Towards Robust Autonomous Driving and Social Robot Navigation via Enhanced Data Utilization

Benjamin Stoler

CMU-CS-25-143

December 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
www.csd.cs.cmu.edu

Thesis Committee:
Jean Oh (Chair)
Sebastian Scherer
Reid Simmons
Jonathan Francis (Bosch Center for Artificial Intelligence)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was in part supported by the Ministry of Trade, Industry and Energy (MOTIE) and the Korea Institute of Advancement of Technology (KIAT) through the International Cooperative R&D programs: P0019782 and P0026022, as well as by Stack AV through research internships which contributed to portions of this thesis.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. or Korean governments, or any other entity.

Keywords: Machine Learning, Robotics, Planning, Prediction, Perception, Autonomous Driving, Social Navigation, Robustness, Safety, Data Utilization, Data Generation, Data Partitioning

Dedicated to my family.

Abstract

Autonomous robots—including self-driving vehicles, sidewalk delivery robots, and more—must navigate among humans in a safe and socially-compliant manner. Current approaches for building and evaluating such autonomous systems rely on data-driven techniques; however, a generalization gap emerges, as methods trained in these traditional paradigms are unable to cope with unexpected real-world scenarios. Therefore, this thesis aims to develop improved methodologies and evaluation settings to increase and assess robustness in autonomous navigation against these challenges, along two key pillars of enhanced data utilization.

First, we introduce scenario characterization and repartitioning schemes, for robustness against out-of-distribution safety-relevant and corner case scenarios. We create a hierarchical characterization method which leverages counterfactual probes to find hidden safety-relevant scenarios in large datasets. We then address the induced generalization gap by incorporating the characterizations into downstream trajectory prediction models' inductive biases. To promote greater interpretability and generalizability, we factorize scenarios into disentangled contexts, creating compositionally novel test sets. We then use modular architectures and auxiliary signals to implicitly reason over and adapt to these settings.

Second, we design targeted scenario modification approaches, to expose and address failure cases and weaknesses of naive autonomy methods. For robustness against perception errors affecting downstream motion prediction, we construct a framework for converting top-down pedestrian trajectory datasets into a more challenging first-person view perspective. We then develop a correction module to account for the resulting errors, trained end-to-end with trajectory prediction approaches. For robustness against adversarial, safety-critical scenarios, we develop a reactive, skill-based adversary policy which leverages a learned, multi-faceted criticality objective to perturb existing scenarios. We then train ego policies in a closed-loop manner against these generated scenarios, demonstrating improved downstream ego performance. Finally, we process and annotate unlabeled and underutilized data sources, to learn human-like behavior from real-world crash videos. We use these learned behavior models to further increase the realism of adversarially perturbed scenarios, as well as the efficacy of closed-loop ego training.

Overall, we find that enhanced data utilization is a key component in developing robust evaluation settings and policy methodologies in autonomous navigation. Because broader machine learning domains exhibit similar data scarcity and out-of-distribution challenges, generalizing these ideas beyond autonomy is likewise promising.

Acknowledgments

I would first like to thank my advisor, Jean Oh, for her steadfast support and encouragement throughout my Ph.D. journey. From taking a chance on me as I transitioned from operating systems research into robotics, to providing endless academic, professional, and personal guidance, I am truly grateful. The work in this thesis would not have been possible without her mentorship.

I would also like to thank Jonathan Francis for his efforts both in research and career discussions. His contributions were invaluable in extending ideas from social robot navigation to autonomous driving in the first place, and in providing much-needed theoretical grounding throughout. I am further grateful to the other members of my thesis committee, Sebastian Scherer and Reid Simmons, for their time, feedback, and many insightful conversations over the years.

I am deeply appreciative of the research collaborators I have worked closely with in this process, for providing their unique perspectives and skills: Soonmin Hwang, Meghdeep Jana, Ingrid Navarro, and Juliet Yang. I additionally thank my colleagues in the roBot Intelligence Group and industry collaborators at Stack AV. Special thanks go to Ian Neal and Baris Kasikci for introducing me to research at the University of Michigan, as well as to Evan Lohn for extensive conversations and musings on machine learning, and being a close friend.

Finally, I want to thank my friends and family for their support in encouraging a healthy work-life balance full of love and adventure: Ryan Goniwiecha, Matias Scharager, Cayden Codel, Joshua Clune, Chad Adelman, Nirjhar Mukherjee, Dylan Caine, and Andrew Wegierski. I am especially grateful to my parents, Joy and Karl; my siblings, Jake and Melissa; and my fiancée, Jada, and our dog, Romeo. Jada's thoughtfulness, kindness, and patience in both this Ph.D. process and beyond has truly meant the world to me, buoying me up from moments of stress and uncertainty, and making moments of peace and joy glow brighter. I feel extremely fortunate for the life I have with my community, and to have such a strong support system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As artificial intelligence (AI) technology advances, more and more autonomous robots are being tasked with operating and navigating among people in shared environments. Such applications span academia and industry, including self-driving vehicles, sidewalk delivery robots, and automated room service in hotels [30, 72, 104, 205]. While these robots can be required to perform a wide variety of interactions with their environments (e.g., lifting a pallet for a warehouse robot, picking up and dropping off a passenger for an autonomous taxi, etc.), an essential challenge for these robots remains the task of autonomous navigation itself, wherein a robot is required to efficiently drive or move to a destination while operating in a manner which is both "safe" (i.e., avoiding collisions and near-misses) as well as "socially compliant" (i.e., behaving non-obstructively and predictably to other entities) [12, 111].

Autonomous robots incur a variety of risks across different task settings, although all are quite severe. Physical risk of injury or death to humans remains a primary concern, as evidenced by multiple fatal collisions caused by autonomous vehicles in recent years [107]. Such high profile incidents also carry the risk of damaging human trust in autonomous robotics more broadly, hampering adoption and delaying potential societal benefits [77, 194]

There are thus several key challenges to be addressed for an autonomously navigating robot, spanning the canonical levels of an autonomy stack [25]. A robot must be adept at sensing and *perceiving* the environment in which it operates, detecting and tracking static and dynamic obstacles and other agents. The autonomous agent must also be able to *predict* the future behavior and intents of relevant external agents, for both humans and other robots. Finally, the agent must intelligently process these intermediate inputs to *plan* its actions and ultimately *control* and actuate the physical hardware. Each layer presents its own challenges and is critical to the overall performance and safety of the system.

Leveraging data has become essential in both developing and evaluating approaches across all levels of the autonomy stack. Because real-world experiments are both expensive and risky, large-scale datasets and high-quality simulators are widely used [154]; with recent advances in machine learning (ML) and AI more broadly, deep learning approaches have become the dominant approach for utilizing this data [49]. In the prototypical data utilization paradigm, datasets comprising recorded human demonstrations are randomly split into training, validation,

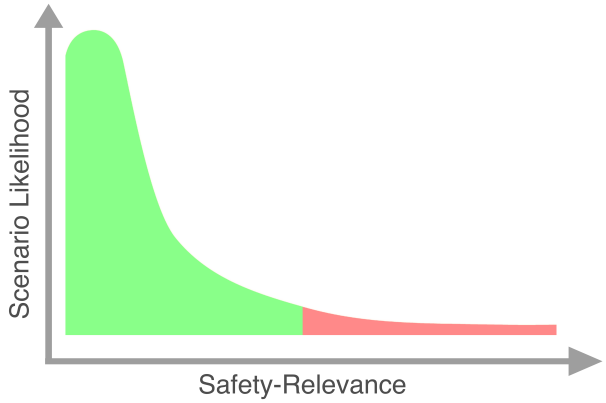


Figure 1.1: Long-tail distribution of scenarios in autonomous navigation. Benign data is far more common than critical data.

and testing sets of scenarios[1], where machine learning models iteratively use the training set to adjust their parameters, the validation set to judge their current progress, and the testing set to assess final performance [6, 54].

However, despite their advantages, pre-recorded datasets and standard data utilization practices also introduce significant challenges. In social navigation and autonomous driving, scenarios tend to follow a heavy-tailed distribution, where safety-critical and other corner cases are exceedingly rare and unlikely to be captured exhaustively during dataset collection, a problem often referred to as the "curse of rarity" [41, 97, 111], as shown in Figure 1.1. Furthermore, datasets and simulations still cannot capture or faithfully reproduce all relevant aspects of real-world scenarios [61]. These limitations lead to generalization and simulation-to-reality (sim2real) gaps, where models which perform quite well in a dataset are "brittle" and non-performant in unexpected real-world scenarios [2, 4], evoking the sentiment among many robotics researchers that "simulations are doomed to succeed" [14].

In this context, improving "robustness"—the ability of a system to maintain performance and safety in the face of these challenges—remains a critical need. By developing and leveraging *enhanced* data utilization approaches that go beyond the aforementioned typical usage frameworks (e.g., artificial distribution shift creation, synthetic long-tail scenario creation, etc.), this thesis aims to fill this need.

1.1 Thesis Statement

Naive utilization of real-world recorded data tends to be insufficient for evaluating and ensuring the robustness of social robot navigation and autonomous driving policies. By adopting enhanced data utilization techniques, policies can achieve greater robustness, maintaining performance across challenges such as out-of-distribution scenarios, perception errors, and adversarial behavior. This thesis explores how these advanced techniques, when paired with increas-

[1] In this thesis, we use the terms "scenario" and "scene" interchangeably.

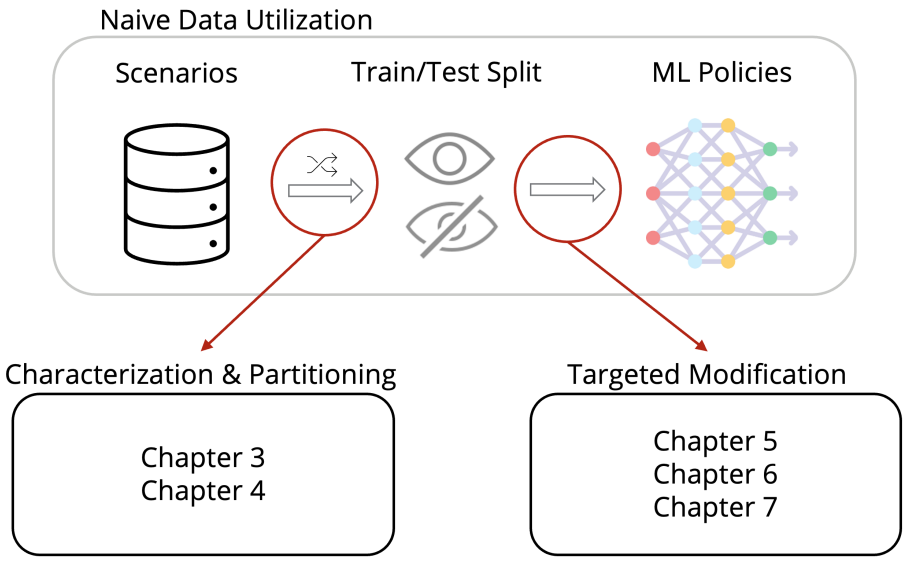


Figure 1.2: This thesis develops "enhanced data utilization" techniques along two key pillars: scenario characterization and partitioning, and targeted scenario modification.

ingly rigorous and realistic evaluation settings, drive methodological advancements—including end-to-end learning, scenario characterization, and adversarial training—to improve robustness across multiple levels of the autonomy stack.

1.2 Contributions and Overview

In this thesis, we present several approaches for enhancing the robustness of different evaluation settings and autonomy policies, across both autonomous driving and social navigation. These developments upon the naive data utilization paradigm are depicted in Figure 1.2. We begin by developing advanced scenario characterization techniques and repartitioning existing datasets—our *first* key pillar of enhanced data utilization. By holding out portions of the datasets as unseen, out-of-distribution test sets, we assess baseline motion prediction approaches and introduce strategies for reducing the incurred drops in performance. We then turn to methodologies for modifying collected data in a realistic and useful way—our *second* key pillar of enhanced data utilization. In social navigation, we reconstruct motion prediction datasets by incorporating perception errors, and develop methods to reduce their impact. Finally, returning to autonomous driving, we contribute approaches for adversarially perturbing background-vehicle behaviors in closed-loop driving simulations, demonstrating that realistic perturbations result in better performing policies. The remainder of this thesis is organized as follows:

In Chapter 2, we introduce common problem formulations, important notation, and key metrics used throughout this thesis. Related work is discussed within each technical chapter in self-contained sections.

In Chapter 3, we mine for hidden safety-relevant driving scenarios, leveraging counterfactual probes and hierarchical scenario characterization to form the basis of these splits. We then

improve motion prediction performance by introducing difficulty-weighted and collision-aware losses. This chapter is based on our IEEE Intelligent Vehicles Symposium (IV) 2024 paper [154], with collaborators Ingrid Navarro, Meghdeep Jana, Soonmin Hwang, Jonathan Francis, and Jean Oh.

In Chapter 4, we factorize driving scenarios into discrete, disentangled ego-centric and social-centric contexts, creating distribution shifts on the basis of compositional novelty of combinations therein. We mitigate the observed drop in motion prediction performance by utilizing modular machine learning architectures and an auxiliary difficulty prediction task. This chapter is based on our paper currently under review with the IEEE Robotics and Automation Letters (RA-L) [155], with collaborators Jonathan Francis and Jean Oh.

In Chapter 5, we re-simulate perfectly annotated social navigation scenarios, captured in a top-down perspective, in an egocentric view with annotations derived from imperfect perception. We then develop a correction module, jointly learned with the downstream motion prediction task, to reduce the impact of these errors. This chapter is based on our IEEE International Conference on Intelligent Robots and Systems (IROS) 2023 paper [153], with collaborators Meghdeep Jana, Soonmin Hwang, and Jean Oh.

In Chapter 6, we adversarially perturb existing driving scenarios by leveraging reactive and human-like skill policies. Through closed-loop training with these generated scenarios, we enhance autonomous driving performance in both safety-critical and normal settings. This chapter is based on our RA-L 2025 paper [156], with collaborators Ingrid Navarro, Jonathan Francis, and Jean Oh.

In Chapter 7, we process unannotated real-world crash videos to learn from human behavior during critical driving scenarios. We then use these insights to ground adversarial perturbation methods and further improve closed-loop autonomous driving performance. This chapter is based on our paper currently under review with the IEEE Transactions on Intelligent Transportation Systems (T-ITS) [157], with collaborators Juliet Yang, Jonathan Francis, and Jean Oh.

Finally, in Chapter 8, we discuss our overall conclusions from this work and highlight some promising future directions.

Chapter 2

Background

We first define the key problem settings, notation, and metrics used throughout this thesis, including both standard formulations as well as our extensions. We primarily focus on two tasks for an autonomous agent—open-loop motion prediction and closed-loop planning in simulation—which use shared dataset sources. We evaluate these tasks in an enhanced data utilization manner via 1) developing artificial distribution shifts by splitting the scenarios into non-uniform train-test sets; and 2) applying perturbations, or modifications, to the scenarios directly in a targeted way.

2.1 Scenario Datasets

In this thesis, we utilize *trajectory* datasets which consist of recorded human demonstrations, as used widely in both social navigation and autonomous driving (AD) [13, 15, 37, 124, 137]. Scenario-based testing via datasets has emerged as a core methodology in autonomy, where alternatives such as on-road testing in AD via a sufficiently large number of miles driven can be prohibitively expensive, risky, and infeasible [75, 101].

We thus consider the set of scenarios that comprise a trajectory dataset as $\mathcal{S}$. We denote $s \in \mathcal{S}$ as a single scenario taken from this corpus, defined as $s = (\mathbf{X}, \mathbf{M}, \texttt{meta})$, where:

i) $\mathbf{X} = \{X_1, X_2, ..., X_N\}$ denotes the trajectories of all scenario participants (i.e., "agents", including pedestrians, cyclists, and vehicles) in s. Each $X_i = \{X_i^{(t)}\}_{t=1}^{T}$ is the trajectory of agent i, consisting of its observed sequence of states over T discrete timesteps, where T is fixed and shared across agents within the scenario. Each state $x_i^{(t)}$ includes the agent's ground-plane position, heading, and velocity, as well as meta information including the type of the agent and a valid bit per timestep indicating whether the agent is present.

ii) $\mathbf{M}$ consists of the relevant *map* context (e.g., road lines, stop sign locations, traffic light locations, etc.). This context includes primarily static information, as fixed locations in the ground-plane, but also includes dynamic elements such as the status of each traffic light at each timestep.

iii) `meta` contains additional meta information that is task-specific. For instance, it may include pairs of agent IDs which are interacting with each other in the scenario, as determined by expert annotators. It may also specify an `ego` ID for the agent which is either the scenario participant which recorded the data in the first place or, more generally, the agent designated as the focus of prediction or control.

Note that depending on the specific dataset and task, some of the above information in s may not be present. For instance, many social navigation datasets neither distinguish between ego and background agents nor provide high-definition map information [124, 137]. Still, recent work, such as UniTraj [40] and trajdata [71], has demonstrated the benefits of adopting a unified format for more effective training and fairer evaluation across datasets.

2.2 Core Autonomy Tasks

2.2.1 Motion Prediction

One key autonomy task is that of motion prediction[1], wherein the future trajectories of agents in a scenario must be predicted, given a brief historical observation. These predictions may be used in the downstream portion of conventional control stacks, to inform an ego-agent's motion planner as it attempts to find possible conflict-free and traffic infraction-free paths. Thus, improving the agent's robustness and its ability to detect possibly safety-critical scenarios is of paramount importance in ensuring the overall acceptable performance of autonomous vehicles in real-world deployments [80].

In the standard motion prediction task, time-varying features in a scenario s are split into a *history* and *future* portion. Given $T = T_{\text{hist}} + T_{\text{fut}}$ representing the total number of timesteps in the scenario, we define $X_i^{\text{hist}} = \{X_i^{(t)}\}_{t=1}^{T_{\text{hist}}}$ as the history portion and $X_i^{\text{fut}} = \{X_i^{(t)}\}_{t=T_{\text{hist}}+1}^{T}$ as the future portion. Let $\mathbf{A}$ be a subset of agents to predict, specified in `meta`. Then, the motion forecasting task is to predict the future ground-plane positions as $\hat{X}_i^{\text{fut}}$ given only $\mathbf{X}^{\text{hist}}$ and $\mathbf{M}$, for each agent $i \in \mathbf{A}$.

Because the ground truth future is available in these datasets, the most commonly used metrics are Average Displacement Error (ADE) and Final Displacement Error (FDE). These metrics can be easily computed on a per-agent basis, for ground truth future track X_i^{fut} and predicted future track $\hat{X}_i^{\text{fut}}$, for each agent i in the scenario. The ℓ_2-distance in the ground plane at each future timestep is then taken; ADE is the average of these distances, while FDE is the final distance. Since future motion is multi-modal, typical models produce K distinct future trajectories along with confidence or probability values that a particular mode is best [9, 150]; in these cases, ADE and FDE are computed only for the mode which best matches the ground truth, in a "best-of-K" style [37, 124, 187].

2.2.2 Closed-Loop Planning

A significant limitation of motion prediction is that it is generally an "open-loop" task; that is, predictions are computed and evaluated after a specified history portion and are not by default updated as the ego agent continues navigating. In contrast, "closed-loop" simulation of an agent is appealing as it (and, optionally, background agents) may react and update their decisions as a scenario unfolds [16, 160].

[1] In this thesis, we use the terms "motion prediction", "motion forecasting", "trajectory prediction", and "trajectory forecasting" interchangeably.

In this thesis, we largely consider the task of re-simulation, where each agent i is instantiated from a base scenario s and follows a behavior functional $\mathcal{B}_i$ in episodic roll-outs. These behaviors may take multiple forms, ranging from fixed trajectories to follow, to reactive policies capable of adapting to others' behaviors while pursuing higher-level objectives. Often, the ego agent assigned by `meta` is controlled using a learning policy and must safely navigate to a given goal, while background agents largely reproduce their original trajectory in s [160, 210]. In a training or evaluation cycle, the k-th sequential roll-out of a base scenario is denoted as $\widetilde{\mathbf{X}}^{(k)}$, where $\widetilde{X}_i^{(k),t}$ refers to the observed position of agent i at timestep t.

Since the ego agent may react to the behavior of other agents during a roll-out, and thus deviate from its original recorded trajectory, direct comparison of $\widetilde{X}_{\text{ego}}^{(k)}$ to the ground truth X_{ego} with ADE and FDE is insufficient. Therefore, task-level metrics, such as the success rate of reaching the goal safely and distributional realism metrics based on kinematic profiles are often utilized [116, 210, 217].

2.3 Enhanced Evaluation Settings

2.3.1 Artificial Distribution Shifts

For a particular evaluation scheme, the total set of all agents in $\mathcal{S}$ must be split to form a training, validation, and test set. In the naive data utilization regime, these agents are split uniformly at random according to some desired size of each set, often at the scenario level of granularity (i.e., all "focal" agents within a given $s \in \mathcal{S}$ are assigned to the same set).

Under distribution shift conditions, $\mathcal{S}$ is instead split into two sets—$\mathcal{S}_{ID}$, representing the in-distribution set, and $\mathcal{S}_{OOD}$ representing the out-of-distribution set—according to some criteria. The task of robust autonomy, then, is to minimize the drop in performance on relevant task metrics (e.g., collision rates, ADE, etc.) when models are tested on $\mathcal{S}_{OOD}$ and $\mathcal{S}_{ID}$, after being trained and validated only on $\mathcal{S}_{ID}$. Furthermore, splits may optionally be performed at the agent level instead of the scenario level, as different focal agents within the same s may experience very different task contexts.

2.3.2 Scenario Modifications

Generating synthetic scenarios and agent behavior therein to address the "curse of rarity" is exceedingly challenging and often results in simulation-to-reality (sim2real) gaps [2, 4]. We therefore focus on targeted *perturbations*, or modifications, of existing scenarios according to carefully-constructed desiderata to expose and mitigate specific weaknesses in naive data utilization paradigms.

We use $\mathcal{P}$ to denote a *perturbation function*, which modifies a base scenario s by altering agent behaviors, possibly conditioned on historical cues. In an open-loop process, this function conditions only on s, that is, the tuple $(\mathbf{X}, \mathbf{M}, \texttt{meta})$. In a closed-loop training or evaluation cycle, this function is permitted to additionally observe all trajectories from up to K past roll-outs on the base scenario, denoted $\{\widetilde{\mathbf{X}}^{(k)}\}_{k=1}^{\leq K}$, each corresponding to a prior episode with a different perturbation of s. These roll-outs are themselves the result of rolling out agents in response to

prior perturbations, forming a sequential interaction history. Concretely, $\mathcal{P}$ is defined as follows:

$$\mathcal{P} : \left(s, \left\{ \widetilde{\mathbf{X}}^{(k)} \right\}_{k=1}^{\leq K} \right) \rightarrow \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_N\}$$

where each behavior functional $\mathcal{B}_i$ follows the definition in Section 2.2.2. The task of robust autonomy in this setting, then, is to maximize task performance on a held-out test set of scenarios, where both training and testing may involve either unmodified or perturbed scenarios, independently.

Chapter 3

Safety-Informed Distribution Shifts

In this chapter, we introduce the first key pillar of enhanced data utilization for robust autonomy: creating artificial distribution shifts based on safety relevance. We demonstrate this approach in the autonomous driving (AD) domain, focusing on motion prediction. As discussed in Chapter 1, large-scale motion prediction datasets such as the Waymo Open Motion Dataset (WOMD) [37] suffer from the "curse of rarity", where safety-critical and other rare events are severely under-represented or entirely absent. Consequently, industry and academia have resorted to validating AD agents via on-road tests [66, 180], where these valuable rare events are also potentially dangerous to other drivers and VRUs, or via simulated experiments [33, 34, 41], wherein the artificial behaviors of agents and inaccurate world physics in the simulators can leave models unprepared and inadequate for real-world deployment [44, 62].

Recently, several works have identified a potential solution to this challenge of robust training, by generating "new" traffic scenarios that serve as training samples for otherwise rare events and/or as difficult test-cases to challenge already-trained models. Unfortunately, despite recent advances in safety-critical scenario generation methods [18, 160, 192], generating non-trivially challenging cases that match the realism, frequency, and difficulty of safety-critical scenarios that agents might encounter in the real world remains an open problem. While we partially address these generation challenges in Chapters 6 and 7, an effective and under-explored alternative lies somewhere in the middle: we propose an approach to mine large-scale datasets of real-world vehicle deployments to find and leverage meaningful *safety-relevant* scenarios that may be hidden in the data. Our key insight is that, in autonomous driving, safety-relevance includes not just scenarios where observed agents act in a safety-critical manner, but also scenarios where agents are able to avoid infractions through proactive maneuvers. Therefore, we propose to leverage counterfactual probes to additionally characterize *what-if* scenarios where these proactive maneuvers were *not* performed. Such fine-grained scenario characterization enables trajectory forecasting models to more easily distill diverse defensive driving skills [148] from existing datasets, e.g., preemptive braking as illustrated in Figure 3.1 (left).

Under this paradigm of scenario characterization, we propose the `SafeShift` framework for identifying and studying the most safety-relevant scenarios in a widely-available autonomous driving dataset. The more extreme scenarios are held out as an out-of-distribution (OOD) test-set, as described in Section 2.3.1, thus acting as a stand-in for the valuable and rare, long-tailed events. In this way, we avoid both the challenges of attempting to generate new safety-critical

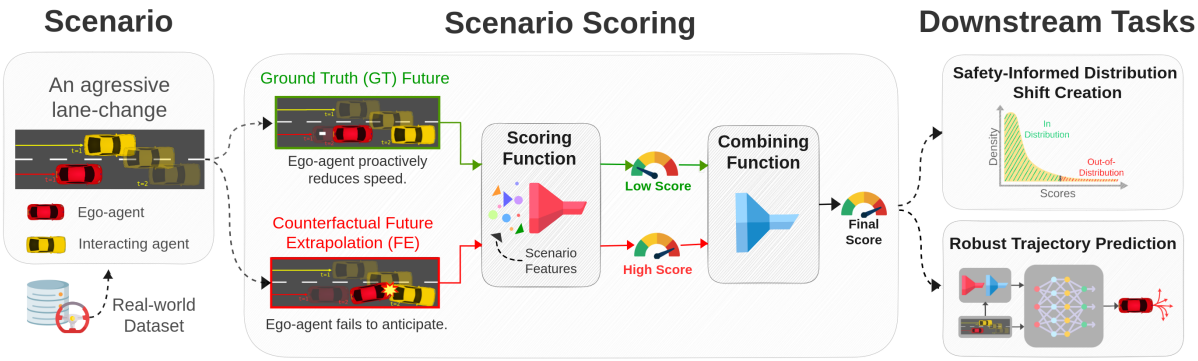


Figure 3.1: An overview of `SafeShift`. Our framework consists of a scoring methodology that uses counterfactual probing to characterize and score scenarios, exploring *what-if* scenarios where proactive maneuvers were not performed, thus resulting in safety-criticality or near misses. We also apply and assesses this scoring approach on two downstream tasks: safety-informed *distribution shift creation*, where challenging scenarios are found and held out for evaluation; and *robust trajectory prediction*, where trajectory prediction algorithms are assessed under this distribution shift and remediated.

scenarios as well as the challenges in performing simulation-to-real transfer; instead, we optimize the usefulness of existing data. To the best of our knowledge, prior work that focuses on creating artificial distribution shifts has not done so based on safety-relevance, instead focusing on, e.g., lane or global location characteristics [42, 200], speed of driving [70], or the city that the data was captured in [70]. Furthermore, prior efforts in scenario characterization under distribution shift settings rely on empirical, dataset-specific heuristics [47, 115, 141].

Our main contributions, illustrated in Figure 3.1, are thus as follows: 1) A versatile approach for scenario characterization in autonomous driving, focused on capturing safety-relevant scenarios; 2) A methodology for scoring safety-criticality for the purposes of crafting a distribution shift, including novel progress in incorporating the aforementioned fuller spectrum of safety-relevance, and improving safety performance therein; and 3) An evaluation of existing socially-aware trajectory prediction approaches in this safety-informed distribution shift setting, utilizing WOMD [37] as an exemplar. Our developed remediation strategy for this setting reduces the predicted trajectories' collision rates by an average of 10%, across the tested models.

This chapter is based on work done with my collaborators [154]. Our code and tools are freely available at `https://github.com/cmubig/SafeShift`.

3.1 Related Work

3.1.1 Socially-Aware Trajectory Prediction

Motion prediction in crowded environments is a well-researched task in the domains of autonomous driving and motion in human crowds [139]. Most current approaches for motion prediction are data-driven, i.e., they focus on characterizing behavior and interactions observed

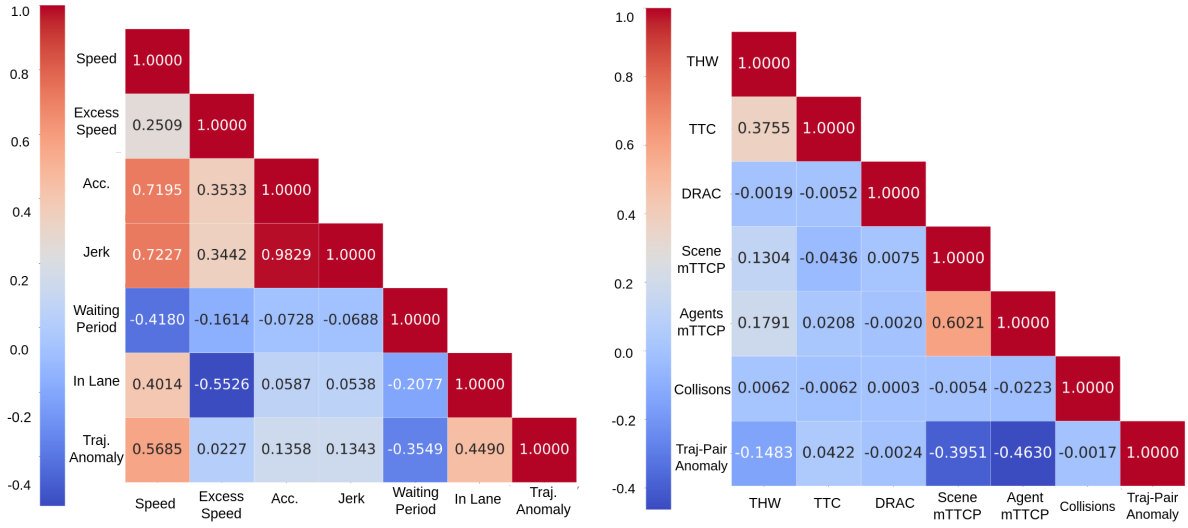
in the data. To capture a multi-modal distribution of possible futures, generative frameworks are frequently used [10, 117, 123, 142, 150, 162]. To model joint behavior and social cues, various techniques such as social pooling [1], rasterized representations [78], and attention-based methods [117, 120, 150, 203] have been employed. Several state-of-the-art techniques have also explored learning richer representations for motion prediction, such as modeling context information as road graphs or polylines [45, 94] and goal conditioning [142].

3.1.2 Robustness Assessment in Trajectory Prediction

One approach to examine robustness for trajectory prediction is robustness to adversarial attacks. Recent studies have shown that state-of-the-art prediction models often lack basic social awareness and collision avoidance when faced with these attacks [140, 182, 211]. A significant disadvantage with these techniques however is that they ultimately rely on simulating realistic agent behavior, which often incurs a simulation-to-real gap [44, 56, 62, 192]. Another approach to ensuring robustness involves studying models' performances under a data domain distribution shift setting, recognizing that AD models will ultimately encounter unseen scenarios in the wild. Some approaches involve identifying domains based on meta characteristics of the scenario, such as road shape characteristics, side-of-driving, and average speeds [42, 70]. Another recent method explores clustering scenarios into domains based on several features, including lane deflection angles, global bounds of the scenario and trajectories, and lane shape information [200]. Many of these works also include domain adaptation or remediation strategies to reduce the impact of the distribution shift, such as by leveraging Frenet coordinates [186, 200], few-shot adaptation [42], or motion-based style transfer [82]. However, to the best of our knowledge, no work has attempted to create distribution shifts based on safety-relevance or study remediation therein.

