Zero-Human Demonstration End-to-end Autonomous Driving with Trajectory Scorer

Zhenxin Li[1,2*], Nadine Chang[3], Wenhao Yao[1,2], Xinglong Sun[3], Zi Wang[3], Maying Shen[3], Jingde Chen[3], Jingyu Song[4], Kailin Li[5], Zuxuan Wu[1,2†], Shiyi Lan[3], and Jose M. Alvarez[3]

[1] Institute of Trustworthy Embodied AI, Fudan University
[2] Shanghai Key Laboratory of Multimodal Embodied AI
[3] NVIDIA
[4] University of Michigan
[5] East China Normal University

Abstract. Human demonstrations are widely considered the cornerstone of end-to-end (E2E) autonomous driving despite human demonstration's scarcity for long-tail and safety-critical scenarios. Nonetheless, current E2E autonomous driving (AD) training paradigms continue to rely on human demonstrations. Imitation learning (IL) requires human demonstrations for training, whereas reinforcement learning (RL) has emerged as a promising alternative to reduce this dependency. However, most existing RL methods for E2E AD still rely implicitly on human demonstrations. A pure rewards-based RL method can overcome the need for human demonstrations, but general RL policy gradient methods suffer from the cold-start problem. In this paper, we propose ZTRS (**Z**ero-human demonstration end-to-end autonomous driving with **TR**ajectory **S**corer) — a complete RL-based E2E planning paradigm trained solely on real-world images and rule-based rewards, entirely without human demonstration. Through our proposed **Exhaustive Policy Optimization (EPO)**, a policy gradient variant tailored for enumerable trajectory actions and dense supervision, ZTRS enables the model to generalize better to long-tail driving scenarios. We demonstrate this generalization through our SOTA performance against IL approaches on both long-tail Navhard and closed-loop HUGSIM datasets. Project page: `https://zhenxinli.net/ZTRS/`.

Keywords: End-to-end Autonomous Driving · Zero-Human Demonstration · Exhaustive Policy Optimization

1 Introduction

Recently, end-to-end (E2E) planners [7, 8, 20, 25, 35], which take in multi-sensor data and output trajectories directly, are the standard approach to autonomous driving (AD). Specifically, two predominant training paradigms have emerged:

*Work done during an internship at NVIDIA. †Corresponding author.

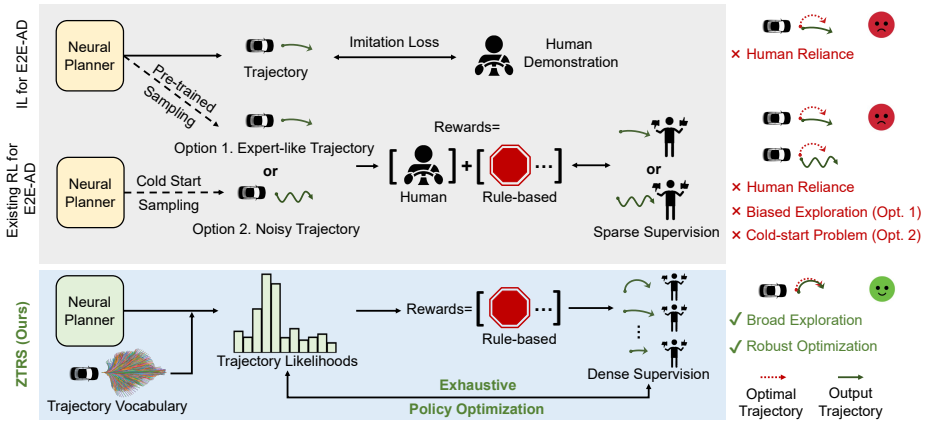

Fig. 1: Paradigms for Training E2E Neural Planners. While current IL and RL rely on human-biased supervision, ZTRS is the first paradigm using zero-human demonstrations for training E2E neural planners. We leverage a dense trajectory vocabulary and Exhaustive Policy Optimization (EPO) to provide dense reward-based supervision across all potential trajectories. EPO effectively resolves the exploration constraints of IL-pretrained models and the cold-start problem of traditional RL, enabling ZTRS to generalize to rare, long-tail scenarios that are scarce in human datasets.

Imitation Learning (IL) and Reinforcement Learning (RL). However, both E2E AD paradigms depend on human demonstrations, which are notably scarce in AD and especially so for safety-critical, long-tail scenarios. On one hand, IL relies exclusively on human demonstrations, leading to a number of additional critical issues: covariate shift [4] and causal confusion [59]. On the other hand, RL methods attempt to mitigate IL's issues by training with reward signals–whether by finetuning a pre-trained IL model [16, 30, 34, 42, 68, 69] or incorporating demonstrations into the reward signal, such as measuring L2 distances between planned and expert trajectories [16, 30, 42]. As a result, bias towards human demonstrations persists and restricts the model's ability to explore beyond the support of expert demonstrations. Ultimately, we encounter the main AD critical challenge in both IL and RL: inability to address safety-critical, long-tail cases where we do not have access to human demonstrations. Thus, in this work, we focus on rewards-based RL to overcome the reliance on human demonstrations.

Generally in RL, current policy gradient methods can be used off-the-self to train a pure rewards-based E2E RL planner without human demonstrations. However, employing common policy gradient methods–such as PPO [21, 48], REINFORCE [54], or GRPO [30, 50]–to generate continuous actions often leads to the common RL cold-start problem: without structured guidance, a random agent struggles to find high-reward trajectories in a large, continuous space [40]. Motivated by overreliance on human demonstration and cold-start limitation, we propose a fully from scratch RL-based **Z**ero-human demonstration end-to-end autonomous driving with **TR**ajectory **S**corer (ZTRS) to achieve: (1) a broad

exploration landscape to cover long-tail scenarios and (2) robust and stable optimization.

As shown in Fig. 1, ZTRS is a neural planner that selects actions from a trajectory vocabulary. The comprehensive nature of this vocabulary that covers nearly all driving actions has proven to improve trajectory generalization [7, 35, 44, 45]. Instead of relying on stochastic exploration, the vocabulary is constructed using cluster anchors from offline datasets [45]. Crucially, we note that we are not training with an explicit human demonstration for each sample, and the reconstructed vocabulary is a constant model input. Leveraging this vocabulary, we propose Exhaustive Policy Optimization (EPO) that trains the planner from scratch by performing an exhaustive evaluation across the entire vocabulary. This dense supervision approach ensures a higher probability of capturing sparse but high-reward trajectory patterns in large spaces, effectively resolving the cold-start problem. To keep the exhaustive evaluation tractable, we use rule-based metrics [3, 12, 31] as reward signals, which are both efficient and well-suited to E2E AD setting. Furthermore, we show that our dense supervision framework facilitates stable reward-driven learning. We also provide a theoretical analysis on EPO's viability to optimize a dense, full action space.

