Raw2Drive: Reinforcement Learning with Aligned World Models for End-to-End Autonomous Driving (in CARLA v2)

Zhenjie Yang[1,3], **Xiaosong Jia**[2†], **Qifeng Li**[1,3], **Xue Yang**[1], **Maoqing Yao**[4], **Junchi Yan**[1,3†]

1. Sch. of CS, Sch. of AIS, Sch. of AI, Shanghai Jiao Tong University
2. Institute of Trustworthy Embodied AI, Fudan University
3. Asynscale AI 4. AgiBot

Abstract

Reinforcement Learning (RL) can mitigate the causal confusion and distribution shift inherent in imitation learning (IL). However, applying RL to end-to-end autonomous driving (E2E-AD) remains an open problem for its training difficulty, and IL is still the mainstream paradigm in both academia and industry. Recently Model-based Reinforcement Learning (MBRL) have demonstrated promising results in neural planning; however, these methods typically require privileged information as input rather than raw sensor data. We fill this gap by designing ***Raw2Drive***, a dual-stream MBRL approach. Initially, we efficiently train an auxiliary privileged world model paired with a neural planner that uses privileged information as input. Subsequently, we introduce a raw sensor world model trained via our proposed ***Guidance Mechanism***, which ensures consistency between the raw sensor world model and the privileged world model during rollouts. Finally, the raw sensor world model combines the prior knowledge embedded in the heads of the privileged world model to effectively guide the training of the raw sensor policy. Raw2Drive is so far the only RL based end-to-end method on CARLA Leaderboard 2.0, and Bench2Drive and it achieves state-of-the-art performance.

1 Introduction

Beyond modular systems, end-to-end autonomous driving models [5, 6, 7, 8] are emerging, where a unified model directly uses raw sensor inputs for planning. As shown in Figure 1, most of these models [9, 10, 11, 12] are based on imitation learning (IL), which trains models to mimic expert demonstrations. However, **IL faces fundamental limitations such as poor generalization to unseen situations [2, 13, 14, 15] and causal confusion [1, 16, 17]**, which occur when the model incorrectly associates actions with the wrong causes. These issues become problematic in complex and dynamic driving scenarios where decision-making must account for delicate interactions with environment.

Reinforcement learning (RL) [18, 19, 20] offers a promising alternative by optimizing driving policies through reward-driven interactions with the dynamic environment [21, 22, 23], beyond the expert demonstration in imitation learning. Recently, RL has been shown to achieve higher performance bounds than IL in various domains, such as AlphaZero [24, 25], and OpenAI-O3/DeepSeek-R1 [26]. These successes highlight **RL's ability to adapt policies for complex decision-making tasks.**

As shown in Table 1, compared to the extensive research in IL for AD, either through modular systems [27, 28] or recent end-to-end approaches [9], RL for AD has been comparatively less

[†] Correspondence Author. This work was in part supported by by NSFC (62222607,62506229) and Natural Science Foundation of Shanghai (25ZR1402268).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

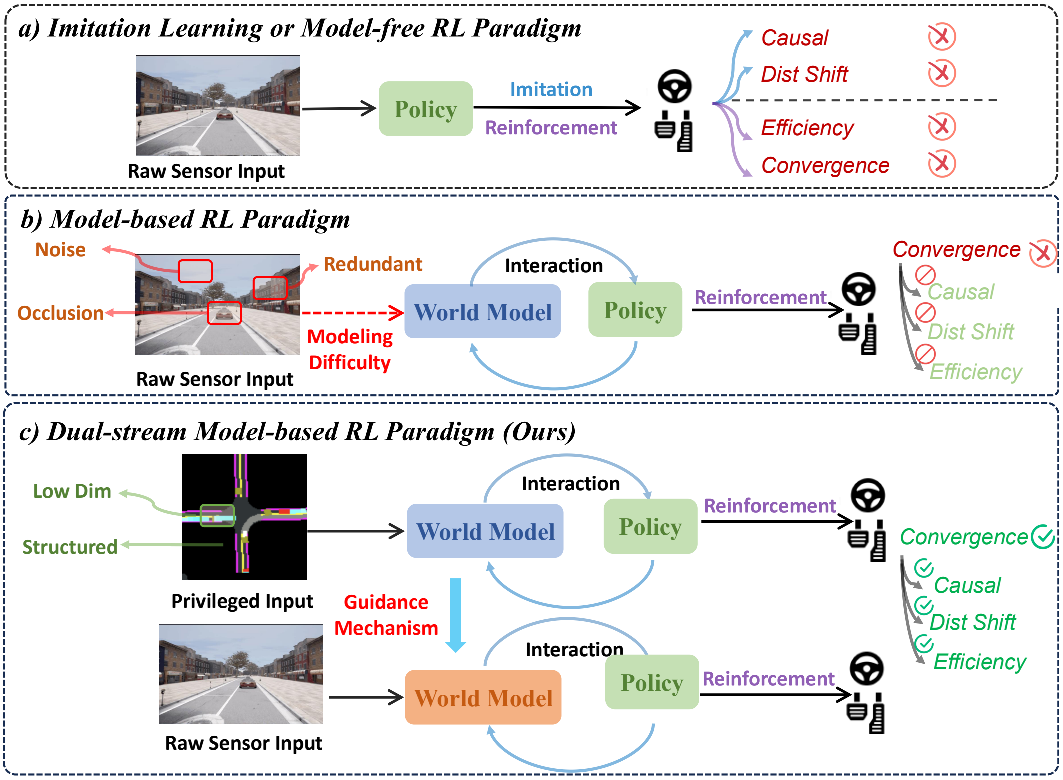

Figure 1: **Comparison of different training paradigms in end-to-end autonomous driving.** **(a) Imitation Learning** suffers from causal confusion [1] and distribution shift [2]. **Model-free Reinforcement Learning** [3] faces efficiency problem and fails to converge. **(b) Model-based Reinforcement Learning**: There are no reported such works for raw sensor input E2E-AD as the raw data can be noisy and redundant, and Think2Drive [4] assumes the privileged ground truth data is given, which cannot be directly applied in real-world AD. **(c)** In ***Raw2Drive***, we propose the first feasible model-based reinforcement learning paradigm for end-to-end autonomous driving. By leveraging low-dimensional, structured privileged input, our approach guides the learning of a world model from raw sensor data, effectively addressing the issues outlined in (a) and (b).

explored. Pioneering work MaRLn [3] faced significant efficiency and convergence issues, requiring approximately 50M steps (57 days of training) with performance lagging far behind contemporary IL based methods [29, 30]. As of today, IL-based methods achieve saturated performance on CARLA Leaderboard 1.0 [31] (abbreviated as CARLA v1), while **none of them can achieve satisfying scores on the more challenging CARLA Leaderboard 2.0** [32] (abbreviated as CARLA v2).

This situation recently changed with the seminal work Think2Drive [4], which introduced a model-based RL (MBRL) approach that successfully solved CARLA v2, with the help of the world model [33]. However, Think2Drive [4] relies on privileged information (ground-truth states of environments) and **there is still no success reported in the field to apply MBRL with raw sensor inputs**. The main challenge lies in the contrast between privileged information, which is compact and facilitates efficient training, and raw sensor data, which is high-dimensional, redundant, and noisy, making the training of world model significantly more difficult. As shown in Table 1, there are no RL based end-to-end methods in CARLA v2, which is the very setting that this paper seeks to challenge. Due to limited space, we discuss more details of **Related Works** in Appendix A.

In this work, we propose ***Raw2Drive***, a dual-stream MBRL method, achieves state-of-the-art performance on CARLA v2 and surpasses IL based methods by a large margin. The training is divided into two stages. In the first stage, we leverage the privileged information to train a privileged world model and a paired neural planner. Then, we further jointly train a raw sensor world model and an end-to-end planner whose input is directly the raw video. In the training of the raw sensor world model, instead

Table 1: **Comparison of settings of mainstream algorithms**. WM means whether to use the world model. SSL means self-supervised learning. CoRL2017 [34] and CARLA v1 [31] include 4 and 10 standard cases, respectively. CARLA v2 [32, 35], on the other hand, introduces 39 additional real-world corner cases, which are significantly more difficult. ROM03 is based on CARLA v1 including 4 very basic scenes.