3.1.3 Critical Scenario Identification in Autonomous Driving

Many existing datasets rely on mining the immense amount of collected data from road-tests for interesting scenarios [15, 21, 37], considering surface-level metrics such as traffic density and kinematic complexity. Therefore, prior frameworks for critical scenario identification (CSI) have been designed to expand upon these initial dataset characterizations [181, 212]. These frameworks typically focus on creating taxonomies for categorizing conflict scenarios, as well as for developing metrics and validation methods to describe and cope with them. In [47], scenarios are instead hierarchically scored along metrics related to anomaly, interestingness, and relevance, better handling more complex maneuvers. Another recent work expands beyond this by defining complexity aspects relating to the road graph layout, surrounding objects, and topology of agents' paths [141]. However, the use of these surrogate metrics for CSI alone, without applying counterfactual reasoning, can fail to identify more subtle safety-relevant scenarios, as illustrated in Figure 3.1. Furthermore, these metrics often rely on empirical weighting and thresholding schemes, as well as on privileged information not uniformly available in AD datasets (e.g., global reference frames, drivable area identification, etc.) [47, 141]; thus they cannot be applied to several key datasets, including WOMD.



(a) Correlation Analysis for Individual Features

(b) Correlation Analysis for Social Features

Figure 3.2: Pearson correlation coefficients for each pair of metrics, showing how the features complement each other. Analysis performed on WOMD [37].

3.2 Scenario Features

We propose a hierarchical scheme as in [47, 141], where low-level, base features are computed within a scenario and then later aggregated to form a score representing a scenario's overall safety-relevance. We consider base features across two main categories: *individual* features related to single-agent behavior and *social* features relevant to the interactions between agents.

For both of these categories, accurate lane assignment is highly important but is nontrivial, e.g., VRUs often do not adhere to lanes. Whereas a simple method of snapping to the best-fitting local lane has been used in previous work [200], we instead leverage a probabilistic approach [145] to find valid lane *sequences* for agents. Additionally, we permit lane assignments based on physically plausible lane deflection angles rather than the lane connectivity graph alone. We excluded some features utilized in previous frameworks and datasets [47, 115], such as driving region-based anomaly detection, that require the knowledge of global, city coordinates which are not generally available across all AD datasets. Instead, to identify anomalies, we utilize a traffic primitive extraction and clustering approach pioneered in [51]. This process produces cluster centers for both single trajectories and trajectory pairs, allowing us to easily measure anomalies.

Individual Features: We primarily focus on metrics derived from relative positional data of a trajectory, such as speed, acceleration, and jerk. We additionally implement metrics to incorporate map context, including waiting period (WP) [207], speed difference with the lane's speed limit, and the percentage of time that the agent is following a lane. Finally, we include a trajectory anomaly value, derived from its distance to the nearest individual traffic primitive cluster.

Social Features: We use widely studied and accepted safety surrogate metrics, as in [47, 149, 171]. These include time headway (THW), time-to-collision (TTC), deceleration rate to avoid

crash (DRAC), and the difference between minimum time to conflict points (ΔmTTCP) in both agent trajectories and road graph locations of interest (e.g., crosswalks, stop signs). We then incorporate a measure of collisions directly, counting situations where two agents' center points or segmented paths overlap at a given timestep. Finally, analogous to the individual trajectory anomaly, we add a trajectory-pair anomaly value using paired traffic primitive clusters.

Our full feature selection, along with a correlation analysis is shown in Figure 3.2. For the individual features, the kinematic-based ones correlate positively, as could be expected, while the other features are largely weakly correlated. Similarly, for the social features, TTC and THW have a weak correlation, as they both involve a leader-follower scenario. The two forms of ΔmTTCP are also relatively strongly correlated, as agent trajectories are required to be somewhat intertwined for both. This analysis implies that the selection and extraction of base features are largely complementary, without excessive overlap in coverage.

3.3 Scenario Scoring

Using the base features described in Section 3.2, we define a safety-relevance scoring function that can characterize a given scenario. We then propose a counterfactual re-scoring approach where we re-characterize the same scenario by taking *what-if* alternatives into account.

3.3.1 Scoring Functions

We start by hierarchically aggregating the base features to create overall trajectory and scenario scores as follows. Let $\mathbf{V}_{ind}$ be the total set of extracted individual features, $\mathbf{V}_{soc}$ be the set of social features, and $v \in \mathbf{V}$ represent a single feature taken from one of these sets (e.g., acceleration, TTC, etc.). Then, let $v_i^{(t)}$ be such an extracted individual feature v for trajectory i at timestep t. Similarly, $v_{i,j}^{(t)}$ denotes a social feature over trajectories i and j together.

To combine these extracted base features, we first convert them to a form in which a larger value corresponds to more safety-relevance (i.e., for features such as speed, we use v directly, but for features such as TTC, we use $1/v$). We then aggregate the individual features into an individual score. We take the maximum value for each metric incurred throughout the trajectory and then linearly combine them according to weights specified in [47]; let these weights be denoted as $\mathbf{W}_{ind}$ and $\mathbf{W}_{soc}$. Then, a trajectory's individual score is expressed in Equation (3.1), where " $\cdot$ " denotes the vector scalar product:

$$\texttt{IndScore}_i = \mathbf{W}_{ind} \cdot \left[\max_t (v_i^{(t)}) \mid v \in \mathbf{V}_{ind} \right] \tag{3.1}$$

Note that we do not perform any sort of value detection thresholding to avoid reliance on empirical decision making. Similarly, for each pair of trajectories, we compute a social score, as follows in Equation (3.2):

$$\texttt{SocScore}_{i,j} = \mathbf{W}_{soc} \cdot \left[\max_t (v_{i,j}^{(t)}) \mid v \in \mathbf{V}_{soc} \right] \tag{3.2}$$

Table 3.1: Trajectory scoring variations.

Variation	IndScore	SocScore
Ground Truth (GT)	$X_{GT}^{(i)}$	$(X_{i,\text{GT}}, X_{j,\text{GT}})$
Future Extrapolated (FE)	$\widetilde{X}_{i,\text{FE}}$	$(\widetilde{X}_{i,\text{FE}}, \widetilde{X}_{j,\text{FE}})$
Asymmetric (AS)	$\widetilde{X}_{i,\text{FE}}$	$(\widetilde{X}_{i,\text{FE}}, X_{j,\text{GT}})$
Combined (CO)	$\max(\texttt{TrajScore}_{\text{GT}}, \texttt{TrajScore}_{\text{FE}})$	
Asymmetric Combined (AC)	$\max(\texttt{TrajScore}_{\text{GT}}, \texttt{TrajScore}_{\text{AS}})$	

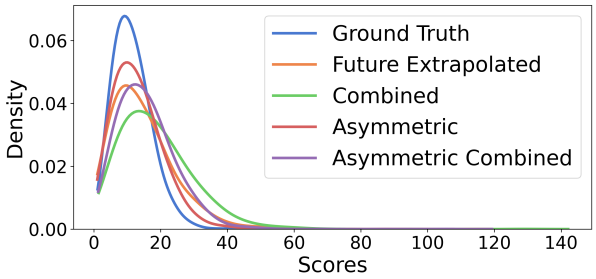


Figure 3.3: PDF of our score variations, exhibiting long-tailed behavior. Analysis performed in WOMD [37].

An agent's trajectory score is then computed by adding together its *individual* score with the *social* score of all trajectory pairs it is involved in:

$$\texttt{TrajScore}_i = \texttt{IndScore}_i + \sum_{j \neq i} \texttt{SocScore}_{i,j} \tag{3.3}$$

We combine these TrajScores into a final ScenarioScore as follows. We begin by taking the weighted sum of all agents' scores in the scenario, where each weight is inversely proportionate to its minimum distance to an agent marked as requiring prediction. Then, to regularize the effect of scenario density, we normalize this total, proportionate to the total number of agents present.

3.3.2 Counterfactual Re-Scoring

The key insight of counterfactual re-scoring is to assess the safety-criticality of a scenario based on potential *what-if* cases in addition to the recorded ground truth event. We hypothesize that the characterization using counterfactual scenarios can capture the hidden risks better than using the ground truth record only, which will subsequently result in improved performance in downstream tasks such as robust trajectory prediction.

To find scenarios beyond just those with high aggregated criticality and/or surrogate criticality values, we wish to perform a counterfactual probe into what could happen if an agent were to simply maintain its current progress within a lane. This represents, e.g., the behavior of a distracted driver ignoring external factors. We craft this probe for an agent i by first extracting its assigned lane sequence in X_i^{hist}. Next, we convert its coordinates to a Frenet frame [186], a coordinate system representing progress and displacements along the given lanes' centerlines.

Finally, we perform a constant-velocity extrapolation in the Frenet frame, to compute a "future extrapolated" trajectory. For agents without a lane assignment, we perform the same steps in Cartesian space. We denote this future extrapolated trajectory as $\widetilde{X}_{i,\text{FE}}$, in contrast with the original ground truth trajectory, $X_{i,\text{GT}}$.

To incorporate this method into the trajectory score in Equation (3.3), we extract the individual and social features of both $X_{i,\text{GT}}$ and $\widetilde{X}_{i,\text{FE}}$. We first compute the individual score using $\widetilde{X}_{i,\text{FE}}$. To compute the social interaction scores, for a pair of interacting agents (i, j), we compute i's social score between $(\widetilde{X}_{i,\text{FE}}, X_{j,\text{GT}})$ and j's score analogously. We denote this *asymmetric* score as $\texttt{TrajScore}_{i,\text{AS}}$. Similarly, we compute the reference ground truth score using exclusively the *GT* trajectories for both agents and denote this as $\texttt{TrajScore}_{i,\text{GT}}$. We then take the maximum value of these two scores into a final *asymmetric combined* trajectory score, $\texttt{TrajScore}_{i,\text{AC}}$. In Table 3.1, we summarize these scoring variations and ablations.

We compute a `ScenarioScore` for these trajectory variations by utilizing the corresponding `TrajScore` (e.g., $\texttt{ScenarioScore}_{\text{FE}}$ uses $\texttt{TrajScore}_{\text{FE}}$ exclusively, etc.). As shown in Figure 3.3, this overall scenario scoring method follows a long-tailed distribution as desired. The scores that incorporate future extrapolation have a much wider spread than just the ground truth, indicating a greater variety of scenarios captured.

3.4 Downstream Tasks

We showcase the utility of our scenario scores from Section 3.3 by applying them to two downstream tasks: 1) creating a safety-informed distribution shift to better evaluate trajectory prediction models; and 2) leveraging the scores to conduct remediation on such models, reducing the incurred drop in performance.

3.4.1 Distribution Shift Creation

We wish to evaluate and improve the robustness of trajectory prediction models when facing scenarios more challenging/safety-critical than those on which they were trained. That is, we must split $\mathcal{S}$ in such a way that $\mathcal{S}_{ID}$ contains relatively low safety-criticality while $\mathcal{S}_{OOD}$ contains the most criticality. Thus, we propose the following approach of splitting $\mathcal{S}$ into the desired safety-informed subsets.

First, we implement a simple uniform, random training/validation/test split to analyze behavior absent of a domain shift context: `Uniform`. Next, as a baseline, we implement the cluster-based domain identification schema from [200], representing a recent approach for domain shift creation that focuses on other aspects of the scenarios instead of safety-relevance: `Clusters`. Finally, we incorporate a safety-informed approach leveraging our schema described in Section 3.3: `Scoring`. We hold out the top 20% scoring scenarios as the test set, then randomly partition the remaining scenarios into training and validation.

3.4.2 Robust Trajectory Prediction

We propose a remediation strategy leveraging the proposed scores in Section 3.3 to increase downstream prediction model performance on challenging, more safety-relevant scenarios. Inspired by the difficulty-weighting of samples, as discussed in [219], we utilize $\texttt{TrajScore}_{i,\text{AC}}$ for each agent i to linearly weigh its contribution to a prediction model's loss function, out of the N total agents in a mini-batch:

$$\texttt{WeightedLoss} = \frac{1}{N}\sum_{i}^{N} \texttt{Loss}_i * \texttt{Score}_{i,\text{AC}} \tag{3.4}$$

Equation (3.4) is then applied after computing the loss function for a given model, but before invoking the optimization pass. This encourages the model to not treat all scenarios and agents' trajectories as equal and to care about more safety-relevant situations. Next, because the future extrapolated score depends only on information available in $\mathbf{X}^{\text{hist}}$, we can incorporate $\texttt{TrajScore}_{i,\text{FE}}$ into a model directly, to add a sense of counterfactual understanding to its inductive biases. We encode this score for each agent i with a simple multilayer perceptron. Then, we concatenate this feature directly with the context encoding representation used in each model (i.e., a function of trajectory histories, lane embedding, etc.) before passing it to the model's trajectory decoding stage.

We also propose to incorporate a collision-aware loss objective within each model. Many models in AD trajectory forecasting produce multi-modal futures, where they output K possible future modes for each agent, along with a scalar, confidence value for each [78, 150, 162]. We add in a cross-entropy (CE) loss objective upon these confidence values, where the "correct" mode is the mode that minimizes collisions with other agents' ground truth futures. In the case where a model already has a CE loss objective (e.g., to minimize the distance to the agent's ground truth future), we linearly weigh the two target values according to a regularization parameter.

3.5 Experimental Setup

Dataset: We utilize WOMD [37] as an exemplar dataset to validate our approach, as it contains a particularly wide variety of scenarios. This variety is highlighted in terms of both geographic and roadway diversity, as well as scenario complexity and traffic density [98, 179]. We utilize a subset from the publicly available training and validation sets from WOMD, consisting of roughly $170k$ scenarios. We consider our three different data splits (Section 3.4.1)—`Uniform`, `Clusters`, `Scoring`—to create $\mathcal{S}_{ID}$ for training and validation (roughly $135k$ scenarios), and $\mathcal{S}_{OOD}$ for testing (roughly $35k$ scenarios).

Baselines: We implement two representative baseline models to validate the efficacy of our distribution shift and remediation strategies. First, we include MTR [150], which, as of this writing, is the current top-performing model on WOMD leaderboards. Second, we implement a version of A-VRNN [117], where we utilize social pooling [1] instead of a graph attention layer for the hidden state refinement. While both models are designed to be "socially-aware," neither is explicitly structured to predict safe futures. We follow the same training procedure performed

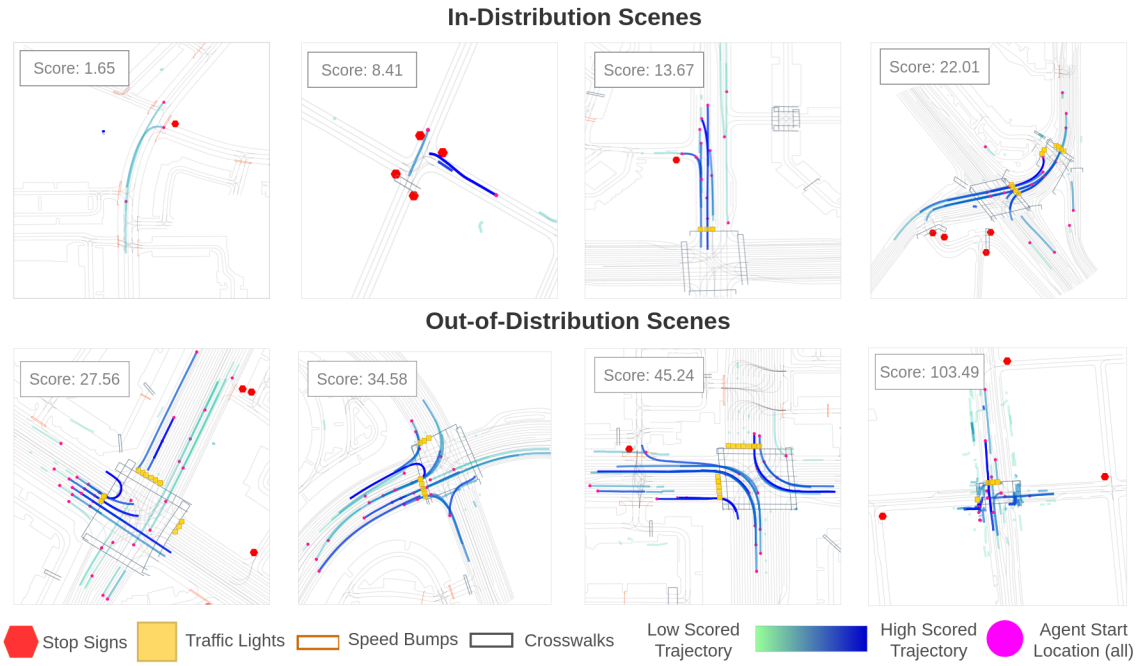


Figure 3.4: Examples of WOMD [150] scenarios by score. In-Distribution and Out-of-Distribution follow our `Scoring` split in Section 3.4.1.

by MTR, where the models are trained for 30 epochs, and learning rate reduction begins after epoch 20.

As a baseline remediation strategy, we implement the Frenet-based domain normalization approach in [200]. This approach converts all coordinates into a trajectory's Frenet frame, before passing the coordinates to a trajectory prediction model. In order to obtain reasonable performance, we use both the Cartesian and Frenet coordinates *together* via concatenation, rather than replacing the former with the latter. We then implement our proposed remediation approach, described in Section 3.4.2 for both models.

Metrics: To measure safety-criticality, we use collision rate (CR), as the average number of collisions of each predicted trajectory to the ground truth of the other agents, as in [81], where collisions with the same external agent over multiple timesteps only count once. We also utilize standard trajectory prediction metrics, as used in the WOMD challenge, including ADE and FDE, along with mean Average Precision (mAP). This final metric categorizes predicted modes into buckets (e.g., straight, stationary, u-turn, etc.), and punishes mode collapse for overlapping predictions.

Table 3.2: Distribution shift experiments in WOMD [37]. ADE / FDE is in meters. Δ_{val} is the change in test collision rate (CR) from the corresponding val CR. A more drastic **increase** is better.

Data Split	Method	Validation Set (In-Distribution)			Testing Set (Out-of-Distribution)		
		ADE / FDE	mAP	CR	ADE / FDE	mAP	CR (Δ_{val})
`Uniform`	GT	- / -	-	0.008	- / -	-	0.009 ($+12.5\%$)
	MTR [150]	0.73 / 1.58	0.30	0.062	0.73 / 1.59	0.31	0.061 (-1.60%)
	A-VRNN [117]	1.80 / 4.63	0.06	0.057	1.82 / 4.67	0.06	0.058 ($+1.80\%$)
`Clusters` [200]	GT	- / -	-	0.009	- / -	-	0.007 (-22.2%)
	MTR	0.69 / 1.50	0.35	0.060	0.71 / 1.55	0.33	0.051 (-15.0%)
	A-VRNN	1.79 / 4.59	0.08	0.062	1.82 / 4.70	0.07	0.049 (-21.0%)
`Scoring` (Ours)	GT	- / -	-	0.005	- / -	-	**0.017** ($\mathbf{+240\%}$)
	MTR	0.72 / 1.59	0.32	0.044	0.74 / 1.59	0.30	**0.100** ($\mathbf{+127\%}$)
	A-VRNN	1.99 / 5.26	0.05	0.042	2.13 / 5.55	0.05	**0.099** ($\mathbf{+136\%}$)

GT: Ground truth tracks

Table 3.3: Robust trajectory prediction experiments in WOMD [37]. ADE / FDE is in meters. Δ_{test} is the change in test CR from the *un-remediated* test CR for each method. A more drastic **decrease** is better.

Data Split	Method	Validation Set (In-Distribution)			Testing Set (Out-of-Distribution)		
		ADE / FDE	mAP	CR	ADE / FDE	mAP	CR (Δ_{test})
`Scoring` (Ours)	GT	- / -	-	0.005	- / -	-	0.017 (-)
	MTR	0.72 / 1.59	0.32	0.044	0.74 / 1.59	0.30	0.100 (-)
	MTR + F+ [200]	0.73 / 1.59	0.32	0.043	0.75 / 1.59	0.30	0.099 (-1.00%)
	MTR + Ours	0.83 / 1.80	0.25	0.037	0.89 / 1.91	0.22	**0.086** ($\mathbf{-14.0\%}$)
	A-VRNN	1.99 / 5.26	0.05	0.042	2.13 / 5.55	0.05	0.099 (-)
	A-VRNN + F+	2.05 / 5.24	0.06	0.041	2.23 / 5.73	0.06	0.103 ($+4.04\%$)
	A-VRNN + Ours	1.76 / 4.61	0.06	0.039	1.91 / 4.94	0.06	**0.093** ($\mathbf{-6.06\%}$)

GT: Ground truth tracks, **F+**: Frenet+ Strategy [200]

3.6 Results

3.6.1 Distribution Shift Results

In Figure 3.4, we highlight some examples of scenarios identified in $\mathcal{S}_{ID}$ and $\mathcal{S}_{OOD}$ for our `Scoring` method described in Section 3.3. The ID scenarios contain both simple scenarios with few interactions, as well as moderately safety-relevant scenarios with lane changes and intersections. The OOD scenarios appear significantly more safety-relevant, with more diverse maneuvers, such as u-turns, larger and more dangerous intersections, and many more VRUs navigating alongside vehicles.

Our quantitative results for the trajectory prediction experiments are summarized in Table 3.2. The metric values reported are averaged over the three classes of vehicles, pedestrians, and cy-

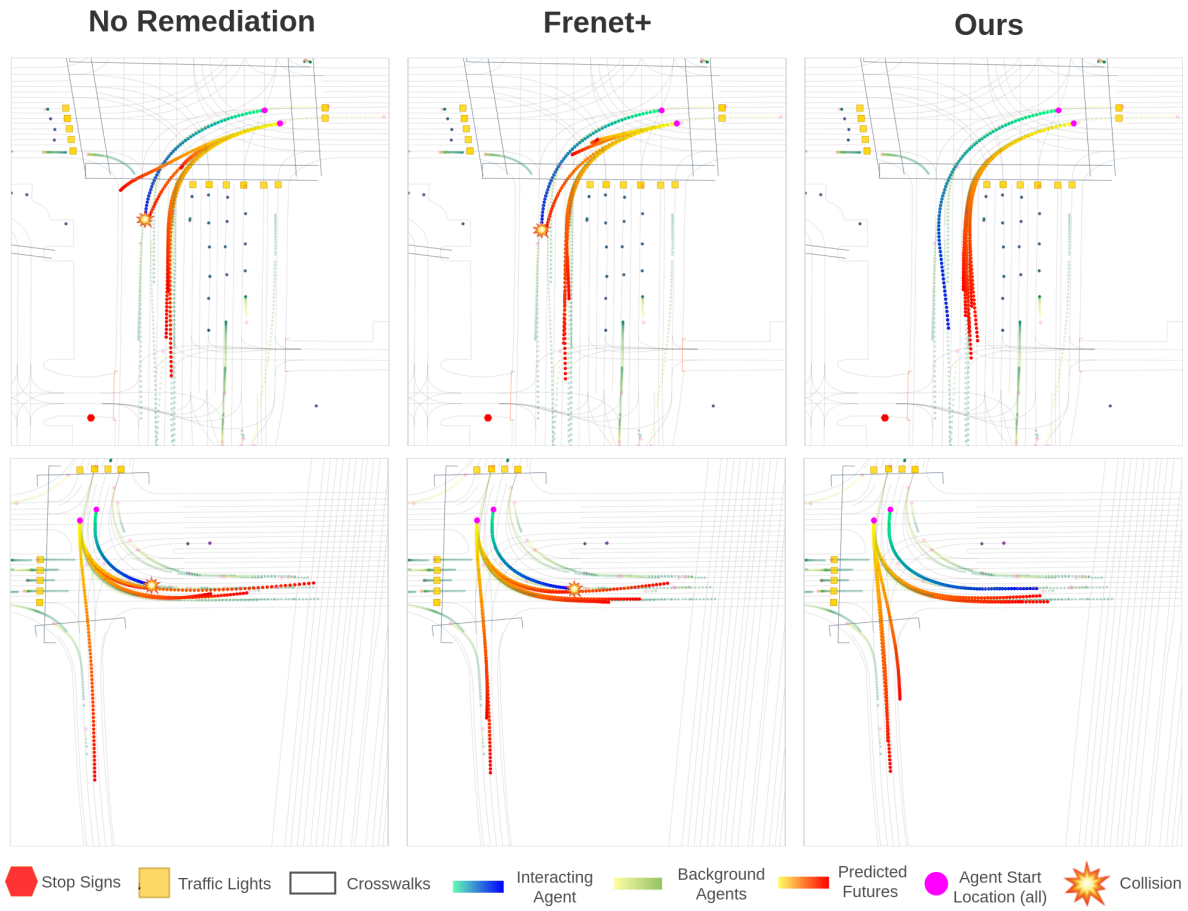


Figure 3.5: Qualitative examples of remediation approaches applied to MTR across two distinct scenarios. Trajectories progress from the pink starting points.

clists. The Δ_{val} value in the final column indicates the increase in collision rate in the OOD test value compared to the ID validation value. In the `Uniform` split, as expected, results between $\mathcal{S}_{ID}$ and $\mathcal{S}_{OOD}$ are quite similar. For the `Clusters` [200] split, we note that while a slight drop in metric performance for ADE / FDE and mAP occurred, the collision rate actually *decreased* from validation to test. We suspect this is because the domains identified by this strategy have no sense of safety-criticality, affirming the importance of using such metrics when selecting scenarios. Finally, our `Scoring` strategy resulted in the largest increase in collision rates between $\mathcal{S}_{ID}$ and $\mathcal{S}_{OOD}$, both in terms of absolute value and percentage change. This increase occurs in both the ground truth tracks, as well as in our tested methods, more than doubling the in-distribution rates.

3.6.2 Robust Trajectory Prediction Results

We show our remediation experiment results in Table 3.3. Our proposed method was the most effective in reducing collisions for the tested models, as shown by the Δ_{test} values. For MTR in

Table 3.4: Scoring strategy ablation study. Results are from using MTR [150] on WOMD [37]. ADE / FDE is in meters. Δ_{val} is the change in test CR from val for the given method. The best distribution shift result is **bolded**.

Ablation Name	Scoring Strategy			Validation Set (In-Distribution)			Testing Set (Out-of-Distribution)		
	GT	FE	AS	ADE / FDE	mAP	CR	ADE / FDE	mAP	CR (Δ_{val})
Ground Truth	✓	-	-	0.72 / 1.57	0.32	0.041	0.75 / 1.64	0.29	0.088 ($+115\%$)
Future Extrapolated	-	✓	-	0.73 / 1.61	0.33	0.046	0.72 / 1.55	0.31	0.097 ($+111\%$)
Combined	✓	✓	-	0.73 / 1.60	0.32	0.048	0.74 / 1.59	0.29	0.098 ($+104\%$)
Asymmetric	-	✓	✓	0.73 / 1.61	0.33	0.044	0.73 / 1.58	0.30	0.099 ($+125\%$)
Asymmetric Combined	✓	✓	✓	0.72 / 1.59	0.32	0.044	0.74 / 1.59	0.30	**0.100 ($+127\%$)**

GT: Ground truth, **FE**: Future extrapolated, **AS**: Asymmetric scoring.

Table 3.5: Remediation strategy ablation study based on our proposed approach in Section 3.4.2 utilizing MTR [150] on WOMD [37]. ADE / FDE is in meters. Δ_{test} is the change in test CR from the *un-remediated* MTR test CR.

Ablation Name	Remediation		Validation Set (In-Distribution)			Testing Set (Out-of-Distribution)		
	SC	CL	ADE / FDE	mAP	CR	ADE / FDE	mAP	CR (Δ_{test})
MTR [150]	-	-	0.72 / 1.59	0.32	0.044	0.74 / 1.59	0.30	0.100 (-)
MTR + Ours (SC only)	✓	-	0.74 / 1.63	0.31	0.046	0.74 / 1.61	0.29	0.103 ($+3.00\%$)
MTR + Ours (CL only)	-	✓	0.81 / 1.77	0.27	0.038	0.88 / 1.92	0.23	0.093 (-7.00%)
MTR + Ours (Full)	✓	✓	0.83 / 1.80	0.25	0.037	0.89 / 1.91	0.22	**0.086 (-14.0%)**

SC: Score incorporation, **CL**: Collision loss objective.

particular, we observe the test collision rates are lowered by 14%, while for A-VRNN, the rates decrease by 6%. This resulted in an average decrease of 10%, reducing the gap to the ground truth collision rate. However, our method does result in a slight decrease in performance on other metrics for MTR. This is likely because MTR has an existing CE loss to select the best mode based on these other metrics, meaning the collision loss objective is in contention with its original objective.

Furthermore, the Frenet+ strategy [200] appeared ineffective in remediating the drop in performance on the `Scoring` data split. We suspect this is due to the presence of more object types than just vehicles; cyclists and pedestrians are often not in lanes, so incorporating such lane information may have been more harmful than beneficial. Additionally, even for vehicles following well-defined lanes, the Frenet+ strategy can still incur collisions, particularly at intersections and unprotected turns.

To gain further insight into the benefits of both the Frenet+ strategy and our remediation approach, we provide qualitative examples in Figure 3.5 using MTR as the prediction model. In these scenarios, the prediction with no remediation results in future modes that collide with an external agent. Meanwhile, the Frenet+ strategy is able to better stay in lanes than the un-remediated approach but still results in collisions. Finally, our remediation approach is able to avoid collisions, while still providing reasonable mode diversity and lane conformance.

3.6.3 Ablation Studies

As shown in Table 3.4, we performed a distribution shift ablation study focusing on the five variations of our scoring strategy discussed in Section 3.3. We utilized MTR as it is the best model according to traditional metrics. Our full method, with asymmetric combined scoring, resulted in the largest increase in collision rate, while still incurring a moderate increase in the other metrics. This result confirms our hypothesis from Section 3.3.2 that our counterfactual probing technique indeed captures a fuller spectrum of safety-relevant scenarios.

We also performed an ablation study focusing on aspects of our remediation strategy, as shown in Table 3.5. While the collision loss objective alone was quite effective, the best performance was achieved utilizing our full approach, incorporating the scores as part of the models' inductive biases and loss weights as well.

3.7 Discussion

Developing autonomous driving trajectory prediction models through real-world datasets, such as WOMD, is often considered insufficient for ensuring robustness and safety. While such datasets provide realistic recorded scenarios, they rarely contain truly safety-relevant scenarios, falling victim to the "curse of rarity." Still, we proposed to further characterize these datasets and find hidden safety-relevant scenarios therein. We thus provided a versatile scenario characterization approach to score scenarios by a hierarchical combination of complementary individual and social features. By performing a counterfactual probe, emulating how a distracted agent may operate, we extended the spectrum of safety-relevance to additionally find hidden risky scenarios, without requiring unrealistic simulation or dangerous real-world testing.

Under a distribution shift setting where the most safety-relevant scenarios were held out as out-of-distribution, we demonstrated that both ground truth, as well as our evaluated trajectory prediction models, incurred a significant increase in collision rates. We further contributed a remediation strategy, achieving a 10% average reduction in prediction collision rates.

Although our remediation strategy proved successful in reducing the test collision rate, the drop in performance was not remediated completely. Incorporating test-time refinement and collaborative sampling techniques, as highlighted in contemporaneous work, could prove a fruitful direction in improving this strategy further [80]. Another interesting future direction of this work would be to utilize our scoring strategy to assess safety-critical scenarios generated in simulation along the axes of realism, frequency, and type of safety-relevance created. Overall, we argue that trajectory prediction datasets remain valuable in assessing safety in autonomous driving, and encourage future work to further this direction.

Chapter 4

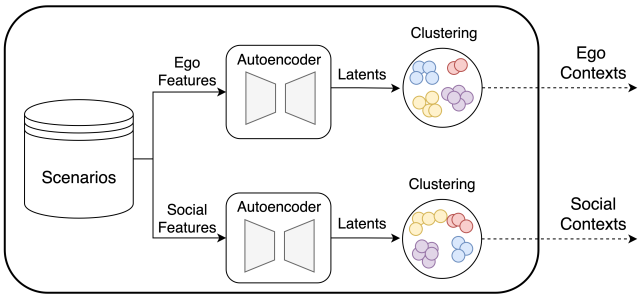
Long-Tail Compositional Zero-Shot Generalization

In this chapter, we further develop the artificial distribution shift pillar of enhanced data utilization, introduced through the `SafeShift` framework in Chapter 3, while continuing to focus on autonomous driving (AD) motion prediction. `SafeShift`, along with other recent work, emulates out-of-distribution (OOD) deployment settings by repartitioning existing datasets into non-uniform train and test splits based on e.g., safety-relevance scores, as in Chapter 3, or cluster identities [201]. However, these bases are often highly entangled (i.e., single axes that conflate many underlying attributes), which results in interpretation and generalization challenges, as emphasized in the broader machine learning literature [7, 85, 177].

