# EGO-CENTRIC JOINT SUPERVISION FOR PLANNING-ORIENTED AUTONOMOUS DRIVING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

End-to-end (E2E) planners predict the future trajectories of the ego and the surrounding agents, yet recent methods supervise each agent marginally against its own trajectory. However, scenes in our training data and real-world driving data often require the planner to consider ego–agent interaction, which a marginal objective does not supervise explicitly. To address this, we propose **CastAD**, which trains a planning-oriented E2E planner under *ego-centric joint supervision*. Unlike conventional joint forecasting objectives, which treat all actors symmetrically, this supervision centers the scene on the ego, whose future is a decision in planning. Only the ego and its interaction-relevant agents, those with any predicted future that overlaps the spatial trajectory the ego followed, are supervised as one scene under the mode closest to the ego's spatial trajectory. A policy optimization stage then refines the stochastic ego policy on the joint-mode ego embeddings, which this supervision trains to encode ego–agent interaction, with rewards scored against map context and logged agent futures. CastAD reaches state-of-the-art results on closed-loop Bench2Drive (**Driving Score 87.53**, **Success Rate 71.82**) and on NAVSIM (**PDMS 91.1**). Video demos are available at `https://castad-anon.github.io`.

## 1 INTRODUCTION

End-to-end (E2E) autonomous driving has emerged as an alternative to modular pipelines that chain perception, prediction, and planning as separate stages (Zhou and Tuzel, 2018; Liu et al., 2023; Moon et al., 2024; Nayakanti et al., 2023; Shi et al., 2022; Dauner et al., 2023): a single model maps sensor inputs directly to future trajectories or control actions (Zeng et al., 2019; Chitta et al., 2023; Hu et al., 2023; Jiang et al., 2023; Sun et al., 2025). Large-scale open datasets such as nuScenes (Caesar et al., 2020), simulators such as CARLA (Dosovitskiy et al., 2017), and planning benchmarks (Jia et al., 2024; Dauner et al., 2024) have made E2E driving an active research area. On these benchmarks, E2E architectures have progressed from dense bird's-eye-view feature maps (Hu et al., 2023; Jiang et al., 2023; Chen et al., 2024b) to sparse instance queries (Sun et al., 2025) and to a single decoder shared by perception, prediction, and planning (Tang et al., 2025; Zhao et al., 2026).

Despite this progression, E2E planners still supervise the ego plan and each surrounding agent's future as separate marginals (Step 1 of Figure 1(a)). Such planners can act incorrectly when the right action depends on how another vehicle responds, as in merging into a busy adjacent lane or overtaking a stopped vehicle through the oncoming lane. Ego and agent queries already exchange information through attention in these planners (Hu et al., 2023; Jiang et al., 2023; Chen et al., 2024b; Sun et al., 2025; Tang et al., 2025; Zhao et al., 2026), so the remaining gap lies in the supervision. A marginal objective cannot penalize jointly inconsistent predictions: each trajectory can be plausible on its own while the scene they form together cannot occur.

In parallel, joint motion forecasting models interacting agents as one scene (Girgis et al., 2021; Luo et al., 2023; Seff et al., 2023) and may appear to offer a solution. Its all-actor joint objective supervises every actor under one shared mode, which penalizes combinations of individually plausible but mutually incompatible trajectories (Ngiam et al., 2022; Wagner et al., 2024). In E2E planning, however, the ego future is a decision, and only a subset of surrounding agents is relevant to it. The planner needs a scene representation centered on the ego: how these agents respond to each ego maneuver, and how the ego should respond to them. The all-actor objective selects the shared mode by the error summed over all actors, including agents irrelevant to the ego, and this mode need not



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

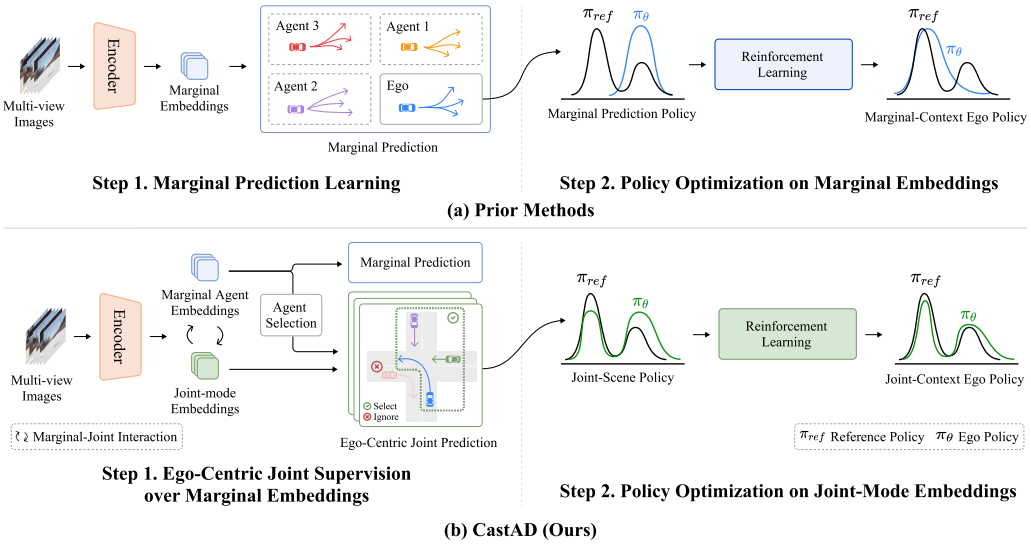


Figure 1: Prior E2E methods vs. CastAD. (a) Prior methods supervise each output against its own marginal target and optionally refine the resulting marginal representations with RL-based policy optimization. (b) CastAD's ego-centric joint supervision uses the ego's spatial trajectory to select interaction-relevant agents and couples them with the ego under the ego-selected mode; policy optimization runs on the resulting joint-mode ego embeddings.

match the maneuver the ego performed. In our experiments, it even lowers mean driving ability below that of the marginally supervised planner (Table 3).

To address this problem, we propose **CastAD**, a framework that uses *ego-centric joint prediction* for planning-oriented E2E planners. It *casts* the ego and its interaction-relevant agents in one supervised scene under a shared mode (Step 1 of Figure 1(b)). An ego-centric joint prediction branch exchanges information with the marginal embeddings and decodes one scene hypothesis of the ego and all agents per mode. Ego-centric joint prediction consists of two parts. First, it selects the interaction-relevant agents: those with any predicted future that overlaps the ego's spatial trajectory, the geometric path the ego followed in the log, independent of timing. Second, it jointly supervises the ego and these agents under the shared mode whose predicted ego trajectory lies closest to this path. The other agents keep their best marginal modes. This *ego-centric joint supervision* reformulates the joint objective for planning and lets the planner explicitly learn only the interactions that matter for the ego plan.

Ego-centric joint supervision also makes policy optimization more effective. Through imitation of logged data, it trains the joint-mode ego embeddings to encode how the ego and the relevant agents interact. Policy optimization then scores these interactions with rewards for safety, together with map compliance, comfort, and long-horizon progress, criteria that pointwise imitation losses underweight (Lu et al., 2023; Li et al., 2026b). Recent methods refine the ego policy with reward or preference signals (Song et al., 2026; Li et al., 2026b; Shang et al., 2025; Lian et al., 2026), yet they operate on marginal or ego-only representations (Step 2 of Figure 1(a)). CastAD applies policy optimization to the joint-mode ego embeddings (Step 2 of Figure 1(b)), where it raises success rate and mean driving ability more than it does on the marginal baseline (Table 3).

CastAD achieves state-of-the-art results on closed-loop Bench2Drive (Jia et al., 2024) and NAVSIM (Dauner et al., 2024). Our Bench2Drive ablations show that most of the gain comes from ego-centric joint supervision. It also makes policy optimization more effective, because the optimization refines a policy distribution that has already learned, from logged data, the interactions the ego needs.

## 2 Related Work

### 2.1 End-to-End Planning and Reinforcement Learning for Autonomous Driving

E2E planners (Hu et al., 2023; Weng et al., 2024; Jiang et al., 2023; Chitta et al., 2023; Sun et al., 2025; Tang et al., 2025) learn planning-oriented representations from sensors. Planners

capture multi-modality with anchor-based (Chen et al., 2024b; Sun et al., 2025; Tang et al., 2025), latent-variable (Zheng et al., 2024), or diffusion- and flow-based (Liao et al., 2025; Xing et al., 2025) designs, and world models (Li et al., 2024; Zheng et al., 2025; Li et al., 2025c;d; Yang et al., 2026; Zhang et al., 2025; 2026a;b) predict future scene evolution. Unified-decoder planners (Tang et al., 2025; Zhao et al., 2026), which predict the ego plan and agent futures, use a marginal objective that supervises each actor against its own target. Reinforcement learning (RL) and preference optimization (Jiang et al., 2025; Lu et al., 2023; Li et al., 2025a; Song et al., 2026; Shang et al., 2025; Li et al., 2026b; Lian et al., 2026; Yang et al., 2025; Gao et al., 2026) refine imitation-trained policies, but act on marginal or ego-only representations.