Critically, EPO demonstrates the following benefit: its ability to generalize to long-tail cases. Since EPO enumerates through nearly all driving actions, it allows ZTRS to learn trajectories often unavailable as expert demonstrations– rare trajectories critical for navigating challenging long-tail cases. This enables ZTRS to better generalize to long-tail test cases and achieve state-of-the-art (SOTA) performance on open-loop safety-critical Navhard. Importantly, ZTRS also achieves SOTA in closed-loop HUGSIM, demonstrating that ZTRS is not merely "hacking" open-loop rewards. Finally, while gaining long-tail generalizability, ZTRS also demonstrates that it can maintain SOTA performance on nominal driving. Our main contributions are as follows:

1. We introduce **ZTRS**, a fully RL-based neural planner for E2E AD that operates without depending on explicit human demonstration. This work demonstrates the feasibility to plan purely from rule-based rewards.
2. We propose **Exhaustive Policy Optimization (EPO)**, an offline reinforcement learning method tailored for enumerable trajectory actions, offering dense supervision and stable convergence.
3. We demonstrate ZTRS's safety-critical, long-tail generalization by achieving SOTA on open-loop Navhard and closed-loop HUGSIM. We further illustrate how EPO resolves cold-start problem as compared to other RL policy methods. Lastly, we show that we are nonetheless able to maintain SOTA performance on par with IL methods for generic open-loop planning, Navtest.

2 Related Work

2.1 Imitation Learning for E2E-AD

Given an offline expert dataset, Imitation Learning (IL) trains a policy to mimic expert behavior. In the end-to-end autonomous driving literature [6], early IL-

based approaches [9] are trained and tested in the closed-loop driving simulator CARLA [15]. Subsequent works improve the closed-loop driving performance with modern neural architectures (e.g. Transformers [56]) [8], intermediate representations [19, 24, 46, 49], and policy distillation [5, 23, 60, 66]. The introduction of UniAD [20] highlighted the strength of IL on real-world sensor data, whose complexity and diversity greatly exceed synthetic data produced by simulators. Building on this foundation, numerous efforts further focus on efficiency [25, 38], multi-modal behaviors [7, 38], vision-language understanding [34, 57], and safety constraints [35].

2.2 Reinforcement Learning for AD

Unlike Imitation Learning, Reinforcement Learning (RL) [53] trains agents to maximize rewards by interacting with the environment, and autonomous driving RL methods generally fall into two categories: symbolic-input methods and sensor-based methods, both relying on simulators. Symbolic-input methods [1, 10, 22, 32, 55, 66] use low-dimensional abstractions (e.g. 3D bounding boxes, maps, traffic signals) and have shown strong results on CARLA [15], nuPlan [18], and Waymax [17]. GigaFlow [10] and CaRL [22] both demonstrated that large-scale RL could train robust policies from scratch. On the other hand, sensor-based approaches operate on raw inputs like images. Early attempts [26] explored real-world training, but faced safety and efficiency challenges, while subsequent works [13, 41, 63] relied on simulated sensor data in CARLA. Despite success in simulators, such approaches could face sim-to-real gaps. Recently, RAD [16] fine-tuned a policy in a 3DGS simulator with high-dimensional real-world images, but still depended on human demonstrations for pre-training and reward computation. Similarly, several other approaches avoid the cold-start problem by fine-tuning an IL-pretrained diffusion policy [30, 34]. In contrast, our approach learns trajectory planning with reward signals from scratch, while operating on high-dimensional real-world sensor data.

2.3 End-to-end Trajectory Scorers

End-to-end Trajectory Scorers introduce a predefined vocabulary containing multiple trajectories, and scores each trajectory to select the most appropriate candidate as the output. Early works [7, 44, 45] score the trajectory candidates through classification based on their distance towards the ground-truth human trajectory. Beyond relying on a single human demonstration, the Hydra-MDP series [31, 35] proposed multi-target hydra-distillation to score trajectories with multiple rule-based metrics, leading to more robust planning capability. SafeFusion [58] synthesizes collision-related scenarios for training a robust planning model and eases the reliance on imitation learning. Other works further introduce multiple approaches to reach more precise and comprehensive trajectory scoring, such as test-time training [51], iterative refinement [64], and diffusion-based trajectory generation [36, 37]. Despite these advancements, these scorers still rely heavily on the imitation of human trajectories. In contrast, our approach fully discards expert demonstrations and achieves strong end-to-end planning performance via Reinforcement Learning.

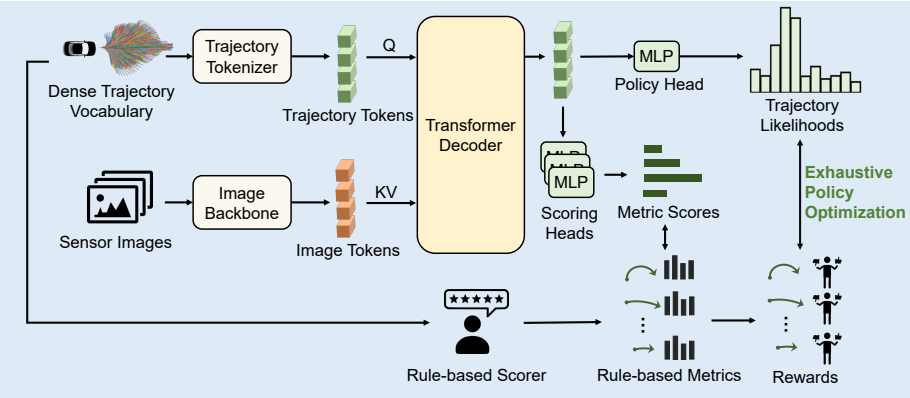

Fig. 2: The Overall Framework of ZTRS. Given offline sensor images and a fixed set of trajectories, ZTRS first tokenizes these two modalities. In a Transformer Decoder, the trajectory tokens attend to image tokens to acquire the context. Finally, scoring heads and a policy head map the trajectory tokens to metric scores and action likelihoods. All components are trained in an end-to-end manner.

3 Methodology

In this section, we elaborate on the framework and the Exhaustive Policy Optimization technique used in ZTRS.

3.1 Preliminary: Trajectory Scorers

As shown in Fig. 2, ZTRS is a trajectory scorer [31, 35–37, 51, 58, 64], whose functionality is to score a discrete vocabulary of trajectories $\mathcal{A} = \{a_i\}_{i=1}^{n}$ instead of regressing to a continuous trajectory. A dense, pre-collected vocabulary encompasses numerous trajectory candidates with natural driving patterns. In this formulation, the goal becomes selecting the most likely trajectory.

ZTRS consists of five modules: an image backbone, a trajectory tokenizer, a Transformer Decoder, a policy head, and several scoring heads. The policy head produces likelihoods for taking each action in $\mathcal{A}$, while the scoring heads predict the scores of rule-based driving metrics. For the choice of the metrics, we follow the practice of prior trajectory scorers (e.g. Hydra-MDP [35] and GTRS [37]) to use the Extended Predictive Driver Model Score (EPDMS, $\mathcal{E}$) [3, 12, 31]. EPDMS comprehensively evaluates multiple aspects of driving (e.g., safety, progress, and rule compliance) and can be efficiently computed for the trajectories in $\mathcal{A}$. Given a state s sampled from an offline dataset $\mathcal{D}$, where s contains sensor data and ego-vehicle status, the forward process involves three steps:

– The image backbone extracts L image tokens $\{x_{img}^{i}\}_{i=1}^{L}$ from a frontal-view image, while the trajectory tokenizer encodes trajectory candidates into queries $\{x_{traj}^{i}\}_{i=1}^{n}$.