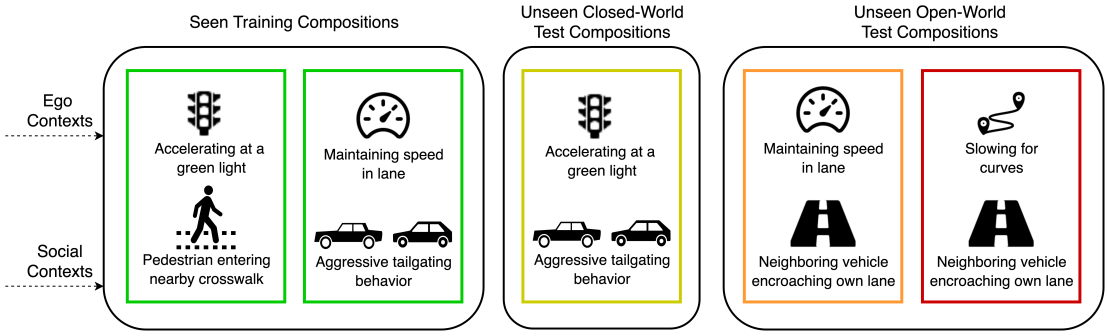
Recently, work in related machine learning tasks, such as labeling objects in images, has developed a relevant framework known as compositional zero-shot learning (CZSL) [109, 127]. In CZSL, object labels in images consist of pairs of object types and attributes. The zero-shot challenge is to train labeling models on some in-distribution (ID) *seen* subset of these pairs and test on both seen and OOD *unseen* pairs. These settings and resulting methodologies have demonstrated finer-grained evaluation of capabilities and improved model generalizability [3, 109]. It is thus appealing to extend the CZSL setting to AD tasks, especially given both the hierarchical manner in which humans operate vehicles [114], as well as the causal factorization of crash risks therein [23, 152]. In the context of driving, however, behavior generalization along safety-relevant, semantically meaningful axes has not yet been explored.

To address this gap, we propose and assess a safety-informed scenario factorization approach, enabling both the creation of challenging OOD evaluation settings and the development of robust, generalization methods for AD motion prediction. Existing studies on human driver error and risk assessment have organized contributing factors into pertinent categories: road infrastructure (road layout, signage quality), vehicle-related issues (mechanical reliability, maintenance), road user conditions (driver experience, mental state), behaviors of other road users (aggression, erraticism), and environmental conditions (lighting, weather) [23, 152]. Unfortunately, large motion prediction datasets typically lack much of this detailed information [37, 187].

We thus derive two axes from available indicators: an "ego" context, capturing both *general* and *safety-relevant* factors (such as kinematics, map features, and deviation from the speed limit), and a "social" context, capturing relative kinematics as well as safety-criticality indicators



(a) Scenario factorization into paired ego and social contexts.



(b) Illustrative toy examples of context compositions.

Figure 4.1: Overview of our framework. (a) Traffic scenarios are factorized and clustered along explicitly disentangled ego and social axes. (b) These contexts are then used to create challenging compositional zero-shot evaluation settings for trajectory prediction, and to enable generalization strategies to enhance OOD robustness.

(including closing speeds and minimum time-to-conflict point difference (mTTCP)), as shown in Figure 4.1a. We then create long-tail, zero-shot compositional closed-world and open-world settings on top of these paired contexts, with these axes serving as analogues to the object and attribute axes in the image-labeling CZSL setting. These settings entail splitting data into seen and unseen portions non-uniformly, holding out novel combinations of ego and social contexts to be tested on, as shown in Figure 4.1b. Note that, unlike the established image-labeling setting, our evaluation focuses on downstream task performance *within* these novel context combinations, rather than predicting a label for a scenario; to our knowledge, this is the **first** extension of CZSL concepts to trajectory prediction.

In these closed-world and open-world settings, we observe a significant OOD performance gap in trajectory prediction, when using WOMD [37] as an exemplar dataset and MTR [150] as a baseline prediction approach. To enhance OOD performance, we then develop and extend domain generalization techniques from the image-labeling CZSL setting. In particular, we adapt task modular gating networks [128] to operate directly in the bottleneck layer of baseline approaches and further enhance the intermediate representation with an auxiliary, difficulty-prediction head.

Our contributions are thus as follows: 1) We introduce a novel, safety-informed scenario fac-

torization approach for autonomous driving, leveraging explicitly disentangled "ego" and "social" axes; 2) We propose new long-tail, zero-shot closed-world and open-world generalization settings upon these axes, where the difficulty-balanced prediction error in the closed-world test setting increases by an average of 5.0% (closed-world) and 14.7% (open-world) over their respective in-distribution performance; and 3) We develop generalization techniques that together reduce these OOD performance gaps to 2.8% and 11.5% respectively, eliminating nearly half the gap in closed-world and one quarter in open-world settings, while improving ID performance by 4.0% and 1.2%.

This chapter is based on our paper currently under review, with a preprint available [155]; upon publication, our code and tools will be made freely available at `https://github.com/cmubig/LongComp`.

4.1 Related Work

4.1.1 Compositional Zero-Shot Learning

Compositional zero-shot learning (CZSL) is increasingly used in computer vision image-labeling as a framework for evaluating human-like generalization to out-of-distribution examples [127]. In the canonical CZSL setting, models are tasked with labeling novel combinations of semantic factors (i.e., object-attribute pairs) not observed during training [109]. To reduce the generalization gap incurred, various strategies, such as task modular architectures [128], conditional attribute learning [175], and attention propagation [76], have demonstrated promising performance. More recently, approaches leveraging large-scale pre-trained foundation models such as CLIP [5, 119], as well as retrieval-augmented generation (RAG) [73], have achieved substantial improvements therein.

In parallel, these evaluation settings and generalization strategies have also recently been extended from static image-labeling to video *action*-labeling, with verb prompts serving as an analogue to attributes [90, 199]. Nevertheless, our focus on both safety-relevant, semantically meaningful factorizations, as well as behavior *generation* under these conditions rather than just labeling, remains largely unexplored. Furthermore, foundation models are less directly applicable in AD, where the heavy-tailed nature of driving behaviors makes large-scale pre-training alone less effective, especially in distribution shift settings [31, 97].

4.1.2 Scenario Characterization and AD Evaluation

For rigorously developing AD systems, scenario characterization and mining approaches are essential, enabling efficient and effective training protocols and more structured evaluation procedures. Identifying scenarios featuring high levels of interaction between traffic participants is particularly common in large dataset curation [37, 115, 187], ensuring that training and evaluation examples are sufficiently challenging. Recent approaches have expanded on this to introduce additional nuanced scenario characteristics, such as inter-agent causality [138], surprise potential [36], and calibrated regret [118]. UniTraj [40] further proposes a stratified evaluation procedure along Kalman-difficulty bins and agent trajectory shapes. Other approaches have explicitly

focused on safety-relevance through automated analyses to capture near-critical scenarios, even while truly safety-critical, long-tail scenarios are absent from recorded datasets [47, 154]. While much of this mining has historically been performed heuristically or with clustering, recent approaches have also explored using large language models and vision language models to identify challenging scenarios and obtain structured scenario descriptors [161, 198, 216].

Beyond scenario understanding, non-uniform training and testing procedures have also been increasingly utilized. Prior work has explored creating artificial distribution shifts on various bases, including road geometry [42], clustered "domains" stemming from agent trajectory and map statistics [200, 201], and overall safety-relevance [154]. Although some prior work, like [176], has studied compositional OOD generalization along axes like weather and time-of-day, these surface-level axes do not consider the diverse behavior-level attributes that make AD uniquely challenging. Concurrently, many works have explored reducing the observed drops in performance, through techniques like Frenet-based domain normalization [200], collision avoidance losses [154], and closed-loop training with synthetic generated scenarios [156, 210]. However, the entangled nature of such prior domain split techniques makes both performance analysis and broader generalization challenging, with remediation techniques often tailored to specific settings. We instead disentangle scenarios along meaningful, safety-informed axes (i.e., ego and social contexts), and extend modular CZSL techniques to target the gap arising from long-tail compositional challenges.

4.2 Preliminaries

In this section, we define relevant notation and task definitions for compositional zero-shot evaluation and generalization in trajectory prediction. Concretely, we utilize our pipeline in Section 4.3.1 and Section 4.3.2 to produce discrete "ego" and "social" context labels for agents, though the definitions and notations here are independent of those details.

Scenario and Data Format: We extend the base scenario definition from Section 2.1, adopting the unified data format and features defined in UniTraj [40]. That is, a particular $X_i^{(t)}$ data point contains ground-plane kinematics (position, velocity, and acceleration), a valid bit indicating whether the agent is present, as well as one-hot encodings both of the agent's type and the current timestep; i.e., an enriched trajectory representation. Information in $\mathbf{M}$ consists of polyline representations of road and traffic control devices, with locations interpolated at a fixed distance interval and a one-hot encoding of the feature type (e.g., lane, stop-sign, crosswalk, etc.), along with other relevant meta information (e.g., speed limit of lanes, lane adjacency graphs, etc.).

Compositional Zero-Shot Trajectory Prediction: In our CZSL setting, we create splits at the agent level, as described in Section 2.3.1, where an agent-centric scenario is created for a given "focal" agent in $\mathbf{A}$. We first obtain paired categorical ego and social context labels from all agents across $\mathcal{S}$. We denote these contexts as $\mathcal{C}_{\text{ego}} = \{c_e^k\}_{k=1}^{N_{\text{ego}}}$ and $\mathcal{C}_{\text{social}} = \{c_s^k\}_{k=1}^{N_{\text{social}}}$ for some finite sizes N_{ego} and N_{social}. Thus the compositional class set is defined as the Cartesian product $\mathcal{C} = \mathcal{C}_{\text{ego}} \times \mathcal{C}_{\text{social}}$, following the analogous formulation for image-labeling in [73]. This set is then divided into two subsets, $\mathcal{C}^{\text{seen}}$ for the known compositions, which make up the training and validation set, and $\mathcal{C}^{\text{unseen}}$ for the unknown compositions that make up the test set. Importantly, there is no overlap between $\mathcal{C}^{\text{seen}}$ and $\mathcal{C}^{\text{unseen}}$, meaning that agent examples in the test set are out-

of-distribution with respect to the training set; that is, agents with contexts in $\mathcal{C}^{\text{seen}}$ are assigned to $\mathcal{S}_{ID}$, while those with contexts in $\mathcal{C}^{\text{unseen}}$ are assigned to $\mathcal{S}_{OOD}$.

In the *closed-world* test setting, $\mathcal{C}^{\text{unseen}}$ consists solely of novel *combinations* of known ego and social contexts; that is, for each paired $(c_e, c_s) \in \mathcal{C}^{\text{unseen}}$, both c_e and c_s are present in $\mathcal{C}^{\text{seen}}$, but never jointly on the same example. In the harder *open-world* test setting, however, we require that for each joint $(c_e, c_s) \in \mathcal{C}^{\text{unseen}}$, at least one of c_e or c_s is completely absent from $\mathcal{C}^{\text{seen}}$. Thus, the task of compositional zero-shot trajectory prediction requires developing a model that performs well on both $\mathcal{C}^{\text{seen}}$ and $\mathcal{C}^{\text{unseen}}$, despite training only on $\mathcal{C}^{\text{seen}}$. Unlike conventional CZSL, which requires *predicting labels* for unseen compositions, our setting requires *generating future behavior* for agents in unseen compositional contexts $(c_e, c_s) \in \mathcal{C}^{\text{unseen}}$.

4.3 Approach

To both construct long-tail, safety-relevant compositional settings for trajectory prediction and enhance generalization performance therein, we first extract relevant ego and social features (Section 4.3.1). We then discretize these features into context labels (Section 4.3.2) and use them to create challenging train/test splits (Section 4.3.3). Finally, we extend baseline trajectory prediction models with new generalization modules to improve out-of-distribution (OOD) performance (Section 4.3.4).

4.3.1 Safety-Relevant Feature Extraction

We begin by extending prior work on scenario characterization, namely, `SafeShift` from Chapter 3 and IMGTP [201], deriving broader sets of both safety-relevant and general attributes and systematically processing them to support downstream discretization. Given a focal agent i in scenario s, we first linearly interpolate X_i^{hist} and each corresponding $X_{j\neq i}^{\text{hist}}$ over agent i's valid timestep bounds. We utilize only information from the *history* portion of s to avoid any leakage from ground truth future trajectories. We then compute and process the following features:

Ego features: These focus on kinematic details, lane information, and traffic control device proximity. We first extract agent i's position, velocity, acceleration, and curvature, relative to its final pose (i.e., its position and heading at T_{hist}). Then, if a valid lane assignment exists based on heading similarity and lateral distance thresholds, we include speed-limit compliance, lane type, and lane-relative, Frenet coordinates of agent i's trajectory. Finally, we obtain distances and relative headings to stop signs, crosswalks, traffic lights, and speed bumps, if such infrastructure is present.

Social features: These instead focus on agent i's interactions with other relevant agents. We start by computing global scalar attributes, like scenario density. Then, for each non-stationary external agent within a given distance threshold to the focal agent, we compute a *geometry* type of the interaction at T_{hist}. We first assign interactions into collinear, parallel, opposite, or crossing types, using longitudinal and lateral distances, as well as relative heading differences. We further break down collinear geometries to leading, trailing, or head-on variants, and all other geometries to left or right variants relative to agent i.

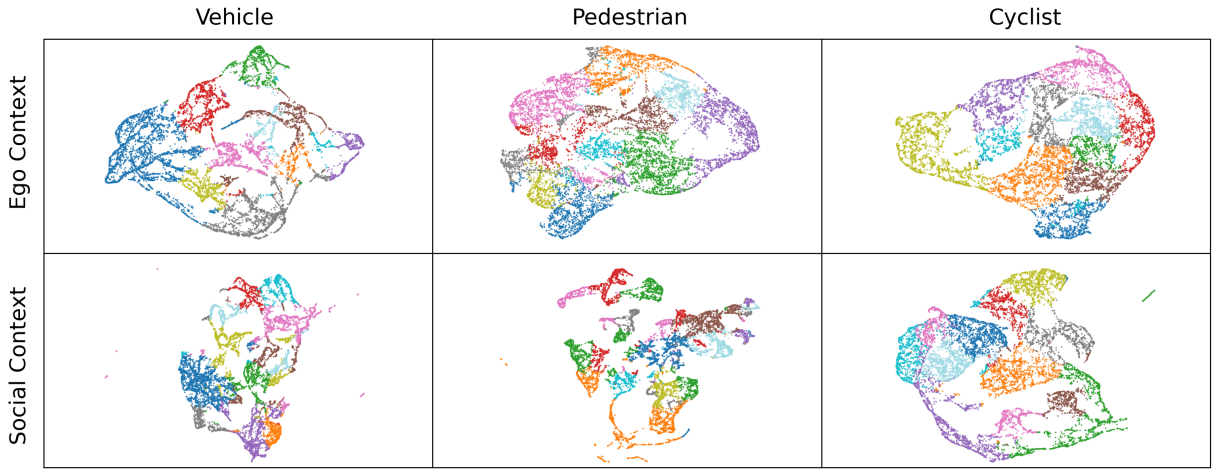


Figure 4.2: UMAP [112] visualizations for ego and social contexts across agent types. Colors correspond to discretized context labels from clustering described in Section 4.3.2, independently per diagram.

Next, we compute time-varying kinematic differences to the focal agent, both relative to agent i's final pose as well as to each of i's poses in X_i^{hist}. We explicitly capture collision-relevance of the interaction via closing speed and linearly projected conflict points at a fixed future horizon (i.e., projected locations of closest separation). For such conflict points, we compute the distance and pose of the point itself, as well as the difference in time-to-conflict-point ($\Delta TTCP$, established in [206]). Finally, we include a categorical feature capturing the object type (i.e., vehicle, pedestrian, or cyclist) of the external agent.

4.3.2 Feature Processing and Context Discretization

A key challenge in discretizing the above ego and social features into (c_e, c_s) context labels is that scenarios contain variable numbers and types of agents, traffic control devices, and other scenario elements, even including cases where none are present. We therefore process the extracted features into consistent, fixed-length representations, better supporting autoencoding and clustering.

We first organize input features into co-occurring groups. For ego features, these groups include the focal agent's kinematics, lane information, the closest instance of each traffic control device type, and the closest instance of each type located ahead of agent i. For social features, these groups include global features and the closest external agent of each geometry type. Within each group, we one-hot encode categorical values, summarize time-varying numerical values with scalar statistics (i.e., mean, standard deviation, minimum and maximum, and average slope), and directly include static numerical values. We then concatenate the features from all groups into a consistently ordered vector for each axis; if any group is absent (e.g., there is no forward stop-sign relative to agent i, or no agent trailing agent i collinearly, etc.), we zero-fill the required features. Finally, we append a valid bit to each group indicating whether or not it was present, producing the final ego and social vectors v_e and v_s.

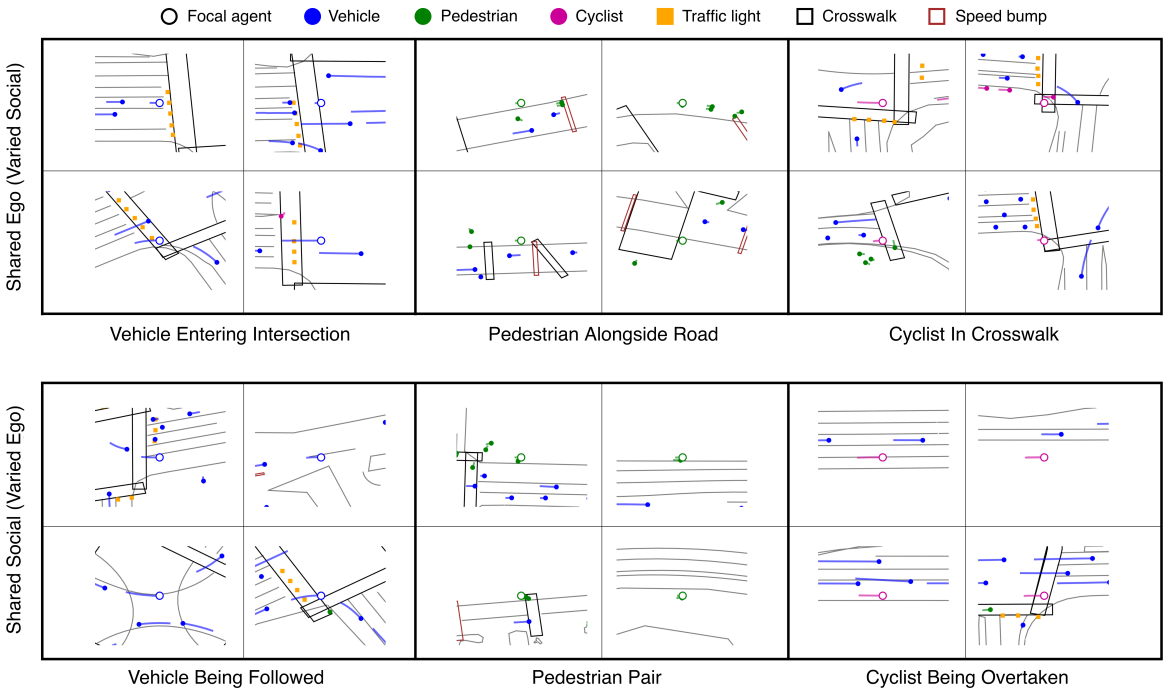


Figure 4.3: Cluster examples, by context and agent behavior types. In each subgrid of 4 examples, one context (e.g., ego) is fixed while the paired context (e.g., social) varies, with a brief caption describing the shared behavior. Agent markers show positions at T_{hist}; the focal agent is shown as a large, open circle, while background agents are smaller and filled.

Next, we train autoencoders for ego and social vectors separately. We further split by object type and train independent models for each type, allowing distinct latent spaces to be learned for e.g., pedestrian focal agents versus vehicle focal agents. Each autoencoder consists of a simple encoder and decoder multi-layer perceptron (MLP), with layer normalization and dropout on hidden layers; the encoder maps down to a low-dimensional latent space and the decoder maps back to the original feature space. That is, we compute $z = \text{Enc}(v)$ and $\tilde{v} = \text{Dec}(z)$. We train the models primarily with a mean-square error (MSE) reconstruction loss between v and $\tilde{v}$, along with a deep embedding clustering (DEC) [190] loss for regularization on the latent z values.

We then obtain discrete ego and social contexts by performing clustering within the latent spaces captured by these autoencoders, using k-means with $k = 11$. We use the Waymo Open Motion Dataset (WOMD) [37] as a representative source of AD scenarios, sampling approximately 20% of the total data. To quantitatively assess cluster and latent space coherence, we compute silhouette scores on held-out sets [147], observing values ranging from 0.31 to 0.50, which indicates a reasonably well-structured space. We also visualize UMAP [112] projections of the resulting spaces in Figure 4.2, showing clear separation and evidence of potential subclusters. We further present cluster examples for each latent space in Figure 4.3, demonstrating intra-cluster consistency alongside paired-context diversity. Ultimately, these intermediate results confirm the validity of our ego and social context derivations, supporting their use as c_e and c_s in downstream compositional tasks.

4.3.3 Closed-World and Open-World Settings

To ensure that held out context combinations indeed contain sufficient long-tail behaviors, we follow UniTraj [40] in considering Kalman difficulty as a reasonably well-calibrated approximation for the overall trajectory prediction challenge. That is, we compute the final displacement error a linear Kalman filter incurs when forecasting the focal agent i, averaged at fixed time horizons (i.e., 2, 4, and 6 seconds in the future). We then average this difficulty value for each clustered context obtained in Section 4.3.2.

Given these context difficulty values, we construct our *open-world* setting by greedily holding out all agents in the highest average-difficulty ego context (regardless of paired social context), as well as those in the highest average-difficulty social context (regardless of paired ego context), repeating this process until a desired test-set size is achieved. For our *closed-world* setting, we follow the same process, but for a given ego or social context that was held out, we add back in examples from *half* of its co-occurring paired contexts. That is, if e.g., all of c_e was originally held out, we add back $(c_e, c_s^{\{1,2,\ldots,N_{\text{social}}/2\}})$.

In both settings, the remaining examples are then split uniformly into training and validation sets of desired sizes. Again using WOMD [37] as a representative dataset, we observe that the average Kalman difficulties in the created test sets are both approximately 20% higher than their corresponding validation set difficulties. This similar magnitude of increase thus allows us to examine the impact of the *compositional* differences between the two long-tail settings, when used in downstream motion prediction experiments.

4.3.4 Generalization Strategies

We propose two techniques for improving performance in the above challenging settings: extending task-modular networks (TMNs) [128] for behavior generation, and refining intermediate representations through a difficulty-prediction auxiliary objective. Given a prototypical encoder-decoder trajectory prediction model, we operate directly in its bottleneck layer for both of these ideas. That is, given an intermediate representation of a focal agent as h, we transform it into an enriched h' before passing it to the model's decoder.

As in the original TMN, the intuition is that a gating network produces weightings to leverage learned modules in novel ways, as appropriate for novel context combinations. However, whereas the original TMN approach ultimately yields a compatibility score between various candidate context labels and the input representation, we refine the original h representation conditioned on fine-grained, sample-level latents.

Our TMN-inspired architecture thus consists of two main components: a sequence of N layers of M multilayer perceptron (MLP) "modules", and a latent-conditioned gating MLP $\mathcal{G}$. The first module layer processes M copies of the initial h vector, while modules in subsequent layers process weighted sums of the previous layer's outputs; these weights are obtained from the gating network's output. A linear projection head then maps the final module layer back to the original input space, producing h'.

To help ensure that such weightings from $\mathcal{G}$ are coherent, even in open-world settings, we condition $\mathcal{G}$ on the structured latent spaces obtained via the autoencoding process described in Section 4.3.2. Note that to avoid information leakage from the test split, we retrain these

autoencoders on the closed-world and open-world train sets described in Section 4.3.3, producing z'_e and z'_s vectors that can safely be used as input to $\mathcal{G}$. We train this component jointly with the baseline prediction model's objectives.

We additionally propose to further refine this h' by enhancing it with difficulty-awareness. We add an auxiliary head, formulated as a simple linear layer on top of h', to map to three scalar values, representing the anticipated Kalman error at 2, 4, and 6 seconds (trained with an MSE regression loss). In this way, we help promote implicit reasoning over difficulty-conditioned decision making (e.g., cautious versus aggressive behavior, etc.), which is especially helpful in our long-tail, safety-informed settings.

4.4 Experiments

Dataset and training details: As in our intermediate results in Section 4.3, we use the Waymo Open Motion Dataset (WOMD) [37] as our high quality source of scenarios, processed into a standardized ScenarioNet [89] format. We sample approximately 500k agent trajectories from these scenarios to perform ego and social context derivation, evaluation setting creation, and trajectory prediction experiments therein. As in the standard WOMD setting, we set T_{hist} to 11 and T_{fut} to 80 timesteps, at 10 Hz.

In the characterization process, we use a heading alignment threshold of 30 degrees and a lateral threshold of 6.5m when considering lane assignments. We additionally set an overall distance threshold of 50m for filtering out irrelevant agents, and use relative heading thresholds of 30 degrees and a collinear lateral threshold of 3.25m when determining geometry types. We also use a 10 second horizon when considering linearly projected conflict points.

When training autoencoders for context discretization, we use a latent dimension of $z = 16$, compressing and reconstructing v_e and v_s with input dimensions of 346 and 1443, resulting in models with approximately 500k and 800k parameters, respectively. We then use $k = 11$ clusters both for the DEC loss module and the k-means computation.

For conducting trajectory prediction experiments, we use the well-established Motion Transformer (MTR) [150] model, scaled to approximately 2.5M total parameters. In our generalization strategies, we utilize $N = 3$ intermediate layers and $M = 12$ modules per layer, where the gating network conditions on both z'_e and z'_s, with dimensions of 16 each. Overall, we add fewer than 100k new parameters to the base MTR model, across both ideas in Section 4.3.4. For finer-grained analyses, we train MTR augmented with the TMN-inspired component and auxiliary head jointly ("MTR + Both"), as well as with each component added separately as ablations ("MTR + TMN-Inspired" and "MTR + Auxiliary"). We train all models for 30 epochs, with an initial learning rate of $1e-3$, a fixed decay schedule, and a batch size of 192.

Metrics and evaluation: We use the standard ADE and FDE metrics, with the number of modes set to six, as in WOMD [37] and Argoverse [187]. We additionally report Brier-FDE values, which penalizes FDE by $(1-p)^2$, where p is the confidence in the best mode, as popularized in the Argoverse challenge.

Although the closed-world and open-world test sets in Section 4.3.3 contain more examples of difficult, long-tail behavior than their corresponding validation sets, the distribution still remains imbalanced with an over-representation of "easy" cases. To better isolate the impacts of

Table 4.1: Generalization results for the compositional settings. `Seen` results are in-distribution while `Unseen` results are out-of-distribution to the train set. Numbers in parentheses indicate relative change from the MTR baseline value in `Seen` (i.e., the setting's generalization gap on that metric). **Lower** numbers are better for all metric values and relative changes.

Setting	Method	Seen ADE	Seen FDE	Seen Brier-FDE	Unseen ADE	Unseen FDE	Unseen Brier-FDE
Closed-World	MTR	2.11 (−)	4.63 (−)	5.12 (−)	2.11 (+0.2%)	4.99 (+7.8%)	5.47 (+6.9%)
	MTR + TMN-Inspired	2.11 (−0.1%)	4.63 (−0.1%)	5.10 (−0.4%)	2.10 (−0.6%)	4.94 (+6.8%)	5.42 (+5.9%)
	MTR + Auxiliary	2.06 (−2.6%)	**4.37** (**−5.6%**)	**4.85** (**−5.2%**)	**2.08** (**−1.5%**)	4.85 (+4.7%)	5.33 (+4.1%)
	MTR + Both	**2.05** (**−2.8%**)	4.41 (−4.7%)	4.89 (−4.5%)	2.11 (+0.1%)	**4.84** (**+4.5%**)	**5.32** (**+3.9%**)
Open-World	MTR	1.97 (−)	4.33 (−)	4.82 (−)	2.12 (+7.8%)	5.15 (+19.0%)	5.65 (+17.4%)
	MTR + TMN-Inspired	2.00 (+1.7%)	4.41 (+1.8%)	4.89 (+1.6%)	2.12 (+7.5%)	5.13 (+18.5%)	5.62 (+16.8%)
	MTR + Auxiliary	2.07 (+5.1%)	4.52 (+4.5%)	5.01 (+4.1%)	2.14 (+8.6%)	5.17 (+19.5%)	5.67 (+17.7%)
	MTR + Both	**1.94** (**−1.5%**)	**4.28** (**−1.0%**)	**4.77** (**−1.0%**)	**2.06** (**+4.8%**)	**5.01** (**+15.6%**)	**5.49** (**+14.1%**)

the zero-shot compositional challenges, as well as the efficacy of our generalization strategies, we thus report all metrics averaged over the Kalman difficulty-class stratification established in UniTraj [40]. Finally, for all experiments, we select the best performing validation Brier-FDE checkpoint and use it to conduct inference on the held-out test sets, obtaining performance measures on both in-distribution compositions in $\mathcal{C}^{\text{seen}}$ and out-of-distribution compositions in $\mathcal{C}^{\text{unseen}}$.

4.5 Results

We present our main experiment results in Table 4.1. To understand the severity of our long-tail compositional zero-shot settings, we first report baseline metric values with MTR alone. As in `SafeShift` , we report each metric along with its relative change from this MTR baseline `Seen` value; error values increase by an average of $\mathbf{5.0}\%$ and $\mathbf{14.7}\%$ in closed-world and open-world settings, respectively. Note that in both settings, the drop in ADE performance is less than FDE, since shorter T_{fut} horizons tend to be easier to predict.

Overall, these results confirm that both our closed-world and open-world zero-shot settings indeed induce significant drops in OOD performance, even for a state-of-the-art model like MTR. Importantly, despite the fact that the relative Kalman difficulty between `Seen` and `Unseen` sets is similar in both settings, the open-world drop in performance was *far* larger than the closed-world drop, clearly demonstrating the impact of compositional setting design, as discussed in Section 4.3.3.

We then assess our generalization strategies proposed in Section 4.3.4. While the TMN-inspired and auxiliary loss components alone improve certain metrics in our settings, using both components achieves the strongest and most consistent performance, highlighting the efficacy of their joint refinement of the hidden representation space. In particular, our full approach reduces the `Unseen` performance gap to an average of $\mathbf{2.8}\%$ and $\mathbf{11.5}\%$ in closed-world and open-world, respectively. Furthermore, performance in the `Seen` setting improved by an average of 4.0% in closed-world and 1.2% in open-world. Hence, our proposed strategy of combining modular reasoning using ego and social representations, along with implicitly considering the difficulty of a given example, improves performance broadly, but especially in the OOD `Unseen` sets.

4.6 Discussion

Robust development and evaluation of trajectory prediction models remain an essential challenge in autonomous driving, especially in the presence of long-tail, safety-critical scenarios encountered in deployment, which are rare or missing altogether in large-scale datasets. We therefore proposed novel evaluation settings on existing datasets, where test sets are constructed to be OOD with respect to training data on the bases of safety-informed context compositions—to our knowledge, the **first** extension of the well-established image-labeling CZSL setting to motion prediction. To create these settings, we developed a factorization framework that disentangles traffic participants along ego and social axes, clustering them into paired context labels. We then used these contexts to build long-tail, zero-shot closed-world and open-world settings. On a state-of-the-art baseline model, we observed performance drops from ID data of 5.0% and 14.7%, respectively; by extending task-modular gating networks and incorporating an auxiliary, difficulty-prediction loss, we reduced these OOD gaps to 2.8% and 11.5%, while also improving ID performance by 4.0% and 1.2%, respectively.

