Trajectory Evaluation via Rollout-Free World Model for End-to-end Autonomous Driving

Junzhe Wang[1*] **Jinqing Zhang**[1,2*] **Qingjie Liu**[1†]
[1]State Key Laboratory of Virtual Reality Technology and Systems,
Beihang University, Beijing, China
[2]Xiaomi EV
qingjie.liu@buaa.edu.cn

Abstract

Although existing candidate trajectory evaluators have substantially improved end-to-end planning accuracy, their underlying mechanism remains suboptimal. Model-based evaluators typically make scoring future-aware by explicitly predicting candidate-conditioned future states over the planning horizon, which we term rollout. However, rollout incurs high inference cost and accumulates prediction errors. We argue that planning does not require reconstructing a full future scene for each candidate, but only future-relevant information for reliable comparison. We therefore propose **RFWorld**, a Rollout-Free World model for end-to-end trajectory evaluation. RFWorld constructs a shared scene memory and shapes it into a future-readable representation via lightweight training-time temporal readout with semantic supervision, keeping training overhead low. At inference time, candidate trajectories directly query the shared memory via sparse spatial readout for scoring, reducing computation by roughly the planning horizon times the number of candidate trajectories, while avoiding recursive rollout errors. Experiments on NAVSIM and Bench2Drive show that RFWorld learns a more planning-aligned world representation and achieves state-of-the-art performance.

1 Introduction

End-to-end autonomous driving [2, 5, 28, 15, 16, 19] has attracted increasing attention by directly mapping sensor observations to future trajectories or low-level control commands, reducing the interface bottlenecks and error accumulation of conventional modular pipelines. Recent planning-oriented methods [3, 31, 26, 37, 44] have further advanced this paradigm from single-trajectory regression to anchor-based, proposal-based, diffusion-based, and flow-based trajectory generation. However, as the action space becomes increasingly multi-modal, a fundamental question remains: given multiple plausible future motions, how should an autonomous driving system evaluate them before execution?

Trajectory evaluation is indispensable for safe and reliable end-to-end planning. Classical methods [8, 32, 6] rely on hand-crafted rules, cost functions, or model predictive objectives. More recent learning-based approaches [25, 24, 29, 38] incorporate rule-based teachers, simulator-derived feedback, or learned score/reward models to distinguish unsafe candidates from human-like but risky ones. Yet many existing evaluators either score trajectories from the current state alone or depend on explicit candidate-conditioned future prediction.

A natural way to make trajectory evaluation future-aware is to introduce world models. Driving world models [1, 13, 34, 35, 9, 49, 48, 4, 42, 47] have been explored for future image generation, occupancy

*Equal contribution.
†Corresponding author.

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

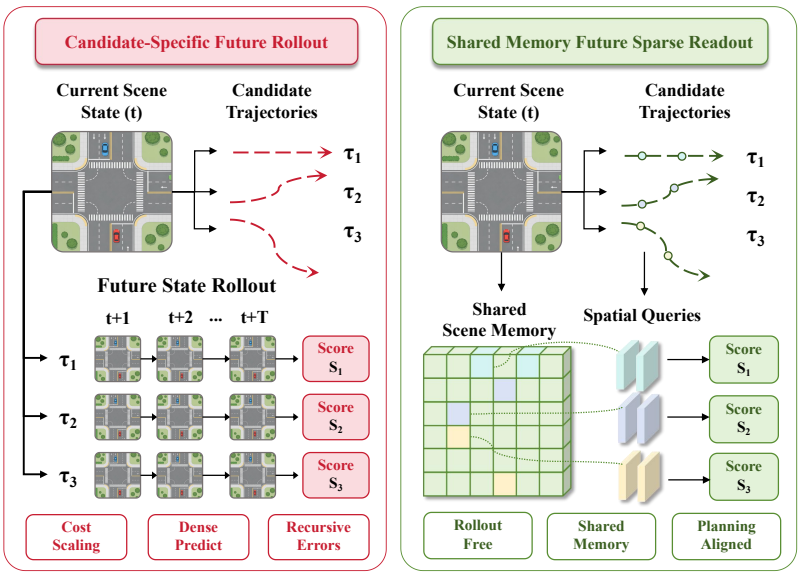


Figure 1: **Two paradigms for future-aware trajectory evaluation.** Conventional methods evaluate each candidate by rolling out its future states separately. In contrast, RFWorld constructs a shared scene memory once and lets all candidate trajectories read from it directly for rollout-free scoring.

forecasting, BEV-state prediction, latent dynamics modeling, and planning-oriented reasoning. In trajectory evaluation, they can imagine the future consequences of each candidate trajectory and estimate its reward or safety score [12, 27, 24]. Existing model-based evaluators typically do so by explicitly predicting candidate-conditioned future states over the planning horizon, which we term *rollout*. However, should future-aware evaluation require rolling out a future scene for every candidate?

We argue that explicit candidate-wise rollout is misaligned with the actual needs of planning. Its inference cost grows with both the planning horizon and the number of candidates, while recursive prediction also makes trajectory scoring vulnerable to accumulated errors. More importantly, planning does not require reconstructing a full future scene for each candidate, but only accessing future-relevant information for reliable comparison.

Motivated by this, we reformulate model-based trajectory evaluation from trajectory-specific future rollout to rollout-free shared-memory readout, as illustrated in Fig. 1. Instead of generating candidate-conditioned future states at inference time, RFWorld constructs a shared scene memory. To make all candidates readable through a common representation, we express their waypoints on the current ego-coordinate BEV plane rather than in independent ego-centric futures, enabling aligned spatial access to a unified memory. During training, temporal readout with semantic supervision shapes this memory into a future-readable representation with predictive temporal structure; during inference, candidate trajectories directly query it through sparse spatial readout for scoring.

This reformulation makes the world model more planning-aligned. Since the shared memory is computed once and reused across candidates, RFWorld avoids candidate-wise dense future generation, substantially reduces inference computation, keeps training overhead low through lightweight temporal shaping, and is less sensitive to compounding rollout errors. In this sense, RFWorld shifts the role of the world model from candidate-conditioned future simulation to constructing a planning-oriented shared memory for trajectory evaluation.

On NAVSIM [7], RFWorld achieves state-of-the-art performance and validates the effectiveness of rollout-free shared-memory trajectory evaluation. We also evaluate RFWorld in closed-loop simulation on the CARLA-based Bench2Drive [18] benchmark.

