

RCG: Safety-Critical Scenario Generation for Robust Autonomous Driving via Real-World Crash Grounding

Benjamin Stoler, Juliet Yang, Jonathan Francis, and Jean Oh

Abstract—Safety-critical scenarios are essential for training and evaluating autonomous driving (AD) systems, yet remain extremely rare in real-world driving datasets. To address this, we propose Real-world Crash Grounding (RCG), a scenario generation framework that integrates crash-informed semantics into adversarial perturbation pipelines. We construct a safety-aware behavior representation through contrastive pre-training on large-scale driving logs, followed by fine-tuning on a small, crash-rich dataset with approximate trajectory annotations extracted from video. This embedding captures semantic structure aligned with real-world accident behaviors and supports selection of adversary trajectories that are both high-risk and behaviorally realistic. We incorporate the resulting selection mechanism into two prior scenario generation pipelines, replacing their handcrafted scoring objectives with an embedding-based criterion. Experimental results show that ego agents trained against these generated scenarios achieve consistently higher downstream success rates, with an average improvement of 9.2% across seven evaluation settings. Qualitative and quantitative analyses further demonstrate that our approach produces more plausible and nuanced adversary behaviors, enabling more effective and realistic stress testing of AD systems. Code and tools will be released publicly.

Index Terms—Autonomous driving, scenario generation, contrastive learning, safety-critical robustness.

I. INTRODUCTION

TO develop and assess autonomous driving (AD) models, a diverse and representative set of traffic scenarios must be used to ensure that such agents behave safely in real-world cases, especially in long-tail, safety-critical scenarios [1]–[3]. To reduce reliance on costly and potentially risky real-world driving testing, researchers and industry practitioners lean on large datasets of collected driving logs [4], [5]. These datasets typically provide agent-centric, trajectory-level motion annotations, enabling supervised learning of motion forecasting and simulation-based driving tasks [6]–[8]. However, safety-critical scenarios, such as actual accident and near-miss cases, are exceedingly scarce in these standard motion datasets [9]–[11]. To address this phenomenon known as “the curse of rarity,” much recent work has turned to adversarial scenario generation to produce these rare events synthetically [12]–[14]. Modifying (or “perturbing”) these benign scenarios is appealing because it retains realistic context from real-world scenes while allowing adversaries to be optimized to trigger failure in fine-grained ways.

To adversarially perturb a scenario with respect to a driving model under development or test (i.e., the “ego” agent

controlled by that model), it would be ideal to be able to *directly* sample appropriate, context-dependent adversary behaviors that sufficiently challenge the ego agent’s decision making. However, due to the curse of rarity, prior art must instead cope in alternative ways. In particular, recent state-of-the-art (SOTA) works such as CAT [12] and SEAL [14] *reason* over candidate adversary behaviors, produced by a pre-trained trajectory generator, 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 too far from being realistic.

To generate realistic safety-critical scenarios, we propose Real-world Crash Grounding (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 [15], [16]. 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) [17] 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

BS, JY, JF, and JO are with the School of Computer Science at Carnegie Mellon University. Emails: {bstoler, jzyang, jmf1, jeanoh}@cs.cmu.edu. JF is also with the Bosch Center for AI.

Manuscript received May xx, 2025.

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.
- 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.

II. RELATED WORK

A. 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 [6]–[8]. 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 [18]–[20], while prediction and planning tasks utilize agent trajectories and map features, often in polyline form [6], [8], [21]. 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 [9]–[11]. Efforts like SafeShift [2] 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 [22] focus on reasoning and forecasting tasks, while others like TADS [15], SUTD [16], and CADP [23] 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., [24], 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 [15] 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.

B. 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 [25]–[27], 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 [11], [14]. Some approaches manually construct starting scenarios based on real-world accident patterns [28]–[31], 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 [25], [32]–[34]. More efficient alternatives employ quality-diversity optimization [10] or adversarial reinforcement learning [13], but depend on hand-crafted metric functions that are difficult to generalize. Similarly, explicitly closed-loop methods such as CAT [12] and SEAL [14] 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.

C. 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 [35]–[38]. These methods often aim to build expressive embedding spaces using high-capacity models such as Transformers [39] or large-language-inspired tokenization schemes [40]. Recent work has also explored the use of skill-space priors to enable more human-like, hierarchical behavior generation [41], [42]. Other approaches attempt to embed safety awareness directly, either by incorporating risk-aware architectural components [43] or by penalizing predicted collisions across trajectory samples [44]. 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 [45], [46]. In autonomous driving, recent works, such as FEND [47] and TRACT [48], apply prototypical contrastive learning (PCL) [17] to enforce structure based on trajectory types and training-stage dynamics. Other approaches, like LIDP [49], rely on traditional contrastive objectives such as the InfoNCE loss [50], 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.

III. PRELIMINARIES

In closed-loop scenario generation, scenarios are created adaptively, based on how a learning agent interacts with an

environment. We define notation here for the purposes of *perturbation-based* safety-critical scenario generation, adapting definitions from SEAL [14]. In this setting, a base scenario from a trajectory-based, real-world dataset is selected and modified according to some desiderata pertaining to the learning agent, so as to increase the criticality of the ensuing interactions. The agent is then rolled out in the modified scenario, receiving some training impetus, and the cycle (i.e., “loop”) repeats. A base scenario is defined as $S = (\mathbf{X}, \mathbf{M}, \text{ego}, \text{adv})$, where:

i) $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$ denotes the trajectories of all traffic participants (i.e., “agents”) 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.

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) *ego* and *adv* specify the agent IDs corresponding to the ego-controlled learning agent and the agent to be adversarially modified, respectively.