### 2.2 Joint Motion Forecasting and Interaction-Aware Planning

Joint motion forecasting targets scene-consistent futures, since marginal futures can be mutually inconsistent when actors interact. Approaches include scene-centric (Ngiam et al., 2022) and query-based (Shi et al., 2022) transformers, conditional or factorized models (Sun et al., 2022; Luo et al., 2023), latent-variable (Girgis et al., 2021) and token-based (Seff et al., 2023) models, marginal-to-joint decoding (Wagner et al., 2026), and self-supervised pretraining (Wagner et al., 2024). Joint objectives (Ngiam et al., 2022; Wagner et al., 2024) supervise all actors under one scene mode chosen by the all-actor winner-takes-all rule, and conditional variants (Ngiam et al., 2022; Sun et al., 2022) take one actor's future as input. A planner, by contrast, produces the ego future and needs agent futures that respond to its maneuver. Also related are the interaction-aware planners SafeDrive (Kim et al., 2026), which conditions a sparse world representation on each candidate ego trajectory, and FlowAD (Guo et al., 2026), which treats ego–scene interaction as a scene-flow feature. Neither supervises agent futures under a mode shared with the ego. CastAD's ego-centric joint supervision couples the ego and its interaction-relevant agents under the ego-selected mode, and policy optimization runs on the resulting joint-mode ego embeddings. Appendix A extends this comparison.

## 3 Method

Marginal supervision lets individually plausible futures be interaction-inconsistent. All-actor joint supervision selects the shared mode by the error summed over all actors, in which the ego's error weighs the same as that of an agent the ego does not interact with. As shown in Figure 2, we augment a query-based E2E planner with an ego-centric joint prediction branch whose supervision selects interaction-relevant agents by the ego's spatial trajectory and binds them to the ego-selected mode (Section 3.2). A policy optimization stage then refines the stochastic ego policy on the joint-mode ego embeddings with planning-oriented feedback (Section 3.3).

### 3.1 Preliminaries

We build on HiP-AD (Tang et al., 2025), a query-based planner that unifies detection, mapping, motion prediction, and planning in a shared task-query architecture. Its decoder refines task queries through temporal, agent-agent, and interaction attention, coupling planning and ego queries with detection and map queries; the resulting marginal embeddings encode actor-specific motion cues. We denote the current marginal embedding of agent $i$ by $\mathbf{x}_i^a \in \mathbb{R}^C$, and the current ego planning embedding by $\mathbf{x}^e \in \mathbb{R}^C$, omitting refinement-layer indices for simplicity. Throughout this section, $N$ denotes the number of surrounding agents and $C$ the feature dimension. This planner represents ego trajectories at multiple granularities. During training, we use the ground-truth spatial trajectory $\tau_{\mathrm{sp}}^e$, the geometric path the ego follows independent of time, as the target for ego-centric mode selection and as a geometric cue for selecting interaction-relevant agents. Optimized for actor-wise prediction, the marginal embeddings do not represent shared scene modes organized around ego motion.

### 3.2 Ego-Centric Joint Prediction

We augment the marginal motion prediction branch with an ego-centric joint prediction module. Conventional joint motion forecasting selects a scene mode by the aggregate error over all actors, targeting scene-level accuracy (Figure 3(a)). Ego-centric joint supervision reformulates this rule for planning: it couples the ego with only the agents relevant to its maneuver and selects the mode by the ego's recorded spatial trajectory alone (Figure 3(b)). This mode ties the ego's trajectory to how

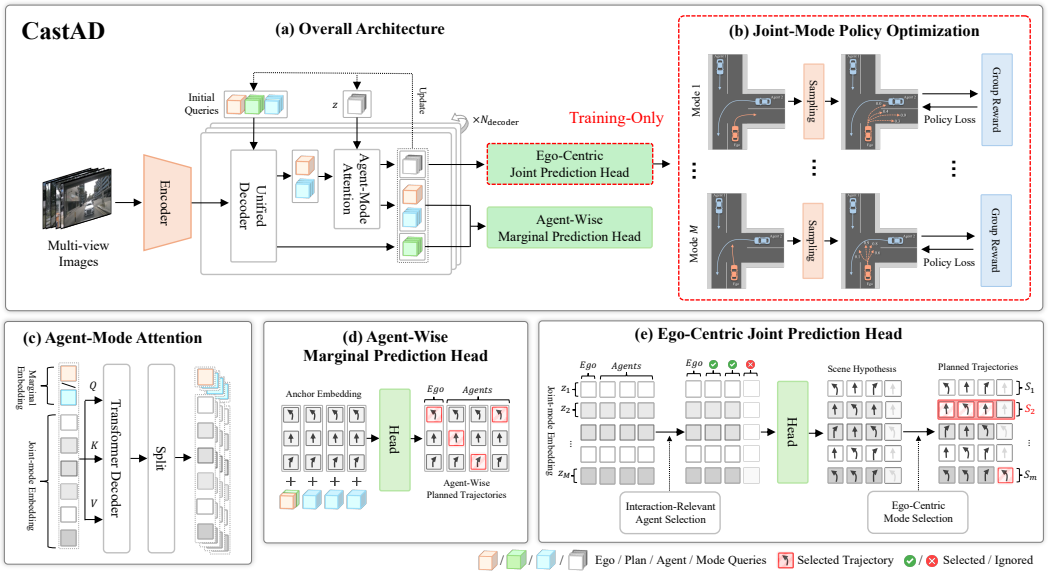


Figure 2: Overview of CastAD. (a) An ego-centric joint prediction branch is attached to the marginal branch of a query-based planner; its prediction head is used only for ego-centric joint supervision during training. (b) A policy optimization stage refines the stochastic ego policy on the joint-mode ego embeddings. (c) Agent-Mode Attention lets marginal and joint-mode embeddings exchange information. (d) The marginal branch keeps its actor-wise targets. (e) Interaction-relevant agents, selected by the ego's spatial trajectory, are supervised under the ego-selected mode.

the relevant agents moved around it. Concretely, the module introduces $M$ ego-centric scene modes, where each mode $m$ predicts an ego trajectory $\hat{\tau}^e_m$ and surrounding-agent trajectories $\{\hat{\tau}^a_{i,m}\}_{i=1}^N$ under the same scene hypothesis.

**Marginal–Joint Embedding Interaction.** Building on the decoded ego and agent embeddings defined in Section 3.1, Agent-Mode Attention enables information exchange between marginal and joint-mode embeddings. First, we introduce joint-mode embeddings to instantiate ego-centric scene modes. For surrounding agent $i$, let $\mathbf{z}^a_{i,m} \in \mathbb{R}^C$ denote the agent joint-mode embedding associated with the $m$-th scene mode, and let $\mathbf{z}^e_m \in \mathbb{R}^C$ denote the ego joint-mode embedding for the same $m$-th mode. We then form ego and agent mode stacks by concatenating each decoded embedding with its $M$ joint-mode embeddings, namely $\mathbf{U}^e = [\mathbf{x}^e; \mathbf{z}^e_1; \ldots; \mathbf{z}^e_M]$ and $\mathbf{U}^a_i = [\mathbf{x}^a_i; \mathbf{z}^a_{i,1}; \ldots; \mathbf{z}^a_{i,M}]$, where $\mathbf{U}^e, \mathbf{U}^a_i \in \mathbb{R}^{(1+M)\times C}$. As illustrated in Figure 2(c), the Agent-Mode Attention module applies attention (Vaswani et al., 2017) to the ego and agent stacks jointly:

$$\bar{\mathbf{U}} = \mathrm{Attn}_{\mathrm{AM}}(\mathbf{U}, \mathbf{U}, \mathbf{U}), \qquad \mathbf{U} = [\mathbf{U}^e; \mathbf{U}^a_1; \ldots; \mathbf{U}^a_N]. \tag{1}$$

In practice, we concatenate the ego stack and all surrounding-agent stacks into one token sequence of shape $[B, (1+N)(1+M), C]$ and apply Agent-Mode Attention to it, so each joint-mode embedding attends to the tokens of the other actors. A learned mode-index embedding shared by the ego and the agents marks the $m$-th token of every actor with the same scene-mode identity. Since the marginal embeddings $\mathbf{x}$ are produced by the unified decoder, they already contain ego, agent, and map context. Agent-Mode Attention then injects entity-level context into each joint-mode embedding $\mathbf{z}$, enabling each mode to capture ego- or agent-specific scene information. The first output token updates the decoded ego or agent embedding, while the remaining tokens become refined joint-mode embeddings. After the final refinement layer, an Ego-Centric Joint Prediction Head decodes the $m$-th ego joint-mode embedding $\mathbf{z}^e_m$ and the corresponding agent joint-mode embeddings $\{\mathbf{z}^a_{i,m}\}_{i=1}^N$ into an ego-centric scene hypothesis $\mathcal{S}_m = (\hat{\tau}^e_m, \hat{\tau}^a_{1,m}, \ldots, \hat{\tau}^a_{N,m})$, where $\hat{\tau}^e_m$ denotes the predicted ego trajectory for mode $m$, and $\hat{\tau}^a_{i,m}$ denotes the predicted trajectory of agent $i$ under the same mode.

