to ensure consistency. A light salience validation process was performed where the boolean salience property for each annotation was double-checked for consistency with our definition of salience. This data collection and validation procedure ensured that we had properly labeled every traffic light as salient or non-salient depending on the traffic scene in the image. The resulting dataset after salience validation is known as the LAVA Salient Lights Dataset. In the process of annotation, the aforementioned definition of salience was used as a standard for annotation, with select frequent ambiguous cases handled according. to the additional criteria below:

- In the case of a frame where the lane the ego-vehicle is in is indeterminate, both traffic lights pertaining to straight and left turns were labeled salient.
- In the case of the ego-vehicle approaching an intersection in the left-most forward-bound lane with an approaching left turn lane opening, both traffic lights pertaining to straight and left turns were labeled salient until it is clear which direction the ego-vehicle will go in.

We collected 30,566 traffic light annotations, with 9,051 salient annotations and 21,515 non-salient annotations. Figure 3 shows the frequency of each light type in the LAVA Salient Lights Dataset.

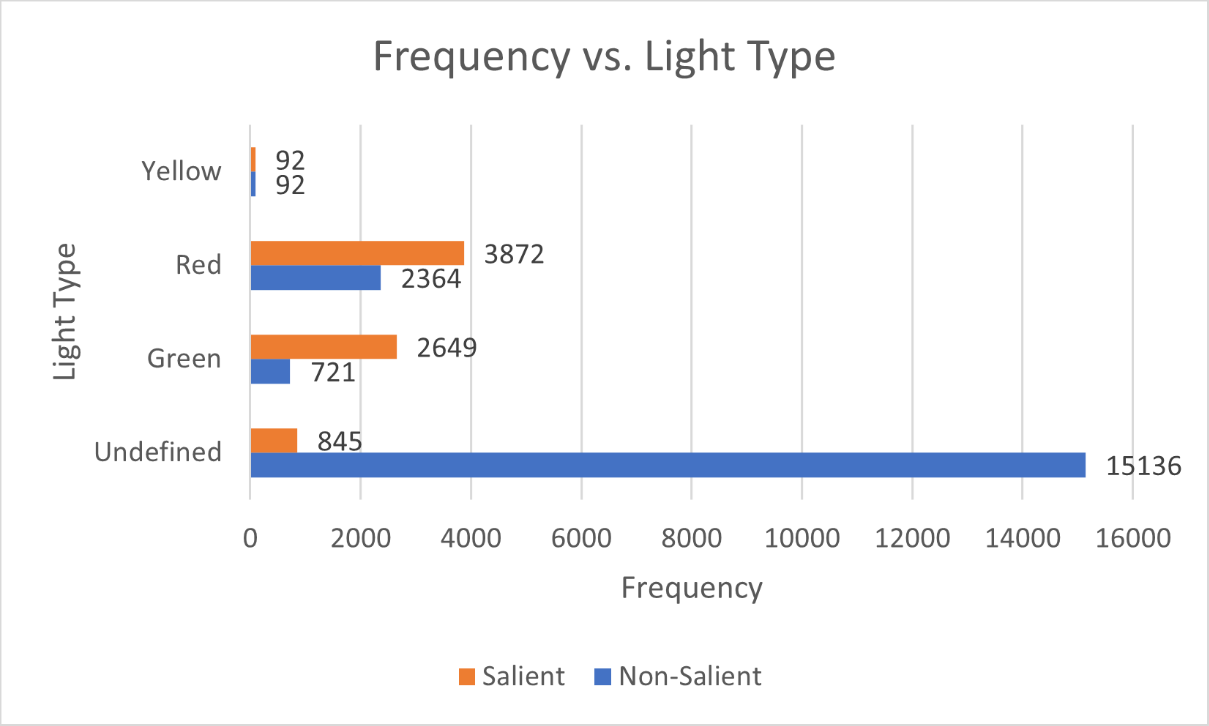


Figure 11.3. Chart of Light Type Frequencies in the LAVA Salient Lights Dataset. The orange rows represent salient lights and the blue row are for non-salient lights. The undefined non-salient light is the most common light type in the dataset, demonstrating just how many non-salient examples that are present in driving scenarios.

11.3.2 Light Detection with Deformable DETR

To perform Traffic Light Detection, we use the end-to-end transformer object detection model Deformable DETR [342]. For the Deformable DETR hyperparameters, we use a ResNet-50 backbone, 300 attention heads, 50 epochs, a learning rate of 0.0002, and a batch size of 2 due to GPU constraints. We also employ gradient clipping and learning rate decay. We define two approaches to perform the Traffic Light Detection:

1. A standard Deformable DETR model trained for traffic light detection

2. Deformable DETR model trained with a custom salient-light loss function.

We compare the two approaches to see if annotating salient traffic lights and focusing performance on such light types increases our detection precision and recall.

11.3.3 Salient-Sensitive Loss for Traffic Lights

The Deformable DETR involves two steps: bounding box regression via a 3 layer Feed-Forward-Network and bounding box binary classification. For this last step, bounding boxes are classified into either two categories: object or foreground. The classification step is trained through a focal loss equation which emphasizes performance on difficult examples. The focal loss equation is:

$$FL(p_t) = -\alpha_{FL}(1-p_t)^{\gamma}\log(p_t) \tag{11.1}$$

This focal loss equation adds a $(1-p_t)^{\gamma}$ factor to the standard cross-entropy loss function, where p_t represents the probability of a ground truth glass. The γ parameter is a focusing parameter which if increased puts more emphasis on more difficult and misclassified examples. α is a hyperparameter is used to balance emphasis on focal loss. We borrow from this focal loss function and customize it such that our loss function emphasizes performance on salient traffic lights. The salient-loss function is defined below:

$$FL(p_t) = -\alpha_{FL}\omega_{SL}(1-p_t)^{\gamma}\log(p_t) \tag{11.2}$$

The equation is similar to focal loss equation except for the addition of ω_{SL}. If the nearest ground truth light detection is salient, we set $\omega_{SL} > 1$ such that the loss function is now stricter and requires greater performance on salient lights. We found that the best value for ω_{SL} on salient examples was 4. Otherwise if the nearest ground truth detection is non-salient, $\omega_{SL} = 1$ and essentially regular focal loss. In considering system limitations, while in general this loss does not combat the learning of the generic focal less, but rather enhances its effects for important objects. However, one relevant consideration is the care placed in defining what is “important". Using this salient-sensitive loss function over the traditional focal loss function can open risks if the saliency property within the dataset is inaccurately or inconsistently annotated, hence why it is important to validate salience annotations as previously mentioned.

11.4 Experimental Evaluation

We evaluate the Deformable DETR Traffic Light Detection Models trained on the LAVA Salient Lights Dataset with and without salient loss. We divided the data at random, with 80% in training, 10% in validation, and 10% in test sets. We trained each model for 50 epochs.

With each detection made by the model, there is a simultaneous confidence score output. We threshold our detections by this confidence score, sweeping across values from 0 to 1 in increments of 0.1. Once we have collected these model predictions, we use IOU with a constant threshold and the ground-truth light detections to calculate the number of correct predictions. Using these sweeping confidence thresholds for detections and using IOU to find the correct predictions, we can generate a precision-recall graph. These charts are shown in Figures 11.4 to 11.6.

How well do these models recall traffic lights under this training regimen? Deformable

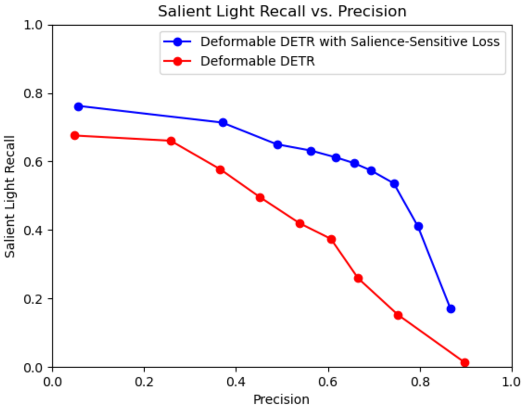


Figure 11.4. Recall of Salient Traffic Lights versus Precision on All Traffic Lights. The model trained with salience-sensitive loss (in blue) outperforms the model trained without, reaching consistently higher values of recall for similar values of precision.

DETR appears to be effective at learning to detect traffic lights as a base model, and Figure 11.4 shows that for salient lights, the salience-sensitive loss increases the performance at all points on the precision-recall curve, even giving a stronger concavity indicating sustained recall even under tighter precision thresholds. Similarly, Figure 11.5 also shows a clear separation and concavity for the model trained with salience-sensitive loss.

Does salience-sensitive loss succeed in prioritizing performance on salient lights versus lights which may be less important to the ego vehicle? Figure 11.6 shows that at high confidence values, the salience-sensitive loss creates a fairly strong difference in performance on salient lights versus all lights in the scene. This means that as the model becomes more scrutinous with an increased confidence threshold, the salient lights remain well-detected compared to the total collection of lights. Qualitative results illustrating model performance are provided in Figure 11.7.

11.5 Concluding Remarks

In autonomous vehicle planning, it is not always the nearest or largest objects which provide the most critical information. Training object detectors with salience-aware methods

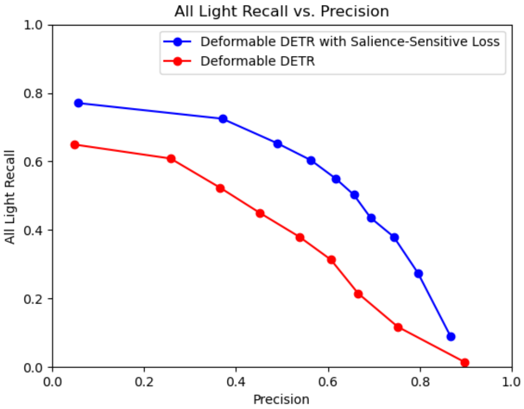


Figure 11.5. Recall of All Traffic Lights versus Precision on All Traffic Lights. The model trained with salience-sensitive loss (in blue) outperforms the model trained without, reaching consistently higher values of recall for similar values of precision.

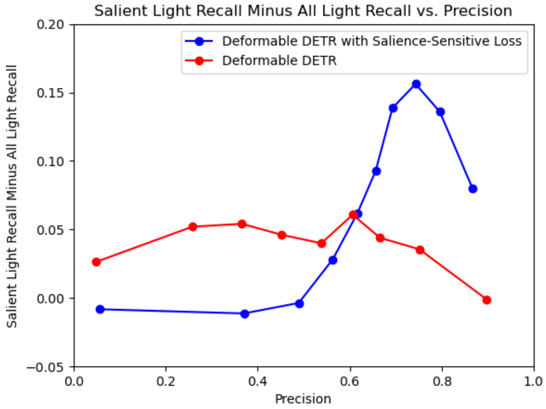


Figure 11.6. The difference in recall on salient lights versus all lights, plotted against precision on all traffic lights. This graph is meant to address the question *Does salience-sensitive loss successfully prioritize salient traffic lights?* Any time the graph is positive, the model is giving stronger recall of salient lights than overall recall. We see that this occurs naturally (red) without salience-sensitive loss, but by adding salience-sensitive loss (blue), higher confidence thresholds lead to recall difference of greater than 5%.

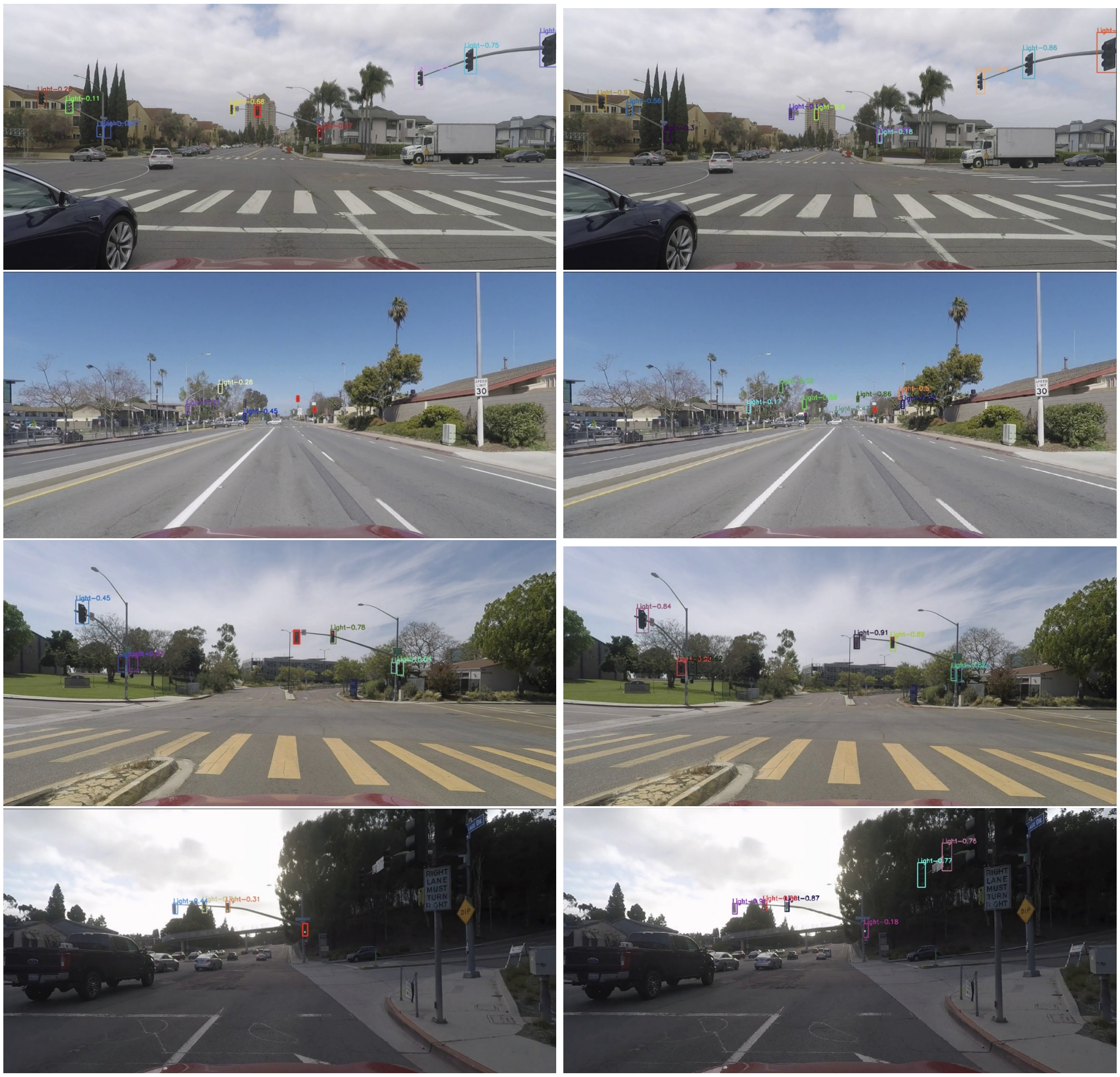


Figure 11.7. Qualitative results of training without salience-sensitive loss (left) and with salience-sensitive loss (right). In each example, a red border is given to signs which are “missed" by the detector. In these examples, the missed sign is salient and critical to the car’s decisions. These signs are sometimes further or smaller than other signs in the scene, making salience an important part of the training process since the “easiest" signs may not be the most important to the vehicle’s planning.

are critical for ensuring that objects critical to the driver’s decisions are emphasized in detection models. Using a transformer like Deformable DETR is a natural choice for this problem since transformers learn what portions of an image it should pay attention to. We have shown the effectiveness of salience-sensitive loss in guiding Deformable DETR toward more accurate object detection for a US traffic lights dataset. To further improve the model performance, we aim to annotate more traffic light examples to expand the LAVA Salient Signs dataset. Further research directions into salience annotation and salience-sensitive classification may continue to improve scene understanding, continuing towards an overall goal of robustly safe autonomous navigation through intersections.

Acknowledgements

Chapter 11, in part, is a reprint of the material as it appears in Greer, Ross, Akshay Gopalkrishnan, Jacob Landgren, Lulua Rakla, Anish Gopalan, and Mohan M. Trivedi. "Robust Traffic Light Detection Using Salience-Sensitive Loss: Computational Framework and Evaluations." Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV) (2023). The dissertation author was the primary investigator and author of this paper.

Part VI

Looking Forward

Chapter 12

Concluding Remarks

In this dissertation, I have presented a selection of ideas which center on the ways that autonomous systems should observe, react, and learn from uncertainty in the driving environment. Because this driving environment is semi-structured but paradoxically dynamic and ever-changing due to the impulses of human agents, it is important that systems are highly adaptable to new or ambiguous information. In my dissertation, I framed such capabilities around the ability of systems to recognize novelty in their observational data. Among the representations of novelty presented in this dissertation, language-based embeddings of visual data was shown to be semantically rich, allowing for unsupervised formation of clusters which contain unique weather conditions, road infrastructure, dynamic objects, vulnerable road users, and even sensor failures, allowing for a variety of applications in autonomous driving system safety and learning. The importance of novelty and the ability of language to capture salient information may be best exemplified by a 2024 feature rollout in Tesla's full-self-driving mode, in which FSD disengagements trigger the interface to ask the human occupant to describe their observations of what happened around the time the system disengaged[1]. I am happy to say that my research shared in Chapter 3 can make these same assessments automatically, without human intervention.

On a practical note, the research presented in this dissertation has applications across the popularized SAE autonomy levels. For example, handling human control takeovers safely is a necessary function of Level 3 driving, where the human may be expected to assume control

[1] https://teslamotorsclub.com/tmc/threads/fsd-audio-disengagement-notes.299566/

when alerted. Object detection and recognition algorithms presented throughout this dissertation have a place in ADAS systems which sit in earlier classification levels. And, towards robust autonomy in the levels beyond, I presented active learning frameworks which can help systems to overcome the human annotation bottleneck which prohibits complete use of colossal volumes of high-dimensional data in safe autonomous driving tasks. These same methods of data curation also have a place in system validation; in the same way that a chain is only as strong as its weakest link, observing system performance (whether perception, prediction, or planning) on “challenge" cases is one way we can measure our progress through the long-tail. In this direction, an immediate appropriate application is the audit of the large datasets used to train detection systems for improved balancing of object classes. As an example, in one of the largest public autonomous driving datasets, only 0.2% of pedestrian instances are observed using wheelchairs, which is the only mobility aid even classified, an important task emphasized in recent years through the research of the CVPR workshop on Accessibility, Vision, and Autonomy [343]. This is especially important because large data systems with severe data imbalance leave those under (or un) represented the most vulnerable to risk from mistakes in perception.

Further, tasks in understanding the driving environment should extend beyond detection-as-perception. Certain objects are more important to driving decisions for the ego-actor than others, an idea that I formulate in the concept of *salience*, and certainly with room for continued research into how salience should be represented, assigned, and used. Besides answering where attention “should" be placed, we may also want to understand where attention “is" placed by human drivers, perhaps helpful in guiding autonomy, and also helpful in noting where human drivers may err in their situational awareness. Further, salience may act as a bridge which connects scene objects (for example, understanding which scene objects are salient to Agent A and which are salient to Agent B may help to form logical groupings for planning and decision-making). As human drivers, we are aware of the relationship between various elements in the observed scene, and our autonomous systems must be able to do the same in a generalizable way.

Our driving environment is semi-structured; like any governing system, we have sets of rules and expectations, but only the ones based in physics are guaranteed to be followed. As some examples, we may perceive and organize lane markings, but these marking impose no physical requirements for drivers to conform. Likewise, a sign that says “No Right On Red" does not impede hurried drivers from continuing their route, as illustrated in Figure 12.1. How do we plan and make decisions under this pervasive ambiguity between what a scene tells us and what agents do? Or, times when the scene presents us with conflicting instructions, as in Figure 12.2? These plans are highly context-dependent, and the safest, most “understandable" decision may vary situation-to-situation. We cannot have frameworks which learn purely from perception of agent actions, nor purely from the scene; some form of reasoning and evaluation beyond perception and obstacle-avoiding control is required for safe, smooth, and human-relatable driving.