Now, let \mathcal{P} denote a *perturbation function*, which modifies a base scenario S by altering the behavior of agent *adv*, possibly conditioned on historical cues. In a closed-loop training or evaluation cycle, the adversary is permitted to observe all trajectories from up to K past roll-outs on the base scenario, denoted $\{\hat{\mathbf{X}}^{(k)}\}_{k=1}^{\leq K}$, each corresponding to a prior episode with a different perturbation of S . These roll-outs are themselves the result of rolling out the ego agent in response to prior adversarial perturbations, forming a sequential interaction history. Concretely, \mathcal{P} is defined as follows:

$$\mathcal{P} : \left(S, \left\{ \hat{\mathbf{X}}^{(k)} \right\}_{k=1}^{\leq K} \right) \rightarrow \mathcal{B}_{\text{adv}}$$

where the planned adversarial behavior functional \mathcal{B}_{adv} may take multiple forms, ranging from a fixed trajectory to follow, to a reactive policy capable of adapting to ego behavior while pursuing higher-level objectives. Importantly, this perturbation process may condition on the original scenario, S , as well as past episode roll-outs, but must not access future planned ego trajectories for the upcoming episode.

IV. 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 [15]. 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) [6], 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 [51] for object detection and SAM2 [52] for instance segmentation. Then, AED [53] is used for 2D object tracking, while we utilize XFeat [54] features to further refine temporal associations through keypoint correspondence. Next, metric depth is estimated with UniDepth [55] and temporally smoothed via VideoDepthAnything [56], allowing us to follow OVM3D-Det [57] 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 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 I. We highlight representative examples of the scenarios in Figure 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*.

V. 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



Fig. 1: Illustrative examples from TADS [15], processed into TADS-trajectory. 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.

TABLE I: 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

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 2.

A. 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) [37] 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.

Hence, given a scenario $S = (\mathbf{X}, \mathbf{M}, \text{ego}, \text{adv})$ and 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 $\tilde{\mathbf{X}}$. The framework produces an overall reconstruction loss, $\mathcal{L}_{\text{recon}}$, comprising trajectory regression and mode selection losses.

B. Safety Classification

To utilize observed behaviors \mathbf{X} in a safety-informed manner, we first derive large-scale safety labels leveraging the SafeShift [2] scenario characterization framework. 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, $d_i = \text{GT}_i - \text{FE}_i$, is then discretized into three classes—“safe”, “neutral”, and “unsafe”—according to Equation (1), where the thresholding value δ is selected to ensure roughly even class distributions across the dataset:

$$y_i = \begin{cases} \text{safe}, & \text{if } d_i < -\delta \\ \text{neutral}, & \text{if } |d_i| \leq \delta \\ \text{unsafe}, & \text{if } d_i > \delta \end{cases} \quad (1)$$

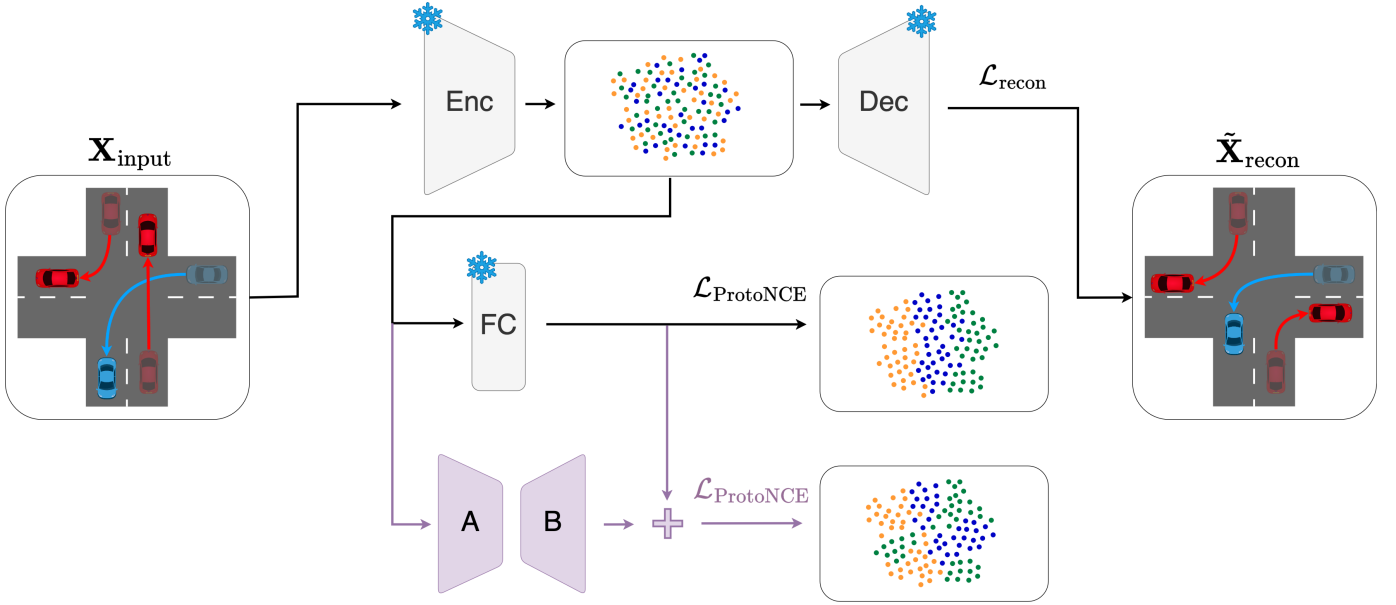


Fig. 2: Embedding space training overview, as described in Section V. Initial pre-training is performed with reconstruction and contrastive loss objectives. Existing weights are then frozen, and adapters are trained on TADS-*traj* using contrastive loss alone (shown in purple). Contrastive regularization is supervised via “safe”, “neutral”, and “unsafe” labels.

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.