Our contributions are summarized as follows:

- We identify a key mismatch in future-aware trajectory evaluation: existing model-based evaluators typically rely on explicit candidate-conditioned future rollout, whereas end-to-end planning

only requires future-relevant information to be directly queryable through a planning-sufficient interface.

- We propose **RFWorld**, a Rollout-Free World model for end-to-end trajectory evaluation. RFWorld constructs a shared scene memory, shapes it through lightweight training-time temporal readout with semantic supervision, and performs inference-time trajectory scoring via sparse spatial readout, thereby substantially reducing inference cost, keeping training overhead low, and mitigating accumulated rollout errors.
- We conduct extensive experiments on the NAVSIM benchmark and show that RFWorld achieves state-of-the-art performance, demonstrating the effectiveness of rollout-free shared-memory world modeling for end-to-end autonomous driving.

2 Related Work

2.1 End-to-end Autonomous Driving

End-to-end autonomous driving directly maps sensor observations to future trajectories or control commands, avoiding the interface bottlenecks of traditional modular pipelines. Representative works such as TransFuser [5], UniAD [16], and VAD [19] establish the importance of sensor fusion, planning-oriented task design, and vectorized scene representations. A broad line of research further improves BEV representations through semantic-aware or geometry-aware learning and auxiliary tasks such as online HD map construction, temporal BEV fusion, and occupancy prediction [22, 17, 30, 39, 45, 46, 41]. Recent studies [31, 21, 10, 3, 26, 37, 44] further improve planning with sparse scene tokens, proposal-centric representations, anchor vocabularies, diffusion policies, flow matching, and reinforcement learning. Most of these methods focus on trajectory generation from current observations, while future-aware candidate evaluation remains less explored.

2.2 Trajectory Evaluation

Trajectory evaluation is crucial for safety-critical autonomous driving, especially when multiple candidate trajectories are available. Rule-based planners and simulators assess candidates using metrics such as collision, drivable-area compliance, progress, time-to-collision, and comfort. Recent end-to-end methods incorporate these signals into learning-based frameworks. Hydra-MDP [25] distills human and rule-based teachers through multiple score heads, DriveDPO [29] points out that per-anchor score regression decouples supervision from direct policy optimization, and DriveSuprim [38] improves selection-based planning through coarse-to-fine filtering and self-distillation. Different from these methods, our work focuses on the future-aware representation and computation used for candidate scoring.

2.3 World Model for Autonomous Driving

World models have been increasingly explored for autonomous driving to model future scene evolution and support anticipatory planning. Image-level driving world models predict future visual observations for scenario generation or planning [35, 4, 42], while occupancy- or latent-space models learn future scene dynamics in more compact representations [49, 23, 50]. WoTE [24] predicts future BEV states for trajectory evaluation and demonstrates the benefit of model-based scoring in BEV space. However, future-aware decision making in these methods is usually coupled with explicit future generation, latent-state prediction, or candidate-conditioned simulation. In contrast, our method shifts the computation from trajectory-specific rollout to shared-memory readout, using future supervision only to shape the memory during training and sparse spatial readout for inference-time trajectory evaluation.

3 Preliminary

3.1 Candidate-based End-to-end Planning

Given the observation at timestamp t, denoted as $\mathcal{O}_t$, the planner first generates a set of candidate trajectories

$$\mathcal{T}_t = \{\tau_i\}_{i=1}^{N},$$

where $\tau_i = \{p_i^k\}_{k=1}^T$ is the i-th candidate trajectory over T future timestamps. The final planning output is selected by

$$\tau^* = \arg\max_{\tau_i \in \mathcal{T}_t} s_i,$$

where s_i is the trajectory-level score of candidate τ_i.

3.2 Shared Memory and Spatio-Temporal Readout

To evaluate candidate trajectories under different future conditions, the planner requires a representation that is both shared across candidates and accessible at specific temporal and spatial locations. We formulate such a representation as a shared memory:

$$\mathcal{M}_t = \Phi_\theta(\mathcal{O}_t) \in \mathbb{R}^{L \times d},$$

where Φ_θ denotes the memory encoder, L is the number of memory tokens, and d is the feature dimension.

We use t for the current absolute timestamp and k for a relative future-step index, corresponding to time $t + k\Delta t$, where Δt is the interval between consecutive steps.

The shared memory is accessed through query-conditioned readout. For future step k, a temporal query q_k^{time} reads out the corresponding time-conditioned feature:

$$\bar{z}_k = \mathcal{R}(q_k^{\text{time}}, \mathcal{M}_t).$$

During training, these temporal readouts are supervised by future-timestamp targets to shape the memory with predictive temporal information. This supervision encourages $\mathcal{M}_t$ to encode future-aware scene context, but it does not define the inference pathway.

During inference, candidate evaluation relies on spatial readout. For the waypoint p_i^k, we construct a spatial query

$$q_{i,k}^{\text{space}} = \psi(p_i^k), \quad z_{i,k} = \mathcal{R}(q_{i,k}^{\text{space}}, \mathcal{M}_t),$$

where $\psi(\cdot)$ is a positional encoding function and $\mathcal{R}(\cdot)$ denotes an abstract memory readout operator. The concrete interaction mechanism is introduced in the method section. In this way, all candidate trajectories access the same temporally-shaped memory through their waypoint-conditioned spatial queries, without constructing independent future representations for each candidate.

3.3 Planning-Sufficient Interface

Under this formulation, the downstream planner operates on a relatively independent planning interface composed of the shared memory and the current ego state. Once $\mathcal{M}_t$ is constructed from the observation, the planner no longer revisits raw perceptual inputs, but evaluates candidate trajectories through memory readout with explicit ego-state conditioning:

$$P_\theta(\tau^* \mid \mathcal{O}_t, \mathcal{T}_t) = P_\theta(\tau^* \mid \mathcal{M}_t, \mathcal{E}_t, \mathcal{T}_t), \quad \mathcal{M}_t = \Phi_\theta(\mathcal{O}_t).$$

Here P_θ denotes the conditional distribution over the selected trajectory induced by the planning model, and $\mathcal{E}_t$ is the current ego state, which is directly fed into the reward encoder. Accordingly, the downstream planner depends on the original input only through the pair ($\mathcal{M}_t$,$\mathcal{E}_t$), rather than revisiting raw perceptual observations.

4 Method

Figure 2 illustrates the overall architecture of RFWorld, which consists of candidate proposal, shared-memory construction with training-time temporal shaping, and inference-time trajectory evaluation via sparse spatial readout.