Perhaps part of the solution will lie in continued use of language embeddings for both scene understanding and planning, allowing for explainable decisions under such modal ambiguity as illustrated in Figures 12.1 and 12.2. When the computer is faced with conflicting modes of information (e.g. modules for sign and light detection understand the red light to prohibit motion, while the behaviors of surrounding vehicles indicate otherwise), modal confusion may occur. Similarly, modal confusion may occur in situations where a construction worker may be waving vehicles to drive through an ambiguously marked area, as in Figure 12.3, or at times when police officers may direct traffic at a broken light. Language may offer a common representation between modes, such that control decisions can both made *and* explained for purposes of validation and possible correction by human agents, whether on-the-fly or in offline learning.

Understanding how to make the control transitions between human driving and autonomy was a focus of this dissertation. In future research, understanding when to make transitions between modes of control will rely on similar principles of recognition of novelty (or sensing, perception, or planning failure), and choosing an appropriate mode of information and modeling to create a safe response. The formulation by [194] of corner cases as a coming-together of multiple modes of ambiguity seems a fitting framework for explainable and safe driving, where an intelligent vehicle may need to synthesize information from driver attention, outside agent

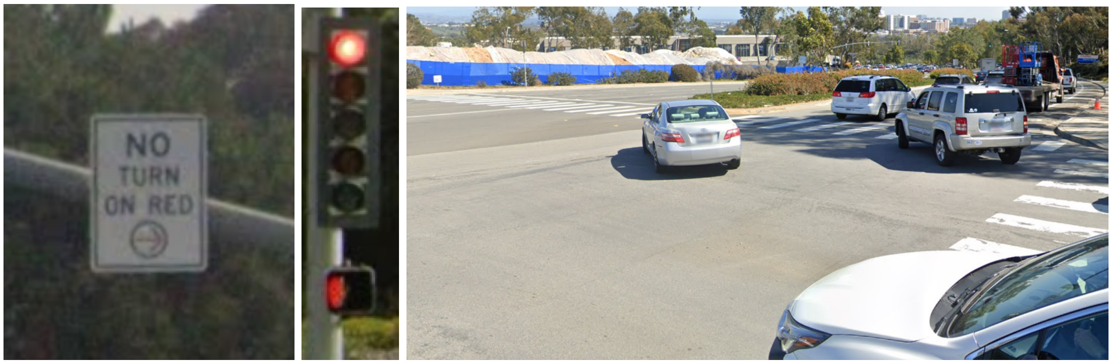


Figure 12.1. What should the autonomous agent do when it detects and understands a sign that says “No Turn On Red", perceives the red light, yet observes vehicles turning? How does this interpretation change when in a construction zone, behind an emergency vehicle, or responding to direction of a traffic-regulating police officer? What scene information would help the machine to make a safe and explainable decision?

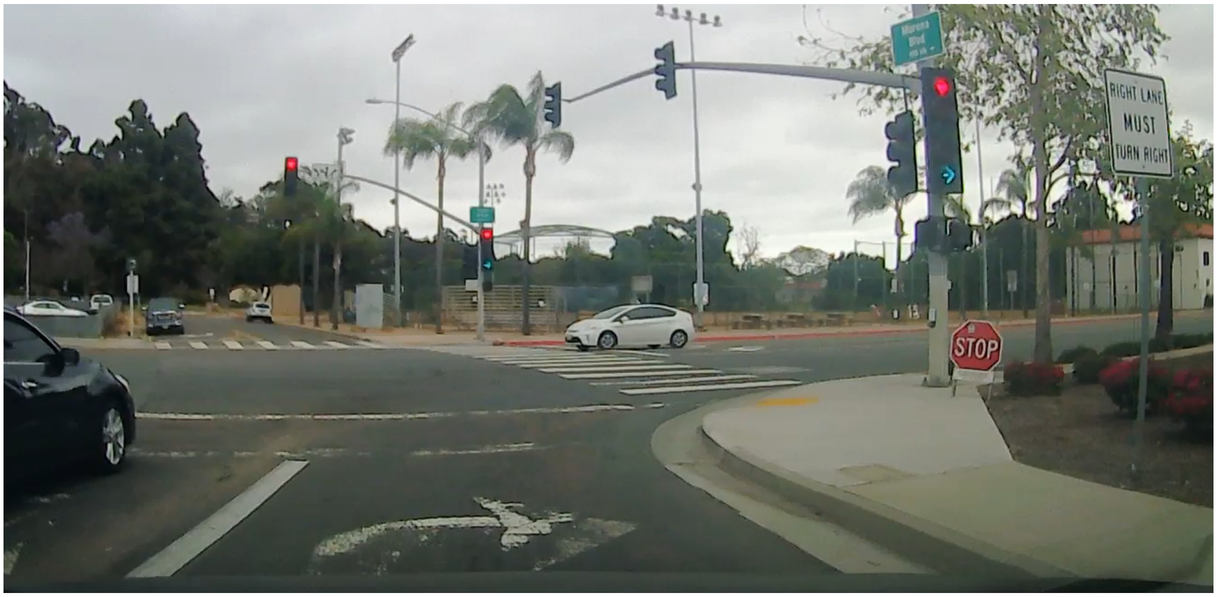


Figure 12.2. What should the driver do on their approach when faced with the installed STOP sign at right, but also a green turn arrow from the traffic light? How do humans assess and make decisions in such scenes, and can this reasoning be replicated in machine intelligence?

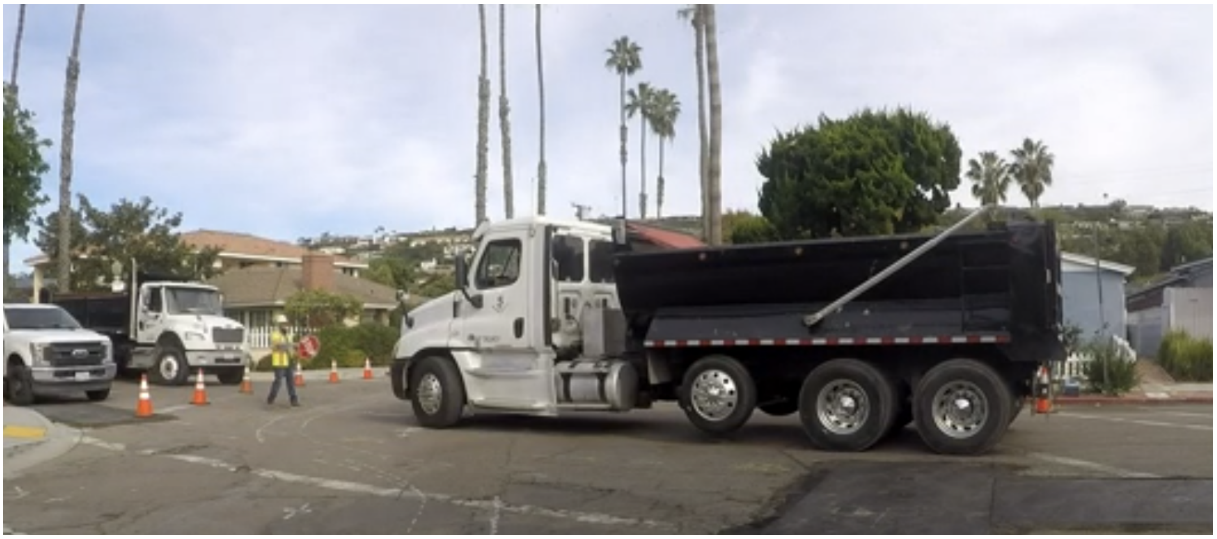


Figure 12.3. How can a machine compromise between the ambiguity of the presented obstacles and openings, presented signs, and gestures of scene agents? Knowing where to prioritize attention (in other words, ranking salience of instructive road objects) will be important for effective and safe autonomous driving.

gestures, scene infrastructure, surround agent interactions, and more; any case, even a frequently-visited street, can become a corner case when new modes of important and novel information are introduced to the scene.

My concluding remarks would not be complete without mention of my appreciation for the ways that human biology and cognition may be echoed in machine intelligence. I began my graduate studies with a fascination over the relationship between our visual system, learning, and convolutional neural networks[2], and through my research, I've become equally taken by the idea of human latent representations, our imaginations, and the process that connects what we see and think to our expressed verbalization. There are many different ways to represent information, each with their own benefits and losses, and humans have learned, planned, and made decisions from a multitude of representations which are now available in digital form for machine intelligence. While I would not make the claim that acting or thinking like a human is what defines intelligence, I would certainly consider systems which do act and think like humans

[2] Though now, especially after seeing the capabilities of the vision Transformer, realizing that the real fascination was more generally with the idea of digital images as representations of the world, and associated learning from visual information

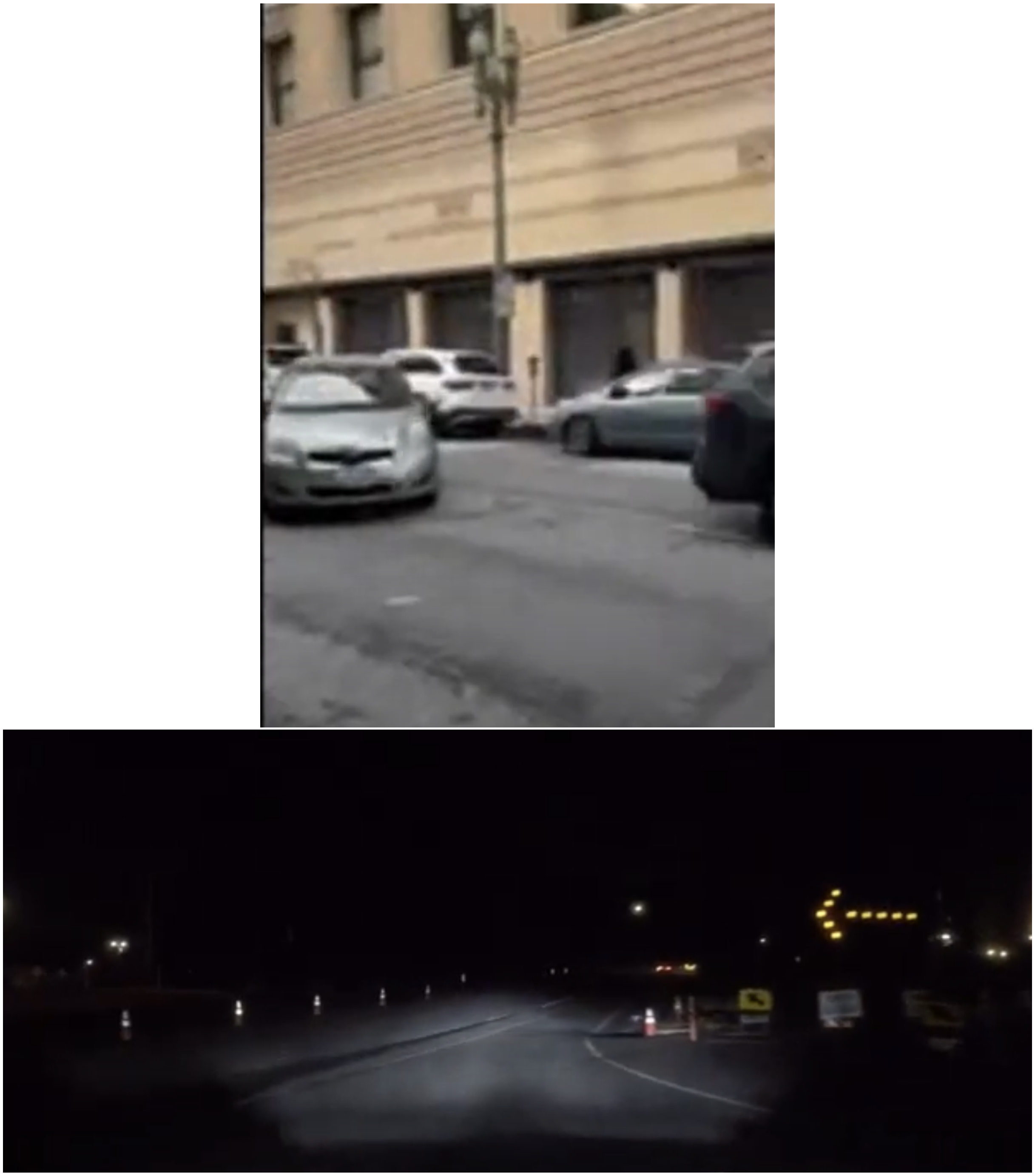


Figure 12.4. How can the vehicle make a decision when presented with conflicting information, such as the one-way street infrastructure, but with surrounding cars flowing in opposite direction, or signs directed the vehicle to enter the oncoming traffic lane? The above scene was encountered by a Waymo vehicle in Los Angeles in early 2024, when a protest caused part of a street to be closed without notice, and the below scene was encountered by the LISA-T testbed in La Jolla in early 2024.

to be intelligent and capable of complex tasks, and I do think the human cognitive lens may be a fruitful ***and explainable*** angle for continued exploration of autonomous systems in the open world.

Part VII

Appendix

Appendix A

Ensemble Learning for Fusion of Multiview Vision with Occlusion and Missing Information: Framework and Evaluations with Real-World Data and Applications in Driver Hand Activity Recognition

A.1 Introduction

Manual (hand-related) activity is a significant source of crash risk while driving; driver distraction contributes to around 65% of safety-critical events (crashes and near crashes) [344], and more than 3,000 deaths in 2022 [345].

Furthermore, given recent consumer adoption of early-stage autonomy in vehicles, driver hand activity has been shown to lead to various incidents even in these semi-autonomous vehicles. Drivers in vehicles supported by partial autonomy show high propensity to engage in distracting activities when supported by automation [346] and show increased likelihood of crashes or near-crashes when engaged in distracting activity [344]. Moreover, it is important to consider the manner of transitions when the driver must take manual control of semi-autonomous vehicles, as drivers demonstrate a slowness or inability to handle these control transitions safely when occupied with non driving-related tasks, often involving the hands [347] [3].

Accordingly, analysis of hand position and hand activity occupation is a useful component to understanding a driver's readiness to take control of a vehicle. Visual sensing through cameras provides a passive means of observing the hands, but its effectiveness varies depending on the camera location.

In this paper, we present a multi-camera sensing framework and machine learning solution, which we apply to the problem of robust driver state monitoring for autonomous driving safety. Our real-world, constrained application represents just one use case for this framework, as it can readily be extended to an ensemble of *N* domain-agnostic data sources and models for similar tasks; accordingly, we provide both a domain-specific and a generalized formulation of the sensing framework and learning problem in the following sections.

Consider an intelligent vehicle which classifies a driver's hand activity for a downstream safety application. By constraints imposed by vehicle manufacturing, we may have multiple cameras (in our case, four) which observe the driver from varying angles: head-on from the steering wheel, diagonally from the rearview mirror, diagonally from the dashboard, and peripherally from the central console. It is readily apparent that, depending on the driver's current position, there are instances where:

1. Only one of the four cameras has any view of the driver's hands, or

2. Multiple cameras have a view of the driver's hands.

An ideal intelligent system would recognize which of the cases is present, and in the former, choose to use the visible information to make an estimate, and in the latter, form an estimate made with the joint information of the multiple views which may be helpfully redundant (both cameras observing the fingers grasping the wheel, from multiple directions) or supplemental (one camera observes the fingers clasped to the wheel rim, the other observes the wrist resting on the wheel center) to the task at hand. This redundancy is closely related to the concept of *homogeneity* in [4]. Because this redundant or supplemental information can be present or absent between instances, we refer to this particular "missing data" phenomenon as *irregular redundancy*. Conceptually, this is similar to situations where data streams which operate under

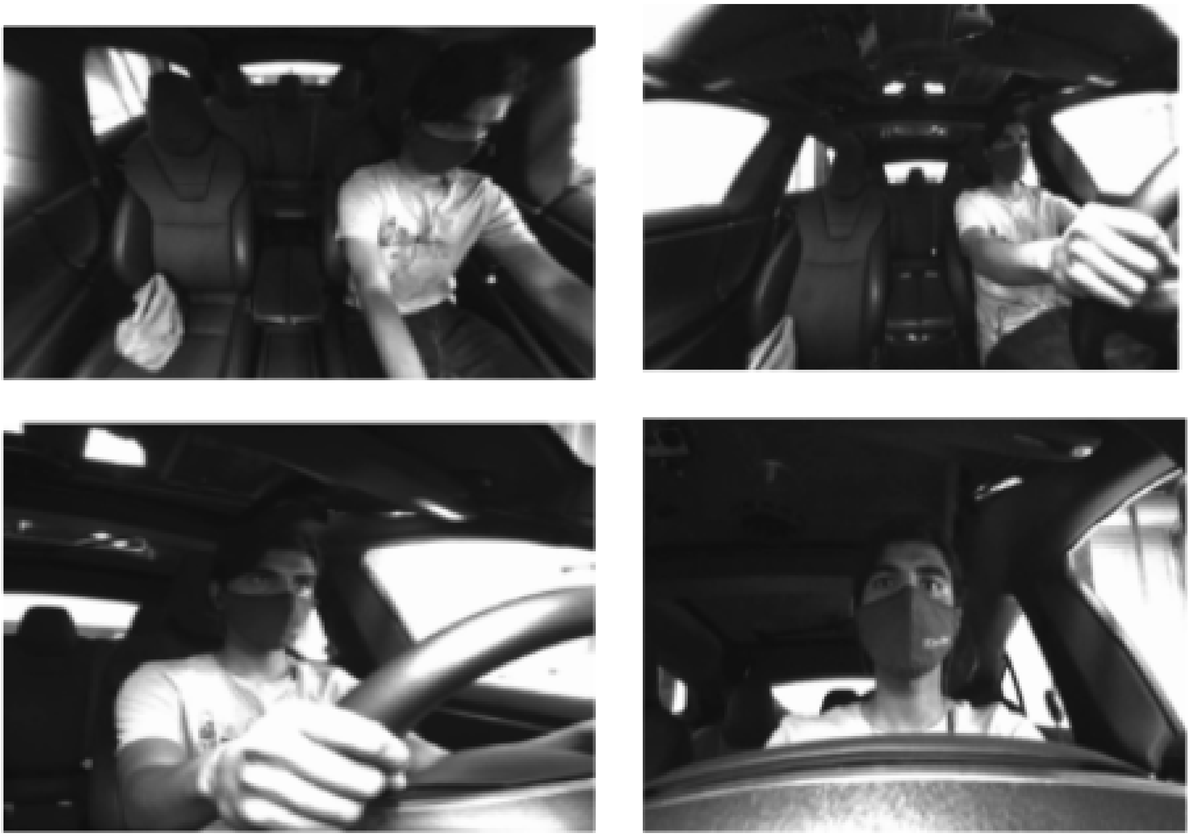


Figure A.1. Multi-view images (clockwise) - rearview, dashboard center, steering view, and dashboard driver, which explain how hands can be missing in certain frames, causing an *irregular redundancy*.

noisy conditions or at different sample rates are provided as input to a model which must provide output despite lost frames due to sampling rate or corruption.

More generally, we may describe this as a problem of *sensor fusion*, by which we must handle data to best leverage the accompanying noise, variance, and redundancy between samples to create an optimal estimate. In this theme, here we pose our framework as a system in which we have multiple data sources of the same event, and our goal is to learn an optimal model which accurately estimates a property of the event. Here, we are left with a few choices:

1. A model learned from one of these sources may tend to provide the best estimate, and we use only this source for future inference.