**Interaction-Relevant Agent Selection.** A key design choice is that we do not force all agents to follow joint-mode supervision. Most surrounding agents do not interact with the ego; assigning them to a shared mode spends supervision where no interaction exists. The all-actor rule degrades driving ability (Table 3). We select an interaction set $\mathcal{A}_{\mathrm{int}}$ using the ground-truth ego spatial trajectory $\tau^e_{\mathrm{sp}}$

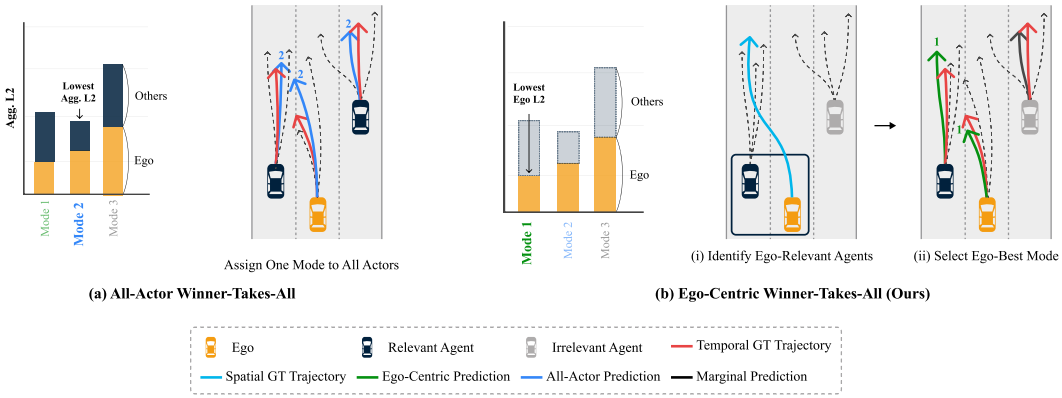


Figure 3: Joint-mode selection. (a) All-actor winner-takes-all picks the mode with the lowest L2 error over all actors, which may miss the ego's maneuver. (b) Ego-centric mode selection picks the mode by ego spatial-trajectory error alone and supervises interaction-relevant agents under it.

and the marginal agent predictions:

$$\mathcal{A}_{\text{int}} = \left\{ i \;\middle|\; \exists k \in \mathcal{K}_i \text{ s.t. collision} \left( \tilde{\tau}^a_{i,k}, \tau^e_{\text{sp}} \right) = \text{true} \right\}, \tag{2}$$

where $\tilde{\tau}^a_{i,k}$ denotes the $k$-th marginal prediction of agent $i$, and $\mathcal{K}_i$ denotes the set of marginal prediction modes considered for collision checking. We use the spatial trajectory because it directly reflects where the ego intends to move, whereas a temporally sampled trajectory entangles the intended path with speed and timing.

**Ego-Centric Mode Selection.** A conventional all-actor winner-takes-all rule selects the scene mode with the lowest aggregate error, e.g., the L2 error summed over all actors (Ngiam et al., 2022; Wagner et al., 2024). In E2E driving, however, we organize the joint representation around the ego trajectory to better align it with the ego planning objective. As shown in Figure 3, we accordingly select the joint mode based on the ego's spatial trajectory and apply the selected mode only to interaction-relevant agents. Formally, we define the selected joint scene mode as:

$$m^\star = \arg\min_m d(\hat{\tau}^e_m, \tau^e_{\text{sp}}), \tag{3}$$

where $d(\cdot, \cdot)$ is a masked trajectory distance. Agents in $\mathcal{A}_{\text{int}}$ are then supervised under the same ego-selected mode $m^\star$, while non-selected agents keep marginal supervision with their own best marginal mode $k^\star_i = \arg\min_k d(\tilde{\tau}^a_{i,k}, \tau^a_{i,\text{gt}})$. This design focuses scene-mode learning on reciprocal dependencies around ego motion. The joint mode classification loss is formulated as:

$$\mathcal{L}_{\text{joint_cls}} = \mathcal{L}^e_{\text{cls}}(m^\star) + \sum_{i \in \mathcal{A}_{\text{int}}} \mathcal{L}^a_{\text{cls},i}(m^\star) + \sum_{i \notin \mathcal{A}_{\text{int}}} \mathcal{L}^a_{\text{cls},i}(k^\star_i), \tag{4}$$

where $\mathcal{L}_{\text{cls}}$ is the focal loss (Lin et al., 2017).

### 3.3 Policy Optimization

The key novelty of this stage is that policy refinement operates on the ego joint-mode embeddings $\mathbf{z}^e_m$ learned through ego-centric joint prediction (Section 3.2), whereas prior methods use marginal or ego-only embeddings. These embeddings are supervised jointly with the interaction-relevant agents. Optimization then acts on features that already encode the interactions that shaped the logged ego behavior. To improve closed-loop driving quality beyond supervised imitation, we implement this policy optimization with a post-training GRPO objective (Shao et al., 2024; Guo et al., 2025), shown in Figure 2(b). Concretely, the ego trajectory head is parameterized as a Gaussian policy. For each joint mode $m$, we sample $G$ ego rollouts and score the $g$-th rollout with a planning-oriented reward $r_{m,g}$, using the penalty-plus-quality decomposition of the Predictive Driver Model Score (PDMS; Dauner et al., 2024), adapted for training (Appendix B). Computing the reward against logged agent futures and map context avoids the compounding errors of rolling out model-predicted agent reactions.

**Gaussian Trajectory Policy.** E2E planners typically formulate future trajectory prediction as a deterministic regression task. To incorporate reinforcement learning, however, the model must

parameterize a trajectory distribution. We therefore interpret the original regression output as the predictive mean $\mu$ and additionally predict the standard deviation $\sigma$ (Yang et al., 2026). The supervised joint regression term is then written as a Gaussian negative log-likelihood:

$$\begin{aligned}\mathcal{L}_{\text{joint_reg}} &= \mathcal{L}^{e}_{\text{reg}}(m^\star) + \sum_{i\in\mathcal{A}_{\text{int}}} \mathcal{L}^{a}_{\text{reg},i}(m^\star) \\ &\quad + \sum_{i\notin\mathcal{A}_{\text{int}}} \mathcal{L}^{a}_{\text{reg},i}(k_i^\star), \\ \mathcal{L}_{\text{reg}}(\tilde{m}) &= \frac{1}{2}\Big(\frac{\big(\mu_{\tilde{m}} - \tau_{\text{gt}}\big)^2}{\sigma^2_{\tilde{m}}} + 2\log\sigma_{\tilde{m}} + \log 2\pi\Big),\end{aligned} \tag{5}$$

where $\mu_{\tilde{m}}$ and $\sigma_{\tilde{m}}$ denote the predicted mean and standard deviation for mode $\tilde{m}$, and $\tau_{\text{gt}}$ denotes the corresponding ground-truth trajectory target. This converts the original trajectory regression head into a stochastic policy head during training; during inference, we use only the predicted mean trajectory $\mu$.

**Policy Optimization over Joint-Mode Embeddings.** We first sample ego-only rollouts for each mode, represented as $\bar{\tau}_m \in \mathbb{R}^{G\times T_{\text{fut}}\times 2}$, where $T_{\text{fut}}$ denotes the prediction horizon. We then normalize the rewards $r_{m,g}$ within each mode group and define the truncated advantage following Zou et al. (2025):

$$\begin{aligned}A_{m,g} &= \frac{r_{m,g} - \text{mean}(r_m)}{\text{std}(r_m)}, \\ \tilde{A}_{m,g} &= \begin{cases} -1, & \text{if an at-fault collision occurs,} \\ \max(0, A_{m,g}), & \text{otherwise.}\end{cases}\end{aligned} \tag{6}$$

We then optimize the clipped GRPO objective over sampled ego rollouts:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{m,g}\big[\min\big(\rho_\theta \tilde{A}_{m,g}, \text{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon)\tilde{A}_{m,g}\big)\big], \tag{7}$$

where $\rho_\theta = \pi_\theta/\pi_{\theta_{\text{old}}}$ is the probability ratio between the current and old policies, and $\epsilon$ controls the update step size for training stability. Compared with plain reward maximization, this group-relative formulation ranks candidate rollouts under the same scene context, which suits interactive planning: what matters is not whether a trajectory is feasible in isolation but whether it is preferable to its alternatives. Importantly, we apply reinforcement learning only to the ego policy, while keeping surrounding-agent prediction supervised. This avoids noisy per-agent rewards and stabilizes the joint scene predictor. Supervising interaction-relevant agents under the ego-selected mode encourages the shared embeddings to capture *why* agents move—the directed dependencies behind observed futures. The full training objective is given in Appendix C.

# 4 Experiments

## 4.1 Experimental Setup

We apply CastAD to one base planner on each of two benchmarks: HiP-AD (Tang et al., 2025) on Bench2Drive (Jia et al., 2024) and DiffusionDrive (Liao et al., 2025) on NAVSIM (Dauner et al., 2024). Bench2Drive is a closed-loop benchmark of 220 routes with interaction-critical scenarios, and NAVSIM scores planning on real-world driving logs (`navtest`, official v1 protocol). On Bench2Drive we report the official closed-loop metrics: Driving Score (DS), Success Rate (SR), Efficiency, and Comfortness. We also report five driving-ability scores and their average, the mean driving ability. Table 3 uses the Bench2Drive-mini subset: 54 of the 220 routes, covering 43 scenario types (route IDs in Appendix D.2). On NAVSIM we report PDMS with its NC, DAC, EP, and TTC components (Appendix D.1).