4.1 Candidate Proposal and Shared Memory Construction

Given the observation at timestamp t, we denote the model input as

$$\mathcal{O}_t = \{\mathcal{H}_t, \mathcal{L}_t, \mathcal{E}_t, \mathcal{I}_t\},$$

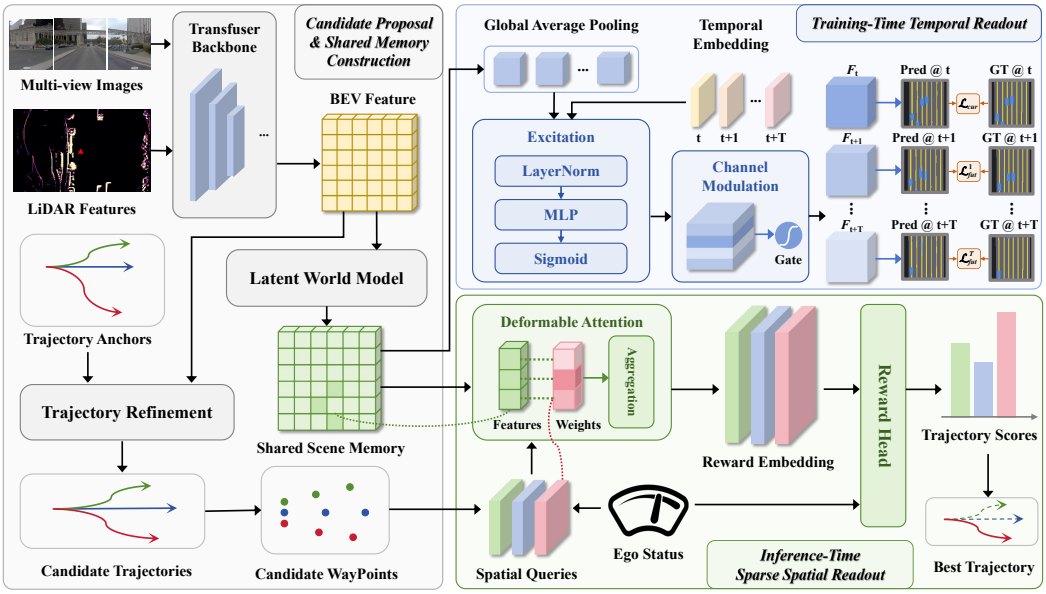


Figure 2: **Overall framework of RFWorld.** The current BEV representation is transformed into a shared memory. During training, temporal readout with semantic supervision shapes the memory into a future-aware representation. During inference, candidate trajectories directly query the shared memory for reward prediction, and the trajectory with highest score is selected as the final planning output.

where $\mathcal{H}_t$ denotes historical frame information, $\mathcal{L}_t$ denotes the current LiDAR feature, $\mathcal{E}_t$ is the current ego state, and $\mathcal{I}_t = \{I_t^v\}_{v=1}^{N_c}$ denotes the current multi-view images. Following the multi-modal fusion paradigm of TransFuser [5], we align historical LiDAR features to the current ego coordinate frame and fuse them with the current visual observations to obtain a BEV scene representation B_t.

For planning, we adopt an anchor-based candidate proposal branch [3, 24]. Let $\mathcal{A} = \{a_i\}_{i=1}^{N}$ denote a predefined vocabulary of trajectory anchors. Each anchor interacts with the current scene representation to predict an offset, yielding the refined candidate trajectory

$$\tau_i = a_i + \Delta_i, \quad i = 1, \ldots, N.$$

Different from candidate-conditioned world models that construct separate future representations for different trajectories [24, 47], our method builds a trajectory-agnostic shared memory from the current BEV representation. Specifically, the current BEV feature B_t is flattened, augmented with spatial positional embeddings, and processed by a Transformer-based [33] latent world model to obtain

$$\mathcal{M}_t = \Phi_{\text{mem}}(\text{Flat}(B_t) + P) \in \mathbb{R}^{L \times d},$$

where P denotes the spatial positional embedding, L is the number of BEV tokens, and d is the hidden dimension. The shared memory is conditioned only on the observation and is reused by both the training and inference branches.

4.2 Training-Time Temporal Readout with Semantic Supervision

To endow the shared memory with predictive temporal structure, we attach a lightweight temporal readout branch during training. Let

$$\bar{\mathcal{M}}_t = \text{Reshape}(\mathcal{M}_t) \in \mathbb{R}^{d \times H \times W}$$

denote the spatial BEV form of the shared memory, where H and W are its height and width ($L = HW$), and let

$$c_t = \text{AvgPool}(\bar{\mathcal{M}}_t) \in \mathbb{R}^d$$

be its globally pooled context. Let $\mathcal{S}$ be the selected supervised temporal indices, including 0 for the current timestamp. For each $k \in \mathcal{S}$, we introduce a learnable temporal embedding e_k and perform SE-style channel modulation [14]:

$$g_k = 2 \cdot \sigma(\mathrm{MLP}(\mathrm{LN}(c_t + e_k))) \in \mathbb{R}^d,$$

$$\bar{\mathcal{M}}_t^{(k)} = \bar{\mathcal{M}}_t \odot g_k.$$

The gate g_k is broadcast across spatial positions to reweight the memory channels. The resulting time-conditioned feature is decoded into a BEV semantic map by the shared semantic decoder $\mathcal{D}$:

$$\hat{Y}_t^{(k)} = \mathcal{D}\Big(\bar{\mathcal{M}}_t^{(k)}\Big).$$

The decoder is shared across all selected timestamps, so temporal diversity is induced solely by the temporal readout. The current prediction $\hat{Y}_t^{(0)}$ is supervised by the current semantic target, while the future predictions $\{\hat{Y}_t^{(k)}\}_{k\in\mathcal{S}\setminus\{0\}}$ are supervised by the selected future semantic targets. We denote the resulting semantic supervision as

$$\mathcal{L}_{\mathrm{sem}} = \mathcal{L}_{\mathrm{cur}} + \sum_{k\in\mathcal{S}\setminus\{0\}} \omega_k \, \mathcal{L}_{\mathrm{fut}}^{(k)},$$

where ω_k denotes the normalized weight of each supervised future timestamp. This branch is used only to shape the shared memory during training and does not define the inference-time planning path.

4.3 Inference-Time Sparse Spatial Readout

During inference, we do not explicitly decode future semantic maps or perform candidate-wise rollout. Instead, for each candidate trajectory $\tau_i = \{p_i^k\}_{k=1}^T$, we directly read the information required for trajectory evaluation from the shared memory through deformable spatial attention [51]. Specifically, each waypoint p_i^k is encoded as a spatial query

$$q_i^k = \psi(p_i^k),$$

where $\psi(\cdot)$ denotes the waypoint encoder. Ego-state context is injected into the query features to modulate the readout. The offsets and attention weights are learned by Γ; bilinear interpolation in V is only the differentiable operation used to sample at the learned continuous locations.