Despite the efficacy of our evaluation settings and generalization strategies, further improvements are still possible. In particular, extending other non CLIP-based approaches from image-labeling CZSL, such as learned attention propagation [76], could boost both ID and OOD performance. Additionally, factorizing along more than two safety-relevant, semantic axes could also increase interpretability, as well as the effectiveness of generalization approaches. We encourage future work to explore these directions.

Chapter 5

First-Person View Error Robustness via Re-Simulated Perspectives

To move beyond repartitioning datasets, as in Chapters 3 and 4, we now introduce the second key pillar of enhanced data utilization for robust autonomy: modifying existing data in a realistic and purposeful way, as described in Section 2.3.2. Specifically, in this chapter, we investigate robustness to perception errors in social navigation motion prediction by re-simulating and re-annotating datasets under imperfect perception conditions.

To develop socially-aware robots that interact in a crowded environment, predicting pedestrian motion is essential [1, 11, 53, 195, 204]. State-of-the-art (SOTA) social navigation trajectory prediction approaches typically utilize datasets that provide full trajectory information of all pedestrians in a bird's-eye view (BEV) scene, unlike their counterparts in autonomous driving [124, 137]. However, BEV observation is unrealistic for agents navigating in the real world, as they must rely on egocentric, first-person view (FPV) sensing. A realistic setting also includes limited field-of-view (FOV), occlusions, and changes in perspective and orientation of the ego-agent.

While collecting top-down data using an overhead camera is relatively convenient, creating a first-person view counterpart is far more challenging for several reasons. To begin with, all participants in the scene would need to wear a camera sensor to record their egocentric views, as well as a location-recording sensor to establish their ground truth locations. Furthermore, such a setting is subject to psychological issues such as the observer (or Hawthorne) effect [69], where people's behaviors in these experiments may not be entirely representative of natural social interaction.

Therefore, to explore first-person view trajectory prediction in a realistic manner, we propose T2FPV, a method for constructing an FPV version of data from a trajectory-only dataset by simulating the agents in high fidelity. The FPV data is collected by having each agent follow their recorded trajectory with a simulated camera attached to them. We perform extensive annotation and post-processing to provide unique information beyond existing FPV datasets as follows: 1) we conduct SOTA detection-and-tracking, giving realistic partial perception of trajectories and enforcing data imputation as a core aspect of the task (in contrast with prior works which simply re-provide ground truth BEV trajectories [11, 32]); 2) our approach utilizes SEANavBench [166], a high-fidelity simulation environment, to provide realistic synthetic images; and 3) we addition-

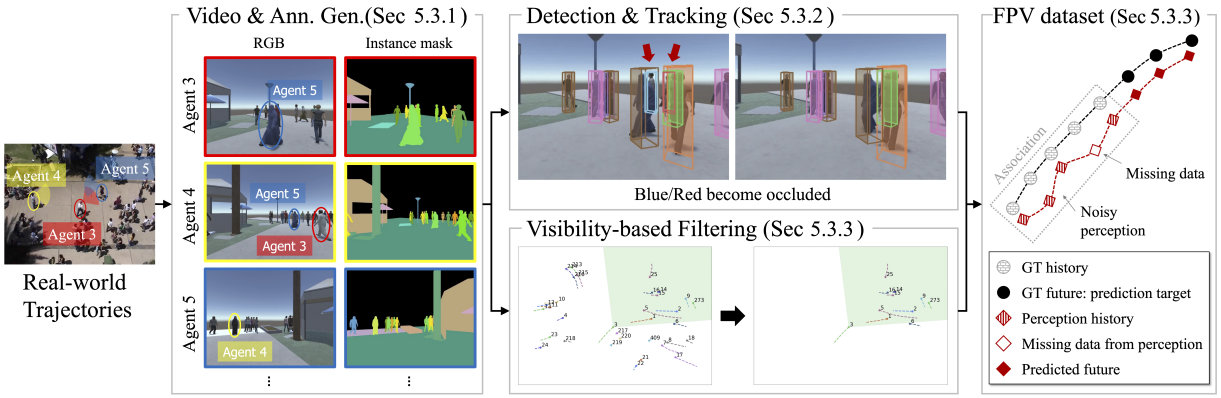


Figure 5.1: **T2FPV Overview**: We generate filtered ground truth tracks and the corresponding D&T tracks from a real-world pedestrian dataset. The downstream task is to predict the future path (black circles) for a set of perceived track histories (striped red diamonds).

ally provide the corresponding ground truth of all observed and missed points of each trajectory, for its history and future (compared to only having information perceived from the camera as in [84, 132, 195]). An overview of our approach is shown visually in Figure 5.1.

To showcase our approach, we construct the T2FPV-ETH dataset based on the ETH/UCY trajectory dataset [124]. In this realistic FPV setting, we observe that a new class of errors is present compared to in BEV. These "FPV errors" arise from occlusion and field-of-view (FOV) limitations of robot sensing, combined with imperfect detection and tracking, resulting in missing observations. When performing trajectory prediction with various SOTA approaches, these errors caused our observed metrics to be significantly worse than what was reported in the BEV setting in the literature[1].

Prior work in pedestrian prediction has largely ignored FPV errors, either throwing out incomplete tracks or relying on simple interpolation over the missing points [202] [184]. Recent work has made significant advancement in data imputation; however, the vast majority of these works focus on artificially missing data [100, 129, 215]. Additionally, these works only focus on imputation as an independent task without considering how it affects prediction performance. Hence, to reduce the FPV errors for improved prediction, we propose a **Co**rrection of **F**PV **E**rrors (CoFE) module that can refine initial imputations via end-to-end training alongside a trajectory prediction approach. We find that our approach decreases prediction displacement errors by more than 10% on average when compared to all tested imputation and forecasting combinations.

Our main contributions are: 1) we propose a method for creating an egocentric view for each agent given a set of trajectories; 2) we generate the T2FPV-ETH dataset, a new first-person view dataset that corresponds to the ETH/UCY dataset; 3) we propose and evaluate CoFE, an end-to-end learned input correction module, which reduces the impact of FPV errors beyond SOTA imputation approaches; and 4) we release our dataset and software tools to promote research in first-person view trajectory prediction.

This chapter is based on work done with my collaborators [153]. Our code and tools are

[1]For instance, Average Displacement Error (ADE) / Final Displacement Error (FDE) performance increased from 0.44m / 0.89m in BEV to 1.51m / 2.08m in FPV for VRNN [9]

freely available at https://github.com/cmubig/T2FPV.

5.1 Related Work

Real-World First-Person Datasets. Various large-scale datasets provide video footage from an ego agent's perspective. [48] is a large-scale first-person view video dataset, with over 3500 hours of footage collected from various sources around the world. Egocentric Basketball Motion Planning [8] provides a wearable camera perspective from multiple people in the scene. However, neither of these datasets are focused on social navigation. They feature many instances of the ego agent walking by themselves or performing an unrelated task (such as carpentry, basketball, etc.) that have inherently different social contexts than navigating in public.

[84] is a dataset of an egocentric pedestrian video stream, providing pose, acceleration, and orientation information. Similarly, [195] uses human camera wearers and pose estimation to create a dataset of 2D targets to predict. However, these approaches only provide a single perspective of an ego agent in each scenario, limiting the diversity of ego behaviors. Also, both lack the ground truth pose information of other agents in the scene, especially due to occlusion and FOV limits.

Self-driving datasets, such as [15, 158], suffer the same problem of not having full ground truth information for training and evaluation. Furthermore, a car's ego-motion is incomparable to a pedestrian's ego-motion in terms of physical characteristics. Additionally, the social interactions and roles between the ego and detected agents are vastly different between the two fields.

Synthetic Pedestrian Datasets and Simulation. Several recent works have generated synthetic data in simulations based on a corresponding real-world dataset. FvTraj [11] uses Unity to render FPV images from ground truth trajectory data [74], but these rendered images consist only of a flat ground plane with no corresponding environment modeled. DeepSocNav [32] generates ego view depth images from ETH/UCY, with a low-fidelity environment model. However, they do not include images from RGB cameras, which are far more common than depth sensors. Furthermore, DeepSocNav [32] and FvTraj [11] do not release any generated images or their in-house simulators, limiting reproducibility and engagement within the research community. Most importantly, both works only use the generated images for augmenting ground truth trajectories when performing prediction; no perception or detection and tracking is used, keeping the task less realistic.

[93] and [166] are relatively high-fidelity simulation environments with scene constructions of ETH/UCY, but both lack first-person views. Furthermore, these approaches also have the same aforementioned limitations as [11, 32] regarding perception and task settings.

Pedestrian Trajectory Prediction. Recent work on trajectory prediction and forecasting has mostly focused on top-down trajectory datasets such as ETH/UCY [124], SDD [137], and inD [13]. [1] uses LSTMs to jointly predict trajectories of all agents, incorporating pooled hidden-state information from neighbors as a social cue. Some approaches, such as AC-VRNN [9], use generative models within a VRNN [29], incorporating social interactions via attentive hidden state refinement. Several works also leverage top-down images explicitly, whether in an RGB form or with added semantic segmentation [110, 188, 204]. SGNet [87] generates coarse step-

wise goals to assist trajectory prediction sequentially. [143] incorporates agent dynamics and environment information and forecasts using a graph-structured recurrent model.

Fewer works have focused on the FPV setting for pedestrians. [195] utilizes FPV to model and predict the trajectory of a single agent directly in pixel-space. [122] creates a spatial visual distribution of objects from FPV, and applies perception and ego-agent trajectory planning in a 2.5D coordinate system. [132] uses a transformer-based architecture, with a graph scene encoding to forecast the camera wearer's trajectory with nearby agents as a cue. Still, none of these works deal with FPV errors when providing the trajectories of detections.

Trajectory Perception Robustness. The field of sequence imputation has had much success with deep learning recently. NAOMI [100] uses a non-autoregressive approach at multiple step sizes to impute missing data in the context of basketball players and billiard ball trajectories. [129] trains imputation and prediction together but still only evaluates them separately rather than as an end-to-end pipeline. [130] evaluates the end-to-end task but only focuses on low-resolution, long-term GPS data, dissimilar to the fine-grained social navigation task. Also, these approaches only deal with artificial missing-completely-at-random (MCAR) data rather than dealing with pathologically missing data due to FPV errors. [215] focuses on real vehicle and human motion trajectories, by transforming existing forecasting challenges into imputation ones. However, similar to the above approaches, they still only apply MCAR masks to ground truth. Furthermore, they only deal with imputation between ground truth points, rather than points which themselves may be erroneous from the perception model.

There have been several recent works in improving the robustness of trajectory forecasting to perception errors. [202] combines refinement via exponential smoothing with trajectory prediction to iteratively re-match observed trajectories with ground truth. [184] reframes the perception pipeline to remove tracking altogether, instead operating directly on detections and affinity matrices. However, both approaches still rely only on simple linear interpolation and extrapolation for missing data. While these are interesting, complementary approaches, we leave them as future work as they primarily focus on tracking and data association errors.

5.2 Preliminaries

In this chapter, we extend the standard motion prediction task described in Section 2.2.1: each agent must predict the trajectories of all agents in their view using only *their own* egocentric information. Hence, observed non-ego tracks may be erroneous compared to the ground truth.

For each agent $i \in \mathbf{A}$, we separately utilize the scenario perturbation function $\mathcal{P}$, defined in Section 2.3.2, to create a new version of s from agent i's re-simulated FPV perspective, denoted s_i. Hence, $\mathcal{P}_i$ produces a set of behaviors where $\mathcal{B}_i$ is simply the original trajectory X_i and each $\mathcal{B}_{j \neq i}$ is the *estimated* trajectory $\widetilde{X}_j$, where positions are output from a detection and tracking module and missing points have been imputed by some method.

Thus, given these observations, the FPV trajectory prediction problem for a given s_i entails predicting the ground truth $\mathbf{X}^{\text{fut}}$, given only X_i^{hist} and $\left\{ \widetilde{X}_j^{\text{hist}} \right\}_{j \neq i}$. In this setting, we consider i to be the ego agent and each $j \neq i$ to be "detected" agents.

5.3 Trajectories to First-Person View

We describe our T2FPV method, demonstrating how we construct first-person view data from an example trajectory dataset, namely the ETH/UCY dataset. This dataset consists of five "folds" of recorded data, in different locations and times: ETH, Hotel, Univ, Zara1, and Zara2.

5.3.1 Video and Annotation Generation

Our approach for creating FPV datasets from real-world trajectory datasets begins with generating videos and ground truth annotations. We use the SEANavBench [166] simulation environment as a starting point for our simulation. SEANavBench consists of high-fidelity pre-modeled scenes for each location within ETH/UCY. We leave these scenes as unchanged as possible, for consistency with prior works using SEANavBench.

As in [11], we enforce a number of assumptions when rendering these tracks. For instance, we orient each pedestrian's gaze with the direction they are traveling in, with spherical linear interpolation for smoother angle changes. Additionally, we mount a camera on each pedestrian at a fixed height of $1.6m$ from their base and assign the following physical characteristics to the camera: $18mm$ focal length, $36 \times 24mm$ sensor, and zero lens shift for the principal point. When rendered at our 640×480 resolution, this results in a vertical FOV of approximately $67°$.

Using the above assumptions, we then render the first-person videos for every person following their track from the original dataset, as well as output an annotation for each agent at every frame. The videos consist of the RGB render, as well as an instance segmentation render, as shown in Figure 5.1, where each object in the scene has been given a unique color. The annotations consist of the agent's ID, pose information, and a list of what other agents can be seen in the camera's view, i.e., the poses of all visible agents in both the camera and world reference frame. This detection list is generated by utilizing the aforementioned segmentation mask to determine agent visibility.

5.3.2 Perception: Detection and Tracking

To perform trajectory prediction in a realistic setting, we employed an off-the-shelf object detector and tracker to produce the observations required. We used a 3D object detector [121] which is SOTA among recent image-only methods which do not require depth information [106], and a simple but effective probabilistic tracker [27]. We made the following changes to both approaches to produce reasonable detection and tracking results.

In DD3D [121], we set the parameters of feature map assignment to use thresholds that fit our ground truths appropriately. We also only used instances that are "visible" (as defined in Section 5.3.3), which helps to filter out heavily occluded instances. For the tracker [27], we changed the matching metric to use BEV IoU (Intersection-over-Union in top-down view) from Mahalanobis distance [113] to associate detections to tracks. We also applied the Kalman filter only to each instance's 3D location and orientation and used state and observation noise covariances calculated from our ground truth data.

Following the common evaluation procedure as in the ETH/UCY trajectory prediction task, we trained one model for each of the five folds, using the other four folds as the training and

Table 5.1: Detection and tracking performance.

Fold	Detection		Tracking	
	AP_{2D}	AP_{BEV}	AMOTA	AMOTP
ETH	96.50	44.10	0.384	1.262
Hotel	94.24	42.56	0.361	1.325
Univ	90.65	67.56	0.318	1.465
Zara1	97.29	90.22	0.709	0.610
Zara2	94.67	73.78	0.517	1.000

validation sets respectively. We then produced tracking results on all ego videos from each fold's test-set.

5.3.3 FPV Dataset Creation

In transitioning from *bird's-eye view* (BEV) to *first-person view* (FPV), given a scene with N agents, we now construct N variations of the original scene, i.e., from each agent's perspective. We begin with the same pre-processing popularized in Social GAN [53], only considering scenes with at least two concurrent agents in a sliding window consisting of $T_{\text{hist}} = 8$ and $T_{\text{fut}} = 12$ timesteps. Then, to account for FPV errors, we redesign the scene as follows, for each agent's perspective.

First, we consider the set of observed tracks from the detection and tracking module. We filter out tracks that are seen by the ego agent for fewer than κ of the first T_{hist} timesteps. Next, we perform an initial imputation on missing values using linear interpolation (as in [184]). This creates a D&T set of tracks, $\widetilde{\mathbf{X}}_i$ consisting of $\widetilde{X}_j$ for each detected agent j from the ego i's perspective, along with X_i itself.

We then consider the set of ground truth (GT) tracks from BEV. We filter out tracks that are impossible to have been seen by the ego agent for fewer than k of the first T_{hist} timesteps (i.e. by having fewer than P pixels visible from instance segmentation). Furthermore, we filter out tracks for which the ground truth is missing pieces of data for any of T_{hist} or T_{fut}. In our creation of T2FPV-ETH, we used $\kappa = 3$ and $P = 100$. This step then creates a GT set of tracks $\mathbf{X}_i \subseteq \mathbf{X}$, containing the ego agent, i, and each agent which is feasibly visible to it.

For each scene, the GT and detected sets of tracks are associated together by performing Hungarian matching, as in [184, 185, 202], based on the mean squared error (MSE) of all pairs between $\mathbf{X}_i^{\text{hist}}$ and $\widetilde{\mathbf{X}}_i^{\text{hist}}$; ego assignment is forced.

5.3.4 Dataset Statistics

We measure the detection and tracking performances of the SOTA methods we employed in Table 5.1. For detection performance, we measure the standard average precision (AP_{2D}) in 2D image space and observe that it performs well. Also, we measure the localization quality of detected objects in 3D space by calculating IoU-based average precision in the top-down view (AP_{BEV}). Both metrics use the same IoU threshold of 0.5. The AP_{BEV} performance is

Table 5.2: T2FPV-ETH statistics.

Fold	Num Ego	Num Dets	Det MSE	FPV Err. Rate
ETH	181	60	2.05	0.44
Hotel	1053	449	2.03	0.51
Univ	24,334	120,072	1.13	0.45
Zara1	5,939	3,686	0.64	0.28
Zara2	17,608	11,775	1.05	0.32

worse than AP_{2D}, which shows the challenge of image-based 3D detection. For tracking, we adopt two popular metrics from [183], Average Multi-Object Tracking Accuracy (AMOTA) and Precision (AMOTP). AMOTA combines false positives, missed targets, and identity switches, and AMOTP measures the misalignment between prediction and ground truth. Although "Univ" shows the worst performance because of the pedestrian density (Table 5.2), the detector and tracker perform reasonably well, as shown qualitatively in Figure 5.1.

Table 5.2 provides a high-level overview of the number of scenes and detections, as created in Section 5.3.3. We note that this table demonstrates a data augmentation effect, as there is now a one-to-one correspondence between each ego agent and a scenario; a single ground truth track is often observed by multiple other agents at once, although with different possible FPV errors and 3D detection locations. These statistics indicate the diversity between the different folds as testing sets, as they have significantly varying scene densities (i.e. detections per ego agents), as well as rate of FPV errors (i.e. number of points needing to be imputed downstream) and difficulty of imputation.

5.4 Proposed Method: CoFE

5.4.1 Motivation

As noted in Section 5.1, existing imputation approaches have two primary deficits when being applied to the field of human trajectory prediction. First, approaches largely use a missing-completely-at-random (MCAR) treatment of the points to be imputed. This assumption does not hold in a setting with FPV errors, as data is missing in a manner pathological to the detection and tracking approach being used as well as compulsory to occlusion and FOV limitations from the ego camera. Second, approaches have full trust in the accuracy of the points around the missing data. This assumption clearly also does not hold in the FPV setting, as the positions of observations are estimated from our approach described in Section 5.3.2.

Therefore, we propose to incorporate a correction module, **Co**rrection of **F**PV **E**rrors (CoFE), between existing imputation approaches and downstream trajectory prediction approach, as shown in Figure 5.2. To make the correction consistent with the trajectory prediction task, we thus train a neural network for both the imputation refinement and the trajectory prediction algorithm itself in an end-to-end (E2E) manner.

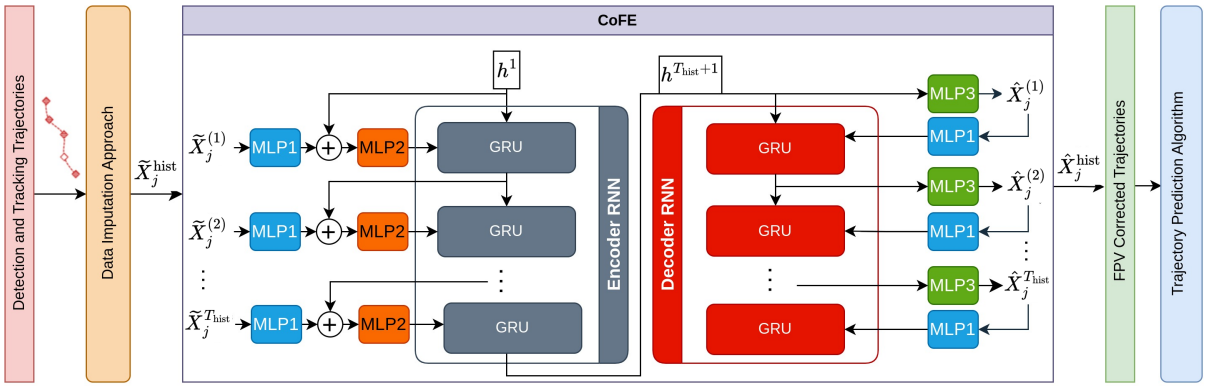


Figure 5.2: **CoFE Approach**: CoFE has an encoder-decoder architecture that refines the imputed trajectories to better account for FPV errors. The corrected trajectories are then passed into a trajectory prediction algorithm.

5.4.2 CoFE Architecture

As shown in Figure 5.2, our architecture is similar to many RNN-based trajectory forecasting approaches, such as VRNN [29]. Our main insight is to use an encoder RNN and decoder RNN back to back. The accumulated hidden state after processing the input then feeds into the decoder to output a "corrected" version of the input. We utilize two different Gated Recurrent Unit (GRU) modules for this purpose.

For a detected agent $j \neq i$, each missing point of $\widetilde{X}_j^{\text{hist}}$ is imputed via the given approach, e.g., NAOMI [100]. Then, each point is transformed into its relative motion from the previous point, in order to be agnostic to absolute coordinates in the given scene. Next, these relative motions are feature extracted with a multilayer perceptron (MLP), labeled `MLP1` in our diagram. These features are concatenated with the hidden state h^t at each timestep in T_{hist}, then fed into `MLP2` and the encoding GRU cell to obtain the next hidden state, h^{t+1}. After processing the entire input, we then switch to decoding with the last hidden state, $h^{T_{\text{hist}}+1}$. We output $\hat{X}_j^{(1)}$ as a prediction for $X_j^{(1)}$, via `MLP3`. We then apply the same feature extractor (i.e., same weights) `MLP1` to this prediction to then feed back into the decoding GRU cell, and repeat this process for all T_{hist} points. Finally, we convert the predicted points back into absolute coordinates. The exact details of this architecture, including the number of layers and hidden units in each MLP, can be seen in our open-sourced implementation.

5.4.3 End-to-End (E2E) Training

We introduce a simple MSE Loss objective to train CoFE itself, between the ground truth X_j^{hist} and corrected estimations $\hat{X}_j^{\text{hist}}$. Then, given the original estimated points with imputation $\widetilde{X}_j^{\text{hist}}$ and also refinements $\hat{X}_j^{\text{hist}}$, we update the points in $\hat{X}_j^{(t)}$ with $\widetilde{X}_j^{(t)}$ for timesteps t where imputation is not required, resulting in $\hat{X}_j'$. This final $\hat{X}_j'$ is then used to train the downstream prediction method (e.g., SGNet [87]) in an end-to-end (E2E) manner, where the loss function being optimized is the sum of the CoFE loss objective and the prediction method's original objective.

Table 5.3: ADE / FDE for each fold and approach tested on T2FPV-ETH dataset. The better result between using CoFE and not is **bolded** and the overall best performance for each prediction method is green. Lower is better.

Unit: meter

Traj. Prediction	Imputation	CoFE (ours)	ETH	Hotel	Univ	Zara1	Zara2	Avg
VRNN [29]	Linear-interp	-	**1.35 / 2.00**	1.30 / 1.73	2.24 / 2.89	1.14 / 1.68	1.54 / 2.10	1.51 / 2.08
	Linear-interp	✓	1.52 / 2.35	**1.06 / 1.53**	**1.65 / 2.10**	**1.06 / 1.63**	**1.27 / 1.62**	**1.31 / 1.84**
	Smooth [202]	-	2.25 / 3.75	1.53 / 2.36	2.17 / 3.00	1.26 / 1.95	1.58 / 2.22	1.76 / 2.65
	Smooth [202]	✓	**1.57 / 2.46**	**1.08 / 1.55**	**1.77 / 2.31**	**1.00 / 1.44**	**1.31 / 1.68**	**1.35 / 1.89**
	NAOMI [100]	-	**1.46 / 2.29**	1.59 / 2.17	1.83 / 2.31	0.96 / 1.57	1.13 / 1.49	1.39 / 1.97
	NAOMI [100]	✓	1.54 / 2.34	**1.09 / 1.55**	**1.58 / 1.97**	**0.92 / 1.39**	**1.11 / 1.39**	**1.25 / 1.73**
A-VRNN [9]	Linear-interp	-	**1.39 / 2.04**	1.31 / 1.75	2.26 / 3.00	1.04 / 1.40	1.47 / 1.93	1.49 / 2.03
	Linear-interp	✓	1.47 / 2.18	**1.16 / 1.72**	**1.54 / 1.88**	**1.03 / 1.48**	**1.31 / 1.69**	**1.30 / 1.79**
	Smooth [202]	-	1.77 / 3.11	1.36 / 1.81	2.27 / 3.34	1.18 / 1.72	1.59 / 2.07	1.63 / 2.41
	Smooth [202]	✓	**1.69 / 2.71**	**1.10 / 1.59**	**1.76 / 2.38**	**1.06 / 1.59**	**1.28 / 1.62**	**1.38 / 1.98**
	NAOMI [100]	-	**1.44 / 2.17**	1.66 / 2.30	1.82 / 2.25	**0.83 / 1.24**	**1.09 / 1.39**	1.37 / 1.87
	NAOMI [100]	✓	1.49 / 2.18	**1.16 / 1.66**	**1.54 / 1.91**	0.88 / 1.31	1.16 / 1.49	**1.25 / 1.71**
SGNet [87]	Linear-interp	-	1.43 / 1.97	0.72 / 1.00	1.48 / 1.73	0.58 / 0.79	0.78 / 0.91	1.00 / 1.28
	Linear-interp	✓	**0.98 / 1.32**	**0.59 / 0.76**	**1.23 / 1.48**	**0.55 / 0.76**	**0.73 / 0.86**	**0.82 / 1.04**
	Smooth [202]	-	1.06 / 1.45	0.73 / 1.04	1.45 / 1.68	0.57 / 0.78	0.79 / 0.93	0.92 / 1.18
	Smooth [202]	✓	**1.03 / 1.41**	**0.61 / 0.80**	**1.28 / 1.54**	**0.56 / 0.77**	**0.74 / 0.86**	**0.84 / 1.08**
	NAOMI [100]	-	**0.90 / 1.28**	0.78 / 0.97	1.21 / 1.43	**0.50 / 0.69**	0.72 / 0.84	0.82 / 1.04
	NAOMI [100]	✓	0.99 / 1.40	**0.59 / 0.78**	**1.16 / 1.39**	0.51 / 0.70	**0.67 / 0.79**	**0.78 / 1.01**

5.5 Experiments

5.5.1 Experimental Setup

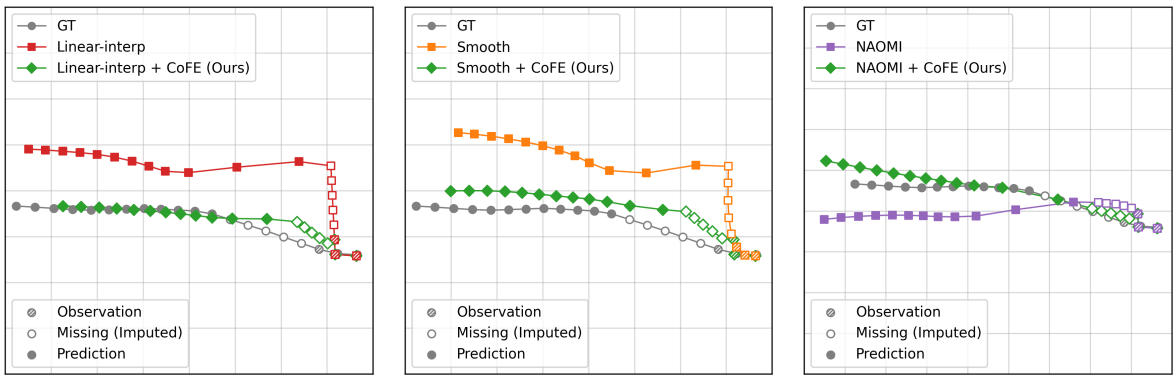
We implemented several representative approaches on the ETH/UCY trajectory prediction task. We selected these algorithms as they stood out along several key techniques common in human trajectory prediction: variational prediction (VRNN [29]), social awareness (A-VRNN [9]), and goal conditioning (SGNet [87]).

For data imputation, we incorporated three commonly used approaches. We selected linear interpolation ("Linear-interp"), a simple but powerful approach used as part of many recent works, such as [184]. We also selected double exponential smoothing ("Smooth"), used in [202], a more complex baseline that better handles dynamic trends in the sequence. Finally, we incorporated NAOMI [100], which is a recent SOTA approach leveraging deep learning.

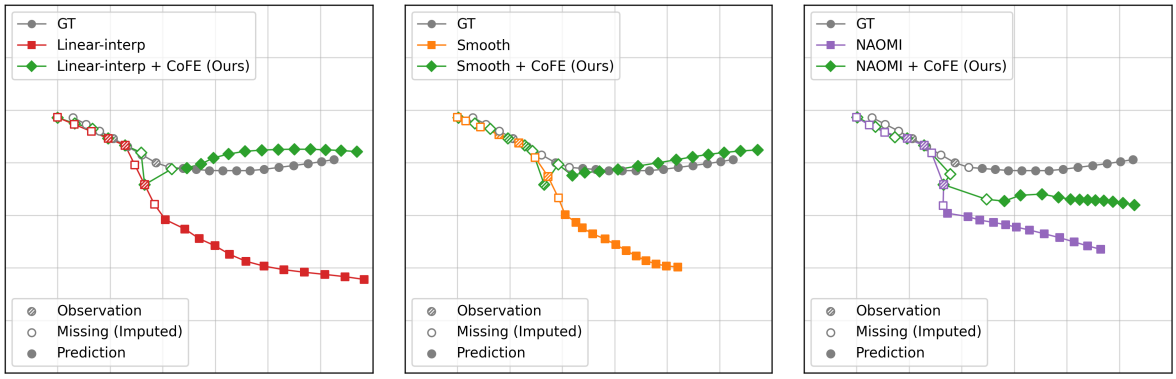
5.5.2 Evaluation Procedure

As in Social GAN [53], we evaluate trajectory predictions using a leave-one-out approach. For each of the five folds, models are trained and validated on data from four of them at a time. Then, the best model according to validation performance is tested on the entirety of the held-out fold.

We train NAOMI [100] separately, following the author's procedure, once per fold. Then, for each combination of imputation techniques and prediction algorithms, we train one predic-



(a) Scenario 1 (From Zara2)



(b) Scenario 2 (From Univ)

Figure 5.3: **Qualitative Results**: We demonstrate the effectiveness of CoFE when applied to different imputation methods, using SGNet [87] for prediction. Grid lines represent 1m.

tion model utilizing CoFE and one model without (i.e., just using the $\widetilde{X}_j^{\text{hist}}$ outputs from the imputation).

As discussed in previous chapters, we use ADE and FDE as our primary metrics. We note that there are other metrics which could be utilized, including collision rate, social comfort level, path complexity, and many more [111]. We chose to focus on the core metrics of the tasks at hand, but suggest that future work in applying these metrics could provide helpful new insight.

5.5.3 Results

We conducted extensive experimentation to assess CoFE's performance when combined with the various imputation and prediction approaches. As shown in Table 5.3, adding the CoFE module is quite effective. When considering the average performance over all dataset folds, all combinations of imputation and prediction algorithms which use CoFE are better than the corresponding versions which bypass it. Furthermore, without CoFE, the average performance is highly variable, dependent on the choice of imputation approach and prediction method. However, with CoFE, these differences become much less pronounced. Note that although performance on most folds in most cases is quite good, CoFE appears to not be as effective on ETH. We suspect

Table 5.4: Ablation study on CoFE applied to SGNet with linear interpolation.