**Implementation.** For Bench2Drive we build on HiP-AD, a sparse query-based planner whose multi-granularity plan queries produce both lateral and longitudinal trajectories; for NAVSIM we build on DiffusionDrive, a diffusion-based planner. We follow their settings unless stated otherwise. On Bench2Drive, the spatial ego trajectory $\tau^e_{\text{sp}}$ is sampled at 2 m spacing, and the predicted temporal and spatial trajectories drive longitudinal and lateral control, respectively.

Table 1: Bench2Drive (Jia et al., 2024) results under open-loop, closed-loop, and driving-ability metrics. Avg. L2 is averaged over the 2 s horizon at 2 Hz. ‡: privileged expert that collected the training data (not ranked); best learned planner in **bold**.

| Method | Open-loop | Closed-loop Metric | | | | Ability (%) ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. L2 (m) ↓ | DS ↑ | SR (%) ↑ | Effic. ↑ | Comf. ↑ | Merging | Overtaking | Emergency Brake | Give Way | Traffic Sign | Mean |
| Think2Drive‡ (2024) | – | 91.85 | 85.41 | – | – | – | – | – | – | – | – |
| UniAD (2023) | 0.73 | 45.81 | 16.36 | 129.21 | 43.58 | 14.10 | 17.78 | 21.67 | 10.00 | 14.21 | 15.55 |
| VAD (2023) | 0.91 | 42.35 | 15.00 | 157.94 | 46.01 | 8.11 | 24.44 | 18.64 | 20.00 | 19.15 | 18.07 |
| SparseDrive (2025) | 0.87 | 44.54 | 16.71 | 170.21 | 48.63 | 12.50 | 17.50 | 20.00 | 20.00 | 23.03 | 18.60 |
| GenAD (2024) | – | 44.81 | 15.90 | – | – | – | – | – | – | – | – |
| DriveTransformer (2025) | **0.62** | 63.46 | 35.01 | 100.64 | 20.78 | 17.57 | 35.00 | 48.36 | 40.00 | 52.10 | 38.60 |
| Hydra-NeXt (2025e) | 0.92 | 73.86 | 50.00 | 197.76 | 20.68 | 40.00 | 64.44 | 61.67 | **50.00** | 50.00 | 53.22 |
| MomAD (2025) | 0.82 | 47.91 | 18.11 | 174.91 | 51.20 | – | – | – | – | – | – |
| VADv2 (2024b) | – | 76.15 | 50.46 | 178.24 | 37.81 | – | – | – | – | – | – |
| SafeDrive (2026) | – | 66.77 | 42.40 | – | – | – | – | – | – | – | – |
| FlowAD (2026) | – | 51.77 | 22.02 | 178.42 | **53.89** | 20.00 | 28.67 | 25.00 | 25.00 | 28.42 | 25.42 |
| *Unified-decoder methods* | | | | | | | | | | | |
| HiP-AD (2025) | 0.69 | 86.77 | 69.09 | **203.12** | 19.36 | 50.00 | 84.44 | **83.33** | 40.00 | **72.10** | 65.98 |
| UniTeD (2026) | – | 87.30 | 70.00 | – | – | – | – | – | – | – | – |
| *RL-refined methods* | | | | | | | | | | | |
| DIVER (2026) | 1.11 | 68.90 | 36.75 | 72.34 | 22.34 | 35.08 | 25.09 | 41.09 | **50.00** | 59.21 | 42.09 |
| ReCogDrive (2026b) | – | 71.36 | 45.45 | 138.18 | 17.45 | 29.73 | 20.00 | 69.09 | 20.00 | 71.34 | 42.03 |
| **CastAD (Ours)** | 0.68 | **87.53** | **71.82** | 172.53 | 34.34 | **60.00** | **86.67** | **83.33** | **50.00** | 71.34 | **70.27** |

**Training.** Training runs in three stages: the detection module for 12 epochs, the planning module with ego-centric joint supervision for 12 more, and policy optimization for 6. We use AdamW (Loshchilov and Hutter, 2019) with weight decay 0.001 and an initial learning rate of $2 \times 10^{-4}$ for the detection and planning stages and $2 \times 10^{-5}$ for the policy optimization stage. We train on 8 NVIDIA A6000 GPUs with a total batch size of 48; for ablations, we halve the schedule to 6 epochs for the planning stage and 3 for policy optimization.

### 4.2 Comparison with State-of-the-Art Methods

**Evaluation on Bench2Drive.** As shown in Table 1, CastAD achieves state-of-the-art closed-loop performance on Bench2Drive, with a Driving Score of 87.53 and a Success Rate of 71.82. It surpasses the previous best, UniTeD, by 0.23 DS and 1.82 SR. It also exceeds the interaction-aware planners SafeDrive and FlowAD (66.77 and 51.77 DS). Over its base planner HiP-AD, it gains 0.76 DS and 2.73 SR and raises mean driving ability from 65.98 to 70.27. The largest driving-ability gains are in Merging and Give Way ($+10.00$ each), both requiring interaction with a neighboring vehicle. No method in Table 1 exceeds 50.00 in Give Way. Half of the Give Way routes require yielding to an emergency vehicle approaching from behind, and these routes may require audio sensing or training targeted at emergency vehicles. Emergency Brake stays at 83.33, and Traffic Sign drops from 72.10 to 71.34. Both abilities hinge on a sudden obstacle or a traffic sign, where no agent's response to the ego plan decides the outcome. Comfortness rises from 19.36 to 34.34. Open-loop L2 is a poor proxy for closed-loop driving (Appendix E). In mean driving ability, the RL-refined DIVER and ReCogDrive (42.09 and 42.03) trail imitation-trained HiP-AD (65.98). This gap suggests that policy optimization alone does not compensate for the representation it refines.

**Evaluation on NAVSIM.** In Table 2, CastAD again attains the highest PDMS, 91.1, and improves every subscore of its base planner DiffusionDrive (88.1): NC by 1.2, DAC by 2.1, EP by 2.7, and TTC by 1.2. Its NC (99.4) and DAC (98.3) are the best among all compared methods. PDMS multiplies the weighted quality terms by these two penalty terms, so an at-fault collision or an off-road segment scales the entire score down. Removing such failures is worth more than the subscore margins suggest. ReCogDrive shows the same arithmetic from the other side: its EP exceeds ours by 2.4, yet its lower NC and DAC leave it behind on PDMS. Among the RL-refined planners, CastAD also has the best TTC (95.9), the remaining subscore that depends on what the surrounding agents do.

### 4.3 Ablation Studies

We ablate the training components of CastAD in Table 3. By default, models use the halved schedule and are evaluated on the 54 routes of Bench2Drive-mini, where one route equals 1.85 SR points. The marginal baseline is HiP-AD reproduced in this setting.

Table 2: NAVSIM (Dauner et al., 2024) results on `navtest` under the official v1 protocol. Best in **bold**.

| Method | PDMS Components | | | | PDMS ↑ |
|---|---|---|---|---|---|
| | NC ↑ | DAC ↑ | EP ↑ | TTC ↑ | |
| UniAD (2023) | 97.8 | 91.9 | 78.8 | 92.9 | 83.4 |
| PARA-Drive (2024) | 97.9 | 92.4 | 79.3 | 93.0 | 84.0 |
| LAW (2025c) | 96.4 | 95.4 | 81.7 | 88.7 | 84.6 |
| GoalFlow (2025) | 98.3 | 93.8 | 79.8 | 94.3 | 85.7 |
| DiffusionDrive (2025) | 98.2 | 96.2 | 82.2 | 94.7 | 88.1 |
| Hydra-MDP++ (2025b) | 97.6 | 96.0 | 80.4 | 93.1 | 86.6 |
| WoTE (2025d) | 98.5 | 96.8 | 81.9 | 94.9 | 88.3 |
| SeerDrive (2025) | 98.4 | 97.0 | 83.2 | 94.9 | 88.9 |
| VADv2 (2024b) | 98.3 | 97.4 | 82.3 | 95.7 | 89.3 |
| SafeDrive (2026) | 99.0 | 97.3 | – | – | 89.6 |
| *Unified-decoder methods* | | | | | |
| HiP-AD (2025) | 98.9 | 96.7 | 81.2 | 96.3 | 88.6 |
| UniTeD (2026) | 98.9 | 97.2 | 84.1 | **96.6** | 90.2 |
| *RL-refined methods* | | | | | |
| DIVER (2026) | 98.5 | 96.5 | 82.6 | 94.9 | 88.3 |
| DriveDPO (2025) | 98.5 | 98.1 | 84.3 | 94.8 | 90.0 |
| ReCogDrive (2026b) | 97.9 | 97.3 | **87.3** | 94.9 | 90.8 |
| WorldRFT (2026) | 97.8 | 96.8 | 81.7 | 94.0 | 87.8 |
| **CastAD (Ours)** | **99.4** | **98.3** | 84.9 | 95.9 | **91.1** |

