What Probing Reveals about Autonomous Driving: Linking Internal Prediction Errors to Ego Planning

Hyeonchang Jeon[1] **Kyungbeom Kim**[1] **Kyung-Joong Kim**[1,‡,*] **Eugene Vinitsky**[2,‡]
[1]**Gwangju Institute of Science and Technology (GIST)** [2]**New York University**
{kevinjeon119, kyungbeom8}@gm.gist.ac.kr
kjkim@gist.ac.kr vinitsky.eugene@nyu.edu

Abstract

Large-scale datasets and fast simulators have improved driving policies, yet strong nominal performance can still mask flawed reasoning and unsafe heuristics. Summary scores from closed-loop simulators offer little insight into the policy, making it hard to tell whether they truly predict surrounding-vehicle motion, how the ego vehicle generates future plans, or whether they rely on brittle heuristics that happen to succeed in nominal scenarios. To better understand these limitations, we probe two capabilities: *prediction*, i.e., where surrounding vehicles will move next, and *planning*, i.e., how the ego vehicle generates future trajectories. To evaluate the presence of these capabilities, we investigate them as a function of scale, asking whether the closed-loop gains from larger datasets and longer simulation training reflect stronger prediction and planning or merely better behavioral heuristics. Using linear probing and targeted perturbations across imitation learning and reinforcement learning policies, we track when these internal signals emerge, plateau, or fail. Despite good closed-loop performance, policies often exhibit weak or delayed surrounding-vehicle predictions during near-collision events, revealing a limitation in the predictive signals available for ego planning. Finally, causal intervention shows that modifying prediction-related representations changes ego-planning and can improve goal progress. Project page: https://kevinjeon.github.io/DrivingProbing/index.html

1 Introduction

The combination of large-scale driving datasets [Ettinger et al., 2021, Caesar et al., 2020, Chang et al., 2019, Wilson et al., 2021] and GPU-accelerated simulation environments [Kazemkhani et al., 2025, Gulino et al., 2023] has enabled rapid advances in data-driven autonomous driving, particularly in learning-based prediction and planning [Shi et al., 2022, Nayakanti et al., 2023, Sima et al., 2024, Philion et al., 2024, Feng et al., 2023, Hu et al., 2022]. However, despite this progress, available driving data remains far smaller than datasets in domains such as NLP and is heavily biased toward nominal, uninteresting scenarios in which drivers simply drive straight [Zhai et al., 2023, Li et al., 2024, Jeon et al., 2024, Makansi et al., 2021]. Furthermore, scenarios that truly test the generalization of driving models are rare, leaving open whether these models learn generalizable rules or instead rely on spurious correlations and shortcut heuristics. These limitations and evaluation gaps raise a question: Are driving agents truly learning the core prediction and planning skills required for safe and robust driving [Makansi et al., 2021, Sun et al., 2024]?

Several natural indicators could assess understanding of driving policies; conversely, their absence would suggest a failure to internalize key skills. A robust driving policy should respond sensibly to changes in its surroundings, distinguish relevant from irrelevant vehicles, and maintain plans for

‡Equal advising.
*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

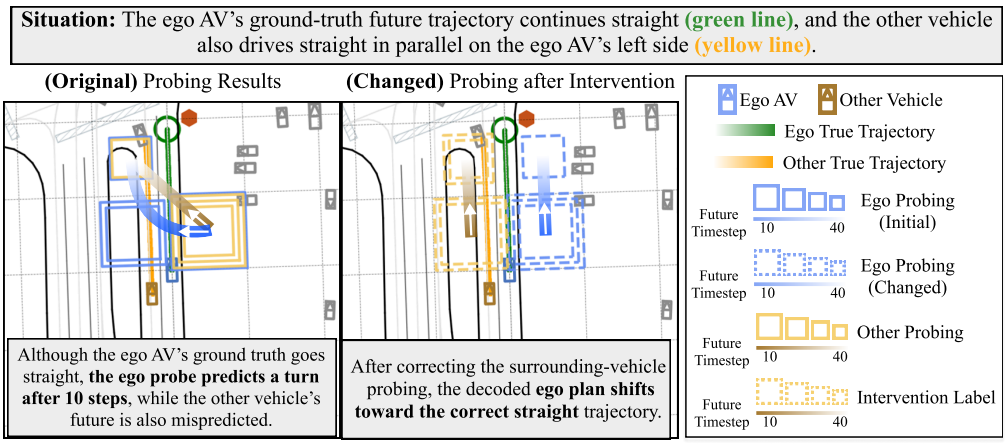


Figure 1: **Recovery of planning with correction of prediction.** The ego AV continues straight, while the other vehicle drives parallel to the ego AV on its left. Gradient arrows simplify the probing results and show the decoded future motion direction over time. **(Original)**: After the 10-step prediction, the decoded ego plan curves away from its straight ground-truth trajectory, while the other vehicle's future is also mispredicted. **(Changed)**: Correcting the surrounding-vehicle prediction shifts the decoded ego plan toward the correct straight trajectory.

safety-critical interactions. For example, removing a lead vehicle should allow the ego to accelerate, while a slowing or newly introduced lead vehicle should induce deceleration. Likewise, the policy should focus on nearby vehicles that may affect its future path rather than distant vehicles with little relevance to the ego plan, and should remain prepared for unexpected interactions such as abrupt cut-ins. These behaviors motivate our focus on two probeable capabilities: surrounding-vehicle prediction and adaptive ego planning.

From these requirements—reasoning about surrounding vehicles and maintaining an internal plan—we narrow our focus to two concrete, probeable competencies. We define *prediction* as inferring the future positions of surrounding vehicles, and *planning* as the adaptiveness of the ego autonomous vehicle (AV)'s planning in response to changes in predictions during decision-time planning [Bush et al., 2025] as shown in Figure 1. However, a driving model can perform well on driving metrics while failing to internalize these capabilities. Indeed, evidence shows that one can perform well on standard benchmarks while entirely neglecting information beyond ego state [Li et al., 2024], or achieve high scores while maintaining incorrect causal models [Sun et al., 2024]. Even if these failure modes are avoided, simple rules like "identify a lead vehicle and follow it at a safe distance" can be effective on most scenes while failing critically on the rare scenarios that make the rule unsafe.

In this paper, we investigate the internals and behavior of driving models ranging from simple behavior cloning (BC) to state-of-the-art imitation learning (IL) and reinforcement learning (RL) policies, to understand how prediction and planning are accessible in their representations. In particular, we study how surrounding-vehicle and ego-planning information becomes accessible and behaviorally relevant as data scale increases. We train driving models at multiple dataset sizes [Zheng et al., 2024, Naumann et al., 2025, Baniodeh et al., 2025] and evaluate each scale using perturbed closed-loop simulation, probing, and targeted interventions.

To test how driving policies depend on surrounding-vehicle context, we use *distance-ordered removal* with *nearest-first* and *farthest-first* settings to distinguish contextual sensitivity from reduced-interaction distribution shift. To quantify what information about nearby agents is linearly accessible from the model's representations, we use *surrounding-vehicle linear probing* [Alain and Bengio, 2016] to decode future positions of nearby agents from internal representations, and benchmark the linearly accessible information against probes trained directly on raw inputs. We further analyze simulated near-collision events to examine how probing performance relates to actual collision outcomes. Lastly, to study how surrounding-vehicle information influences planning, we use *ego-linear probing* to decode the ego AV's planned future positions and perform a decision-time *intervention experiment* that modifies internal representations through these probes, testing how modifying prediction-related representations changes ego planning [Bush et al., 2025].

Our results show that prediction and planning are linearly accessible in driving model representations, but remain imperfect in ways that nominal closed-loop performance can obscure. Perturbation and intervention analyses further reveal brittleness to reduced-interaction settings and show that prediction-related representations are behaviorally connected to ego planning. Our contributions are as follows:

- **Representation analysis across scales:** Using linear probes, we find that prediction and planning become linearly accessible in driving model representations, but surrounding-vehicle prediction remains imperfect and does not consistently improve with scale, especially at longer horizons.
- **Perturbed closed-loop evaluation:** Perturbed closed-loop evaluation shows limited generalization to uncommon single-AV observations and additional brittleness when nearby-agent context is removed from originally interactive scenes.
- **Near-collision analysis and causal intervention:** Prediction signals become strongest only near collision, while causal interventions along prediction-related directions causally alter ego planning and can improve avoidance responses or goal progress.

2 Related Work

2.1 Planning in Autonomous Driving

Autonomous driving research has access to abundant datasets of human behavior [Caesar et al., 2020, Chang et al., 2019, Ettinger et al., 2021]. However, these datasets are heavily skewed toward nominal, low-interaction scenarios, leaving open the question of whether standard closed-loop simulation—typically evaluated on the same distribution—can truly measure robust behavior in safety-critical interactions. Standard planning models [Nayakanti et al., 2023, Chai et al., 2020, Seff et al., 2023] are widely used for both prediction and planning, yet current evaluations rarely test whether they use surrounding-vehicle information in a timely and safety-relevant manner. Recent studies [Li et al., 2024, Zhai et al., 2023] show that source datasets are dominated by nominal straight-driving scenarios, where the single ego AV often suffices to achieve strong closed-loop performance. As a result, closed-loop metrics can be misleading, indicating high performance even when the model largely ignores other vehicles. This motivates alternative evaluations that assess whether surrounding-vehicle information is accessible in model representations and behaviorally relevant to planning. In this paper, we study how prediction and planning abilities in driving models are reflected in their representations and analyze perturbed simulation results using linear probing.

2.2 Data Scaling Trends

With the rise of foundation models, scaling laws have become an important research topic [Henighan et al., 2020, Tian et al., 2024, Bharadhwaj et al., 2024]. They study how performance varies with dataset size, model capacity, and compute resources, especially for transformer-based models. This perspective has recently been extended to autonomous driving [Zheng et al., 2024, Naumann et al., 2025, Baniodeh et al., 2025]. For example, [Zheng et al., 2024] introduced ONE-Drive and showed that augmenting rare scenarios is more effective than sampling random scenarios. [Naumann et al., 2025] studied end-to-end scaling through larger models and additional camera inputs. [Baniodeh et al., 2025] found stronger alignment between open-loop and closed-loop metrics and identified compute-optimal model sizes. Similarly, we study how autonomous driving performance scales with dataset size and evaluate models with both metrics. Unlike prior work, however, we focus on how prediction and planning capabilities themselves scale with data.

2.3 Interpretability in Deep Learning

Linear probing measures information accessible from learned representations through a simple linear readout [Alain and Bengio, 2016, Mikolov et al., 2013]. It has been used to study planning in model-free RL and supervised agents [Kim et al., 2018, Guez et al., 2019], and in autonomous driving to analyze affordance prediction and interpretable control from frozen policy encoders [Xiao et al., 2021]. More recent work combines probing with sparse autoencoders (SAEs) to disentangle latent factors and interpret or control ego motion [Bricken et al., 2023, Tas and Wagner, 2025].

In this paper, we use linear probing to test whether future information is accessible through a simple linear readout, rather than recoverable by a highly expressive decoder. More expressive probes capture broader recoverability, while attribution [Kim and Canny, 2017] and SAE [Bricken et al., 2023, Tas and Wagner, 2025] analyses address complementary questions about input influence and latent structure. Accordingly, our conclusions concern only *linearly accessible* prediction and planning information.

3 Approaches

In this section, we analyze how driving policies represent and use surrounding-vehicle information for decision making: (i) whether information about safety-relevant agents is linearly accessible, (ii) whether such predictions are timely in near-collision situations, and (iii) how prediction representations causally influence planning via targeted interventions.

3.1 Experimental Setup

Models. We compare a transformer-based behavior cloning policy similar to Gulino et al. [2023] (BC) and a PPO-based RL policy [Schulman et al., 2017, Cornelisse et al., 2025b] (RL) across data scales. We additionally apply our linear probing and causal intervention analyses to SMART [Wu et al., 2024] (IL), a multi-agent trajectory prediction model, demonstrating that our probing-based framework extends to state-of-the-art IL models. Detailed descriptions are provided in Appendix A.

Dataset. For the BC and RL models, we train our model using the Waymo Open Motion Dataset (WOMD) [Ettinger et al., 2021]. The dataset consists of over $400K$ driving scenes, each containing 9 seconds of trajectory data sampled at 10 Hz, with up to 128 cars per scene. Due to limitations of the underlying simulator used to replay the trajectories, we exclude scenes containing traffic lights and overpasses, filtering the dataset to $\sim 90K$ scenes. Each observation is partially observable and includes information about the ego vehicle, surrounding vehicles (up to 127), and the map in an ego-centric view. As action labels are not available in the dataset, we derive them through inverse kinematics. For the IL model, we set the same setting in SMART [Wu et al., 2024]. We train using the WOMD dataset with 11 historical steps as input, including global coordinate, heading information, and predict the remaining future trajectories for the target. Since the IL model predicts multi-agent trajectories, it can predict up to 32 vehicles per scenario.

More details on the dataset can be found in Appendix B. For a detailed analysis of closed-loop simulations, we first preprocess trajectories to remove those with infeasible kinematics (likely due to noise during data collection), and then classify the remaining trajectories into mutually exclusive categories (*Straight*, *Turn*, *Reverse*, and *Uncategorized*). (See details in Appendix B.3)

Linear Probing. To interpret the driving model's internal representation, we perform a linear probing experiment. We select the best model across seeds for each dataset scale and train a linear classifier to predict future positions. The ground truth is set as the vehicle's positions (ego or surrounding vehicle) 1 to 4 seconds ahead (corresponding to 10, 20, 30, and 40 steps). The positions are discretized into 64 labels by dividing the ego vehicle's current field of view into an 8×8 grid along the x- and y-axes, normalized with respect to the ego AV's current position. To understand what information is retained in the internal network layers, we train linear classifiers on both the raw input and representations from the early and late attention layers. We evaluate performance using the F1 score to account for data imbalance, as well as per-trajectory-type accuracy to investigate how linearly accessible information differs across cases. For the IL model, which predicts multi-agent trajectories, there is no predefined ego agent. We therefore sample one vehicle as the ego and select its nearest neighboring vehicle as the surrounding vehicle. Since our goal is to analyze how information about surrounding vehicles contributes to ego-trajectory prediction, this setup provides a natural egocentric perspective for evaluating the use of information from other agents. The linear probing setting is described in Appendix D.1 and Appendix E.1.

3.2 Data Scaling Trends

To investigate the emergence of planning and prediction ability, we train the BC and RL models with three different random seeds while gradually increasing the dataset size, and evaluate them in the GPUDrive simulator [Kazemkhani et al., 2025] on unseen scenarios. For RL, we train on up to

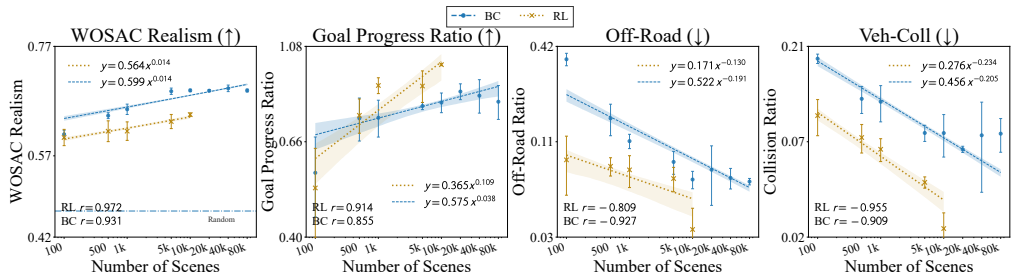


Figure 2: **Power-law-like scaling trends of BC and RL models**: We evaluate the WOSAC, collision metrics (Off-Road and Veh-Coll), and goal progress rate. Curves are fitted with $y = ax^b$, and r denotes the Pearson correlation coefficient computed in log–log space.