Method	Venue	Input	Scheme	E2E	w/ WM	Benchmark	Corner Case
Chen [36]	ITSC 2019	Privileged	RL	No	No	Roundabout	No
MaRLn [3]	CVPR 2020	Raw	RL	Yes	No	CoRL 2017	No
Roach [37]	ICCV 2021	Privileged	RL	No	No	Carla v1	No
UniAD [9]	CVPR 2023	Raw	IL	Yes	No	Carla v1	No
Think2Drive [4]	ECCV 2024	Privileged	RL	No	Yes	Carla v2	Yes
LAW [38]	ICLR 2025	Raw	IL+SSL	Yes	Yes	Carla v1	No
AdaWM [39]	ICLR 2025	Privileged	RL	No	Yes	ROM03	No
DriveTrans [40]	ICLR 2025	Raw	IL	Yes	No	Carla v2	Yes
Raw2Drive(Ours)	NeurIPS 2025	Raw	RL	Yes	Yes	Carla v2	Yes

of reconstructing computationally expensive multi-view videos, it is guided by the alignment with the frame-wise feature of the trained privileged world model, facilitating an otherwise extremely difficult training process. We further propose the ***Guidance Mechanism*** to enforce alignment, ensuring consistency in future state predictions during rollouts across both world models. Additionally, it leverages the prior knowledge embedded in the heads of the privileged world model to effectively guide the training of the raw sensor policy. **The contributions are as follows:**

- To our best knowledge, Raw2Drive is the first MBRL framework for E2E-AD, i.e. from raw image input to planning, beyond existing IL [9, 6] or privileged input based RL approaches [37, 4].
- Raw2Drive achieves state-of-the-art performance on the challenging CARLA v2 and Bench2Drive and surpasses IL methods by a large margin, validating the power of RL.
- We only use 64 H800 GPU days in total to deliver our final planner, and the cost can be further saved to 40 GPU days when Think2Drive is reused which dismisses our phase I training. In comparison, IL-based UniAD costs about 30 GPU days yet it only solves even 3∼4 corner cases in CARLA v2. We believe it is significantly less than the imitation learning approaches used in industry and the hope is our work could provide an orthogonal reference for industry applications.

2 Problem Formulation & Related Works

In Figure 2, we give a problem formulation where there are two streams: (I) privileged observations o_t with ground-truth bounding boxes, and HD-Map. (II) raw sensor observations $\hat{o}_t$ with onboard sensors such as cameras, LiDAR, and IMU. Note that **privileged observations are only accessible during training** to facilitate learning. During inference, only raw sensor observations are allowed.

3 Method

Instead of directly adopting a classic MBRL structure like Dreamer V3 [41], ***Raw2Drive*** is a dual-stream MBRL framework, as shown in Figure 2. It consists of four key components: two world models and two corresponding policy models. Detailed symbol definitions are in Table 2.

The rationale behind the two stream design is that: **raw sensor data (e.g., high resolution multi-view videos) are high-dimensional and complex, presenting significant challenges to directly train a world model with low error** [42, 43]. Thus, rather than building the raw sensor world model from scratch, we first build the privileged stream as an auxiliary since the learning of the world model under structured and low-dimensional conditions is much easier and there are already some successes [4]. The raw sensor stream's learning process is guided by the trained privileged stream, which eases the task of extracting the decision-making related information from raw sensor data. The ***Guidance Mechanism*** is carefully designed to avoid cumulative error and train-val gaps.

3.1 Privileged Stream

Privileged Input: As shown in Figure 2 (a), similar to Roach [37] and Think2Drive [4], we utilize time-sequenced BEV semantic masks o_t as input. Since it is commonly used in existing works, we leave details of the privileged input in Appendix C.1.

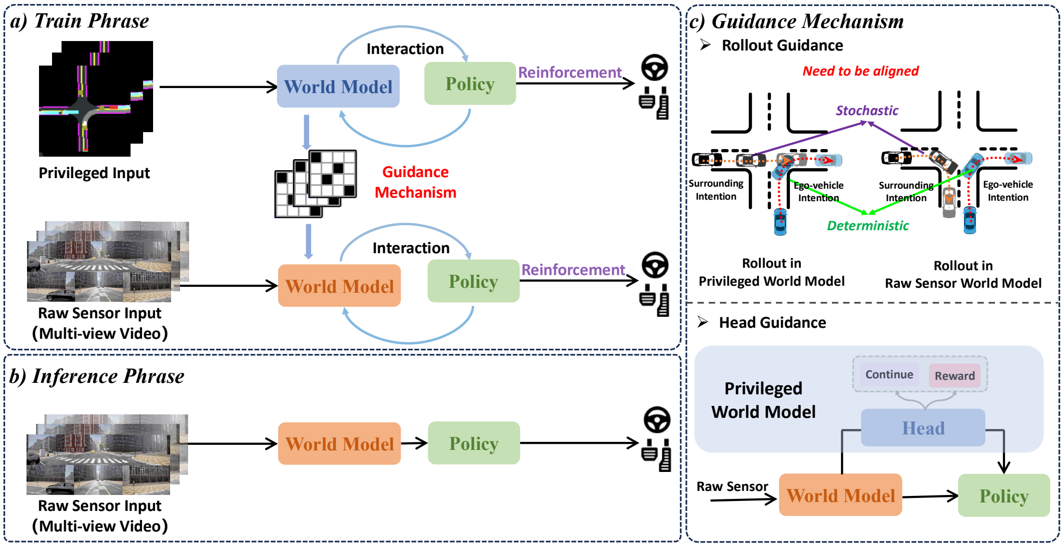

Figure 2: **The Overall Pipeline of Raw2Drive.** (a) During training, we use privileged input to train the privileged world model and paired policy. Then, the privileged world model is used to guide the training of the raw sensor stream. (b) During inference, only raw sensor inputs are available, which aligns with real-world autonomous driving. (c) The guidance mechanism consists of two parts: (I) Rollout Guidance to ensure future modeling consistency; (II) Head Guidance to ensure the supervision for raw sensor policy is accurate and stable.

Table 2: **Notations of Dual Stream World Models in *Raw2Drive*.** WM means the privileged world model and $\hat{\text{WM}}$ means the raw sensor world model. t denotes the time-step. Both world models consist of an Encoder, RSSM, and Heads. The privileged world model is similar to DreamerV3 [41] while raw sensor world model has a tailored encoder $\hat{Enc}$ and only has a tailored decoder head $\hat{Dec}$. ① and ② represent the operation during training and inference respectively.

Dual Stream				Privileged WM	Raw Sensor $\hat{\text{WM}}$
Observation				o_t	$\hat{o}_t$
Encoder State				$e_t = \text{Enc}(o_t)$	$\hat{e}_t = \hat{\text{Enc}}(\hat{o}_t)$
World Model	RSSM	Deterministic State		$h_t = f_\theta(h_{t-1}, a_{t-1}, s_{t-1})$	$\hat{h}_t = \hat{f}_\theta(\hat{h}_{t-1}, a_{t-1}, s_{t-1})$ ① $\hat{h}_t = \hat{f}_\theta(\hat{h}_{t-1}, \hat{a}_{t-1}, \hat{s}_{t-1})$ ②
		Stochastic State	Train	$s_t = q_\theta(h_t, e_t)$	$\hat{s}_t = \hat{q}_\theta(\hat{h}_t, \hat{e}_t)$
			Infer	$s_t = p_\theta(h_t)$	$\hat{s}_t = \hat{p}_\theta(\hat{h}_t)$
	Heads	Reward		$r_t = \text{Reward}(h_t, s_t)$	/
		Decoder		$d_t = \text{Decoder}(h_t, s_t)$	$\hat{d}_t = \hat{\text{Decoder}}(\hat{h}_t, \hat{s}_t)$
		Continue		$c_t = \text{Continue}(h_t, s_t)$	/

Privileged World Model: As shown in the upper part of Figure 3, the same as Dreamer V3 [41], the privileged world model WM is composed of the Encoder, Recurrent State-Space Model (RSSM) [44] and three heads, defined in the left part of Table 2. These components are used (I) for rollout to train the privileged policy by RL; (II) to guide the training of the end-to-end raw sensor stream.

Privileged Policy: As shown in the upper part of Figure 3, the privileged policy is composed of actor-critic networks and is trained through rollouts, same as in Dreamer V3 [33].

3.2 Raw Sensor Stream

Raw Sensor Input: As in Figure 2 (a), raw sensor input $\hat{o}_t$ in this work is multi-view images and IMU. We adopt BEVFormer [45] as the encoder $\hat{\text{Enc}}(\hat{o}_t)$ of the raw sensor stream, which outputs grid-shaped BEV features for the ease of receiving guidance from privileged streams. The details of the inputs are in Appendix C.1.