C. 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 [47], [58]. 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 [17], and applied in TrACT [48] and FEND [47], 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 :

$$\mathcal{L}_{\text{inst}} = - \sum_{i=1}^B \frac{1}{N_{\text{po},i}} \sum_{i^+=1}^{N_{\text{po},i}} \log \frac{\exp(v_i \cdot v_{i^+}^+ / \tau)}{\sum_{j=1}^B \exp(v_i \cdot v_j / \tau)}, \quad (2)$$

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:

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

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:

$$\phi_j = \frac{\sum_{i=1}^{N_{\text{class},j}} \|v_i - c_j\|_2}{N_{\text{class},j} \log(N_{\text{class},j} + \alpha)} \quad (4)$$

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 following multi-objective loss:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{ProtoNCE}}, \quad (5)$$

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

D. Fine-Tuning Representations

We now aim to calibrate our learned behavior representation by fine-tuning using the small but highly-informative dataset

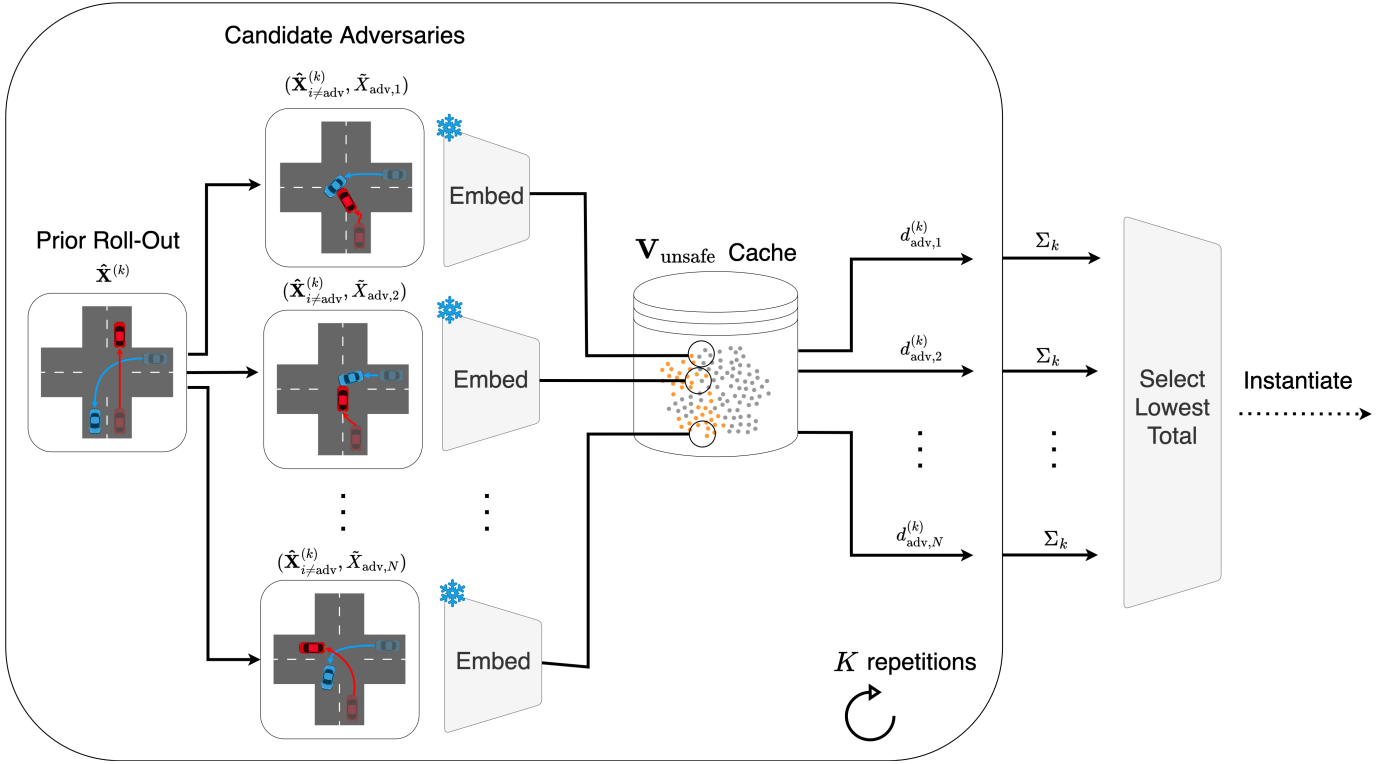


Fig. 3: RCG adversarial scenario selection, as described in Section VI. Candidate behaviors supplant observed behaviors in K prior roll-outs, then are projected into the \mathbf{V} -space and ranked by k -NearestNeighbor distance to “unsafe” embeddings.

processed in Section IV, 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 [59], [60]. Indeed, empirically we observe that full fine-tuning leads to rapid memorization and degradation of generalization performance. To mitigate this, we employ a LoRA [61] 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:

$$v_i = \text{normalize}(\text{FC}_{\text{proj}}(z_i) + B A z_i) \quad (6)$$

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 pre-trained backbone.

VI. REAL-WORLD CRASH-GROUNDED ADVERSARIES

To generate realistic yet critical driving scenarios, we propose **Real-world Crash Grounding (RCG)** as an adversarial

selection mechanism, defined over our behavior embedding space created in Section V. Because this embedding space captures high-level semantic structure among 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 III. This overall process is visually represented in Figure 3.