10K scenes, at which point performance has nearly converged, as shown in prior work [Cornelisse et al., 2025b]. To support the generality of our evaluation, we additionally report the Waymo Open Sim Agents Challenge (WOSAC) metrics [Montali et al., 2023]. For closed-loop simulation, we use three metrics: vehicle collision (Veh-Coll), off-road rate, and goal progress ratio (calculated as $1 - d_{\text{final}}/d_{\text{initial}}$, where d_{final} and d_{initial} denote the final and initial distances to the goal), respectively.

As shown in Figure 2, both BC and RL exhibit clear performance improvements with scale and power-law-like trends over the evaluated range. For BC, the gains largely saturate beyond approximately 20K scenes, consistent with trends reported in prior work [Baniodeh et al., 2025]. RL achieves substantially lower Veh-Coll and off-road rates and reaches approximately 99% goal progress at 10K scenes. The fitted curves show strong correlations in log–log space, suggesting that a power-law fit provides a useful empirical description of scaling over the evaluated range, rather than establishing a general scaling law. See Appendix C.1 and Appendix C.3 for additional results.

3.3 Linear Probing for Surrounding Vehicles Prediction

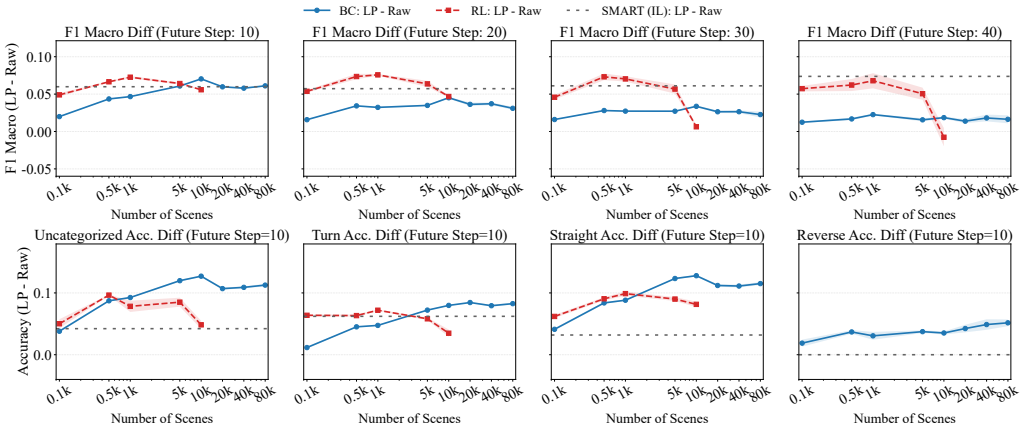


Figure 3: **Performance metrics with surrounding-vehicle linear probing.** Top: F1 Macro score difference between linear probes on trained model representations and the raw-input baseline (LP − Raw) across future steps $fs \in 10, 20, 30, 40$, where fs denotes the prediction horizon in timesteps. Bottom: trajectory-type accuracy difference at $fs = 10$ for $Straight$, $Turn$, $Reverse$, and $Uncategorized$ cases. Higher values are better for both metrics. The $Reverse$ case of RL is omitted because no $Reverse$ cases are available for the RL setting.

To understand how the driving model uses surrounding-vehicle information internally, we train linear probes on the intermediate representations of our trained model to assess the accessibility of prediction and planning capabilities, using the linear probing method [Alain and Bengio, 2016]. We train the linear probes with three different random seeds while gradually increasing the dataset size. Since the models differ in architecture and input dimensionality (i.e., the RL model uses a single timestep, whereas the BC model uses five timesteps), directly comparing raw probing performance across models can be misleading. Therefore, for cross-model comparison, we report the relative

probing gain, computed as the difference between the trained model's probe and the raw-input probe (LP − Raw). This measures how much additional information is linearly accessible from the learned representation beyond what a linear readout of the raw input provides.

As shown in Figure 3, BC probing gains increase up to 10K scenes and then largely plateau, broadly following the closed-loop scaling trend in Section 3.2. In contrast, RL does not improve monotonically: from 5K to 10K scenes, driving and ego-plan probing improve while surrounding-vehicle probing weakens. At 10K scenes, RL remains above BC at the 20-, 30-, and 40-step horizons but falls below BC at the 10-step horizon. This may reflect increasing specialization toward ego-centric control, reducing the linear accessibility of surrounding-vehicle futures. Lastly, SMART (IL) is comparable to 80K BC at 10 steps but retains stronger LP–Raw gains at longer horizons, suggesting better long-horizon accessibility of surrounding-vehicle information. Type-wise results show larger BC gains for Straight and Uncategorized than for Turn and Reverse, while RL changes little with scale. SMART's advantage is more apparent in long-horizon F1 than in type accuracy. More detailed results are provided in Appendix D.

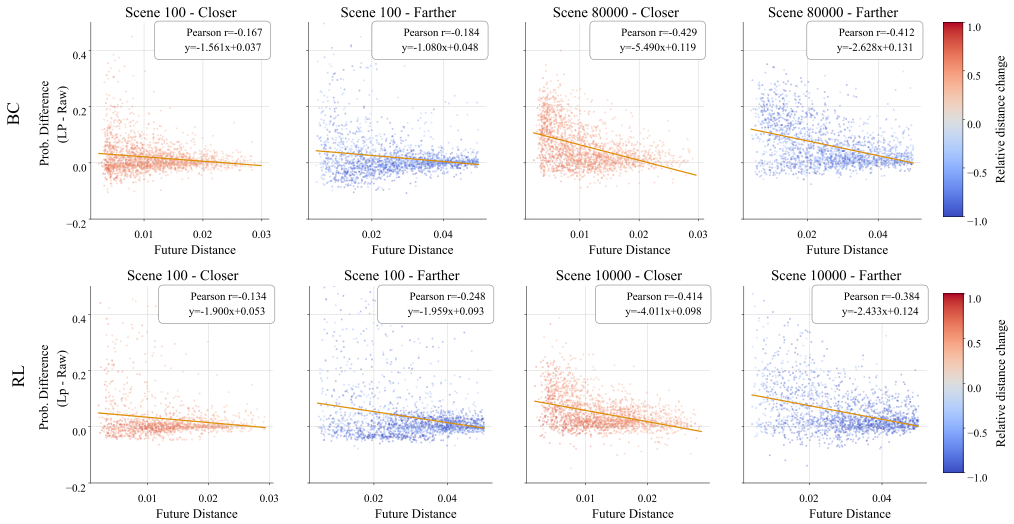
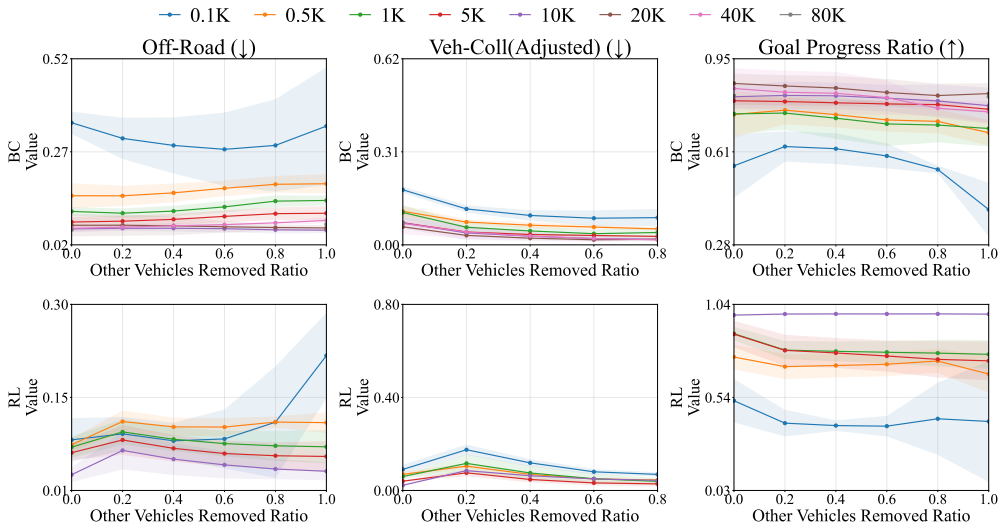


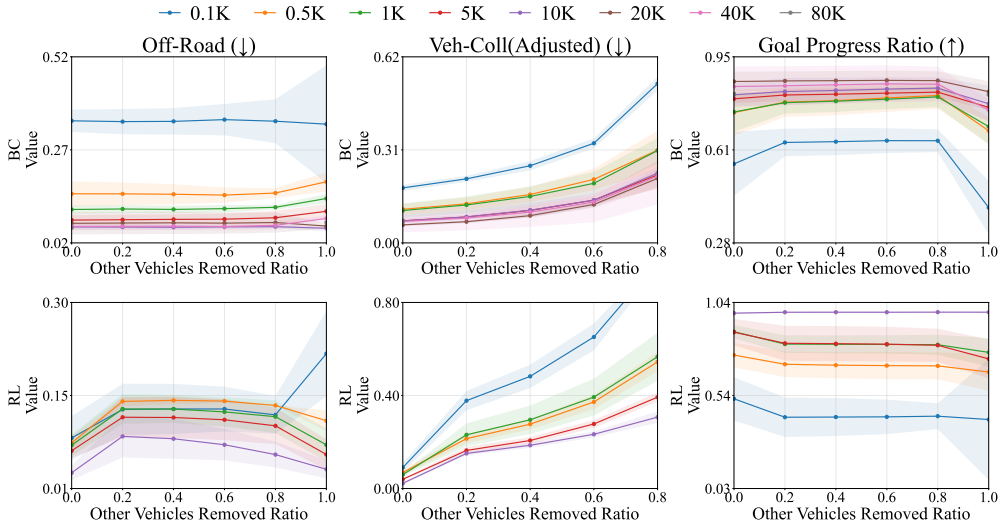
Figure 4: **Correlation between linear-probe and future ego–other distance.** Vertical: probability gap (LP – raw); horizontal: future ego–other distance. Orange: regression line; r: correlation.

However, this nominal understanding does not directly assess predictive ability, since only a few vehicles are actually important among the many surrounding vehicles in each scenario. To address this, we further analyze whether the surrounding-vehicle linear probe is stronger for vehicles that will become closer to the ego in the future. In Figure 4, we plot the correlation between linear-probe prediction and future ego–other distance on both BC and RL models. The vertical axis shows the difference in predicted probability for the true label between the LP and the raw-input probe, while the horizontal axis shows the future distance 10 steps ahead. Points are colored by relative distance change, $\frac{current-future}{current}$: samples that become more than 40% closer are shown in red, whereas those that become more than 40% farther are shown in blue, with opacity increasing with the magnitude of the change.

At a small scale (100 scenes), the linear-probe advantage over the raw-input probe is only weakly associated with future ego–other distance, indicating limited sensitivity to future interaction relevance. With more training data (10,000 and 80,000 scenes), the association becomes markedly more negative, indicating that the probing advantage becomes stronger for vehicles that remain close or move closer to the target. However, among nearby vehicles, the probing signal remains similar for both approaching and receding nearby vehicles. This suggests that scaling increasingly differentiates distant from nearby vehicles in the probing signal, but it does not clearly distinguish which nearby vehicles will become most interaction-relevant, i.e., those that will move even closer to the target.



(a) **Nearest-first removal.**



(b) **Farthest-first removal.**

Figure 5: **Distance-ordered surrounding-vehicle removal.** Surrounding vehicles are removed progressively in nearest-first and farthest-first order and evaluate BC and RL policies across removal ratios using off-road rate, goal progress ratio, and adjusted vehicle-collision rate, where collision is normalized by the remaining-vehicle ratio $(1 - p)$ where p is partner removal ratio.

3.4 Sensitivity to Surrounding-Vehicle Removal

If driving models use surrounding-vehicle information selectively, removing distant vehicles should have less effect than removing nearby vehicles that are more likely to affect ego planning. To test this, we progressively remove surrounding vehicles in nearest-first and farthest-first order, as shown in Figure 5 in 10,000 validation scenarios. We evaluate off-road rate, goal progress ratio, and vehicle-collision rate. Because removing vehicles mechanically reduces opportunities for collision, an adjusted collision rate is used, $\text{Veh-Coll}_{\text{adj}} = \text{Veh-Coll}/(1 - p)$, where p denotes the fraction of surrounding vehicles removed.

The two removal strategies produce a clear asymmetry. Farthest-first removal has little effect on goal progress through high removal ratios, indicating relative insensitivity to distant-agent context. In contrast, nearest-first removal increasingly degrades goal progress at high removal ratios, even as the adjusted vehicle-collision rate decreases. This suggests that the degradation reflects brittleness to the removal of nearby-agent context rather than increased interaction difficulty. Note that the

vehicle-collision rate increases under farthest-first removal because nearby vehicles are maintained, so the raw number of collisions remains almost the same while the total number of vehicles decreases.

We additionally analyze random removal and naturally occurring single-AV scenes in Appendix F.1. Natural single-AV scenes already perform worse than normal scenes, while removing vehicles from originally interactive scenes causes a further decline. Together, these results suggest both limited generalization to reduced-interaction settings and sensitivity to nearby-agent context, rather than a general spurious reliance on surrounding vehicles.

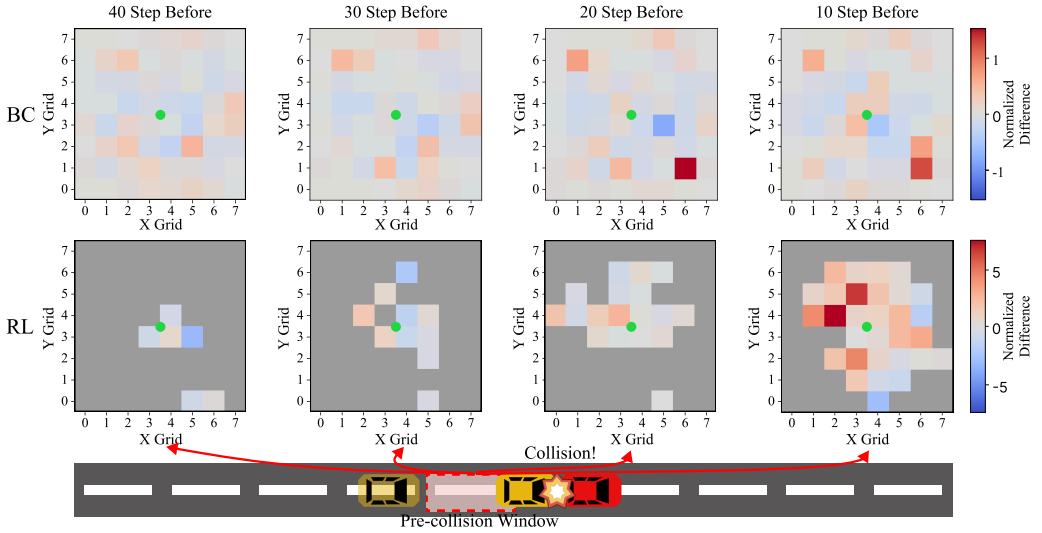


Figure 6: **Near-collision Analysis.** Heatmaps show the normalized change in surrounding-vehicle probing signal relative to the full-episode average, $(w - a_{\text{grid}})/a_{\text{grid}}$, at 40, 30, 20, and 10 steps before collision. Red denotes increased probing signal and blue denotes decreased signal; dark gray cells indicate insufficient collision samples. The grid is ego-centric and spans $[-50, 50]$ m along both axes. Top: BC; Bottom: RL.