- In the Transformer Decoder, the trajectory queries attend to image tokens.
- The policy head maps the attended trajectory queries $\{x^i_{traj}\}_{i=1}^n$ to likelihoods $\pi(\cdot|s)$, and m scoring heads map them to m rule-based scores $\{\mathcal{S}_i(\cdot|s)\}_{i=1}^m$.

During training, the scoring heads are trained with binary classification losses against $\mathcal{E}(s,\cdot)$, while the policy head is trained with Exhaustive Policy Optimization (See Sec. 3.3). At inference, the final trajectory $a \in \mathcal{A}$ is chosen using a weighted average of the likelihoods $\pi(\cdot|s)$ and predicted metric scores $\{\mathcal{S}_i(\cdot|s)\}_{i=1}^m$, following Hydra-MDP [31, 35].

3.2 Revisiting the Policy Gradient Theorem

To optimize a trajectory scorer with only rewards, we start with a simplified one-step policy optimization problem where the action space is a finite discrete set $\mathcal{A}$, and π is a policy parameterized by θ. In the online RL setting [53], the action a is sampled with $\pi(a|s)$, where s is the current state. In our offline RL setting [29], the state s is sampled from an offline dataset $\mathcal{D}$. Following the notation of GAE [47], the policy gradient g is defined as

$$g := \mathbb{E}\left[\Psi(s,a)\nabla_\theta \log \pi_\theta(a \mid s)\right], \tag{1}$$

where the advantage function $\Psi(s,a)$ can represent many quantities, such as the cumulative return or the state-action value function. The gradient can be equivalently written as

$$\begin{aligned} g &= \mathbb{E}\left[\Psi(s,a)\nabla_\theta \log \pi_\theta(a \mid s)\right] && (2)\\ &= \sum_{a' \in \mathcal{A}} \Psi(s,a')\pi_\theta(a' \mid s)\, \nabla_\theta \log \pi_\theta(a' \mid s) && (3)\\ &= \sum_{a' \in \mathcal{A}} \Psi(s,a')\pi_\theta(a' \mid s)\, \frac{\nabla_\theta \pi_\theta(a' \mid s)}{\pi_\theta(a' \mid s)} && (4)\\ &= \sum_{a' \in \mathcal{A}} \Psi(s,a')\nabla_\theta \pi_\theta(a' \mid s). && (5) \end{aligned}$$

This formulation matches the classical Policy Gradient Theorem [54]. Notably, the summation over all actions in $\mathcal{A}$ suggests that if the advantage function Ψ can be computed for each action at state s, policy optimization can be carried out directly on action likelihoods rather than log-likelihoods, as shown in Eq. 5.

3.3 Exhaustive Policy Optimization

When the action space $\mathcal{A}$ is a set of trajectories covering almost all driving possibilities, policy optimization can be formulated as maximizing the objective for an offline dataset $\mathcal{D}$:

$$\mathbb{E}_{s\sim\mathcal{D},\, a\sim\mathcal{A}}\left[\Psi(s,a)\right]. \tag{6}$$

This is consistent with the one-step policy optimization problem in Sec. 3.2 if Ψ is derived from open-loop reward signals, which do not require additional environment interaction. This objective can be optimized either in the log-likelihood form on a sampled action (Eq. 1) or in the likelihood form on the entire action space (Eq. 5). The latter formulation,

$$g := \sum_{\substack{a' \in \mathcal{A} \\ s \sim \mathcal{D}}} \Psi(s, a') \nabla_\theta \pi_\theta(a' \mid s) \tag{7}$$

provides much denser supervision for the policy as it explicitly enumerates all possibilities. Eq. 7 also defines our proposed *Exhaustive Policy Optimization (EPO)*, a variant of policy gradient tailored for offline data and enumerable actions. Specifically, EPO optimizes the policy by exhaustively considering every possible action $a \in \mathcal{A}$ and its respective advantage $\Psi(s, a)$. This formulation naturally enables broad exploration when the action space is $\mathcal{A}$ is dense. Meanwhile, EPO skips the sampling procedure, removing the need for IL pre-training and cold-start sampling. This dense, full-action-space optimization process is generally impractical in online RL but allows our framework to fully leverage off-policy trajectories for stable and exploratory learning. An overview of the pipeline is shown in Fig. 2.

To compute the advantage function Ψ without relying on human demonstrations, we again adopt the EPDMS metric $\mathcal{E}$ for its efficient and comprehensive evaluation of trajectories. Our empirical findings suggest that using the score $\mathcal{E}(s, \cdot)$ as $\Psi(s, \cdot)$ enables better long-tail, closed-loop generalization than baselines trained with human demonstrations. Furthermore, the scores $\mathcal{E}(s, \cdot)$ can also be reused for each state $s \in \mathcal{D}$ as long as the action space is fixed throughout the training process, which greatly improves training efficiency. To further enforce temporal consistency, we subtract a correction term $b(s_t, a_t, a_{t-1}) = \lambda \mathbb{1}\left[\mathrm{EC}(a_{t-1}, a_t)\right]$, where λ is a constant, $a_{t-1} = \underset{a}{\mathrm{argmax}} \pi(a|s_{t-1})$, and EC indicates the violation of Extended Comfort (EC) thresholds [31]:

$$\Psi(s_t, a_t) = \mathcal{E}(s_t, a_t) - b(s_t, a_t, a_{t-1}). \tag{8}$$

Note that this formulation penalizes inconsistent predictions more strongly than the one used in [3, 31]. Finally, Ψ is normalized to zero mean and unit variance following [21, 50].

4 Experiments

In this section, we quantitatively show that ZTRS can generalize to long-tail and safety-critical scenarios. Through our ablation studies, we show the necessity of our design choices. Finally, we qualitatively illustrate what type of long-tail and safety-critical cases ZTRS addresses.

4.1 Datasets and metrics

Datasets. We conduct experiments on three benchmarks covering open-loop, closed-loop, and long-tail testing: Navhard [3], HUGSIM [67], and Navtest [12].

Our first dataset is Navhard, its long-tail and safety-critical scenarios makes it a suitable dataset to evaluation long-tail generalization. Navhard [3] uses pseudo-simulation on the long-tail challenging scenarios in NAVSIM, resulting in the total evaluation to a two-stage paradigm: 1) first stage follows the original NAVSIM evaluation, 2) second stage adopts 3D Gaussian Splatting (3DGS) [27, 33] to synthesize subsequent driving scenarios, resulting in 244 initial scenarios and 4164 synthetic scenarios.

HUGSIM [67] is a closed-loop driving benchmark that does not contain human demonstrations; thus an appropiate dataset to test non-human demonstration method. It features images synthesized with 3DGS and integrates multiple driving datasets, including KITTI-360 [39], Waymo [52], nuScenes [2], and Pandaset [61], into a collection of 345 driving scenarios. These scenarios are categorized by difficulty into four levels: easy, medium, hard, and extreme. Specifically, HUGSIM released 49 easy scenarios for regular driving, 126 medium scenarios with inserted vehicles, and 86 hard scenarios as well as 84 extreme scenarios with aggressive vehicles.