Given query q_i^k and its reference point r_i^k on the BEV plane, the reward encoder predicts a set of sampling offsets $\Delta_{i,k}^m$ and attention weights $\alpha_{i,k}^m$:

$$\{\Delta_{i,k}^m,\ \alpha_{i,k}^m\}_{m=1}^M = \Gamma(q_i^k),$$

where $\Gamma(\cdot)$ denotes the sampling-parameter prediction head, and M is the number of sampling points. The waypoint feature is then obtained by sparse weighted sampling from $\bar{\mathcal{M}}_t$:

$$z_i^k = \sum_{m=1}^{M} \alpha_{i,k}^m \, V\big(\bar{\mathcal{M}}_t,\ r_i^k + \Delta_{i,k}^m\big),$$

where $V(\cdot)$ denotes bilinear sampling on the shared memory. The waypoint features are aggregated along the trajectory horizon to produce a trajectory embedding e_i, which is further fused with ego-state context before reward prediction.

Following prior trajectory-evaluation frameworks [24], we predict an imitation reward and a set of simulation rewards:

$$r_i^{\mathrm{im}} = \mathrm{Head}_{\mathrm{im}}(e_i), \qquad r_i^{\mathrm{sim}} = \mathrm{Head}_{\mathrm{sim}}(e_i).$$

The former captures similarity to expert behavior, whereas the latter evaluates the candidate under simulator-derived safety and planning criteria. At inference time, these rewards are aggregated into a final trajectory score, and the highest-scoring trajectory is selected as the planning output.

4.4 Training Objective

We train the model with a compact multi-term objective consisting of trajectory, semantic, imitation-reward, and simulation-reward supervision. For trajectory prediction, only the candidate anchor closest to the ground-truth trajectory is supervised, yielding the trajectory loss $\mathcal{L}_{\mathrm{traj}}$. The semantic supervision in Sec. 4.2 is denoted as $\mathcal{L}_{\mathrm{sem}}$, where the semantic maps are supervised by a focal loss. For imitation reward learning, we construct a soft target distribution from the relative distances between candidate trajectories and the expert trajectory, and optimize the predicted imitation reward with a cross-entropy-style objective, yielding $\mathcal{L}_{\mathrm{im}}$. For simulation reward learning, the predicted heads for NC, DAC, EP, TTC, and Comfort are supervised by simulator-derived targets using BCE loss; the additional EP margin ranking regularization is absorbed into $\mathcal{L}_{\mathrm{sim}}$ for brevity.

The overall training objective is

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{traj}}\mathcal{L}_{\mathrm{traj}} + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}} + \lambda_{\mathrm{im}}\mathcal{L}_{\mathrm{im}} + \lambda_{\mathrm{sim}}\mathcal{L}_{\mathrm{sim}},$$

where λ_{traj}, λ_{sem}, λ_{im}, and λ_{sim} are the corresponding loss weights.

5 Experiments

5.1 Benchmark

NAVSIM. We evaluate RFWorld on NAVSIM-v1 [7], which is built upon real-world sensor data and emphasizes safety-critical and behaviorally diverse planning scenarios. The official trainval split contains 85,109 training samples and 18,179 validation samples; the official test split contains 12,146 scenarios. For each test scenario, the model selects exactly one final trajectory, and each metric is computed once for that scenario; reported values are averages over the 12,146 scenarios. The primary metric is the Predictive Driver Model Score (PDMS), which aggregates No At-Fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (Comf.). We report PDMS together with all constituent metrics. For normalized subscores in $[0, 1]$, NAVSIM-v1 computes $\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot (5\,\mathrm{TTC} + 5\,\mathrm{EP} + 2\,\mathrm{Comf.})/12$.

Bench2Drive. We evaluate RFWorld on Bench2Drive [18], a closed-loop autonomous driving benchmark in CARLA 0.9.15, using the official 220 test routes. We report Success Rate (%) and Driving Score.

5.2 Implementation Details

NAVSIM. We train and evaluate RFWorld on NAVSIM-v1 under the official data split. For fair comparison with prior work, we adopt the same ResNet-34 [11] backbone as Transfuser [5]. The model takes LiDAR features, ego states, and multi-view images as input, and predicts scores over 1024 candidate trajectory anchors. NAVSIM experiments are conducted on 2 RTX 3090 GPUs with a batch size of 16. We train the model for 50 epochs using AdamW with a base learning rate of 1×10^{-4}. Unless otherwise specified, these settings are shared across the NAVSIM experiments.

Bench2Drive. The input consists of three front-facing cameras, four consecutive LiDAR frames, the ego state, and a navigation command. The three images are stitched and resized to 1024×256; LiDAR points are rasterized into a 256×256 BEV grid covering 32 m in each direction at 0.25 m resolution. We initialize from the original RFWorld pretrained weights and train with AdamW for 30,000 iterations using batch size 8, learning rate 5×10^{-6}, weight decay 10^{-4}, and gradient clipping at 5. At inference, 1,024 candidate trajectories are ranked by their predicted rewards and the highest-scoring trajectory is selected at every simulation step. A vehicle PID controller converts it into steering, throttle, and brake commands by tracking the sixth trajectory point, with throttle capped at 0.55.

5.3 Comparison with SOTA

NAVSIM. As shown in Table 1, RFWorld achieves a PDMS of 90.5 under the ResNet-34 setting, with the highest PDMS, NC, and TTC among the compared learned methods. These results suggest that our future-aware modeling and trajectory evaluation pipeline can more accurately capture future scene evolution and more effectively suppress unsafe trajectory candidates.

Table 1: Comparison with state-of-the-art end-to-end driving methods on NAVSIM. The best results are denoted in **bold**. WM indicates whether the method explicitly employs a world model. For fair comparison, we report the scores using the same ResNet-34 backbone.