Behavior Scoring. Recall that $\mathcal{P} : (S, \{\hat{\mathbf{X}}^{(k)}\}_{k=1}^K) \rightarrow \mathcal{B}_{\text{adv}}$ maps a base scenario S and up to K historical roll-outs $\{\hat{\mathbf{X}}^{(k)}\}_{k=1}^K$ to a desired adversary behavior \mathcal{B}_{adv} , which is then rolled-out in S to construct the next training or evaluation scenario. We begin by sampling a set of N_{cand} candidate adversary trajectories from a pre-trained trajectory predictor π_{gen} , which models plausible future trajectories conditioned on a fixed history portion of S , following the formulation in [12], [14]; that is, we collect $\{\tilde{\mathbf{X}}_{\text{adv},i}\}_{i=1}^{N_{\text{cand}}} \sim \pi_{\text{gen}}(\mathbf{X}_{\text{adv}} | S)$. For each candidate $\tilde{\mathbf{X}}_{\text{adv},i}$ and each historical ego trajectory $\hat{\mathbf{X}}_{\text{ego}}^{(k)}$, 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 [12], and obtain its embedding $v_{\text{adv},i}^{(k)}$ using the encoder defined in Section V-D.

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

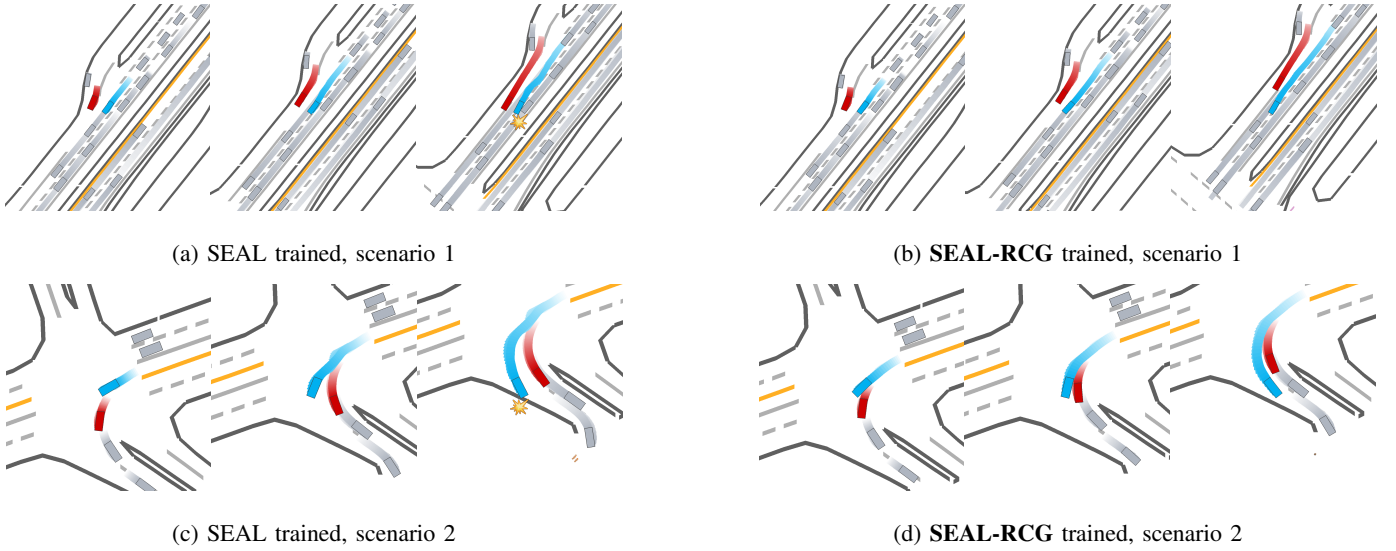


Fig. 4: Qualitative examples of closed-loop ego driving in **unmodified** hard scenarios from WOMD. In each row, the left depicts ego agents trained with SEAL alone; the right shows agents trained using RCG (Section VI) instantiated with SEAL. In Scenario 1 (top), our **ego** agent avoids a merging **adversary** 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.

evaluating the semantic plausibility of proposed adversarial behaviors. For each candidate, we then compute the average KNN distance across the K perturbed scenarios:

$$d_{\text{adv},i} = \frac{1}{K} \sum_{k=1}^K \text{KNN_dist}(v_{\text{adv},i}^{(k)}, \mathbf{V}_{\text{unsafe}}) \quad (7)$$

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 [12], defined as the minimum distance from each historical ego trajectory $\hat{X}_{\text{ego}}^{(k)}$ to a candidate $\tilde{X}_{\text{adv},i}$, averaged over the K rollouts. We retain the $N_{\text{int}} < N_{\text{cand}}$ candidates with the lowest average collision-closeness, forming the set $\{\tilde{X}_{\text{adv},i}\}_{i=1}^{N_{\text{int}}}$, and finally select the one with the minimum $d_{\text{adv},i}$ from this set as \tilde{X}_{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}_{adv} is instantiated using the selected trajectory \tilde{X}_{adv}^* , and may correspond to open-loop behavior (as in [12]) or a closed-loop reactive policy that utilizes the trajectory as a goal (as in [14]).

VII. 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 VII-A, RQ2 corresponding to Section VII-B, and RQ3 to Section VII-C. Ablation studies in Section VIII 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.

A. 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 VI. This directly assesses whether scenarios generated by our approach, compared to those from prior SOTA baselines (i.e., GOOSE [13], CAT [12], and SEAL [14]), provide more effective training stimuli, when tested in both unmodified and adversarially-perturbed scenarios.

We adopt the training and evaluation setup described in SEAL [14], using MetaDrive [62] as the simulator and focusing on the reinforcement learning of an ego agent in non-trivial but non-critical base scenarios from WOMD [6]. The learning framework, as established in SEAL, builds on ReSkill [42], a hierarchical reinforcement learning approach that combines offline skill learning with online residual fine-tuning. A continuous skill space is learned from heuristic

TABLE II: 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 [6]. 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
<i>None</i> (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)

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

Training Pipeline	Success (\uparrow)	Crash (\downarrow)	OoR (\downarrow)
<i>None</i> (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 IV: Average performance (%) of ego agents across ablations. “CL Pre” and “TADS FT” are contrastive pre-training (Section V-C) and fine-tuning (Section V-D); “Dist.” is the distance measure used in Equation (7) in Section VI. Selected adversary behaviors are instantiated with SEAL.