Metrics. Navhard and Navtest evaluate open-loop planning with EPDMS $\mathcal{E}(s, a)$, which is defined as an aggregation of multiple driving metrics:

$$\mathcal{E}(s,a) = \left(\prod_{m \in S_{\text{pen}}} m(s,a) \right) \cdot \left(\frac{\sum_{m \in S_{\text{avg}}} w_m \, m(s,a)}{\sum_{m \in S_{\text{avg}}} w_m} \right) \tag{9}$$

where s is the current state and a is a 4-second trajectory. The penalty metric set S_{pen} is applied multiplicatively and includes No-at-fault Collisions (NC), Drivable Area Compliance (DAC), Driving Direction Compliance (DDC), and Traffic Light Compliance (TLC), while the weighted metric set S_{avg} contains Time-to-Collision (TTC), Ego Progress (EP), Lane Keeping (LK), and History Comfort (HC). The extended comfort (EC) from [31] is also used as a weighted metric to promote temporally consistent driving. w_m is the aggregation weight for metric m. Note that human filtering is used in [3] for calculating $\mathcal{E}$ but not in our training. For HUGSIM, HD-Score is used for closed-loop evaluation. It aggregates Route Completion (RC) with the sub-metrics NC, DAC, TTC, HC across an episode of length T:

$$\text{HD-Score} = RC \cdot \sum_{t=1}^{T} \left[\left(\prod_{m \in \{NC, DAC\}} m(s_t, \tilde{a}_t) \right) \cdot \left(\frac{\sum_{m \in \{TTC, HC\}} w_m \, m(s_t, \tilde{a}_t)}{\sum_{m \in \{TTC, HC\}} w_m} \right) \right], \tag{10}$$

where $\tilde{a}$ is the ego-vehicle acceleration and steering angle transformed from a trajectory.

Navtest [12] is the common open-loop evaluation dataset for NAVSIM. We evaluate on this dataset to showcase ZTRS' ability to handle more nominal driving. NAVSIM contains 103k and 12k diverse and challenging driving scenarios for

model training (Navtrain) and evaluation (Navtest), and introduces simulation-based metrics to better review closed-loop planning capability through open-loop evaluation. During evaluation, the output trajectory is evaluated by a simulator to get rule-based simulation metric scores related to multiple driving aspects, such as traffic rule compliance, comfort, and progress.

4.2 Implementation Details

All our models are trained on the Navtrain split with 24 NVIDIA A100 GPUs, while the synthetic data from Navhard and HUGSIM are not used for training. Models are trained for 15 epochs with a total batch size of 528, using a learning rate and weight decay of 2×10^{-4} and 0.0. The frontal view with center-cropped front-left and front-right views are concatenated as the input image, which is then resized to 512×2048. The hyperparameter λ in the correction term b is set to 0.2. By default, the action space used in our method has 16384 trajectories, each spanning 4 seconds at 10Hz. These trajectories are obtained through K-means clustering on the nuPlan dataset [18]. Following [31, 35, 64], we default to use the DD3D-pretrained [43] V2-99 [28] as the image backbone in our experiments. The ViT-L [14] backbone is pretrained from Depth-Anything [62].

4.3 Main Results

Navhard is a difficult long-tail benchmark as its distribution widely differs from training split distribution. Thus, overcoming this distribution mismatch with strong dataset performance highlights how a method can generalize from one distribution to another. In Tab. 1, we quantitatively that ZTRS' dense supervision with EPO can generalize to even Navhard's distribution containing safety-critical scenarios. ZTRS achieves SOTA performance across various settings. Equipped with V2-99 and EVA-ViT-L backbones, ZTRS significantly outperforms the strong GTRS-Dense baseline by +3.8 EPDMS and +2.8 EPDMS, respectively. In particular, these gains are achieved despite the use of additional regularization and dynamic trajectory candidates in GTRS-Dense, which are not required by our framework.

HUGSIM is a safety-critical closed-loop dataset where human demonstrations are not available. In Tab. 2, we show how ZTRS, even trained with open-loop rewards, can transfer successfully to a closed-loop setting and overcome the training lack of human demonstrations. ZTRS demonstrates superior robustness in closed-loop, achieving SOTA results on the HUGSIM benchmark with an overall 42.6 RC and 28.9 HD-Score. Notably, these results are achieved through zero-shot closed-loop transfer, as the model is trained purely on the open-loop Navtrain dataset without exposure to HUGSIM's synthetic interactive scenarios. Achieving better closed-loop performance than its IL baseline GTRS-Dense indicates that ZTRS is not learning to "hack" open-loop rewards through EPO. Instead, EPO facilitates generalization to closed-loop settings.

Navtest. Finally, we show in Tab. 3 that ZTRS performs on par with SOTA IL planners in nominal driving scenarios found in Navtest. We note we are able

Table 1: Performance on the Navhard Benchmark. ZTRS achieves SOTA results, demonstrating our ability to generalize to different distributions, including safety-critcal cases in Navhard. We note that PDM-Closed uses ground-truth symbolic inputs for planning. *H-Demo: human demonstration.*

Backbone	Method	H-Demo	Inputs	Planner Type	EPDMS ↑
-	PDM-Closed [11]	-	Privileged Info.	Rule-based Planner	51.3
ResNet34	LTF [8]	✓	Image	Regression-based Planner	23.1
	DiffusionDrive [38]	✓	LiDAR+Image	Diffusion-based Planner	27.5
	ZTRS (Ours)	✗	Image	Trajectory Scorer	**39.9**
V2-99	DriveSuprim [64]	✓	Image	Trajectory Scorer	42.1
	GTRS-Dense [37]	✓	Image	Trajectory Scorer	41.7
	ZTRS (Ours)	✗	Image	Trajectory Scorer	**45.5**
EVA-ViT-L	DriveSuprim [64]	✓	Image	Trajectory Scorer	44.7
	GTRS-Dense [37]	✓	Image	Trajectory Scorer	43.4
	ZTRS (Ours)	✗	Image	Trajectory Scorer	**46.2**

Backbone	Method	H-Demo	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑
Stage 1: Real-world Scenarios											
-	PDM-Closed [11]	-	94.4	98.8	100	99.5	100	93.5	99.3	87.7	36.0
ResNet34	LTF [8]	✓	97.3	80.2	97.8	99.3	83.4	96.2	92.9	**97.8**	71.1
	DiffusionDrive [38]	✓	96.8	86.0	98.8	99.3	**84.0**	95.8	**96.7**	97.6	**79.6**
	ZTRS (Ours)	✗	**98.4**	**92.9**	**99.3**	**100.0**	62.8	**98.7**	93.3	97.3	43.1
V2-99	DriveSuprim [64]	✓	98.9	95.1	99.2	99.6	**76.1**	**99.1**	94.7	97.6	**54.2**
	GTRS-Dense [37]	✓	98.7	95.8	99.4	99.3	72.8	98.7	95.1	96.9	40.4
	ZTRS (Ours)	✗	**98.9**	**97.6**	**100.0**	**100.0**	66.7	98.9	**96.2**	96.7	44.0
EVA-ViT-L	DriveSuprim [64]	✓	98.7	98.0	99.1	99.8	75.9	98.7	94.7	**97.6**	**49.8**
	GTRS-Dense [37]	✓	97.6	95.8	99.7	99.8	**77.2**	97.8	**95.3**	97.3	46.7
	ZTRS (Ours)	✗	**99.1**	**98.9**	**100.0**	**100.0**	65.6	**98.9**	94.4	95.8	38.2