Algorithm	CoFE	Train E2E	Impute Only	ADE / FDE
SGNet [87]	-	-	-	1.00 / 1.28
	✓	No	No	1.11 / 1.43
	✓	No	Yes	0.98 / 1.27
	✓	Yes	No	0.94 / 1.22
	✓	Yes	Yes	**0.82 / 1.04**

that this is because, as shown in Table 5.2, there are only a small number of detected tracks in the first place (60), so performance is more variable and sensitive to any individual prediction's error.

To gain further insight into CoFE's performance, we performed qualitative analyses. As seen in Figure 5.3-(b), the imputation approach (NAOMI [100]) trusts surrounding points in the data, performing an extrapolation and thus, does not effectively capture the FPV errors. When paired with CoFE, the approach is more effective at capturing underlying temporal and spatial patterns in the data, correcting the FPV errors which results in better downstream prediction.

We also performed an ablation study, shown in Table 5.4, to assess aspects of our design choices. We focused on SGNet [87] combined with linear interpolation for the study, and found clearly that the E2E training was vital in obtaining top performance. We further find that focusing on imputation only in the prediction phase (i.e., replacing non-imputed points to create $\hat{X}'_j$ as described in Section 5.4.3) also has a significant effect in improving performance.

5.6 Discussion

In existing work, pedestrian trajectory prediction has been mainly studied under a complete information assumption. In this chapter, we introduced a first-person view trajectory prediction problem where agents need to make predictions based on partial, imprecise information. To promote this research direction, we presented T2FPV, a method for generating high-fidelity egocentric datasets for pedestrian navigation by leveraging existing real-world trajectory datasets. In this setting, FPV-specific errors arose due to imperfect detection and tracking, occlusions, and FOV limitations of the camera. To address these errors, we proposed CoFE, a module that further refines imputation of missing data in an end-to-end manner with trajectory forecasting algorithms. Our method reduced the impact of such FPV errors on downstream prediction performance, decreasing displacement error by 11.73%, averaging over all combinations of imputation techniques and prediction approaches tested. We also showed that E2E training of CoFE is essential in achieving this performance increase. Our constructed T2FPV-ETH dataset provides a benchmark for human trajectory prediction from detection and tracking results, which is a more natural and realistic setting. Therefore, we argue that incorporating such realism throughout the perception pipeline is an important direction to move toward in enabling robots to navigate in the real world.

Although SEANavBench is a high-fidelity environment, we do note that further effort in

improving its sensory realism could be useful. Realism could be enhanced not just by increasing the 3D-modeling asset and animation qualities, but also by further improving alignment between the reproduced scenery and the original locations. Additionally, for associating D&T tracks with their corresponding GT tracks, we relied on Hungarian matching on our tracking output directly. This decreased the number of correctly matched trajectories, due to identity association errors of detections. Incorporating affinity-based techniques from [184] or performing the full re-tracking algorithm from [202] could be a promising way to even further reduce FPV errors.

Chapter 6

Skill-Enabled Safety-Critical Scenario Generation

In this chapter, we continue to develop the scenario modification pillar of enhanced data utilization, shifting focus from perception-level robustness in social navigation (Chapter 5), to behavior-level robustness in autonomous driving (AD). While Chapters 3 to 5 utilized open-loop motion prediction as a representative downstream task, we now transition to evaluating closed-loop agent performance in simulation, more directly assessing the decision-making layer of the autonomy stack. Although validation of system behavior under normal operating circumstances is valuable, testing AD behavior under *safety-critical* and other corner-case circumstances is vital for Safety of the Intended Functionality (SOTIF) standards [68, 151, 212]. Given the "curse of rarity" of safety-critical scenarios in AD datasets, as discussed in Chapters 1 and 3, programmatically *generating* safety-critical scenarios is thus necessary. To ensure that generated scenarios retain realistic properties, it is appealing to perturb the behavior of one or more agents in a principled way, rather than using first principles to painstakingly assemble a scenario from scratch [17, 34, 63, 210]. In this setting, one agent is referred to as the *ego* agent, while the modified background traffic participants are *adversary* agent(s), who attempt to attack the ego in some way.

State-of-the-art (SOTA) approaches in perturbation-based adversarial scenario generation have coupled a dynamic scenario generation framework with an ego control policy being trained with closed-loop objectives [163, 197, 210], in contrast with previous less-efficient staged approaches [136, 191]. These approaches can still be sub-optimal, however, in that they can struggle to provide *useful* training stimuli to a closed-loop agent. In particular, we identify three key issues in recent SOTAs: 1) they have a limited view of safety-criticality, e.g., focusing only on inducing collisions or near-misses; 2) they lack reactivity to an ego agent's behavior diversity; and 3) their optimization objectives tend to maximize "unrealistic" and overly-aggressive adversarial behavior, limiting their usefulness for balanced model training.

Therefore, in this chapter, we propose and evaluate a method for **S**kill-**E**nabled **A**dversary **L**earning (SEAL), which yields significantly improved downstream ego behavior, in closed-loop training with safety-critical scenario generation. Our method addresses the identified limitations in prior art by introducing two novel components, as shown in Figure 6.1. First, we introduce a learned objective function to *anticipate* how a reactive ego agent will respond to a candidate

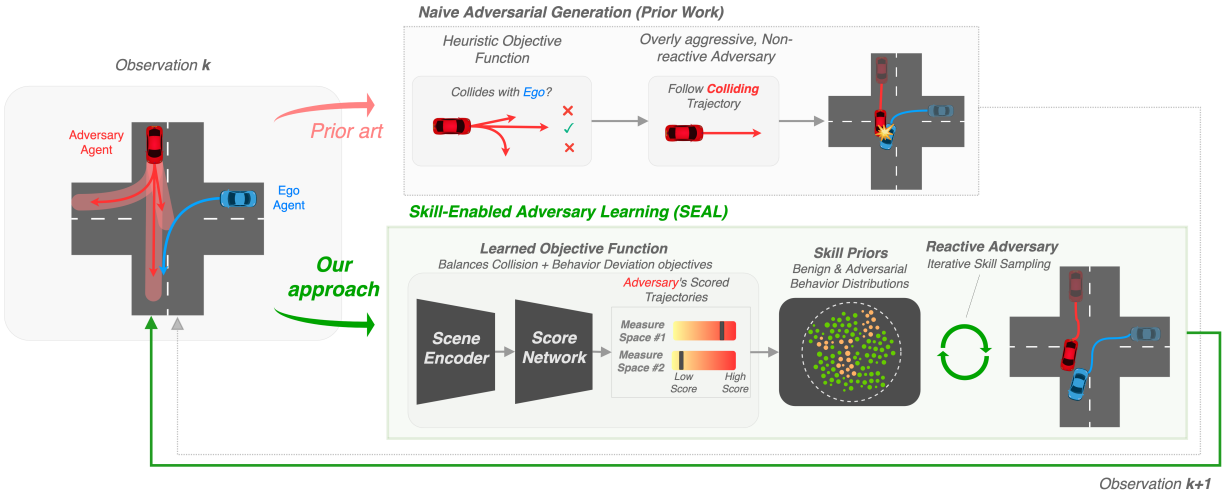


Figure 6.1: An overview of SEAL. Our scenario generation approach leverages a learned objective function and an adversarial skill-based, reactive policy, for improved adversary realism and more effective closed-loop training, leading to safer autonomous driving agents, compared to previous approaches such as CAT [210] and GOOSE [134].

adversarial agent behavior. We quantify both collision closeness and induced ego behavior deviation, thus providing a broadened understanding of safety-criticality. Second, we develop a skill-enabled, reactive adversary policy; in particular, inspired by human cognition, we leverage a hierarchical framework that is akin to how humans operate vehicles [114] and we create an adversarial prior that selects human-like *skill primitives* to increase criticality while maintaining realism.

Furthermore, we argue that safety-critical scenario generation should be evaluated based on behavior realism and usefulness for ego policy improvement, not just induced criticality. Prior work often assesses ego policies on generated scenarios where safety-critical behavior remains effectively *in-distribution* with respect to training data and heuristic perturbations [56, 210]. To address this, we build on the `SafeShift` framework established in Chapter 3, to identify real (non-generated) but safety-relevant scenarios, enabling a more realistic, out-of-distribution evaluation. While in-distribution performance is informative, real-world performance on challenging scenarios is ultimately most important.

In summary, this chapter comprises three main contributions: 1) We propose two novel techniques for safety-critical perturbation: (i) a learned objective function to select candidate trajectories; and (ii) an adversarial skill-based, reactive policy for more realism in adversary behavior; 2) We design an improved evaluation setting for closed-loop training, utilizing real-world safety-relevant scenarios in contrast to just in-distribution generated scenarios; and 3) We provide results on several key experiments, showing an increase of more than 20% in ego task success rate over SOTA baselines, across scenarios generated closed-loop by our proposed framework, across scenarios generated closed-loop by previous SOTA baseline frameworks, *and* across real-world safety-relevant scenarios.

This chapter is based on work done with my collaborators [156]. Our code and tools are

freely available at `https://github.com/cmubig/SEAL`.

6.1 Related Work

6.1.1 Scenario Generation in Autonomous Driving

Approaches for generating scenarios that reproduce the distribution of *normal* driving behavior have been extensively explored. Some methods ensure the diversity of generated traffic behavior [159, 192], while others aim for controllability through rule-based or language-driven specifications [102, 217, 218]. However, due to the rarity of safety-critical events in recorded data [41, 97, 154], other approaches have focused on directly generating corner-case scenarios by injecting adversarial behaviors. Earlier works in safety-critical scenario generation relied on gradient-based optimization approaches with access to vehicle dynamics [56, 136, 173], a limitation in model-free settings. Other methods, such as diffusion-based approaches [22, 191], are compute-intensive and impractical to be used in a closed-loop manner. Efficient methods like CAT [210] and GOOSE [134], which leverage trajectory prediction priors and reinforcement learning (RL) respectively, prioritize simple collision objectives and are non-reactive to the ego agent. Similarly, [208] employs reactive adversaries but focuses only on collisions for criticality and defines realism via proximity to ground truth trajectories, making it sensitive to distribution shifts. In contrast, our approach efficiently generates reactive, nuanced adversarial behavior across multiple axes of criticality, providing a stronger closed-loop training signal.

6.1.2 Robust Training and Evaluation in Autonomous Driving

Several techniques for robustifying AD policies against safety-critical and out-of-distribution scenarios have been explored. Formal methods, such as Hamilton-Jacobi (HJ) reachability, have been utilized in various driving tasks, but struggle with dimension scaling [24, 131]. Similarly, domain randomization has been used as a form of data augmentation (e.g., randomizing vehicle control parameters [172] or scenario initial states [65]) but requires excessive sampling to cover a sufficient domain size. Thus, adaptive stress testing [41, 79] and adversarial training have been increasingly used, either as a fine-tuning scheme [136, 191] or in a fully closed-loop training pipeline [173, 210], providing continuous feedback to an ego agent. However, these approaches still tend to optimize for naive collision objectives alone.

Evaluation of robust training and scenario generation approaches is crucial. Many works evaluate generated scenarios against fixed rule-based or replay ego planners alone [22, 33, 134, 136, 160], offering limited insights into the efficacy of adversarial agents against more sophisticated ego agents. Additionally, adversarially-trained ego policies are often tested on scenarios perturbed by the same adversarial method used in training [56, 173, 191, 210], leading to in-distribution evaluations. Conversely, we focus on out-of-distribution evaluation of well-trained, reactive ego policies, in both adversarial scenarios perturbed by *other* SOTA approaches, as well as real safety-relevant scenarios.

Out-of-distribution evaluation has been well-explored in AD trajectory prediction [70, 123, 154, 200], but these approaches often aim to characterize an entire scenario without focusing on

a single ego driver or identifying a specific adversary. In AD control tasks, some prior work has explored out-of-distribution settings, such as CARNOVEL [42, 43], which tests unseen scenario types like roundabouts. Additionally, Lu et al. [103] evaluate across real-world scenarios of various difficulty levels, but do not hold out the hardest scenarios during training. Our approach thus addresses this gap by offering a more comprehensive and rigorous evaluation, across a wide set of adversarial and real-world scenarios.

6.2 Preliminaries

In this section, we extend the closed-loop motion planning and scenario modification definitions, from Section 2.2.2 and Section 2.3.2, to define our closed-loop adversarial perturbation task. Given a base scenario $s = (\mathbf{X}, \mathbf{M}, \texttt{meta})$, we clarify that `meta` includes `ego` and `adv` IDs to refer, respectively, to the agent IDs of the ego vehicle (to be controlled in simulation) and the adversarial vehicle (to be perturbed to induce criticality).

In this task, a base scenario, $s \in \mathcal{S}$, is selected and modified by the perturbation function $\mathcal{P}$, outputting $\mathcal{B}_{\text{adv}}$ so as to increase the anticipated criticality of the ensuing interactions with respect to the ego learning agent, while maintaining realism. The ego agent is then rolled out in the modified scenario, according to $\mathcal{B}_{\text{ego}}$, receiving some training impetus to update its behavior parameters, and the cycle (i.e., "loop") repeats. All other agents simply reproduce their original trajectories $\{X_j \mid j \notin \{\texttt{ego}, \texttt{adv}\}\}$ from s.

Importantly, $\mathcal{P}$ may condition on the original scenario, s, as well as past episode roll-outs, $\left\{\widetilde{\mathbf{X}}^{(k)}\right\}_{k=1}^{\leq K}$, but must not access future planned ego trajectories for an upcoming episode. In practice, during training, we maintain a queue of the most recent K perturbation roll-outs for each base scenario; during evaluation, we instead run K sequential perturbation–simulation steps and use the final roll-out as the adversarial scenario.

6.3 Approach: Skill-Enabled Adversary Learning for Scenario Generation

To increase scenario criticality while preserving realism, we propose the **S**kill-**E**nabled **A**dversary **L**earning (SEAL) approach for perturbation-based scenario generation. Similar to CAT [210], SEAL employs a probabilistic trajectory predictor π_{gen} to sample plausible future adversary trajectories conditioned on a fixed history portion of s. That is, we sample a set of N_{cand} candidate adversary trajectories as $\{\hat{X}_{\text{adv},i}\}_{i=1}^{N_{\text{cand}}} \sim \pi_{\text{gen}}(X_{\text{adv}} \mid s)$. However, directly selecting and executing one of these samples, as in CAT, has three main limitations: 1) it measures criticality only via collisions, ignoring, e.g., forced ego hard braking or swerving; 2) it prevents reactivity to ego decisions; and 3) it often generates non-human-like behavior, driving straight at the ego with no avoidance. SEAL addresses these with a learned objective for flexible trajectory selection and an adversarial skill policy for more human-like and reactive behavior.

6.3.1 Learned Objective Function

Many previous works rely on heuristic approaches to select the *best* trajectory from a candidate set to be assigned to the behavior of the adversary agent, i.e., directly setting $\mathcal{B}_{\text{adv}}$ equal to some proposed $\hat{X}^{*}_{\text{adv}} \in \{\hat{X}_{\text{adv},i}\}_{i=1}^{N_{\text{cand}}}$. For instance, CAT [210] compares bounding box overlaps across the previous K episodes in all candidate routes, selecting the one which collides with the most previous ego roll-outs at the earliest timestep or is closest to a collision, otherwise. We instead aim to select among candidate trajectories in a more flexible way that captures both closeness to collision as well as likelihood of anticipated ego behavior deviation (e.g., causing the ego to swerve or execute a hard-brake maneuver).

We frame the problem as a supervised regression task. First, we build a dataset of simulated outcomes, where in each base scenario, we roll-out and observe all N_{cand} trajectory pairs of ego and adversarial agents, ($\widetilde{X}^{(K+1)}_{\text{ego}}$, $\hat{X}^{(K+1)}_{\text{adv},i}$). To keep ego behavior as a black-box in downstream closed-loop training, we have the ego follow a reactive heuristic policy during this stage. We then obtain ground truth values from the collected demonstrations, using the following scoring functions, similar to measure functions used in prior work [63, 154]:

$$f_{\text{coll}} = \exp\left(-\frac{1}{b}\min_{t}\left|\left|\widetilde{X}^{(k),t}_{\text{ego}} - \widetilde{X}^{(k),t}_{\text{adv}}\right|\right|_2\right) \tag{6.1}$$

$$f_{\text{diff}} = 1 - \exp\left(-\frac{1}{b}\sum_{t}\left|\left|\widetilde{X}^{(k-1),t}_{\text{ego}} - \widetilde{X}^{(k),t}_{\text{ego}}\right|\right|_2\right), \tag{6.2}$$

where $b \in \mathbb{R}$ is a hyperparameter controlling sensitivity to distance values. Both Equation (6.1) and Equation (6.2) map to $[0, 1]$, where 1 indicates maximal criticality and 0 indicates minimal. Equation (6.1) captures collision closeness between the ego and adversary over a given roll-out, while Equation (6.2) captures ego behavior difference between two episodes. However, instead of only assessing past episodes, we propose to *predict* these measures for a roll-out yet to happen by training a neural network, π_{score} (detailed in Section 6.3.3). This π_{score} network aims to predict f_{coll} and f_{diff} conditioned on a previous $\widetilde{X}^{(k)}_{\text{ego}}$ and the proposed $\hat{X}^{(K+1)}_{\text{adv},i}$. The final score for ranking candidate trajectories is the sum of the predicted f_{coll} and f_{diff} values from π_{score}, averaged over the K previous ego roll-outs. By using π_{score} in place of additional heuristic simulation, we enable scoring to be conditioned on actual ego policy roll-outs and substantially reduce runtime overhead.

6.3.2 Adversarial Skill Learning

We design a reactive policy π_{adv}, to guide the adversary's behavior $\mathcal{B}_{\text{adv}}$, unlike recent works [134, 210], where the selected adversary follows a predefined trajectory. This adversarial policy observes and acts in a closed-loop simulator alongside the ego policy. In this context, skill-based hierarchical policies are appealing approaches as they capture maneuvers at a higher abstraction, compared to the low-level actions of a simulator, corresponding more closely to how humans operate vehicles [114].

We build upon prior work, ReSkill [133], which utilizes expert demonstrations to extract paired observation and action sequences as state-conditioned "skills" which are then embedded using a Variational AutoEncoder (VAE). Additionally, a state-conditioned prior network is

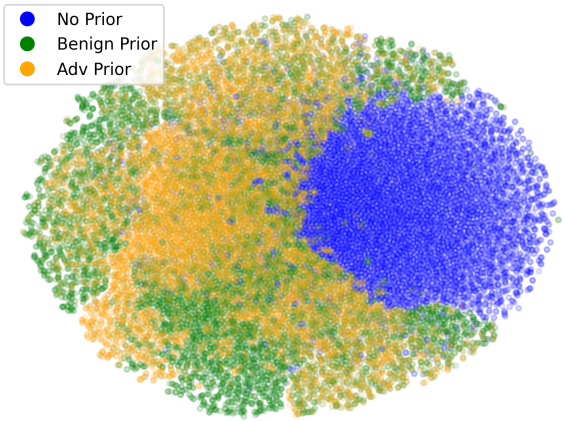


Figure 6.2: Skill space visualized with t-SNE [169]. Benign and adversarial priors map to several regions representing useful, human-like skills, with meaningful separation and overlap.

trained to map from a state to a useful location in the VAE's latent space to be decoded into a reconstructed skill for the agent to follow.

In our work, we separate the demonstrated skills into adversarial (i.e., those ending in a collision or near-miss) and benign skills (i.e., those avoiding a collision while staying on road). We use a sliding-window partitioning scheme that excludes segments starting within twice the skill horizon before an out-of-road event, and labels as adversarial those within the same window before a collision. This reflects the intuition that not only the final skill but also preceding behavior contributes to unsafe outcomes. We then train two prior networks in parallel with a shared-skill VAE: benign skills flow through a "benign" prior while adversarial skills flow through an analogous "adversarial" prior. In this way, the adversarial agent policy, π_{adv}, leverages the adversarial prior to select skills likely to lead to safety-critical outcomes. Furthermore, because each prior is implemented as a real-valued non-volume preserving transformation trained on observed data (as in ReSkill), sampled noise vectors bijectively correspond to plausible, in-distribution behaviors. Figure 6.2 visualizes the learned skill spaces over uniformly sampled states; regions of overlap correspond to skills which may be useful to both an adversarial and benign agent (e.g., lane-keeping, smooth kinematics, etc.) while distinct regions correspond to skills only useful for that particular agent (e.g., for an adversary: cutting-off another vehicle, hard-braking in a dangerous way, etc.).

To integrate this skill module with the trajectory generation and ranking discussed in Section 6.3.1, we first select the highest ranking candidate trajectory, as $\hat{X}^*_{\text{adv}}$. We derive goals and subgoals from this selected trajectory to provide to π_{adv} as navigation information. Skills are then executed in a hierarchical manner as in [133]: at the start of the episode or when a skill has completed, a new skill is selected based on the current observation and adversarial prior. The agent then decodes that skill, in a closed-loop manner, into raw actions. To further increase safety-criticality, the adversary initially exactly follows $\hat{X}^*_{\text{adv}}$ before switching to this adversarial skill policy at a fixed offset before the anticipated point of maximal collision risk.

6.3.3 SEAL Implementation Details

For training and validating both the learned objective function and skill spaces, we leverage the well-established Waymo Open Motion Dataset (WOMD) [37] dataset, as well as a subset of scenarios therein labeled by Waymo as containing interacting agents. A further subset of 500 of these scenarios has been used by prior work, and we henceforth refer to this set as `WOMD-Normal` [67, 210]. We split these scenarios into 400 training and 100 evaluation examples.

For π_{gen}, we utilize a pre-trained DenseTNT [50] trajectory prediction model, as used by CAT. π_{gen} takes as input the first one second of $\mathbf{X}$, as well as the static `meta` and map information $\mathbf{M}$, and produces 32 candidate eight-second future adversary paths. We use the MetaDrive simulator [88] and its included IDM policy [165] as the heuristic reactive agent to collect imperfect demonstration data, described and utilized in both Section 6.3.1 and Section 6.3.2. For data augmentation, **all** agents in the scenario follow the IDM policy and produce useful demonstrations, rather than collecting examples from solely the ego. We extract subgoals from each trajectory using MetaDrive's default waypoint logic, placing checkpoints every 8 meters as navigation input for π_{adv}.

We implement π_{score} as a VectorNet-style polyline encoder [45], followed by a multilayer perceptron decoder to the predicted values of f_{coll} and f_{diff}. We use an MSE loss objective on the sum of the two values, ensuring equal weight to both predicted measures. For π_{adv}, we leverage the skill embedding framework from [133], with identical architectures and loss functions across our two parallel prior networks. We empirically set the hyperparameter b in Equation (6.1) and Equation (6.2) to 8, use a skill time horizon of 10 steps, and fix K to 5 (consistent with CAT).

6.4 Experimental Setup

We leverage SEAL to generate scenarios for two primary purposes: providing data augmentation during closed-loop training of reinforcement learning (RL) agent policies, and providing a means of evaluating such agents' capabilities.

6.4.1 Policy Training

For closed-loop training of an ego agent policy, we leverage the `WOMD-Normal` set along with the MetaDrive simulator [88], described in Section 6.3.3. Then, we follow the curriculum training approach proposed by CAT [210], where a random base scenario $\mathbf{S}$ from the train split is selected and has a random chance of being perturbed; this perturbation chance increases throughout the training process. Agents observe the environment via simulated LiDAR returns and navigation information based on their original destination in $\mathbf{X}$. Agents act on the environment with normalized steering and acceleration forces as $\mathbf{a}$; the ego and adversarial agents follow either a policy or a predefined trajectory, while all other agents follow their original trajectory in $\mathbf{X}$.

We utilize ReSkill [133] as our underlying RL algorithm, a recent SOTA approach in hierarchical RL. We use our skill space built in Section 6.3.2, utilizing the benign prior rather than the adversarial one. The low-level action learned by the ReSkill agent is a remediating $\Delta\mathbf{a}$ ad-

justment to the action decoded based on the current skill and state pair, $\mathbf{a}'$, while the high-level action selects the noise vector to be passed to the prior. Thus, the action sent to the environment is $\mathbf{a} = \mathbf{a}' + \Delta\mathbf{a}$. Actions are performed at a 10Hz rate, and all agents are trained for one million timesteps in total, empirically sufficient for consistent policy convergence.

6.4.2 Evaluation Settings

Many previous works evaluate agent performance, in-distribution, on a held-out subset of their *own* generated scenarios [56, 173, 191, 210]. For additional comprehensiveness, we propose to utilize the `SafeShift` framework, for identifying real-world safety-relevant base scenarios, denoted as `WOMD-SafeShift-Hard`. We start by identifying scenarios containing interacting agents labeled by Waymo. We then apply `SafeShift` 's hierarchical scoring to these agents and select scenarios where the *interacting* agents have trajectory scores in the top 20th percentile across WOMD, randomly sampling 100 scenarios therein. The ego and adversary agents are assigned to the interacting agents with the higher and lower trajectory score, respectively.

We baseline SEAL against two recent SOTA safety-critical scenario generation approaches, that can be utilized in a closed-loop manner: CAT [210] and GOOSE [134]. CAT heuristically chooses a trajectory from π_{gen} to apply to the adversarial agent; we use the same π_{gen} function for both CAT and SEAL, for fairness. GOOSE learns to iteratively modify control points of a NURBS [105] curve fit to the original adversary's trajectory, observing the outcome of each roll-out. We train GOOSE against the MetaDrive IDM agent using the `WOMD-Normal` training set and GOOSE's "deceleration" task goal—induce a collision while maintaining kinematic feasibility. For consistency, we limit the number of GOOSE policy steps (i.e., observed roll-outs) to $K = 5$.

6.4.3 Metrics

Within MetaDrive, episodes are terminated when the ego agent either arrives safely at its goal (`Success`), collides with another agent (`Crash`), or violates an off-road constraint (i.e., crosses a road edge or yellow median; `Out of Road`). As such, we report these corresponding rates as the key metrics for ego performance (as in Section 2.2.2), following prior work [210].

For evaluating generated scenario quality, we examine the induced ego `Success` rate, across all tested ego methods. We derive a realism metric based on distributional measures, following MixSim [160] and other prior work [116, 217]. In particular, we utilize the Wasserstein distance (WD) over adversarial "profiles"—normalized histograms constructed from the adversary's yaw rates, acceleration values, and out-of-road rates. All WD values are computed via comparison to profiles derived from the original X_{adv} in s, which we average to compute an overall `Realism` meta-metric. We also report relative collision velocities along the contact normal, as in [136], along with head-on collision rates and severe head-on rates (where severity is defined as collision velocity exceeding 5 m/s, thereby filtering out low-speed, glancing incidents).

Table 6.1: Ego performance on adversarially-perturbed (a, b, c) and unmodified, real-world (d, e) scenarios. `WOMD-Normal` are WOMD [37] scenarios with basic interactive agents labeled by Waymo; `WOMD-SafeShift-Hard` refers to `SafeShift` -mined real scenarios in WOMD. Adversarially-perturbed scenarios use `WOMD-Normal` as base scenarios, in both training and evaluation settings. *Higher* success rates and lower crash and out of road rates are better. Ego realism scores are shown in (f), averaged over settings (a–e) using Wasserstein distance (WD); lower is better.

(a) WOMD-Normal, GOOSE-Gen

Training	Success	Crash	Out of Road
None	0.59 (0.00)	0.41 (0.00)	0.00 (0.00)
No Adv	0.41 (0.06)	0.37 (0.02)	0.23 (0.04)
GOOSE	0.37 (0.07)	0.35 (0.09)	0.30 (0.17)
CAT	0.35 (0.03)	0.27 (0.02)	0.39 (0.06)
SEAL	**0.44** (0.04)	0.27 (0.00)	0.27 (0.00)

(b) WOMD-Normal, CAT-Gen

Training	Success	Crash	Out of Road
None	0.18 (0.00)	0.82 (0.00)	0.00 (0.00)
No Adv	0.32 (0.01)	0.46 (0.02)	0.22 (0.01)
GOOSE	0.25 (0.10)	0.47 (0.02)	0.31 (0.04)
CAT	0.32 (0.03)	0.32 (0.03)	0.40 (0.00)
SEAL	**0.42** (0.02)	0.32 (0.04)	0.24 (0.02)

(c) WOMD-Normal, **SEAL**-Gen

Training	Success	Crash	Out of Road
None	0.32 (0.00)	0.68 (0.00)	0.00 (0.00)
No Adv	0.33 (0.03)	0.50 (0.05)	0.21 (0.04)
GOOSE	0.26 (0.08)	0.46 (0.00)	0.27 (0.06)
CAT	0.31 (0.00)	0.34 (0.04)	0.36 (0.02)
SEAL	**0.38** (0.04)	0.36 (0.01)	0.25 (0.06)

(d) WOMD-Normal

Training	Success	Crash	Out of Road
None	1.00 (0.00)	0.00 (0.00)	0.00 (0.00)
No Adv	0.48 (0.02)	0.21 (0.01)	0.28 (0.04)
GOOSE	0.44 (0.13)	0.23 (0.03)	0.34 (0.10)
CAT	0.50 (0.02)	0.15 (0.06)	0.36 (0.10)
SEAL	**0.59** (0.01)	0.15 (0.00)	0.27 (0.01)

(e) WOMD-SafeShift-Hard

Training	Success	Crash	Out of Road
None	0.97 (0.00)	0.01 (0.00)	0.02 (0.00)
No Adv	0.28 (0.05)	0.38 (0.05)	0.33 (0.02)
GOOSE	0.19 (0.04)	0.42 (0.06)	0.36 (0.04)
CAT	0.24 (0.00)	0.38 (0.03)	0.37 (0.05)
SEAL	**0.38** (0.02)	0.29 (0.02)	0.33 (0.04)

(f) Aggregate Realism

Training	Yaw WD	Acc WD	Road WD
None	0.014	0.269	0.004
No Adv	0.147	**3.041**	**0.252**
GOOSE	0.152	3.052	0.312
CAT	0.154	3.050	0.374
SEAL	**0.146**	3.074	0.270

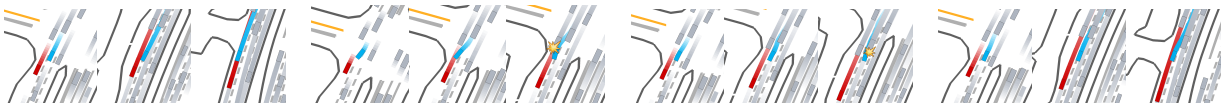
Table 6.2: Scenario generation quality. Results are averaged over all tested ego models. WD measures are Wasserstein distances over adversary behavior; a lower value indicates greater realism. Lower collision velocities (m/s) and head-on rates are better. A *lower* ego `Success` is better, as this table assesses safety-critical effectiveness.

Eval Scenario Type	Ego Success	Realism WD	Yaw WD	Acc WD	Road WD	Coll. Vel.	Head-On	Head-On (Sev.)
WOMD-Normal	60.0%	0.056	0.120	0.020	0.027	2.849	06.7%	04.2%
WOMD-SafeShift-Hard	41.3%	0.069	0.116	0.044	0.043	2.206	00.0%	00.0%
WOMD-Normal, GOOSE-Gen	43.0%	0.401	0.124	0.601	0.482	4.744	13.2%	11.7%
WOMD-Normal, CAT-Gen	**29.6%**	0.167	0.123	0.305	0.074	4.136	**07.1%**	05.9%
WOMD-Normal, **SEAL**-Gen	31.9%	**0.108**	**0.121**	**0.157**	**0.049**	**2.950**	09.2%	**03.6%**

6.5 Results

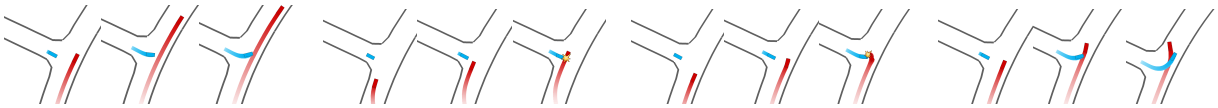
We report the median and interquartile range (IQR) over four seeds, for greater statistical robustness. These statistical summaries are computed independently over each metric, so `Success`, `Crash`, and `Out of Road` may not sum to 100%. We also evaluate a non-reactive ego replay policy (`Replay`), which rolls out the original X_{ego} trajectory, as well as a ReSkill [133] agent trained without any adversarial scenario generation (`No Adv`). Note that due to re-simulation limitations, `Replay` in `WOMD-Normal` and `WOMD-SafeShift-Hard` may have a nonzero failure rate.