Table 3: Ablation of the supervision rule and policy optimization on Bench2Drive-mini (Tang et al., 2025). All joint rows use the same ego-centric joint prediction branch architecture; "–" in the cue column of the joint rows denotes all-actor winner-takes-all supervision. T/S: ground-truth temporal/spatial ego cue for interaction-relevant agent selection; Pred(S): predicted spatial cue; †: reproduced. Best in **bold**; shaded: full model.

| Supervision | Ego Cue | Policy Optim. | DS ↑ | SR (%) ↑ | Mean Abil. (%) ↑ |
|---|---|---|---|---|---|
| Marginal | – | – | 83.12 | 62.96 | 63.83 |
| (HiP-AD†) | – | ✓ | 87.43 | 64.81 | 66.38 |
| Joint | – | – | 84.76 | 64.81 | 56.29 |
| | T | – | 86.90 | 68.52 | 66.92 |
| | Pred(S) | – | 86.47 | 66.67 | 65.40 |
| | S | – | 88.61 | 75.93 | 72.66 |
| | S | ✓ | **91.05** | **79.63** | **76.51** |

Table 4: Ablation of the policy optimization scope on the full 220-route Bench2Drive (Jia et al., 2024) set. Best in **bold**.

| Policy Optimization Scope | DS ↑ | SR (%) ↑ | Mean Abil. (%) ↑ |
|---|---|---|---|
| All interaction-relevant agents | 86.90 | 65.45 | 65.36 |
| **Ego only (Ours)** | **87.53** | **71.82** | **70.27** |

**Supervision Rule.** We first replace ego-centric joint supervision (spatial cue) with the all-actor winner-takes-all rule of joint motion forecasting (Figure 3(a)), keeping the ego-centric joint prediction branch and its parameter count fixed. The replacement costs 3.85 DS, 11.12 SR, and 16.37 points of mean driving ability. The last figure is the informative one: in mean driving ability, conventional all-actor joint supervision (56.29) falls below the marginal baseline (63.83), which has no joint branch. Two things change together here, the rule that picks the mode and the set of agents supervised under it, and both plausibly contribute. The all-actor rule picks the mode that minimizes the error summed over all actors, including agents outside the interaction set, and this mode need not be the one that best explains the ego's trajectory. It then supervises every agent under this mode, whereas the ego-centric rule supervises only the interaction-relevant agents under it. Under the ego-centric rule, the same branch raises mean driving ability to 72.66. Every ego-centric variant in Table 3, including those with the temporal or predicted cue, exceeds both the all-actor rule and the marginal baseline on DS, SR, and mean driving ability.

**Agent-Selection Cue.** With ego-centric mode selection fixed, we vary the cue that decides which agents share the selected mode. Moving from the temporal cue (T) to the spatial one (S) adds 1.71 DS, 7.41 SR, and 5.74 mean driving ability, the pattern Section 3.2 predicts. A temporally sampled ego trajectory barely advances while the ego waits, so the agent it is waiting for does not overlap it and is left out of the interaction set, exactly in the scenes where the interaction decides the outcome. Swapping the ground-truth spatial cue for the model's own prediction (Pred(S)) costs 2.14, 9.26, and 7.26 on the same three metrics, which places the cue at the center: when the ego cue itself is predicted, its errors enter the supervision and are then trained on.

**Ego-Centric Joint Supervision and Policy Optimization.** With the ground-truth spatial cue, we next separate what the supervision contributes from what policy optimization contributes. Applied to the marginal baseline, policy optimization is worth 4.31 DS, 1.85 SR, and 2.55 mean driving ability. Ego-centric joint supervision, with no policy optimization at all, is worth 5.49, 12.97, and 8.83. That model already leads the policy-optimized marginal baseline by 1.18 DS, 11.12 SR, and 6.28 mean driving ability. Policy optimization yields a stronger planner when it refines the representation learned under ego-centric joint supervision: 91.05 DS, 79.63 SR, and 76.51 mean driving ability, against 87.43, 64.81, and 66.38 when it refines the marginal representation. Running the same policy optimization on top of it adds 3.70 SR and 3.85 mean driving ability, against 1.85 and 2.55 on the marginal baseline; only the DS gain is smaller (2.44 vs. 4.31), from a starting point already 5.49 DS

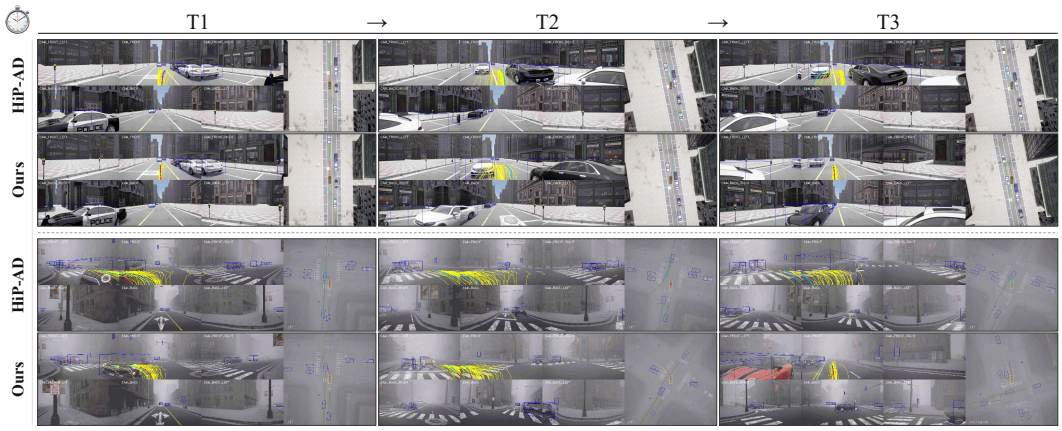


Figure 4: Qualitative results on Bench2Drive (Jia et al., 2024) over consecutive frames. Sky-blue: spatial trajectory; red: selected temporal trajectory; yellow: remaining temporal candidates. Top: CastAD passes through the gap between two vehicles. Bottom: it completes the left turn without stopping, whereas the marginally supervised baseline remains stopped in the middle of the intersection. Click the image to view the video demos.

higher. The split is consistent across the three metrics: supervision accounts for 5.49 of the 7.93 DS, 12.97 of the 16.67 SR, and 8.83 of the 12.68 mean driving ability that separate the full model from the marginal baseline.

**Scope of Policy Optimization.** Finally, on all 220 routes with the full schedule (Table 4), we extend the ego-only policy objective of CastAD to all interaction-relevant agents. This extension lowers DS, SR, and mean driving ability by 0.63, 6.37, and 4.91. We attribute this drop to an asymmetry: the ego trajectory is the decision being optimized, whereas agent futures are evidence of the scene's response and help most when anchored to logged behavior.

### 4.4 Qualitative Results

Figure 4 follows CastAD and the marginally supervised baseline through two interaction-critical Bench2Drive scenarios over three consecutive frames. In the top panel, the ego has to get through a narrowing stretch before the vehicle ahead closes it. CastAD moves through early and is clear by the last frame, whereas the baseline stops as soon as a vehicle appears ahead of it and does not recover, ending in a deadlock. In the bottom panel, the ego has to turn left across an intersection while other vehicles are moving through it. CastAD keeps moving without colliding with them and completes the turn, whereas the baseline chooses to stop when they cross its turning path and remains stopped in the middle of the intersection. A marginal objective assigns each nearby vehicle's future to its own best mode, so the baseline's supervision does not pair the ego plan with that vehicle's co-occurring future. In both scenes the vehicles that decide the outcome are the ones ego-centric joint supervision binds to the ego's own mode. More examples, including NAVSIM collision cases, are in Appendix F.

## 5 Conclusion

We presented CastAD, which teaches E2E planners ego–agent dependencies through ego-centric joint supervision of the ego and its interaction-relevant agents as one scene under the mode that best explains the ego's spatial trajectory. The ablations trace most of the gain over the marginal baseline to this supervision. Policy optimization on joint-mode ego embeddings yields larger Success Rate and mean driving ability gains than on marginal ones. CastAD achieves state-of-the-art results on Bench2Drive and NAVSIM. However, two limitations remain (Appendix G). The interaction set depends on marginal predictions overlapping the logged spatial path, so an agent that influences the ego without such an overlap keeps marginal supervision. The reward scores rollouts against logged agent futures, which do not react to the sampled ego plan. We plan to address both limitations in future work.

### AI USE STATEMENT

In this work, we used generative AI tools for language editing, including grammar correction and sentence-level rewriting. We did not use generative AI tools to generate research ideas, to design the methodology or the experiments, to analyze experimental results, or to produce scientific claims, and the remaining tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools to improve the clarity and readability of the manuscript. We have reviewed all AI-assisted work: every AI-assisted language edit was checked by the authors to confirm that it preserved the intended technical meaning and the scientific claims. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

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 *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 11618–11628. IEEE, 2020.