CL Pre	TADS FT	Dist.	Success (\uparrow)	Crash (\downarrow)	OoR (\downarrow)
–	–	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

expert demonstrations, while a high-level policy maps observations (including simulated LiDAR, navigation goals, and ego localization) to latent skill vectors at 1 Hz. A low-level policy, conditioned on the current skill, maps observations at 10 Hz to steering and acceleration commands. During episode roll-out, the skill space and both policies are frozen, while a residual policy is trained online to refine the low-level actions, producing a final control signal to send to the simulator.

Following SEAL, we use 400 base 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 VIII-A.

B. Embedding Space Creation Details

When implementing Section V, we pre-train our representation space using the WOMD [6] dataset, leveraging the MTR [37] implementation and training tools provided in UniTraj [63]. To ensure compatibility with this pipeline, we convert our TADS-`traj` dataset into the ScenarioNet [64] 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 [17]. 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 [65].

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 VIII-B.

C. Adversarial Generation Details

We generate adversarial candidate trajectories, as described in Section VI, using a pre-trained DenseTNT [66] model as the generative policy π_{gen} , following prior work [12], [14]. For each adversary agent, we sample $N_{\text{cand}} = 32$ candidate trajectories and retain the top $N_{\text{int}} = 6$ highly interactive behaviors.

To instantiate adversarial behaviors, we integrate our grounded objective into both CAT [12] and SEAL [2] pipelines by directly replacing their respective trajectory scoring objectives. Candidate selection is performed using the distance-based criterion in Equation (7), with $N_{\text{KNN}} = 8$ neighbors for SEAL and $N_{\text{KNN}} = 15$ for CAT. The larger neighborhood for

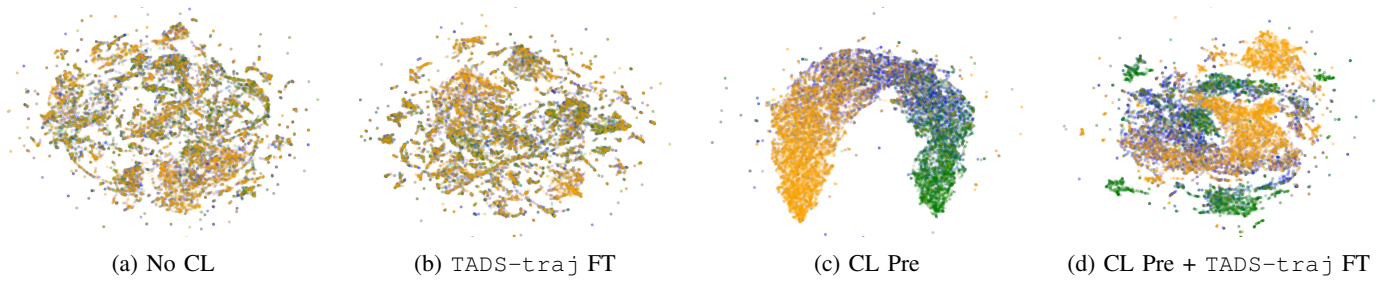


Fig. 5: UMAP [67] projections of the learned representation space, progressing through Section V. Each point is an agent behavior, colored by safety-label “safe”, “neutral”, or “unsafe”. Contrastive pre-training with large-scale data (CL, section V-C and fine-tuning with focused crash data in TADS-trajectory (FT, section V-D) reveal increasingly structured clusters and sub-clusters, aligned with safety semantics.

TABLE V: 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 \uparrow	Davies-Bouldin \downarrow	Linear Probe Acc. \uparrow
No CL	0.177	1.768	44.1%
TADS-trajectory FT	0.171	1.734	43.6%
CL Pre	0.189	1.624	46.5%
CL Pre + TADS-trajectory FT	0.198	1.278	46.8%

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 [12], [14]. We evaluate the quality of the generated adversarial trajectories $\hat{X}_{adv}^{(K)}$ 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 [14] and detailed in Section VIII-C.

VIII. RESULTS

A. Closed-Loop Training Results

Table II 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 outperform agents trained with GOOSE [13]-generated scenarios, as well as those trained without perturbations (“No Adv”).

Qualitatively, Figure 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 III

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 IV, breaking down the training pipeline along components introduced in Section V and Section VI. First, we ablate the contrastive pre-training described in Section V-C, training the embedding model with either \mathcal{L}_{recon} alone or the full loss from Equation (5). We then ablate the TADS-trajectory contrastive fine-tuning phase described in Section V-D, either omitting or including it. Finally, we ablate the distance measure used for candidate scoring against the embedding cache, as formalized in Equation (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.

B. 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 V and projected from WOMB input scenarios.

Figure 5 illustrates how the representation evolves across training stages. Without contrastive learning (Figure 5a), the space lacks any clear organization with respect to safety. Applying prototypical contrastive regularization (Figure 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-trajectory dataset (Figure 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 5b) fails to meaningfully reshape the space, supporting the claim that our full pipeline is necessary.