Beyond the single AV setting, the perturbed simulation offers only a rough, indirect view of how the model handles surrounding vehicles. To examine how linearly decodable surrounding-vehicle information changes in safety-critical situations, we therefore conduct a near-collision analysis using the surrounding-vehicle linear probe. We define a pre-collision window W of 10 steps and compare the probing signal within this window against the model's average probing signal over the full validation episodes. For each spatial grid cell $grid$, we compute the normalized difference $(w - a_{grid})/a_{grid}$, where a_{grid} is the corresponding full-episode average. We filter out grid cells with fewer than 100 collision cases and mark them in dark gray.

Figure 6 visualizes this normalized difference at 40, 30, 20, and 10 steps before collision. Red grids denote spatial regions whose surrounding-vehicle probing signal increases relative to the whole episode average, whereas blue grids denote regions whose signal decreases. Therefore, early anticipation of a dangerous interaction would appear as a strong positive signal around the relevant nearby vehicle at earlier horizons, such as 40, 30, or 20 steps before the event. In contrast, for the BC model, the signal remains weak and diffuse at earlier horizons and becomes clearly concentrated near the ego only in the final 10 steps before collision. Although this late increase indicates that the nearby colliding vehicle becomes strongly linearly decodable, it is already too late for effective avoidance; therefore, these episodes remain planning failures.

The RL model shows a similar qualitative pattern, but collision cases are much rarer, resulting in substantially fewer samples and larger dark-gray filtered regions. Despite this sparsity, the strongest positive signal again appears mainly in the final 10-step window. Overall, both BC and RL appear to use surrounding-vehicle information near a collision. Still, the probing signal for the relevant vehicle becomes strong only shortly before collision to avoid the crash. We analyze near-collision events for off-road in Appendix F.2.

3.5 Intervention Test for Adaptiveness of Ego AV Planning

How does the model's planning change when it has more accurate predictions, and how does it adapt to changes in surrounding-vehicle predictions? To test this, we first apply linear probing to the ego AV's planning, as in Section 3.3 (which we refer to as *ego-linear probing*); results are in Appendix E. Inspired by Bush et al. [2025], we intervene on an earlier-layer surrounding-vehicle representation g_o using the linear-probe direction w_l^s associated with future-position label l. Let $g = [g_o;\, g_{-o}]$, where $g_o \in \mathbb{R}^{128}$ denotes the intervened representation and g_{-o} the remaining representations:

$$g'_o = g_o + \frac{\alpha}{|S|} \sum_{s \in S} w_l^s, \qquad g' = [g'_o;\, g_{-o}], \tag{1}$$

where α controls intervention strength. We then propagate the modified representation and decode the resulting ego plan:

$$h' = f(g'), \qquad p' = e(h'). \tag{2}$$

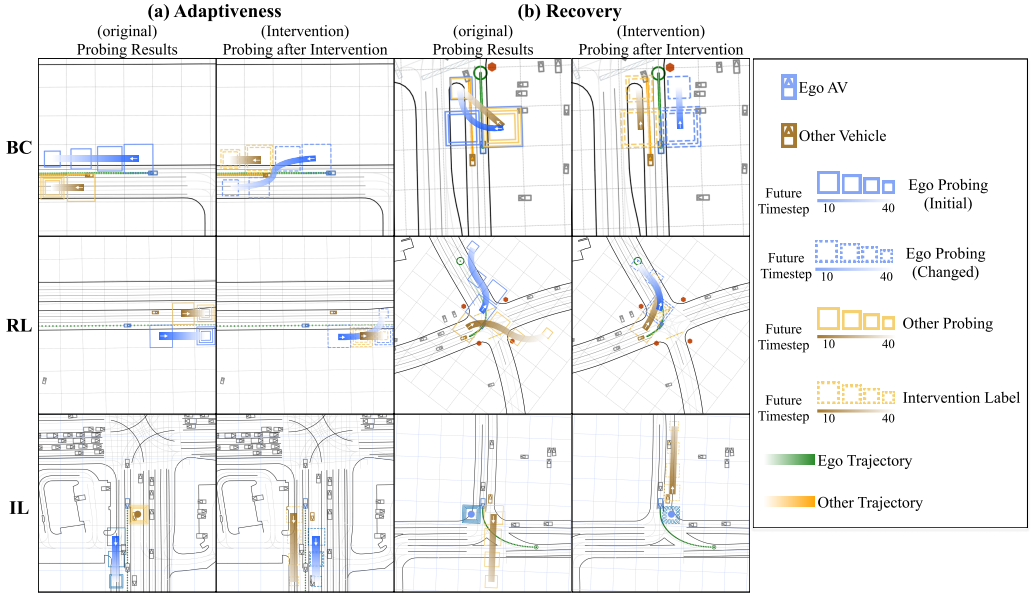


Figure 7: **Intervention experiment for adaptiveness and recovery (10 to 40 future timesteps)**: Examples from BC and RL models; each column pair compares the original probing result with the probing result after intervention. (a) **Adaptiveness**: we perturb the surrounding-vehicle prediction so that it overlaps with the ego's predicted path, and test whether the ego plan changes to avoid the induced conflict. (b) **Recovery**: we correct an erroneous surrounding-vehicle prediction and examine whether the ego plan moves toward the ground-truth trajectory.

In the intervention experiment, we define two settings. In *adaptiveness*, we induce surrounding-vehicle predictions that conflict with the ego's decoded path and test whether the ego moves away from the induced trajectory. In *recovery*, we correct erroneous predictions toward the ground-truth future and test whether ego goal progress improves. Both are compared against magnitude-matched random controls.

Adaptiveness. Let $\mathbf{x}_t^{\text{tgt}}$ denote the intervention-defined surrounding-vehicle trajectory and $\mathbf{x}_t^{\text{ego,base}}$ and $\mathbf{x}_t^{\text{ego,int}}$ the paired ego trajectories. We define

$$\text{CAP} = \frac{1}{H} \sum_{t=1}^{H} \left(\left\| \mathbf{x}_t^{\text{ego,int}} - \mathbf{x}_t^{\text{tgt}} \right\|_2 - \left\| \mathbf{x}_t^{\text{ego,base}} - \mathbf{x}_t^{\text{tgt}} \right\|_2 \right), \tag{3}$$

where positive CAP indicates increased separation from the induced trajectory.

For BC, the semantic intervention yields a mean CAP of 2.61 m, compared with 0.92 m for the random control, resulting in a 1.69 m gap across 1,433 samples. Ego trajectories also change more under

the semantic intervention, with an ADE of 3.67 m compared with 1.91 m under the random control, showing a directionally meaningful planning response beyond magnitude-matched perturbations.

For RL, the downstream effect is weaker. Mean CAP is 0.022 m under the semantic intervention and 0.010 m under the random control. Moreover, the semantic intervention induces an ego-trajectory ADE of 0.280 m, which is smaller than the 0.357 m induced by the random control. Nevertheless, the semantic intervention increases the probed prediction's match to the intervention label in 58.7% of scenes, compared with only 8.2% under the random control. Thus, the intervention alters the prediction-related representation in the intended direction but does not consistently propagate to ego planning. This weaker propagation may be related to max pooling in RL, whereas the attention-based BC architecture may propagate such changes more flexibly.

Recovery. We measure recovery by the change in goal progress:

$$\mathrm{GPG} = \mathrm{gp}^{\mathrm{int}} - \mathrm{gp}^{\mathrm{base}}. \tag{4}$$

For BC, across 5,023 scenes, the semantic intervention improves goal progress by 4.8 percentage points, while the random control yields only a 0.9-point improvement. The semantic advantage becomes more pronounced for severe failures with $\mathrm{gp}^{\mathrm{base}} \leq 0.5$, where the gap increases to 11.6 points. However, collision and off-road rates increase slightly, indicating improved recovery without uniform safety gains.

For RL, most scenes already achieve high baseline goal progress, limiting the overall recovery effect. Among low-progress scenes whose outcomes change, however, the semantic intervention improves GPG by 0.413, compared with 0.313 under the random control. This suggests that semantic interventions can facilitate recovery from severe RL failures, although their downstream effect remains weaker than in BC. Figure 7 shows representative examples across BC, RL, and IL. Additional examples, including IL and explicit success criteria, are provided in Appendix G.

4 Conclusion

In this paper, we investigate how driving models preserve and use surrounding-vehicle information for ego planning. As dataset size increases, probing signals become more selective for nearby over distant agents, while prediction- and planning-related information remains imperfect despite strong nominal closed-loop performance. Furthermore, our perturbed closed-loop evaluation reveals limited generalization to uncommon single-AV settings and additional brittleness to the removal of nearby-agent context. In particular, during near-collision events, safety-relevant prediction signals often become strongly linearly accessible only shortly before collision, suggesting that the models do not reliably capture critical interactions early enough. Finally, targeted interventions show that prediction-related representations are behaviorally relevant to ego planning. When surrounding-vehicle representations are perturbed toward potential conflicts, the ego increases its separation from the induced trajectory, while correcting erroneous predictions improves goal progress, particularly when the original policy makes little progress. However, these recovery interventions also incur modest increases in collision and off-road rates, indicating that better prediction does not uniformly translate into safer planning.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-16902996, RS-2026-25520248). This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2026-25614465, AI Star Fellowship Support Program, No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)). This research was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020535, The Competency Development Program for Industry Specialist). We appreciate the high-performance GPU computing support of HPC-AI Open Infrastructure via GIST SCENT. This work was supported by GIST-IREF from Gwangju Institute of Science and Technology (GIST). This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Mustafa Baniodeh, Kratarth Goel, Scott Ettinger, Carlos Fuertes, Ari Seff, Tim Shen, Cole Gulino, Chenjie Yang, Ghassen Jerfel, Dokook Choe, et al. Scaling laws of motion forecasting and planning–a technical report. *arXiv preprint arXiv:2506.08228*, 2025.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Thomas Bush, Stephen Chung, Usman Anwar, Adrià Garriga-Alonso, and David Krueger. Interpreting emergent planning in model-free reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99. PMLR, 2020.

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.

Daphne Cornelisse, Spencer Cheng, Pragnay Mandavilli, Julian Hunt, Kevin Joseph, Waël Doulazmi, Valentin Charraut, Aditya Gupta, Joseph Suarez, and Eugene Vinitsky. PufferDrive: A fast and friendly driving simulator for training and evaluating RL agents. `https://github.com/Emerge-Lab/PufferDrive`, 2025a. Version 2.0.0. Equal contribution by the first two authors.

Daphne Cornelisse, Aarav Pandya, Kevin Joseph, Joseph Suárez, and Eugene Vinitsky. Building reliable sim driving agents by scaling self-play. *arXiv preprint arXiv:2502.14706*, 2025b.

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.

Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 3567–3575. IEEE, 2023.

Arthur Guez, Mehdi Mirza, Karol Gregor, Rishabh Kabra, Sébastien Racanière, Théophane Weber, David Raposo, Adam Santoro, Laurent Orseau, Tom Eccles, et al. An investigation of model-free planning. In *International conference on machine learning*, pages 2464–2473. PMLR, 2019.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36:7730–7742, 2023.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022.

Hyeongseok Jeon, Sanmin Kim, Abi Rahman Syamil, Junsoo Kim, and Dongsuk Kum. Beyond the data imbalance: Employing the heterogeneous datasets for vehicle maneuver prediction. In *European Conference on Computer Vision*, pages 38–53. Springer, 2024.

Saman Kazemkhani, Aarav Pandya, Daphne Cornelisse, Brennan Shacklett, and Eugene Vinitsky. GPUDrive: Data-driven, multi-agent driving simulation at 1 million FPS. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=ERv8ptegFi`.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *2017 IEEE international conference on computer vision (ICCV)*, pages 2961–2969. IEEE, 2017.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873, 2024.

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.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

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*, 36:59151–59171, 2023.

Alexander Naumann, Xunjiang Gu, Tolga Dimlioglu, Mariusz Bojarski, Alperen Degirmenci, Alexander Popov, Devansh Bisla, Marco Pavone, Urs Muller, and Boris Ivanovic. Data scaling laws for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2571–2582, 2025.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023.

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.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024.

Liting Sun, Rebecca Roelofs, Ben Caine, Khaled S Refaat, Ben Sapp, Scott Ettinger, and Wei Chai. Causalagents: A robustness benchmark for motion forecasting. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6820–6827. IEEE, 2024.

Omer Sahin Tas and Royden Wagner. Words in motion: Extracting interpretable control vectors for motion transformers. In *International Conference on Learning Representations*, volume 2025, pages 87191–87214, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

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.

Yi Xiao, Felipe Codevilla, Christopher Pal, and Antonio Lopez. Action-based representation learning for autonomous driving. In *Conference on Robot Learning*, pages 232–246. PMLR, 2021.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

Yupeng Zheng, Zhongpu Xia, Qichao Zhang, Teng Zhang, Ben Lu, Xiaochuang Huo, Chao Han, Yixian Li, Mengjie Yu, Bu Jin, et al. Preliminary investigation into data scaling laws for imitation learning-based end-to-end autonomous driving. *arXiv preprint arXiv:2412.02689*, 2024.

A Model and Training Details

A.1 BC Details

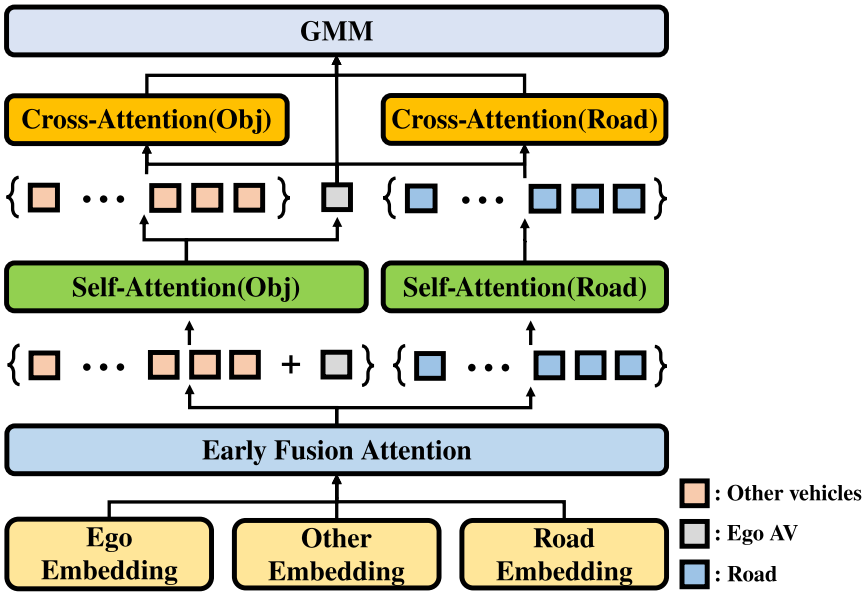


Figure 8: **Behavior Cloning model architecture**: Overall architecture of the behavior cloning model. Ego vehicle, other vehicles, and road features are first embedded and fused via early-fusion attention. The fused representations are refined via self- and cross-attention modules and finally modeled with a Gaussian Mixture Model (GMM).