Raphael Chekroun, Marin Toromanoff, Sascha Hornauer, and Fabien Moutarde. GRI: General reinforced imitation and its application to vision-based autonomous driving. *Robotics*, 12(5):127, 2023.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024b.

Kashyap Chitta, Aditya Prakash, and Andreas Geiger. NEAT: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15793–15803, 2021.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2023.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning (CoRL)*, pages 1268–1281. PMLR, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 28706–28719, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, pages 1–16. PMLR, 2017.

Yinfeng Gao, Qichao Zhang, Deqing Liu, Zhongpu Xia, Guang Li, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, Da-Wei Ding, and Dongbin Zhao. PerlAD: Towards enhanced closed-loop end-to-end autonomous driving with pseudo-simulation-based reinforcement learning. *arXiv preprint arXiv:2603.14908*, 2026.

Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. *arXiv preprint arXiv:2104.00563*, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Mingzhe Guo, Yixiang Yang, Chuanrong Han, Rufeng Zhang, Shirui Li, Ji Wan, and Zhipeng Zhang. FlowAD: Ego-scene interactive modeling for autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17853–17862, 2023.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, volume 37, pages 819–844, 2024.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2025, pages 67227–67243, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. AlphaDrive: Unleashing the power of VLMs in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

Jungho Kim, Jiyong Oh, Seunghoon Yu, Hongjae Shin, Donghyuk Kwak, and Jun Won Choi. SafeDrive: Fine-grained safety reasoning for end-to-end driving in a sparse world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Derun Li, Changye Li, Yue Wang, Jianwei Ren, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, et al. Learning personalized driving styles via reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025a.

Hongchen Li, Tianyu Li, Jiazhi Yang, Haochen Tian, Caojun Wang, Lei Shi, Mingyang Shang, Zengrong Lin, Gaoqiang Wu, Zhihui Hao, Xianpeng Lang, Jia Hu, and Hongyang Li. PlannerRFT: Reinforcing diffusion planners through closed-loop and sample-efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-MDP++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025b.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European conference on computer vision*, pages 142–158. Springer, 2024.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, volume 2025, pages 42942–42959, 2025c.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27137–27146. IEEE, 2025d.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026b.

Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and Jose M Alvarez. Hydra-next: Robust closed-loop driving with open-loop training. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27305–27314. IEEE, 2025e.

Zhexi Lian, Haoran Wang, Xuerun Yan, Weimeng Lin, Xianhong Zhang, Yongyu Chen, and Jia Hu. Fine-tuning is not enough: A parallel framework for collaborative imitation and reinforcement learning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026.

Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. CIRL: Controllable imitative reinforcement learning for vision-based self-driving. In *European Conference on Computer Vision (ECCV)*, pages 584–599, 2018.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12037–12047. IEEE, 2025.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. ieee, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Wenjie Luo, Cheolho Park, Andre Cornman, Benjamin Sapp, and Dragomir Anguelov. JFP: Joint future prediction with interactive multi-agent modeling for autonomous driving. In *Proceedings of The 7th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1457–1467. PMLR, 2023.

Seokha Moon, Hyun Woo, Hongbeen Park, Haeji Jung, Reza Mahjourian, Hyung-gun Chi, Hyerin Lim, Sangpil Kim, and Jinkyu Kim. Visiontrap: Vision-augmented trajectory prediction guided by textual descriptions. In *European Conference on Computer Vision*, pages 361–379. Springer, 2024.

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

Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. In *International Conference on Learning Representations (ICLR)*, 2022.

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*, 2023.

Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and Zhaoxiang Zhang. DriveDPO: Policy learning via safety DPO for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems*, 2022.

Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22432–22441, 2025.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. DIVER: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3708096.

Qiao Sun, Xin Huang, Junru Gu, Brian C Williams, and Hang Zhao. M2I: From factored marginal trajectory prediction to interactive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6543–6552, 2022.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025.

Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 25605–25615. IEEE, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Royden Wagner, Omer Sahin Tas, Marvin Klemp, and Carlos Fernandez. JointMotion: joint self-supervision for joint motion prediction. In *Conference on Robot Learning (CoRL)*, 2024.

Royden Wagner, Omer Sahin Tas, Felix Hauser, Marlon Steiner, Dominik Strutz, Abhishek Vivekanandan, Jaime Villa, Yinzhe Shen, Carlos Fernandez, and Christoph Stiller. RetroMotion: Retrocausal motion forecasting models are instructable. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4302–4311, 2026.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15449–15458, 2024.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1602–1611. IEEE, 2025.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11649–11657, 2026.

Zhewen Yang, Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, Jie Zhou, and Jiwen Lu. READ: End-to-end autonomous driving made safer with efficient reinforcement learning. OpenReview, `https://openreview.net/forum?id=UOYkiemcUP`, 2025.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. AutoDrive-$R^2$: Incentivizing reasoning and self-reflection capacity for VLA model in autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026.

Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8660–8669, 2019.

Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. ResWorld: Temporal residual world model for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2026a.

Lingjun Zhang, Changjie Wu, Linzhe Shi, Jiangyang Li, Jiaxin Liu, Lei Yang, Hang Zhang, Mu Xu, and Hong Wang. DeepSight: Long-horizon world modeling via latent states prediction for end-to-end autonomous driving. *arXiv preprint arXiv:2605.10564*, 2026b.

Bo Zhao, Xinting Zhao, Naifan Li, Erkang Cheng, and Haibin Ling. UniTeD: Unified temporal diffusion for joint perception and planning in autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2026.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. GenAD: Generative end-to-end autonomous driving. In *European Conference on Computer Vision (ECCV)*, pages 87–104. Springer, 2024.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28632–28642. IEEE, 2025.

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499. Ieee, 2018.

Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets. *arXiv preprint arXiv:2412.09602*, 2024.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. DiffusionDriveV2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

# APPENDIX

## A ADDITIONAL RELATED WORK

### A.1 END-TO-END PLANNING FOR AUTONOMOUS DRIVING

End-to-end autonomous driving learns planning-oriented representations directly from sensor observations, offering an alternative to modular pipelines (Chen et al., 2024a; Chitta et al., 2023; 2021; Hu et al., 2023; Jiang et al., 2023; Sun et al., 2025; Zimmerlin et al., 2024; Tang et al., 2025; Zheng et al., 2024). Early methods such as TransFuser (Chitta et al., 2023) and NEAT (Chitta et al., 2021) predict planning-relevant waypoints from fused sensor features or BEV-based attention-field representations. More recent approaches incorporate stronger intermediate structures. UniAD (Hu et al., 2023) unifies perception, prediction, and planning in a planning-oriented framework, while VAD (Jiang et al., 2023) and subsequent works (Sun et al., 2025; Liao et al., 2025; Tang et al., 2025; Zheng et al., 2024) improve vectorized scene modeling, efficiency, multimodal ego planning, or generative future modeling. However, these methods supervise the ego plan and agent futures against separate targets, so reciprocal dependencies in highly interactive driving scenarios may be captured only partially or implicitly. World-model approaches such as DeepSight (Zhang et al., 2026b), which predicts latent BEV features for consecutive future frames, model how the scene evolves but likewise supervise the ego plan against its own target.

### A.2 REINFORCEMENT LEARNING FOR END-TO-END PLANNING

Although end-to-end driving is dominated by imitation learning (IL), reinforcement learning (RL) provides supervisory signals for objectives beyond behavior cloning, such as safety and human-preference alignment (Chekroun et al., 2023; Jiang et al., 2025; Liang et al., 2018; Lu et al., 2023; Li et al., 2025a; Shang et al., 2025; Wang et al., 2025; Li et al., 2026b; Yuan et al., 2026). Early methods such as CIRL (Liang et al., 2018) and GRI (Chekroun et al., 2023) combine expert demonstrations with RL for vision-based autonomous driving. More recently, BC-SAC (Lu et al., 2023), TrajHF (Li et al., 2025a), and DriveDPO (Shang et al., 2025) refine supervised planners with RL or preference objectives for robustness, personalized driving style, and safety, respectively. PlannerRFT (Li et al., 2026a) reinforces diffusion planners through closed-loop, sample-efficient fine-tuning, and PaIR-Drive (Lian et al., 2026) runs imitation and reinforcement learning as parallel branches to avoid sequential fine-tuning drift. READ (Yang et al., 2025) fine-tunes DiffusionDrive on NAVSIM with a GRPO-style objective over a surrogate mode-selection probability and a decomposed reward. PerlAD (Gao et al., 2026) performs RL in a rendering-free, vector-space pseudo-simulation whose agents are driven by a learned prediction world model. For vision-language-action (VLA) models, recent works further improve reasoning and planning beyond imitation-centric training (Wang et al., 2025; Li et al., 2026b; Yuan et al., 2026; Jiang et al., 2025). However, these approaches refine the policy on marginal or ego-only representations, which receive no supervision for ego–agent consistency. CastAD performs policy optimization on joint-mode ego embeddings, which are trained to encode interaction-relevant agent futures, and scores each candidate ego rollout against logged agent futures and map context.