2. Models learned independently from each of these sources can each provide an independent estimate, and we can interpret their respective estimates to reach a group-informed consensus estimate.

3. A single model can be learned simultaneously between the sources, exploiting moments of redundancy and uncertainty in the data sources, such that the model provides an estimate with intelligence in selecting relevant features from data sources at any given instance dependent on the state of the other sources.

This question, described as the *multimodal reasoning* problem [4], is thoroughly investigated in the work of Seeland and Mäder [348], as will be discussed on the following section. Their analysis on multi-view classification utilizes datasets with complete data; here we seek to extend their work by answering a further question critical to real-world, real-time tasks: can models and learning paradigms generalize to cases of multiple data sources when significant data is missing?

This problem of missing, corrupted, or asynchronized multi-modal data is found in many domains, ranging from biomedical imaging modalities like photoacoustic and computed tomography and optical microscopy [349] to autonomous systems dealing with temporallycalibrated LiDAR, vision, and radar [350] and identification of crop disease from satellite imagery with significantly different capture frequencies or resolutions [351].

In our analysis, we examine “best-of-*N*" performance from collections of *N* independent models, as well as schemes which negotiate between the logits of *N* independent models, and a model which learns between hidden features derived from the *N* data sources jointly, known as *late fusion*. We adopt the term “ensemble" to refer to the *N* models respectively learned from the *N* data sources, which may be combined to generate a prediction. Critically, under our condition of irregular redundancy, the number of views available varies between instances, thus requiring the introduction of our method for multi-view ensemble learning with missing data. Further, because there are multiple simultaneous tasks involved in driver monitoring, we examine the *task relevance* of each modality [4] in our analysis.

To summarize our contributions, we (1) perform comparative analysis between single-view, ensemble voting-based, and late-fusion learning on data from four real-world, continuous-estimation safety tasks, using sensors operating with irregular redundancy, (2) provide a generalized formulation of the real-world, real-time multi-modal problem such that our methods can be applied to similar tasks in both autonomous driving and other domains, and (3) evaluate the performance of these models with respect to human-centric safety systems by examining task performance on human drivers outside of the training datasets.

A.2 Related Research

A.2.1 Sensor Fusion

Sensor fusion describes integration of data from multiple sensor sources, like LiDAR or cameras, towards a task. In the intelligent vehicles domain, research in methods of combination of output from multiple sensors to improve tasks in prediction and estimation is well-established. Chen et al. designed a Multi-Vew 3D Network (MV3D) that fuses LiDAR point cloud and RGB image data to perform 3D Object Detection in autonomous driving scnearios [356]. Their deep

Table A.1. Multiview Fusion Methods. Homogeneity (from [4]) refers to the extent that the abstract information presented in one view is equivalent to the information presented in another, toward the intended task(s). High homogeneity is highly redundant, while medium homogeneity refers to cases where some combinations of views may have the same information, but this information may be exluded from other views. Low homogeneity refers to situations where information between views is primarily supplemental. Our presented method is notable in having only medium homogeniety in support of its task, and frequent appearance of incomplete sets.

Method	Modalities	Tasks	Homogeneity	Incompleteness
Various Fusions [348]	2-5 RGB	1	High	None
Late Fusion [352]	4 IR + 1 RGB	1	High	None
Temporal Score Fusion [353]	3 RGB	1	High	None
Late Fusion [354]	5 RGB	1	Medium	None
Slice Fusion [355]	Depth Slices	1	Low	None
Ours	4 IR	4	Medium	Frequent

fusion of camera and LiDAR data uses FractalNet, a CNN architecture that is an alternative to other state-of-the-art CNNs like ResNet [357]. Similarly, Liang et al. fuse LiDAR and image feature maps into using a continuous convolution fusion layer [358]. This fusion process creates a birds-eye-view (BEV) feature map that is fed into a 3D Object Detection Model. Pointpainting is another prominent example of a fusion process of LiDAR and image data [359]. Pointpainting takes image data and performs semantic segmentation to compactly summarize the features of the image. To fuse the LiDAR and image data, the LiDAR data is projected onto the semantic segmentation output. In all these methods, the sensors are LiDARs and cameras. There is a different class of work which aims to do sensor fusion using the same modality or sensor type, but with data collected from different sensors or sensor views, like [360], which combines LiDAR point clouds from the birds-eye-view and perspective view to learn fused features. In our work, we learn image features fused from different camera views. The features can be combined at different stages in the network giving rise to different ensembles, as explained in the next section.

A.2.2 Ensemble Learning

In addition to fusion of output from N sensors to reduce uncertainty of the observed information, we also explore ways that the models learned from the input of these N sensors can share information during the learning process, such that the collective ensemble is optimized to the task.

We present here the Sagi and Rokach's survey definition of Ensemble Learning:

> “Ensemble learning is an umbrella term for methods that combine multiple inducers to make a decision,...The main premise of ensemble learning is that by combining multiple models, the errors of a single inducer will likely be compensated by other inducers, and as a result, the overall prediction performance of the ensemble would be better than that of a single inducer." [361]

It is worth noting that there are a variety of methods for generating such ensembles [361]. For example, the high-level learning system may:

1. Vary the training data provided to the inducers [362] [363] [364] [365],
2. Vary the model architecture between inducers [366] [367] [368],
3. Vary the learning methodology [369] or hyper-parameters [370] between inducers, or
4. Vary some combination of the above between inducers.

In this research, we freeze the model architecture, learning methodology, and hyper-parameters; we vary only the training data provided to the inducers. However, in this case, the training data is not sampled or refactored from some shared pool; instead, each ensemble member has access to its own set of training data. These training data are not independent, though; the training data is unified as a *collection* per instance, where each member of the collection is a different representation of the same base observation (e.g. different cameras taking simultaneous photos of the same object).

From the ensemble of inducers, inductions can be combined and learned-from in a variety of ways:

Bayesian model averaging and combination

Bayesian Model Averaging (BMA) allows formation of predictions with many candidate models without losing information like an all-or-none technique. Using Bayesian model averaging, the probability of a prediction y given training data D can be defined as:

$$p(y|D) = \sum_{k=1}^{K} p(f_k|D)p(y|f_k,D) \tag{A.1}$$

where f_k is the prediction of the kth model. The posterior probabilities $p(f_k|D)$ can be treated as weights w_k for each of the separate models since $\sum_{k=1}^{K} p(f_k|D) = 1$. Previously researched applications using these methods include weather forecasting [371], flood insurance rate maps [372], risk assessment in debris flow [373], and crop management [374].

As mentioned by Monteith et al., Bayesian Model Averaging can be more thought of as a model selection algorithm, as ultimately the importance of each model is determined by the posterior probability weight [375]. To develop an approach that is more inherent in ensemble learning, there are various strategies that can be used for model combination rather than model averaging. Bayesian model combination has found success in reinforcement learning by combining multiple expert models [376], speech recognition [377], and other tasks (notably functioning on non-probabilistic models and combinations of models observing different datasets [378]).

Voting, Weighted Majority Algorithm

In ensemble learning, it is crucial to learn the value of each individual model by assigning different weights to these models in order to increase performance. In this algorithm, weighted votes are collected from each model of the ensemble. Then, a class prediction is made based on which prediction has the highest vote. All models which made incorrect predictions will be discounted by a factor β where $0 < \beta < 1$ [379]. Weighted majority algorithms have been used to combine model predictions to identify power quality disturbances for a hydrogen energy-based microgrid [380], calendar scheduling [381], profitability and pricing [382], and other applications. The weighted majority algorithm is used to combine the predictions from the expert models and see how effective each expert model is.

In addition to single-view results, we compare performance in our hand classification task under naive voting, Bayesian model combination, and weighted majority voting.

A.2.3 Machine Learning from Multiple Cameras

Seeland and Mäder thoroughly investigate image classification performance gains afforded by network fusion at different process levels (early, late, and score-based) when using multiple views of an object [348]. They apply their methodology to datasets comprising cars (shot from 5 views), plants (shot from 2 views), and ants (shot from 3 views). They find that late fusion provides the strongest performance gain for the car and plant datasets, and that an early fusion (slightly misnamed in this case, as it occurs at the final convolution) leads to a very marginal gain compared to late fusion on the ant dataset. In general, the authors results support late fusion as the dominant methodology, with early fusion often leading to worse performance compared to baseline. In their research, each data instance is referred to as a “collection" (i.e. collection of N images). Critically, each collection analyzed is *complete*; that is, no view is missing from any given collection. This is where our problem framework and approach differs; as illustrated in Figure A.2, we consider situations in which collections may be incomplete, and seek to learn correct labels despite missing data.

The late fusion approach for visual patterns has found success in multiple application domains, as presented in Table A.1, and even in other domains such as cross-modal information retrieval [383] [384] [385] [386].

A.2.4 Driver Hand Activity Classification

Shortcomings of Non-Camera Methods

At the time of writing, most commercial in-vehicle systems which monitor the driver’s hands use pressure and torque sensors embedded in the steering wheel to detect the presence of a driver’s hands. However, this method of sensing leaves multiple safety vulnerabilities:

1. Especially in the case of non-capacitative sensing, these sensors can be spoofed by placing

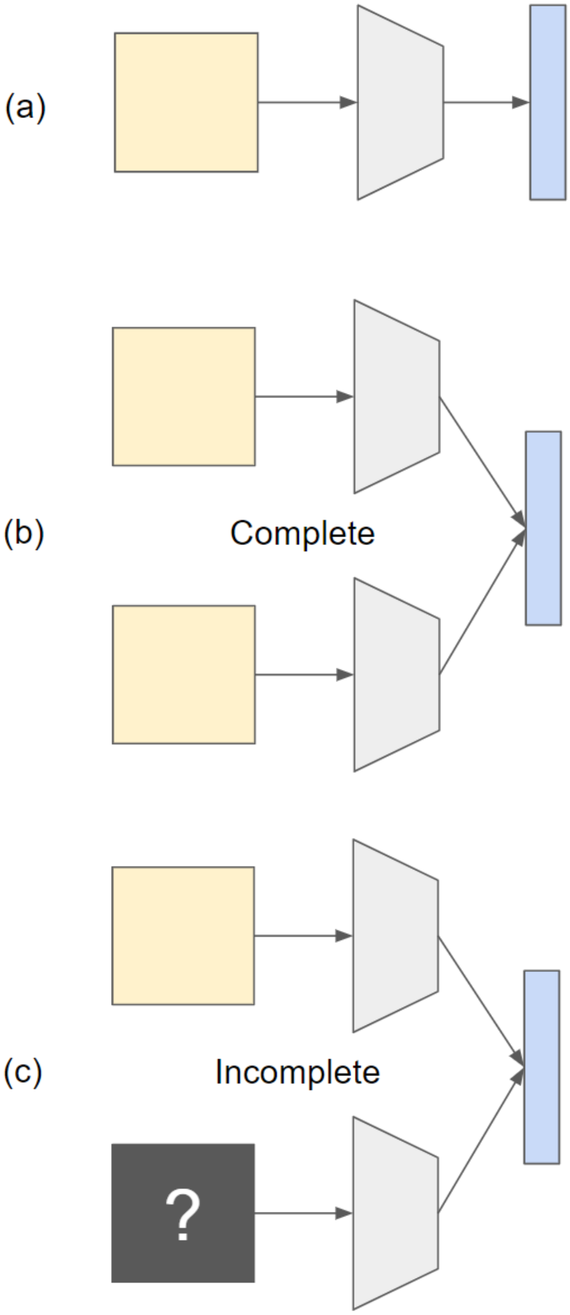


Figure A.2. While traditional learning problems (a) may seek to learn a model (gray) which makes a prediction (blue) from input (yellow), in the multi-modal setting (b), we seek to learn a model which makes a prediction from multiple inputs. However, in the case where a sensor fails, becomes occluded, or operates at a different rate, the input set goes from *complete* to *incomplete*. In this research, we explore techniques for dealing with such incomplete sets (c), important for systems which are relied upon for always-online output prediction.

weighted objects on the wheel, leading to recent fatal accidents.

2. When effective for determining if the hands are on or off the wheel, these sensors still cannot distinguish between different hand activities taking place off of the wheel, and recognizing these activities is critical for estimating important metrics like driver readiness and take-over time. Hand locations and held objects imply hand activities, crucial to inferring a driver's state, and this information is lost when reduced to hands-on-wheel and hands-off-wheel.

Camera Methods

Camera-based methods of driver hand analysis allow for observation of the hands without steering wheel engagement. Past systems for classification of driver activity and identification of driver distraction use traditional machine learning approaches; for instance, Ohn-Bar et al. demonstrated systems that utilize both static and dynamic hand activity cues in order to classify activity in one of three regions [387] and extracts various hand cues in ROI and fuses them using an SVM classifier [244]. Borgi et al. use infrared steering wheel images to detect hands using a histogram-based algorithm [388]. More recent works expand on the aforementioned classifiers and utilize deep learning in order to identify and classify driver distraction in a more robust manner. Eraqi et al., among others, have developed systems that operate in real time to identify driver distraction in a CNN-based localization method [389]. Shahverdy et al. also use a CNN-based system in order to differentiate between driving styles (normal, aggressive, etc.) in order to alert the driver accordingly [390]. Building on this, Weyers et al. demonstrate a system for driver activity recognition based on analysis of key body points of the driver and a recurrent neural network [391], and Yang et al. further demonstrate a spatial and temporal-stream based CNN to classify a driver's activity and the object/device causing driver distraction [392]. A comprehensive survey outlining the current driver behavior analysis using in-vehicle cameras was done by Wang et al [393].

Recent pose detection models provide another helpful tool in understanding the hands of the driver. As defined by Dang et al., 2D pose detection involves detecting important human body parts from images or videos [394]. Chen et al. describes that there are three ways to define human poses: skeleton-based models, contour-based models, and 3D-based volume models [395]. Our research uses a skeleton based-model, which describes the human body by identifying locations of joints of the body through 2D-coordinates. A deep learning approach to pose detection through a skeleton based-model is to first detect the human location through object detection models like Faster-RCNN and then perform pose estimation on a cropped version of the human. Some successful approaches to pose detection include HRNet, which is successful for pose detection problems since it maintains high-resolution representations of the input image throughout a deep convolutional neural network [396]. Toshev and Szegedy perform this pose estimation by implementing the model DeepPose, which refines initial joint predictions via a Deep Neural Network regressor using higher resolution sub images [397]. Yang et al. design a Pyramid Residual Model for pose estimation which learns convolutional filters on various scales from input features [398].

Though consumer and commercial vehicles have begun integrating inside-facing cameras for a variety of tasks, such as attention monitoring and distraction alerts, these methods are not without their own challenges. A single camera may be well-suited to a particular task, but different situations may call for different camera placements. While one view may be ideal for a particular task within design constraints, this view may sacrifice a complete view of a different driver aspect and may not offer redundancies if a camera is obstructed or blocked. For example, an ideal hand view (taken from above the driver) would not be suitable for assessing a driver's eyegaze, but a camera that can see the driver's eyes may also have at least a partial view of the driver's hands.

Safety and Advanced Driver Assistance Systems

Recent works in safety and advanced driver assistance systems utilize deep learning techniques in order to perform driver analysis. In particular, deep learning allows researchers to extract driver state information and determine if they are distracted through analyzing driver characteristics such as eye-gaze, hand activity, or posture [393]. Estimating driver readiness is another vital aspect to safe partial autonomy, and a key component to understanding driver readiness is hand activity, as a distracted driver often has their hands off the wheel or on other devices like a phone. Illustrated in Figure A.4, Rangesh et al. [3] and Deo & Trivedi [199] show that driver hand activity is the most important component of models for prediction of driver readiness and takeover time, two metrics critical to safe control transitions in autonomous vehicles [28] [27]. Such driver-monitoring models take hand activity classes and held-object classes as input, among other components, as illustrated in Figure A.3. These classes can be inferred from models such as HandyNet [239] and Part Affinity Fields [238], using individual frames of a single camera view as input. Critically, this view is taken to be above the driver, centered in the cabin and directed towards the lap– a typically unobstructed view of the hands.

Application of multi-view and multi-modal learning to safe, intelligent vehicles ([399], [217]) brings two benefits: increased flexibility in field-of-view for individual component cameras, and increased accuracy in classification for observable activity. Both benefits arise from the ability of the system to reason between views, allowing occluded or otherwise compromised images from one view to be substantiated by images from additional views in cooperation.

A.3 Methods

The general hand activity inference stage is organized in four steps: multi-view capture, pose extraction, hand cropping, and CNN-based classification.

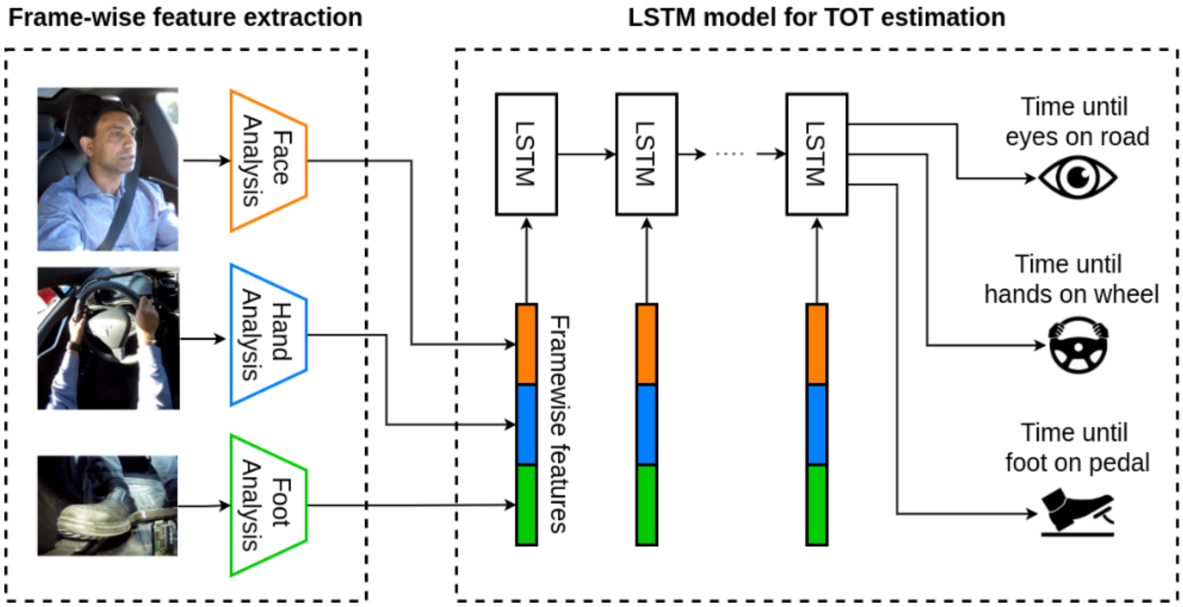


Figure A.3. As illustrated in [2], analyzing logits of hand activity and location classes play a useful role in predicting a driver's readiness to take control of a vehicle.

A.3.1 Pre-processing Steps

Feature Extraction: Pose and Hands

The inference stage is illustrated in Figure A.5. Following data capture, we extract the pose of the driver in each frame, where "pose" is a collection of 2D keypoint coordinates associated with the driver's body, such as the wrists, elbows, shoulders, eyes, etc. This problem is broken into two steps: first, we must detect the driver in the frame, then detect the driver's pose. Each step requires its own neural network; for driver detection, we first use the Faster-RCNN [400] model with Feature Pyramid Networks [401], using a ResNet-50 backbone [402] to detect the driver. We note that this network will output any humans detected in the frame, so we apply a post processing step (based on the camera view) to only include detections corresponding to the driver's seat. For joint detection, we employ the HRNet [403], a robust top-down pose detection model,which predicts 2D coordinates of various points of the body such as the wrists, elbows, shoulders, eyes, etc. The results of driver and keypoint detection are illustrated in Figure A.7.