In our behavior cloning framework, the model is conditioned on three types of inputs: ego vehicle state, other vehicles' state, and road context features in Table 2. Each input modality is first embedded into a latent representation through dedicated encoders:

$$h^{ego} = f_{ego}(x^{ego}), \tag{5}$$
$$h^{other} = f_{other}(x^{other}), \tag{6}$$
$$h^{road} = f_{road}(x^{road}) \tag{7}$$

where $f_{ego}, f_{other}, f_{road}$ denotes a scene encoder applied to the corresponding features, implemented as a multi-layer perceptron with layer normalization and non-linear activations:

$$z_1 = Wx + b, \quad \text{(Linear)} \tag{8}$$
$$z_2 = \text{Dropout}(z_1), \quad \text{(Dropout)} \tag{9}$$
$$z_3 = \text{LN}(z_2), \quad \text{(Layer Normalization)} \tag{10}$$
$$h = \phi(z_3), \quad \text{(Activation, e.g., Tanh)} \tag{11}$$

The embedded representations are first fused through an early fusion attention module, which enables the ego state to jointly attend to both road and object-level features:

$$H^{ego}, H^{other}, H^{road} = \text{Attn}([h^{ego}, h^{other}, h^{road}]). \tag{12}$$

To refine the fused representation in a modality-specific manner, self-attention layers are applied independently to object-level and road-level embeddings:

$$\tilde{H}^{ego}, \tilde{H}^{other} = \text{SelfAttn}([H^{ego}, H^{other}]), \tag{13}$$
$$\tilde{H}^{road} = \text{SelfAttn}(H^{road}). \tag{14}$$

To enhance modality-specific interaction and enable more informed policy learning, cross-attention layers are applied, allowing the ego representation to selectively attend to road-level and object-level

features:

$$\tilde{H}^{ego}_{other} = \text{CrossAttn}([\tilde{H}^{ego}, \tilde{H}^{other}]), \tag{15}$$

$$\tilde{H}^{ego}_{road} = \text{CrossAttn}([\tilde{H}^{ego}, \tilde{H}^{road}]). \tag{16}$$

Finally, the aggregated representation—obtained by concatenating the refined ego embedding with the outputs of the road- and object-conditioned cross-attention modules ($\tilde{H}^{ego}, \tilde{H}^{ego}_{other}, \tilde{H}^{ego}_{road}$) is fed into an output head parameterizing a Gaussian Mixture Model (GMM):

$$p(y \mid x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y \mid \mu_k, \Sigma_k), \tag{17}$$

where y denotes the predicted future action of the ego vehicle, and π_k, μ_k, Σ_k are the mixture weights, means, and covariances of the k-th Gaussian component, respectively. This probabilistic formulation enables the model to capture multimodal action distributions and inherent uncertainties in motion prediction. During training, the model parameters are optimized by minimizing the negative log-likelihood (NLL) of expert demonstrations under the predicted GMM distribution. Specifically, given an expert action y^*, the likelihood under the mixture model is

$$p(y^* \mid x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y^* \mid \mu_k, \Sigma_k), \tag{18}$$

and the loss function is defined as

$$\mathcal{L}_{\text{GMM}} = -\mathbb{E}_{(x,y^*)\sim\mathcal{D}} \left[\log p(y^* \mid x)\right], \tag{19}$$

where $\mathcal{D}$ denotes the dataset of expert demonstrations. This objective encourages the model to assign high probability density to expert actions, thereby aligning the predicted distribution with the expert policy.

A.2 BC Training Settings

Across all experiments of dataset size, the BC model has 1.4M parameters and is trained with a batch size of 512, a hidden layer size of 128, and a 6-component GMM head. We train with a learning rate of 0.0005 and apply a weight decay. The different settings across different dataset sizes are in Table 1. We used 4 A100 GPUs to train BC models.

Table 1: BC Training Details.

#Scenes	#Vehicles	#Samples	Grad. Steps	#Scenes	#Vehicles	#Samples	Grad. Steps
100	0.7K	44K	20,000	10000	68K	4.2M	400,000
500	3K	215K	50,000	20000	136K	8.4M	550,000
1000	7K	431K	100,000	40000	270K	16.9M	700,000
5000	33K	2.1M	250,000	80000	542K	33.8M	850,000

A.3 RL Training Settings

Across all experiments, we train a feed-forward IPPO policy [Schulman et al., 2017, Cornelisse et al., 2025b] with 1.2M parameters and a hidden dimension of 128. Training is performed with a batch size of 65,536 and a minibatch size of 4,096 for 2 update epochs, using a learning rate of 0.0003. We set the discount factor to 0.99, the GAE to 0.95, the clipping coefficient to 0.2, the entropy coefficient to 0.0001, and the value loss coefficient to 0.3. We scale the number of training steps with dataset size, using 0.1B, 0.2B, 0.3B, 0.75B, and 1B steps for 100, 500, 1,000, 5,000, and 10,000 scenes, respectively. We used 4 A100 GPUs to train RL models.

B Dataset Details

B.1 Observation Features

The observation space provides a multi-modal representation composed of three key features: *ego state features*, describing the ego vehicle's kinematics and goal-related information; *surrounding vehicle features*, encoding surrounding dynamic surrounding vehicles; *road graph features*, capturing static road topology and structural elements (road edge, road line, road lane, crosswalk, speed bump, stop sign, and None as padding). For all experiments, we set the maximum number of agents per scenario to 128, and we consider the nearest 200 road points. The detailed specifications of these observation features, including their type, constituent variables, dimensionality, and description, are summarized in Table 2.

Table 2: Observation features specifications.

Type	Feature	Dimension	Description
ego state	Speed	$(1,)$	Ego vehicle speed
	Size	$(2,)$	Vehicle length and width
	Goal position	$(2,)$	Relative goal coordinates (x,y)
	Collision state	$(1,)$	1 if collided, 0 otherwise
surrounding vehicle	Speed	$(\text{max num agents} - 1, 1)$	Partner vehicle speed
	Size	$(\text{max num agents} - 1, 2)$	Length and width
	Goal position	$(\text{max num agents} - 1, 2)$	Relative goal coordinates (x,y)
	Collision state	$(\text{max num agents} - 1, 1)$	1 if collided, 0 otherwise
road graph	Segment position	$(\text{top k road points}, 2)$	Road point coordinates
	Segment size	$(\text{top k road points}, 3)$	Length, width, and height
	Segment orientation	$(\text{top k road points}, 1)$	Orientation of the road segment
	Segment type	$(\text{top k road points}, 8)$	One-hot encoded road point type

B.2 Inverse Kinematics Model

B.2.1 Delta Dynamics Model

In our simulation environment, we employ a delta dynamics model [Gulino et al., 2023] to update the agent states. Unlike models that apply displacements directly in the global frame, our approach defines actions in the *local coordinate frame* of the agent, which aligns with its heading direction. At each timestep t, the action is represented as

$$a_t = (\Delta x_t, \Delta y_t, \Delta \psi_t), \tag{20}$$

where Δx_t and Δy_t denote the forward and lateral displacements relative to the agent's orientation, and $\Delta \psi_t$ is the change in yaw angle. To apply the action in the global coordinate frame, we rotate the local displacement using the current yaw ψ_t of the agent:

$$\begin{bmatrix} \Delta x_t^{\text{global}} \\ \Delta y_t^{\text{global}} \end{bmatrix} = R(\psi_t) \begin{bmatrix} \Delta x_t \\ \Delta y_t \end{bmatrix}, \quad \Delta \psi_t^{\text{global}} = \Delta \psi_t, \tag{21}$$

where $R(\psi_t)$ is the 2D rotation matrix. The global displacements are then added to the agent's current state (x_t, y_t, ψ_t) to obtain the updated trajectory $(x_{t+1}, y_{t+1}, \psi_{t+1})$. For inverse dynamics, the procedure is reversed: we compute the displacement between consecutive states in the global frame and project it back into the local frame using $R(-\psi_t)$. This formulation allows actions to be expressed relative to the agent's forward-facing direction, making them more natural for imitation learning and policy optimization tasks.

B.2.2 Bicycle Model

Following the kinematic bicycle model [Gulino et al., 2023], we define an agent's current state information as $s = (x, y, \theta, v_x, v_y)$, which includes the x, y positions in the coordinate space, the yaw angle θ, and the velocities in the X and Y directions. The action is defined as (a, κ), where a

denotes the longitudinal acceleration and κ the steering curvature. Let $v = \sqrt{v_x^2 + v_y^2}$ denote the speed magnitude.

For the inverse kinematics, given the state information of two consecutive states $s = (x, y, \theta, v_x, v_y)$ and $s' = (x', y', \theta', v'_x, v'_y)$, we estimate the acceleration a and steering curvature κ as

$$a = \frac{v' - v}{\Delta t}$$

$$\kappa = \frac{\theta' - \theta}{d}, \qquad d = \sqrt{(x' - x)^2 + (y' - y)^2},$$

where $v' = \sqrt{{v'_x}^2 + {v'_y}^2}$. This formulation provides a compact approximation of car-like motion for extracting action labels from logged trajectories.

B.3 Trajectory Types Distributions

Straight refers to trajectories in which changes in Δy and Δyaw remain below predefined thresholds. *Turn* denotes the presence of a contiguous interval with notable variations in Δy and Δyaw. *Reverse* captures cases where Δx is negative for at least half of the trajectory. *Uncategorized* encompasses all remaining trajectories that do not satisfy the criteria above. The thresholds and criteria employed for this categorization are empirically determined using domain knowledge and a preliminary inspection of the dataset. As shown in Figure 9, the trajectory distribution is highly imbalanced, with nominal scenarios—particularly *Uncategorized* and *Straight*—accounting for more than 80% of the data, whereas *Reverse* cases constitute less than 1%.

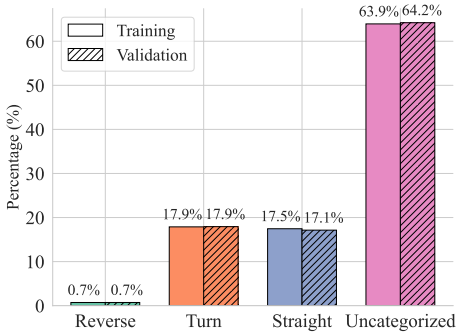


Figure 9: **Distributions of 4 trajectory types on training and validation set.**

To label the trajectory type, we extract the action values $(\Delta x, \Delta y, \Delta yaw)$ of whole trajectories. The training dataset has 542K samples, and the validation dataset has 69 K samples. Empirically, we first filtered out trajectories whose maximum lateral offset, $\max(\Delta y)$, exceeded 0.5 and whose maximum yaw change, $\max(\Delta\text{yaw})$, exceeded 0.2. We then labeled the remaining trajectories as follows: a trajectory was labeled **Straight** if the peaks of both Δy and Δyaw were below 0.01; **Turn** if the fraction of timesteps with $\Delta y > 0.035$ and $\Delta\text{yaw} > 0.025$ was at least 15% of the trajectory; **Reverse** if the proportion of timesteps with $\Delta x < -0.01$ was at least 50%; and **Uncategorized** otherwise.

C Additional Scaling Results

C.1 Training and Validation Scenes Results

Table 3: Driving performance metrics across dataset scales of BC model.

Num Scenes	Dataset	Goal Rate	Off-Road	Veh-Coll	Goal Progress Ratio
100	Training	0.320 ± 0.062	0.306 ± 0.050	0.125 ± 0.031	0.606 ± 0.149
	Validation	0.302 ± 0.034	0.352 ± 0.030	0.183 ± 0.011	0.561 ± 0.115
500	Training	0.568 ± 0.133	0.127 ± 0.052	0.092 ± 0.012	0.785 ± 0.092
	Validation	0.521 ± 0.111	0.156 ± 0.033	0.111 ± 0.018	0.745 ± 0.083
1000	Training	0.553 ± 0.038	0.074 ± 0.019	0.083 ± 0.025	0.780 ± 0.063
	Validation	0.543 ± 0.017	0.114 ± 0.011	0.107 ± 0.024	0.748 ± 0.072
5000	Training	0.572 ± 0.097	0.074 ± 0.014	0.070 ± 0.011	0.797 ± 0.019
	Validation	0.573 ± 0.093	0.086 ± 0.013	0.073 ± 0.007	0.795 ± 0.015
10000	Training	0.615 ± 0.167	0.058 ± 0.008	0.076 ± 0.020	0.808 ± 0.043
	Validation	0.615 ± 0.166	0.067 ± 0.008	0.073 ± 0.019	0.809 ± 0.041
20000	Training	0.710 ± 0.020	0.075 ± 0.031	0.062 ± 0.006	0.854 ± 0.039
	Validation	0.709 ± 0.018	0.077 ± 0.030	0.060 ± 0.002	0.857 ± 0.035
40000	Training	0.567 ± 0.163	0.067 ± 0.006	0.073 ± 0.034	0.836 ± 0.076
	Validation	0.566 ± 0.159	0.069 ± 0.010	0.071 ± 0.036	0.839 ± 0.073
80000 (Full)	Training	0.583 ± 0.033	0.066 ± 0.004	0.076 ± 0.014	0.808 ± 0.074
	Validation	0.587 ± 0.034	0.065 ± 0.003	0.073 ± 0.015	0.813 ± 0.072

Table 4: Driving performance metrics across dataset scales of the RL model.

Num Scenes	Dataset	Goal Rate	Off-Road	Veh-Coll	Goal Progress Ratio
100	Validation	0.491 ± 0.123	0.088 ± 0.034	0.091 ± 0.020	0.519 ± 0.119
500	Validation	0.743 ± 0.058	0.081 ± 0.010	0.069 ± 0.012	0.757 ± 0.067
1000	Validation	0.875 ± 0.038	0.077 ± 0.017	0.060 ± 0.008	0.885 ± 0.036
5000	Validation	0.861 ± 0.101	0.068 ± 0.011	0.040 ± 0.003	0.882 ± 0.072
10000	Validation	0.989 ± 0.005	0.034 ± 0.011	0.023 ± 0.005	0.986 ± 0.005

As shown in Table 3, the BC model improves consistently on almost all metrics up to 20,000 scenes, after which performance largely saturates. The generalization gap between the training and validation sets is also mostly resolved beyond 20,000 scenes. By contrast, Table 4 shows gradual improvement in all metrics for RL except the goal progress ratio. This indicates a weaker scaling trend than in BC, likely because the RL objective is more directly tied to destination-reaching behavior.

C.2 WOSAC metrics of Scaling Trends

WOSAC metrics are widely used to evaluate human likeness in autonomous driving. We therefore examine how these metrics change with data scale and whether their trends can be empirically described by a power-law fit. As shown in Figure 10, all five metrics exhibit strong correlations with data scale and power-law-like trends over the evaluated range. The four WOSAC sub-metrics—Realism Meta, Kinematic, Interactive, and Map-based—improve steadily up to approximately 10K scenes, after which the gains largely saturate. Our Realism Meta score approaches 0.7, and minADE reaches approximately 1.1, both within a reasonable range and close to the 2023 leaderboard. This saturation also appears in the BC surrounding-vehicle probing results, which show limited improvement beyond about 10K scenes. Finally, our random baseline matches that reported in

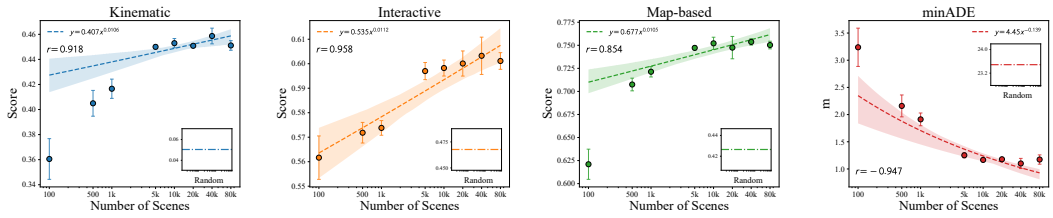
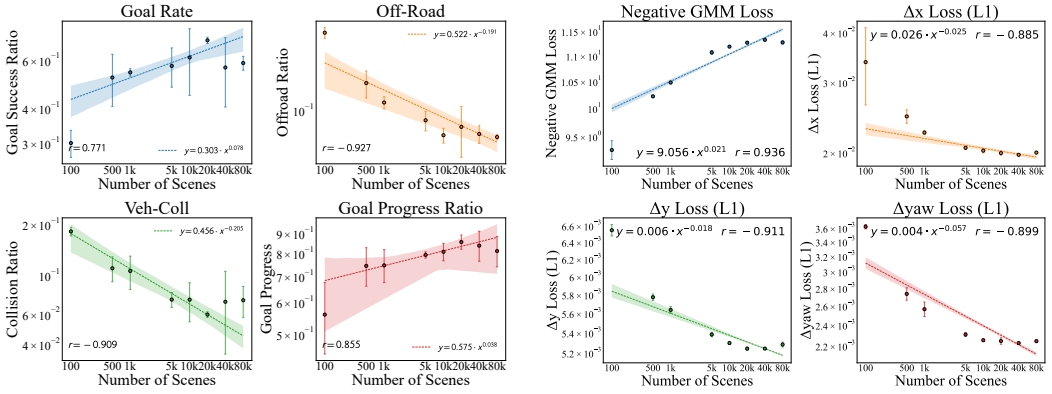


Figure 10: **Power-law-like scaling trends for WOSAC metrics**: We evaluate the realism meta score, kinematic score, interactive score, map-based score, and minADE for 1,000 scenes in the validation set.