Method	WM	Supervision	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Human	–	–	100.0	100.0	99.9	**100.0**	87.5	94.8
Ego Status MLP	×	Human	93.0	77.3	83.6	**100.0**	62.8	65.6
VADv2 [3]	×	Human	97.2	89.1	91.6	**100.0**	76.0	80.9
UniAD [16]	×	Human	97.8	91.9	92.9	**100.0**	78.8	83.4
LTF [5]	×	Human	97.4	92.8	92.4	**100.0**	79.0	83.8
Transfuser [5]	×	Human	97.7	92.8	92.8	**100.0**	79.2	84.0
DRAMA [40]	×	Human	98.0	93.1	94.8	**100.0**	80.1	85.5
GoalFlow [37]	×	Human	98.3	93.8	94.3	**100.0**	79.8	85.7
Hydra-MDP [25]	×	Human & Rule	98.3	96.0	94.6	**100.0**	78.7	86.5
Hydra-MDP++ [20]	×	Human & Rule	97.6	96.0	93.1	**100.0**	80.4	86.6
DiffusionDrive [26]	×	Human	98.2	96.2	94.7	**100.0**	82.2	88.1
DriveSuprim [38]	×	Human & Rule	97.8	97.3	93.6	**100.0**	**86.7**	89.9
DriveDPO [29]	×	Human & Rule	98.5	**98.1**	94.8	99.9	84.3	90.0
LAW [23]	✓	Human	96.4	95.4	88.7	99.9	81.7	84.6
WoTE [24]	✓	Human & Rule	98.5	96.8	94.9	99.9	81.9	88.3
ResWorld [47]	✓	Human	98.9	96.5	95.6	**100.0**	83.1	89.0
RFWorld (Ours)	✓	Human & Rule	**99.0**	97.8	**96.0**	**100.0**	84.6	**90.5**

Table 2: Closed-loop results on the official 220-route Bench2Drive [18] test set.

Method	Success Rate (%)↑	Driving Score↑
AD-MLP [43]	0.00	18.05
UniAD [16]	16.36	45.81
VAD [19]	15.00	42.35
TCP [36]	30.00	59.90
RFWorld	**31.36**	**62.38**

Bench2Drive. As shown in Table 2, RFWorld achieves a Success Rate of 31.36% and a Driving Score of 62.38, exceeding TCP by 1.36 percentage points and 2.48 score points, respectively, and leading the compared baselines on both metrics.

5.4 Ablation Study

Effect of world model and future supervision. We first study whether the improvement comes from introducing a world model itself or from learning a predictive representation with explicit future supervision. The four rows in Table 3 differ only in the two indicated switches; the backbone, anchors, optimization schedule, split, and evaluation protocol are fixed. Directly adding the world model without future supervision does not improve performance, suggesting that a future branch optimized only through the final planning objective is insufficient to learn stable and informative predictive features.

In contrast, explicitly supervising future readout features consistently improves all major metrics. In the WM × / Future Sup. ✓ setting, temporal semantic readout supervises the backbone BEV feature directly, without the world model Transformer. Future supervision alone improves PDMS from 88.9 to 89.4, while the complete RFWorld model reaches 90.5, indicating complementary roles for future supervision and shared-memory world modeling.

Effect of the number of inference anchors. We study how increasing the number of inference-time candidate trajectories affects performance and cost. For fair comparison, both WoTE and RFWorld are trained with 1024 anchors and a batch size of 16, and different numbers of anchors are randomly sampled at test time. Due to their different training memory demands, WoTE is trained on two A6000

Table 3: Ablation study on world modeling and future supervision. "WM" denotes whether the world model branch is enabled, and "Future Sup." denotes whether future readout features are explicitly supervised during training. The best results are shown in **bold**.

WM	Future Sup.	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
×	×	98.7	94.4	95.2	99.8	83.4	88.9
×	✓	98.8	97.1	95.4	**100.0**	83.7	89.4
✓	×	98.5	96.5	95.1	99.8	82.3	88.4
✓	✓	**99.0**	**97.8**	**96.0**	**100.0**	**84.6**	**90.5**

Table 4: Ablation study on inference-time candidate anchors. "#Traj." denotes the number of candidate trajectories randomly sampled at test time. Latency is measured per sample on a single RTX 3090 GPU. The best results are shown in **bold**.

Method	#Traj.	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑	Latency↓
WoTE [24]	256	98.4	96.7	94.6	99.9	82.3	88.3	37.0 ms
WoTE [24]	512	98.3	96.3	94.5	99.8	83.4	88.4	41.2 ms
WoTE [24]	1024	98.4	97.1	94.8	**100.0**	82.5	88.7	49.2 ms
RFWorld	256	98.8	97.0	95.7	99.9	83.9	89.6	35.1 ms
RFWorld	512	98.9	97.7	95.5	**100.0**	84.4	90.2	36.2 ms
RFWorld	1024	**99.0**	**97.8**	**96.0**	**100.0**	**84.6**	**90.5**	37.5 ms

GPUs (48 GB each), whereas RFWorld is trained on two RTX 3090 GPUs (24 GB each). Inference latency is measured on a single RTX 3090 GPU.

As shown in Table 4, RFWorld improves consistently as the candidate set grows from 256 to 1024, indicating that richer candidate coverage is more effectively utilized in RFWorld. This trend is consistent with the underlying complexity difference. Let N denote the number of candidate trajectories, T the planning horizon, L the number of BEV tokens, and M the number of sampling points in sparse readout. WoTE scales as $O(NT \cdot \mathcal{C}_{\mathrm{WM}}(L))$, approximately $O(NTL^2d)$ when transformer world-model computation dominates, because future prediction is repeated for every candidate over the whole horizon. In contrast, RFWorld constructs the shared memory once with cost $O(\mathcal{C}_{\mathrm{WM}}(L))$ and scores candidates with sparse spatial readout at $O(NTMd)$. Since $M \ll L$, RFWorld removes the dominant NT factor from future-state generation. The same advantage appears in training memory: under the 1024-anchor, batch-size-16 setting, RFWorld uses 8.7 GB GPU memory, compared with 33.9 GB for WoTE. Consistently, RFWorld remains faster at all tested anchor counts, with latency increasing only from 35.1 ms to 37.5 ms as N grows from 256 to 1024, while keeping the evaluation protocol fixed across all settings.

5.5 Qualitative Comparison

We further provide qualitative comparisons with WoTE [24] in Fig. 3. These cases indicate a broader advantage of RFWorld over rollout-based trajectory evaluation paradigms, where early rollout inaccuracies may accumulate over time and induce unreliable future assessments. In contrast, RFWorld better suppresses such compounding errors, leading to safer and more robust planning behaviors.