Backbone	Method	H-Demo	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑
Stage 2: Synthetic Scenarios											
-	PDM-Closed [11]	-	88.1	90.6	96.3	98.5	100	83.1	73.7	91.5	25.4
ResNet34	LTF [8]	✓	79.4	69.0	85.6	98.5	83.8	76.7	47.9	97.0	70.6
	DiffusionDrive [38]	✓	80.1	72.8	84.4	98.4	**85.9**	76.6	46.4	96.3	**72.8**
	ZTRS (Ours)	✗	**91.8**	**88.3**	**96.2**	**98.9**	54.6	**90.2**	**56.4**	**98.2**	63.0
V2-99	DriveSuprim [64]	✓	87.9	88.8	89.6	98.8	**80.3**	86.0	53.5	97.1	56.1
	GTRS-Dense [37]	✓	**91.4**	89.2	94.4	98.8	69.5	**90.1**	54.6	94.1	49.7
	ZTRS (Ours)	✗	91.1	**90.4**	**95.8**	**99.0**	63.6	89.8	**60.4**	**97.6**	**66.1**
EVA-ViT-L	DriveSuprim [64]	✓	89.5	89.6	92.9	98.5	**78.9**	86.4	55.3	96.5	52.7
	GTRS-Dense [37]	✓	**91.9**	91.3	92.7	**99.0**	72.7	90.4	53.8	94.1	41.6
	ZTRS (Ours)	✗	91.2	**94.4**	**96.3**	98.7	59.4	**90.5**	**57.3**	**97.5**	**63.0**

to achieve SOTA despite the fact that reported metrics include ego progress (EP), which measures the L2 distance to the expert trajectories, none of which we ever trained with. Thus, we see a notable gap in our EP score compared to the IL methods. Even so, with a lightweight ResNet34 backbone, ZTRS achieves an EPDMS of 83.2, outperforming models with advanced trajectory scoring techniques like DriveSuprim [64]. With higher-capacity backbones like ViT-L and V2-99, ZTRS maintains competitive results (86.2 EPDMS and 85.3 EPDMS) against DriveSuprim and surpasses its IL baseline GTRS-Dense [37] by a clear margin (+1.5 EPDMS and +1.3 EPDMS).

Table 2: Zero-shot Performance on the HUGSIM Benchmark. UniAD and VAD are trained on nuScenes [2], which partially overlap with the evaluation scenarios. Other methods are evaluated with zero-shot transfer. *H-Demo: human demonstration.*

Method	H-Demo	Zero-shot	Easy		Medium		Hard		Extreme		Overall	
			RC ↑	HD-Score ↑	RC ↑	HD-Score ↑	RC ↑	HD-Score ↑	RC ↑	HD-Score ↑	RC ↑	HD-Score ↑
VAD [25]	✓	✗	38.7	24.3	27.0	9.9	25.5	10.4	23.0	8.2	27.9	12.3
UniAD [20]	✓	✗	58.6	48.7	41.2	29.5	**40.4**	**27.3**	**26.0**	**14.3**	40.6	28.9
LTF [8]	✓	✓	68.4	52.8	40.7	24.6	36.9	19.8	25.5	8.1	41.4	24.8
GTRS-Dense [37]	✓	✓	62.0	57.8	40.5	35.2	23.9	16.2	13.7	5.8	34.5	28.2
ZTRS (Ours)	✗	✓	**74.8**	**66.9**	**46.1**	**37.9**	31.6	21.1	20.0	9.8	**42.0**	**32.9**

Table 3: Performance on the Navtest Benchmark. ZTRS performs on par with SOTA IL planners in nominal scenarios. Notably, while Imitation Learning methods relying on huamn demonstration (H-Demo) achieve higher Ego Progress (EP) by mimicking human trajectories, ZTRS remains highly competitive across most metrics despite never being trained on such trajectories.

Backbone	Method	H-Demo	NC ↑	DAC ↑	DDC ↑	TL ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
-	Human Agent	-	100	100	99.8	100	87.4	100	100	98.1	90.1	90.3
-	Ego Status MLP	✓	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0
ResNet34	Transfuser [8]	✓	96.9	89.9	97.8	99.7	87.1	95.4	92.7	**98.3**	**87.2**	76.7
	HydraMDP++ [31]	✓	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
	DriveSuprim [64]	✓	97.5	96.5	99.4	99.6	**88.4**	96.6	**95.5**	**98.3**	77.0	83.1
	ZTRS (Ours)	✗	**97.9**	**98.0**	**99.7**	**99.8**	80.1	**96.9**	94.9	98.1	79.0	**83.2**
ViT-L	HydraMDP++ [31]	✓	98.5	98.5	99.5	99.7	87.4	97.9	95.8	98.2	75.7	85.6
	DriveSuprim [64]	✓	98.4	98.6	99.6	99.8	**90.5**	97.8	**97.0**	**98.3**	**78.6**	**87.1**
	GTRS-Dense [37]	✓	**99.0**	97.8	99.3	**99.9**	86.3	**98.3**	95.1	**98.3**	71.9	84.7
	ZTRS (Ours)	✗	98.2	**99.1**	**99.7**	99.8	86.9	97.5	96.6	98.2	78.2	86.2
V2-99	HydraMDP++ [31]	✓	**98.4**	98.0	99.4	99.8	87.5	**97.7**	95.3	**98.3**	77.4	85.1
	DriveSuprim [64]	✓	97.8	97.9	99.5	**99.9**	**90.6**	97.1	96.6	**98.3**	**77.9**	**86.0**
	GTRS-Dense [37]	✓	97.6	98.5	99.5	**99.9**	89.5	97.2	**96.8**	97.2	57.2	84.0
	ZTRS (Ours)	✗	97.8	**99.4**	**99.8**	99.8	84.1	97.0	96.2	98.2	77.2	85.3

4.4 Ablation Studies

RL Policy Gradient Comparisons. We revisit the common cold-start RL problem, where policy gradient methods struggle to find high-rewards in an large continuous space without proper guidance. This is especially visible in early training stages. As such, we analyze our EPO against other policy gradient methods in the beginning epochs, as seen in Fig. 3. We evaluate against three policy methods within a discrete trajectory scoring framework: REINFORCE [54] and two common settings of known high-performer GRPO [50]. To ensure fair comparison, all baselines sample from the same trajectory vocabulary. In Fig. 3, we observe that EPO demonstrates stable, non-volatile performance on both Navhard and Navtest. Its strong initial performance illustrates how EPO's dense supervision enables us to successfully find high-reward trajectories. Conversely, REINFORCE shows noticeably weaker performance, especially on Navtest. Importantly, although GRPO is widely acknowledged for its performance delivery, we not only see both GRPO settings' poor performance but also their instability

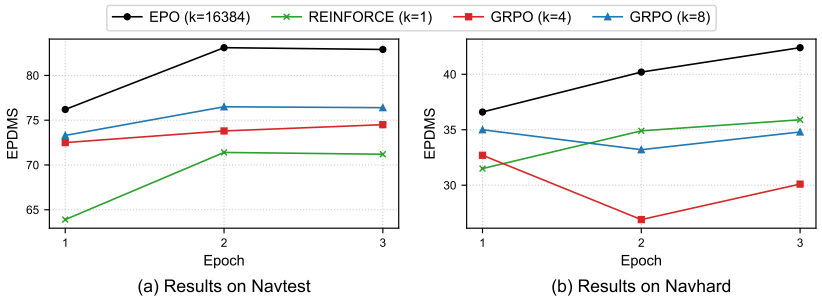

Fig. 3: Comparisons on Different RL Policy Gradient Methods. k: Group size for sampled trajectories per training iteration. EPO enables stable optimization on both nominal driving (Navtest) and long-tail cases (Navhard). EPO's early out performance against baselines indicate that we are able to find high-reward trajectories early on, illustrating our ability to mitigate the cold-start problem.