### A.3 JOINT PREDICTION FOR MULTI-AGENT INTERACTION

In interactive driving, independently predicted marginal futures can be mutually inconsistent. Joint models address this by predicting scene-consistent futures. Scene Transformer (Ngiam et al., 2022) introduces a scene-centric transformer for joint, marginal, and conditional prediction, while M2I (Sun et al., 2022) improves tractability by factorizing interactions into influencer–reactor pairs. Other interaction-aware approaches include JFP (Luo et al., 2023), which learns consistent multi-agent trajectories; AutoBots (Girgis et al., 2021), which models joint trajectory distributions with latent-variable set transformers; MTR (Shi et al., 2022), which combines intention localization with refinement; MotionLM (Seff et al., 2023), which autoregressively models discrete motion tokens; and JointMotion (Wagner et al., 2024), which pretrains joint predictors via self-supervision. More recently, RetroMotion (Wagner et al., 2026) decomposes forecasting into marginal and pairwise joint predictions and supports instruction-based modification. Complementary to these forecasting methods, SafeDrive (Kim et al., 2026) adds fine-grained safety reasoning to end-to-end driving

by estimating agent-specific collision risks and drivable-region compliance over future timesteps. Relative to influencer–reactor factorizations such as M2I, the ego-centric rule of CastAD can be read as fixing the ego as the influencer, but it differs in two ways. First, M2I infers the roles per agent pair, whereas CastAD always anchors on the ego because the ego is the actor that plans. Second, the anchor acts only on supervision: the ego's ground-truth path selects the supervised mode and the supervised agent set during training. At deployment, no conditional prediction is run. The joint-mode embeddings still interact with the marginal embeddings through Agent-Mode Attention. Overall, these works address interaction-aware forecasting or interaction-aware planning without supervising agents under a mode selected by the ego. CastAD organizes joint supervision around the ego plan and uses the resulting representation to align the driving policy for closed-loop planning.

## B Reward Details

**Planning-Oriented Reward.** To avoid the compounding errors of rolling out model-predicted agent reactions, we score each sampled ego rollout against map context and logged agent futures. The logged agents reacted to the logged ego, so the reward is non-reactive to the sampled rollouts. On Bench2Drive, map context comes from the Bench2Drive HD map, and agent boxes and futures come from the ground-truth annotations. It follows the penalty-plus-quality form of the NAVSIM Predictive Driver Model Score (PDMS) (Dauner et al., 2024):

$$r = \underbrace{\Big( \prod_{p \in \{\mathrm{NC,DAC,DD}\}} \mathrm{score}_p \Big)}_{\text{safety/rule penalties}} \times \underbrace{\left( \frac{\sum_{w \in \{\mathrm{EP,TTC,C}\}} \mathrm{weight}_w \times \mathrm{score}_w}{\sum_{w \in \{\mathrm{EP,TTC,C}\}} \mathrm{weight}_w} \right)}_{\text{driving-quality term}}, \tag{8}$$

where NC denotes no at-fault collision, DAC drivable-area compliance, DD driving-direction compliance, EP ego progress, TTC time-to-collision within bound, and C comfort. The weights are $5.0$, $5.0$, and $2.0$ for EP, TTC, and C. Discrete subscores tie rollouts in a group and remove the ranking signal. We make DD, DAC, TTC, and comfort continuous. The at-fault-collision subscore stays discrete because at-fault collisions are safety-critical.

**Continuous Subscore Modifications.** The continuous subscores give denser rewards and fewer ties among rollouts sampled in the same scene group, which suits GRPO's relative comparison.

**Driving-direction compliance.** Let $d_{\mathrm{opp}}$ denote the maximum accumulated opposite-direction progress. We replace the original three-level score with a linear interpolation between the compliance threshold $2.0\,\mathrm{m}$ and the violation threshold $6.0\,\mathrm{m}$:

$$\mathrm{score}_{\mathrm{DD}} = \begin{cases} 1.0, & d_{\mathrm{opp}} \le 2.0, \\ 1 - \frac{d_{\mathrm{opp}} - 2.0}{6.0 - 2.0}, & 2.0 < d_{\mathrm{opp}} < 6.0, \\ 0.0, & d_{\mathrm{opp}} \ge 6.0. \end{cases}$$

**Drivable-area compliance.** The original subscore is zero whenever any off-road timestep occurs. We use the fraction of timesteps that remain within the drivable area:

$$\mathrm{score}_{\mathrm{DAC}} = 1 - \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{\text{ego off drivable area at } t\}.$$

**Time-to-collision.** The binary TTC score becomes a continuous score based on the earliest TTC-infraction timestep:

$$\mathrm{score}_{\mathrm{TTC}} = \frac{\min(t_{\mathrm{TTC}}, T)}{T},$$

where $t_{\mathrm{TTC}}$ is the earliest TTC-infraction timestep and $T$ is the rollout horizon. If no infraction occurs, we set $t_{\mathrm{TTC}} = T$, so the score is $1$.

**Comfort.** Each comfort metric receives a continuous compliance score, and the minimum of these scores replaces the hard conjunction over comfort thresholds:

$$s_k = \exp\left( -\frac{\Delta_k}{\alpha_k} \right), \qquad \mathrm{score}_{\mathrm{C}} = \min_k s_k,$$

Table A1: Bench2Drive-mini route IDs used for the closed-loop ablations.

| Split | Route IDs |
|---|---|
| 43 scenario routes | 1711, 1773, 1790, 1825, 1852, 1956, 2050, 2082, 2084, 2115, 2127, 2144, 2164, 2201, 2204, 2273, 2373, 2390, 2416, 2509, 2534, 2664, 2709, 2790, 3086, 3248, 3364, 3436, 3464, 3540, 3561, 3936, 14194, 14842, 17563, 17752, 23658, 23695, 23771, 23901, 26458, 28087, 28099 |
| 11 random routes | 1792, 2086, 2129, 2283, 2539, 2668, 26406, 26956, 27494, 27532, 28154 |

where $k$ ranges over longitudinal acceleration, lateral acceleration, jerk magnitude, longitudinal jerk, yaw acceleration, and yaw rate, with the nuPlan comfort bounds $[-4.05, 2.40]\,\mathrm{m/s^2}$, $\pm 4.89\,\mathrm{m/s^2}$, $\pm 8.37\,\mathrm{m/s^3}$, $\pm 4.13\,\mathrm{m/s^3}$, $\pm 1.93\,\mathrm{rad/s^2}$, and $\pm 0.95\,\mathrm{rad/s}$, respectively. $\Delta_k$ is the largest amount by which metric $k$ exceeds its bound over the rollout, and $\alpha_k$ is the magnitude of the upper bound of metric $k$.

# C Training Objective

During policy optimization, the final objective adds the joint and GRPO losses to the marginal objective $\mathcal{L}_{\mathrm{e2e}}$ for detection, mapping, planning, and marginal motion prediction:

$$\mathcal{L} = \mathcal{L}_{\mathrm{e2e}} + \lambda_{\mathrm{joint}}(\mathcal{L}_{\mathrm{joint_reg}} + \mathcal{L}_{\mathrm{joint_cls}}) + \lambda_{\mathrm{rl}}\mathcal{L}_{\mathrm{GRPO}}, \tag{9}$$

where $\lambda_{\mathrm{joint}}$ and $\lambda_{\mathrm{rl}}$ weight the joint supervision and the GRPO objective. The planning stage sets $\lambda_{\mathrm{rl}} = 0$. The marginal objective $\mathcal{L}_{\mathrm{e2e}}$ keeps actor-wise prediction reliable, ego-centric joint supervision targets the ego–agent dependencies that $\mathcal{L}_{\mathrm{e2e}}$ cannot express, and GRPO refines the ego policy with the planning-oriented reward (Appendix B).

# D Additional Experimental Details

## D.1 Dataset and Metrics

**Bench2Drive and Metrics.** Our primary evaluation uses Bench2Drive (Jia et al., 2024), a closed-loop benchmark designed to stress interactive driving. It provides 950 clips for training and 50 clips for open-loop validation, while closed-loop performance is measured on 220 predefined routes. We report the official metrics: Driving Score (DS), Success Rate (SR), Efficiency, and Comfortness. We further break performance down into the five driving abilities of Table 1: Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign. We conduct the ablations of Table 3 on Bench2Drive-mini (Tang et al., 2025), a 54-route subset that covers 43 scenario types (Appendix D.2).

**NAVSIM and Metrics.** We also evaluate on NAVSIM (Dauner et al., 2024), a planning-oriented benchmark based on data-driven non-reactive simulation. Under the official NAVSIM v1 protocol, each predicted ego trajectory is rolled out for 4 s, while background actors follow their logged future trajectories. The primary NAVSIM metric is the Predictive Driver Model Score (PDMS), which summarizes planning performance by combining safety-critical penalties with a weighted quality term (Dauner et al., 2024). We also report its component metrics: no at-fault collision (NC), drivable-area compliance (DAC), ego progress (EP), and time-to-collision within bound (TTC). NC and DAC capture safety and rule compliance, whereas EP and TTC reflect route progress and collision margin. The NAVSIM planning metrics also form the basis of the reward used in our policy optimization stage (Section 3.3 and Appendix B).