6 Limitations and Future Work

RFWorld has several limitations. First, its Transformer-based latent world model is not well suited to higher-resolution future prediction on dense BEV features; replacing dense BEV features with sparse scene tokens may improve both scalability and efficiency. Second, the quality of the shared memory remains dependent on the form of memory-shaping supervision, and the semantic targets are currently needed during training, motivating future work on pseudo-labeling and self-supervised temporal objectives. Third, observed failures mainly occur when the lightweight perception backbone produces inaccurate scene features or when relevant road geometry and traffic participants lie near or beyond

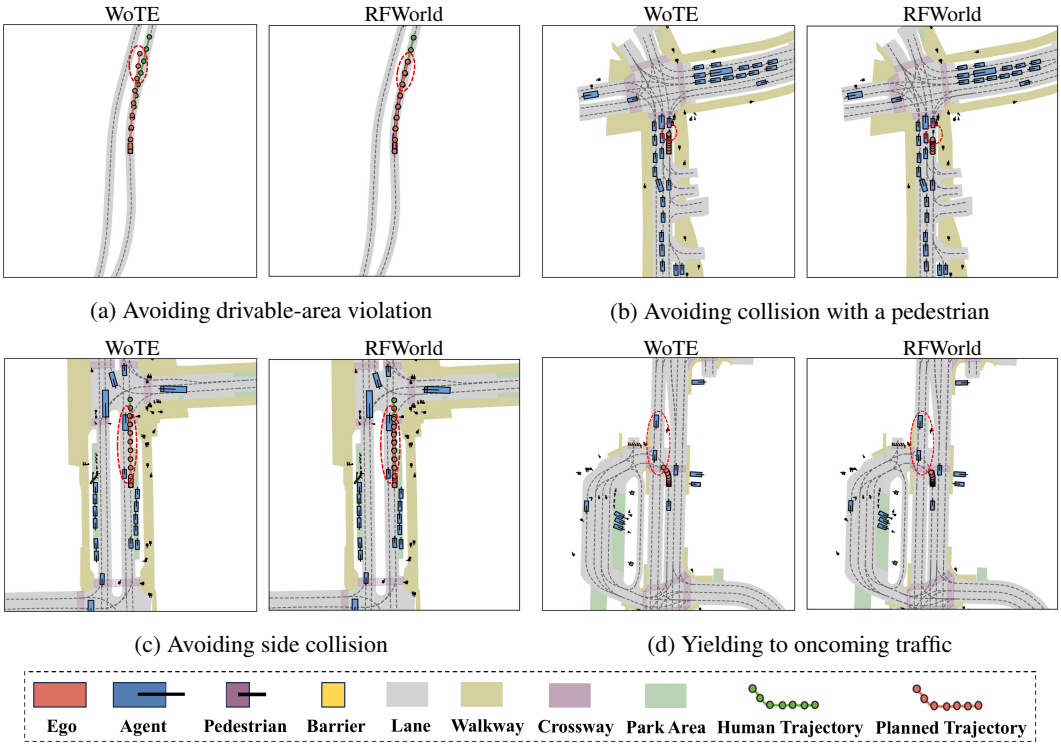


Figure 3: Qualitative comparison between WoTE [24] and RFWorld on NAVSIM-v1. Legend from DriveDPO [29]. (a) RFWorld avoids leaving the drivable area and remains within the lane boundaries. (b) RFWorld decelerates in advance and safely avoids the pedestrian. (c) RFWorld stays close to the lane centerline and avoids potential collision with the adjacent-lane vehicle. (d) RFWorld better accounts for oncoming traffic and yields before turning, thereby avoiding potential future risk.

the observation range. These errors can propagate to trajectory scoring and lead to drivable-area departures or collisions.

7 Conclusion

In this paper, we revisit model-based trajectory evaluation for end-to-end autonomous driving and identify a key mismatch in existing future-aware evaluators: they often rely on explicit candidate-conditioned future rollout, although trajectory selection only requires future-relevant information for candidate comparison. To address this issue, we propose RFWorld, a rollout-free world model for end-to-end trajectory evaluation. RFWorld constructs a shared scene memory, shapes it through training-time temporal readout with semantic supervision, and performs inference-time trajectory scoring via deformable spatial reward readout under explicit ego-state conditioning. Extensive experiments on the NAVSIM benchmark demonstrate that RFWorld provides a more planning-aligned world representation for future-aware trajectory evaluation and achieves state-of-the-art performance. These results show that replacing candidate-specific future rollout with shared-memory readout is an effective direction for end-to-end autonomous driving.

Acknowledgments and Disclosure of Funding

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020016 and National Natural Science Foundation of China under Grant No. 62576023. The authors declare no competing interests.

References

[1] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15590–15599, 2021.

[2] 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.

[3] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[4] Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. DrivingGPT: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26890–26900, 2025.

[5] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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

[7] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, pages 28706–28719, 2024.

[8] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[9] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems*, 2024.

[10] Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. iPad: Iterative proposal-centric end-to-end autonomous driving. *arXiv preprint arXiv:2505.15111*, 2025.

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

[12] Mikael Henaff, Alfredo Canziani, and Yann LeCun. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. In *International Conference on Learning Representations*, 2019.

[13] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[15] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.

[16] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

[17] Junjie Huang and Guan Huang. BEVDet4D: Exploit temporal cues in multi-camera 3D object detection. *arXiv preprint arXiv:2203.17054*, 2022.

[18] 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.

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

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

[21] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *International Conference on Learning Representations*, 2025.

[22] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An online hd map construction and evaluation framework. In *IEEE International Conference on Robotics and Automation*, pages 4628–4634, 2022.

[23] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, 2025.

[24] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27137–27146, 2025.

[25] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

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

[27] Guanren Qiao, Guorui Quan, Rongxiao Qu, and Guiliang Liu. Modelling competitive behaviors in autonomous driving under generative world model. In *European Conference on Computer Vision*, pages 19–36. Springer, 2024.

[28] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *European Conference on Computer Vision*, pages 414–430. Springer, 2020.

[29] Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and Zhaoxiang Zhang. DriveDPO: Policy learning via safety dpo for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, 2025.

[30] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.

[31] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2025.

[32] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drive-Dreamer: Towards real-world-driven world models for autonomous driving. In *European Conference on Computer Vision*, 2024.

[35] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[36] 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 *Advances in Neural Information Processing Systems*, 2022.

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

[38] Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11910–11918, 2026.

[39] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. FlashOCC: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023.

[40] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

[41] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7356–7365, 2024.

[42] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. FutureSightDrive: Thinking visually with spatio-temporal CoT for autonomous driving. In *Advances in Neural Information Processing Systems*, 2025.

[43] 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.