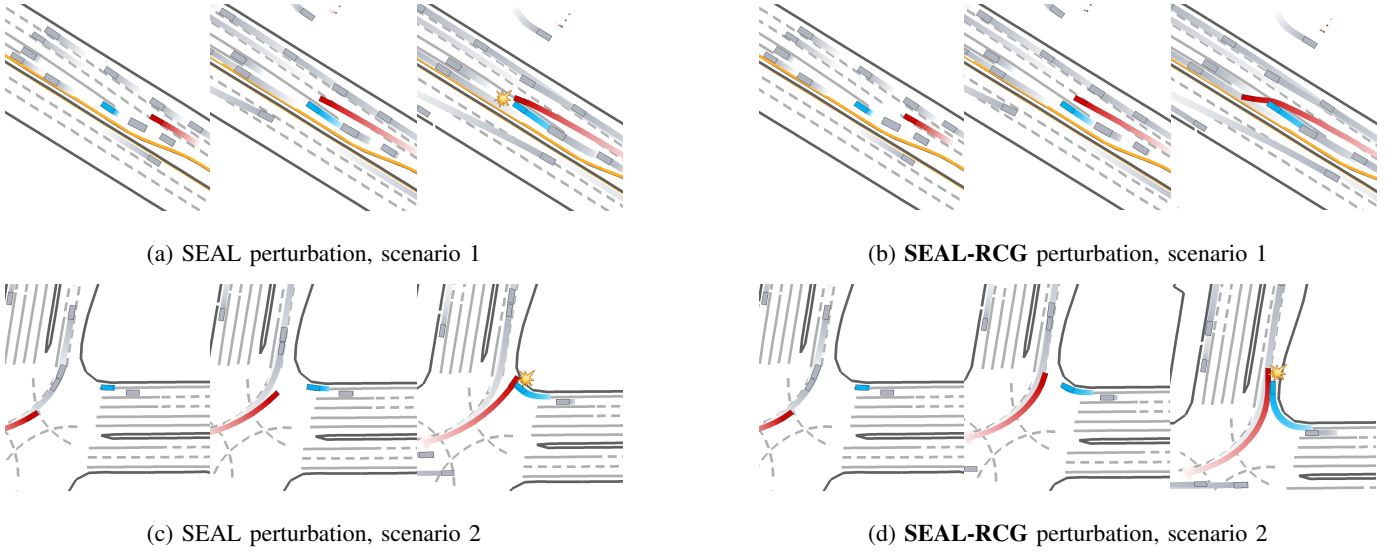


Fig. 6: Qualitative examples of adversarial **perturbation** against an ego replay policy. In each row, the left shows perturbations from SEAL alone, and the right shows SEAL-RCG, applied to the same base scenario. In Scenario 1 (top), the **adversary** merges directly into the **ego** under SEAL, whereas 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.

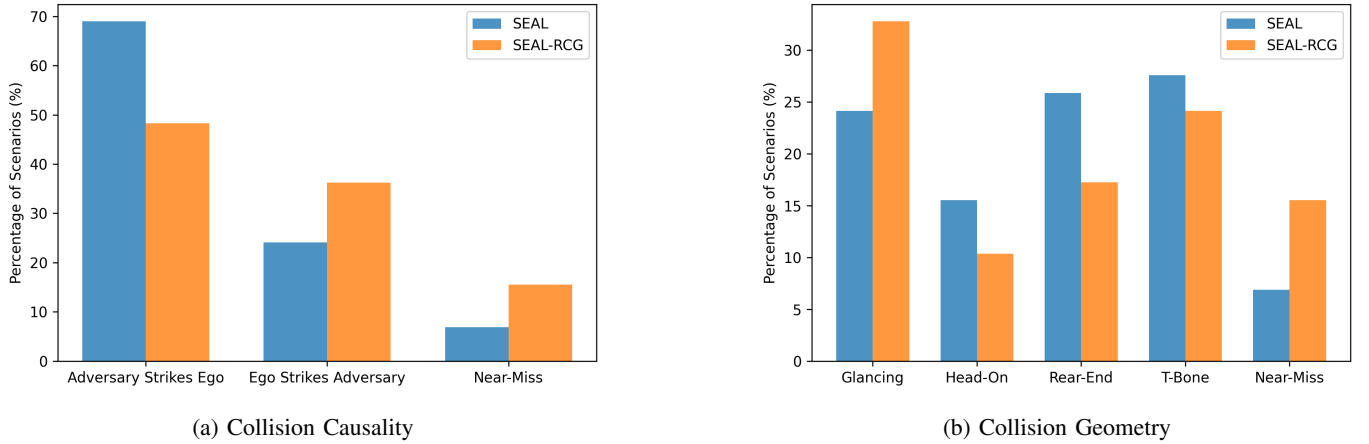


Fig. 7: Taxonomizations of generated adversarial interactions across matched base scenarios, against an ego replay policy.

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} clusters) 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 [68], [69]. For informativity, we train linear probes to predict safety labels on the same held-out validation set used in Section V. Results are reported in Table V; 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.

C. Generated Scenario Results

We highlight representative qualitative examples of adversarial scenario generation in Figure 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. 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

TABLE VI: 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 (\downarrow)	Yaw WD (\downarrow)	Acc WD (\downarrow)
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

and fewer t-bone or in-line collisions, better aligning with adversary maneuvers that *reduce* impact severity, as observed in real-world crash data such as TADS-*traj*.

Quantitatively, as shown in Table VI, 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 VIII-A, 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 [14]. 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**.

IX. CONCLUSION

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.