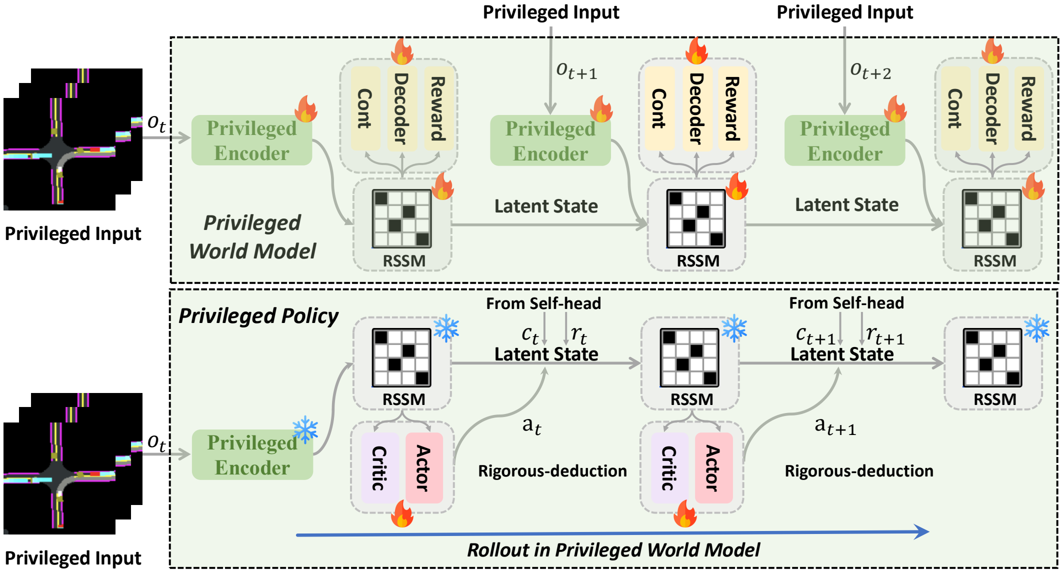

Figure 3: **Training of Privileged World Model and Policy**. The privileged world model WM is trained with time-sequenced BEV semantic masks as inputs. The privileged policy π is trained through rollouts in the privileged world model. The reward r_t and continuous flag c_t are generated by the heads of the privileged world model with the privileged input o_t.

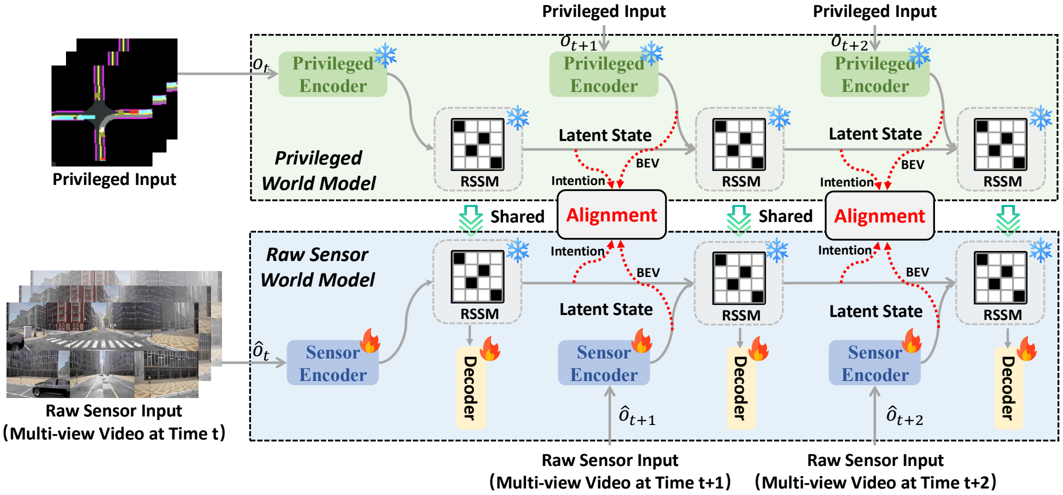

Figure 4: **Training of Raw Sensor World Model**. During training, the spatial-temporal feature of the privileged world model serves as supervision instead of reconstructing multi-view video so that the learning could focus on decision related information. RSSM parameters are initialized from the privileged world model.

Raw Sensor World Model: As shown in Figure 4, the raw sensor world model has a similar architecture as the privileged world model except for the encoder $\hat{Enc}$ and heads. The different encoders are used to process different inputs. As for the heads, **we only use the decoder head which provides supervision signals based on BEV semantic masks** instead of directly reconstructing multi-view videos. Additionally, we find that **learning rewards or continuous flag (both only one scalar) could be harmful for the raw sensor stream**. Specifically, we observe that two adjacent similar frames could have very different reward and continuous labels, which is confusing and thus hampers convergence. We analyze these effects in Section 4.4.

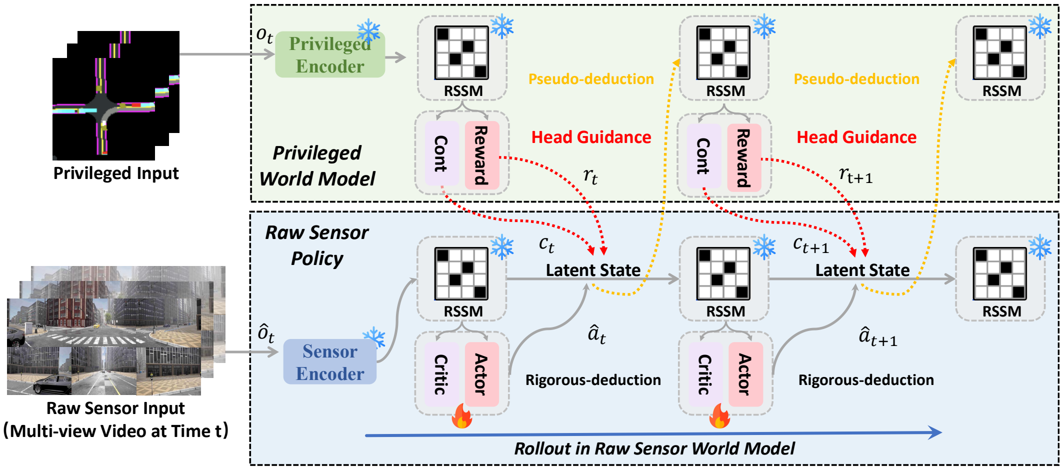

Figure 5: **Training of Raw Sensor Policy.** The raw sensor policy is trained through RL within the dual-stream world model. The raw model operates under strict deduction, while the reward r_t and continuation flag c_t are derived from the privileged model via pseudo-deduction.

Raw Sensor Policy: As in Figure 5, the raw sensor policy is trained by RL with the dual-stream world model. During rollouts, the guidance mechanism detailed in Section 3.3 ensures consistency between the two world models in future predictions. We adopt the heads in the privileged world model to obtain the reward and continuous flag to provide more accurate and stable supervisor signals.

3.3 Guidance Mechanism

In the previous section, we introduced Raw2Drive's dual-stream MBRL framework. In this section, we give details of the guidance mechanism about (I) how the privileged stream guide the learning process of both the raw sensor world and policy; (II) how to alleviate cumulative errors and train-val gaps of the dual-stream paradigm.

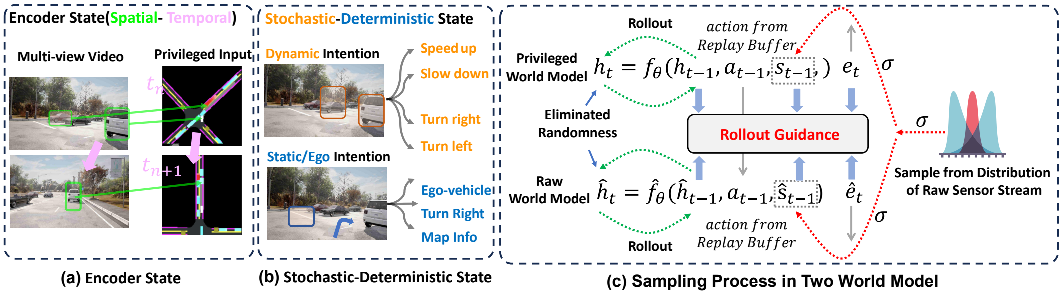
Encoder State(Spatial- Temporal)
Multi-view Video
Privileged Input
t_n
t_{n+1}
(a) Encoder State
Stochastic-Deterministic State
Dynamic Intention
Speed up
Slow down
Turn right
Turn left
Static/Ego Intention
Ego-vehicle
Turn Right
Map Info
(b) Stochastic-Deterministic State
Rollout
action from Replay Buffer
Privileged World Model
$h_t = f_\theta(h_{t-1}, a_{t-1}, s_{t-1},)$
e_t
σ
Eliminated Randomness
Rollout Guidance
σ
Raw World Model
$\hat{h}_t = \hat{f}_\theta(\hat{h}_{t-1}, a_{t-1}, \hat{s}_{t-1})$
$\hat{e}_t$
σ
Sample from Distribution of Raw Sensor Stream
action from Replay Buffer
Rollout
(c) Sampling Process in Two World Model

Figure 6: **State Variables Aligned and Sampling Process in the Rollout Guidance**. (a) The encoder state is aligned temporally and spatially. (b) The deterministic state and stochastic state is aligned to maintain dynamic and static intention consistency. (c) **Eliminating Cumulative Errors Caused By Sampling**. During the rollout process, when deducting next states, we only sample once for the stochastic state - from the distribution of the raw sensor stream. The sampled state is fed into both streams to eliminate the randomness and thus the alignment is more stable.