PufferDrive [Cornelisse et al., 2025a], providing a reference for interpreting the absolute scale of these scores.

C.3 Additional results of Scaling Trends



(a) **Closed-loop scaling.** Goal/collision vs data; dashed: power-law fit; r: correlation.

(b) **Open-loop scaling.** GMM and action L1 losses vs data.

In this section, we present additional power-law-like scaling trends for closed-loop metrics in Figure 11a and open-loop metrics in Figure 11b. We further examine these trends across trajectory types in Figure 12. In most cases, the log–log correlation coefficient r is high, indicating that the power-law fit provides a useful empirical description of performance changes with data scale over the evaluated range. For *Reverse* trajectories, the metrics show larger variance due to the small number of samples, and no collisions are observed beyond 5,000 scenes.

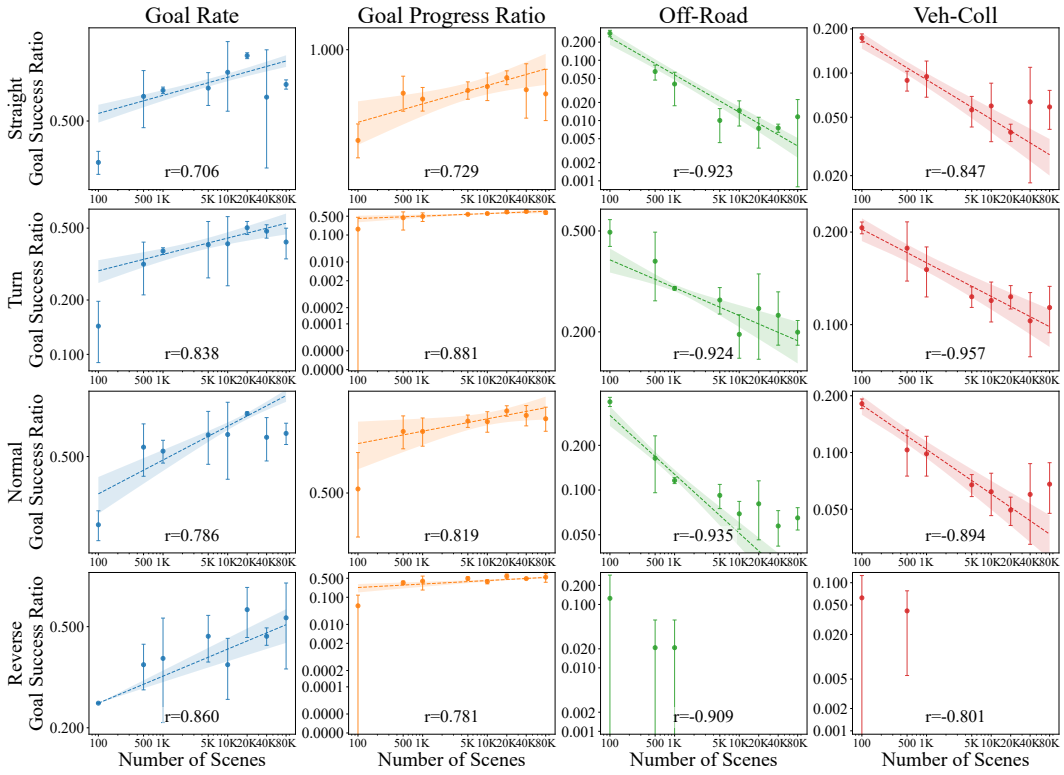


Figure 12: **Power-law-like scaling trends for simulation results by cases (BC)**: Closed-loop simulation results for BC with data scaling. Missing points correspond to zero-valued metrics.

D Additional Surrounding-vehicle probing results

D.1 Linear Probing Setting

Table 5: Linear Probing Setting Details for Surrounding-Vehicle Prediction.

#Scenes	Gradient (Step)	#Samples (fs@10)	#Samples (fs@20)	#Samples (fs@30)	#Samples (fs@40)
100	5K	235K	184K	140K	104K
500	12K	983K	768K	588K	437K
1000	20K	1.8M	1.4M	1.1M	824K
5000	50K	8.6M	6.7M	5.1M	3.8M
10,000	75K	17.6M	13.7M	10.5M	7.8M
20,000	100K	35.2M	27.4M	21M	15.7M
40,000	125K	70M	54.6M	41.9M	31.2M
80,000	150K	140.4M	110M	84.1M	62.7M

Across all experiments with varying dataset sizes, we train with a learning rate of 0.0015 and a batch size of 256. Table 5 shows the settings across dataset sizes. For SMART (IL), we trained the linear probe for 15,000 gradient steps with the learning rate of 0.001 and a batch size of 16. For the raw-input model, we use agent-centric features: position, heading, velocity, and type, with 11 historical steps.

D.2 Detailed Results of Linear Probing

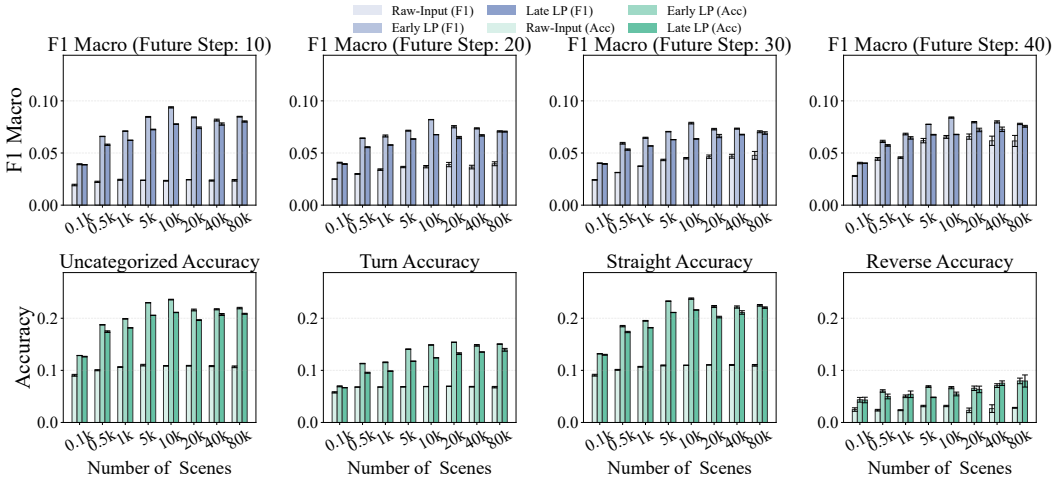


Figure 13: **Full performance metrics with surrounding-linear probing (BC)**: Light colors are raw-input probing and darker colors are BC probing(earlier layer and late layer) with error bars of standard deviation across different seeds. For convenience, we refer to linear probing in the earlier layer of the BC model as *early LP* and in the later layer as *late LP*. The top row is the F1 score across future step $\{10, 20, 30, 40\}$ and the bottom row is accuracy with labeled cases (Future step = 10).

In this section, we present the full linear probing results for BC and RL, shown in Figure 13 and Figure 14. Surrounding-vehicle information becomes more linearly accessible in BC as training scale increases, but its absolute predictive accuracy remains limited. Although both early and late LP consistently outperform the raw-input probe, the overall F1 score remains below 10%, indicating limited linear decodability of surrounding-vehicle futures. This weakness is also reflected in type accuracy: BC performs better on relatively easy cases such as *Straight* and *Uncategorized*, but degrades substantially on more complex cases such as *Turn* and *Reverse*, showing that future positions are more linearly decodable for common Straight and Uncategorized trajectories than for Turn and Reverse cases. Interestingly, as the dataset grows, BC probing gains concentrate more on the

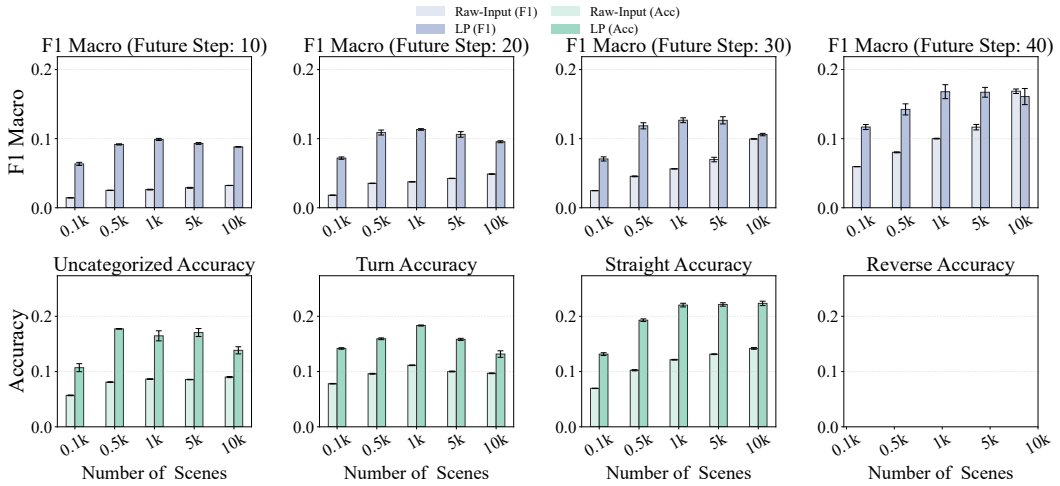


Figure 14: **Full performance metrics with surrounding-linear probing (RL)**: Light colors are raw-input probing and darker colors are RL probing with error bars of standard deviation across different seeds. For convenience, we refer to linear probing as *LP*. The top row is the F1 score across future steps $\{10, 20, 30, 40\}$, and the bottom row is accuracy with labeled cases (Future step = 10). Note that since there are no success cases of $Reverse$, so no probing results in $Reverse$.

near horizon, whereas the raw-input probe continues to improve at farther future steps. This suggests that the BC's probing gains become increasingly concentrated at shorter horizons. The widening gap between early and late LP at larger scales further indicates that near-future behavior becomes increasingly linearly accessible from deeper representations.

By contrast, the RL model achieves consistently higher F1 scores than BC across all future horizons, suggesting that it learns higher linear decodability of surrounding-vehicle futures. The fact that the raw-input probe is also stronger than in BC further suggests that self-play trajectories themselves are more predictable, indicating that RL agents may interact in a more mutually predictable manner. However, this advantage is concentrated on the near future. As the dataset grows, RL largely maintains its performance at the 10-step horizon, while showing weaker gains or slight degradation at longer horizons. This implies that the probing results become increasingly concentrated at short horizons, while long-horizon information becomes less linearly accessible. One possible explanation is that, unlike BC, which uses stacked observations over multiple past steps, RL operates on single-step inputs and therefore has less access to temporal context for longer-horizon prediction. Overall, these results suggest that RL exhibits stronger linear decodability of surrounding-vehicle information, particularly at short horizons.

Additionally, we introduce the additional results of linear probing as in the Table 6 and Table 7. Note that the RL model does not have a late LP, since it fuses at an earlier layer. From the Table 6, we report the F1 Macro score across all future timesteps. In Table 7, we report the full results of type accuracy.

As shown in Table 6, the BC model shows clear saturation after 10,000 scenes in both *Info Loss* and *LP - Raw*. Both values also decrease as the future step increases, indicating that the probing gain is concentrated on the near horizon. The positive *Info Loss* further suggests that surrounding-vehicle information is more linearly accessible from earlier layers than from later layers, likely in favor of other signals such as ego planning. By contrast, the RL model exhibits a much larger *LP - Raw* gap even at 100 scenes, indicating that surrounding-vehicle information becomes linearly accessible at smaller training scales than in BC. This may help explain why RL shows relatively stable collision behavior. However, this advantage is concentrated at short horizons, while longer-horizon gains shrink substantially as scale increases.

As shown in Table 7, the BC model exhibits increasingly larger type-wise probing gains with scale, especially for *Normal* and *Straight* cases, while *Turn* remains consistently more difficult and *Reverse* stays weak and unstable. The positive *Info Loss* across most types further suggests that surrounding-vehicle information is more linearly accessible in earlier than later layers. By contrast,

Table 6: F1 Macro differences across future steps. (a) BC, (b) RL. $Info\ Loss$ is (Early LP $-$ Late LP), and $LP - Raw$ is the difference between probing and the raw-input model. The RL table only has $LP - Raw$.

(a) BC

	FS=10		FS=20		FS=30		FS=40	
	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw
100	0.0005	0.0199	0.0013	0.0158	0.0007	0.0160	0.0001	0.0124
500	0.0080	0.0435	0.0086	0.0343	0.0061	0.0280	0.0039	0.0168
1000	0.0088	0.0467	0.0086	0.0323	0.0078	0.0272	0.0037	0.0226
5000	0.0121	0.0608	0.0080	0.0349	0.0077	0.0271	0.0099	0.0156
10000	0.0161	0.0704	0.0144	0.0451	0.0151	0.0337	0.0162	0.0185
20000	0.0099	0.0597	0.0103	0.0363	0.0066	0.0264	0.0074	0.0138
40000	0.0038	0.0579	0.0067	0.0371	0.0057	0.0265	0.0071	0.0181
80000	0.0046	0.0610	0.0002	0.0310	0.0012	0.0227	0.0023	0.0163

(b) RL

#Scenes	FS=10	FS=20	FS=30	FS=40
100	0.0490	0.0537	0.0459	0.0573
500	0.0665	0.0735	0.0731	0.0621
1000	0.0726	0.0757	0.0703	0.0679
5000	0.0641	0.0637	0.0567	0.0505
10000	0.0558	0.0468	0.0063	-0.0077

Table 7: Accuracy differences by action types. (a) BC at future step 10, (b) RL at future step 10. $Info\ Loss$ is (Early LP $-$ Late LP), and $LP - Raw$ is the difference between probing and the raw-input model.

(a) BC (Future step = 10)

	Normal		Turn		Straight		Reverse	
	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw
100	0.0021	0.0380	0.0025	0.0116	0.0020	0.0411	0.0004	0.0188
500	0.0132	0.0872	0.0176	0.0451	0.0112	0.0838	0.0104	0.0368
1000	0.0173	0.0925	0.0169	0.0474	0.0132	0.0882	-0.0039	0.0306
5000	0.0242	0.1198	0.0231	0.0722	0.0217	0.1233	0.0207	0.0373
10000	0.0247	0.1271	0.0246	0.0798	0.0218	0.1279	0.0121	0.0352
20000	0.0194	0.1070	0.0218	0.0845	0.0203	0.1120	0.0022	0.0424
40000	0.0100	0.1090	0.0128	0.0793	0.0104	0.1111	-0.0047	0.0488
80000	0.0109	0.1127	0.0110	0.0826	0.0044	0.1151	0.0002	0.0516

(b) RL (Future step = 10)

	Normal	Turn	Straight	Reverse
	LP - Raw	LP - Raw	LP - Raw	LP - Raw
100	0.0501	0.0638	0.0619	–
500	0.0965	0.0633	0.0905	–
1000	0.0783	0.0720	0.0988	–
5000	0.0850	0.0581	0.0900	–
10000	0.0484	0.0348	0.0815	–

RL shows relatively strong *LP - Raw* gains even at small scales, indicating earlier emergence of useful surrounding-vehicle representations. However, unlike in BC, these gains do not increase monotonically with scale; instead, they weaken at 10K scenes for several types. Overall, BC improves more steadily with data, whereas type-specific surrounding-vehicle information becomes linearly accessible at smaller scales in RL but does not sustain the same scaling trend.