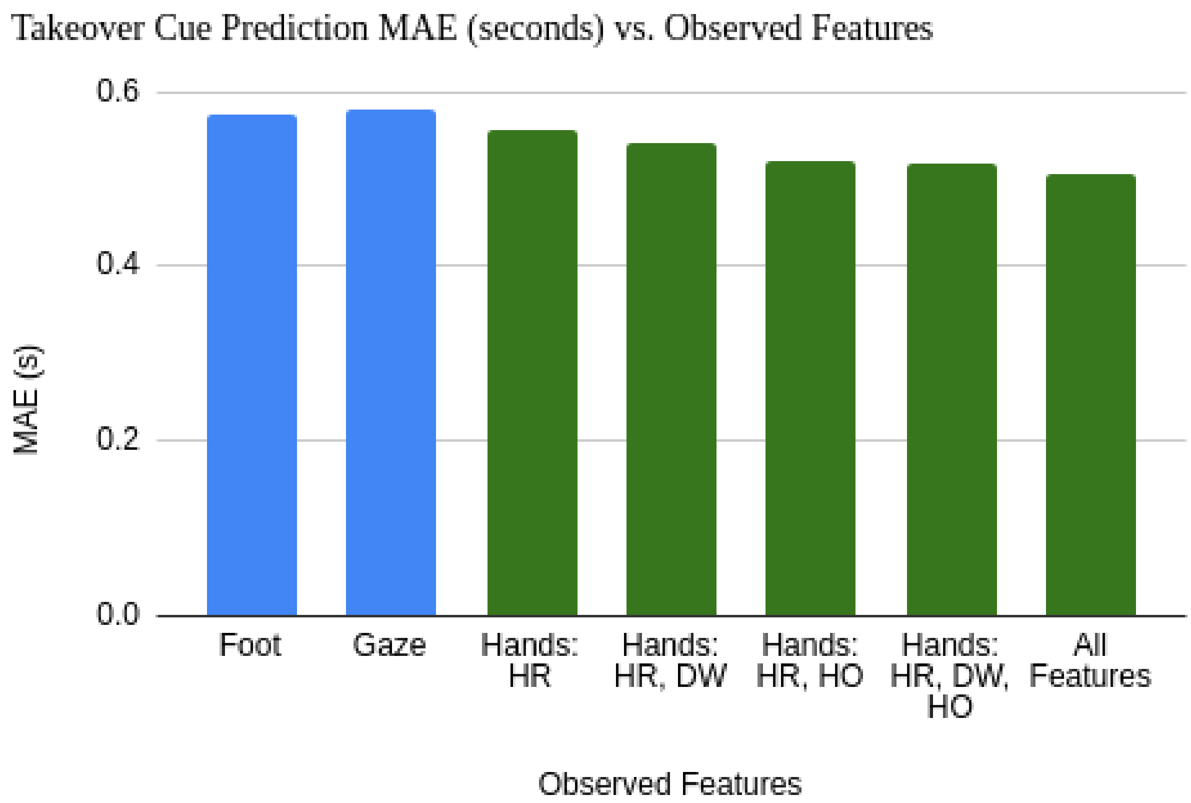


Figure A.4. In an ablative study, Rangesh et al. [3] show that various individual features and combinations of features associated with the hands, including hand region (HR), distance to wheel (DW), and held object (HO) are most informative to models for predicting cues associated with vehicle takeovers from automated to manual control. In the case of control transitions, these fractional-second gains are critical for a driver's reflexes to safety alerts.

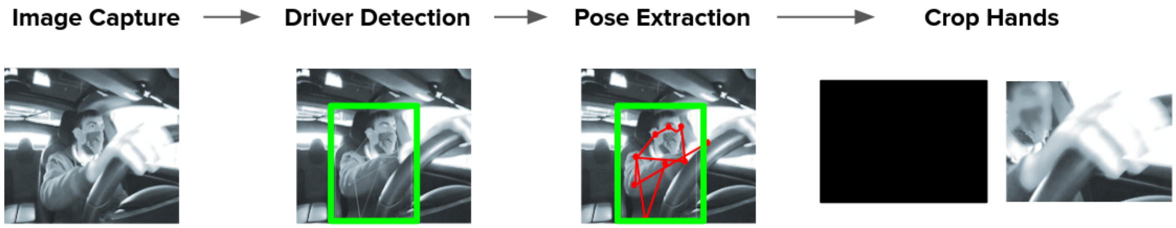


Figure A.5. The image preprocessing pipeline prior to learning involves four steps, carried out individually from each camera stream. First, the image is captured, then, the driver is detected and their pose extracted, allowing for crops around the hands to be generated. In this example, because the left hand is not visible to the particular camera, the method of single imputation is used to replace the frame with a frame of zeros. We note that because the method uses only the image of the hands towards its learning, it is possible to anonymize the driver by blurring the face, as we have done in the above example, for the cropped frames that serve as model input.

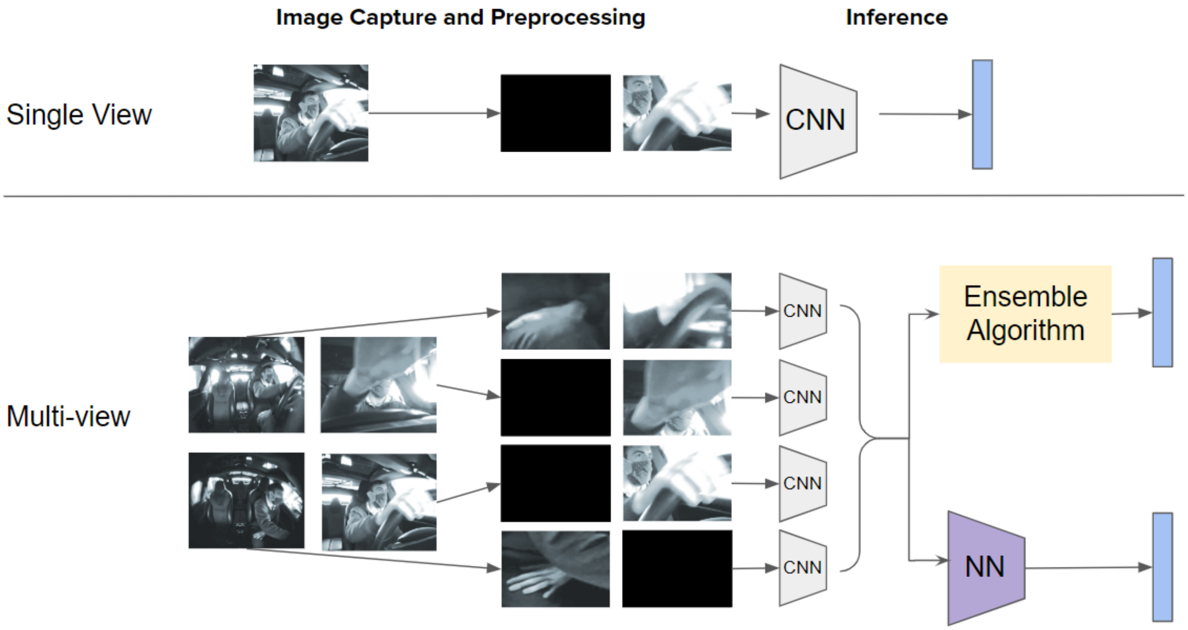


Figure A.6. Classification pipeline. Following image capture, we perform image processing to detect the driver using Faster-RCNN with Feature Pyramid Networks (FPN) with a ResNet-50 backbone, extract the driver pose using HR-Net, and crop the hands 100px from center of wrist joints. In the Inference stage, we utilize CNNs for classification, beginning from a pre-trained ResNet fine-tuned on our dataset. For the single view model, we make direct inference, and for the multi-view models, we pass the logits to ensemble algorithms, or pass the CNN-output feature maps to a neural network for late fusion. In our experiments, we use Bayesian Combination and Weighted Majority Averaging as the Ensemble Learning algorithms, and Late Fusion via fully-connected neural network laters.

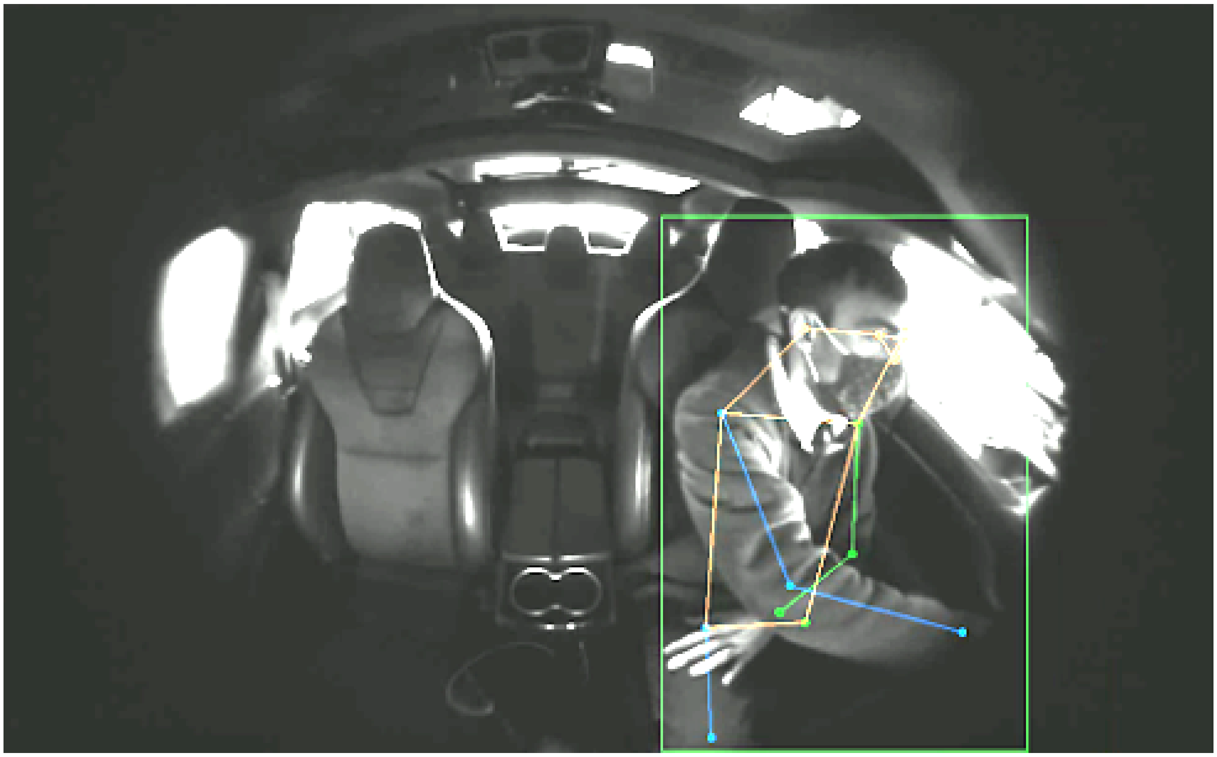


Figure A.7. Prior to classifying driver hand activity, the system must detect the driver. We use Faster-RCNN to generate the bounding box shown in green. Following driver detection, we apply HRNet to identify the 2D pose skeleton, shown as keypoints and connecting lines on the driver's body.

Hyperparameter Selection: Hand Crop Dimensions

We crop images around each of the hands, centered at the wrist and extending 100 pixels in each direction. The width of the crop is a hyperparameter which can be changed to add or reduce spatial context. Only these hand crops are fed into the activity classification pipelines.

A.3.2 Single-View Models

The cropped images from a particular camera are classified by two convolutional neural networks trained on images of that view. One network outputs probabilities that the hands are holding one of three objects: Phone, Beverage, Tablet; or holding nothing. The second network (identical in architecture to the first, except for number of classes) predicts the probability that the hand is in one of five hand location classes: Steering Wheel, Lap, Air, Radio, or Cupholder. The classes Radio and Cupholder are reserved for the right hand only. For single-view model evaluation, the network infers the hands to be classified according to the class of maximal probability.

In cases where there is no image available, the model is provided an image of proper dimension containing only the value 0. This is a variation of the method referred to as single-imputation [349], in which a single value is used to replace any instances of missing data. The intention behind this decision is that the network will learn a prior over the training data in situations when the view is occluded; that is, each time a blank image is presented, it infers that the sample should be classified in one of the typically occluded positions, with probability representative of the distribution of the training data.

A.3.3 Naive Voting

In the naive voting scheme, all four single-view models make a prediction using their respective image from a given collection, noting that up to $N-1$ images may be blank. The prediction made by the network is taken to be

$$y = \arg\max_i \left(\sum_{j=1}^{N} \frac{1}{N} p_{ij}\right), \tag{A.2}$$

where M is the number of classes, N the number of models, and p_{ij} the probability of the ith class from the jth model. This method gives each model equal vote.

A.3.4 Weighted Majority Voting

Using Weighted Majority Voting, we seek to combine the decisions of the 4 models weighted by a discount factor d_i. This discount factor is based on the number of mistakes m_i made by the model during validation:

$$d_i = 1 - \frac{m_i}{\sum\limits_{i=1}^{N} m_i} \tag{A.3}$$

Then, each collection prediction is made using

$$y = \arg\max_i \left(\sum_{i=1}^{N} d_i p_i\right). \tag{A.4}$$

A.3.5 Bayesian Model Combination

Using Bayesian Model Combination, we combine the decisions of the 4 models weighted by a factor representing the likelihood of the particular model given the observed data, $P_i \sim p(f_i|D)$. In cases where the hands are not detected in a certain view i, then we consider model f_i to have low likelihood; therefore, we set P_i to zero in such situations. If n models have P_i as zero, then the P_i of remaining models is distributed uniformly as

$$P_i = \frac{1}{N-n} \tag{A.5}$$

where N is total number of views.

$$y = \arg\max_i (\sum_{i=1}^{N} P_i p_i) \tag{A.6}$$

A.3.6 Multi-view Late Fusion

For the late fusion scheme, we use a neural network architecture composed of four parallel sets of convolutional layers (ResNet-50 backbones), which act on each of the four image views. Following the convolutional layers, each parallel track is fed to its own fully-connected layer of 512 nodes (followed by a ReLU activation). These layers are joined together by a fully-connected layer with 2048 nodes (followed by a softmax activation); this is the point of fusion, where the features extracted from the four views are combined and the relationships between the multiple views are learned.

We call this late fusion as it is done at the penultimate layer, late in the pipeline. This was done to make sure the fusion could leverage high level features present deeper in the pipeline. We use two fused models as before; one which outputs probabilities that the hand is holding one of the 3 objects and another which outputs the location of the hand. The maximal probability class is chosen as the classification output.

A.4 Experimental Evaluation

A.4.1 Comparison of Single-View and Ensemble Techniques

Using four cameras, we collect a dataset of 19 subjects engaged in various hand placements and object-related activities.

Altogether, we collect approximately 81,000 frames corresponding to hand zone activity, and 128,000 frames corresponding to held object activity. We divide these into training, validation, and test sets using approximately 80%, 10%, and 10% of the data respectively (with marginal differences to account for dropped frames). The distribution of the data between views

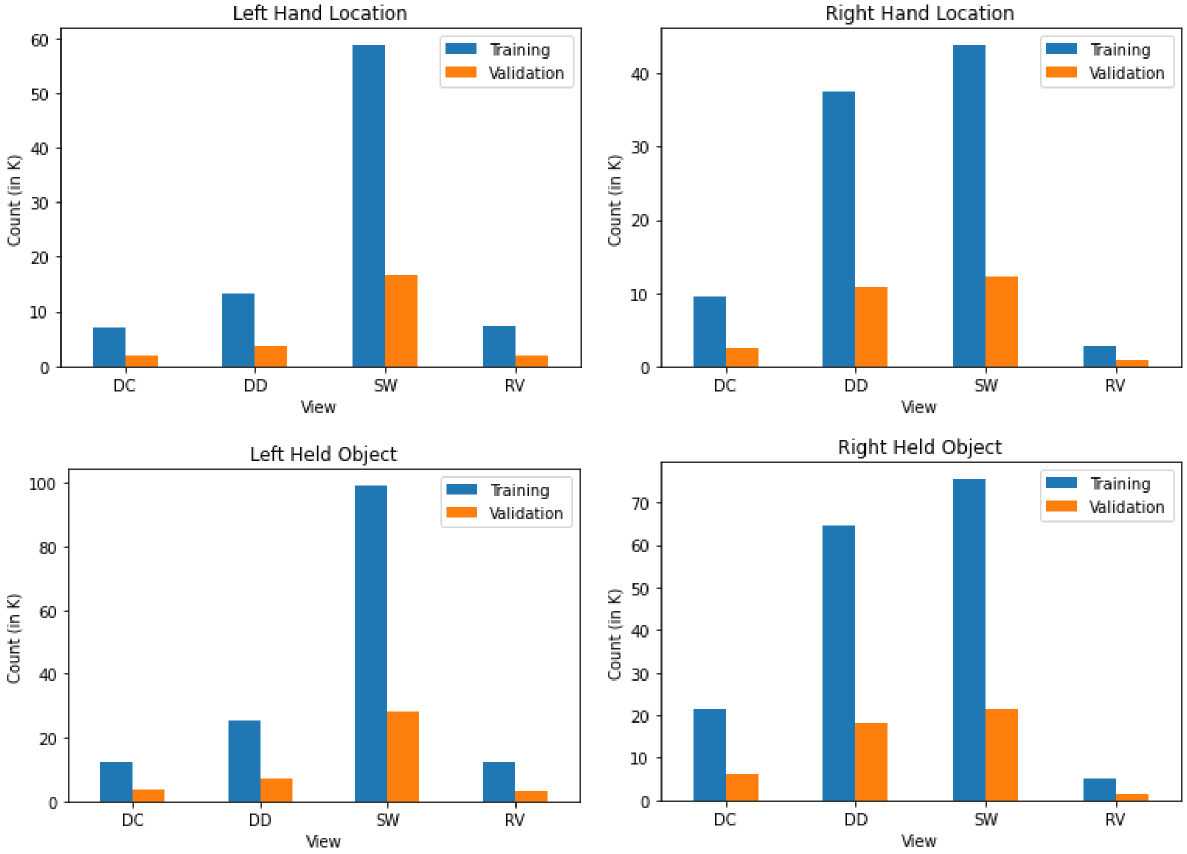


Figure A.8. Distribution of collected samples for locations (Left and Right hand) and held objects (Left and Right hand) from 4 views Dashboard Driver (DD), Dashboard Center (DC), Steering Wheel (SW) and Rear View (RV)

is shown in Figure A.8. We note that the challenges of selecting camera views for this task are readily apparent in the proportions of the data; one camera view (rearview) has significantly less frames where the pose is reliably estimated, while the steering wheel view has many. However, the availability of frames does not necessarily correspond to the ability of that view to be informative to the task at hand nor generalizability to other tasks in the autonomous driving domain.

Using this data, we trained the above-described neural networks for hand location and held object classification into the defined zones (3 location zones for the left hand, 5 location zones for the right hand, and 4 held objects [including null] for each hand).

Table A.2. Classification accuracies (averaged across all classes) of baseline single-camera views. The rows represent, for each task, the performance of the best-performing and worst-performing of the *N* camera view models, as well as the average performance across views.