Rollout Guidance. In MBRL framework, the rollout of world model is crucial for effective policy learning. Thus, under the proposed dual-stream framework, the first type of guidance is Rollout Guidance, which adopts the trained privileged world model provides supervisions during the entire rollout process of the raw sensor world model. Specifically, as shown in Figure 6, there are three components in the rollout process: (1) Encoded State $\hat{e}$, (2) Deterministic State $\hat{h}$, (3) Stochastic State $\hat{s}$. At each timestep t, both stream encode the observation (either privileged or raw sensor) into Encoded State. However, due to the high dimensionality and inherent redundancy of multi-view video data, its encoded state $\hat{e}$ of the raw sensor stream may exhibit instability and errors. To address this, we introduce a loss function to align it with the privileged encoded state e, ensuring spatial consistency at each timestep, called **Spatial-Temporal Alignment Loss**.

Meanwhile, as illustrated in the upper part of Figure 2 (c), and Figure 6 (b), during the rollout process, the deterministic and stochastic states serve distinct modeling purposes. The deterministic state primarily predicts the ego vehicle's state, while the stochastic state focuses on anticipating the behaviors of other traffic participants, such as sudden braking or deceleration. To ensure the consistency of states in the rollout of the two world models, additional supervisory signals are introduced to ensure alignment with the privileged world model. Specifically, the deterministic state employs L2 loss to maintain prediction consistency, whereas the stochastic state is sampled from an independent one-hot distribution. KL divergence [46] is then used to constrain the distributions of the two information streams, ensuring they remain as similar as possible. Furthermore, alignment is enforced across all timesteps, called **Abstract-State Alignment Loss**. The overall loss function is:

$$\mathcal{L}_{\text{Rollout}} = \beta_e \sum_t \sum_{i=0}^{\text{grid num}} \text{MSE}(e_t^i, \hat{e}_t^i) + \sum_t (\beta_s \text{KL}(s_t, \hat{s}_t) + \beta_h \text{MSE}(h_t, \hat{h}_t)) \qquad (1)$$

where grid num means the number of grids under BEV-view and β_e, β_h, β_s are the loss weights of encoder state, deterministic state, and stochastic state. Notably, in the standard RSSM [41], the deduction of stochastic states employs sampling during rollout. **This sampling process can be detrimental for the training of raw sensor world model, as difference caused by randomness during sampling between the two models could accumulate over time**. As a result, in the later timesteps, using states from the privileged world model as supervision signals is confusing for the raw sensor world model to align with. To this end, as shown in Figure 6, we only conduct sampling from the distribution of raw sensor stream and directly feed the sampled variable into the privileged world model to deduct its next state. In this way, the cumulative errors caused by randomness is eliminated and thus is beneficial for the trianing of raw sensor model.

Head Guidance. During the training of the raw sensor world model, as mentioned in Sec 3.2, only the decoder head is trained. We omit the reward head and continuous head learning because directly training the two heads with raw sensor inputs has a convergence issue. Adjacent frames in the video are highly similar while value of reward and continuous flag could fluctuate abruptly, as in Figure 9. As a result, it is difficult for the network to learn stable patterns.

For MBRL, reward and continuous flag play a crucial role in guiding raw policy training. Thus, **we use the accurate reward and continuous flag from the privileged world model during the training of raw sensor policy.** As shown in Figure 2 (c) Lower Part and Figure 5 , at each timestep t, the raw sensor world model executes the action a_t obtained from the raw policy $\hat{\pi}$, transitioning the system to next latent state. At the same time, the privileged world model conducts the same action to rollout. Since ***Rollout Guidance*** imposes the consistency between the two world models, the reward r_t and the continuation flag c_t from the privileged world model could be directly used. Finally, the resulting sequence serves as the training data for optimizing the raw sensor policy. Notably, we also adopt the technique to eliminate randomness mentioned in Section 4.4 so that the reward and continuous flag is accurate. Furthermore, we also use the trained privileged policy to collect to replay buffer and to distill action distributions to raw sensor policy. We summarize our training pipeline in Appendix B.

4 Experiments

4.1 Datasets and Benchmark

We employ the CARLA simulator [34] (version 0.9.15.1) for closed-loop driving performance evaluation. Note that during evaluation, the model only has access to raw sensor observations and is prohibited from utilizing privileged observations. Experimental details are in Appendix C.

Leaderboard 2.0 [34]: It includes two long routes, *validation* and *devtest*. Each of which comprises several routes with lengths of 7-10 kilometers and containing a series of complex corner cases. *As driving inherently follows a Markov decision process [47], evaluating performance over long routes is unnecessary. Moreover, the penalty mechanism employed in scoring [4] fails to accurately reflect the true evaluation capability of the model.*

Table 3: **Performance on Carla Official Town13 Validation and Devtest Benchmark**. *denotes expert feature distillation. As discussed in carla-garage [48] and Section 4.3, long routes evaluation in Leaderboard 2.0 can't reflect the actual driving performance [35].

Method	**Venue**	**Scheme**	**Modality**	**Closed-loop Metric**					
				DS ↑		RC (%) ↑		IS ↑	
				Validation	Devtest	Validation	Devtest	Validation	Devtest
AD-MLP [49]	Arxiv 2023	IL	State	0.00	0.00	0.00	0.00	0.00	0.00
UniAD-Base [9]	CVPR 2023	IL	Image	0.15	0.00	0.51	0.07	0.23	0.04
VAD [10]	ICCV 2023	IL	Image	0.17	0.00	0.49	0.06	0.31	0.04
DriveTrans [40]	ICLR 2025	IL	Image	0.85	0.68	1.42	2.13	0.33	0.35
TCP-traj* [14]	NeurIPS 2022	IL	Image	0.31	0.02	0.89	0.11	0.24	0.05
ThinkTwice* [16]	CVPR 2023	IL	Image	0.50	0.64	1.23	1.78	0.35	**0.43**
DriveAdapter* [17]	ICCV 2023	IL	Image	0.92	0.87	1.52	2.43	0.42	0.37
Raw2Drive (Ours)	NeurIPS 2025	RL	Image	**4.12**	**3.56**	**9.32**	**6.04**	**0.43**	0.42
Think2Drive [4]	ECCV 2024	RL Expert	Privileged*	43.8	56.8	78.2	98.6	0.73	0.92

Table 4: **Performance on Bench2Drive Multi-Ability Benchmark**. * denotes expert feature distillation. IL represents imitation learning. RL represents reinforcement learning. RL expert Think2Drive [4] uses privileged information for training.

Method	**Venue**	**Scheme**	**Modality**	**Ability** (%) ↑					
				Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	**Mean**
TCP-traj* [14]	NeurIPS 2022	IL	Image	8.89	24.29	51.67	40.00	46.28	34.22
AD-MLP [49]	Arxiv 2023	IL	State	0.00	0.00	0.00	0.00	4.35	0.87
UniAD-Base [9]	CVPR 2023	IL	Image	14.10	17.78	21.67	10.00	14.21	15.55
ThinkTwice* [16]	CVPR 2023	IL	Image	27.38	18.42	35.82	**50.00**	54.23	37.17
VAD [10]	ICCV 2023	IL	Image	8.11	24.44	18.64	20.00	19.15	18.07
DriveAdapter* [17]	ICCV 2023	IL	Image	28.82	26.38	48.76	**50.00**	56.43	42.08
DriveTrans [40]	ICLR 2025	IL	Image	17.57	35.00	48.36	40.00	52.10	38.60
Raw2Drive (Ours)	NeurIPS 2025	RL	Image	**43.35**	**51.11**	**60.00**	**50.00**	**62.26**	**53.34**
Think2Drive [4]	ECCV 2024	RL Expert	Privileged*	81.27	83.92	90.24	90.00	87.67	86.26

Table 5: **Results on Bench2Drive Closed-loop Benchmark.** *denotes expert feature distillation. * denotes expert feature distillation. IL represents imitation learning. RL represents reinforcement learning. RL expert Think2Drive [4] uses privileged information for training.

Method	Venue	Scheme	Modality	**Closed-loop Metric**			
				DS ↑	SR(%) ↑	Efficiency ↑	Comfort ↑
TCP-traj* [14]	NeurIPS 2022	IL	Image	59.90	30.00	76.54	18.08
AD-MLP [49]	Arxiv 2023	IL	State	18.05	0.00	48.45	22.63
UniAD-Base [9]	CVPR 2023	IL	Image	45.81	16.36	129.21	43.58
VAD [10]	ICCV 2023	IL	Image	42.35	15.00	157.94	**46.01**
ThinkTwice* [16]	CVPR 2023	IL	Image	62.44	31.23	69.33	16.22
DriveAdapter* [17]	ICCV 2023	IL	Image	64.22	33.08	70.22	16.01
GenAD [50]	ECCV 2024	IL	Image	44.81	15.90	-	-
DriveTrans [40]	ICLR 2025	IL	Image	63.46	35.01	100.64	20.78
MomAD [51]	CVPR 2025	IL	Image	44.54	16.71	170.21	48.63
Raw2Drive (Ours)	NeurIPS 2025	RL	Image	**71.36**	**50.24**	**214.17**	22.42
Think2Drive [4]	ECCV 2024	RL Expert	Privileged	91.85	85.41	269.14	25.97

Bench2Drive [35]: A more comprehensive and fair benchmark, it includes 220 short routes with one challenging corner case per route for analysis of different driving abilities. Following Bench2Drive, we use 1,000 routes under diverse weather conditions for RL training.