REFERENCES

- [1] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, “Did we test all scenarios for automated and autonomous driving systems?” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2950–2955.
- [2] B. Stoler, I. Navarro, M. Jana, S. Hwang, J. Francis, and J. Oh, “Safeshift: Safety-informed distribution shifts for robust trajectory prediction in autonomous driving,” in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, pp. 1179–1186.
- [3] J. Sun, H. Zhang, H. Zhou, R. Yu, and Y. Tian, “Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance,” *IEEE transactions on intelligent transportation systems*, vol. 23, no. 9, pp. 14088–14103, 2021.
- [4] N. Webb, D. Smith, C. Ludwick, T. Victor, Q. Hommes, F. Favaro, G. Ivanov, and T. Daniel, “Waymo’s safety methodologies and safety readiness determinations,” *arXiv preprint arXiv:2011.00054*, 2020.
- [5] N. Kalra and S. 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*, vol. 94, pp. 182–193, 2016.
- [6] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. 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*, 2021, pp. 9710–9719.
- [7] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles,” *arXiv preprint arXiv:2106.11810*, 2021.
- [8] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes *et al.*, “Argoverse 2: Next generation datasets for self-driving perception and forecasting,” *arXiv preprint arXiv:2301.00493*, 2023.
- [9] H. X. Liu and S. Feng, “Curse of rarity for autonomous vehicles,” *nature communications*, vol. 15, no. 1, p. 4808, 2024.
- [10] P. Huang, W. Ding, B. Stoler, J. Francis, B. Chen, and D. 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)*. IEEE, 2025.
- [11] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, “A survey on safety-critical driving scenario generation—a methodological perspective,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6971–6988, 2023.
- [12] L. Zhang, Z. Peng, Q. Li, and B. Zhou, “Cat: Closed-loop adversarial training for safe end-to-end driving,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2357–2372.
- [13] J. Ransiek, J. Plaum, J. Langner, and E. Sax, “Goose: Goal-conditioned reinforcement learning for safety-critical scenario generation,” in *2024 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2024.
- [14] B. Stoler, I. Navarro, J. Francis, and J. Oh, “Seal: Towards safe autonomous driving via skill-enabled adversary learning for closed-loop scenario generation,” *arXiv preprint arXiv:2409.10320*, 2024.
- [15] Y. Chai, J. Fang, H. Liang, and W. Silamu, “Tads: a novel dataset for road traffic accident detection from a surveillance perspective,” *The Journal of Supercomputing*, vol. 80, no. 18, pp. 26226–26249, 2024.
- [16] L. Xu, H. Huang, and J. 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*, 2021, pp. 9878–9888.
- [17] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, “Prototypical contrastive learning of unsupervised representations,” in *2021 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2021.