## D.2 Bench2Drive-mini for Ablation Studies

For closed-loop ablation experiments we use the Bench2Drive-mini split (Tang et al., 2025) of Bench2Drive (Jia et al., 2024). Evaluating every ablation on all 220 routes is computationally expensive, so we use a 54-route subset corresponding to roughly 25% of the full benchmark. The subset contains one representative route from each of 43 of the 44 Bench2Drive scenario types, together with 11 additional randomly selected routes. Table A1 lists the exact route IDs used for all Bench2Drive-mini ablations.

Table A2: Open-loop evaluation on the nuScenes (Caesar et al., 2020) *val* set under the VAD (Jiang et al., 2023) metric protocol. †: reproduced.

| Method | L2 (m) ↓ | | | | Collision (%) ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. |
| VAD-Base (2023) | 0.41 | 0.70 | 1.05 | 0.72 | 0.07 | 0.17 | 0.41 | 0.22 |
| GenAD (2024) | 0.28 | 0.49 | 0.78 | 0.52 | 0.08 | 0.14 | 0.34 | 0.19 |
| SparseDrive (2025) | 0.29 | 0.58 | 0.96 | 0.61 | 0.01 | 0.05 | 0.18 | 0.08 |
| DriveTransformer (2025) | 0.16 | 0.30 | 0.55 | 0.33 | 0.01 | 0.06 | 0.15 | 0.07 |
| MomAD (2025) | 0.31 | 0.57 | 0.91 | 0.60 | 0.01 | 0.05 | 0.22 | 0.09 |
| HiP-AD (2025) | 0.28 | 0.53 | 0.87 | 0.56 | 0.01 | 0.05 | 0.15 | 0.07 |
| HiP-AD† (2025) | 0.31 | 0.63 | 1.05 | 0.66 | 0.00 | 0.03 | 0.14 | 0.06 |
| **CastAD (Ours)** | 0.28 | 0.56 | 0.94 | 0.59 | 0.02 | 0.02 | 0.19 | 0.08 |

### D.3 Implementation Details

**Bench2Drive.** Our implementation uses a sparse query-based planner with multi-granularity ego queries, following the released baseline configuration (Tang et al., 2025) unless otherwise specified. Input images are resized to $640 \times 352$, and target points and high-level commands are embedded into the planner. The planner receives ego status during both training and inference. The spatial trajectory $\tau^e_{\text{sp}}$ is sampled at 2 m waypoint intervals. The predicted temporal and spatial trajectories drive longitudinal and lateral control, respectively; ego-centric joint supervision and policy optimization are used only during training. At inference, Agent-Mode Attention still exchanges information between the marginal and joint-mode embeddings. The output of the Ego-Centric Joint Prediction Head is not used for planning. Training runs in three stages: the detection module for 12 epochs, the planning module with ego-centric joint supervision for another 12 epochs, and policy optimization for 6 epochs. All Bench2Drive experiments run on 8 NVIDIA RTX A6000 GPUs with a total batch size of 48. We use $M = 6$ joint modes, $|\mathcal{K}_i| = 1$ marginal mode per agent in the interaction-set definition of Section 3.2, $G = 8$ rollouts per mode over a horizon of $T_{\text{fut}} = 6$ waypoints at 2 Hz, a clip range of $\epsilon = 0.2$, and $\lambda_{\text{rl}} = 0.3$. For $\lambda_{\text{joint}}$, the joint terms reuse the marginal loss weights: 1.0 for ego regression and 0.2 for ego classification and for agent regression and classification. The initial learning rate is $2 \times 10^{-4}$ for the detection and planning stages and $2 \times 10^{-5}$ for the policy optimization stage, with the AdamW optimizer (Loshchilov and Hutter, 2019) and a weight decay of 0.001. For the Bench2Drive-mini ablations of Table 3 we train all models under the same setting, with 6 epochs for the planning stage and 3 epochs for the policy optimization stage; Table 4 uses the full schedule.

**NAVSIM.** For NAVSIM we follow the implementation of DiffusionDrive (Liao et al., 2025). We use ResNet-34 (He et al., 2016) as the image encoder and add rasterized BEV LiDAR as input. Three front-view camera images are downsampled and concatenated into an input resolution of $1024 \times 256$. We train the model from scratch with ego-centric joint supervision on the `navtrain` split for 80 epochs using AdamW (Loshchilov and Hutter, 2019) with a learning rate of $6 \times 10^{-4}$. We then optimize it for 20 epochs with the policy optimization stage of Section 3.3.

## E Additional Quantitative Results

**Open-Loop Error Does Not Predict Closed-Loop Behavior.** In Table 1, DriveTransformer (Jia et al., 2025) attains the best Avg. L2 (0.62) yet only 63.46 Driving Score, whereas Hydra-NeXt (Li et al., 2025e), with a far larger displacement error (0.92), scores 73.86. CastAD's trajectory error of 0.68 is slightly below the 0.69 of its base method, yet it obtains the best Driving Score and Success Rate. Average displacement weights every frame equally. Lane-following frames likely outnumber the yielding and merging frames that our objective targets.

**Evaluation in the nuScenes Open-Loop Setting.** We also evaluate CastAD in the open-loop setting on nuScenes (Caesar et al., 2020), as shown in Table A2. Compared with the reproduced baseline (Tang et al., 2025), CastAD lowers the average L2 error (0.59 vs. 0.66) at a slightly higher average collision rate (0.08% vs. 0.06%). Open-loop metrics are known to correlate weakly with closed-loop driving (Dauner et al., 2023), so we read this table only as evidence that ego-centric joint supervision does not trade away open-loop imitation accuracy. The interaction-aware improvements that CastAD targets are measured by the closed-loop results of Table 1.

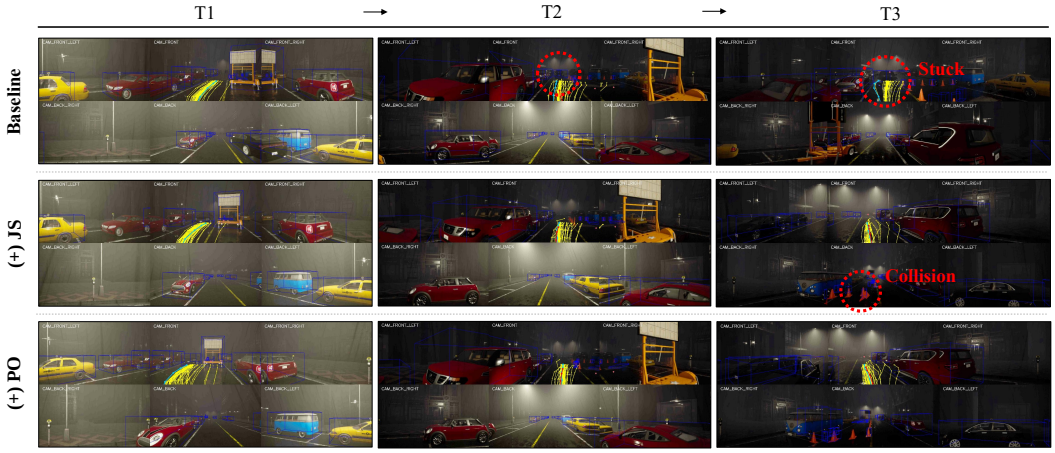


Figure A1: Qualitative examples of interaction-aware planning under module ablations, over consecutive frames in one interactive scenario. (+) JS adds ego-centric joint supervision to the baseline. (+) PO further adds policy optimization. Sky-blue: spatial trajectory; red: selected temporal trajectory; yellow: remaining temporal candidates.




Figure A2: Qualitative NAVSIM (Dauner et al., 2024) collision cases. Green: CastAD; red: DiffusionDrive (Liao et al., 2025); purple: ground truth. In BEV, every box other than the purple ego box at $t = 0$ is drawn at the same future step, and the collision-related object is orange.

## F Additional Qualitative Results

**Qualitative Comparison of Components.** Figure A1 compares qualitative behaviors under different module configurations. The baseline (Tang et al., 2025) moves into locally free space without accounting for its interaction with the surrounding agents and becomes stuck in front of another vehicle. With ego-centric joint supervision, which ties the relevant agent futures to the ego plan, the

planner handles the interaction better but can still fail, e.g., by colliding with a traffic cone. With policy optimization, the ego policy is further aligned using planning-oriented feedback, producing safer and more coherent behavior.

**NAVSIM Collision Cases.** Figure A2 shows four NAVSIM scenes in which DiffusionDrive collides with the orange object and CastAD avoids it.

## G LIMITATIONS

Three limitations bound the claims of this paper. First, interaction-relevant agents are those whose predicted futures overlap the ego's spatial trajectory. An agent that strongly influences the ego plan without such an overlap keeps its best marginal mode, so joint supervision does not yet capture the dependency between this agent and the ego. Second, CastAD has not yet been applied to real-world closed-loop driving: Bench2Drive runs in CARLA simulation, and NAVSIM scores plans on real-world logs without reactive agents. Third, the policy-optimization reward scores sampled ego rollouts against logged agent futures, which do not react to these rollouts. Our next step is reinforcement learning with reactive rollouts, in which each sampled ego plan shapes the behavior of surrounding agents.