4.2 Metric

We employ the official metrics of CARLA [34] for evaluation. **Infraction Score** (IS) measures the number of infractions made along the route, accounting for interactions with pedestrians, vehicles, road layouts, red lights, etc. **Route Completion** (RC) indicates the percentage of the route completed by the autonomous agent. **Driving Score (DS)**, is calculated as the product of Route Completion and Infraction Score. In Bench2Drive [35], additional metric **Success Rate** is used, which measures the proportion of successfully completed routes within the allotted time and without traffic violations.

4.3 Comparison with State-of-the-Art Works

As in Table 3, similar to the conclusions of Bench2Drive, the long-road evaluation in CARLA 2.0 fails to accurately reflect real driving performance due to its cumulative penalty scoring mechanism. In its devtest/validation routes, the first scenario is ***ParkingExit***, where traditional imitation learning methods struggle to solve the task. This is primarily because these models typically rely on L2/L1 loss for training while ***ParkingExit*** requires large steering angles, which are challenging to learn effectively due to the inherent imbalance in the data distribution. Additionally, when a collision causes a blockage, the evaluation stops immediately, preventing the continuation of subsequent scenarios. This does not mean that the performance is poor in other scenarios. Therefore, we mainly focus on the short-route closed-loop evaluation results of Bench2Drive. As shown in Table 4, our method achieves SOTA performance in multi-ability benchmark. In Bench2Drive's closed-loop benchmark, as shown in Table 5, Raw2Drive achieves SOTA performance in raw sensor end-to-end methods.

4.4 Ablation Study

Following Bench2Drive [35, 52] and DriveTransformer [40], the tiny validation set ***Dev10*** comprises 10 carefully selected clips from the official 220 routes. These clips are chosen to be both challenging and representative, with low variance. *To avoid overfitting Bench2Drive-220, we use Dev10 for ablations, repeating each experiment 3 times and reporting the average.*

Choice of Heads in Raw Sensor World Model. Due to the high-dimensional redundancy of image information and the fact that both the reward and continue flags are represented as scalars, the network struggles to achieve stable convergence during training. To further investigate this, we conduct an ablation study on the raw sensor world model heads. As shown in Table 6, the decoder head provides effective supervision, significantly enhancing the learning of the raw sensor world model. In contrast, the reward and continuation heads introduce ambiguity that hinders convergence, leading to suboptimal world model training and degraded policy performance. We provide a detailed visualization in the Appendix H.2.

Table 6: **Ablation on the Raw Sensor World Model Heads**. Only the decoder head is used.

Heads			DS↑	SR↑
Decoder	Reward	Continue		
×	×	×	17.4	1.2/10
✓	×	×	**83.5**	**7.5/10**
✓	✓	×	46.6	3.4/10
✓	✓	✓	34.5	2.2/10

Table 7: **Ablation on the Abstract-State Alignment in Rollout Guidance**.

Latent State			DS↑	SR↑
Encoder	Deterministic	Stochastic		
×	×	×	0.0	0.0/10
✓	×	×	36.4	2.4/10
✓	✓	×	38.3	2.8/10
✓	✓	✓	**83.5**	**7.5/10**

Abstract-State Alignment in Rollout Guidance. We conduct an ablation study to examine the role of Abstract-State Alignment in rollout guidance. As shown in Table 7, since the world model is trained during rollouts, any state misalignment can lead to discrepancies between the predictions of the two world models. Our results show that when any component of the rollout guidance is missing, the model can only handle simple tasks, such as moving straight or turning left, while failing to address more complex interactive corner cases. This misalignment significantly hinders the training of the raw sensor-based world model. All three components of rollout guidance components are thus essential for robust learning and performance. Loss comparisons are provided in Appendix F.

Spatial-Temporal Alignment in Rollout Guidance. The results on Spatial-Temporal Alignment (based on Dev10) is shown in Table 8. Spatial Alignment ensures consistency between image representations and BEV representations—removing it is similar to "driving blind." Temporal Alignment maintains the consistency of future predictions across time. Both components are essential and complementary for a stable world model rollout.

Table 8: **Ablation on the Spatial-Temporal Alignment in Rollout Guidance.**

Spatial Alignment	Temporal Alignment	DS↑	SR(%)↑
×	×	0.0	0.0/10
✓	×	13.6	1.2/10
×	✓	9.24	0.8/10
✓	✓	83.5	7.5/10

Head Guidance. For this ablation study, we assume that the raw sensor world model is trained with three heads. As shown in Table 9, both Setting I and Setting II follow this configuration, aligning with the setup in the last row of Table 6 and reflecting the same considerations and conclusions drawn in Ablation 4.4. Our results indicate that head guidance provides some improvement in policy training. However, the additional heads increase training complexity and degrade performance. Thus, Raw2Drive adopts only the decoder head with head guidance.

Table 9: **Ablation on head guidance for raw sensor policy learning.** D/R/C: decoder, reward, continuation; HG: head guidance.

Method	D	R	C	HG	DS↑	SR↑
Raw2Drive (Ours)	✓	×	×	✓	**83.5**	**7.5/10**
Setting I	✓	✓	✓	✓	34.5	2.2/10
Setting II	✓	✓	✓	×	26.4	1.6/10

Shared Parameter. To investigate the impact of parameter sharing, we conduct an ablation study on whether the Recurrent State-Space Model (RSSM) and the decoder head should share parameters. As shown in Table 10, we compare different configurations where these components are either shared or kept separate. Our findings indicate that sharing parameters between the RSSM and the decoder head leads to better performance. This suggests that parameter sharing facilitates more efficient representation learning, improving the consistency and generalization of the world model.

Table 10: **Ablation Study on the Sharing Parameters (RSSM and Decoder Head)**.

Method	DS↑	SR↑
Raw2Drive (Ours)	**83.5**	**7.5/10**
w/o Shared RSSM	53.2	5.4/10
w/o Shared Head	65.6	6.1/10

Table 11: **Ablation Study on Raw Sensor Policy Training.** Compare two strategies: (1) directly using the privileged policy, and (2) fine-tuning the privileged policy on raw sensor inputs.

Method	DS↑	SR↑
Directly Use Privileged Policy	58.4	5.6/10
Fine-tune Privileged Policy (Ours)	**83.5**	**7.5/10**

Policy Finetuning. To evaluate the necessity of the raw sensor world model, we conduct an ablation study by comparing policy fine-tuning with directly using the privileged policy. Specifically, we analyze the performance difference between policies trained with and without fine-tuning under the raw sensor world model. As shown in Table 11, directly applying the pre-trained privileged policy without fine-tuning results in suboptimal performance, as the policy lacks adaptation to the raw sensor world model's learned dynamics. In contrast, fine-tuning the policy within this world model leads to significant improvements, particularly in handling complex interactive tasks. This result underscores the importance of the raw sensor-based world model for effective policy learning and adaptation.

Real-time Inference. We conducted latency analysis for each module. In end-to-end autonomous driving, the perception backbone (e.g., surround-view image encoder) typically dominates the overall latency. Our world model and policy are highly efficient (each under 2ms), while the raw sensor stream is mainly bottlenecked by the vision encoder (e.g., BEVFormer). The results in Appendix G.

5 Conclusion

We propose ***Raw2Drive***, the first end-to-end model-based reinforcement learning method in autonomous driving. Our approach introduces a novel dual-stream architecture and a guidance mechanism to effectively enable MBRL learning. The approach is fulfilled by the careful treatment of the two world models for the raw and privileged information, respectively, and achieves the state-of-the-art performance on recently released benchmarks. We hope this work serves as a stepping stone toward exploring reinforcement learning for end-to-end autonomous driving.

Limitations: In our setting, the privileged input is ground truth bounding boxes and HD-Map. And in real-world autonomous driving, for industry, the ground truth bounding boxes and HD-Map can be from human annotation or advanced perception algorithms. Reinforcement learning in the real world is also a technical issue that would be solved by 3DGS [53, 54, 55] or a diffusion-based simulator [56] in the future. While CARLA remains the only viable closed-loop simulator for RL research at present, our work focuses on policy learning and introduces a dual-stream world model design that is conceptually decoupled from the specific simulator.

Social Impact: Raw2Drive presents an efficient reinforcement learning framework for end-to-end autonomous driving, mitigating key issues of imitation learning such as causal confusion and distribution shift. By learning robust world models from raw sensors, it enhances the safety, reliability, and generalization of autonomous driving systems.