[44] Dongkun Zhang, Jiaming Liang, Ke Guo, Sha Lu, Qi Wang, Rong Xiong, Zhenwei Miao, and Yue Wang. CarPlanner: Consistent auto-regressive trajectory planning for large-scale reinforcement learning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17239–17248, 2025.

[45] Jinqing Zhang, Yanan Zhang, Qingjie Liu, and Yunhong Wang. SA-BEV: Generating semantic-aware bird's-eye-view feature for multi-view 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3348–3357, 2023.

[46] Jinqing Zhang, Yanan Zhang, Yunlong Qi, Zehua Fu, Qingjie Liu, and Yunhong Wang. GeoBEV: Learning geometric BEV representation for multi-view 3D object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9960–9968, 2025.

[47] Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. ResWorld: Temporal residual world model for end-to-end autonomous driving. In *The Fourteenth International Conference on Learning Representations*, 2026.

[48] Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. BEVWorld: A multimodal world model for autonomous driving via unified BEV latent space. *arXiv preprint arXiv:2407.05679*, 2024.

[49] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D occupancy world model for autonomous driving. In *European Conference on Computer Vision*, 2024.

[50] Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, XianPeng Lang, and Dongbin Zhao. World4Drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28632–28642, 2025.

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

A Further Ablation Study

Effect of the number of supervised future timestamps. We further study how many future timestamps should be supervised during training to effectively shape the shared memory. As shown in Table 5, planning performance improves consistently as more future timestamps are supervised. For intermediate settings, timestamps are randomly sampled from both the earlier and later halves of the horizon, which encourages temporally balanced predictive shaping. This result suggests that future supervision is essential for injecting predictive temporal structure into the shared memory.

Table 5: Ablation study on the number of supervised future timestamps. "#Future Sup." denotes the number of future timestamps used for semantic supervision during training. "0" corresponds to the setting with world model but without future supervision. The best results are shown in **bold**.

#Future Sup.	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
0	98.5	96.5	95.1	99.8	82.3	88.4
2	98.8	97.0	95.7	**100.0**	83.7	89.5
4	99.0	97.3	95.9	99.9	84.0	90.0
6	99.0	97.4	**96.2**	99.9	84.3	90.3
8	**99.0**	**97.8**	96.0	**100.0**	**84.6**	**90.5**

When no future timestamp is supervised, the world model branch is optimized only indirectly through the final planning objective, which is insufficient to reliably expose future-relevant information for downstream trajectory evaluation. Once future supervision is introduced, performance improves steadily as more timestamps are covered. We attribute this trend to the fact that a denser set of supervised future readouts provides stronger temporal constraints on the shared memory, encouraging it to encode scene evolution over a broader horizon.

B Additional Visualization Results

B.1 Readout Semantic Maps During Training

To further examine whether the training-time temporal readout indeed shapes the shared memory into a future-aware representation, we visualize in Fig. 4 the semantic maps decoded from the readout features at four future timestamps. To make the temporal differences easier to inspect, representative dynamic objects are highlighted by red dashed boxes in both the predicted maps and the corresponding ground truth maps. Since the semantic decoder is shared across all timestamps, the differences across columns are induced solely by the time-conditioned readout rather than by timestamp-specific decoding heads.

As shown in Fig. 4, the predicted maps consistently preserve the major scene semantics, including the drivable region, centerline structure, and the coarse layout of surrounding vehicles. More importantly, the highlighted regions show that the readout features capture temporally evolving vehicle configurations across future timestamps, rather than simply reproducing the current scene. Although minor deviations in shape and occupancy remain, the overall temporal evolution is largely consistent with the ground truth.

These observations provide qualitative evidence that temporal readout with future semantic supervision injects predictive temporal structure into the shared memory. In other words, the readout branch learns semantically meaningful future features during training, which helps shape a planning-oriented memory for subsequent trajectory evaluation.

B.2 Temporal Structure of Readout Gates

To further understand how the temporal readout modulates the shared memory across timestamps, we analyze the learned readout gates from two complementary perspectives. We first examine channel-wise temporal modulation, which reveals how individual channels are reweighted when the readout moves from the current timestamp to future timestamps. We then study the global statistics of the gate distribution, which summarize how the overall selectivity of temporal readout evolves over the prediction horizon.

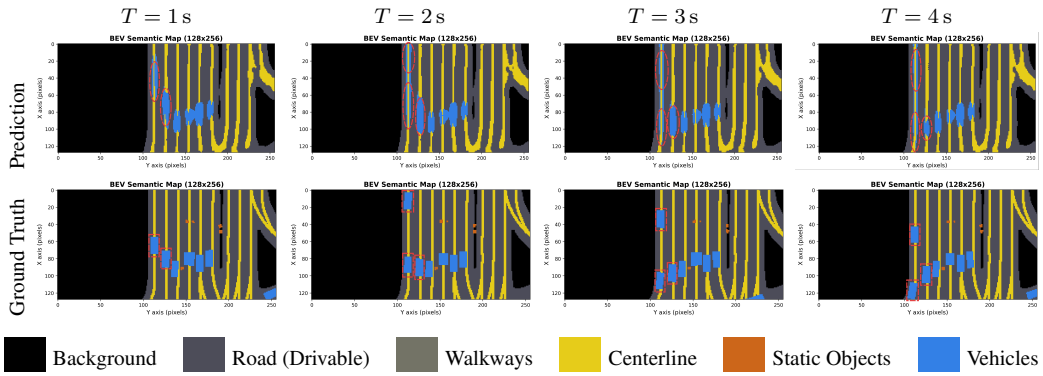
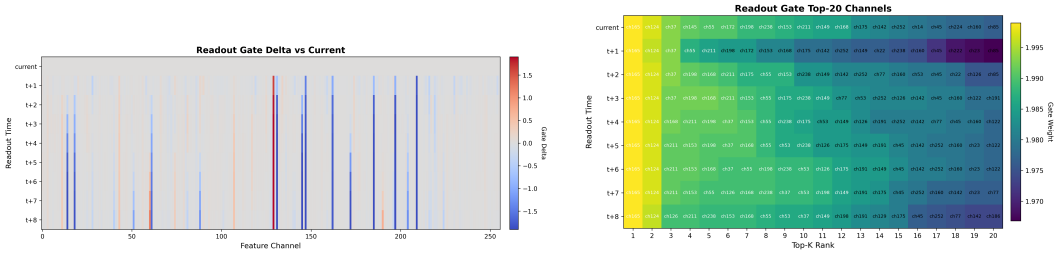


Figure 4: Visualization of the readout semantic maps during training. The top row shows the semantic maps predicted from the time-conditioned readout features, and the bottom row shows the corresponding ground-truth semantic maps. From left to right, the columns correspond to four future timestamps at $T = 1\,\mathrm{s}$, $2\,\mathrm{s}$, $3\,\mathrm{s}$, and $4\,\mathrm{s}$. Red dashed boxes highlight representative dynamic regions to facilitate comparison of their temporal evolution.