D.3 Type Accuracy of Future Steps

Figure 15 shows the corresponding results for surrounding-vehicle probing. Here, accuracies are overall lower than in the ego case, and the gap between early and late LP is smaller and less stable, especially for $Turn$ and $Reverse$. However, there is no tendency for type accuracy in RL as in 16.

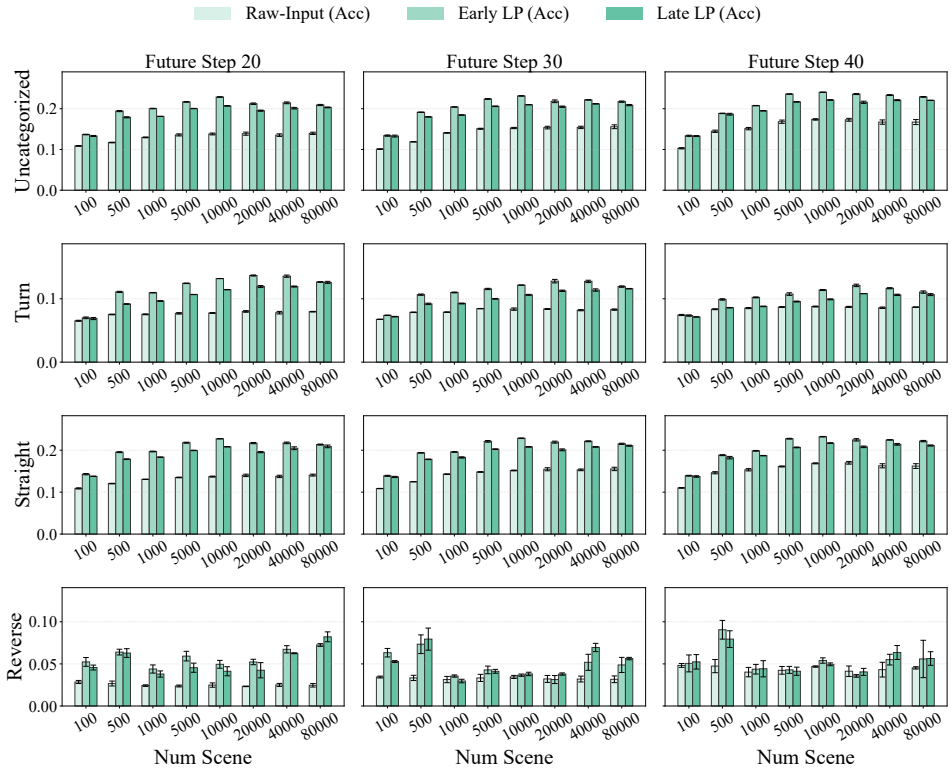


Figure 15: **Type accuracy of other future steps (20, 30, 40) (Other) (BC)**: The row is type ($Normal$, $Turn$, $Straight$, and $Reverse$) and the column is future step (20, 30, 40).

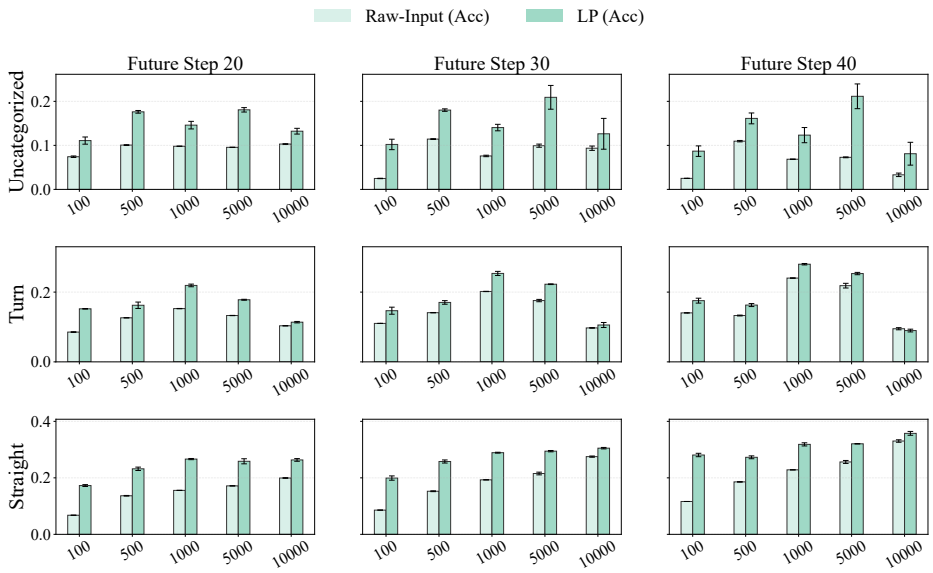


Figure 16: **Type accuracy of other future steps (20, 30, 40) (Other) (RL)**: The row is type ($Normal$, $Turn$, $Straight$, and $Reverse$) and the column is future step (20, 30, 40).

E Additional Ego probing results

E.1 Linear Probing Setting with Data Scaling

Table 8: Linear Probing Setting Details for Ego Prediction.

#Scenes	Gradient (Step)	#Samples (fs@10)	#Samples (fs@20)	#Samples (fs@30)	#Samples (fs@40)
100	5K	37K	31K	26K	20K
500	12K	181K	152K	125K	99K
1000	20K	355K	297K	244K	193K
5000	50K	1.7M	1.4M	1.2M	937K
10,000	75K	3.5M	2.9M	2.3M	1.9M
20,000	100K	7M	5.8M	4.8M	3.8M
40,000	125K	13.9M	11.6M	9.5M	756K
80,000	150K	27.7M	23.2M	19M	15.1M

Across all experiments with varying dataset sizes, we train with a learning rate of 0.0015 and a batch size of 256. The different settings across different dataset sizes are in Table 8.

E.2 Detailed Results of Linear Probing

In this section, we report the full results of ego AV linear probing as in Figure 17 and 18. The overall F1 Macro score exceeds 50%, indicating that future ego positions are strongly linearly decodable from the learned representation. As the dataset size increases, later-layer probes outperform early-layer probes, indicating that future ego-position information is more linearly accessible in deeper representations. This trend contrasts with the linear probing of surrounding vehicles, where early LP probes surpass late LP probes (early LP $>$ late LP). Interestingly, future ego positions in Reverse cases become strongly linearly decodable, even though they are not reliably decoded from raw inputs.

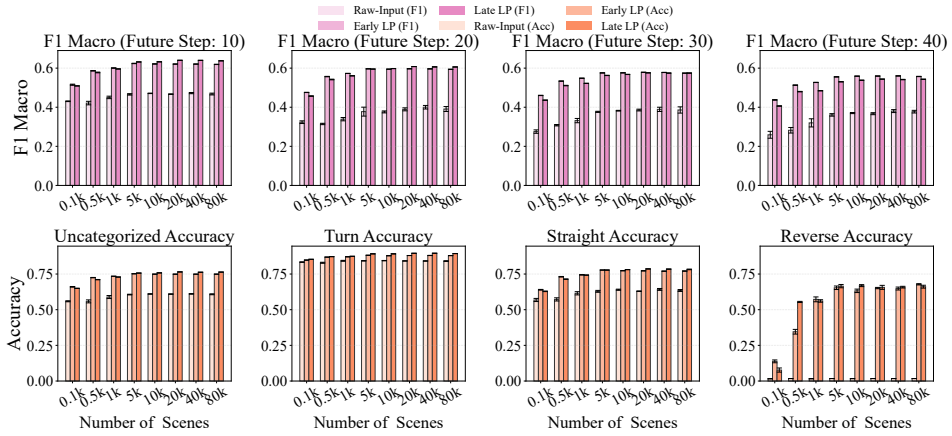


Figure 17: **Performance metrics with ego AV linear probing (BC)**: Light colors are raw-input probing and darker colors are BC probing (earlier layer and late layer) with error bars of standard deviation across different seeds. The top row is the F1 score across future steps $\{10, 20, 30, 40\}$ and the bottom row is accuracy with labeled cases (Future step = 10).

E.3 Type Accuracy of Future Steps

Figure 19 summarizes the accuracy of ego linear probing across trajectory types ($Uncategorized$, $Turn$, $Straight$, $Reverse$), future steps (20, 30, 40), and dataset scales. Across all types and future steps, both early and late LP substantially outperform the raw-input baseline. The gains are most

Table 9: F1 Macro differences across future steps. (a) BC, (b) RL. $Info\ Loss$ is (Early LP $-$ Late LP), and $LP - Raw$ is the difference between trained model probing and the raw-input model.

(a) BC

	FS=10		FS=20		FS=30		FS=40	
	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw
100	0.0060	0.0847	0.0186	0.1522	0.0241	0.1849	0.0312	0.1784
500	0.0086	0.1652	0.0150	0.2425	0.0226	0.2255	0.0330	0.2307
1000	0.0047	0.1502	0.0122	0.2339	0.0262	0.2167	0.0424	0.2068
5000	-0.0078	0.1659	0.0012	0.2190	0.0130	0.1992	0.0251	0.1946
10000	-0.0109	0.1615	-0.0030	0.2218	0.0082	0.1936	0.0210	0.1888
20000	-0.0192	0.1731	-0.0113	0.2188	0.0026	0.1930	0.0158	0.1925
40000	-0.0190	0.1674	-0.0100	0.2063	0.0032	0.1887	0.0186	0.1799
80000	-0.0185	0.1701	-0.0121	0.2151	-0.0004	0.1888	0.0143	0.1800

(b) RL

#Scenes	FS=10	FS=20	FS=30	FS=40
100	0.2207	0.3234	0.2487	0.1218
500	0.2872	0.3494	0.2616	0.2799
1000	0.1998	0.2476	0.2240	0.1567
5000	0.1422	0.1859	0.1823	0.1801
10000	0.2893	0.3191	0.3394	0.2771

Table 10: Accuracy differences by action types. (a) BC at future step 10, (b) RL at future step 10. $Info\ Loss$ is (Early LP $-$ Late LP), and $LP - Raw$ is the difference between model probing and the raw-input model.

(a) BC (Future step = 10)

	Normal		Turn		Straight		Reverse	
	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw	Info Loss.	LP - Raw
100	0.0109	0.1016	-0.0056	0.0196	0.0113	0.0717	0.0610	0.1219
500	0.0141	0.1666	-0.0036	0.0431	0.0172	0.1589	-0.2088	0.5369
1000	0.0054	0.1454	-0.0031	0.0314	0.0022	0.1296	0.0113	0.5561
5000	-0.0047	0.1505	-0.0102	0.0488	0.0001	0.1492	-0.0115	0.6478
10000	-0.0090	0.1479	-0.0130	0.0476	-0.0077	0.1411	-0.0365	0.6517
20000	-0.0159	0.1554	-0.0156	0.0534	-0.0132	0.1563	-0.0049	0.6392
40000	-0.0136	0.1517	-0.0155	0.0541	-0.0142	0.1423	-0.0097	0.6406
80000	-0.0145	0.1556	-0.0141	0.0527	-0.0115	0.1477	0.0166	0.6605

(b) RL (Future step = 10)

	Normal	Turn	Straight	Reverse
	LP - Raw	LP - Raw	LP - Raw	LP - Raw
100	0.0712	0.1789	0.2450	–
500	0.1996	0.2314	0.1587	–
1000	0.1269	0.0982	0.1205	–
5000	0.0353	0.0844	0.0604	–
10000	0.1562	0.3351	0.0744	–

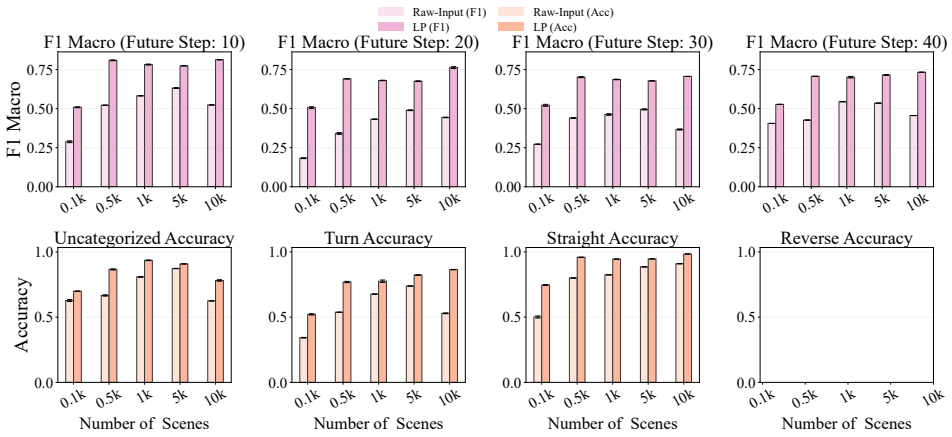


Figure 18: **Performance metrics with ego AV linear probing (RL)**: Light colors are raw-input probing and darker colors are RL probing with error bars of standard deviation across different seeds. The top row is the F1 score across future steps $\{10, 20, 30, 40\}$ and the bottom row is accuracy with labeled cases (Future step = 10).

significant for the $Straight$ and $Reverse$ cases for BC, and performance gradually improves, then saturates as the number of scenes increases. At the same time, accuracy consistently declines as the future horizon lengthens. The figure 15 shows the corresponding results for surrounding-vehicle probing. Here, accuracies are overall lower than in the ego case, and the gap between early and late LP is smaller and less stable, especially for $Turn$ and $Reverse$.

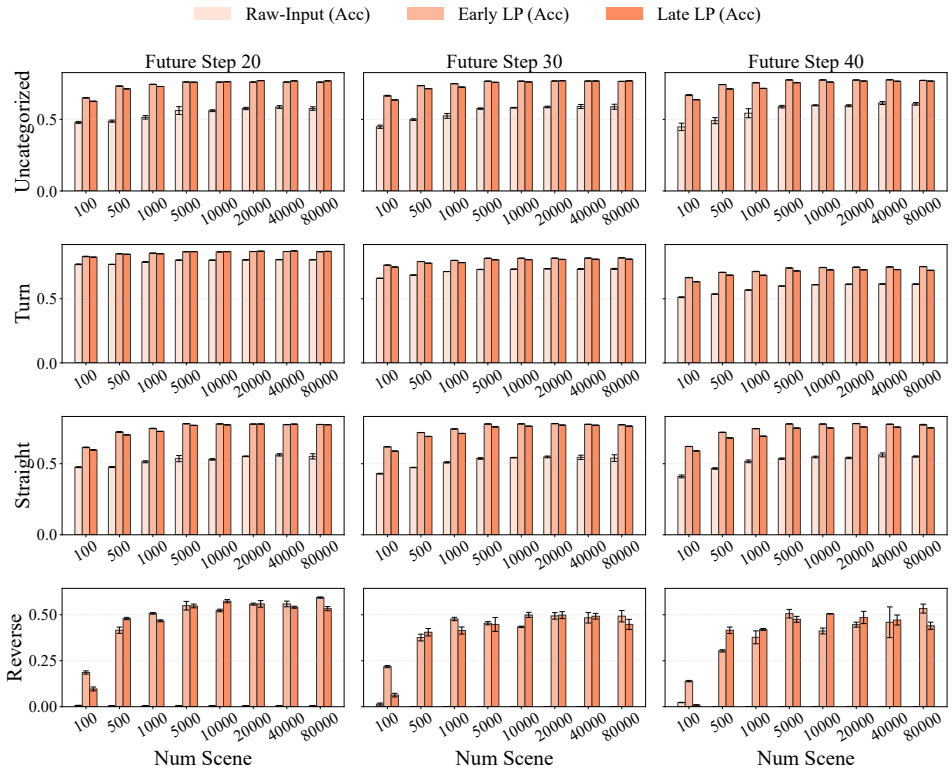


Figure 19: **Type accuracy of other future steps (20, 30, 40 (Ego) (BC)**: The row is type ($Uncategorized$, $Turn$, $Straight$, and $Reverse$) and the column is future step (20, 30, 40).