References

[1] Sidney D'Mello, Blair Lehman, Reinhard Pekrun, and Art Graesser. Confusion can be beneficial for learning. *Learning and Instruction*, 29:153–170, 2014.

[2] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 33:2564–2575, 2020.

[3] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020.

[4] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking in latent world model for quasi-realistic autonomous driving (in carla-v2). In *ECCV*, 2024.

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

[6] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2023.

[7] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling, 2023.

[8] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Jia Zeng, Zhiqi Li, Jiazhi Yang, Hanming Deng, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2151–2170, 2023.

[9] 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 *CVPR*, pages 17853–17862, 2023.

[10] 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. *ICCV*, 2023.

[11] Xiaosong Jia, Liting Sun, Hang Zhao, Masayoshi Tomizuka, and Wei Zhan. Multi-agent trajectory prediction by combining egocentric and allocentric views. In *Conference on Robot Learning*, pages 1434–1443. PMLR, 2022.

[12] Xiaosong Jia, Li Chen, Penghao Wu, Jia Zeng, Junchi Yan, Hongyang Li, and Yu Qiao. Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach. In *Conference on Robot Learning*, pages 910–920. PMLR, 2023.

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

[14] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*, 2022.

[15] Xiaosong Jia, Liting Sun, Masayoshi Tomizuka, and Wei Zhan. Ide-net: Interactive driving event and pattern extraction from human data. *IEEE robotics and automation letters*, 6(2):3065–3072, 2021.

[16] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023.

[17] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[19] Yazhe Niu, Yuan Pu, Zhenjie Yang, Xueyan Li, Tong Zhou, Jiyuan Ren, Shuai Hu, Hongsheng Li, and Yu Liu. Lightzero: A unified benchmark for monte carlo tree search in general sequential decision scenarios. *Advances in Neural Information Processing Systems*, 36:37594–37635, 2023.

[20] Yuan Pu, Yazhe Niu, Jiyuan Ren, Zhenjie Yang, Hongsheng Li, and Yu Liu. Unizero: Generalized and efficient planning with scalable latent world models. *arXiv preprint arXiv:2406.10667*, 2024.

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

[22] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[23] Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):15453–15461, Apr. 2025.

[24] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[25] Yazhe Niu, Yuan Pu, Zhenjie Yang, Xueyan Li, Tong Zhou, Jiyuan Ren, Shuai Hu, Hongsheng Li, and Yu Liu. Lightzero: A unified benchmark for monte carlo tree search in general sequential decision scenarios. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[27] Mobileye. Mobileye under the hood. `https://www.mobileye.com/ces-2022/`, 2022.

[28] NVIDIA. NVIDIA DRIVE End-to-End Solutions for Autonomous Vehicles. `https://developer.nvidia.com/drive`, 2022.

[29] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pages 66–75. PMLR, 2020.

[30] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] Get started with Leaderboard 1.0 — leaderboard.carla.org. `https://leaderboard.carla.org/get_started_v1/`, 2022.

[32] Get started with Leaderboard 2.0 — leaderboard.carla.org. `https://leaderboard.carla.org/get_started/`, 2023.

[33] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dreamer: Efficient reinforcement learning through latent imagination. In *International Conference on Machine Learning*, 2020.

[34] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[35] 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 *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[36] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*, pages 2765–2771. IEEE, 2019.

[37] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021.

[38] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhaneddin Yaman, Liu Ren, and Junshan Zhang. Adawm: Adaptive world model based planning for autonomous driving, 2025.

[40] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[41] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[42] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

[43] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. *arXiv preprint arXiv:2401.12592*, 2024.

[44] Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Trimpe Sebastian. Probabilistic recurrent state-space models. In *International conference on machine learning*, pages 1280–1289. PMLR, 2018.

[45] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *ECCV*, 2022.

[46] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[47] Bernhard Jaeger, Daniel Dauner, Jens Beißwenger, Simon Gerstenecker, Kashyap Chitta, and Andreas Geiger. Carl: Learning scalable planning policies with simple rewards. *arXiv preprint arXiv:2504.17838*, 2025.

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

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

[50] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. *arXiv preprint arXiv: 2402.11502*, 2024.

[51] 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. *arXiv preprint arXiv:2503.03125*, 2025.

[52] GitHub - Thinklab-SJTU/Bench2Drive: [NeurIPS 2024 Datasets and Benchmarks Track] Closed-Loop E2E-AD Benchmark Enhanced by World Model RL Expert — github.com. `https://github.com/Thinklab-SJTU/Bench2Drive`. [Accessed 01-03-2025].

[53] Zipei Ma, Junzhe Jiang, Yurui Chen, and Li Zhang. Béziergs: Dynamic urban scene reconstruction with bézier curve gaussian splatting. In *ICCV*, 2025.

[54] Xianliang Huang, Jiajie Gou, Shuhang Chen, Zhizhou Zhong, Jihong Guan, and Shuigeng Zhou. Iddr-ngp: Incorporating detectors for distractors removal with instant neural radiance field. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1343–1351, 2023.

[55] Shuhang Chen, Xianliang Huang, Zhizhou Zhong, Juhong Guan, and Shuigeng Zhou. A focused human body model for accurate anthropometric measurements extraction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22658–22667, 2025.

[56] Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger, Xiangyu Yue, and Li Chen. Resim: Reliable world simulation for autonomous driving. *arXiv preprint arXiv:2506.09981*, 2025.

[57] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

[58] Xiaosong Jia, Shaoshuai Shi, Zijun Chen, Li Jiang, Wenlong Liao, Tao He, and Junchi Yan. Amp: Autoregressive motion prediction revisited with next token prediction for autonomous driving. *arXiv preprint arXiv:2403.13331*, 2024.

[59] Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Zhao Huang, Yipeng Wu, Die Zuo, Jibin Peng, Ziyuan Zhong, Xin Wang, et al. Trajectory-llm: A language-based data generator for trajectory prediction in autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[60] Yutao Zhu, Xiaosong Jia, Xinyu Yang, and Junchi Yan. Flatfusion: Delving into details of sparse transformer-based camera-lidar fusion for autonomous driving. *arXiv preprint arXiv:2408.06832*, 2024.

[61] Han Lu, Xiaosong Jia, Yichen Xie, Wenlong Liao, Xiaokang Yang, and Junchi Yan. Activead: Planning-oriented active learning for end-to-end autonomous driving. *arXiv preprint arXiv:2403.02877*, 2024.

[62] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. Llm4drive: A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

[63] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025.

[64] Xiaosong Jia, Penghao Wu, Li Chen, Yu Liu, Hongyang Li, and Junchi Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):13860–13875, 2023.

[65] Junqi You, Xiaosong Jia, Zhiyuan Zhang, Yutao Zhu, and Junchi Yan. Bench2drive-r: Turning real world data into reactive closed-loop autonomous driving benchmark by generative model. *arXiv preprint arXiv:2412.09647*, 2024.

[66] Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

[67] Fengshuo Bai, Hongming Zhang, Tianyang Tao, Zhiheng Wu, Yanna Wang, and Bo Xu. Picor: Multi-task deep reinforcement learning with policy correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6728–6736, Jun. 2023.

[68] Fengshuo Bai, Rui Zhao, Hongming Zhang, Sijia Cui, Ying Wen, Yaodong Yang, Bo Xu, and Lei Han. Efficient preference-based reinforcement learning via aligned experience estimation. *arXiv preprint arXiv:2405.18688*, 2024.

[69] Fengshuo Bai, Yu Li, Jie Chu, Tawei Chou, Runchuan Zhu, Ying Wen, Yaodong Yang, and Yuanpei Chen. Retrieval dexterity: Efficient object retrieval in clutters with dexterous hand. *arXiv preprint arXiv:2502.18423*, 2025.

[70] Zhaowei Zhang, Fengshuo Bai, Qizhi Chen, Chengdong Ma, Mingzhi Wang, Haoran Sun, Zilong Zheng, and Yaodong Yang. Amulet: Realignment during test time for personalized preference adaptation of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[71] Lu Li, Jiafei Lyu, Guozheng Ma, Zilin Wang, Zhenjie Yang, Xiu Li, and Zhiheng Li. Normalization enhances generalization in visual reinforcement learning. *arXiv preprint arXiv:2306.00656*, 2023.

[72] Ming Zhang, Shenghan Zhang, Zhenjie Yang, Lekai Chen, Jinliang Zheng, Chao Yang, Chuming Li, Hang Zhou, Yazhe Niu, and Yu Liu. Gobigger: A scalable platform for cooperative-competitive multi-agent interactive simulation. In *The Eleventh International Conference on Learning Representations*, 2023.

[73] Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

[74] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pages 726–737. PMLR, 2023.

[75] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023.

A Related Works

A.1 End-to-End Autonomous Driving Based on Imitation Learning

Imitation Learning (IL) [5, 57, 58, 59, 60, 61, 62, 63] is a foundational method for training agents by mimicking expert behavior, especially suitable for end-to-end autonomous driving tasks. IL aims to replicate human driving actions, thus reducing system complexity and simplifying the decision-making process from perception to control.