- [18] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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*, 2020, pp. 2446–2454.
- [19] X. Zhang, Z. Li, Y. Gong, D. Jin, J. Li, L. Wang, Y. Zhu, and H. Liu, “Openmpd: An open multimodal perception dataset for autonomous driving,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2437–2447, 2022.
- [20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nusenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [21] R. Krajewski, J. Bock, L. Kloecker, and L. 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)*. IEEE, 2018, pp. 2118–2125.
- [22] J. Fang, L.-l. Li, J. Zhou, J. Xiao, H. Yu, C. Lv, J. Xue, and T.-S. Chua, “Abductive ego-view accident video understanding for safe driving perception,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 030–22 040.
- [23] A. P. Shah, J.-B. Lamare, T. Nguyen-Anh, and A. 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)*. IEEE, 2018, pp. 1–9.
- [24] X. Yan, Z. Zou, S. Feng, H. Zhu, H. Sun, and H. X. Liu, “Learning naturalistic driving environment with statistical realism,” *Nature communications*, vol. 14, no. 1, p. 2037, 2023.
- [25] S. Suo, K. Wong, J. Xu, J. Tu, A. Cui, S. Casas, and R. Urtasun, “Mixsim: A hierarchical framework for mixed reality traffic simulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9622–9631.
- [26] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, “Bits: Bi-level imitation for traffic simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2929–2936.
- [27] J. Ge, J. Zhang, C. Chang, Y. Zhang, D. Yao, and L. Li, “Task-driven controllable scenario generation framework based on aog,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 6186–6199, 2024.
- [28] J. M. Scanlon, K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, “Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain,” *Accident Analysis & Prevention*, vol. 163, p. 106454, 2021.
- [29] B. Zhu, P. Zhang, B. Liu, X. Dai, W. Deng, “Hazardous scenario enhanced generation for automated vehicle testing based on optimization searching method,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7321–7331, 2021.
- [30] X. Li, S. Teng, B. Liu, X. Dai, X. Na, and F.-Y. Wang, “Advanced scenario generation for calibration and verification of autonomous vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3211–3216, 2023.
- [31] W. Ding, Y. Cao, D. Zhao, C. Xiao, and M. Pavone, “Realgen: Retrieval augmented generation for controllable traffic scenarios,” in *European Conference on Computer Vision*. Springer, 2024, pp. 93–110.
- [32] C. Xu, D. Zhao, A. Sangiovanni-Vincentelli, and B. Li, “Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles,” in *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023.
- [33] W.-J. Chang, F. Pittaluga, M. Tomizuka, W. Zhan, and M. Chandraker, “Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.00391>
- [34] D. Remppe, J. Philion, L. J. Guibas, S. Fidler, and O. 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*, 2022, pp. 17 305–17 315.
- [35] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. 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*. Springer, 2020, pp. 624–641.
- [36] Z. Zhang, A. Liniger, C. Sakaridis, F. Yu, and L. V. Gool, “Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 57 481–57 499, 2023.
- [37] S. Shi, L. Jiang, D. Dai, and B. Schiele, “Motion transformer with global intention localization and local movement refinement,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.
- [38] W. Wu, X. Feng, Z. Gao, and Y. Kan, “Smart: Scalable multi-agent real-time motion generation via next-token prediction,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 114 048–114 071, 2024.
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [40] R. Tian, B. Li, X. Weng, Y. Chen, E. Schmerling, Y. Wang, B. Ivanovic, and M. Pavone, “Tokenize the world into object-level knowledge to address long-tail events in autonomous driving,” *arXiv preprint arXiv:2407.00959*, 2024.
- [41] C. Hao, C. Weaver, C. Tang, K. Kawamoto, M. Tomizuka, and W. Zhan, “Skill-critic: Refining learned skills for hierarchical reinforcement learning,” *IEEE Robotics and Automation Letters*, 2024.
- [42] K. Rana, M. Xu, B. Tidd, M. Milford, and N. Sünderhauf, “Residual skill policies: Learning an adaptable skill-based action space for reinforcement learning for robotics,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2095–2104.
- [43] Q. Wang, D. Xu, G. Kuang, C. Lv, S. E. Li, and B. Nie, “Risk-aware vehicle trajectory prediction under safety-critical scenarios,” *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [44] T. van der Heiden, N. S. Nagaraja, C. Weiss, and E. Gavves, “Safecritic: Collision-aware trajectory prediction,” *arXiv preprint arXiv:1910.06673*, 2019.
- [45] H. Hu, X. Wang, Y. Zhang, Q. Chen, and Q. Guan, “A comprehensive survey on contrastive learning,” *Neurocomputing*, p. 128645, 2024.
- [46] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, “A survey on self-supervised learning: Algorithms, applications, and future trends,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [47] Y. Wang, P. Zhang, L. Bai, and J. 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*, 2023, pp. 1400–1409.
- [48] J. Zhang, M. Pourkeshavarz, and A. Rasouli, “Tract: A training dynamics aware contrastive learning framework for long-tail trajectory prediction,” in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, pp. 3282–3288.
- [49] C. Liu, J. Yu, and Q. 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)*. IEEE, 2024, pp. 1352–1357.
- [50] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [51] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *European Conference on Computer Vision*. Springer, 2024, pp. 38–55.
- [52] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, “Sam 2: Segment anything in images and videos,” *arXiv preprint arXiv:2408.00714*, 2024.
- [53] Z. Fang, C. Liang, X. Zhou, S. Zhu, and X. Li, “Associate everything detected: Facilitating tracking-by-detection to the unknown,” *arXiv preprint arXiv:2409.09293*, 2024.
- [54] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, “Xfeat: Accelerated features for lightweight image matching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2682–2691.
- [55] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu, “Unidepth: Universal monocular metric depth estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 106–10 116.
- [56] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, “Video depth anything: Consistent depth estimation for super-long videos,” *arXiv preprint arXiv:2501.12375*, 2025.
- [57] R. Huang, H. Zheng, Y. Wang, Z. Xia, M. Pavone, and G. Huang, “Training an open-vocabulary monocular 3d detection model without 3d data,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 72 145–72 169, 2024.
- [58] O. Makansi, Ö. Çiçek, Y. Marrakchi, and T. Brox, “On exposing the challenging long tail in future prediction of traffic actors,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 147–13 157.
- [59] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J. Low, L. Bing, and L. Si, “On the effectiveness of adapter-based tuning for pretrained language model adaptation,” in *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)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 2208–2222. [Online]. Available: <https://aclanthology.org/2021.acl-long.172/>
- [60] Y. Lin, X. Ma, X. Chu, Y. Jin, Z. Yang, Y. Wang, and H. Mei, “Lora dropout as a sparsity regularizer for overfitting control,” *arXiv preprint arXiv:2404.09610*, 2024.
 - [61] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models,” in *2022 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2022.
 - [62] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, “Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3461–3475, 2022.
 - [63] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi, “Unitraj: A unified framework for scalable vehicle trajectory prediction,” in *European Conference on Computer Vision*. Springer, 2024, pp. 106–123.
 - [64] Q. Li, Z. M. Peng, L. Feng, Z. Liu, C. Duan, W. Mo, and B. Zhou, “Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling,” *Advances in neural information processing systems*, vol. 36, pp. 3894–3920, 2023.
 - [65] V. Lialin, N. Shivagunde, S. Muckatira, and A. Rumshisky, “Relora: High-rank training through low-rank updates,” in *2024 IEEE International Conference on Learning Representations (ICLR)*. IEEE, 2024.
 - [66] J. Gu, C. Sun, and H. Zhao, “DenseTnt: End-to-end trajectory prediction from dense goal sets,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.
 - [67] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.
 - [68] R. Chua, A. Ghosh, C. Kaplanis, B. Richards, and D. Precup, “Learning successor features the simple way,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 49 957–50 030, 2024.
 - [69] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Clustering algorithms and validity measures,” in *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001*. IEEE, 2001, pp. 3–22.

Benjamin Stoler is a Ph.D. candidate in the Computer Science Department at Carnegie Mellon University. Benjamin’s research focuses on safety and robustness in autonomous driving and social robot navigation. Benjamin obtained his B.S. and M.S. degrees in Computer Science from the College of Engineering at the University of Michigan.

Juliet Yang is an undergraduate student in the Computer Science Department at Carnegie Mellon University. Juliet is interested in exploring ways to improve the robustness of autonomous driving systems.

Jonathan Francis is a Senior Research Scientist at the Bosch Center for Artificial Intelligence (BCAI), Courtesy Faculty in the Robotics Institute at Carnegie Mellon University (CMU), and the Robotics Research Program Lead in the Carnegie-Bosch Institute, College of Engineering, at CMU. Jonathan’s research focuses on developing autonomous systems that learn safe, robust, multimodal, and transferrable representations of the world and are therefore capable of acquiring, adapting, and improving their own skills/behavior when deployed to diverse scenarios. Jonathan received his Ph.D. from the School of Computer Science at CMU and his B.S. and M.S. in Electrical & Computer Engineering at CMU.

Jean Oh is an Associate Research Professor at the Robotics Institute at Carnegie Mellon University (CMU) and head of the roBot Intelligence Group (BIG). Jean’s research goal is to develop technologies to remind us of “what makes us humans,” promoting humanity such as *safety*, *creativity*, and *compassion* in various problem domains including self-driving, safe aviation, and arts. Jean’s works on social navigation and creative robotics have won several best paper awards at various robotics conferences such as IEEE International Conference on Robotics and Automation (ICRA) and ACM/IEEE International Conference on Human-Robot Interaction (HRI), and featured in media around the world including New York Times and CNET. Jean received PhD at CMU, MS at Columbia University, and BS at Yonsei University.