Downstream Performance. Our closed-loop training results are summarized in Table 6.1. SEAL-trained policies average a $\mathbf{21.5\%}$ **increase** in `Success` rate relative to the top baseline in each setting, achieving a strong balance between `Crash` and `Out of Road` rates. While a baseline-trained policy may have slightly better performance on one failure type, it is achieved by sacrificing performance against the other. Compared to GOOSE and CAT, SEAL training



(a) Ego Replay (b) GOOSE [134]-Trained (c) CAT [210]-Trained (d) **SEAL**-Trained

Figure 6.3: Qualitative examples of **driving policies**. The **blue** ego is a learned agent while the **red** adversary is fixed. (a) shows the original human trajectory from `WOMD-SafeShift-Hard`, while (b), (c), and (d) show ego behaviors learned in different pipelines.



(a) No Perturbation (b) GOOSE [134]-Gen (c) CAT [210]-Gen (d) **SEAL**-Gen

Figure 6.4: Qualitative examples of **scenario perturbation**. The **blue** ego follows a fixed replay policy while the **red** adversary is modified. (a) shows the original `WOMD-Normal` scenario, while (b), (c), and (d) show perturbations generated by GOOSE, CAT, and SEAL, respectively.

yields more realistic yaw and road compliance but less realistic acceleration, indicating stronger braking and more disciplined in-lane maneuvering to manage criticality. Despite high kinematic realism, `No Adv` egos crash frequently due to lack of experience in challenging scenarios and resulting poor reactivity.

We highlight qualitative examples of ego behavior in Figure 6.3, showcasing how different training regimes influence the execution of the same *benign* skills, in a scenario drawn from the `WOMD-SafeShift-Hard` set of real, safety-relevant scenarios. While all ego policies operate within the same offline-learned skill space, their online adaptation differs across training methods. The `Replay` ego depicts the ground truth human trajectory, which merges safely into the right lane. The GOOSE-trained ego initiates the merge too early and fails to recover in time, resulting in a collision. The CAT-trained ego begins to merge later but hesitates under pressure, slows down, and is rear-ended. In contrast, the SEAL-trained ego merges with a sufficient gap while accommodating a close-following tail vehicle, resulting in a smooth and safe maneuver. These differences highlight how SEAL's more realistic and nuanced adversarial training scenarios better prepare ego policies to navigate challenging interactions effectively.

Scenario Generation Quality. To directly assess scenario generation quality, we aggregate metrics in Table 6.2, averaged over all ego methods. Although CAT scenarios induce a lower ego `Success` rate and raw head-on rate than SEAL scenarios, SEAL scenarios exhibit the highest `Realism` among scenario generation approaches, a $\mathbf{35.3\%}$ **improvement**, contributing to SEAL-trained policies' superior downstream performance.Furthermore, SEAL's collision velocities and severe head-on rates are far lower than baseline approaches.

We also showcase qualitative examples of the tested scenario generation approaches in Figure 6.4, using a fixed ego `Replay` policy to isolate differences in adversary behavior. CAT and GOOSE both produce aggressive trajectories that lead to collisions: CAT stops and turns directly into the ego, while GOOSE swerves across the lane and slows in the ego's path to force

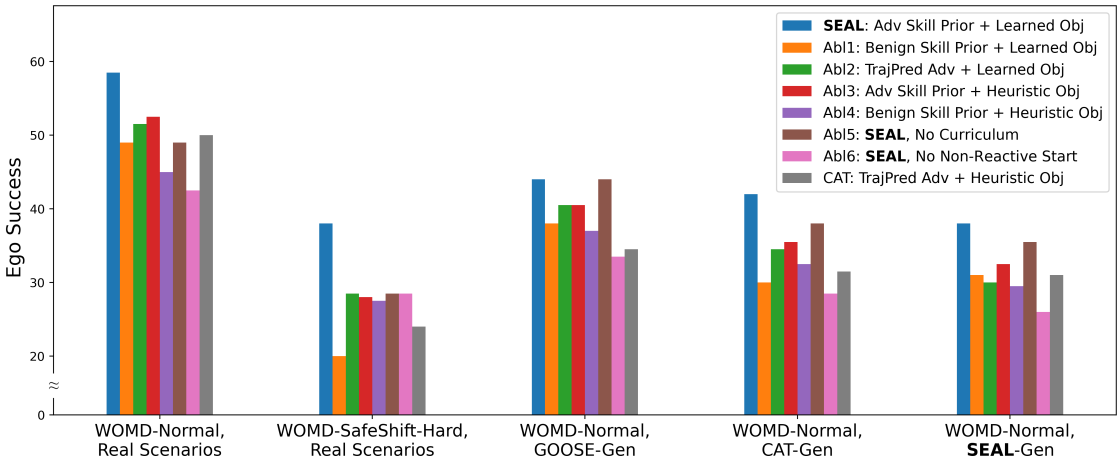


Figure 6.5: Ablation study on SEAL scenario generation training pipelines. Our full approach with learned objectives (Section 6.3.1) and adversarial skill policies (Section 6.3.2) produces the strongest downstream agents, across all five evaluation settings.

a t-bone. In contrast, the SEAL adversary exhibits more nuanced behavior, slowing down to let the ego catch up, moving away at the last moment to induce a near-miss, and thus demonstrating interesting *adversarial* skill behavior.

Ablation Studies. To further investigate how different components of SEAL affect downstream training, we perform extensive ablation studies shown in Figure 6.5, as well as comparing against CAT as it is a slightly stronger baseline than GOOSE. We study the effect of our learned objective function by comparing it to the heuristic, bounding box overlap approach used by CAT (`Learned Obj` and `Heuristic Obj`, respectively). Similarly, we compare our adversarial skill policy (`Adv Skill Prior`) with a benign prior variant (`Benign Skill Prior`) and a predefined trajectory following policy (`TrajPred Adv`). We also compare SEAL against two additional ablations: one trained *without* curriculum, and another *without* the initial non-reactive start. Our full SEAL approach performs best across all settings; both the learned objective function and adversarial skill policy are essential, while the curriculum and non-reactive start further improve performance.

6.6 Discussion

As autonomous driving (AD) systems advance, ensuring safety remains essential. While recent safety-critical scenario generation techniques show promise, they often lack the realism, reactivity, and nuance needed to provide strong training signals for closed-loop agents. We thus introduced Skill-Enabled Adversary Learning (SEAL) as a perturbation-based safety-critical scenario generation approach, combining a learned objective function and an adversarial skill policy. In all test settings—across both real-world challenging scenarios and generated scenarios by SEAL and other SOTA methods—SEAL-trained policies achieved significantly higher success rates, with a more than 20% relative increase. Upon deeper analysis, SEAL-generated scenarios contain less aggressive but more realistic adversaries, helping to explain the observed ego agent

improvements. We argue that realism metrics, downstream task utility, and out-of-distribution evaluation settings are vital in assessing adversarially-perturbed scenarios.

While SEAL is quite effective, further improvements are still possible. Incorporating finer-grained metrics into the objective function could enable more adaptive and controllable generation beyond safety-criticality alone. Additionally, enhancing realism metrics to reflect human decision-making at the skill-level could provide deeper insights into scenario quality. We encourage future work to explore these topics.

Chapter 7

Real-World Crash Grounding for Improved Safety-Critical Scenario Generation

In this final technical chapter, we further advance the scenario modification pillar of enhanced data utilization, as introduced in Chapters 5 and 6, by focusing on more realistic safety-critical perturbations of autonomous driving (AD) scenarios. Our SEAL approach, established in Chapter 6, along with other recent state-of-the-art (SOTA) work like CAT [210], *reasons* over candidate adversary behaviors produced by a pre-trained trajectory generator. This reasoning is then used to select a behavior to roll-out with respect to expert criteria (e.g., attempting to maximize collision closeness to the ego, or forcing the ego to perform harsh avoidance maneuvers, etc.). Despite these advances, generated scenarios by such approaches tend to still be overly aggressive and thus insufficiently realistic.

To generate more realistic safety-critical scenarios, we propose **R**eal-world **C**rash **G**rounding (RCG), which begins by leveraging a large, well-annotated dataset of everyday driving to model moderately unsafe behaviors. To capture more extreme but realistic failures, we reach beyond conventional datasets for autonomous driving prediction and control and we draw from minimally-annotated video corpora originally collected for traffic scene understanding tasks, which include real crash and near-miss cases [20, 193]. This raises the key question of how to meaningfully incorporate such non-aligned data; RCG addresses this by using a representation learning framework to unify and structure behaviors across both sources.

This representation is constructed in two stages. First, we collect and heuristically classify large-scale "safe", "neutral", and "unsafe" behaviors, from everyday but safety-relevant scenarios. We train the embedding using a prototypical contrastive learning (PCL) [86] objective that encourages local structure between contextually similar behaviors while separating distinct safety classes. This enables pre-training over diverse scene contexts while defining a structured notion of "unsafe" behavior. Second, we fine-tune the representation on a small number of real accident and near-miss interactions. Since these examples lack motion annotations, we use off-the-shelf perception tools to extract approximate trajectories that meaningfully refine the embedding around real-world unsafe behaviors.

Once trained, our embedding space yields an adversarial selection process by quantifying

how closely a generated trajectory matches real unsafe behaviors in context, ultimately serving as an optimization objective. This objective can then be easily integrated on top of existing scenario generation approaches. By guiding adversarial agents to maximize realistic criticality, we generate safety-critical scenarios that are both more plausible as well as effective for training and evaluating driving models. Thus, RCG also addresses a broader methodological challenge: using representation learning to extract useful structure from non-aligned, weakly-labeled data, enabling it to inform downstream tasks despite minimal task-specific annotation.

Our contributions are thus as follows: 1) A safety-informed representation space that captures contextualized driving behaviors with global and local structure, and enables integration of non-aligned data sources not previously used or intended for this task; 2) A crash-grounded scoring objective for adversarial scenario generators that improves the plausibility and practical usability of generated scenarios; and 3) Extensive validation showing that ego agents trained in closed-loop with our generated adversaries display more robust performance, improving driving success rates by an average of 9.2% across all tested environments and base scenario generation approaches.

This chapter is based on our paper currently under review, with a preprint available [157]; upon publication, our code and tools will be made freely available at `https://github.com/cmubig/RCG`.

7.1 Related Work

7.1.1 Autonomous Driving Scenario Curation

Much work in autonomous driving has focused on curating large-scale real-world driving logs for training and evaluating perception, prediction, and planning modules [16, 37, 187]. These datasets are typically multi-modal, with sensor data and annotations tailored to specific downstream tasks. For instance, perception often relies on image and LiDAR inputs with dense labels [15, 158, 213], while prediction and planning tasks utilize agent trajectories and map features, often in polyline form [37, 83, 187]. While rich and widely used, such datasets overwhelmingly capture nominal behavior and suffer from the well-known "curse of rarity," i.e., the near-absence of safety-critical events [34, 63, 97]. Efforts like [154] attempt to surface more safety-relevant scenes from these corpora, but truly critical interactions remain sparse or entirely missing.

To address this gap, several datasets have been released that capture real-world accidents or near-misses, often from dashcams or fixed-position surveillance footage. Ego-centric video datasets such as MM-AU [38] focus on reasoning and forecasting tasks, while others like TADS [20], SUTD [193], and CADP [146] compile surveillance footage of crashes with varying levels of annotation. These datasets are typically small in scale and offer coarse labels like crash timing, agent roles; only a few, e.g., [196], include a small number of trajectory-level annotations, which are required for most driving behavior learning approaches. In this work, we approximately annotate safety-critical examples from TADS [20] and incorporate them as a fine-tuning stage in a contrastive learning pipeline, enabling semantic grounding on real-world accident structure without requiring full supervision.

7.1.2 Closed-Loop and Adversarial Scenario Generation

Alongside dataset curation, much work has explored generating synthetic autonomous driving scenarios directly. While many methods focus on reproducing normal driving behavior [46, 160, 192], generating adversarial behavior is far more challenging due to the curse of rarity and distribution shift to benign scenarios, and must carefully balance diversity, fidelity, and usability for ego training [34, 156]. Some approaches manually construct starting scenarios based on real-world accident patterns [35, 91, 144, 220], but these often require significant effort and domain expertise, and are limited to a narrow set of scenario types.

Thus, many approaches have turned to adversarial perturbation of real driving data, though these too face limitations. Several works rely on diffusion or black-box optimization methods that are computationally intensive and impractical for closed-loop ego policy development [22, 136, 160, 191]. More efficient alternatives employ quality-diversity optimization [63] or adversarial reinforcement learning [134], but depend on hand-crafted metric functions that are difficult to generalize. Similarly, explicitly closed-loop methods such as CAT [210] and SEAL [156] select behaviors from lightweight generative priors, but over-optimize for collision proximity or induced ego deviation, often resulting in unrealistic adversary behavior. Our work follows this closed-loop adversarial selection paradigm, while emphasizing perturbations *grounded* in a more generalized representation of real-world crash structure.

7.1.3 Representation Learning for Driving Behavior

Learning representations of agent behavior is a foundational component of many autonomous driving systems, particularly in tasks such as motion forecasting and simulation-based policy learning. A common approach leverages encoder-decoder architectures that *implicitly* capture driving semantics in a latent bottleneck [19, 150, 189, 214]. These methods often aim to build expressive embedding spaces using high-capacity models such as Transformers [170] or large-language-inspired tokenization schemes [164]. Recent work has also explored the use of skill-space priors to enable more human-like, hierarchical behavior generation [57, 133]. Other approaches attempt to embed safety awareness directly, either by incorporating risk-aware architectural components [174] or by penalizing predicted collisions across trajectory samples [168]. While these techniques often improve downstream task performance, the resulting latent representations themselves remain difficult to interpret or align with safety-relevant structure.

To imbue learned embeddings with semantic structure more explicitly, contrastive learning has become an increasingly popular strategy in the machine learning community [52, 60]. In autonomous driving, recent works, such as FEND [178] and TRACT [209], apply prototypical contrastive learning (PCL) [86] to enforce structure based on trajectory types and training-stage dynamics. Other approaches, like LIDP [96], rely on traditional contrastive objectives such as the InfoNCE loss [167], augmenting trajectory features to capture individualized driving patterns and style. Building on these ideas, we combine contrastive pre-training with crash-grounded fine-tuning to produce a representation space structured explicitly around safety-class separation, and further refined to reflect real-world accident behaviors.

Figure 7.1: Illustrative examples from TADS [20], processed into `TADS-traj`. In the top example, a vehicle (ID 5, "safe") can be seen approaching from the left, while executing a braking maneuver to reduce the severity of the collision, while a bike (ID 1, "unsafe") fails to react, increasing the criticality. In the bottom example, a vehicle (ID 2, "unsafe") swerves to avoid a car (ID 1, "unsafe") cutting in front of it, causing a third vehicle (ID 4, "safe") to swerve and brake to avoid an accident itself.

7.2 Accident Dataset Processing

To provide safety-critical examples for constructing our embedding space, we curate a small set of real-world accident scenarios derived from the TADS dataset [20]. Originally designed for road-traffic accident detection in CCTV footage, TADS comprises a relatively diverse collection of high-quality surveillance videos capturing vehicular accidents. These fixed, third-person perspectives enable more stable perception, which is essential to capturing the nuanced actions and reactions displayed by traffic participants in response to developing criticality. These types of interactions include subtle failures of anticipation, missed perceptions, and even actions taken by colliding participants to reduce the severity of an inevitable collision; crucially, such behaviors are not represented in large-scale trajectory-level datasets such as the Waymo Open Motion Dataset (WOMD) [37], which are sourced from non-critical driving logs. Incorporating such accident scenarios is thus essential for modeling high-risk behaviors that lie outside the typical training distribution, and enabling grounded adversarial scenario-generation.

Trajectory Extraction. To extract trajectory-level annotations from TADS, we leverage recent advances in off-the-shelf perception tools, which make it increasingly feasible to process video-only datasets without extensive manual labeling. We apply a pipeline of recent foundation models, starting with GroundingDINO [99] for object detection and SAM2 [135] for instance segmentation. Then, AED [39] is used for 2D object tracking, while we utilize XFeat [126] features to further refine temporal associations through keypoint correspondence. Next, metric depth is estimated with UniDepth [125] and temporally smoothed via VideoDepthAnything [26], allowing us to follow OVM3D-Det [64] to estimate and orient 3D bounding boxes. We further smooth these agent bounding box sequences spatio-temporally to yield stable agent trajectories.

Trajectory Filtering. Following automated processing, we conduct a manual filtering and refinement step. From the roughly $1,000$ available videos in TADS, we exclude scenarios that are perception-degraded, feature unavoidable collisions, or contain insufficient context before the

Table 7.1: Distribution of annotated agent roles and observed maneuvers across 385 accident-involved agents. Note, multiple maneuvers may be assigned to a single agent.

Annotation Type	Category	Count
Role	Aggressor	155
	Recipient	162
	Bystander	68
Maneuver	Neutral (none)	40
	Neutral (brake)	7
	Neutral (swerve)	5
	Safe (brake)	142
	Safe (swerve)	86
	Safe (speed-up)	5
	Safe (backup)	1
	Unsafe (none)	83
	Unsafe (brake)	14
	Unsafe (swerve)	20

incident, selecting instead those that highlight subtle or reactive behaviors in the moments preceding an accident. This produces approximately 144 crash-containing *scenarios*, from which we extract 385 accident-involved agent *trajectories*. Each trajectory is manually labeled according to its role in the incident: "aggressor", "recipient", or "bystander". In addition, we annotate one or more observed maneuvers per agent. Each maneuver is classified by both a discrete behavioral type (i.e., brake, swerve, none, etc.) and a high-level safety class (safe, neutral, or unsafe), based on counterfactual reasoning about how the agent's active behavior impacted criticality.

A detailed breakdown of role and maneuver labels is shown in Table 7.1. We highlight representative examples of the scenarios in Figure 7.1, illustrating the sort of complex, nuanced behaviors that are essential for training. We denote this processed and filtered version of TADS as `TADS-traj`.

7.3 Learning A Safety-Informed Embedding Space

Understanding safety-relevant driving behaviors requires an embedding space that captures how agents act under both benign and critical conditions. We aim to learn such a space in a way that supports downstream adversarial scenario-generation, by embedding agent trajectories such that unsafe behaviors are meaningfully structured and distinguishable from safe or neutral ones. This section describes how we construct this safety-aware embedding through four key stages. First, we train a large-capacity, agent-centric model on large-scale trajectory data using a proxy reconstruction objective, producing a general-purpose behavioral encoder. Then, we derive heuristic safety labels and apply contrastive regularization to organize the embedding space and define a structured notion of unsafe behavior. Finally, we refine this space by fine-tuning on a curated set of real-world crash trajectories, adapting the representation to better reflect real safety-critical interactions. This overall process is highlighted in Figure 7.2.

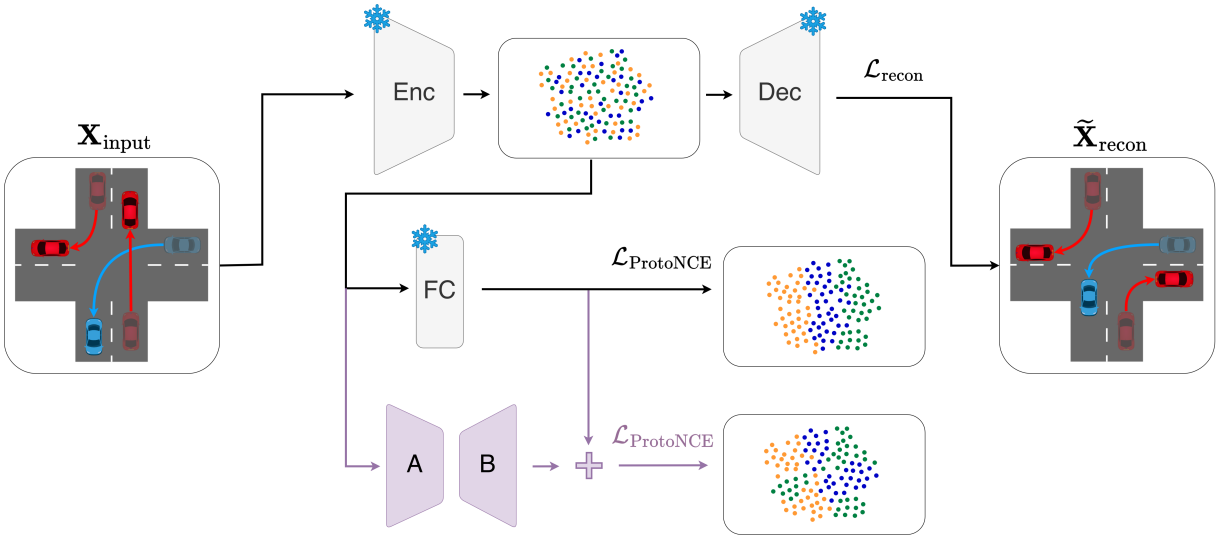


Figure 7.2: Embedding space training overview, as described in Section 7.3. Initial pre-training is performed with reconstruction and contrastive loss objectives. Existing weights are then frozen, and adapters are trained on `TADS-traj` using contrastive loss alone (shown in purple). Contrastive regularization is supervised via "safe", "neutral", and "unsafe" labels.

7.3.1 Behavior Encoding

We begin by constructing a large-capacity encoder that produces dense representations of agent behavior from observed trajectories. Our backbone builds on the Motion Transformer (MTR) [150] framework, a state-of-the-art encoder-decoder approach originally designed for multi-agent trajectory prediction. To adapt this architecture for our purposes, we make two key modifications: 1) we replace its agent history conditioning with a full-trajectory conditioning, enabling the encoder to capture complete behavioral signatures; and 2) we remove map conditioning from the input, to support learning from lower-fidelity settings (i.e., real-world accident videos) where map information may be unreliable or unavailable. These changes preserve the strong interaction modeling and expressive capacity of MTR while tailoring it to our goal of behavior representation.

Using the closed-loop adversarial perturbation task defined in Chapter 6, we begin by considering a scenario $s = (\mathbf{X}, \mathbf{M}, \texttt{meta})$, with `ego` and `adv` IDs in `meta`. Given a target behavior $X_i \in \mathbf{X}$, we follow the standard MTR encoder structure to compute a context-conditioned behavior embedding $z_i = \text{Enc}(X_i,\ \mathbf{X})$, using full trajectories in $\mathbf{X}$ transformed to the local reference frame of agent i at a fixed timestep. We retain the original MTR decoder architecture, $\text{Dec}(z_i)$, and training setup, using its multi-modal output head to reconstruct the input trajectory from z_i, creating $\widetilde{\mathbf{X}}_{\text{recon}}$. The framework produces an overall reconstruction loss, $\mathcal{L}_{\text{recon}}$, comprising trajectory regression and mode selection losses.

7.3.2 Safety Classification

To utilize observed behaviors $\mathbf{X}$ in a safety-informed manner, we first derive large-scale safety labels leveraging the `SafeShift` scenario characterization framework, developed in Chapter 3. For each agent, we compute heuristic safety-relevance scores by aggregating various low-level indicators (e.g., time-to-collision, time headway, trajectory anomaly detection, etc.), applied to both the observed trajectory ("ground truth"), and a counterfactual extrapolation in which the agent in question proceeds passively ("future extrapolated"; i.e., continuing at constant velocity in its current lane). We denote the resulting scores as GT_i and FE_i, respectively.

Next, to estimate the impact of an agent's *active* behavior on downstream safety, we compute the difference between these two scores. This difference, $\texttt{diff}_i = \text{GT}_i - \text{FE}_i$, is then discretized into three classes—"safe", "neutral", and "unsafe"—according to Equation (7.1), where the thresholding value δ is selected to ensure roughly even class distributions across the dataset:

$$y_i = \begin{cases} \text{safe}, & \text{if } \texttt{diff}_i < -\delta \\ \text{neutral}, & \text{if } |\texttt{diff}_i| \leq \delta \\ \text{unsafe}, & \text{if } \texttt{diff}_i > \delta \end{cases} \tag{7.1}$$

Importantly, although counterfactual alternatives are used to compute these labels, we do not use counterfactual data for training. The representation is trained entirely on unaltered, observed behaviors, preserving the realism of the input trajectories.

7.3.3 Contrastive Regularization

To organize the learned representations around safety-relevant behavior, we incorporate supervised contrastive learning via the above labels. Architecturally, we project the $z_i \in \mathbf{Z}$ embeddings obtained from $\text{Enc}(X_i, \mathbf{X})$ through an *additional* fully-connected (FC) layer, denoted FC_{proj}, as in prior work [108, 178]. Then, we ℓ_2-normalize the resulting vector, to obtain v_i. In this way, we can perform regularization on the space spanned by all embeddings $v_i \in \mathbf{V}$ without disrupting the core features in the $\mathbf{Z}$-space which are useful for trajectory reconstruction.

We utilize the standard $\mathcal{L}_{\text{ProtoNCE}}$ objective pioneered in PCL [86], and applied in TrACT [209] and FEND [178], leveraging both prototype-level and instance-level contrastive losses: $\mathcal{L}_{\text{ProtoNCE}} = \mathcal{L}_{\text{inst}} + \mathcal{L}_{\text{proto}}$.
That is, for a particular batch of size B:

$$\mathcal{L}_{\text{inst}} = -\sum_{i=1}^{B} \frac{1}{N_{\text{po},i}} \sum_{i^+=1}^{N_{\text{po},i}} \log \frac{\exp(v_i \cdot v_i^+/\tau)}{\sum_{j=1}^{B} \exp(v_i \cdot v_j/\tau)}, \tag{7.2}$$

where, for a particular sample i, $N_{\text{po},i}$ refers to the number of positive samples (i.e., samples with the same class label) within the current batch, i^+ indexes one such positive sample, and τ is a hyperparameter controlling the temperature of the loss. Since all v_i values are normalized, taking their dot product "$\cdot$" serves to indicate similarity. The prototypical loss term is defined as follows:

$$\mathcal{L}_{\text{proto}} = -\sum_{i=1}^{B} \log \frac{\exp(v_i \cdot c_i/\phi_i)}{\sum_{j=1}^{3} \exp(v_i \cdot c_j/\phi_j)}, \tag{7.3}$$

where c_i and ϕ_i denote the prototype centroid and concentration associated with sample i's class, and c_j and ϕ_j are those corresponding to each safety class $j \in \{\text{safe}, \text{ neutral}, \text{ unsafe}\}$. Both values are updated once per epoch; centroids c_j are smoothed via momentum with coefficient η to promote stability. Centroids are computed over all current embeddings v_i associated with a particular class, while concentration estimates ϕ_j are defined as follows, where $N_{\text{class},j}$ denotes the total number of samples associated with class j and α is a non-negative scalar hyperparameter:

$$\phi_j = \frac{\sum_{i=1}^{N_{\text{class},j}} ||v_i - c_j||_2}{N_{\text{class},j} \log(N_{\text{class},j} + \alpha)} \tag{7.4}$$

Combined, these two terms have a synergistic effect in organizing the representation space, guiding the space to reflect both fine-grained similarities within safety classes and broader structural separation between them. Our overall training procedure is then to optimize the followinge multi-objective loss:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{ProtoNCE}}, \tag{7.5}$$

where λ is a tunable hyperparameter that balances reconstruction fidelity against the strength of the safety-aware contrastive regularization.

7.3.4 Fine-Tuning Representations

We now aim to calibrate our learned behavior representation by fine-tuning using the small but highly-informative dataset processed in Section 7.2, `TADS-traj`. Prior work has shown that adapter-based fine-tuning approaches are more robust to overfitting than full fine-tuning, particularly in low-data regimes [58, 95]. Indeed, empirically we observe that full fine-tuning leads to rapid memorization and degradation of generalization performance. To mitigate this, we employ a LoRA [59] adapter at the encoder bottleneck, inserted in parallel to the contrastive projection head, FC_{proj}, to act upon z_i.

Given $z_i \in \mathbb{R}^{d_1}$ and $\text{FC}_{\text{proj}}(z_i) \in \mathbb{R}^{d_2}$, we follow the original LoRA formulation, introducing a learnable down-projection matrix $A \in \mathbb{R}^{d_1 \times r}$ and up-projection matrix $B \in \mathbb{R}^{r \times d_2}$, where r is the adapter rank. The adapter output is added to the original projection, and the result is ℓ_2-normalized to yield the final embedding:

$$v_i = \text{normalize}\left(\text{FC}_{\text{proj}}(z_i) + BAz_i\right) \tag{7.6}$$

The up-projection weights B are initialized to zero, so that v_i initially matches the pretrained projection. During fine-tuning, only A and B are updated; all other encoder and decoder parameters remain frozen. Training is performed via the contrastive loss $\mathcal{L}_{\text{ProtoNCE}}$ alone, encouraging additional structure in the representation $\mathbf{V}$-space while preserving the generality of the pretrained backbone.

7.4 Real-World Crash-Grounded Adversaries

To generate realistic yet critical driving scenarios, we propose **R**eal-world **C**rash **G**rounding (RCG) as an adversarial selection mechanism, defined over our behavior embedding space created in Section 7.3. Because this embedding space captures high-level semantic structure among

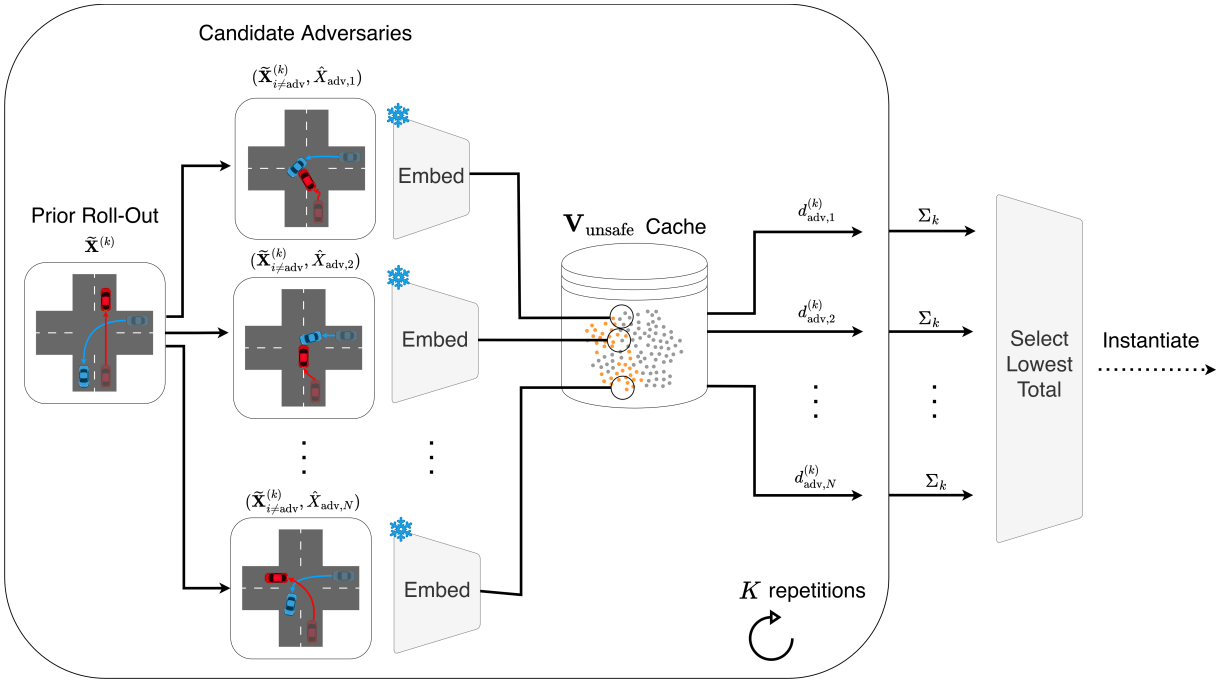


Figure 7.3: RCG adversarial scenario selection, as described in Section 7.4. Candidate behaviors supplant observed behaviors in K prior roll-outs, then are projected into the $\mathbf{V}$-space and ranked by k-NearestNeighbor distance to "unsafe" embeddings.

agent behaviors, organizing them by safety class while still preserving local variations, we propose a distance measure based on k-NearestNeighbors (KNN) to N_{KNN} known "unsafe" embeddings. We then use this KNN-based distance measure to select over *candidate* adversarial behaviors to roll-out against the ego agent, operationalizing the safety-critical perturbation function $\mathcal{P}$ introduced in Section 2.3.2. This overall process is visually represented in Figure 7.3.