Some approaches [14, 16, 17, 64] typically involve collecting large amounts of human driving data and using supervised learning to directly map sensor inputs (such as camera, LiDAR data, etc.) to control outputs (such as steering angle, throttle, and brake). The expert driving actions are recorded as state-action pairs, and imitation learning is used to train a model that approximates this behavior. The classic TCP model [14] uses a single camera input in a dual-branch design that integrates trajectory prediction and multi-step control within a single framework. It flexibly combines the outputs of both branches based on prior planning to optimize the final control signal. The latest UniAD [9] approach goes further by using surround-view camera inputs and leveraging the query feature of transformer architectures, integrating detection, tracking, mapping, trajectory prediction, occupancy grid prediction, and planning into a single differentiable end-to-end pipeline. This design sets a new benchmark for integrating tasks in autonomous driving. To accelerate grid-based scene representation in UniAD, VAD [10] proposes a fully vectorized end-to-end framework that uses vectorized representations for agent motion and map elements, significantly improving task efficiency.

Although behavioral cloning methods perform well in simple driving environments, they still cannot successfully complete long-term sequence decisions in complex traffic scenarios [35, 49, 65], which also called corner case. The reason is that its network uses regression loss to generate future trajectories, and scenes with large turns will be averaged by straight road scenes. For example, the recent closed-loop Bench2Drive [35] evaluation showed that supervised learning models are sensitive to shifts in data distribution; they perform well in simple driving interactions but tend to accumulate errors in dynamic traffic settings, causing the model to deviate from the intended driving path.

A.2 End-to-End Autonomous Driving based on Model-free Reinforcement Learning

Model-free reinforcement learning [37, 3, 66, 67, 68, 69, 70, 71, 72, 73] is another key approach for end-to-end autonomous driving, optimizing decision-making through rewards rather than direct imitation. Early RL work, such as MaRLn [3], demonstrated the potential of reinforcement learning in autonomous driving. However, it required extensive pre-training, with around 20M (23 days) of training in a single town or 50M (57 days) in multi-towns to solve only 4 standard cases in CoRL 2017 [34]. The complexity of perception in autonomous driving makes it difficult to optimize perception and control jointly, increasing the challenge of training end-to-end systems with RL. To reduce the impact of perception on decision training, some classic RL methods, such as Roach [37], use 2D Bird's Eye View (BEV) representations as input observations, simplifying the task of learning from raw sensor data and focusing on high-level environmental features. And then use RL methods to collect offline data for training the end-to-end method [14, 74, 75].

A.3 End-to-End Autonomous Driving based on Model-based Reinforcement Learning

Model-free RL suffers challenges in complex driving scenarios or corner cases that require long-term memory. Think2Drive [4] combines Roach's [37] 2D BEV inputs with a world model, significantly improving sampling efficiency and effectively modeling sequential interactions, leading to strong performance in complex traffic scenarios. To our best knowledge, there is no model-based RL end-to-end method in autonomous driving.

B Details of Training Pipeline

As shown in Alogrithm 1,the training pipeline of Raw2Drive consists of two stages. In Stage I, we use privileged observations for MBRL training. The world model and behavior policy are updated alternately.

In Stage II, we train the raw sensor world with the help of the proposed guidance mechanism. Compared to Stage I, the world model is trained with an additional loss derived from the rollout guidance. For behavior policy training, we fine-tune the privileged policy by interacting with the raw sensor world model with head guidance.

Algorithm 1: Training Pipeline of Raw2Drive

```
Privileged Observation $o$, World Model WM, Policy $\pi$
Raw Sensor Observation $\hat{o}$, World Model $\hat{\text{WM}}$, Policy $\hat{\pi}$
Privileged Replay Buffer $B$, Replay Buffer $\hat{B}$ Training iterations $N$
Stage 1: Privileged World Model and Policy Training
for $i = 1$ to $N$ do
    Collect trajectories $(o_t, a_t, r_t, c_t, o_{t+1})$ with current $\pi$ to interact with simulator and store them in
      Buffer $B$;
    Sample a trajectory $(o_{t:T}, a_{t:T}, r_{t:T}, c_{t:T}, o_{t+1:T+1})$ from Privileged Replay Buffer $B$;
    Train Priviledged World Model WM with prediction loss $\mathcal{L}_{pred}$, dynamics loss $\mathcal{L}_{dyn}$, and
      representation loss $\mathcal{L}_{rep}$ with weights $\beta_{pred}$, $\beta_{dyn}$, $\beta_{rep}$;
    $\mathcal{L}_{\text{WM}} = \mathbb{E}\left[ \sum_{t=1}^{T} \left(\beta_{\text{pred}}\mathcal{L}_{\text{pred}} + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}\right) \right]$;
    Sample a trajectory $o_{t:T}$ from Privileged Replay Buffer $B$;
    Rollout in Priviledged World Model to obtain the predicted latent state $(h_{t:T}, s_{t:T})$, as well as the
      predicted reward and continuous flag. These predictions are then combined to construct the trajectory:
      $(o_{t:T}, a_{t:T}, r_{t:T}, c_{t:T}, h_{t+1:T+1}, s_{t+1:T+1})$;
    Train privileged policy $\pi$ based on the actor-critic algorithm with the above trajectory which rollouts in
      Priviledged World Model;
Stage 2: Raw Sensor World Model and Policy Training
for $i = 1$ to $N$ do
    Collect trajectories $(o_t, \hat{o}_t, \hat{a}_t, \hat{r}_t, \hat{c}_t, \hat{o}_{t+1})$ by using raw sensor policy $\hat{\pi}$ to interact with simulator and
      store them in Raw Replay Buffer $\hat{B}$;
    Sample a trajectory $(o_{t:t+T}, \hat{o}_{t:t+T}, \hat{a}_{t:t+T}, \hat{r}_{t:t+T}, \hat{c}_{t:t+T}, \hat{o}_{t+1:t+T+1})$ from Raw Replay Buffer $\hat{B}$;
    Train Raw Sensor World Model $\hat{\text{WM}}$ with additional loss in Rollout Guidance for latent state
      alignment;
    $\mathcal{L}_{\hat{\text{WM}}} = \mathcal{L}_{\text{WM}} + \mathcal{L}_{\text{Rollout}}$;
    Sample a trajectory $\hat{o}_{t:T}$ from Raw Replay Buffer $\hat{B}$;
    Rollout in Raw Sensor World Model to obtain the predicted latent state $(\hat{h}_{t:T}, \hat{s}_{t:T})$. Using Head
      Guidance to obtain the predicted reward and continuous flag. These predictions are then combined to
      construct the trajectory: $(\hat{o}_{t:T}, \hat{a}_{t:T}, \hat{r}_{t:T}, \hat{c}_{t:T}, \hat{h}_{t+1:T+1}, \hat{s}_{t+1:T+1})$;
    Train Raw Sensor Policy $\hat{\pi}$ based on the actor-critic algorithm with the above trajectory which rollouts
      in Raw Sensor World Model;
```

C Details of the Experiment

C.1 Details of the Dual Stream Input

For the privileged inputs, we utilize BEV semantic segmentation masks $\in \{0,1\}^{H \times W \times C}$ as image input and ego vehicle info as vector input. Each channel in the BEV masks represents the presence of a specific type of object. It is generated from the privileged information obtained from the simulator and consists of C masks of size $H \times W$. Note that the $C = 43$ channels of semantic segmentation masks correspond to static objects (e.g. roads, lines, lanes, and the ego vehicle) and dynamic objects (signs, lights, pedestrians, vehicles, and obstacles).

For the raw sensor inputs, we use BEVFormer [45] as the encoder surrounding RGB images to achieve grid-shape BEV features.

C.2 Action Space in Raw2Drive

To simplify the action space, we design a set of 39 discrete actions. The full list of actions is provided in Table 12.

Table 12: **The Discrete Actions.** The continuous action space is decomposed into 39 discrete actions, each for specific values of throttle, steer, brake and reverse. Each action is rational and legitimate.

Throttle	Brake	Steer	Reverse	Throttle	Brake	Steer	Reverse	Throttle	Brake	Steer	Reverse
0	0	1	False	0.3	0	-0.2	False	0	0	0.1	False
0.7	0	-0.5	False	0.3	0	-0.1	False	0	0	0.3	False
0.7	0	-0.3	False	0.3	0	0	False	0	0	0.6	False
0.7	0	-0.2	False	0.3	0	0.1	False	0	0	1.0	False
0.7	0	-0.1	False	0.3	0	0.2	False	0.5	0	-0.5	True
0.7	0	0	False	0.3	0	0.3	False	0.5	0	-0.3	True
0.7	0	0.1	False	0.3	0	0.5	False	0.5	0	-0.2	True
0.7	0	0.2	False	0.3	0	0.7	False	0.5	0	-0.1	True
0.7	0	0.3	False	0	0	-1	False	0.5	0	0	True
0.7	0	0.5	False	0	0	-0.6	False	0.5	0	0.1	True
0.3	0	-0.7	False	0	0	-0.3	False	0.5	0	0.2	True
0.3	0	-0.5	False	0	0	-0.1	False	0.5	0	0.3	True
0.3	0	-0.3	False	1	0	0	False	0.5	0	0.5	True

C.3 Model Configuration

We implement the model using PyTorch. Both world models are trained with a learning rate of 1e-5, weight decay of 0.00, and the AdamW optimizer. The behavior policy is trained with a learning rate of 3e-5, weight decay of 0.00, also using AdamW. The weights β_e, β_h, and β_s are set to 10, 5, and 10, respectively.