Table 4: Ablation study on different learning methods and targets. $\hat{\mathcal{E}}$ represents using the trajectory with the maximum ground-truth EPDMS as the imitation target. The Extended Comfort (EC) metric highlights that incorporating the correction term b into the reward function is essential for achieving temporal consistency.

IL	RL	Target	NC ↑	DAC ↑	DDC ↑	TL ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
✓	✗	Human	98.5	98.7	98.9	99.9	88.5	98.2	97.0	98.3	80.5	86.2
✓	✗	$\hat{\mathcal{E}}$	96.6	96.8	99.5	99.6	88.3	96.0	96.7	92.2	18.5	76.7
✗	✓	$\mathcal{E}$	97.5	99.2	**99.8**	99.7	**89.3**	96.9	**96.8**	98.0	53.8	84.2
✗	✓	$\mathcal{E} - b$	**97.8**	**99.4**	**99.8**	**99.8**	84.1	**97.0**	96.2	**98.2**	**77.2**	**85.3**

through their performance fluctuations, which is undesirable due to potential model collapse. This is likely due to GRPO's unrepresentative subset trajectory sampling and the prevalence of negative gradients in sampled trajectory groups.

Learning Paradigms Comparison. In Tab. 4, we study the effects of different learning paradigms and targets. When using the trajectory with the maximum EPDMS score (i.e., $\hat{\mathcal{E}} = \underset{a}{\text{argmax}}\mathcal{E}(s,a)$) as the imitation target, performance drops significantly compared to the IL baseline, as many trajectories in $\mathcal{A}$ can achieve high EPDMS and a single target fails to capture the underlying pattern. Using likelihoods over the entire action space mitigates this issue, improving EPDMS by 7.5%, but introduces serious oscillation, as indicated by the low EC metric. This highlights the need for our correction term b to enforce temporal consistency. Using $\mathcal{E} - b$ as the reward increases EC by 23.4%.

Action Space Size Ablation. Tab. 5 shows the relationship between the size of the action space and evaluation data. Models using the full action space during inference achieve the best results on real-world data, as reflected by EPDMS on Navtest and Navhard, while shrinking the action space improves performance on the simulated portion. This finding is consistent with GTRS [37]:

Table 5: The relationship between the size of the action space and evaluation data. $EPDMS_1$ measures the Stage 1 real-world scenarios of Navhard, while $EPDMS_2$ measures the Stage 2 simulated scenarios.

Backbone	$\|\mathcal{A}\|$ for training	$\|\mathcal{A}\|$ for inference	Navtest	Navhard		
			EPDMS ↑	$EPDMS_1$ ↑	$EPDMS_2$ ↑	EPDMS ↑
V2-99	8192	8192	84.6	73.3	57.4	43.0
	16384	16384	**85.3**	**74.9**	57.1	43.4
	16384	8192	82.0	74.2	**60.7**	**45.5**
ViT-L	8192	8192	84.6	73.7	55.9	41.9
	16384	16384	**86.2**	**76.1**	50.5	38.8
	16384	8192	84.3	73.4	**59.9**	**45.0**

Table 6: Planning Performance as a Function of Training Initialization on Navtest. Results across different initialization seeds show that our Exhaustive Policy Optimization (EPO) exhibits significantly lower variance and higher overall performance, indicating that our offline RL approach is stable and robust to initialization.

Policy Optimization Method	**Backbone**	**Seed**	**EPDMS ↑**
REINFORCE	V2-99	0	**75.0**
		1	72.6
EPO (ours)	V2-99	0	85.3
		1	**85.5**

regardless of whether the model is trained with human demonstrations, reducing model complexity enhances generalization on unseen simulated data.

Sensitivity to Training Initialization. Here, we evaluate the training stability and variance of our proposed Exhaustive Policy Optimization (EPO) against the classical policy gradient method, REINFORCE. As shown in Tab. 6, the REINFORCE baseline exhibits significant performance fluctuations across different initialization seeds (75.0 vs. 72.6 EPDMS), highlighting the randomness of stochastic sampling in a large trajectory space. In contrast, EPO achieves substantially higher performance and stability, yielding a marginal variance of 0.2 EPDMS. Since EPO performs policy updates by evaluating a comprehensive and fixed trajectory vocabulary exhaustively at each training step, it provides a more deterministic optimization landscape and reduces such fluctuations.

4.5 Qualitative Results

Fig. 4 and Fig. 5 show visualization results on the open-loop Navtest and the closed-loop HUGSIM, respectively. Interestingly, although ZTRS is trained without human demonstrations, ZTRS learns driving patterns that resemble human trajectories from rule-based rewards in nominal driving scenarios, such as long-term path following. On the other hand, when facing complex urban interactions, as shown in Fig. 4 (d), ZTRS can crucially make different but safer trajectory plans than the human demonstration. In the red box's leftmost scenario, while both the human and the agent navigate around a cyclist, ZTRS exhibits a more