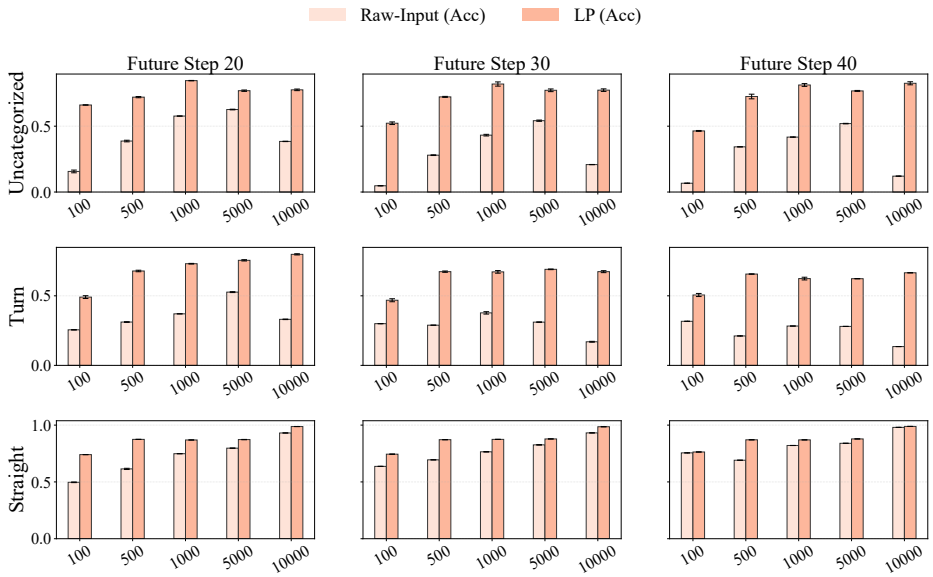


Figure 20: **Type accuracy of other future steps (20, 30, 40 (Ego) (RL)**: The row is type ($Uncategorized$, $Turn$, and $Straight$) and the column is future step (20, 30, 40).

F Further Analysis

F.1 Additional Perturbed Simulation

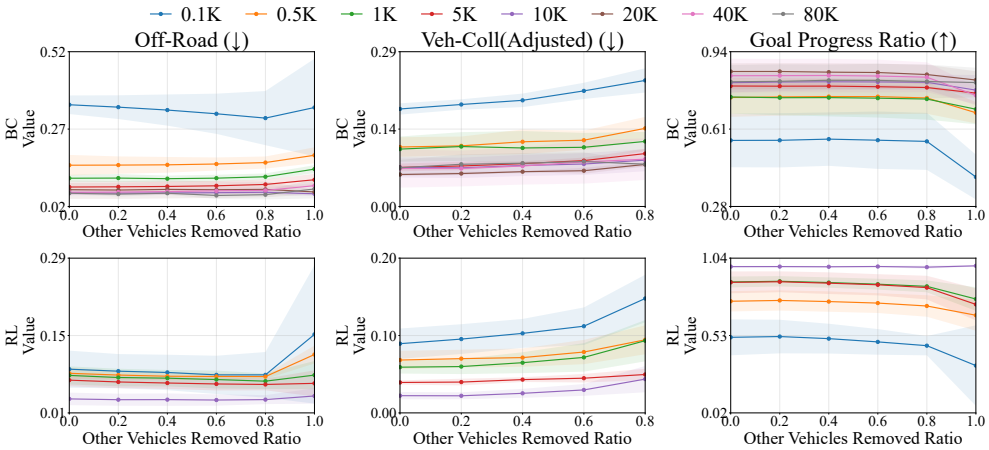


Figure 21: **Perturbed simulation results**: We test the BC model and RL model by randomly removing the surrounding vehicles with a ratio p. We adjust the ratio by multiplying $\frac{1}{ratio}$ for vehicle collision.

We provide additional controls to distinguish sensitivity to removed vehicle context from limited generalization to uncommon reduced-interaction settings. Single-AV observations are uncommon but not absent: approximately 1.5–1.8% of trajectories contain no surrounding vehicles for the full episode, while about 5–6% of RL single-step observations and 4–5% of BC five-frame observation windows contain no partners as in Table 12.

Because random removal alone cannot separate perturbation effects from the rarity of single-AV inputs, we compare normal, naturally occurring single-AV, and perturbed single-AV scenes in Table 11. As in Table 11, goal progress for BC decreases from $56.1 \pm 11.5\%$ to $47.8 \pm 5.1\%$ and $40.4 \pm 9.6\%$, respectively, while RL decreases from $51.9 \pm 11.9\%$ to $39.7 \pm 29.6\%$ and $33.2 \pm 26.6\%$. Natural single-AV scenes therefore already perform worse than normal scenes, while perturbing originally interactive scenes causes a further decline. Off-road differences between natural and perturbed single-AV scenes are small relative to their variance.

We also report random-removal results in Figure 21. Goal progress generally decreases as scenes approach the single-AV setting, with smaller degradation at larger data scales. Together with the distance-ordered results in Section 3.4, these controls suggest both limited generalization to reduced-interaction settings and brittleness to removed nearby-agent context, rather than a general spurious reliance on surrounding vehicles.

Table 11: Performance under normal, naturally occurring single-AV, and perturbed single-AV settings. Values are mean ± standard deviation.

Method	Metric	Normal	Natural Single-AV	Perturbed Single-AV
BC	Goal Progress (%)	56.1 ± 11.5	47.8 ± 5.1	40.4 ± 9.6
BC	Off-Road (%)	35.2 ± 3.0	36.7 ± 17.1	34.3 ± 15.8
RL	Goal Progress (%)	51.9 ± 11.9	39.7 ± 29.6	33.2 ± 26.6
RL	Off-Road (%)	8.8 ± 3.4	16.9 ± 14.2	15.2 ± 12.6

Table 12: Frequency of single-AV observations across dataset scales.

# Scenes	Full Episode	1-Step	5-Step
100	1.80%	6.16%	5.14%
1K	1.64%	5.03%	4.21%
10K	1.74%	4.96%	4.15%
40K	1.73%	5.04%	4.22%
80K	1.71%	5.03%	4.22%

F.2 Additional Near-Collision Analysis

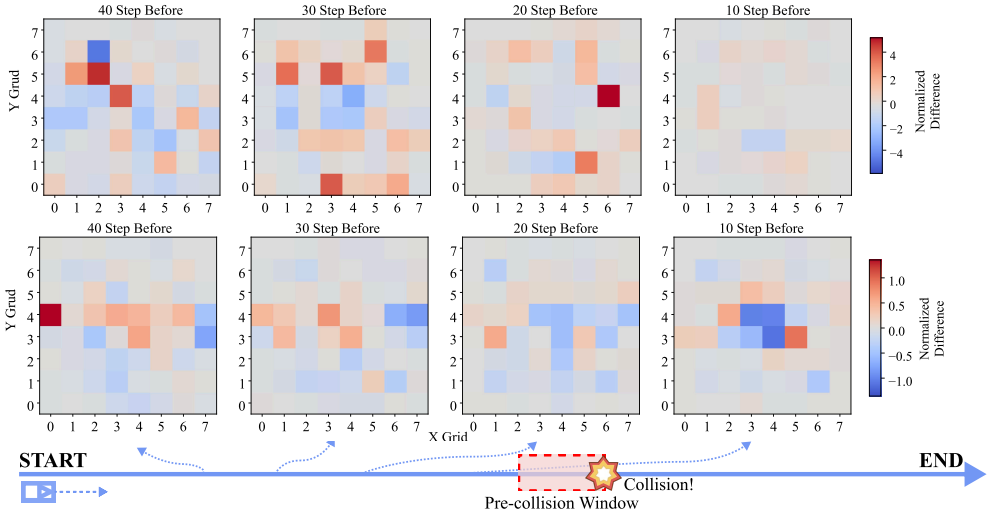


Figure 22: **Near-collision Analysis. (Off-road)** Each heatmap shows the normalized difference between the predicted probability in the pre-collision window w and that of each spatial grid over the full episode, computed as $\frac{w-a_{\text{grid}}}{a}$, at 10 to 40 steps before collision. The red dot marks the ego vehicle position. Dark gray cells indicate grids that were filtered out due to insufficient collision samples. Top: RL, Bottom: BC

G Intervention Results

Table 13: Intervention outcomes for adaptiveness and recovery. Success denotes $\mathrm{CAP} > 0$ for adaptiveness and $\mathrm{GPG} > 0$ for recovery. Mean reports CAP (m) or GPG. Random denotes the norm-matched control.

Model	Intervention	Case	Success	Total	Mean
BC	Semantic	Adaptiveness	1,301 (90.8%)	1,433	$+2.606$ m
	Random	Adaptiveness	1,036 (72.3%)	1,433	$+0.916$ m
	Semantic	Recovery	3,132 (62.4%)	5,023	$+0.048$
	Random	Recovery	2,762 (55.0%)	5,023	$+0.009$
RL	Semantic	Adaptiveness	111 (4.9%)	2,284	$+0.022$ m
	Random	Adaptiveness	156 (6.8%)	2,284	$+0.010$ m
	Semantic	Recovery	88 (11.3%)	778	$+0.007$
	Random	Recovery	111 (14.3%)	778	$+0.010$

Table 14: Intervention effects across different baseline conditions. For adaptiveness, close and far split scenes by the median baseline distance to the intervention-defined partner trajectory. Relative gap is the CAP gap normalized by the baseline distance. For recovery, we report GPG among scenes whose outcomes change, stratified by baseline goal progress.

(a) BC: Adaptiveness

Subset	n	CAP gap	Relative gap
Close	717	$+1.13$ m	$+10.6\%$
Far	716	$+2.25$ m	$+7.9\%$
All	1,433	$+1.69$ m	$+8.7\%$

(b) RL: Adaptiveness

Subset	n	CAP gap	Relative gap
Close	1,142	-0.032 m	-0.39%
Far	1,142	$+0.054$ m	$+0.24\%$
All	2,284	$+0.011$ m	$+0.07\%$

(c) BC: Recovery

	Semantic		Random	
gp_0	n	GPG	n	GPG
$[0, .5]$	431	$+.218$	430	$+.099$
$(.5, .9]$	2,323	$+.059$	2,323	$+.012$
$(.9, 1]$	2,256	$+.004$	2,256	$-.012$

(d) RL: Recovery

	Semantic		Random	
gp_0	n	GPG	n	GPG
$[0, .5]$	29	$+.413$	39	$+.313$
$(.5, .9]$	28	$-.038$	33	$+.016$
$(.9, 1]$	126	$-.042$	129	$-.037$

We report intervention results under the adaptiveness and recovery criteria in Table 13. Labels are constructed from linear-probe futures of the ego and nearby partners at 10, 20, 30, and 40 steps, considering up to the three closest vehicles within 50 m.

Adaptiveness considers scenes in which a nearby partner's predicted future overlaps the ego plan. We intervene by shifting the partner prediction toward the ego plan, provided that the target differs sufficiently from its original prediction. *Recovery* considers scenes in which both the ego and a nearby partner are incorrectly predicted relative to their logged futures. We then intervene on an eligible partner toward its ground-truth future. Scenes satisfying both criteria are assigned uniformly to one category, while those satisfying neither remain unlabeled.

We evaluate whether the intervention changes the probed prediction and, more importantly, whether it induces a beneficial behavioral response. For adaptiveness, success is defined as $\mathrm{CAP} > 0$, indicating that the intervened ego moves farther from the intervention-defined partner trajectory. For recovery, success is defined as $\mathrm{GPG} > 0$, where $\mathrm{GPG} = \mathrm{gp}^{\mathrm{int}} - \mathrm{gp}^{\mathrm{base}}$, indicating improved goal progress after intervention.

Table 15: Intervention diagnostics for adaptiveness. Ego LP changed measures whether the probed prediction changes after intervention, and label match measures whether it changes toward the intervention-defined label.

Model	Intervention	Ego LP changed	Label match ↑	Ego moved
BC	Semantic	100.0%	78.3%	100.0%
	Random	99.8%	10.7%	100.0%
RL	Semantic	93.1%	58.7%	10.7%
	Random	70.7%	8.2%	13.2%

Table 16: Open-loop intervention results for IL on 10,000 validation scenes. CAP proxy is computed from linear-probe cell centers and does not represent a closed-loop trajectory response.

Intervention	Ego LP changed	Label match	Mean CAP	CAP > 0	CAP $= 0$
Semantic	42.0%	56.3%	$+5.84$ m	70.0%	28.8%
Random	41.9%	0.8%	-0.12 m	34.2%	29.4%

For BC and RL, recovery is evaluated using GPG from closed-loop simulation. We do not report GPG for IL because it is an open-loop forecaster without a navigation goal or policy rollout.

IL intervention. We evaluate the IL model (SMART) on 10,000 validation scenes at the same four prediction horizons. Since IL does not produce closed-loop trajectories, we use an open-loop CAP proxy computed from the ego linear-probe cells. Specifically, $\mathrm{CAP}^{\mathrm{proxy}}$ measures the change in the ego probe's distance from the intervention-defined partner cell after intervention, averaged across the four horizons. A positive value indicates that the ego probe moves farther from the intervention-defined cell.

As shown in Table 16, the semantic intervention yields a mean CAP proxy of $+5.84$ m with a 70.0% positive-CAP rate, while the random control yields -0.12 m with a 34.2% positive-CAP rate. The label-match rate is also substantially higher under the semantic intervention, reaching 56.3% compared with 0.8% under the random control. Because the CAP proxy is computed from probe cells rather than closed-loop trajectories, it reflects a directional change in the probe readout rather than behavioral adaptation.

G.1 Failure Case

We analyze the failure cases, as illustrated in Figure 23, where there is no change compared to the initial prediction of ego planning. Interestingly, most cases increase in speed. As in cases (a) and (d), we make the front vehicles go faster so that the ego vehicle can increase its speed. However, the BC model remains on the same grid as before. Especially in case (d), even if we move multiple vehicles that could affect the ego's future position, the BC model remains in the same grid. In the case of (b) and (c), we increase the following vehicles' speed so that the ego prediction should speed up. However, there is no change in the BC model. In contrast, the RL policies have difficulty changing lanes, as seen in (e), (f), and (g). In (h), we expect the RL policy to increase speed, but it fails to do so. Lastly, in the IL model, as shown in (i), the ego vehicle remains in the grid before intervention, and the other vehicles' predictions are completely wrong. We restore the other vehicle to move in the correct direction, but the ego planner changes direction incorrectly. The same situation occurs in (j); we correct the vehicle prediction, but suddenly, ego planning goes the wrong way.

G.2 Additional Intervention Cases

As shown in Figure 24, panels (a)–(d) present additional BC adaptiveness cases, while panels (e)–(h) present RL adaptiveness cases. For the BC cases, panels (a) and (c) illustrate change lane interventions. In (a), the intervention induces a lane change through a single interaction with the yellow vehicle. In (c), the red vehicle follows from behind while the yellow vehicle changes lanes, requiring the

ego to maintain speed and change lanes as well; the BC model successfully performs this behavior. Panels (b) and (d) show slowdown interventions. In (b), when the yellow lead vehicle is kept in the ego's future grid cells, the ego remains behind it and slows its progress. In (d), we place multiple surrounding-vehicle trajectories in front of the ego, and the BC model responds by slowing down accordingly.