Method	LH Location	RH Location	LH Held Object	RH Held Object
Worst-of-N	0.442	0.212	0.322	0.289
Average-of-N	0.593	0.458	0.513	0.581
Best-of-N	0.952	0.785	0.952	0.836

Single-View Models

We evaluate the four single view models on images from the test set, including blank images when no image is available. From this, we compute an average model accuracy by taking the average of each per-class accuracy for each of the four tasks (left hand location, right hand location, left hand held object, right hand held object). For each task, we report the performance of the best-performing model, the worst-performing model, and the average across the four models. This highlights foremost the importance of camera view selection for this particular task, but also provides a point of comparison to see how the ensemble learning and fusion methods may enhance the overall performance of the models to their task. Results are provided in Table A.2.

Ensemble Methods: Naive Voting, Weighted Majority Voting, Bayesian Model Combination, and Multi-View Late Fusion

We evaluate the four methods described in the Methods section, as well as an additional method which employs both Weighted Majority Voting and Bayesian Model Combination simultaneously. We evaluate the performance of these models on two different sets: first, only on collections which have all *N* images available, and second, on collections with any number of images (1 to *N*) available. Results are provided in Tables A.4 and A.5.

Importantly, only a very small fraction (less than 3%) of each of our task test set collections are complete, as shown in Table A.3. In fact, some task classes are never simultaneously

Table A.3. Test set size for different tasks, and percentage of test set collections which are complete.

Task	Test Set Size	% Complete
LH Location	9,193	2.43%
RH Location	9,205	1.67%
LH Held Object	15,486	2.59%
RH Held Object	14,624	2.22%

Table A.4. Classification accuracies (averaged across all classes) of different ensemble methods on four hand classification tasks, evaluated only when all *N* views are available. In this “complete-view-only" test set, 2 classes from left hand location, 4 from right hand location, and 1 each from left and right hand held object are completely unrepresented. Performance on the held object tasks may be poor due to the uncertainty in less-informative views bringing down the overall confidence of the system towards the correct class (or artificially raising confidence in the incorrect class). Naive voting may outperform weighted majority voting when challenging examples found in the validation set may be unrepresented in this test set, thereby discounting models which would otherwise be “correct". This table also serves to illustrate how often frames are missing in these tasks, demonstrating the importance of a method which is robust to missing data.

Method	LH Location	RH Location	LH Object	RH Object
Naive Voting	1.000	0.981	0.509	0.426
Weighted Majority Voting	0.991	0.987	0.403	0.410
Late Fusion	1.000	1.000	0.978	0.995

observed from all views, so results in Table A.4 indicate performance on a limited number of classes from the actual task at hand, and at that, only for complete collections! While the models may be great at making inference when they have a clear view of the object of interest, this suggests a significant performance gap for a safety system expected to make continuous inference across all classes, not just inference when data is complete. By contrast, Table A.5 represents performance across *every* sample of the test set. We include both tables to illustrate the point that while the voting-based methods begin to fail, the late fusion method performs just as well even when data is missing from a collection.

Table A.5. Classification accuracies (averaged across all classes) of different ensemble methods on four hand classification tasks, with 1 to *N* views available in each collection.

Method	LH Location	RH Location	LH Object	RH Object
Naive Voting	0.470	0.205	0.277	0.334
Weighted Majority Voting	0.443	0.201	0.269	0.316
Bayesian Model Combination	0.397	0.338	0.366	0.360
WMV+BMC	0.398	0.338	0.358	0.363
Late Fusion	**0.991**	**0.988**	**0.978**	**0.986**

Our original question was: can these methods overcome situations where data is missing from a collection? Table A.5 provides our answer. When data is missing, the voting-based methods struggle significantly due to the falsely-placed confidence given to the model output. The largest challenge with these approaches is recognizing which view is dominantly correct in a particular situation and leveraging that view appropriately; otherwise, too much weight may be given to a model which has false confidence, and a model's vote may be a reflection of the intrinsic difficulty of that particular view. Able to better leverage information between views, the best performance comes from the multi-view late fusion approach. The late fusion model both (1) maintains near-perfect performance on the four tasks, even when 1 to $N-1$ frames are missing from the collection, and (2) exceeds performance of all single-view cameras for each task. These two results suggest that a late-fusion model is successfully learning complementary information that is unavailable in a single-view; that is, the model is effective in combining different sources of information to make a better-informed prediction on the task. Additionally, it is able to do so despite missing data, suggesting that the model has learned to leverage remaining sources of information when frames are dropped.

In prior work, Greer et al. [404] show that multi-view late fusion models give superior results over single-view models because the network can learn from more perspectives. Late fusion is particularly effective as all the camera views have high-level richer features deeper in the pipeline. The multi-view late fusion model was successful in classifying zones and objects when the training and test subjects were same. But in real-world scenarios, models need to generalize to unseen subjects. We elaborate on our approach for evaluating performance on cross-subject classifications in the next section, and provide recommendations for such systems in the following discussion.

A.4.2 Multiple Subject Validation: Generalizing to Unseen Drivers

In the first set of experiments, we show that multi-view late fusion models give superior results over single-view models because the network can learn from more perspectives. Late fusion is particularly effective as all the camera views have high-level richer features deeper in the pipeline. The multi-view late fusion model was successful in classifying zones and objects when the training and test subjects were same. But in real-world scenarios, models need to generalize to unseen subjects. Here, we evaluate performance on cross-subject classifications, and provide recommendations for such systems in the following discussion.

Greer et al. [404] evaluate the late-fusion model performance on a substantial set of test data derived from the same capture system and subjects as the training and validation data, but in intelligent vehicle applications, it may be impractical to collect training data on each individual driver. An ideal model would generalize to all drivers that may use the vehicle.

A typical risk in end-to-end learning on overparameterized systems involving human subjects is that such a deep neural network is not typically “explainable" [405] [383]. When the model learns from humans, it can overfit to particular features associated with an individual subject, rather than learning actual patterns of interest (e.g. the model becomes really good at learning how to recognize Subject A’s hand holding Subject A’s cell phone, rather than a more general prototype of any hand holding any cell phone).

Machine learning models are commonly evaluated using k-fold cross validation, but this evaluation has shortcomings when data from the same subjects are contained in both train and validation sets, since (as described above) the model can overfit to the subject’s unique signature instead of the latent activity. Accordingly, techniques of subject cross validation are preferred [406]. In typical k-fold cross validation, data is divided into k sets, and each of these k sets have a turn being left out of the training process (used only for evaluation). The summary statistics to describe the goodness of the model is then the average model performance on the k validation sets.

In our case, we utilize a dataset of 19 subjects. Here, we discuss evaluation choices made on splitting the data. We first constrain evaluation such that any subject being used in validation is unseen during training. We note that early stopping is controlled by a subset of the training data to prevent significant overfitting; while it may be beneficial to let yet another unseen subject (or subjects) determine the training stop-point, this introduces the bias of model performance to that particular driver (or drivers), which will not necessarily translate to performance on the unseen evaluation driver.

We use varying values of k on each task to handle computational constraints, rotating a left-out subject from each of the k model trainings for each task. For each model, we take the average accuracy among all of the classification categories (the so-called *macro-averaged precision*), and then average this value among the k models. We evaluate using $k = 8$ for the left and right hand location tasks, $k = 17$ for the left hand held object task, and $k = 13$ for the right hand held object task.

We report this averaged performance for each of the four single camera views as well as the late-fusion multiview model, in Table A.2.

We first note that, with the exception of the rearview-mounted camera on two left hand tasks, single-view models do not seem to generalize well across subjects; classification on the unseen subject tends to collapse to a few classes, likely due to overfit to a nearest-neighbor image in the training data. Practically speaking, this would suggest that models for driver state estimation which rely on a single camera would indeed benefit from fine-tuning on data from the driver of interest; we know that the model can train to near-perfect accuracy on data it has seen, it's the generalizability that causes the issue.

Now, to our primary question: can late-fusion multiview models overcome the generalizability challenge? Our results suggest that the late-fusion multiview model does outperform the best of the single-view models for unseen drivers on right-hand related tasks, though the rearview-mirror placed camera is excellent at left-hand related tasks. The late-fusion multiview model exceeds the average between the four single cameras on every task.

- For the left hand location task, the late-fusion multiview model is 9.6% less accurate than the best-performing rearview model, but 33% more accurate than the average across camera views.
- For the right hand location task, the late-fusion multiview model is 10.8% more accurate than the best-performing rearview model, and 45% more accurate than the average across camera views.
- For the left hand held object task, the late-fusion multiview model is 6% less accurate than the best-performing rearview model, but 30% more accurate than the average across camera views.
- For the right hand held object task, the late-fusion multiview model is 4.3% more accurate than the best-performing dashboard-center-view model, and 15% more accurate than the average across camera views.

A.5 Discussion and Concluding Remarks

System designers often have interest in selecting optimal number (and placement) of sensors for a given task, and in this research, we explored methods of leveraging the irregular redundancy of multiple sensors observing the same scene. While one camera placed expertly may be sufficient at a single task (say, observing the hands), there are many other tasks relevant to safe driving, such as estimating eyegaze, passenger seating occupation and positioning, and distraction identification.

Table A.6. Classification accuracies (averaged across all classes) of single-camera and late-fusion multiview models on four hand activity tasks. Evaluations are averaged over 19 models in which the evaluated subject is unseen during training, with mean and variance provided. View 1 is taken from the dashboard center, view 2 from the dashboard facing the driver, view 3 from the steering wheel, and view 4 from the rearview mirror. LFM refers to Late Fusion Multiview.

View	LH Location	RH Location	LH Object	RH Object	Average
1	0.333, 0.001	0.229, 0.007	0.320, 0.004	0.442, 0.006	0.331
2	0.217, 0.023	0.090, 0.008	0.276, 0.001	0.292, 0.004	0.219
3	0.317, 0.017	0.296, 0.027	0.259, 0.006	0.253, 0.08	0.281
4	**0.861**, 0.024	0.664, 0.059	**0.729**, 0.019	0.350, 0.014	0.651
LFM	0.765, 0.031	**0.772**, 0.015	0.699, 0.023	**0.485**, 0.049	**0.680**

What this framework contributes is a method of making stronger inference when information is missing from one source, and the system can recognize and leverage the fact the information is missing to then make better use of information in other sources. In fact, missing information often informs the other models; if a hand is not visible to one camera, then it is more likely to be within the view of another. Further considerations for enhanced accuracy include exploring weighting schemes for weighted majority voting, hyperparameter sweeps for crop sizes and model architecture, and model likelihood estimation for Bayesian model averaging.

Late-fusion approaches which use our method of replacement of missing data with a zero-placeholder may effectively learn a prior distribution given a missed reading from a sensor or camera. This is particularly relevant in cases where multiple perspectives are necessary for complete observation, or when multimodal systems are used which sample at different rates. We see plenty of examples of this in existing technology; many phones and laptop computers use both RGB and IR cameras for securely identifying the user, and thermal cameras are often used as an additional modality for medical applications, but cameras operating on different spectra (or media) typically operate at different rates.

No camera perfectly captures an event, but by using ensemble learning and fusion, safety systems (where every inch of accuracy counts) may exploit the benefits in redundancy and completeness of multi-view or multi-modal observations.

A.5.1 Efficient systems of multiple models

In this research, we evaluate performance of four models related to the driver's hands: two for held object (one for each hand), and two for hand location (one for each hand). While each model may function independently, cascading the models gives a better idea of a driver's current activity. For example, an image of the hand does not necessarily need to pass through both a held-object and hand-location model. In applications, the models can be cascaded such that first a held object can be determined, and if it is the case that no object is held, the image is passed forward to the location classification module. In fact, some applications may successfully “short-circuit" for efficiency depending on their use; a left-hand holding a cell phone may be sufficient to send an advisory without necessary inference on its location nor the right hand's activity.

Of further note, the system bottleneck most strongly occurs at the level of 2D pose estimation. To review, the model first detects the driver, then estimates the driver's pose, and from this pose classifies smaller regions pertaining to the hands (or, for other applications, eyes and other keypoints of interest). Fortunately, a system will only need to pass through this bottleneck once per inference time, since the remaining downstream models all utilize the same predicted pose information (and are much less computationally expensive). Because this system is modular, continued research from the computer vision community on efficient 2D pose estimation will translate directly and smoothly to performance gains in such human driver analysis systems.

Further research may benefit from an analysis of the Vision Transformer architecture for this problem, since the Transformer is particularly adept at selecting which features should be attended to. However, the Transformer is notably computationally expensive, so any performance gains must be balanced with increased inference delays to meet application requirements. The application of attention maps for (near) human-explainable reasoning from multimodal streams is considered an open challenge within multimodal learning [4], and these sample tasks and irregularly redundant data sources may be a strong candidate for future experiments.

A.5.2 Onboard vs. Cloud Processing

There is strong interest in moving compute from onboard processing toward cloud-based computing of driver monitoring data[1], but of key concern for consumer support and adoption of such data schemes is the preservation of driver privacy. To this end, we highlight that our presented framework allows for extraction of particular features (rather than complete images), which then allows for the anonymization via blurring or pixel value adjustment of the driver's face or similar privacy-sensitive content before sharing toward network computers, since these components are unused in model training and inference.

A.5.3 System design recommendations from experimental results

Our results lead us to the following system design recommendations for applications involving camera-based driver state estimation:

- When possible, collect data and finetune models using the driver of interest. Generalizability is a difficult task since the real-world may violate the i.i.d. assumptions that allow for excellent performance from neural networks. Unseen data may not come from the same distribution as prior training; the simpler case is to fit the model to data that most closely matches the expect distribution (i.e. images of the intended driver).

- If design constraints allow, opt for multiple cameras observing the driver to leverage complementary information between views, alternative views of occluded zones, and redundant information to provide improved accuracy and generalizability.

- If restricted to a single camera view, an overhead view from a camera placed near the rearview mirror may be optimal. If unavailable, a view facing the driver from behind the

[1] https://2023.ieee-iv.org/automotive-game-day

steering wheel may provide the best performance on estimating whether the driver's hands are on the wheel or elsewhere. However, this view is less well-suited to infer what the driver is doing with their hands if off-wheel; for this, a camera view facing the cabin from the rearview area or dashboard is better suited. The selected view should be informed by the intended application use case.

- While outside the scope of this research, we encourage applications implementing this framework to explore different hyperparameter values for crop size around the hands (or other features of interest). Differences in camera distance affect how much of the hand (and surrounding context) is visible within a particular crop size, and it may be worthwhile to vary these sizes for specific use cases depending on objects and locations of interest.

A.5.4 Post-processing considerations for downstream applications

Systems that seek to reliably estimate the state of the driver's hands (or similar driver attributes) will have to apply robust thresholds and denoising techniques to distinguish between genuine distractions and momentary lapses in attention.

Filtering

We suggest low-pass filtering to reduce the effects of noisy patterns from inference (that is, small "blips" between classes for fractions of a second). This allows for a more steady prediction result by averaging over moving windows of time, where the window size is a hyperparameter that can be tuned based on observation of the duration of a typical inference mistake made by the network.

Thresholding

We also suggest a thresholding step to distinguish between monetary lapses of attention (such as a driver quickly reaching for an object), versus an elongated period of distraction, which warrants an alert. The permissible interval of sustained distraction is another hyperparameter that should be tuned according to the goals of the automaker or driver policy.

Alerting

If it is decided that the driver may be distracted, the system can then issue a standard request for driver attentiveness, or employ other downstream safety mechanisms. It is recommended that the alert system employs a method of alerting aligned to the standards of human-machine interface research; these techniques are outside the scope of this research, but we emphasize that this is a modular endpiece, and the presented framework can be applied for any downstream alerting mechanism.

A.5.5 Additional applications

The hand activity framework requires multi-view capture, driver detection and pose estimation in its upstream steps. These tasks can be used towards several additional safety critical scenarios. As the driver detection step detects all individuals in the car, it can be used to estimate seat occupancy or passenger positioning. The cameras also capture driver gaze which can provide another signal towards driver attentiveness. The pose estimation module can provide data for studying safe airbag deployment in crashes. We believe demonstrating the effectiveness of multi-camera hand distraction can lead to further research in these applications to create holistic, robust, end-to-end systems for driver safety.

There are many further layers of analysis to problems of irregular redundancy; in this research, we move beyond complete sets to emphasize approaches which are applicable toward incomplete sets. Future work should incorporate temporal dynamics into this analysis, towards making systems which show even stronger generality to new subjects. However, this temporal dependency differs from that described in [4], where the goal is “to accumulate multimodal information across time so that long-range cross-modal interactions can be captured through storage and retrieval from memory" – rather, we seek to retain short-range information from the collective representation, such that iterative predictions are consistent with prior predicted states. There is further promise in the ability of ensemble techniques to generalize to an entirely different (and relevant) class of what is “unseen": in addition to generalizing to new subjects, domain-adaptive ensemble methods have also been shown to be effective learners to entirely new views [407], making them highly appropriate towards driver monitoring domain tasks, where the same views may not be guaranteed between vehicle designs.

We conclude that the late fusion technique is a strong baseline toward problems where multiple data streams, possibly under noise and dropped instances, are sampled simultaneously for continuous task inference.

Acknowledgements

Appendix A, in part, is a reprint of the material as it appears in Ross Greer and Mohan M. Trivedi. "Ensemble Learning for Fusion of Multiview Vision with Occlusion and Missing Information: Framework and Evaluations with Real-World Data and Applications in Driver Hand Activity Recognition." arXiv preprint arXiv:2301.12592 (2023). The dissertation author was the primary investigator and author of this paper.

Bibliography

[1] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, pp. 201–221, 1994.

[2] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Predicting take-over time for autonomous driving with real-world data: Robust data augmentation, models, and evaluation," *arXiv preprint arXiv:2107.12932*, 2021.

[3] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 231–236, IEEE, 2021.

[4] P. P. Liang, A. Zadeh, and L.-P. Morency, "Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions," *arXiv preprint arXiv:2209.03430*, 2022.

[5] M. Liu, E. Yurtsever, X. Zhou, J. Fossaert, Y. Cui, B. L. Zagar, and A. C. Knoll, "A survey on autonomous driving datasets: Data statistic, annotation, and outlook," *arXiv preprint arXiv:2401.01454*, 2024.

[6] H. Li, Y. Li, H. Wang, J. Zeng, P. Cai, H. Xu, D. Lin, J. Yan, F. Xu, L. Xiong, *et al.*, "Open-sourced data ecosystem in autonomous driving: the present and future," *arXiv preprint arXiv:2312.03408*, 2023.

[7] D. Bogdoll, F. Schreyer, and J. M. Zöllner, "Ad-datasets: a meta-collection of data sets for autonomous driving," *arXiv preprint arXiv:2202.01909*, 2022.

[8] T. Fingscheidt, H. Gottschalk, and S. Houben, *Deep neural networks and data for automated driving: Robustness, uncertainty quantification, and insights towards safety*. Springer Nature, 2022.

[9] H. Gottschalk, M. Rottmann, and M. Saltagic, "Does redundancy in ai perception systems help to test for super-human automated driving performance?," *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, pp. 81–106, 2022.

[10] H. X. Liu and S. Feng, ""curse of rarity" for autonomous vehicles," *arXiv preprint arXiv:2207.02749*, 2022.

[11] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick, "An application-driven conceptualization of corner cases for perception in highly automated driving," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 644–651, IEEE, 2021.

[12] C. Gruhl, B. Sick, and S. Tomforde, "Novelty detection in continuously changing environments," *Future Generation Computer Systems*, vol. 114, pp. 138–154, 2021.

[13] P. Koopman, A. Kane, and J. Black, "Credible autonomy safety argumentation," in *27th Safety-Critical Systems Symposium*, pp. 34–50, 2019.

[14] J.-A. Bolte, A. Bar, D. Lipinski, and T. Fingscheidt, "Towards corner case detection for autonomous driving," in *2019 IEEE Intelligent vehicles symposium (IV)*, pp. 438–445, IEEE, 2019.