(a) Channel-wise gate differences relative to the current timestamp.

(b) Top-20 channels with the largest gate values at each timestamp.

Figure 5: Channel-wise analysis of temporal readout gates. Left: gate differences between each future timestamp and the current timestamp. Right: Top-20 channels ranked by gate value at each timestamp.

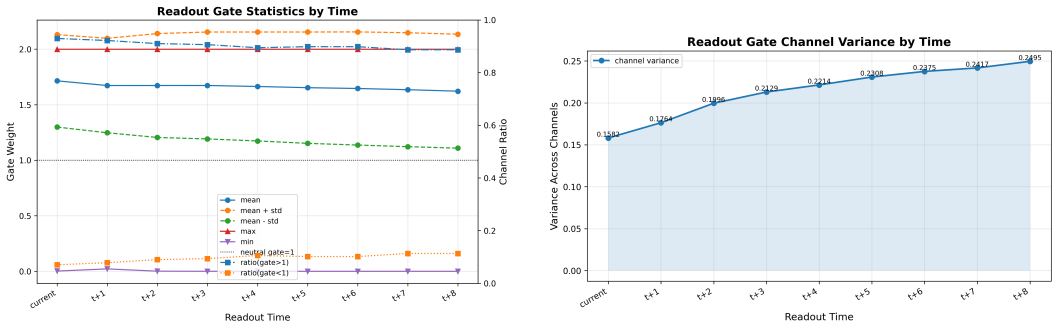
B.2.1 Channel-wise Temporal Modulation

We first investigate the channel-level behavior of temporal readout in Fig. 5. The gate-delta heatmap visualizes, for each future timestamp, the difference between its gate values and those at the current timestamp. A clear pattern can be observed: most channels remain close to the current readout, while only a relatively small subset exhibits noticeable positive or negative deviations. This suggests that temporal readout does not construct future representations independently for each timestamp. Instead, it performs structured modulation on top of a shared memory, preserving a common channel basis while selectively adjusting a subset of channels according to the queried timestamp.

The Top-20 visualization provides a complementary view of this behavior. Several highly weighted channels remain consistently active across timestamps, indicating that part of the channel basis is shared by all temporal readouts. Meanwhile, the lower-ranked channels change gradually as the timestamp moves farther into the future, showing that temporal conditioning mainly acts by refining the supporting channels around a stable core set. Taken together, these results indicate that the temporal readout follows a "shared basis + timestamp-specific modulation" pattern, which is consistent with the design goal of RFWorld: the shared memory stores reusable scene structure, while the readout gate exposes timestamp-relevant predictive information through lightweight channel reweighting.

B.2.2 Distributional Statistics Across Timestamps

We next summarize the temporal readout behavior at the distribution level in Fig. 6. The gate statistics show that the mean gate value stays above the neutral value across all timestamps, and



(a) Summary statistics of gate values across timestamps.

(b) Variance of gate values across channels at each timestamp.

Figure 6: Distributional analysis of temporal readout gates. Left: aggregated statistics of gate values at each timestamp. Right: channel-wise variance of gate values at each timestamp.

that the proportion of channels with gate values larger than the neutral value consistently dominates the proportion below it. This indicates that the temporal readout mainly operates by enhancing informative channels rather than suppressing the majority of channels. At the same time, as the readout timestamp moves farther into the future, the mean gate value decreases slightly and the fraction of down-weighted channels grows modestly, suggesting that longer-horizon readout becomes progressively more selective.

This tendency is further corroborated by the variance plot. The variance of gate values across channels increases steadily with the timestamp, indicating that the channel weights become more dispersed for farther-future readout. In other words, the readout gate applies increasingly differentiated reweighting over channels as the timestamp advances. This observation complements the channel-wise analysis above: near-future readout remains closer to the current feature basis, whereas farther-future readout requires stronger channel discrimination to extract more predictive temporal information from the shared memory.

Overall, the two levels of visualization form a coherent picture. Temporal readout in RFWorld is neither static nor arbitrary. It preserves a stable shared channel basis across timestamps, while gradually increasing channel-wise selectivity as the queried timestamp moves farther into the future. This behavior supports the main-text claim that future supervision shapes a shared memory with predictive temporal structure, and that such structure can be exposed efficiently through sparse timestamp-conditioned readout rather than explicit future-state rollout.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are consistent with the method and the NAVSIM and Bench2Drive results. See Abstract, Secs. 1 and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of the proposed method, including scalability, supervision dependence, and perception/observation failures, are discussed in Sec. 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper does not present theoretical results requiring formal assumptions or proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper specifies the model design, training objectives, benchmark, evaluation metrics, implementation details, and ablation settings. See Secs. 4 and 5.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The RFWorld implementation and documentation are publicly available at `https://github.com/Rupertail/RFWorld`. The repository provides the code and instructions for reproducing the reported experiments; the NAVSIM and Bench2Drive data remain available under their respective official access terms.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The paper describes the benchmark, evaluation metrics, model inputs, backbone, optimizer, learning rate, batch size, training schedule, and closed-loop protocol. See Sec. 5.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Across five NAVSIM training seeds, PDMS is 90.4676 ± 0.0838 (mean $\pm$ sample standard deviation), with a two-sided 95% Student's t confidence interval of $[90.3635, 90.5717]$, assuming independent and approximately normally distributed run scores. These statistics are reported here only; the main tables and closed-loop results do not include corresponding uncertainty estimates.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Secs. 5.2 and 5.4 report the NAVSIM hardware, training settings, memory, and inference latency, but do not report complete training times, total project compute, or the closed-loop hardware.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: To the best of our knowledge, the research described in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: The current submission does not include a dedicated discussion of potential positive and negative societal impacts.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper does not release high-risk assets such as large generative models or scraped datasets that require special safeguards.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [No]

Justification: We cite the datasets, models, and prior methods used in the paper; the NAVSIM and Bench2Drive assets remain subject to their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper does not introduce a separate new dataset or asset package. The implementation and its documentation are provided in the public RFWorld repository at `https://github.com/Rupertail/RFWorld`.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: This work does not involve crowdsourcing or human-subject research requiring IRB review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the core methodology in this research.

Guidelines:

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