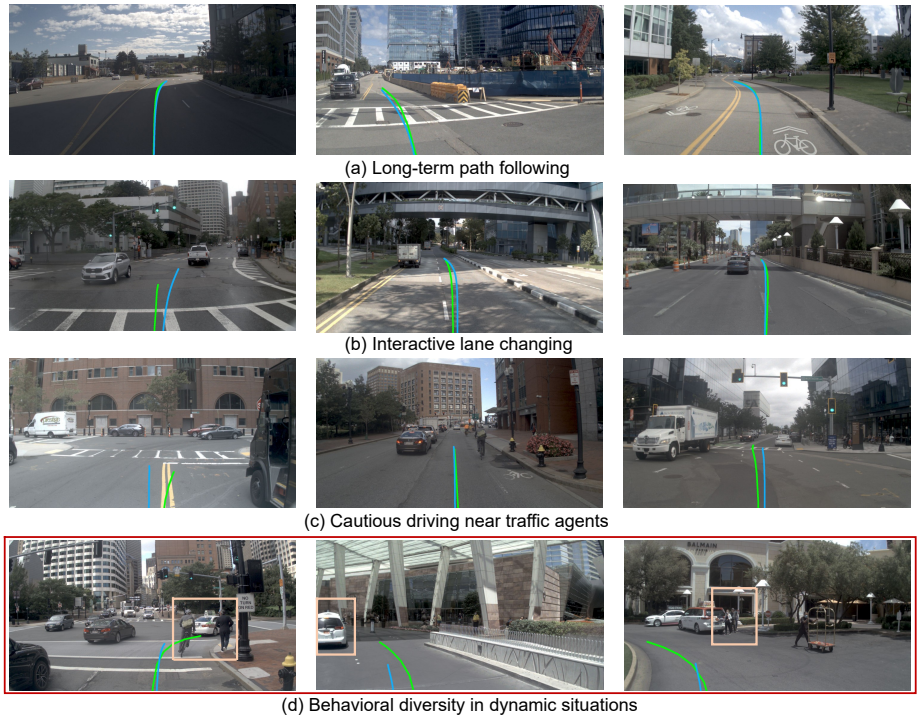

Fig. 4: Visualizations of planned trajectories and the human trajectory on Navtest. In nominal scenarios, ZTRS plans similarly to humans, exhibiting long-term path following. In complex cases, ZTRS demonstrates sophisticated behaviors such as interactive lane changing, cautious navigation near traffic agents, and behavioral diversity in dynamic situations where it plans differently but more safely than humans.

pronounced nudging behavior, providing even more lateral clearance than the human driver to ensure safety. Further, the second scenario shows that ZTRS decelerates after detecting the cut-in behavior of the leading vehicle. In the third scenario, ZTRS shows it can anticipate the luggage handlers' movements, opting for a more conservative path than that of the human driver.

Moreover, ZTRS manages to navigate safely in safety-critical driving scenarios without closed-loop training or adaptation, as shown in Fig. 5. Even under the extreme conditions depicted in Fig. 5 (c), where the ego agent must overtake a parked car while facing an oncoming vehicle, ZTRS can safely complete the route. This demonstrates the strong closed-loop driving ability and the robust safety margin maintained by ZTRS in highly complex long-tail situations.

5 Conclusion

We proposed ZTRS, an effective end-to-end autonomous driving paradigm that eliminates the need for human demonstrations during training. By relying solely

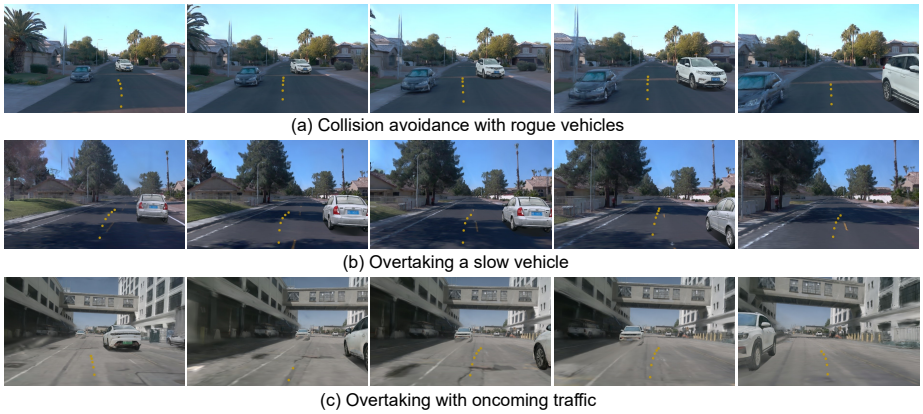

Fig. 5: Visualizations of planned trajectories on the challenging closed-loop driving benchmark HUGSIM. Here, we illustrate ZTRS can plan trajectories even in the shown long-tail scenarios, including safety critical, collision avoidance cases.

on offline data, reward-driven supervision, and a newly proposed exhaustive policy optimization (EPO), ZTRS successfully addresses the cold-start problem while enabling robust planning directly from high-dimensional sensor inputs, without human demonstrations. Our approach achieves strong SOTA performance on long-tail driving benchmarks such as NAVHARD, while maintaining top-level results on standard (nominal) driving benchmarks such as NAVSIM and HUGSIM. These findings demonstrate that well-designed reward mechanisms can potentially replace human demonstrations in training reliable end-to-end autonomous driving systems.

Limitations and Discussions. Despite ZTRS achieving state-of-the-art performance on challenging planning and closed-loop driving benchmarks, it still lags behind imitation-based baselines in certain open-loop scenarios. EPO is also most direct when the planner acts over a fixed, enumerable trajectory vocabulary, since this allows rewards to be pre-computed and reused during training. Extending the same idea to continuous action spaces is therefore an important direction for future work. One practical route is to connect continuous trajectory generators with the discrete reward table through nearest-neighbor lookup, as suggested by recent continuous end-to-end planners [65].

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 62427819) and the Science and Technology Commission of Shanghai Municipality (No. 24511103100).

References

[1] Booher, J., Rohanimanesh, K., Xu, J., Isenbaev, V., Balakrishna, A., Gupta, I., Liu, W., Petiushko, A.: Cimrl: Combining imitation and reinforcement learning for safe autonomous driving. arXiv preprint arXiv:2406.08878 (2024)

[2] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)

[3] Cao, W., Hallgarten, M., Li, T., Dauner, D., Gu, X., Wang, C., Miron, Y., Aiello, M., Li, H., Gilitschenski, I., et al.: Pseudo-simulation for autonomous driving. CoRL (2025)

[4] Chang, J., Uehara, M., Sreenivas, D., Kidambi, R., Sun, W.: Mitigating covariate shift in imitation learning via offline data with partial coverage. NeurIPS **34**, 965–979 (2021)

[5] Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: CoRL. pp. 66–75. PMLR (2020)

[6] Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[7] Chen, S., Jiang, B., Gao, H., Liao, B., Xu, Q., Zhang, Q., Huang, C., Liu, W., Wang, X.: Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. arXiv preprint arXiv:2402.13243 (2024)

[8] Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[9] Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: ICRA. pp. 4693–4700. IEEE (2018)

[10] Cusumano-Towner, M., Hafner, D., Hertzberg, A., Huval, B., Petrenko, A., Vinitsky, E., Wijmans, E., Killian, T., Bowers, S., Sener, O., et al.: Robust autonomy emerges from self-play. arXiv preprint arXiv:2502.03349 (2025)

[11] Dauner, D., Hallgarten, M., Geiger, A., Chitta, K.: Parting with misconceptions about learning-based vehicle motion planning. In: Conference on Robot Learning. pp. 1268–1281. PMLR (2023)

[12] Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., Geiger, A., Chitta, K.: Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In: NeurIPS (2024)

[13] Delavari, E., Khanzada, F.K., Kwon, J.: A comprehensive review of reinforcement learning for autonomous driving in the carla simulator. arXiv preprint arXiv:2509.08221 (2025)