[15] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos, "Video anomaly detection in spatiotemporal context," in *2010 IEEE International Conference on Image Processing*, pp. 705–708, IEEE, 2010.

[16] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, "Systematization of corner cases for visual perception in automated driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1257–1264, IEEE, 2020.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[18] I. Jatzkowski, D. Wilke, and M. Maurer, "A deep-learning approach for the detection of overexposure in automotive camera images," in *2018 21St international conference on intelligent transportation systems (ITSC)*, pp. 2030–2035, IEEE, 2018.

[19] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, "Lost and found: detecting small road hazards for self-driving vehicles," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1099–1106, IEEE, 2016.

[20] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1025–1032, IEEE, 2017.

[21] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving," in *proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

[22] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the unexpected via image resynthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2152–2161, 2019.

[23] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Conference on Robot Learning*, pp. 384–393, PMLR, 2020.

[24] E. Capellier, F. Davoine, V. Cherfaoui, and Y. Li, "Evidential deep learning for arbitrary lidar object classification in the context of autonomous driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1304–1311, IEEE, 2019.

[25] R. V. Chakravarthy, H. Liu, and A. M. Pavy, "Open-set radar waveform classification: Comparison of different features and classifiers," in *2020 IEEE International Radar Conference (RADAR)*, pp. 542–547, 2020.

[26] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *European Conference on Computer Vision*, pp. 335–352, Springer, 2022.

[27] M. L. Cummings and B. Bauchwitz, "Safety implications of variability in autonomous driving assist alerting," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 8, pp. 12039–12049, 2021.

[28] R. Greer, N. Deo, A. Rangesh, P. Gunaratne, and M. Trivedi, "Safe control transitions: Machine vision based observable readiness index and data-driven takeover time prediction," *27th International Technical Symposium on the Enhanced Safety of Vehicles (ESV)*, 2023.

[29] C. S. Vallon and F. Borrelli, "Data-driven strategies for hierarchical predictive control in unknown environments," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1434–1445, 2022.

[30] A. Hekimoglu, P. Friedrich, W. Zimmer, M. Schmidt, A. Marcos-Ramiro, and A. Knoll, "Multi-task consistency for active learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3415–3424, 2023.

[31] G. Hacohen, A. Dekel, and D. Weinshall, "Active learning on a budget: Opposite strategies suit high and low budgets," *arXiv preprint arXiv:2202.02794*, 2022.

[32] R. Greer, B. Antoniussen, M. V. Andersen, A. Møgelmose, and M. M. Trivedi, "The why, when, and how to use active learning in large-data-driven 3d object detection for safe autonomous driving: An empirical exploration," *arXiv preprint arXiv:2401.16634*, 2024.

[33] A. Roitberg, C. Ma, M. Haurilet, and R. Stiefelhagen, "Open set driver activity recognition," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1048–1053, IEEE, 2020.

[34] R. T. Calumby, R. da Silva Torres, and M. A. Gonçalves, "Diversity-driven learning for multimodal image retrieval with relevance feedback," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2197–2201, IEEE, 2014.

[35] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[36] G. Singh, S. Akrigg, M. Di Maio, V. Fontana, R. J. Alitappeh, S. Saha, K. Jeddisaravi, F. Yousefi, J. Culley, T. Nicholson, *et al.*, "Road: The road event awareness dataset for autonomous driving," *arXiv preprint arXiv:2102.11585*, 2021.

[37] A. Ghita, B. Antoniussen, W. Zimmer, R. Greer, C. Creß, A. Møgelmose, M. M. Trivedi, and A. C. Knoll, "Activeanno3d–an active learning framework for multi-modal 3d object detection," *arXiv preprint arXiv:2402.03235*, 2024.

[38] Y. Wu, I. Kozintsev, J.-Y. Bouguet, and C. Dulong, "Sampling strategies for active learning in personal photo retrieval," in *2006 IEEE International Conference on Multimedia and Expo*, pp. 529–532, IEEE, 2006.

[39] Z. Liang, X. Xu, S. Deng, L. Cai, T. Jiang, and K. Jia, "Exploring diversity-based active learning for 3d object detection in autonomous driving," *arXiv preprint arXiv:2205.07708*, 2022.

[40] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. Nesnas, and M. Pavone, "Semantic anomaly detection with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1035–1055, 2023.

[41] O. Çatal, S. Leroux, C. De Boom, T. Verbelen, and B. Dhoedt, "Anomaly detection for autonomous guided vehicles using bayesian surprise," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8148–8153, IEEE, 2020.

[42] A. Xie, F. Tajwar, A. Sharma, and C. Finn, "When to ask for help: Proactive interventions in autonomous reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16918–16930, 2022.

[43] A. Chen, A. Sharma, S. Levine, and C. Finn, "You only live once: Single-life reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14784–14797, 2022.

[44] D. Xiao, M. Dianati, W. G. Geiger, and R. Woodman, "Review of graph-based hazardous event detection methods for autonomous driving systems," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[45] D. Mahapatra, A. Poellinger, L. Shao, and M. Reyes, "Interpretability-driven sample selection using self supervised learning for disease classification and segmentation," *IEEE transactions on medical imaging*, vol. 40, no. 10, pp. 2548–2562, 2021.

[46] H. Wang, W. Wang, S. Yuan, and X. Li, "Uncovering interpretable internal states of merging tasks at highway on-ramps for autonomous driving decision-making," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2825–2836, 2021.

[47] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with llms: Fusing object-level vector modality for explainable autonomous driving," *arXiv preprint arXiv:2310.01957*, 2023.

[48] Y. Cui, S. Huang, J. Zhong, Z. Liu, Y. Wang, C. Sun, B. Li, X. Wang, and A. Khajepour, "Drivellm: Charting the path toward full autonomous driving with large language models," *IEEE Transactions on Intelligent Vehicles*, 2023.

[49] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[50] X. Li and Y. Guo, "Adaptive active learning for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 859–866, 2013.

[51] H. Abualsaud, S. Liu, D. B. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, "Laneaf: Robust multi-lane detection with affinity fields," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7477–7484, 2021.

[52] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," *Pattern Recognition*, vol. 130, p. 108796, 2022.

[53] R. Greer, A. Gopalkrishnan, J. Landgren, L. Rakla, A. Gopalan, and M. Trivedi, "Robust traffic light detection using salience-sensitive loss: Computational framework and evaluations," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, 2023.

[54] R. Greer, J. Isa, N. Deo, A. Rangesh, and M. M. Trivedi, "On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 636–644, 2022.

[55] R. Greer, A. Gopalkrishnan, N. Deo, A. Rangesh, and M. Trivedi, "Salient sign detection in safe autonomous driving: Ai which reasons over full visual context," *27th International Technical Symposium on the Enhanced Safety of Vehicles (ESV)*, 2023.

[56] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[57] S. H. Nair, E. H. Tseng, and F. Borrelli, "Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5267–5272, IEEE, 2022.

[58] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[59] D. Coelho, M. Oliveira, and V. Santos, "Rlad: Reinforcement learning from pixels for autonomous driving in urban environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–9, 2023.

[60] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1468–1476, 2018.

[61] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE intelligent vehicles symposium (IV)*, pp. 1179–1184, IEEE, 2018.

[62] C. Chen, X. Chen, C. Guo, and P. Hang, "Trajectory prediction for autonomous driving based on structural informer method," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2023.

[63] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 165–170, IEEE, 2021.

[64] R. Greer, N. Deo, and M. Trivedi, "Trajectory prediction in autonomous driving with a lane heading auxiliary loss," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4907–4914, 2021.

[65] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *arXiv preprint arXiv:2001.03093*, 2020.

[66] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011.

[67] A. Gopalkrishnan, R. Greer, M. Keskar, and M. Trivedi, "Robust detection, assocation, and localization of vehicle lights: A context-based cascaded cnn approach and evaluations," *arXiv preprint arXiv:2307.14571*, 2023.

[68] R. Greer, A. Gopalkrishnan, M. Keskar, and M. M. Trivedi, "Patterns of vehicle lights: Addressing complexities of camera-based vehicle light datasets and metrics," *Pattern Recognition Letters*, 2024.

[69] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[70] F. Galatolo., M. Cimino., and G. Vaglini, "Generating images from caption and vice versa via clip-guided generative latent space search," *Proceedings of the International Conference on Image Processing and Vision Engineering*, 2021.

[71] X. Zhu, R. Zhang, B. He, Z. Guo, Z. Zeng, Z. Qin, S. Zhang, and P. Gao, "Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2639–2650, 2023.

[72] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, “Simple but effective: Clip embeddings for embodied ai,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

[73] M. Singh, E. Curran, and P. Cunningham, “Active learning for multi-label image annotation,” tech. rep., University College Dublin. School of Computer Science and Informatics, 2009.

[74] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[75] N. Kulkarni, A. Rangesh, J. Buck, J. Feltracco, M. Trivedi, N. Deo, R. Greer, S. Sarraf, and S. Sathyanarayana, “Create a large-scale video driving dataset with detailed attributes using amazon sagemaker ground truth,” 2021.

[76] W. Zimmer, J. Birkner, M. Brucker, H. T. Nguyen, S. Petrovski, B. Wang, and A. C. Knoll, “Infradet3d: Multi-modal 3d object detection based on roadside infrastructure camera and lidar sensors,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2023.

[77] W. Zimmer, C. Creß, H. T. Nguyen, and A. C. Knoll, “Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception,” in *2023 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2023.

[78] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[79] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” *arXiv preprint arXiv:1109.2378*, 2011.

[80] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, “Fast optimal leaf ordering for hierarchical clustering,” *Bioinformatics*, vol. 17, no. suppl_1, pp. S22–S29, 2001.

[81] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *arXiv preprint arXiv:2304.08485*, 2023.

[82] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.*, “Mistral 7b,” *arXiv preprint arXiv:2310.06825*, 2023.

[83] S. Wang, Y. Sun, Z. Wang, and M. Liu, “St-tracknet: A multiple-object tracking network using spatio-temporal information,” *IEEE Transactions on Automation Science and Engineering*, 2022.

[84] H. Cai, Z. Zhang, Z. Zhou, Z. Li, W. Ding, and J. Zhao, “Bevfusion4d: Learning lidar-camera fusion under bird’s-eye-view via cross-modality guidance and temporal aggregation,” *arXiv preprint arXiv:2303.17099*, 2023.

[85] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 2774–2781, IEEE, 2023.

[86] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. M. Alvarez, "Focalformer3d: focusing on hard instance for 3d object detection," in *Proceedings of the IEEE/CVF International Conference On Computer Vision*, pp. 8394–8405, 2023.

[87] Y. Xie, C. Xu, M.-J. Rakotosaona, P. Rim, F. Tombari, K. Keutzer, M. Tomizuka, and W. Zhan, "Sparsefusion: Fusing multi-modal sparse representations for multi-sensor 3d object detection," *arXiv preprint arXiv:2304.14340*, 2023.

[88] N. Kulkarni, A. Rangesh, J. Buck, J. Feltracco, M. M. Trivedi, N. Deo, R. Greer, S. Sarraf, and S. Sathyanarayana, "Create a large-scale video driving dataset with detailed attributes using amazon sagemaker ground truth: Lisa amazonmlsl vehicle attributes (lava) dataset," *AWS Machine Learning Blog*, June 2021.

[89] R. Greer, L. Rakla, S. Desai, A. Alofi, A. Gopalkrishnan, and M. Trivedi, "Champ: Crowdsourced, history-based advisory of mapped pedestrians for safer driver assistance systems," *arXiv preprint arXiv:2301.05842*, 2023.

[90] H. Gottschalk, M. Rottmann, and M. Saltagic, "Does redundancy in ai perception systems help to test for super-human automated driving performance?," *arXiv preprint arXiv:2112.04758*, 2021.

[91] B. Pes, "Learning from high-dimensional biomedical datasets: the issue of class imbalance," *IEEE Access*, vol. 8, pp. 13527–13540, 2020.

[92] H. B. Lee, T. Nam, E. Yang, and S. J. Hwang, "Meta dropout: Learning to perturb latent features for generalization," in *Eighth International Conference on Learning Representations, ICLR 2020*, International Conference on Learning Representations, 2020.

[93] S. Dubnov and R. Greer, *Deep and shallow: Machine learning in music and audio.* CRC Press, 2023.

[94] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable active learning for object detection," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1430–1435, 2020.

[95] R. Greer, J. Isa, N. Deo, A. Rangesh, and M. M. Trivedi, "On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset," in *2022 Winter Conference on Applications of Computer Vision (WACV)*.

[96] S. Lefevre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 32–42, 2015.

[97] R. Greer, A. Gopalkrishnan, M. Keskar, and M. M. Trivedi, “Patterns of vehicle lights: Addressing complexities of camera-based vehicle light datasets and metrics,” *Pattern Recognition Letters*, 2024.

[98] A. Doshi and M. M. Trivedi, “Examining the impact of driving style on the predictability and responsiveness of the driver: Real-world and simulator analysis,” in *2010 IEEE Intelligent Vehicles Symposium*, pp. 232–237, IEEE, 2010.

[99] A. Balachandran, M. Brown, S. M. Erlien, and J. C. Gerdes, “Predictive haptic feedback for obstacle avoidance based on model predictive control,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 26–31, 2015.

[100] W. Zimmer, A. Rangesh, and M. Trivedi, “3d bat: A semi-automatic, web-based 3d annotation toolbox for full-surround, multi-modal data streams,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1816–1821, IEEE, 2019.

[101] J. Lee, S. Walsh, A. Harakeh, and S. L. Waslander, “Leveraging pre-trained 3d object detection models for fast ground truth generation,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2504–2510, IEEE, 2018.

[102] C. Liu, X. Qian, X. Qi, E. Y. Lam, S.-C. Tan, and N. Wong, “Map-gen: An automated 3d-box annotation flow with multimodal attention point generator,” in *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 1148–1155, IEEE, 2022.

[103] R. Greer, L. Rakla, A. Gopalkrishnan, and M. Trivedi, “Multi-view ensemble learning with missing data: Computational framework and evaluations using novel data from the safe autonomous driving domain,” *arXiv preprint arXiv:2301.12592*, 2023.

[104] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, “Deep active learning for efficient training of a lidar 3d object detector,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 667–674, IEEE, 2019.

[105] A. Moses, S. Jakkampudi, C. Danner, and D. Biega, “Localization-based active learning (local) for object detection in 3d point clouds,” in *Geospatial Informatics XII*, vol. 12099, pp. 44–58, SPIE, 2022.

[106] Y. Luo, Z. Chen, Z. Fang, Z. Zhang, M. Baktashmotlagh, and Z. Huang, “Kecor: Kernel coding rate maximization for active 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18279–18290, 2023.

[107] A. Hekimoglu, M. Schmidt, and A. Marcos-Ramiro, “Monocular 3d object detection with lidar guided semi supervised active learning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2346–2355, 2024.

[108] S. Hwang, S. Kim, Y. Kim, and D. Kum, “Joint semi-supervised and active learning via 3d consistency for 3d object detection,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4819–4825, IEEE, 2023.

[109] S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, “Advanced active learning strategies for object detection,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 871–876, IEEE, 2020.

[110] J. Yuan, B. Zhang, X. Yan, T. Chen, B. Shi, Y. Li, and Y. Qiao, “Bi3d: Bi-domain active learning for cross-domain 3d object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15599–15608, 2023.

[111] M. Meyer and G. Kuschk, “Automotive radar dataset for deep learning based 3d object detection,” in *2019 16th european radar conference (EuRAD)*, pp. 129–132, IEEE, 2019.

[112] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. Van Gool, “Towards a weakly supervised framework for 3d point cloud object detection and annotation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4454–4468, 2021.

[113] G. Villalonga and A. M. L. Pena, “Co-training for on-board deep object detection,” *IEEE Access*, vol. 8, pp. 194441–194456, 2020.

[114] A. Almin, L. Lemarié, A. Duong, and B. R. Kiran, “Navya3dseg-navya 3d semantic segmentation dataset design & split generation for autonomous vehicles,” *IEEE Robotics and Automation Letters*, 2023.

[115] L. Chen, X. He, X. Zhao, H. Li, Y. Huang, B. Zhou, W. Chen, Y. Li, C. Wen, and C. Wang, “Gocomfort: Comfortable navigation for autonomous vehicles leveraging high-precision road damage crowdsensing,” *IEEE Transactions on Mobile Computing*, 2022.

[116] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

[117] D. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning,” in *Machine learning proceedings 1994*, pp. 148–156, Elsevier, 1994.

[118] T. Scheffer, C. Decomain, and S. Wrobel, “Active hidden markov models for information extraction,” in *International Symposium on Intelligent Data Analysis*, pp. 309–318, Springer, 2001.

[119] B. Settles, M. Craven, and S. Ray, “Multiple-instance active learning,” *Advances in neural information processing systems*, vol. 20, pp. 1289–1296, 2007.

[120] A. I. Schein and L. H. Ungar, “Active learning for logistic regression: an evaluation,” *Machine Learning*, vol. 68, no. 3, pp. 235–265, 2007.

[121] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, 2008.

[122] S. Sivaraman and M. M. Trivedi, “A general active-learning framework for on-road vehicle recognition and tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[123] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study," *Machine vision and applications*, vol. 25, no. 3, pp. 599–611, 2014.

[124] R. K. Satzoda and M. M. Trivedi, "Multipart vehicle detection using symmetry-derived analysis and active learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 926–937, 2015.

[125] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[126] N. Singh, H. Hukkelås, and F. Lindseth, "Deep active learning for autonomous perception," in *NIKT: Norsk IKT-konferanse for forskning og utdanning 2020*, Bibsys Open Journal Systems, 2020.

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

[128] A. Hekimoglu, M. Schmidt, A. Marcos-Ramiro, and G. Rigoll, "Efficient active learning strategies for monocular 3d object detection," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 295–302, IEEE, 2022.

[129] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *CoRR*, vol. abs/1903.11027, 2019.

[130] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations," in *2015 IEEE intelligent vehicles symposium (IV)*, pp. 330–335, IEEE, 2015.

[131] F. A. Schmidt, *Crowdproduktion von Trainingsdaten: Zur Rolle von Online-Arbeit beim Trainieren autonomer Fahrzeuge*. No. 417, Study der Hans-Böckler-Stiftung, 2019.

[132] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," *Advances in neural information processing systems*, vol. 31, 2018.

[133] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*, pp. 3–12, Springer, 1994.

[134] V. Nguyen, M. H. Shaker, and E. Hüllermeier, "How to measure uncertainty in uncertainty sampling for active learning," *https://doi.org/10.1007/s10994-021-06003-9*, 2021.

[135] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," *arXiv preprint arXiv:2205.13542v2*, 2022.

[136] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, , and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted window," *ICCV*, 2021.

[137] Y. Yan, Y. Mao, , and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, 2018.

[138] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detectio," *CVPR*, 2017.

[139] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," *arXiv preprint arXiv:1708.02383*, 2017.

[140] T. Lew, A. Sharma, J. Harrison, A. Bylard, and M. Pavone, "Safe active dynamics learning and control: A sequential exploration–exploitation framework," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2888–2907, 2022.

[141] V. Chen, M.-K. Yoon, and Z. Shao, "Task-aware novelty detection for visual-based deep learning in autonomous systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11060–11066, IEEE, 2020.

[142] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[143] E. Ohn-Bar and M. M. Trivedi, "What makes an on-road object important?," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3392–3397, IEEE, 2016.

[144] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger, "Label efficient visual abstractions for autonomous driving," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2338–2345, IEEE, 2020.