Behavior Scoring. Recall that $\mathcal{P}$ maps a base scenario, s, and up to K historical roll-outs, $\{\widetilde{\mathbf{X}}^{(k)}\}_{k=1}^{\leq K}$, to a desired adversary behavior $\mathcal{B}_{\texttt{adv}}$, which is then rolled-out in s alongside $\mathcal{B}_{\text{ego}}$ to construct the next training or evaluation scenario. Using the same formulation as in SEAL, we first sample a set of N_{cand} candidate adversary trajectories from a pre-trained trajectory predictor π_{gen}, resulting in $\{\hat{X}_{\text{adv},i}\}_{i=1}^{N_{\text{cand}}}$. For each candidate $\hat{X}_{\text{adv},i}$ and each historical ego trajectory $\widetilde{X}_{\text{ego}}^{(k)}$, we construct a perturbed scenario $s_i^{(k)}$ by replacing the ego and adversary trajectories in the original scenario's trajectory set X, as in CAT [210], and obtain its embedding $v_{\text{adv},i}^{(k)}$ using the encoder defined in Section 7.3.4.

To enable scoring, we precompute and cache the embeddings of all "unsafe" base scenarios, denoted $\mathbf{V}_{\text{unsafe}}$. This frozen cache provides a consistent empirical reference for evaluating the semantic plausibility of proposed adversarial behaviors. For each candidate, we then compute the average KNN distance across the K perturbed scenarios:

$$d_{\text{adv},i} = \frac{1}{K}\sum_{k=1}^{K} \texttt{KNN\_dist}(v_{\text{adv},i}^{(k)}, \mathbf{V}_{\text{unsafe}}) \tag{7.7}$$

where `KNN_dist` denotes the mean Euclidean distance to the embedding's N_{KNN} nearest neighbors in $\mathbf{V}_{\text{unsafe}}$.

Behavior Selection. Since "unsafe" in our framework denotes general behaviors that elevate criticality, rather than a scalar notion of risk, emphasizing *local* similarity via KNN produces a more precise and flexible objective than distance to class-level prototypes. To further increase interaction potential, we apply a heuristic collision-closeness score to each candidate, following [210], defined as the minimum distance from each historical ego trajectory $\widetilde{X}_{\text{ego}}^{(k)}$ to a candidate $\hat{X}_{\text{adv},i}$, averaged over the K roll-outs. We retain the $N_{\text{int}} < N_{\text{cand}}$ candidates with the lowest average collision-closeness, forming the set $\{\hat{X}_{\text{adv},i}\}_{i=1}^{N_{\text{int}}}$, and finally select the one with the minimum $d_{\text{adv},i}$ from this set as $\hat{X}_{\text{adv}}^{*}$.

This procedure ensures that the selected adversary behavior is both contextually plausible (in embedding space) and likely to be physically proximate to the ego (in trajectory space, subject to ego reactivity), promoting meaningful and nontrivial interaction. The final behavior functional $\mathcal{B}_{\text{adv}}$ is instantiated using the selected trajectory $\hat{X}_{\text{adv}}^{*}$, and may correspond to open-loop behavior (as in [210]) or a closed-loop reactive policy that utilizes the trajectory as a goal (as in [156]).

7.5 Experiment Setup

We validate our approach through a sequence of experiments designed to address three research questions:

- **RQ1:** Does training the ego agent against RCG adversaries result in more robust closed-loop performance?
- **RQ2:** Does our learned behavior embedding meaningfully organize behaviors in a safety-aware manner?
- **RQ3:** Does RCG adversary selection lead to more plausible and effective perturbations than baseline methods?

Each of the following subsections supports one of these research questions, with RQ1 to Section 7.5.1, RQ2 corresponding to Section 7.5.2, and RQ3 to Section 7.5.3. Ablation studies in Section 7.6 further support the validity of these design choices. Although we evaluate all components of the pipeline, our downstream ego training experiments addressing RQ1 provide the most direct and substantial empirical validation of the method's impact.

7.5.1 Closed-Loop Ego Training

To evaluate the downstream utility of our behavior embedding, we perform closed-loop training of an ego agent in the generated safety-critical scenarios from Section 7.4. This directly assesses whether scenarios generated by our approach, compared to those from prior SOTA baselines (i.e., GOOSE [134], CAT [210], and SEAL [156]), provide more effective training stimuli, when tested in both unmodified and adversarially-perturbed scenarios.

We adopt the training and evaluation setup described in SEAL in Chapter 6, using MetaDrive [88] as the simulator, and focusing on the reinforcement learning of an `ego` agent using ReSkill [133] in non-trivial but non-critical base scenarios from WOMD [37]. We use the same 400 base

Table 7.2: Success rate (%) of ego agents trained under different scenario generation pipelines, across seven scenario evaluation types. "Normal" and "Hard" refer to unmodified base scenarios from WOMD [37]. Columns 3–7 correspond to adversarially-perturbed variants of "Normal", each generated by the corresponding method. Each cell reports the mean and standard deviation over four training seeds; higher is better.

Training Pipeline	**Evaluation Setting**						
	Normal	Hard	GOOSE	CAT	SEAL	CAT-RCG	SEAL-RCG
None (Replay)	100.0 (0.0)	97.0 (0.0)	59.0 (0.0)	18.0 (0.0)	32.0 (0.0)	49.0 (0.0)	47.0 (0.0)
No Adv	49.8 (3.7)	29.0 (4.9)	41.0 (3.5)	31.5 (1.8)	31.2 (3.9)	35.8 (3.0)	34.8 (4.7)
GOOSE	43.5 (8.1)	23.0 (8.2)	36.3 (6.2)	25.7 (7.0)	26.2 (7.9)	31.0 (7.2)	32.8 (7.8)
CAT	50.5 (3.6)	29.5 (10.7)	36.8 (5.5)	33.0 (6.3)	32.0 (3.0)	39.2 (6.1)	36.2 (4.8)
CAT-RCG	**52.2 (6.4)**	**33.2 (4.5)**	**41.2 (2.0)**	**38.2 (5.4)**	**35.8 (2.9)**	**41.8 (2.9)**	**40.0 (6.6)**
SEAL	54.5 (6.3)	36.5 (10.2)	39.2 (6.9)	34.0 (3.2)	35.8 (5.1)	37.5 (5.2)	41.0 (4.3)
SEAL-RCG	**55.5 (4.0)**	**36.5 (3.3)**	**46.0 (2.4)**	**38.0 (2.7)**	**38.2 (6.2)**	**43.8 (2.2)**	**41.8 (4.4)**

scenarios for training, with 100 held-out `WOMD-Normal` scenarios and 100 additional safety-relevant base scenarios (`WOMD-Hard`) for evaluation, performing training and evaluation over four independent seeds. We report performance on both sets of unmodified scenarios, as well as on perturbed versions of the `WOMD-Normal` scenes generated by each baseline method and our own approach. Evaluation focuses on ego safety metrics, detailed further in Section 7.6.1.

7.5.2 Embedding Space Creation Details

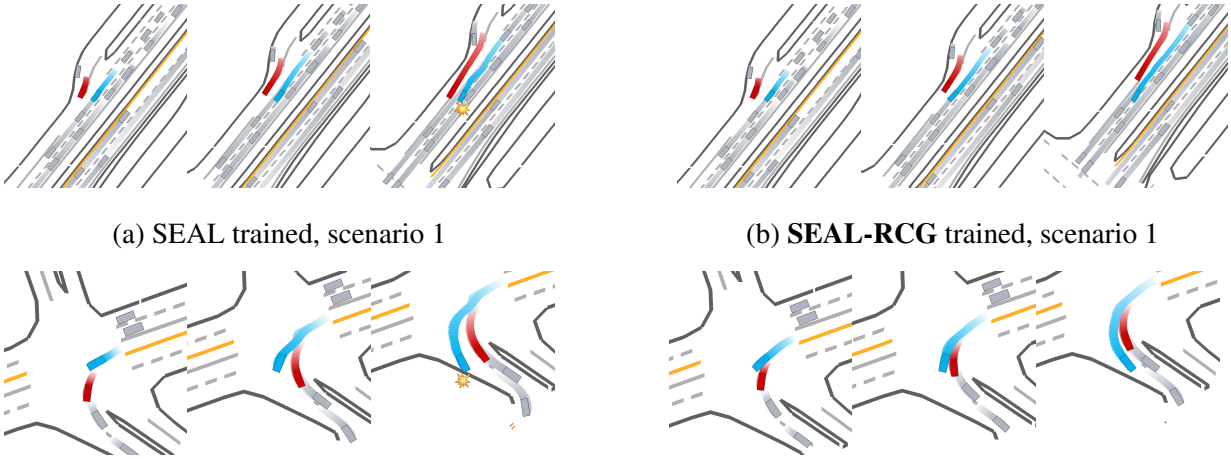
When implementing Section 7.3, we pre-train our representation space using the WOMD [37] dataset, leveraging the MTR [150] implementation and training tools provided in UniTraj [40]. To ensure compatibility with this pipeline, we convert our `TADS-traj` dataset into the ScenarioNet [89] format, as required by UniTraj. We apply a prototype momentum coefficient of $\eta = 0.8$, contrastive temperature $\tau = 0.05$, and set the concentration parameter $\alpha = 10$, following [86]. The contrastive loss weight is tuned to $\lambda = 10$.

While LoRA is typically used in low-rank regimes, our architecture's relatively small hidden sizes (i.e., $d_1 = 256$ and $d_2 = 16$) permits full-rank adaptation ($r = 16$). We find this performs better than lower values for r; in our case, LoRA primarily mitigates overfitting from full fine-tuning, consistent with findings that LoRA may under-perform in low-dimensional settings [92].

To assess the learned embedding space, we evaluate its qualitative structure and quantitative properties, comparing multiple configurations and ablations of the training pipeline, as described in Section 7.6.2.

7.5.3 Adversarial Generation Details

We generate adversarial candidate trajectories, as described in Section 7.4, using a pre-trained DenseTNT [50] model as the generative policy π_{gen}, following prior work [156, 210]. For each adversary agent, we sample $N_{\text{cand}} = 32$ candidate trajectories and retain the top $N_{\text{int}} = 6$ highly interactive behaviors.



(a) SEAL trained, scenario 1

(b) **SEAL-RCG** trained, scenario 1

(c) SEAL trained, scenario 2

(d) **SEAL-RCG** trained, scenario 2

Figure 7.4: Qualitative examples of closed-loop ego driving in **unmodified** hard scenarios from WOMD. In each row, the left depicts ego agents trained with SEAL alone; the right shows agents trained using RCG (Section 7.4) instantiated with SEAL. In Scenario 1 (top), our **ego** agent avoids a merging **adversary** while the SEAL-only agent overreacts and crashes into a **background** vehicle. In Scenario 2 (bottom), our agent navigates a double left-turn cleanly, while the SEAL-only agent veers too wide and leaves the drivable surface.

To instantiate adversarial behaviors, we integrate our grounded objective into both CAT [210] and SEAL [156] pipelines by directly replacing their respective trajectory scoring objectives. Candidate selection is performed using the distance-based criterion in Equation (7.7), with $N_{\text{KNN}} = 8$ neighbors for SEAL and $N_{\text{KNN}} = 15$ for CAT. The larger neighborhood for CAT reduces over-reliance on local structure during candidate selection, since it cannot adapt during roll-out; in contrast, SEAL tolerates smaller neighborhoods due to its ability to adjust behavior online.

We perturb scenarios in an iterative manner, up to a maximum number of previous roll-outs $K = 5$, following prior work [156, 210]. We evaluate the quality of the generated adversarial trajectories $\widetilde{X}_{\text{adv}}^{(K)}$ based on their interaction characteristics, criticality with respect to the ego agent (i.e., induced lack of safety), as well as Wasserstein distance (WD)–based realism, following the protocol from SEAL [156] and detailed in Section 7.6.3.

7.6 Results

7.6.1 Closed-Loop Training Results

Table 7.2 shows our main quantitative results, highlighting overall task success rate improvements with RCG. Both SEAL-RCG and CAT-RCG substantially outperform their respective baselines, with an average success rate gain of 9.2%, confirming **RQ1**. Performance is strictly equal or better across all seven evaluation settings, and standard deviations are generally smaller, indicating more consistent behavior. For completeness, RCG-enhanced pipelines also outper-

Table 7.3: Average success rate (%), crash rate (%), and out-of-road (OoR) rate (%) of ego agents over all evaluation settings reported in Table 7.2.

Training Pipeline	**Success** (↑)	**Crash** (↓)	**OoR** (↓)
None (Replay)	57.4	42.3	00.3
No Adv	36.1	39.0	24.9
GOOSE	31.2	37.9	30.9
CAT	36.8	**28.5**	34.7
CAT-RCG	**40.4**	28.6	**31.0**
SEAL	39.8	31.3	28.9
SEAL-RCG	**42.8**	**30.7**	**26.5**

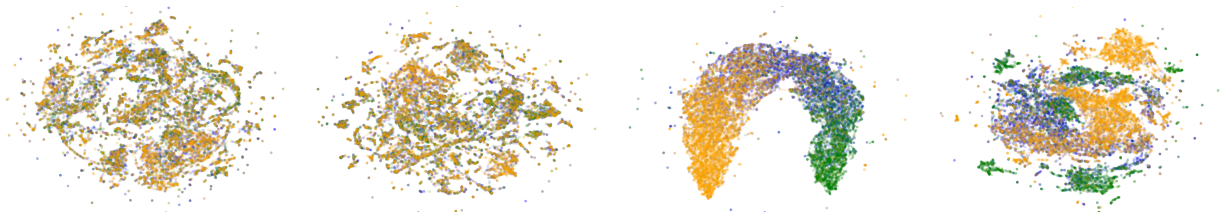
Table 7.4: Average performance (%) of ego agents across ablations. "CL Pre" and "TADS FT" are contrastive pre-training (Section 7.3.3) and fine-tuning (Section 7.3.4); "Dist." is the distance measure used in Equation (7.7) in Section 7.4. Selected adversary behaviors are instantiated with SEAL.

CL Pre	**TADS FT**	**Dist.**	**Success** (↑)	**Crash** (↓)	**OoR** (↓)
–	–	KNN	33.9	32.1	34.1
✓	–	KNN	38.4	30.7	30.9
–	✓	KNN	39.2	**30.6**	30.2
✓	✓	Proto	41.6	31.7	26.7
✓	✓	KNN	**42.8**	30.7	**26.5**

form agents trained with GOOSE [134]-generated scenarios, as well as those trained without perturbations ("No Adv").

Qualitatively, Figure 7.4 illustrates how ego agents trained with RCG adversaries outperform those trained without. In both scenarios, the baseline SEAL-trained policy overreacts to the adversary—swerving into background vehicles or leaving the drivable area—whereas the SEAL-RCG trained policy successfully navigates the challenging scenario. Table 7.3 provides a deeper breakdown by failure type; while RCG-based agents achieve similar crash rates to their respective baselines, they show substantial improvements in out-of-road rates, highlighting that prior approaches struggle to balance both failure modes.

To understand how various components of RCG contribute to final performance, we provide an ablation study in Table 7.4, breaking down the training pipeline along components introduced in Section 7.3 and Section 7.4. First, we ablate the contrastive pre-training described in Section 7.3.3, training the embedding model with either $\mathcal{L}_{\text{recon}}$ alone or the full loss from Equation (7.5). We then ablate the `TADS-traj` contrastive fine-tuning phase described in Section 7.3.4, either omitting or including it. Finally, we ablate the distance measure used for candidate scoring against the embedding cache, as formalized in Equation (7.7), by replacing the k-NearestNeighbor function with distance to the "unsafe" prototype center. As shown in the bottom row, all components are required for peak performance.



(a) No CL (b) `TADS-traj` FT (c) CL Pre (d) CL Pre + `TADS-traj` FT

Figure 7.5: UMAP [112] projections of the learned representation space, progressing through Section 7.3. Each point is an agent behavior, colored by safety-label "safe", "neutral", or "unsafe". Contrastive pre-training with large-scale data (CL, section 7.3.3 and fine-tuning with focused crash data in `TADS-traj` (FT, section 7.3.4) reveal increasingly structured clusters and sub-clusters, aligned with safety semantics.

Table 7.5: Quantitative analysis of embedding space structure. Clustering metrics (Silhouette score and Davies-Bouldin index) are computed over per-class KMeans sub-clustering (averaged over $N_{km} \in \{3, 4, 5, 6\}$), while linear probes assess classification accuracy. Results are averaged over three seeds.

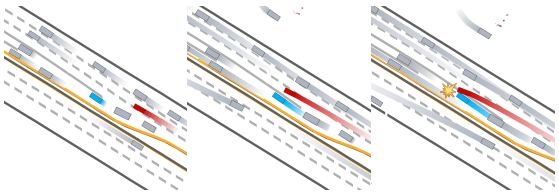
Method	**Silhouette ↑**	**Davies-Bouldin ↓**	**Linear Probe Acc. ↑**
No CL	0.177	1.768	44.1%
`TADS-traj` FT	0.171	1.734	43.6%
CL Pre	0.189	1.624	46.5%
CL Pre + `TADS-traj` FT	**0.198**	**1.278**	**46.8%**

7.6.2 Embedding Space Analysis

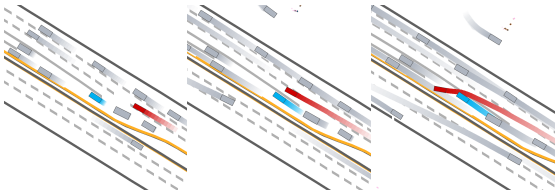
We analyze the learned behavior embedding space by evaluating both its qualitative structure and quantitative properties. All evaluations focus on the contrastive feature embeddings v_i, derived from the training pipeline described in Section 7.3 and projected from WOMD input scenarios.

Figure 7.5 illustrates how the representation evolves across training stages. Without contrastive learning (Figure 7.5a), the space lacks any clear organization with respect to safety. Applying prototypical contrastive regularization (Figure 7.5c) introduces coarse semantic structure aligned with safety labels, providing an initial organization where distance reflects behavioral risk. This is essential for downstream use, where candidate behaviors are scored by proximity to known unsafe examples. Fine-tuning on the `TADS-traj` dataset (Figure 7.5d) further refines this space, introducing sub-cluster structure that captures more *granular* distinctions within each safety class. This refinement arises because the critical trajectories from TADS introduce higher-severity interactions and contextually unsafe behavior, expanding the variation within each class in ways that the contrastive objective can retain. Conversely, fine-tuning without the initial PCL stage (Figure 7.5b) fails to meaningfully reshape the space, supporting the claim that our full pipeline is necessary.

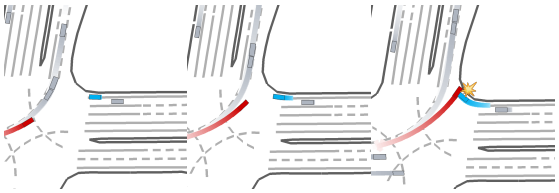
Quantitatively, we assess the structure of the embedding space using unsupervised clustering quality and supervised probe informativity. For clustering, we perform KMeans (with N_{km} clus-



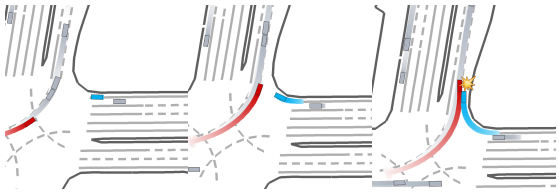
(a) SEAL perturbation, scenario 1



(b) **SEAL-RCG** perturbation, scenario 1



(c) SEAL perturbation, scenario 2



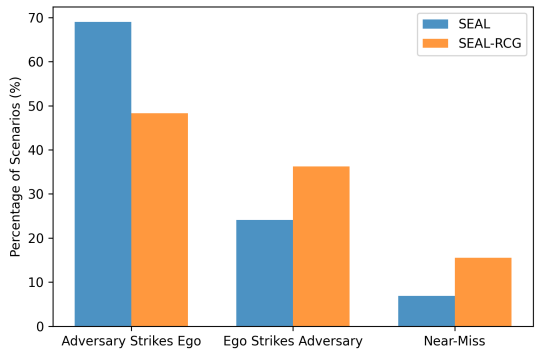
(d) **SEAL-RCG** perturbation, scenario 2

Figure 7.6: Qualitative examples of adversarial **perturbation** against an ego replay policy. In each row, the left shows perturbations from SEAL alone, and the right shows SEAL-RCG, applied to the same base scenario. In Scenario 1 (top), the **adversary** merges directly into the **ego** under SEAL, whereas our method produces a near-miss cut-in. In Scenario 2 (bottom), SEAL causes the ego to t-bone an adversary that drives unrealistically across the lane toward the road edge, whereas our method produces a more subtle, glancing collision as both agents turn into the same lane.

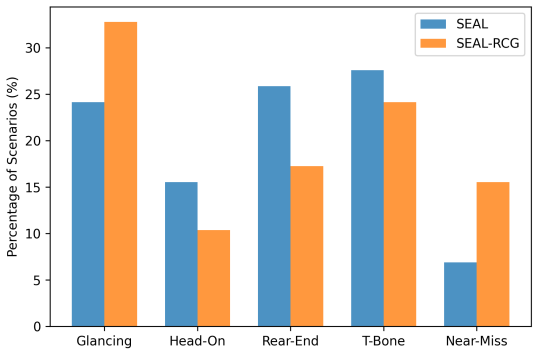
ters) within each safety class and report the average Silhouette score and Davies-Bouldin index on these sub-clusters, two standard metrics for capturing intra-class coherence and inter-class separation [28, 55]. For informativity, we train linear probes to predict safety labels on the same held-out validation set used in Section 7.3. Results are reported in Table 7.5; a higher Silhouette score and probe accuracy is better, while a lower Davies-Bouldin index is better. All results are averaged over three seeds, with clustering metrics further averaged over $N_{km} \in \{3, 4, 5, 6\}$. Overall, our full method improves across all metrics, confirming **RQ2** by demonstrating the stronger intrinsic capabilities of the learned representation.

7.6.3 Generated Scenario Results

We highlight representative qualitative examples of adversarial scenario generation in Figure 7.6, comparing SEAL alone to SEAL-RCG on matched base scenarios. Our method induces more nuanced behaviors, including near-misses and subtle collisions, in contrast to the overtly aggressive and often implausible maneuvers produced by SEAL. To compare broader behavioral characteristics, we analyze perturbation structure across a shared subset of base scenarios where both approaches produce critical or near-critical outcomes, as shown in Figure 7.7. Our approach increases the rate of near-misses and ego-initiated collisions, suggesting that RCG adversaries induce high-risk interactions that are more contingent on ego behavior, rather than forcing direct collisions. Geometrically, our method produces more glancing impacts and fewer t-bone or in-line collisions, better aligning with adversary maneuvers that *reduce* impact severity, as observed



(a) Collision Causality



(b) Collision Geometry

Figure 7.7: Taxonomizations of generated adversarial interactions across matched base scenarios, against an ego replay policy.

Table 7.6: Scenario generation criticality and realism results. **Lower** ego success is better. Wasserstein Distance (WD) on yaw and acceleration reflects alignment with real-world behavior. Averages are computed over all tested ego agents.

Eval. Setting	Ego Success (↓)	Yaw WD (↓)	Acc WD (↓)
Normal	55.5%	0.134	0.032
Hard	37.1%	0.122	0.050
GOOSE	42.1%	0.138	0.617
CAT	**32.1%**	0.136	0.291
CAT-RCG	39.2%	0.135	0.281
SEAL	33.2%	0.135	0.160
SEAL-RCG	39.1%	**0.135**	**0.117**

in real-world crash data such as `TADS-traj`.

Quantitatively, as shown in Table 7.6, baseline approaches like CAT and SEAL produce more extreme scenarios than RCG-based methods, as the former directly optimize for criticality. Still, as shown in Section 7.6.1, training ego policies on these highly critical examples does not necessarily lead to greater downstream ego performance. We additionally evaluate distributional realism via Wasserstein distances on yaw and acceleration, following SEAL [156]. Unlike SEAL, we do not penalize adversary out-of-road behavior, as such maneuvers are common in real-world crashes (e.g., in `TADS-traj`) and do not necessarily indicate implausibility in safety-critical settings. These distances, computed against ground truth X_{adv} trajectories, serve as a rough proxy for low-level behavioral realism. Under this measure, SEAL-RCG achieves the strongest alignment with real-world distributions, with CAT-RCG also showing modest gains. While these metrics capture distributional similarity, they overlook key aspects of interaction *quality*, further motivating the causal and geometric analyses above. Thus, taken together, these qualitative and quantitative results support **RQ3**.

7.7 Discussion

Scenario generation is essential for developing and validating autonomous driving systems, but existing approaches often produce interactions that are overly simplistic or behaviorally unrealistic. We introduced Real-world Crash Grounding (RCG), a framework for generating safety-critical scenarios by guiding adversarial perturbations using a behavior embedding grounded in real-world crash data. We integrated RCG into two prior generation pipelines, CAT and SEAL, replacing their handcrafted scoring mechanisms with our embedding-based selection criterion. Ego agents trained against RCG-perturbed scenarios achieved consistently higher success rates, with an average improvement of 9.2% across seven evaluation settings. Further analysis showed that the resulting scenarios elicited more causally and geometrically nuanced interactions, better reflecting real-world failure modes.

While our approach improves scenario quality and ego robustness, further enhancements remain possible. Scaling to additional crash video sources and leveraging advances in perception models could improve the quality and diversity of approximate trajectory annotations. Future work may also explore ego-side applications of the representation, such as using distance to "safe" embeddings to guide ego maneuver selection in closed-loop settings.

Chapter 8

Conclusion

In this thesis, we hypothesized that enhanced data utilization techniques were key components for increasing robustness in real-world autonomous driving and social robot navigation. We thus presented several solutions for identifying and addressing the weaknesses of naive data utilization. Our first pillar involved creating artificial distribution shifts by repartitioning datasets to evaluate and improve task performance and safety under real-world out-of-distribution conditions. These shifts were created on the bases of overall safety-relevance as well as disentangled ego and social contexts, with our remediation strategies recovering much of the lost performance. Our second pillar involved targeted scenario modifications to expose and address system weaknesses. We first developed methods to reduce the impact of perception errors in first-person view settings through re-simulation of originally perfect top-down annotations, along with learned correction modules. Then, we adversarially perturbed scenarios in increasingly human-like and real-world grounded ways, demonstrating that closed-loop training with these perturbations yields stronger performance in both normal and safety-critical settings. Through these results, we demonstrated that enhanced data utilization was highly beneficial in developing robust evaluation settings and policy methodologies.

While we discussed scoped limitations in each chapter, we nevertheless identify overarching limitations and future directions here. Our approach to robustness entails statistically increasing performance (or reducing performance degradation) in challenging settings, but makes no theoretical guarantees on abilities, a core challenge in data-based approaches. Such guarantees, e.g., through formal verification or reachability analyses, would be extremely helpful in applying and extending this thesis's work to industry settings. Another promising direction is unifying perception, prediction, and decision-making within a single learned world model. Advances in photorealistic scene reconstruction, such as Gaussian splatting or neural radiance fields, could be combined with our behavior-level realism methods explored in this thesis to enable coherent end-to-end training and evaluation.

Beyond robustness in autonomous navigation, our work on enhanced data utilization can also be extended to broader machine learning domains. In robotic manipulation, for example, zero-shot generalization continues to be a critical challenge, and could benefit from applying our factorized, compositional formulations and notions of long-tail evaluation. Similarly, in music transcription, where high-quality labeled data is scarce, existing data augmentation methods largely focus on audio-level realism (analogous to sensor realism in robotics). A focus on se-

mantically and musically meaningful perturbations (analogous to agent behavior perturbations explored in this thesis) could further improve real-world performance. This thesis represents initial steps towards improving the safety and robustness of autonomous robots, and we strongly encourage continued industry, academic, and government efforts to further develop these data utilization threads.