For the RL cases, panels (e) and (g) show change lane interventions, whereas panels (f) and (h) show slowdown interventions. In (e), the yellow and red vehicles originally occupy implausible positions, but when they are placed on the ego's predicted path, the ego changes its trajectory to avoid them. In (f) and (h), we modify previously implausible surrounding-vehicle trajectories to make them more feasible, and the RL model correspondingly reduces speed. Finally, in (g), after aligning an initially unrealistic trajectory, the ego's predicted motion is adjusted so that it no longer overlaps in time with the intervened vehicle.

For the IL cases, panels (i) and (k) show that the ego vehicle successfully adjusts its planning when we intervene in the surrounding-vehicle prediction to align with the original planning. In the cases of (j) and (l), we intervene in the ego's planning to stay on a further trajectory, so we expect the ego to change plans to slow down. The ego plans successfully to slow down to avoid a collision.

G.3 Additional Recovery Case

As shown in Figure 25, we describe the additional results of recovery cases. To evaluate the correction of the recovery label, we visualize the correctly recovered cases, which we label based on the surrounding vehicle trajectories and the random labels of other vehicles. In panel (a), the original prediction of the yellow vehicle is totally wrong, and the red vehicle is slightly wrong at 30 and 40 steps after. Therefore, we recover the red and yellow vehicle predictions to straighten and verify that the ego vehicle increases its speed slightly faster than before. Interestingly, when we assign a random label, the ego-vehicle prediction is the same as the initial prediction. Panel (b) examples show that the other vehicle prediction overlaps, so the ego prediction needs to avoid other vehicles. Therefore, we change the other vehicle's prediction along a straight line and show that the ego prediction changes in the same way, whereas random labeling tends to stay in the same grid for longer. In panel (c), which depicts the RL model's intervention, we recover the other vehicle's trajectory to follow the ego vehicle. Then, the ego vehicle adjusts to stay at the destination and avoid overwhelming it, consistent with the corrected prediction that the yellow vehicle will not overtake the ego vehicle, while the random label has the same planning as before the intervention. In panel (d), the RL completely mispredicts the vehicle in front. When we recover that it is feasible, the model did not predict the off-road area in 20 timesteps after. However, the random label indicates the wrong destination after 40 timesteps. Lastly, in (e), we recover the ego's trajectory as we approach a nearby vehicle to go straight. However, when we intervene randomly, the ego plan becomes corrupted and zigzags.

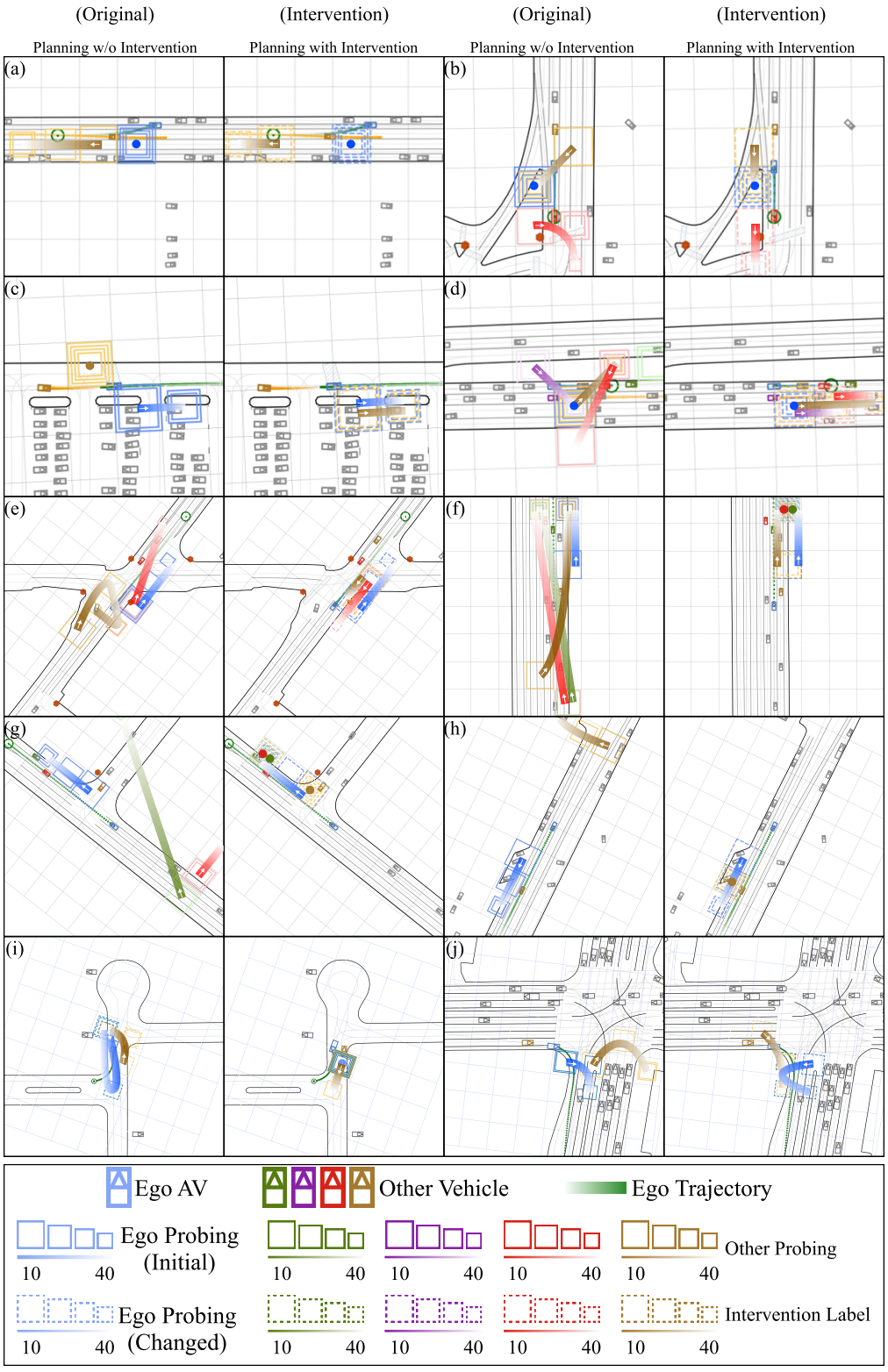


Figure 23: **Failure cases of intervention**: The visualization examples for failure cases of changing labels. (a)-(d): BC, (e)-(h): RL, (i)-(j): IL

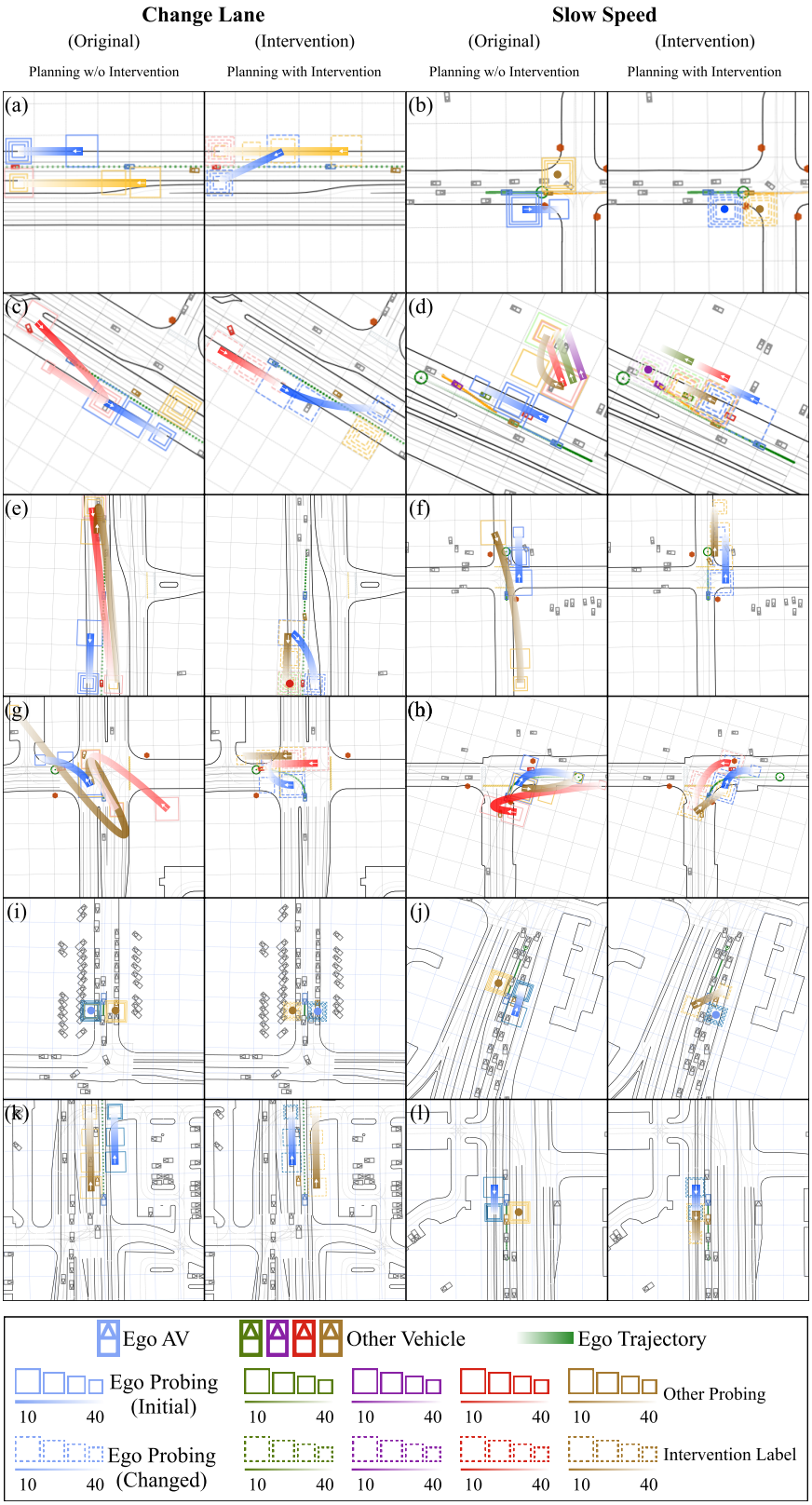


Figure 24: **Additional Examples for Adaptiveness cases**: The visualization examples for adaptiveness cases. **Change Lane**: ego linear probing changes the lane to avoid collision. **Slowing Speed**: When we disturb the trajectory, the model slows down the speed. (a)-(d): BC, (e)-(h): RL, (i)-(l): IL

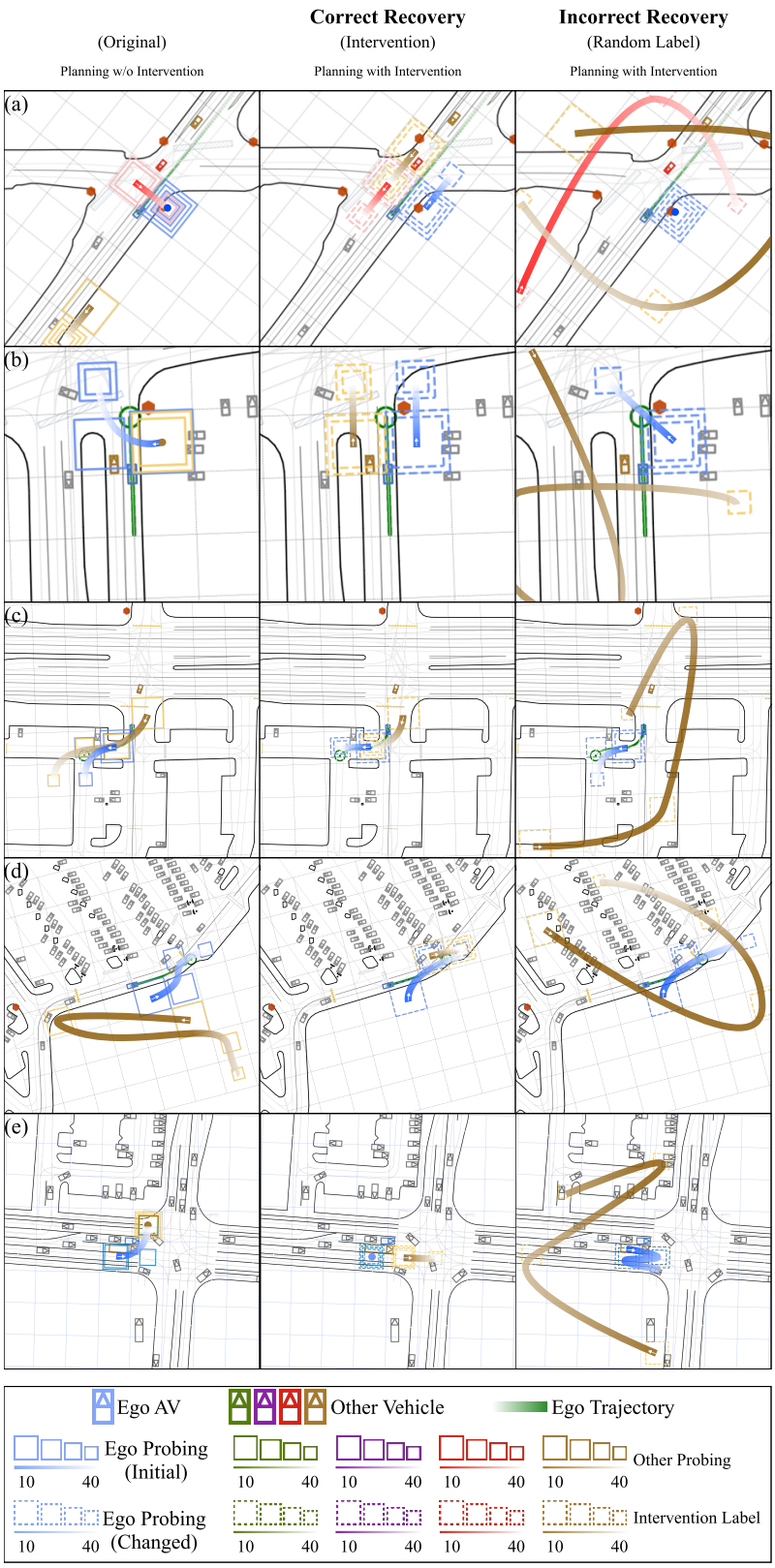


Figure 25: **Additional Examples for Recovery cases**: The visualization examples for recovery cases. **Correct Recovery**: When we change the label to align with the correct expert trajectory of the surrounding vehicle. **Incorrect Recovery**: When we set the label to a random label. (a)-(b): BC, (c)-(d): RL, (e): IL

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's scope and contributions. The paper analyzes how autonomous driving models represent and use information about surrounding vehicles, using linear probing, perturbed closed-loop simulations, near-collision analysis, and intervention experiments. These analyses support the central claims that prediction and planning capabilities emerge with scale, that nominal closed-loop success can mask weaknesses in surrounding-vehicle reasoning, and that improving internal predictions can causally improve ego planning. The experimental sections are well aligned with these stated goals.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of the work and future works in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: In this paper, we do not propose new theorems. In section 3.2, we conduct the scaling-law experiment and show that the driving models follow this scale law.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper provides model settings in Appendix B and experiment settings in Appendix C.1, D.1, and E.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Although the data and code will not accompany in the submission version, they will be released upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We briefly introduce the experiment settings in Section 3.1. Detailed settings are in the Appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include the error bar and standard deviation in all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix A.2 and A.3, we recorded the computing resources of our main experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We summarize the paper's contribution to the autonomous driving field. We analyzed the black-box deep learning autonomous driving models foucsed on utilization of surrounding vehicle information.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our experiments were conducted on a simulation, not in the real world.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets used in this paper are cited within this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our experiment is conducted in a simulation without human subjects. So, our research does not need for IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.