[145] S. Dasgupta, "Two faces of active learning," *Theoretical computer science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[146] R. Greer and M. Trivedi, "Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets," *arXiv preprint arXiv:2402.07320*, 2024.

[147] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *International Conference on Machine Learning*, 2021.

[148] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[149] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2090–2096, IEEE, 2019.

[150] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” *arXiv preprint arXiv:1910.05449*, 2019.

[151] D. Ridel, N. Deo, D. Wolf, and M. Trivedi, “Scene compliant trajectory forecast with agent-centric spatio-temporal grids,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2816–2823, 2020.

[152] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, “Multi-head attention with joint agent-map representation for trajectory prediction in autonomous driving,” *arXiv preprint arXiv:2005.02545*, 2020.

[153] S. Casas, C. Gulino, S. Suo, and R. Urtasun, “The importance of prior knowledge in precise multimodal prediction,” *arXiv preprint arXiv:2006.02636*, 2020.

[154] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” *arXiv preprint arXiv:2007.13732*, 2020.

[155] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[156] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, “Sophie: An attentive gan for predicting paths compliant to social and physical constraints,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.

[157] J. Amirian, J.-B. Hayet, and J. Pettré, “Social ways: Learning multi-modal distributions of pedestrian trajectories with gans,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[158] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, “Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks,” in *Advances in Neural Information Processing Systems*, pp. 137–146, 2019.

[159] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, “Multi-agent tensor fusion for contextual trajectory prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12126–12134, 2019.

[160] E. Wang, H. Cui, S. Yalamanchi, M. Moorthy, F.-C. Chou, and N. Djuric, “Improving movement predictions of traffic actors in bird’s-eye view models using gans and differentiable trajectory rasterization,” *arXiv preprint arXiv:2004.06247*, 2020.

[161] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, “Desire: Distant future prediction in dynamic scenes with interacting agents,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–345, 2017.

[162] B. Ivanovic and M. Pavone, “The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2375–2384, 2019.

[163] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, “Implicit latent variable model for scene-consistent motion forecasting,” *arXiv preprint arXiv:2007.12036*, 2020.

[164] N. Rhinehart, K. M. Kitani, and P. Vernaza, “R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 772–788, 2018.

[165] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, “Precog: Prediction conditioned on goals in visual multi-agent settings,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2830, 2019.

[166] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C.-N. Straehle, “Conditional flow variational autoencoders for structured sequence prediction,” *arXiv preprint arXiv:1908.09008*, 2019.

[167] A. Bhattacharyya, C.-N. Straehle, M. Fritz, and B. Schiele, “Haar wavelet based block autoregressive flows for trajectories,” *arXiv preprint arXiv:2009.09878*, 2020.

[168] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, “Planning-based prediction for pedestrians,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3931–3936, IEEE, 2009.

[169] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, “Activity forecasting,” in *European Conference on Computer Vision*, pp. 201–214, Springer, 2012.

[170] Y. Zhang, W. Wang, R. Bonatti, D. Maturana, and S. Scherer, “Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories,” *arXiv preprint arXiv:1810.07225*, 2018.

[171] N. Deo and M. M. Trivedi, “Trajectory forecasts in unknown environments conditioned on grid-based plans,” *arXiv preprint arXiv:2001.00735*, 2020.

[172] M. Niedoba, H. Cui, K. Luo, D. Hegde, F.-C. Chou, and N. Djuric, “Improving movement prediction of traffic actors using off-road loss and bias mitigation,” in *Workshop on’Machine Learning for Autonomous Driving’at Conference on Neural Information Processing Systems*, 2019.

[173] F. A. Boulton, E. C. Grigore, and E. M. Wolff, “Motion prediction using trajectory sets and self-driving domain knowledge,” *arXiv preprint arXiv:2006.04767*, 2020.

[174] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, “Human motion trajectory prediction: A survey,” *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[175] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf, “A literature review on the prediction of pedestrian behavior in urban scenarios,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3105–3112, IEEE, 2018.

[176] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, “Covernet: Multimodal behavior prediction using trajectory sets,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14074–14083, 2020.

[177] S. Casas, C. Gulino, R. Liao, and R. Urtasun, “Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9491–9497, IEEE, 2020.

[178] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, “Deep kinematic models for kinematically feasible vehicle trajectory predictions,” in *International Conference on Robotics and Automation (ICRA)*, pp. 10563–10569, IEEE, 2020.

[179] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, “Argoverse: 3d tracking and forecasting with rich maps,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

[180] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” *arXiv preprint arXiv:1903.11027*, 2019.

[181] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

[182] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[183] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: a survey,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5595–5637, 2017.

[184] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, “Self-driving cars,” *Computer*, vol. 50, no. 12, pp. 18–23, 2017.

[185] S. Kim, H. Jeon, J. W. Choi, and D. Kum, “Diverse multiple trajectory prediction using a two-stage prediction network trained with lane loss,” *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2038–2045, 2022.

[186] J. Rückin, F. Magistri, C. Stachniss, and M. Popović, “Semi-supervised active learning for semantic segmentation in unknown environments using informative path planning,” *IEEE Robotics and Automation Letters*, 2024.

[187] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” *arXiv preprint arXiv:1708.00489*, 2017.

[188] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, “Multi-class active learning by uncertainty sampling with diversity maximization,” *International Journal of Computer Vision*, vol. 113, pp. 113–127, 2015.

[189] H. Lu, X. Jia, Y. Xie, W. Liao, X. Yang, and J. Yan, “Activead: Planning-oriented active learning for end-to-end autonomous driving,” *arXiv preprint arXiv:2403.02877*, 2024.

[190] M. Rottmann and M. Reese, “Automated detection of label errors in semantic segmentation datasets via deep learning and uncertainty quantification,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3214–3223, 2023.

[191] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai, “Addressing the item cold-start problem by attribute-driven active learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 631–644, 2019.

[192] R. K. Satzoda and M. M. Trivedi, “Drive analysis using vehicle dynamics and vision-based lane semantics,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 9–18, 2014.

[193] M. Van Ly, S. Martin, and M. M. Trivedi, “Driver classification and driving style recognition using inertial sensors,” in *2013 IEEE intelligent vehicles symposium (IV)*, pp. 1040–1045, IEEE, 2013.

[194] K. Rösch, F. Heidecker, J. Truetsch, K. Kowol, C. Schicktanz, M. Bieshaare, B. Sick, and C. Stiller, “Space, time, and interaction: A taxonomy of corner cases in trajectory datasets for automated driving,” in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 86–93, IEEE, 2022.

[195] N. Deo, E. Wolff, and O. Beijbom, “Multimodal trajectory prediction conditioned on lane-graph traversals,” in *Conference on Robot Learning*, pp. 203–212, PMLR, 2022.

[196] A. Gopalkrishnan, R. Greer, and M. Trivedi, “Multi-frame, lightweight & efficient vision-language models for question answering in autonomous driving,” *arXiv preprint arXiv:2403.19838*, 2024.

[197] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger, “Learning situational driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11296–11305, 2020.

[198] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger, "Exploring data aggregation in policy learning for vision-based urban autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11763–11773, 2020.

[199] N. Deo and M. M. Trivedi, "Looking at the driver/rider in autonomous vehicles to predict take-over readiness," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 41–52, 2019.

[200] R. Greer, A. Gopalkrishnan, S. Mandadi, P. Gunaratne, M. M. Trivedi, and T. D. Marcotte, "Vision-based analysis of driver activity and driving performance under the influence of alcohol," *FAST-Zero*, 2023.

[201] K. Kircher and C. Ahlstrom, "Minimum required attention: a human-centered approach to driver inattention," *Human factors*, vol. 59, no. 3, pp. 471–484, 2017.

[202] B. S. Jensen, M. B. Skov, and N. Thiruravichandran, "Studying driver attention and behaviour for three configurations of gps navigation in real traffic driving," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1271–1280, 2010.

[203] P. Tice, S. Dey Tirtha, and N. Eluru, "Driver attention and the built environment initial, findings from a naturalistic driving study," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 65, pp. 1077–1081, SAGE Publications Sage CA: Los Angeles, CA, 2021.

[204] J. Gaspar and C. Carney, "The effect of partial automation on driver attention: A naturalistic driving study," *Human factors*, vol. 61, no. 8, pp. 1261–1276, 2019.

[205] S. Benedetto, M. Pedrotti, L. Minin, T. Baccino, A. Re, and R. Montanari, "Driver workload and eye blink duration," *Transportation research part F: traffic psychology and behaviour*, vol. 14, no. 3, pp. 199–208, 2011.

[206] T. C. Ojsteršek, "Eye tracking use in researching driver distraction: A scientometric and qualitative literature review approach," *Journal of Eye Movement Research*, vol. 12, no. 3, 2019.

[207] C. Isaza, K. Anaya, C. Fuentes-Silva, J. P. Z. de Paz, A. Rizzo, and A.-I. Garcia-Moreno, "Dynamic set point model for driver alert state using digital image processing," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19543–19563, 2019.

[208] J. D. Eastwood, A. Frischen, M. J. Fenske, and D. Smilek, "The unengaged mind: Defining boredom in terms of attention," *Perspectives on Psychological Science*, vol. 7, no. 5, pp. 482–495, 2012.

[209] S. G. Charlton and N. J. Starkey, "Driving without awareness: The effects of practice and automaticity on attention and driving," *Transportation research part F: traffic psychology and behaviour*, vol. 14, no. 6, pp. 456–471, 2011.

[210] J. J. Bernstein and J. Bernstein, “Texting at the light and other forms of device distraction behind the wheel,” *BMC public health*, vol. 15, no. 1, pp. 1–5, 2015.

[211] “Automated vehicles for safety.”

[212] B. Williams, “Automated driving levels,” 2021.

[213] S. M. Casner, E. L. Hutchins, and D. Norman, “The challenges of partially automated driving,” *Communications of the ACM*, vol. 59, no. 5, pp. 70–77, 2016.

[214] M. Kyriakidis, J. C. de Winter, N. Stanton, T. Bellet, B. van Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat, *et al.*, “A human factors perspective on automated driving,” *Theoretical Issues in Ergonomics Science*, pp. 1–27, 2017.

[215] J. S. Warm, R. Parasuraman, and G. Matthews, “Vigilance requires hard mental work and is stressful,” *Human factors*, vol. 50, no. 3, pp. 433–441, 2008.

[216] E. Murphy-Chutorian and M. M. Trivedi, “Hyhope: Hybrid head orientation and position estimation for vision-based driver head tracking,” in *2008 IEEE Intelligent Vehicles Symposium*, pp. 512–517, IEEE, 2008.

[217] A. Tawari and M. M. Trivedi, “Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos,” in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 344–349, 2014.

[218] A. Tawari, K. H. Chen, and M. M. Trivedi, “Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 988–994, IEEE, 2014.

[219] S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, “Real-time gaze estimator based on driver’s head orientation for forward collision warning system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 254–267, 2011.

[220] B. Vasli, S. Martin, and M. M. Trivedi, “On driver gaze estimation: Explorations and fusion of geometric and data driven approaches,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 655–660, IEEE, 2016.

[221] L. Fridman, P. Langhans, J. Lee, and B. Reimer, “Driver gaze region estimation without use of eye movement,” *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 49–56, 2016.

[222] L. Fridman, J. Lee, B. Reimer, and T. Victor, “‘owl’and ‘lizard’: patterns of head pose and eye pose in driver gaze classification,” *IET Computer Vision*, vol. 10, no. 4, pp. 308–314, 2016.

[223] S. Vora, A. Rangesh, and M. M. Trivedi, “On generalizing driver gaze zone estimation using convolutional neural networks,” in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 849–854, IEEE, 2017.

[224] S. Vora, A. Rangesh, and M. M. Trivedi, "Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 254–265, 2018.

[225] R. A. Naqvi, M. Arsalan, G. Batchuluun, H. S. Yoon, and K. R. Park, "Deep learning-based gaze detection system for automobile drivers using a nir camera sensor," *Sensors*, vol. 18, no. 2, p. 456, 2018.

[226] S. Jha and C. Busso, "Probabilistic estimation of the gaze region of the driver using dense classification," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 697–702, Nov 2018.

[227] A. Rangesh, B. Zhang, and M. M. Trivedi, "Driver gaze estimation in the real world: Overcoming the eyeglass challenge," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1054–1059, IEEE, 2020.

[228] A. Tawari, S. Martin, and M. M. Trivedi, "Continuous head movement estimator for driver assistance: Issues, algorithms, and on-road evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 818–830, 2014.

[229] A. Doshi and M. M. Trivedi, "Head and eye gaze dynamics during visual attention shifts in complex environments," *Journal of vision*, vol. 12, no. 2, pp. 9–9, 2012.

[230] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *2015 IEEE 18th international conference on intelligent transportation systems*, pp. 2953–2958, IEEE, 2015.

[231] E. Ohn-Bar and M. Trivedi, "In-vehicle hand activity recognition using integration of regions," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 1034–1039, IEEE, 2013.

[232] E. Ohn-Bar and M. M. Trivedi, "Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 1245–1250, IEEE, 2014.

[233] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE transactions on intelligent transportation systems*, vol. 15, no. 6, pp. 2368–2377, 2014.

[234] G. Borghi, E. Frigieri, R. Vezzani, and R. Cucchiara, "Hands on the wheel: a dataset for driver hand detection and tracking," in *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pp. 564–570, IEEE, 2018.

[235] A. Rangesh, E. Ohn-Bar, and M. M. Trivedi, "Hidden hands: Tracking hands with an occlusion aware tracker," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 19–26, 2016.

[236] N. Deo, A. Rangesh, and M. Trivedi, "In-vehicle hand gesture recognition using hidden markov models," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 2179–2184, IEEE, 2016.

[237] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–7, 2015.

[238] K. Yuen and M. M. Trivedi, "Looking at hands in autonomous vehicles: A convnet approach using part affinity fields," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 361–371, 2019.

[239] A. Rangesh and M. M. Trivedi, "Handynet: A one-stop solution to detect, segment, localize & analyze driver hands," *arXiv preprint arXiv:1804.07834*, 2018.

[240] C. Tran, A. Doshi, and M. M. Trivedi, "Pedal error prediction by driver foot gesture analysis: A vision-based inquiry," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 577–582, IEEE, 2011.

[241] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012.

[242] A. Rangesh and M. Trivedi, "Forced spatial attention for driver foot activity classification," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[243] A. Rangesh and M. Trivedi, "Forced spatial attention for driver foot activity classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[244] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 660–665, IEEE, 2014.

[245] C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel, "Driver-activity recognition in the context of conditionally autonomous driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 1652–1657, IEEE, 2015.

[246] A. Behera, A. Keidel, and B. Debnath, "Context-driven multi-stream lstm (m-lstm) for recognizing fine-grained activity of drivers," in *Pattern Recognition*, pp. 298–314, 2019.

[247] A. Roitberg, M. Haurilet, S. Reiß, and R. Stiefelhagen, "Cnn-based driver activity understanding: Shedding light on deep spatiotemporal representations," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[248] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. Reiß, M. Voit, and R. Stiefelhagen, "Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2801–2810, 2019.

[249] S. Reiß, A. Roitberg, M. Haurilet, and R. Stiefelhagen, "Deep classification-driven domain adaptation for cross-modal driver behavior recognition," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1042–1047, IEEE, 2020.

[250] A. Behera, Z. Wharton, A. Keidel, and B. Debnath, "Deep cnn, body pose and body-object interaction features for drivers' activity monitoring," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[251] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3182–3190, 2015.

[252] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 3118–3125, IEEE, 2016.

[253] S. Martin, S. Vora, K. Yuen, and M. M. Trivedi, "Dynamics of driver's gaze: Explorations in behavior modeling & maneuver prediction," 2018.

[254] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, "Predicting driver maneuvers by learning holistic features," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 719–724, IEEE, 2014.

[255] A. Doshi and M. Trivedi, "A comparative exploration of eye gaze and head motion cues for lane change intent prediction," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 49–54, IEEE, 2008.

[256] V. A. Shia, Y. Gao, R. Vasudevan, K. D. Campbell, T. Lin, F. Borrelli, and R. Bajcsy, "Semiautonomous vehicular control using driver modeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2696–2709, 2014.

[257] K. Driggs-Campbell, V. Shia, and R. Bajcsy, "Improved driver modeling for human-in-the-loop vehicular control," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1654–1661, IEEE, 2015.

[258] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1108–1120, 2016.

[259] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 340–350, 2007.

[260] Y. Liang and J. D. Lee, “A hybrid bayesian network approach to detect driver cognitive distraction,” *Transportation research part C: emerging technologies*, vol. 38, pp. 146–155, 2014.

[261] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, “Real-time system for monitoring driver vigilance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, 2006.

[262] N. Li and C. Busso, “Predicting perceived visual and cognitive distractions of drivers with multimodal features,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 51–65, 2015.

[263] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, “Online driver distraction detection using long short-term memory,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574–582, 2011.

[264] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ““take over!” how long does it take to get the driver back into the loop?,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, pp. 1938–1942, SAGE Publications Sage CA: Los Angeles, CA, 2013.

[265] B. K.-J. Mok, M. Johns, K. J. Lee, H. P. Ive, D. Miller, and W. Ju, “Timing of unstructured transitions of control in automated driving,” in *2015 IEEE intelligent vehicles symposium (IV)*, pp. 1167–1172, IEEE, 2015.

[266] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler, “How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving,” in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 58, pp. 2063–2067, Sage Publications Sage CA: Los Angeles, CA, 2014.

[267] C. Gold, M. Körber, D. Lechner, and K. Bengler, “Taking over control from highly automated vehicles in complex traffic situations: the role of traffic density,” *Human factors*, vol. 58, no. 4, pp. 642–652, 2016.

[268] M. Körber, C. Gold, D. Lechner, and K. Bengler, “The influence of age on the take-over of vehicle control in highly automated driving,” *Transportation research part F: traffic psychology and behaviour*, vol. 39, pp. 19–32, 2016.

[269] H. Clark and J. Feng, “Age differences in the takeover of vehicle control and engagement in non-driving-related activities in simulated driving with conditional automation,” *Accident Analysis & Prevention*, vol. 106, pp. 468–479, 2017.

[270] S. Petermeijer, P. Bazilinskyy, K. Bengler, and J. De Winter, “Take-over again: Investigating multimodal and directional tors to get the driver back into the loop,” *Applied ergonomics*, vol. 62, pp. 204–215, 2017.

[271] G. Huang, C. Steele, X. Zhang, and B. J. Pitts, “Multimodal cue combinations: a possible approach to designing in-vehicle takeover requests for semi-autonomous driving,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 63, pp. 1739–1743, SAGE Publications Sage CA: Los Angeles, CA, 2019.

[272] E. Dogan, M.-C. Rahal, R. Deborne, P. Delhomme, A. Kemeny, and J. Perrin, “Transition of control in a partially automated vehicle: Effects of anticipation and non-driving-related task involvement,” *Transportation research part F: traffic psychology and behaviour*, vol. 46, pp. 205–215, 2017.

[273] A. Eriksson and N. A. Stanton, “Takeover time in highly automated vehicles: noncritical transitions to and from manual control,” *Human factors*, vol. 59, no. 4, pp. 689–705, 2017.

[274] F. Naujoks, C. Purucker, K. Wiedemann, and C. Marberger, “Noncritical state transitions during conditionally automated driving on german freeways: Effects of non–driving related tasks on takeover time and takeover quality,” *Human factors*, vol. 61, no. 4, pp. 596–613, 2019.

[275] C. Braunagel, W. Rosenstiel, and E. Kasneci, “Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 10–22, 2017.