Bibliography

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 3.1.1, 3.5, 5, 5.1

[2] Francesco Amigoni, Matteo Luperto, and Viola Schiaffonati. Toward generalization of experimental results for autonomous robots. *Robotics and Autonomous Systems*, 90:4–14, 2017. 1, 2.3.2

[3] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 33:1462–1473, 2020. 4

[4] Stephen Balakirsky, Stefano Carpin, George Dimitoglou, and Benjamin Balaguer. From simulation to real robots with predictable results: Methods and examples. *Performance evaluation and benchmarking of intelligent systems*, pages 113–137, 2009. 1, 2.3.2

[5] Wentao Bao, Lichang Chen, Heng Huang, and Yu Kong. Prompting language-informed distribution for compositional zero-shot learning. In *European Conference on Computer Vision*, pages 107–123. Springer, 2024. 4.1.1

[6] Y Bengio. Learning deep architectures for AI, 2009. 1

[7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 4

[8] Gedas Bertasius, Aaron Chan, and Jianbo Shi. Egocentric basketball motion planning from a single first-person image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018. 5.1

[9] Alessia Bertugli, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. AC-VRNN: Attentive Conditional-VRNN for multi-future trajectory prediction. *Computer Vision and Image Understanding*, 2021. 2.2.1, 1, 5.1, 5.3, 5.5.1

[10] Manoj Bhat, Jonathan Francis, and Jean Oh. Trajformer: Trajectory prediction with local self-attentive contexts for autonomous driving. *arXiv preprint arXiv:2011.14910*, 2020. 3.1.1

[11] Huikun Bi, Ruisi Zhang, Tianlu Mao, Zhigang Deng, and Zhaoqi Wang. How can I see my future? FvTraj: Using first-person view for pedestrian trajectory prediction. In *European*

Conference on Computer Vision, 2020. 5, 5.1, 5.3.1

[12] Amitai Y Bin-Nun, Patricia Derler, Noushin Mehdipour, and Radboud Duintjer Tebbens. How should autonomous vehicles drive? Policy, methodological, and social considerations for designing a driver. *Humanities and social sciences communications*, 9(1):1–13, 2022. 1

[13] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections. *IEEE Intelligent Vehicles Symposium*, 2020. 2.1, 5.1

[14] Rodney A Brooks and Maja J Mataric. Real robots, real learning problems. In *Robot Learning*, pages 193–213. Springer, 1993. 1

[15] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2.1, 3.1.3, 5.1, 7.1.1

[16] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 2.2.2, 7.1.1

[17] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022. 6

[18] Yulong Cao, Danfei Xu, Xinshuo Weng, Zhuoqing Mao, Anima Anandkumar, Chaowei Xiao, and Marco Pavone. Robust trajectory prediction against adversarial attacks. In *Conference on Robot Learning*, pages 128–137. PMLR, 2023. 3

[19] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 624–641. Springer, 2020. 7.1.3

[20] Yachuang Chai, Jianwu Fang, Haoquan Liang, and Wushouer Silamu. TADS: A novel dataset for road traffic accident detection from a surveillance perspective. *The Journal of Supercomputing*, 80(18):26226–26249, 2024. (document), 7, 7.1.1, 7.1, 7.2

[21] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 3.1.3

[22] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. SAFE-SIM: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries, 2024. 6.1.1, 6.1.2, 7.1.2

[23] Samuel G Charlton, Nicola J Starkey, John A Perrone, and Robert B Isler. What's the

risk? A comparison of actual and perceived driving risk. *Transportation research part F: traffic psychology and Behaviour*, 25:50–64, 2014. 4

[24] Bingqing Chen, Jonathan Francis, Jean Oh, Eric Nyberg, and Sylvia L Herbert. Safe autonomous racing via approximate reachability on ego-vision. *arXiv preprint arXiv:2110.07699*, 2021. 6.1.2

[25] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[26] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint arXiv:2501.12375*, 2025. 7.2

[27] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3D multi-object tracking for autonomous driving. *arXiv, vol. abs/2001.05673*, 2020. 5.3.2

[28] Raymond Chua, Arna Ghosh, Christos Kaplanis, Blake Richards, and Doina Precup. Learning successor features the simple way. *Advances in Neural Information Processing Systems*, 37:49957–50030, 2024. 7.6.2

[29] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015. 5.1, 5.4.2, 5.3, 5.5.1

[30] Ítalo Renan da Costa Barros and Tiago Pereira Nascimento. Robotic mobile fulfillment systems: A survey on recent developments and research opportunities. *Robotics and Autonomous Systems*, 137:103729, 2021. 1

[31] Wei Dai, Shengen Wu, Wei Wu, Zhenhao Wang, Sisuo Lyu, Haicheng Liao, Limin Yu, Weiping Ding, Runwei Guan, and Yutao Yue. Large foundation models for trajectory prediction in autonomous driving: A comprehensive survey. *arXiv preprint arXiv:2509.10570*, 2025. 4.1.1

[32] Juan Pablo de Vicente and Alvaro Soto. DeepSocNav: Social navigation by imitating human behaviors. *RSS Workshop on Social Robot Navigation 2021*, 2021. 5, 5.1

[33] Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2250. IEEE, 2020. 3, 6.1.2

[34] Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—A methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6971–6988, 2023. 3, 6, 7.1.1, 7.1.2

[35] Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, and Marco Pavone. Realgen: Retrieval augmented generation for controllable traffic scenarios. In *European Conference on Computer Vision*, pages 93–110. Springer, 2024. 7.1.2

[36] Wenhao Ding, Sushant Veer, Karen Leung, Yulong Cao, and Marco Pavone. Surprise potential as a measure of interactivity in driving scenarios. *arXiv preprint arXiv:2502.05677*,

2025. 4.1.2

[37] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. (document), 2.1, 2.2.1, 3, 3.1.3, 3.2, 3.3, 3.5, 3.2, 3.3, 3.4, 3.5, 4, 4.1.2, 4.3.2, 4.3.3, 4.4, 6.3.3, 6.1, 7.1.1, 7.2, 7.5.1, 7.2, 7.5.2

[38] Jianwu Fang, Lei-lei Li, Junfei Zhou, Junbin Xiao, Hongkai Yu, Chen Lv, Jianru Xue, and Tat-Seng Chua. Abductive ego-view accident video understanding for safe driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22030–22040, 2024. 7.1.1

[39] Zimeng Fang, Chao Liang, Xue Zhou, Shuyuan Zhu, and Xi Li. Associate everything detected: Facilitating tracking-by-detection to the unknown. *arXiv preprint arXiv:2409.09293*, 2024. 7.2

[40] Lan Feng, Mohammadhossein Bahari, Kaouther Messaoud Ben Amor, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *European Conference on Computer Vision*, pages 106–123. Springer, 2024. 2.1, 4.1.2, 4.2, 4.3.3, 4.4, 7.5.2

[41] Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023. 1, 3, 6.1.1, 6.1.2

[42] Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020. 3, 3.1.2, 4.1.2, 6.1.2

[43] Jonathan Francis, Bingqing Chen, Weiran Yao, Eric Nyberg, and Jean Oh. Distribution-aware goal prediction and conformant model-based planning for safe autonomous driving. *arXiv preprint arXiv:2212.08729*, 2022. 6.1.2

[44] Jonathan Francis, Nariaki Kitamura, Felix Labelle, Xiaopeng Lu, Ingrid Navarro, and Jean Oh. Core challenges in embodied vision-language planning. *Journal of Artificial Intelligence Research*, 74:459–515, 2022. 3, 3.1.2

[45] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 3.1.1, 6.3.3

[46] Jingwei Ge, Jiawei Zhang, Cheng Chang, Yi Zhang, Danya Yao, and Li Li. Task-driven controllable scenario generation framework based on AOG. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):6186–6199, 2024. 7.1.2

[47] Christoph Glasmacher, Robert Krajewski, and Lutz Eckstein. An automated analysis framework for trajectory datasets. *arXiv preprint arXiv:2202.07438*, 2022. 3, 3.1.3, 3.2,

3.2, 3.3.1, 4.1.2

[48] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5.1

[49] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020. 1

[50] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 6.3.3, 7.5.3

[51] Aritra Guha, Rayleigh Lei, Jiacheng Zhu, XuanLong Nguyen, and Ding Zhao. Robust unsupervised learning of temporal dynamic vehicle-to-vehicle interactions. *Transportation research part C: emerging technologies*, 142:103768, 2022. 3.2

[52] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 7.1.3

[53] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 5, 5.3.3, 5.5.2

[54] Isabelle Guyon et al. A scaling law for the validation-set training-set size ratio. *AT&T Bell Laboratories*, 1(11), 1997. 1

[55] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering algorithms and validity measures. In *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001*, pages 3–22. IEEE, 2001. 7.6.2

[56] Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. King: Generating safety-critical driving scenarios for robust imitation via kine-

matics gradients. In *European Conference on Computer Vision*, pages 335–352. Springer, 2022. 3.1.2, 6, 6.1.1, 6.1.2, 6.4.2

[57] Ce Hao, Catherine Weaver, Chen Tang, Kenta Kawamoto, Masayoshi Tomizuka, and Wei Zhan. Skill-critic: Refining learned skills for hierarchical reinforcement learning. *IEEE Robotics and Automation Letters*, 2024. 7.1.3

[58] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.172. 7.3.4

[59] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *2022 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2022. 7.3.4

[60] Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. A comprehensive survey on contrastive learning. *Neurocomputing*, page 128645, 2024. 7.1.3

[61] Xuemin Hu, Shen Li, Tingyu Huang, Bo Tang, Rouxing Huai, and Long Chen. How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[62] Peide Huang, Xilun Zhang, Ziang Cao, Shiqi Liu, Mengdi Xu, Wenhao Ding, Jonathan Francis, Bingqing Chen, and Ding Zhao. What went wrong? closing the sim-to-real gap via differentiable causal discovery. In *Conference on Robot Learning*, pages 734–760. PMLR, 2023. 3, 3.1.2

[63] Peide Huang, Wenhao Ding, Ben Stoler, Jonathan Francis, Bingqing Chen, and Ding Zhao. CaDRE: Controllable and diverse generation of safety-critical driving scenarios using real-world trajectories. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, arXiv Preprint arXiv:2403.13208. IEEE, 2025. 6, 6.3.1, 7.1.1, 7.1.2

[64] Rui Huang, Henry Zheng, Yan Wang, Zhuofan Xia, Marco Pavone, and Gao Huang. Training an open-vocabulary monocular 3d detection model without 3d data. *Advances in Neural Information Processing Systems*, 37:72145–72169, 2024. 7.2

[65] Wenhui Huang, Haochen Liu, Zhiyu Huang, and Chen Lv. Safety-aware human-in-the-loop reinforcement learning with shared control for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 6.1.2

[66] WuLing Huang, Kunfeng Wang, Yisheng Lv, and FengHua Zhu. Autonomous vehicles testing methods review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 163–168. IEEE, 2016. 3

[67] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. Versatile scene-consistent traffic scenario generation as optimization with diffusion.

arXiv preprint arXiv:2404.02524, 2024. 6.3.3

[68] International Organization for Standardization. Road Vehicles—Safety of the Intended Functionality. `https://www.iso.org/obp/ui/#iso:std:77490:en`, 2022. ISO Standard 21448:2022. 6

[69] Bahar Irfan, James Kennedy, Séverin Lemaignan, Fotios Papadopoulos, Emmanuel Senft, and Tony Belpaeme. Social psychology and human-robot interaction: An uneasy marriage. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2018. 5

[70] Masha Itkina and Mykel Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning*, pages 606–617. PMLR, 2023. 3, 3.1.2, 6.1.2

[71] Boris Ivanovic, Guanyu Song, Igor Gilitschenski, and Marco Pavone. trajdata: A unified interface to multiple human trajectory datasets. *Advances in Neural Information Processing Systems*, 36:27582–27593, 2023. 2.1

[72] Dylan Jennings and Miguel Figliozzi. Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel. *Transportation Research Record*, 2673(6):317–326, 2019. 1

[73] Chenchen Jing, Yukun Li, Hao Chen, and Chunhua Shen. Retrieval-augmented primitive representations for compositional zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2652–2660, 2024. 4.1.1, 4.2

[74] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018. 5.1

[75] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016. 2.1

[76] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Alain Pagani, Didier Stricker, and Muhammad Zeshan Afzal. Learning attention propagation for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3828–3837, 2023. 4.1.1, 4.6

[77] Bing Cai Kok and Harold Soh. Trust in robots: Challenges and opportunities. *Current Robotics Reports*, 1(4):297–309, 2020. 1

[78] Stepan Konev, Kirill Brodt, and Artsiom Sanakoyeu. MotionCNN: A strong baseline for motion prediction in autonomous driving, 2022. 3.1.1, 3.4.2

[79] Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J Kochenderfer. Adaptive stress testing for autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2018. 6.1.2

[80] Parth Kothari and Alexandre Alahi. Safety-compliant generative adversarial networks for human trajectory forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4251–4261, 2023. 2.2.1, 3.7

[81] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds:

A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23 (7):7386–7400, 2021. 3.5

[82] Parth Kothari, Danya Li, Yuejiang Liu, and Alexandre Alahi. Motion style transfer: Modular low-rank adaptation for deep motion forecasting. In *Conference on Robot Learning*, pages 774–784. PMLR, 2023. 3.1.2

[83] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. IEEE, 2018. 7.1.1

[84] Alexei A. Efros Krishna Kumar Singh, Kayvon Fatahalian. KrishnaCam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *IEEE Winter Conference on Applications of Computer Vision*, 2016. 5, 5.1

[85] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017. 4

[86] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *2021 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2021. 7, 7.1.3, 7.3.3, 7.5.2

[87] Kaidong Li, Nina Y. Wang, Yiju Yang, and Guanghui Wang. SGNet: A super-class guided network for image classification and object detection. *Conference on Robots and Vision*, 2021. (document), 5.1, 5.4.3, 5.3, 5.5.1, 5.3, 5.4, 5.5.3

[88] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022. 6.3.3, 6.4.1, 7.5.1

[89] Quanyi Li, Zhenghao Mark Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36:3894–3920, 2023. 4.4, 7.5.2

[90] Rongchang Li, Zhenhua Feng, Tianyang Xu, Linze Li, Xiao-Jun Wu, Muhammad Awais, Sara Atito, and Josef Kittler. C2c: Component-to-composition learning for zero-shot compositional action recognition. In *European Conference on Computer Vision*, pages 369–388. Springer, 2024. 4.1.1

[91] Xuan Li, Siyu Teng, Bingzi Liu, Xingyuan Dai, Xiaoxiang Na, and Fei-Yue Wang. Advanced scenario generation for calibration and verification of autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 8(5):3211–3216, 2023. 7.1.2

[92] Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *2024 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2024. 7.5.2

[93] Junwei Liang, Lu Jiang, Kevin P. Murphy, Ting Yu, and Alexander G. Hauptmann. The

garden of forking paths: Towards multi-future trajectory prediction. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 5.1

[94] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–556. Springer, 2020. 3.1.1

[95] Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. Lora dropout as a sparsity regularizer for overfitting control. *arXiv preprint arXiv:2404.09610*, 2024. 7.3.4

[96] Chunyu Liu, Jianjun Yu, and Qiang Lin. LIDP: Contrastive learning of latent individual driving pattern for trajectory prediction. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1352–1357. IEEE, 2024. 7.1.3

[97] Henry X Liu and Shuo Feng. Curse of rarity for autonomous vehicles. *nature communications*, 15(1):4808, 2024. 1, 4.1.1, 6.1.1, 7.1.1

[98] Mingyu Liu, Ekim Yurtsever, Xingcheng Zhou, Jonathan Fossaert, Yuning Cui, Bare Luka Zagar, and Alois C Knoll. A survey on autonomous driving datasets: Data statistic, annotation, and outlook. *arXiv preprint arXiv:2401.01454*, 2024. 3.5

[99] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 7.2

[100] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems*, 32, 2019. 5, 5.1, 5.4.2, 5.3, 5.5.1, 5.5.2, 5.5.3

[101] Guannan Lou, Yao Deng, Xi Zheng, Mengshi Zhang, and Tianyi Zhang. Testing of autonomous driving systems: Where are we and where should we go? In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 31–43, 2022. 2.1

[102] Jack Lu, Kelvin Wong, Chris Zhang, Simon Suo, and Raquel Urtasun. SceneControl: Diffusion for controllable traffic scene generation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16908–16914. IEEE, 2024. 6.1.1

[103] Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7553–7560. IEEE, 2023. 6.1.2

[104] Georgina Lukanova and Galina Ilieva. Robots, artificial intelligence, and service automation in hotels. In *Robots, Artificial Intelligence, and Service Automation in Travel, Tourism and Hospitality*, pages 157–183. Emerald Publishing Limited, 2019. 1

[105] Weiyin Ma and J P Kruth. NURBS curve and surface fitting for reverse engineering. *The International Journal of Advanced Manufacturing Technology*, 14:918–927, 1998. 6.4.2

[106] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: A survey. *arXiv preprint arXiv:2202.02980*, 2022. 5.3.2

[107] Carl Macrae. Learning from the failure of autonomous and intelligent systems: Accidents, safety, and sociotechnical sources of risk. *Risk analysis*, 42(9):1999–2025, 2022. 1

[108] Osama Makansi, Özgün Çiçek, Yassine Marrakchi, and Thomas Brox. On exposing the challenging long tail in future prediction of traffic actors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13147–13157, 2021. 7.3.3

[109] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5222–5230, 2021. 4, 4.1.1

[110] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. *IEEE/CVF International Conference on Computer Vision*, 2020. 5.1

[111] Christoforos I. Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 2023. 1, 1, 5.5.2

[112] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. (document), 4.2, 4.3.2, 7.5

[113] Goeffrey J McLachlan. Mahalanobis distance. *Resonance*, 1999. 5.3.2

[114] Nathan Medeiros-Ward, Joel M Cooper, and David L Strayer. Hierarchical control and driving. *Journal of experimental psychology: General*, 143(3):953, 2014. 4, 6, 6.3.2

[115] Tobias Moers, Lennart Vater, Robert Krajewski, Julian Bock, Adrian Zlocki, and Lutz Eckstein. The exiD dataset: A real-world trajectory dataset of highly interactive highway scenarios in Germany. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 958–964. IEEE, 2022. 3, 3.2, 4.1.2

[116] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36, 2024. 2.2.2, 6.4.3

[117] Alessio Monti, Alessia Bertugli, Simone Calderara, and Rita Cucchiara. DAG-net: Double attentive graph neural network for trajectory forecasting. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 2551–2558. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9412114. 3.1.1, 3.5, 3.2

[118] Kensuke Nakamura, Ran Tian, and Andrea Bajcsy. A general calibrated regret metric for detecting and mitigating human-robot interaction failures. *CoRR*, 2024. 4.1.2

[119] Nihal V Nayak, Peilin Yu, and Stephen H Bach. Learning to compose soft prompts for compositional zero-shot learning. In *International Conference on Learning Representa-*

tions, 2023. 4.1.1

[120] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2021. 3.1.1

[121] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3D object detection? In *IEEE/CVF International Conference on Computer Vision*, 2021. 5.3.2

[122] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric future localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 5.1

[123] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 282–298. Springer, 2020. 3.1.1, 6.1.2

[124] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009. 2.1, 2.2.1, 5, 5.1

[125] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 7.2

[126] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024. 7.2

[127] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4051–4070, 2022. 4, 4.1.1

[128] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602, 2019. 4, 4.1.1, 4.3.4

[129] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 5, 5.1

[130] Kyle K. Qin, Yongli Ren, Wei Shao, Brennan Lake, Filippo Privitera, and Flora D. Salim. Multiple-level point embedding for solving human trajectory imputation with prediction.

ACM Trans. Spatial Algorithms Syst., 2023. 5.1

[131] Zhizhen Qin, Tsui-Wei Weng, and Sicun Gao. Quantifying safety of learning-based self-driving control using almost-barrier functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12903–12910. IEEE, 2022. 6.1.2

[132] Jianing Qiu, Lipeng Chen, Xiao Gu, Frank P-W Lo, Ya-Yen Tsai, Jiankai Sun, Jiaqi Liu, and Benny Lo. Egocentric human trajectory forecasting with a wearable camera and multi-modal fusion. *IEEE Robotics and Automation Letters*, 2022. 5, 5.1

[133] Krishan Rana, Ming Xu, Brendan Tidd, Michael Milford, and Niko Sünderhauf. Residual skill policies: Learning an adaptable skill-based action space for reinforcement learning for robotics. In *Conference on Robot Learning*, pages 2095–2104. PMLR, 2023. 6.3.2, 6.3.3, 6.4.1, 6.5, 7.1.3, 7.5.1

[134] Joshua Ransiek, Johannes Plaum, Jacob Langner, and Eric Sax. GOOSE: Goal-conditioned reinforcement learning for safety-critical scenario generation. In *2024 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2024. 6.1, 6.1.1, 6.1.2, 6.3.2, 6.4.2, 6.3b, 6.4b, 7.1.2, 7.5.1, 7.6.1

[135] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 7.2

[136] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17305–17315, 2022. 6, 6.1.1, 6.1.2, 6.4.3, 7.1.2

[137] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016. 2.1, 5, 5.1

[138] Rebecca Roelofs, Liting Sun, Ben Caine, Khaled S Refaat, Ben Sapp, Scott Ettinger, and Wei Chai. CausalAgents: A robustness benchmark for motion forecasting using causal relationships. *arXiv preprint arXiv:2207.03586*, 2022. 4.1.2

[139] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 3.1.1

[140] Saeed Saadatnejad, Mohammadhossein Bahari, Pedram Khorsandi, Mohammad Saneian, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Are socially-aware trajectory prediction models really socially-aware? *Transportation research part C: emerging technologies*, 141:103705, 2022. 3.1.2

[141] Abbas Sadat, Sean Segal, Sergio Casas, James Tu, Bin Yang, Raquel Urtasun, and Ersin Yumer. Diverse complexity measures for dataset curation in self-driving. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8609–8616. IEEE, 2021. 3, 3.1.3, 3.2

[142] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi,

and Silvio Savarese. SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1349–1358. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00144. 3.1.1

[143] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *European Conference on Computer Vision*, 2020. 5.1

[144] John M Scanlon, Kristofer D Kusano, Tom Daniel, Christopher Alderson, Alexander Ogle, and Trent Victor. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accident Analysis & Prevention*, 163:106454, 2021. 7.1.2

[145] Julian Schmidt, Julian Jordan, David Raba, Tobias Welz, and Klaus Dietmayer. MEAT: Maneuver extraction from agent trajectories. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1810–1816. IEEE, 2022. 3.2

[146] Ankit Parag Shah, Jean-Bapstite Lamare, Tuan Nguyen-Anh, and Alexander Hauptmann. CADP: A novel dataset for CCTV traffic camera based accident analysis. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–9. IEEE, 2018. 7.1.1

[147] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748. IEEE, 2020. 4.3.2

[148] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017. 3

[149] Jiajun Shen and Guangchuan Yang. Crash risk assessment for heterogeneity traffic and different vehicle-following patterns using microscopic traffic flow data. *Sustainability*, 12 (23):9888, 2020. 3.2

[150] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35:6531–6543, 2022. (document), 2.2.1, 3.1.1, 3.4.2, 3.5, 3.4, 3.2, 3.4, 3.5, 4, 4.4, 7.1.3, 7.3.1, 7.5.2

[151] Qunying Song, Emelie Engström, and Per Runeson. Industry practices for challenging autonomous driving systems with critical scenarios. *ACM Transactions on Software Engineering and Methodology*, 33(4):1–35, 2024. 6

[152] Neville A Stanton and Paul M Salmon. Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems. *Safety Science*, 47(2):227–237, 2009. 4

[153] Benjamin Stoler, Meghdeep Jana, Soonmin Hwang, and Jean Oh. T2FPV: Dataset and method for correcting first-person view errors in pedestrian trajectory prediction. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4037–4044. IEEE, 2023. 1.2, 5

[154] Benjamin Stoler, Ingrid Navarro, Meghdeep Jana, Soonmin Hwang, Jonathan Francis, and Jean Oh. SafeShift: Safety-informed distribution shifts for robust trajectory prediction in autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1186. IEEE, 2024. 1, 1.2, 3, 4.1.2, 6.1.1, 6.1.2, 6.3.1, 7.1.1

[155] Benjamin Stoler, Jonathan Francis, and Jean Oh. Longcomp: Long-tail compositional zero-shot generalization for robust trajectory prediction. *arXiv preprint arXiv:2511.10411*, 2025. 1.2, 4

[156] Benjamin Stoler, Ingrid Navarro, Jonathan Francis, and Jean Oh. Seal: Towards safe autonomous driving via skill-enabled adversary learning for closed-loop scenario generation. *IEEE Robotics and Automation Letters*, 10(9):9320–9327, 2025. 1.2, 4.1.2, 6, 7.1.2, 7.4, 7.5.1, 7.5.3, 7.6.3

[157] Benjamin Stoler, Juliet Yang, Jonathan Francis, and Jean Oh. Rcg: Safety-critical scenario generation for robust autonomous driving via real-world crash grounding. *arXiv preprint arXiv:2507.10749*, 2025. 1.2, 7

[158] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 5.1, 7.1.1

[159] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409, 2021. 6.1.1

[160] Simon Suo, Kelvin Wong, Justin Xu, James Tu, Alexander Cui, Sergio Casas, and Raquel Urtasun. Mixsim: A hierarchical framework for mixed reality traffic simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9622–9631, 2023. 2.2.2, 3, 6.1.2, 6.4.3, 7.1.2

[161] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. In *7th Annual Conference on Robot Learning*, 2023. 4.1.2

[162] Xiaocheng Tang, Soheil Sadeghi Eshkevari, Haoyu Chen, Weidan Wu, Wei Qian, and Xiaoming Wang. Golfer: Trajectory prediction with masked goal conditioning MnM network. *arXiv preprint arXiv:2207.00738*, 2022. 3.1.1, 3.4.2

[163] Hanlin Tian, Kethan Reddy, Yuxiang Feng, Mohammed Quddus, Yiannis Demiris, and Panagiotis Angeloudis. Enhancing autonomous vehicle training with language model integration and critical scenario generation. *arXiv preprint arXiv:2404.08570*, 2024. 6

[164] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. *arXiv preprint arXiv:2407.00959*, 2024. 7.1.3

[165] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 6.3.3

[166] Nathan Tsoi, Mohamed Hussein, Olivia Fugikawa, J. D. Zhao, and Marynel Vázquez. An

approach to deploy interactive robotic simulators on the web for HRI experiments: Results in social robot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. 5, 5.1, 5.3.1

[167] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 7.1.3

[168] Tessa van der Heiden, Naveen Shankar Nagaraja, Christian Weiss, and Efstratios Gavves. SafeCritic: Collision-aware trajectory prediction. *arXiv preprint arXiv:1910.06673*, 2019. 7.1.3

[169] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008. (document), 6.2

[170] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 7.1.3

[171] Katja Vogel. A comparison of headway and time to collision as safety indicators. *Accident Analysis & Prevention*, 35(3):427–433, 2003. ISSN 0001-4575. doi: 10.1016/S0001-4575(02)00022-2. 3.2

[172] Kevin L Voogd, Jean Pierre Allamaa, Javier Alonso-Mora, and Tong Duy Son. Reinforcement learning from simulation to real world autonomous driving using digital twin. *IFAC-PapersOnLine*, 56(2):1510–1515, 2023. 6.1.2

[173] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021. 6.1.1, 6.1.2, 6.4.2

[174] Qingfan Wang, Dongyang Xu, Gaoyuan Kuang, Chen Lv, Shengbo Eben Li, and Bingbing Nie. Risk-aware vehicle trajectory prediction under safety-critical scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2025. 7.1.3

[175] Qingsheng Wang, Lingqiao Liu, Chenchen Jing, Hao Chen, Guoqiang Liang, Peng Wang, and Chunhua Shen. Learning conditional attributes for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2023. 4.1.1

[176] Tsun-Hsuan Wang, Alaa Maalouf, Wei Xiao, Yutong Ban, Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. Drive anywhere: Generalizable end-to-end autonomous driving with multi-modal foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6687–6694. IEEE, 2024. 4.1.2

[177] Xin Wang, Hong Chen, Zihao Wu, Wenwu Zhu, et al. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 4

[178] Yuning Wang, Pu Zhang, Lei Bai, and Jianru Xue. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1400–1409, 2023. 7.1.3, 7.3.3

[179] Yuning Wang, Zeyu Han, Yining Xing, Shaobing Xu, and Jianqiang Wang. A survey on datasets for the decision making of autonomous vehicles. *IEEE Intelligent Transportation Systems Magazine*, 2024. 3.5

[180] Nick Webb, Dan Smith, Christopher Ludwick, Trent Victor, Qi Hommes, Francesca Favaro, George Ivanov, and Tom Daniel. Waymo's safety methodologies and safety readiness determinations. *arXiv preprint arXiv:2011.00054*, 2020. 3

[181] Hendrik Weber, Julian Bock, Jens Klimke, Christian Rösener, Johannes Hiller, Robert Krajewski, Adrian Zlocki, and Lutz Eckstein. A framework for definition of logical scenarios for safety assurance of automated driving. *Traffic Injury Prevention*, 20:S65–S70, June 2019. doi: 10.1080/15389588.2019.1630827. 3.1.3

[182] Erica Weng, Hana Hoshino, Deva Ramanan, and Kris Kitani. Joint metrics matter: A better standard for trajectory forecasting. *CoRR*, abs/2305.06292, 2023. doi: 10.48550/arXiv.2305.06292. 3.1.2

[183] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D multi-object tracking: A baseline and new evaluation metrics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020. 5.3.4

[184] Xinshuo Weng, Boris Ivanovic, Kris Kitani, and Marco Pavone. Whose track is it anyway? Improving robustness to tracking errors with affinity-based trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5, 5.1, 5.3.3, 5.5.1, 5.6

[185] Xinshuo Weng, Boris Ivanovic, and Marco Pavone. MTP: Multi-hypothesis tracking and prediction for reduced error propagation. In *IEEE Intelligent Vehicles Symposium*, 2022. 5.3.3

[186] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*, pages 987–993. IEEE, 2010. 3.1.2, 3.3.2

[187] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 2.2.1, 4, 4.1.2, 4.4, 7.1.1

[188] Conghao Wong, Beihao Xia, Ziming Hong, Qinmu Peng, and Xinge You. View Vertically: A hierarchical network for trajectory prediction via fourier spectrums. In *European Conference on Computer Vision*, 2022. 5.1

[189] Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. *Advances in Neural Information Processing Systems*, 37:114048–114071, 2024. 7.1.3

[190] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016. 4.3.2

[191] Chejian Xu, Ding Zhao, Alberto Sangiovanni-Vincentelli, and Bo Li. Diffscene:

Diffusion-based safety-critical scenario generation for autonomous vehicles. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. 6, 6.1.1, 6.1.2, 6.4.2, 7.1.2

[192] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. BITS: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2929–2936. IEEE, 2023. 3, 3.1.2, 6.1.1, 7.1.2

[193] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9878–9888, 2021. 7, 7.1.1

[194] Zhigang Xu, Kaifan Zhang, Haigen Min, Zhen Wang, Xiangmo Zhao, and Peng Liu. What drives people to accept automated vehicles? Findings from a field experiment. *Transportation research part C: emerging technologies*, 95:320–334, 2018. 1

[195] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 5, 5.1

[196] Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X Liu. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14 (1):2037, 2023. 7.1.1

[197] Xuemeng Yang, Licheng Wen, Yukai Ma, Jianbiao Mei, Xin Li, Tiantian Wei, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, et al. DriveArena: A closed-loop generative simulation platform for autonomous driving. *arXiv preprint arXiv:2408.00415*, 2024. 6

[198] Yi Yang, Qingwen Zhang, Kei Ikemura, Nazre Batool, and John Folkesson. Hard cases detection in motion prediction by vision-language foundation models. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2405–2412. IEEE, 2024. 4.1.2

[199] Gefan Ye, Lin Li, Kexin Li, Jun Xiao, et al. Zero-shot compositional action recognition with neural logic constraints. *arXiv preprint arXiv:2508.02320*, 2025. 4.1.1

[200] Luyao Ye, Zikang Zhou, and Jianping Wang. Improving the generalizability of trajectory prediction models with frenet-based domain normalization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11562–11568. IEEE, 2023. 3, 3.1.2, 3.2, 3.4.1, 3.5, 3.2, 3.3, 3.6.1, 3.6.2, 4.1.2, 6.1.2

[201] Luyao Ye, Zikang Zhou, Jianping Wang, Yung-Hui Li, Nien-Yi Jan, Yi-Rong Lin, and Yen-Cheng Lin. IMGTP: A unified framework for improving and measuring the generalizability of trajectory prediction models. *IEEE Transactions on Intelligent Vehicles*, 2024. 4, 4.1.2, 4.3.1

[202] Rui Yu and Zihan Zhou. Towards robust human trajectory prediction in raw videos. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. 5, 5.1, 5.3.3, 5.3, 5.5.1, 5.6

[203] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. *CoRR*, abs/2103.14023, 2021. 3.1.1

[204] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *European Conference on Computer Vision*, 2022. 5, 5.1

[205] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access : practical innovations, open solutions*, 8:58443–58469, 2020. 1

[206] Wei Zhan, Liting Sun, Di Wang, Yinghan Jin, and Masayoshi Tomizuka. Constructing a highly interactive vehicle motion dataset. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6415–6420. IEEE, 2019. 4.3.1

[207] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019. 3.2

[208] Chris Zhang, Sourav Biswas, Kelvin Wong, Kion Fallah, Lunjun Zhang, Dian Chen, Sergio Casas, and Raquel Urtasun. Learning to drive via asymmetric self-play. In *European Conference on Computer Vision*, pages 149–168. Springer, 2024. 6.1.1

[209] Junrui Zhang, Mozhgan Pourkeshavarz, and Amir Rasouli. Tract: A training dynamics aware contrastive learning framework for long-tail trajectory prediction. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3282–3288. IEEE, 2024. 7.1.3, 7.3.3

[210] Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning*, pages 2357–2372. PMLR, 2023. 2.2.2, 4.1.2, 6, 6.1, 6, 6.1.1, 6.1.2, 6.3, 6.3.1, 6.3.2, 6.3.3, 6.4.1, 6.4.2, 6.4.3, 6.3c, 6.4c, 7, 7.1.2, 7.4, 7.4, 7.5.1, 7.5.3

[211] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15159–15168, 2022. 3.1.2

[212] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, José Manuel Gaspar Sánchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, et al. Finding critical scenarios for automated driving systems: A systematic mapping study. *IEEE Transactions on Software Engineering*, 49(3):991–1026, 2022. 3.1.3, 6

[213] Xinyu Zhang, Zhiwei Li, Yan Gong, Dafeng Jin, Jun Li, Li Wang, Yanzhang Zhu, and Huaping Liu. OpenMPD: An open multimodal perception dataset for autonomous driving. *IEEE Transactions on Vehicular Technology*, 71(3):2437–2447, 2022. 7.1.1

[214] Zhejun Zhang, Alexander Liniger, Christos Sakaridis, Fisher Yu, and Luc V Gool. Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding. *Advances in Neural Information Processing Systems*, 36:57481–57499, 2023. 7.1.3

[215] Cong Zhao, Andi Song, Yuchuan Du, and Biao Yang. TrajGAT: A map-embedded graph attention network for real-time vehicle trajectory imputation of roadside perception.

Transportation research part C: emerging technologies, 142:103787, 2022. 5, 5.1

[216] Xiaoji Zheng, Lixiu Wu, Zhijie Yan, Yuanrong Tang, Hao Zhao, Chen Zhong, Bokui Chen, and Jiangtao Gong. Large language models powered context-aware motion prediction in autonomous driving. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 980–985. IEEE, 2024. 4.1.2

[217] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, pages 144–177. PMLR, 2023. 2.2.2, 6.1.1, 6.4.3

[218] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566. IEEE, 2023. 6.1.1

[219] Xiaoling Zhou, Ou Wu, Weiyao Zhu, and Ziyang Liang. Understanding difficulty-based sample weighting with a universal difficulty measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 68–84. Springer, 2022. 3.4.2

[220] Bing Zhu, Peixing Zhang, Jian Zhao, and Weiwen Deng. Hazardous scenario enhanced generation for automated vehicle testing based on optimization searching method. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7321–7331, 2021. 7.1.2