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[15] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
[16] Gao, H., Chen, S., Jiang, B., Liao, B., Shi, Y., Guo, X., Pu, Y., Yin, H., Li, X., Zhang, X., et al.: Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. arXiv preprint arXiv:2502.13144 (2025)
[17] Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., et al.: Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. NeurIPS **36**, 7730–7742 (2023)
[18] H. Caesar, J. Kabzan, K.T.e.a.: Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In: CVPR ADP3 workshop (2021)
[19] Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., Shotton, J.: Model-based imitation learning for urban driving. NeurIPS **35**, 20703–20716 (2022)
[20] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR. pp. 17853–17862 (2023)
[21] Huang, S., Dossa, R.F.J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., AraÃšjo, J.G.: Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. Journal of Machine Learning Research **23**(274), 1–18 (2022)
[22] Jaeger, B., Dauner, D., Beißwenger, J., Gerstenecker, S., Chitta, K., Geiger, A.: Carl: Learning scalable planning policies with simple rewards. arXiv preprint arXiv:2504.17838 (2025)
[23] Jia, X., Gao, Y., Chen, L., Yan, J., Liu, P.L., Li, H.: Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In: ICCV. pp. 7953–7963 (2023)
[24] Jia, X., Wu, P., Chen, L., Xie, J., He, C., Yan, J., Li, H.: Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In: CVPR. pp. 21983–21994 (2023)
[25] Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)
[26] Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.M., Lam, V.D., Bewley, A., Shah, A.: Learning to drive in a day. In: ICRA. pp. 8248–8254. IEEE (2019)
[27] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
[28] Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 0–0 (2019)

[29] Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020)
[30] Li, D., Ren, J., Wang, Y., Wen, X., Li, P., Xu, L., Zhan, K., Xia, Z., Jia, P., Lang, X., et al.: Finetuning generative trajectory model with reinforcement learning from human feedback. arXiv preprint arXiv:2503.10434 (2025)
[31] Li, K., Li, Z., Lan, S., Xie, Y., Zhang, Z., Liu, J., Wu, Z., Yu, Z., Alvarez, J.M.: Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. arXiv preprint arXiv:2503.12820 (2025)
[32] Li, Q., Jia, X., Wang, S., Yan, J.: Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In: ECCV. pp. 142–158. Springer (2024)
[33] Li, T., Qiu, Y., Wu, Z., Lindström, C., Su, P., Nießner, M., Li, H.: Mtgs: Multi-traversal gaussian splatting. arXiv preprint arXiv:2503.12552 (2025)
[34] Li, Y., Xiong, K., Guo, X., Li, F., Yan, S., Xu, G., Zhou, L., Chen, L., Sun, H., Wang, B., et al.: Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. arXiv preprint arXiv:2506.08052 (2025)
[35] Li, Z., Li, K., Wang, S., Lan, S., Yu, Z., Ji, Y., Li, Z., Zhu, Z., Kautz, J., Wu, Z., et al.: Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. arXiv preprint arXiv:2406.06978 (2024)
[36] Li, Z., Wang, S., Lan, S., Yu, Z., Wu, Z., Alvarez, J.M.: Hydra-next: Robust closed-loop driving with open-loop training. arXiv preprint arXiv:2503.12030 (2025)
[37] Li, Z., Yao, W., Wang, Z., Sun, X., Chen, J., Chang, N., Shen, M., Wu, Z., Lan, S., Alvarez, J.M.: Generalized trajectory scoring for end-to-end multimodal planning. arXiv preprint arXiv:2506.06664 (2025)
[38] Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., et al.: Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In: CVPR. pp. 12037–12047 (2025)
[39] Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE TPAMI **45**(3), 3292–3310 (2022)
[40] Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Overcoming exploration in reinforcement learning with demonstrations. In: ICRA. pp. 6292–6299. IEEE (2018)
[41] Nehme, G., Deo, T.Y.: Safe navigation: Training autonomous vehicles using deep reinforcement learning in carla. arXiv preprint arXiv:2311.10735 (2023)
[42] Noguchi, C., Yamamoto, T.: Offline reinforcement learning for end-to-end autonomous driving. arXiv preprint arXiv:2512.18662 (2025)
[43] Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)
[44] Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: Covernet: Multimodal behavior prediction using trajectory sets. In: CVPR. pp. 14074–14083 (2020)

[45] Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV. pp. 194–210. Springer (2020)
[46] Renz, K., Chitta, K., Mercea, O.B., Koepke, A.S., Akata, Z., Geiger, A.: Plant: Explainable planning transformers via object-level representations. In: CoRL (2022)
[47] Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
[48] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
[49] Shao, H., Wang, L., Chen, R., Li, H., Liu, Y.: Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In: CoRL. pp. 726–737. PMLR (2023)
[50] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
[51] Sima, C., Chitta, K., Yu, Z., Lan, S., Luo, P., Geiger, A., Li, H., Alvarez, J.M.: Centaur: Robust end-to-end autonomous driving with test-time training. arXiv preprint arXiv:2503.11650 (2025)
[52] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2446–2454 (2020)
[53] Sutton, R.S., Barto, A.G., et al.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
[54] Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. NeurIPS **12** (1999)
[55] Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-end model-free reinforcement learning for urban driving using implicit affordances. In: CVPR. pp. 7153–7162 (2020)
[56] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
[57] Wang, J., Zhang, X., Xing, Z., Gu, S., Guo, X., Hu, Y., Song, Z., Zhang, Q., Long, X., Yin, W.: He-drive: Human-like end-to-end driving with vision language models. arXiv preprint arXiv:2410.05051 (2024)
[58] Wang, Z., Lan, S., Sun, X., Chang, N., Li, Z., Yu, Z., Alvarez, J.M.: Enhancing autonomous driving safety with collision scenario integration. arXiv preprint arXiv:2503.03957 (2025)
[59] Wen, C., Lin, J., Darrell, T., Jayaraman, D., Gao, Y.: Fighting copycat agents in behavioral cloning from observation histories. NeurIPS **33**, 2564–2575 (2020)
[60] Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. NeurIPS **35**, 6119–6132 (2022)

[61] Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., et al.: Pandaset: Advanced sensor suite dataset for autonomous driving. In: ITSC. pp. 3095–3101. IEEE (2021)
[62] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. arXiv preprint arXiv:2401.10891 (2024)
[63] Yang, Z., Jia, X., Li, Q., Yang, X., Yao, M., Yan, J.: Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). arXiv preprint arXiv:2505.16394 (2025)
[64] Yao, W., Li, Z., Lan, S., Wang, Z., Sun, X., Alvarez, J.M., Wu, Z.: Drivesuprim: Towards precise trajectory selection for end-to-end planning. arXiv preprint arXiv:2506.06659 (2025)
[65] Yao, W., Sun, X., Li, Z., Lan, S., Wang, Z., Alvarez, J.M., Wu, Z.: Had: Combining hierarchical diffusion with metric-decoupled rl for end-to-end driving. arXiv preprint arXiv:2604.03581 (2026)
[66] Zhang, Z., Liniger, A., Dai, D., Yu, F., Van Gool, L.: End-to-end urban driving by imitating a reinforcement learning coach. In: ICCV. pp. 15222–15232 (2021)
[67] Zhou, H., Lin, L., Wang, J., Lu, Y., Bai, D., Liu, B., Wang, Y., Geiger, A., Liao, Y.: Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. arXiv preprint arXiv:2412.01718 (2024)
[68] Zhou, Z., Cai, T., Zhao, Seth Z.and Zhang, Y., Huang, Z., Zhou, B., Ma, J.: Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. NeurIPS (2025)
[69] Zou, J., Chen, S., Liao, B., Zheng, Z., Song, Y., Zhang, L., Zhang, Q., Liu, W., Wang, X.: Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. arXiv preprint arXiv:2512.07745 (2025)