[276] A. Lotz and S. Weissenberger, “Predicting take-over times of truck drivers in conditional autonomous driving,” in *International Conference on Applied Human Factors and Ergonomics*, pp. 329–338, Springer, 2018.

[277] F. Berghöfer, C. Purucker, F. Naujoks, K. Wiedemann, and C. Marberger, “Prediction of take-over time demand in highly automated driving. results of a naturalistic driving study prediction of take-over time demand in conditionally automated driving-results of a real world driving study,” *Proceedings of the Human Factors and Ergonomics Society Europe*, 2019.

[278] N. Du, F. Zhou, E. M. Pulver, D. M. Tilbury, L. P. Robert, A. K. Pradhan, and X. J. Yang, “Predicting driver takeover performance in conditionally automated driving,” *Accident Analysis & Prevention*, vol. 148, p. 105748, 2020.

[279] S. Hwang, A. G. Banerjee, and L. N. Boyle, “Predicting driver’s transition time to a secondary task given an in-vehicle alert,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[280] E. Pakdamanian, S. Sheng, S. Baee, S. Heo, S. Kraus, and L. Feng, “Deeptake: Prediction of driver takeover behavior using multimodal data,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2021.

[281] K. Yuen, S. Martin, and M. M. Trivedi, “Looking at faces in a vehicle: A deep cnn based approach and evaluation,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 649–654, IEEE, 2016.

[282] K. Yuen and M. M. Trivedi, “Looking at hands in autonomous vehicles: A convnet approach using part affinity fields,” *arXiv preprint arXiv:1804.01176*, 2018.

[283] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017.

[284] A. Tawari, P. Mallela, and S. Martin, “Learning to attend to salient targets in driving videos using fully convolutional rnn,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3225–3232, 2018.

[285] H. Zhu, T. Misu, S. Martin, X. Wu, and K. 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)*, pp. 6210–6216, IEEE, 2021.

[286] R. Greer, A. Gopalkrishnan, J. Landgren, L. Rakla, A. Gopalan, and M. Trivedi, “Robust traffic light detection using salience-sensitive loss: Computational framework and evaluations,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, 2023.

[287] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles,” *arXiv preprint arXiv:2106.11810*, 2021.

[288] H. Wang, Z. Liu, Y. Li, T. Li, L. Chen, C. Sima, Y. Wang, S. Jiang, F. Wen, H. Xu, *et al.*, “Road genome: A topology reasoning benchmark for scene understanding in autonomous driving,” *arXiv preprint arXiv:2304.10440*, 2023.

[289] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural Networks*, no. 0, pp. –, 2012.

[290] C. Ertler, J. Mislej, T. Ollmann, L. Porzi, G. Neuhold, and Y. Kuang, “The mapillary traffic sign dataset for detection and classification on a global scale,” in *European Conference on Computer Vision*, pp. 68–84, Springer, 2020.

[291] A. Møgelmose, D. Liu, and M. M. Trivedi, “Traffic sign detection for us roads: Remaining challenges and a case for tracking,” in *17th International IEEE Conference on Intelligent Transportation Systems*, pp. 1394–1399, 2014.

[292] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-sign detection and classification in the wild,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[293] D. Temel, M. Chen, and G. AlRegib, “Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[294] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," in *Scandinavian conference on image analysis*, pp. 238–249, Springer, 2011.

[295] L. Huang, "Chinese traffic sign database."

[296] V. I. Shakhuro and A. Konouchine, "Russian traffic sign images dataset," *Computer optics*, vol. 40, no. 2, pp. 294–300, 2016.

[297] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, "A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection," *IEEE Access*, vol. 8, pp. 29742–29754, 2020.

[298] J. Cao, J. Zhang, and X. Jin, "A traffic-sign detection algorithm based on improved sparse r-cnn," *IEEE Access*, vol. 9, pp. 122774–122788, 2021.

[299] Á. Arcos-García, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, pp. 158–165, 2018.

[300] N. Fulton, N. Hunt, N. Hoang, and S. Das, "Formal verification of end-to-end learning in cyber-physical systems: Progress and challenges," *arXiv preprint arXiv:2006.09181*, 2020.

[301] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, "Explaining autonomous driving by learning end-to-end visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[302] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," *arXiv preprint arXiv:2005.02545*, 2020.

[303] D. Guo, M. Moh, and T.-S. Moh, "Vision-based autonomous driving for smart city: a case for end-to-end learning utilizing temporal information," in *International Conference on Smart Computing and Communication*, pp. 19–29, Springer, 2020.

[304] G. V. Alves, L. Dennis, and M. Fisher, "A double-level model checking approach for an agent-based autonomous vehicle and road junction regulations," *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, p. 41, 2021.

[305] Y. Guo, W. Feng, F. Yin, T. Xue, S. Mei, and C.-L. Liu, "Learning to understand traffic signs," in *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, (New York, NY, USA), p. 2076–2084, Association for Computing Machinery, 2021.

[306] I. Dua, T. A. John, R. Gupta, and C. Jawahar, "Dgaze: Driver gaze mapping on road," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5946–5953, 2020.

[307] I. Kotseruba and J. K. Tsotsos, "Behavioral research and practical models of drivers' attention," *arXiv preprint arXiv:2104.05677*, 2021.

[308] F. Lateef, M. Kas, and Y. Ruichek, "Saliency heat-map as visual attention for autonomous driving using generative adversarial network (gan)," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[309] J. Su, C. Xia, and J. Li, "Exploring driving-aware salient object detection via knowledge transfer," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2021.

[310] A. Pal, S. Mondal, and H. 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*, pp. 11883–11892, 2020.

[311] C. Li, S. H. Chan, and Y.-T. 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)*, pp. 10711–10718, IEEE, 2020.

[312] Z. Zhang, A. Tawari, S. Martin, and D. Crandall, "Interaction graphs for object importance estimation in on-road driving videos," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8920–8927, 2020.

[313] N. Kulkarni, A. Rangesh, J. Buck, J. Feltracco, M. M. Trivedi, N. Deo, R. Greer, S. Sarraf, and S. Sathyanarayana, "Lisa amazon-mlsl vehicle attributes (lava) dataset," Jun 2021.

[314] R. K. Satzoda and M. M. Trivedi, "Drive analysis using vehicle dynamics and vision-based lane semantics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 9–18, 2015.

[315] K. Clemens, "Geocoding with openstreetmap data," *GEOProcessing 2015*, p. 10, 2015.

[316] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[317] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[318] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 231–236, 2021.

[319] P. Rau, C. Becker, and J. Brewer, "Approach for deriving scenarios for safety of the intended functionality," in *Proc. ESV*, pp. 1–15, 2019.

[320] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural networks*, vol. 32, pp. 323–332, 2012.

[321] M. M. Trivedi, T. L. Gandhi, and K.-S. Huang, “Distributed interactive video arrays for event capture and enhanced situational awareness,” *IEEE Intelligent Systems*, pp. 58–66, 2005.

[322] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda, and P. Hanagal, “Traffic sign detection and recognition using a cnn ensemble,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, 2019.

[323] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, “Traffic sign detection and recognition using fully convolutional network guided proposals,” *Neurocomputing*, vol. 214, pp. 758–766, 2016.

[324] O. N. Manzari, A. Boudesh, and S. B. Shokouhi, “Pyramid transformer for traffic sign detection,” in *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 112–116, 2022.

[325] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, “Evaluation of deep neural networks for traffic sign detection systems,” *Neurocomputing*, vol. 316, pp. 332–344, 2018.

[326] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European Conference on Computer Vision*, pp. 213–229, 2020.

[327] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2021.

[328] P. Koopman, R. Hierons, S. Khastgir, J. Clark, M. Fisher, R. Alexander, and J. A. McDermid, “Certification of highly automated vehicles for use on uk roads: Creating an industry-wide framework for safety,” in *Proc. ESV*, 2019.

[329] A. Tawari, A. Møgelmose, S. Martin, T. B. Moeslund, and M. M. Trivedi, “Attention estimation by simultaneous analysis of viewer and view,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 1381–1387, IEEE, 2014.

[330] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, “On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems,” *Computer Vision and Image Understanding*, vol. 134, pp. 130–140, 2015.

[331] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi, “Unsupervised learning of important objects from first-person videos,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1956–1964, 2017.

[332] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[333] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, “Traffic light detection: A learning algorithm and evaluations on challenging dataset,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2341–2345, IEEE, 2015.

[334] A. Møgelmose, D. Liu, and M. M. Trivedi, “Detection of us traffic signs,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3116–3125, 2015.

[335] S. Gautam and A. Kumar, “Image-based automatic traffic lights detection system for autonomous cars: a review,” *Multimedia Tools and Applications*, pp. 1–48, 2023.

[336] Z. Shi, Z. Zou, and C. Zhang, “Real-time traffic light detection with adaptive background suppression filter,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 690–700, 2015.

[337] K. Behrendt, L. Novak, and R. Botros, “A deep learning approach to traffic lights: Detection, tracking, and classification,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1370–1377, IEEE, 2017.

[338] M. Weber, P. Wolf, and J. M. Zöllner, “Deeptlr: A single deep convolutional network for detection and classification of traffic lights,” in *2016 IEEE intelligent vehicles symposium (IV)*, pp. 342–348, IEEE, 2016.

[339] Z. Ennahhal, I. Berrada, and K. Fardousse, “Real time traffic light detection and classification using deep learning,” in *2019 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–7, IEEE, 2019.

[340] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, “Vision for looking at traffic lights: Issues, survey, and perspectives,” *IEEE transactions on intelligent transportation systems*, vol. 17, no. 7, pp. 1800–1815, 2016.

[341] A. Fregin, J. Muller, U. Krebel, and K. Dietmayer, “The driveu traffic light dataset: Introduction and comparison with existing datasets,” in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3376–3383, IEEE, 2018.

[342] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2020.

[343] J. Zhang, M. Zheng, M. Boyd, and E. Ohn-Bar, “X-world: Accessibility, vision, and autonomy meet,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9762–9771, 2021.

[344] N. Zangi, R. Srour-Zreik, D. Ridel, H. Chasidim, and A. Borowsky, "Driver distraction and its effects on partially automated driving performance: A driving simulator study among young-experienced drivers," *Accident Analysis & Prevention*, vol. 166, p. 106565, 2022.

[345] D. o. H. United States, C. f. D. C. Human Services, and Prevention, "Distracted driving," *Centers for Disease Control and Prevention*, 2022.

[346] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proceedings of the National Academy of Sciences*, vol. 113, no. 10, pp. 2636–2641, 2016.

[347] F. Naujoks, S. Höfling, C. Purucker, and K. Zeeb, "From partial and high automation to manual driving: Relationship between non-driving related tasks, drowsiness and take-over performance," *Accident Analysis & Prevention*, vol. 121, pp. 28–42, 2018.

[348] M. Seeland and P. Mäder, "Multi-view classification with convolutional neural networks," *Plos one*, vol. 16, no. 1, p. e0245230, 2021.

[349] S. Wang, M. E. Celebi, Y.-D. Zhang, X. Yu, S. Lu, X. Yao, Q. Zhou, M.-G. Miguel, Y. Tian, J. M. Gorriz, *et al.*, "Advances in data preprocessing for biomedical data fusion: An overview of the methods, challenges, and prospects," *Information Fusion*, vol. 76, pp. 376–421, 2021.

[350] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[351] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, "Computer vision, iot and data fusion for crop disease detection using machine learning: a survey and ongoing research," *Remote Sensing*, vol. 13, no. 13, p. 2486, 2021.

[352] B. Silva, F. R. Barbosa-Anda, and J. Batista, "Multi-view fine-grained vehicle classification with multi-loss learning," in *2021 IEEE international conference on autonomous robot systems and competitions (ICARSC)*, pp. 209–214, IEEE, 2021.

[353] S. B. Negrete, H. Arai, K. Natsume, and T. Shibata, "Multi-view image-based behavior classification of wet-dog shake in kainate rat model," *Frontiers in Behavioral Neuroscience*, vol. 17, p. 1148549, 2023.

[354] A. B. Khajwal, C.-S. Cheng, and A. Noshadravan, "Post-disaster damage classification based on deep multi-view image fusion," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 4, pp. 528–544, 2023.

[355] L. Wu, A. Chen, P. Salama, K. W. Dunn, and E. J. Delp, "An ensemble learning and slice fusion strategy for three-dimensional nuclei instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1884–1894, 2022.

[356] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[357] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[358] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 641–656, 2018.

[359] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4604–4612, 2020.

[360] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*, pp. 923–932, PMLR, 2020.

[361] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[362] P. K. Chan and S. J. Stolfo, "A comparative evaluation of voting and meta-learning on partitioned data," in *Machine Learning Proceedings 1995*, pp. 90–98, Elsevier, 1995.

[363] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Learning ensembles from bites: A scalable and accurate approach," *The Journal of Machine Learning Research*, vol. 5, pp. 421–451, 2004.

[364] L. Rokach, "Genetic algorithm-based feature set partitioning for classification problems," *Pattern Recognition*, vol. 41, no. 5, pp. 1676–1700, 2008.

[365] K. M. Ting, J. R. Wells, S. C. Tan, S. W. Teng, and G. I. Webb, "Feature-subspace aggregating: ensembles for stable and unstable learners," *Machine Learning*, vol. 82, no. 3, pp. 375–397, 2011.

[366] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, and E. Xun, "Ensemble of deep neural networks with probability-based fusion for facial expression recognition," *Cognitive Computation*, vol. 9, no. 5, pp. 597–610, 2017.

[367] H. G. Ayad and M. S. Kamel, "On voting-based consensus of cluster ensembles," *Pattern Recognition*, vol. 43, no. 5, pp. 1943–1953, 2010.

[368] L. Deng and J. Platt, "Ensemble deep learning for speech recognition," in *Proc. interspeech*, 2014.

[369] G. Brown, J. L. Wyatt, P. Tino, and Y. Bengio, "Managing diversity in regression ensembles.," *Journal of machine learning research*, vol. 6, no. 9, 2005.

[370] S.-W. Lin and S.-C. Chen, "Parameter determination and feature selection for c4. 5 algorithm using scatter search approach," *Soft Computing*, vol. 16, no. 1, pp. 63–75, 2012.

[371] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using bayesian model averaging to calibrate forecast ensembles," *Monthly weather review*, vol. 133, no. 5, pp. 1155–1174, 2005.

[372] T. Huang and V. Merwade, "Uncertainty analysis and quantification in flood insurance rate maps using bayesian model averaging and hierarchical bma," *Journal of Hydrologic Engineering*, vol. 28, no. 2, p. 04022038, 2023.

[373] M. Tian, H. Fan, Z. Xiong, and L. Li, "Data-driven ensemble model for probabilistic prediction of debris-flow volume using bayesian model averaging," *Bulletin of Engineering Geology and the Environment*, vol. 82, no. 1, pp. 1–16, 2023.

[374] S. Fei, Z. Chen, L. Li, Y. Ma, and Y. Xiao, "Bayesian model averaging to improve the yield prediction in wheat breeding trials," *Agricultural and Forest Meteorology*, vol. 328, p. 109237, 2023.

[375] K. Monteith, J. L. Carroll, K. Seppi, and T. Martinez, "Turning bayesian model averaging into bayesian model combination," in *The 2011 International Joint Conference on Neural Networks*, pp. 2657–2663, IEEE, 2011.

[376] M. Gimelfarb, S. Sanner, and C.-G. Lee, "Reinforcement learning with multiple experts: A bayesian model combination approach," *Advances in neural information processing systems*, vol. 31, 2018.

[377] A. Sankar, "Bayesian model combination (baycom) for improved recognition," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1, pp. I–845, IEEE, 2005.

[378] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *Artificial Intelligence and Statistics*, pp. 619–627, PMLR, 2012.

[379] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and computation*, vol. 108, no. 2, pp. 212–261, 1994.

[380] G. Bayrak, A. Küçüker, and A. Yılmaz, "Deep learning-based multi-model ensemble method for classification of pqds in a hydrogen energy-based microgrid using modified weighted majority algorithm," *International Journal of Hydrogen Energy*, 2022.

[381] A. Blum, "Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain," *Machine Learning*, vol. 26, no. 1, pp. 5–23, 1997.

[382] Y. Braouezec, "Committee, expert advice, and the weighted majority algorithm: An application to the pricing decision of a monopolist," *Computational Economics*, vol. 35, no. 3, pp. 245–267, 2010.

[383] P. P. Liang, Y. Lyu, G. Chhablani, N. Jain, Z. Deng, X. Wang, L.-P. Morency, and R. Salakhutdinov, "Multiviz: Towards visualizing and understanding multimodal models," in *The Eleventh International Conference on Learning Representations*, 2022.

[384] L. V. B. Beltrán, J. C. Caicedo, N. Journet, M. Coustaty, F. Lecellier, and A. Doucet, "Deep multimodal learning for cross-modal retrieval: One model for all tasks," *Pattern Recognition Letters*, vol. 146, pp. 38–45, 2021.

[385] S. Wan, Z. Gao, H. Zhang, C. Xiaojun, C. Chen, and A. Tefas, "Editorial paper for pattern recognition letters vsi on cross model understanding for visual question answering," *Pattern Recognition Letters*, vol. 160, pp. 9–10, 2022.

[386] Z. Xue and R. Marculescu, "Dynamic multimodal fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2574–2583, 2023.

[387] E. Ohn-Bar, S. Martin, and M. M. Trivedi, "Driver hand activity analysis in naturalistic driving studies: challenges, algorithms, and experimental studies," *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 041119–041119, 2013.

[388] G. Borghi, E. Frigieri, R. Vezzani, and R. Cucchiara, "Hands on the wheel: A dataset for driver hand detection and tracking," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 564–570, 2018.

[389] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks," *Journal of Advanced Transportation*, vol. 2019, 2019.

[390] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 2020.

[391] P. Weyers, D. Schiebener, and A. Kummert, "Action and object interaction recognition for driver activity classification," pp. 4336–4341, 2019.

[392] L. Yang, T.-Y. Yang, H. Liu, X. Shan, J. Brighton, L. Skrypchuk, A. Mouzakitis, and Y. Zhao, "A refined non-driving activity classification using a two-stream convolutional neural network," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15574–15583, 2020.

[393] J. Wang, W. Chai, A. Venkatachalapathy, K. L. Tan, A. Haghighat, S. Velipasalar, Y. Adu-Gyamfi, and A. Sharma, "A survey on driver behavior analysis from in-vehicle cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10186–10209, 2021.

[394] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, 2019.

[395] H. Chen, R. Feng, S. Wu, H. Xu, F. Zhou, and Z. Liu, "2d human pose estimation: A survey," *arXiv preprint arXiv:2204.07370*, 2022.

[396] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5693–5703, 2019.

[397] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.

[398] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *proceedings of the IEEE international conference on computer vision*, pp. 1281–1290, 2017.

[399] A. Roitberg, K. Peng, Z. Marinov, C. Seibold, D. Schneider, and R. Stiefelhagen, "A comparative analysis of decision-level fusion for multimodal driver behaviour understanding," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1438–1444, IEEE, 2022.

[400] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[401] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[402] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[403] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2019.

[404] R. Greer, L. Rakla, A. Gopalkrishnan, and M. Trivedi, "Multi-view ensemble learning with missing data: Computational framework and evaluations using novel data from the safe autonomous driving domain," 2023.

[405] E. Sachdeva, N. Agarwal, S. Chundi, S. Roelofs, J. Li, B. Dariush, C. Choi, and M. Kochenderfer, "Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning," *arXiv preprint arXiv:2309.06597*, 2023.

[406] A. Dehghani, T. Glatard, and E. Shihab, "Subject cross validation in human activity recognition," *arXiv preprint arXiv:1904.02666*, 2019.

[407] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008–8018, 2021.