C.4 Reward Design in Raw2Drive

We adopt the reward design and reward shaping approach from Think2Drive [4].

D Details of the losses in World Model

Train the World Model W_p using prediction loss $\mathcal{L}_{pred}$, the dynamics loss $\mathcal{L}_{dyn}$, and the representation loss $\mathcal{L}_{rep}$, the loss weights are respectively β_{pred}, β_{dyn}, β_{rep}:

$$\begin{aligned}
\mathcal{L}_{\text{pred}}(\theta) &\doteq -\ln p_\theta(e_t \mid s_t, h_t) - \ln p_\theta(r_t \mid s_t, h_t) \\
&\quad - \ln p_\theta(c_t \mid s_t, h_t) \\
\mathcal{L}_{\text{dyn}}(\theta) &\doteq \max\left(1, \text{KL}\left[\text{sg}\left(q_\theta(s_t \mid h_t, e_t)\right) \parallel p_\theta(s_t \mid h_t)\right]\right) \\
\mathcal{L}_{\text{rep}}(\theta) &\doteq \max\left(1, \text{KL}\left[q_\theta(s_t \mid h_t, e_t) \parallel \text{sg}\left(p_\theta(s_t \mid h_t)\right)\right]\right)
\end{aligned} \tag{2}$$

E Details of the losses in Behavior Policy

The critic network p_ψ estimates the distribution of future returns R_t^λ, while the value function v_t represents the expected value of the return at state s_t, train Critic network in the Privileged Behavior Policy π_p by maximum likelihood loss:

$$\begin{aligned}
\mathcal{L}(\psi) &\doteq -\sum_{t=1}^{T} \ln p_\psi(R_t^\lambda | s_t) \\
R_t^\lambda &\doteq r_t + \gamma c_t \left((1-\lambda) v_t + \lambda R_{t+1}^\lambda\right) \\
R_T^\lambda &\doteq v_T
\end{aligned} \tag{3}$$

Train the actor network through policy optimization, using entropy regularization $\mathcal{H}$, exponential moving average (EMA) for smoothing, entropy regularization coefficient η and the sg operation for gradient stability.

$$\begin{aligned}
\mathcal{L}(\theta) \doteq &-\sum_{t=1}^{T} \text{sg}\left(\frac{(R_t^\lambda - v_\psi(s_t))}{\max(1, S)}\right) \log \pi_\theta(a_t | s_t) \\
&+ \eta \mathcal{H}\left[\pi_\theta(a_t | s_t)\right] \\
S \doteq &\text{EMA}\left(\text{Per}(R_t^\lambda, 95) - \text{Per}(R_t^\lambda, 5), 0.99\right)
\end{aligned}$$

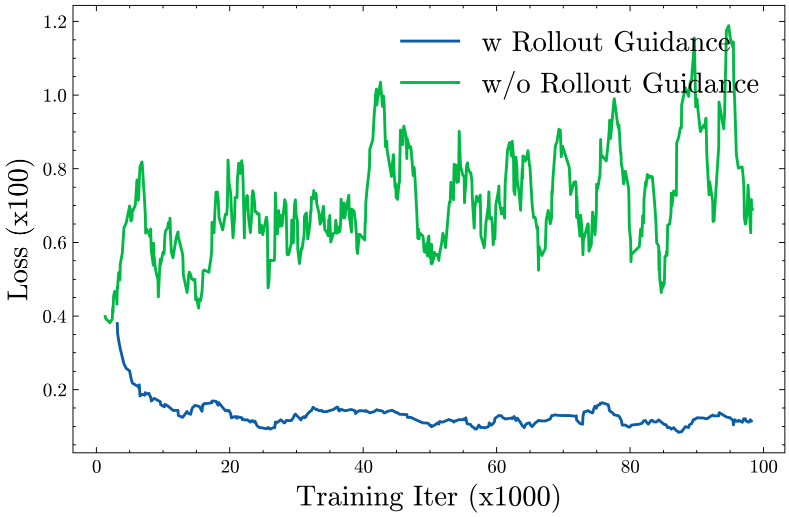

Figure 7: **Comparison of the Raw Sensor World Model Loss w/ and w/o Guidance**.

F Rollout Guidance

As shown in Figure 7, the rollout guidance ensures consistency between the dual-stream world models during rollouts, playing a critical role in training the raw sensor world model; without it, the network struggles to converge.

G Inference Latency

Table 13: **Latency Comparison between Privileged and Raw Sensor Streams.**

Method	Modality	Encoder Latency (ms)	World Model Latency (ms)	Policy Latency (ms)
Privileged Stream	BEV State	2 (5×Conv)	2	2
Raw Sensor Stream	Multi-view images	600 (BEVFormer)	2	2

H Visualization

H.1 Decoder Output by the World Model

We visualize the output of the raw sensor decoder to evaluate its reconstruction quality. In our dual-stream architecture, the raw sensor world model replaces expensive video reconstruction with a more efficient BEV reconstruction. As shown in the Figure 8, under the guidance mechanism, the raw sensor world model successfully learns to reconstruct the BEV representation with reasonable accuracy. However, due to the inherent limitations of the camera sensor, the model struggles to reconstruct occluded regions or areas beyond its direct line of sight.

H.2 Perception Confusion

As shown in Figure 9, adjacent frames exhibit a high degree of similarity. In the upper section, the ego vehicle is in a corner case of ***ParkingExit***. Despite minimal visual differences between adjacent frames, identical actions can yield opposite rewards, causing model confusion. In the lower section, as the ego vehicle nears task completion, the visual change remains subtle. However, the binary nature of the continuation flag (0 or 1) provides limited information, further hindering convergence.

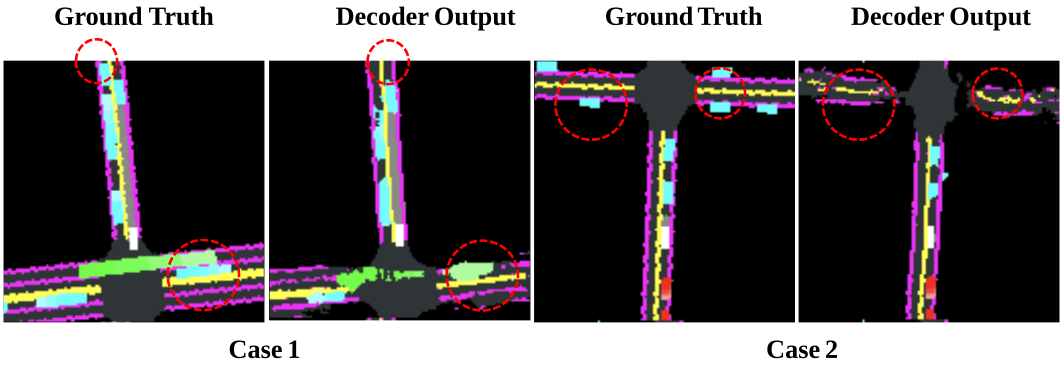

Figure 8: **Ground Truth VS Raw Sensor Decoder Output by the World Model** (The red circle is the blind spot of the camera).

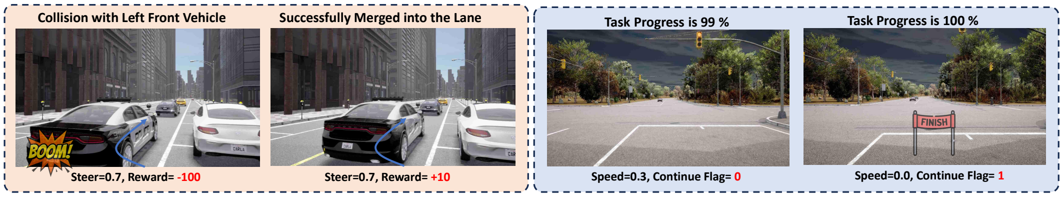
Collision with Left Front Vehicle
Successfully Merged into the Lane
BOOM!
Steer=0.7, Reward= -100
Steer=0.7, Reward= +10
Task Progress is 99 %
Task Progress is 100 %
FINISH
Speed=0.3, Continue Flag= 0
Speed=0.0, Continue Flag= 1

Figure 9: **Perception Confusion in Reward and Continuous Head**.