DriveFuture: Future-Aware Latent World Models for Autonomous Driving

Yufeng Hong[1†] **Xiaotian Zhou**[1†] **Yingyan Li**[2] **Xiangpo Zhou**[3]
Lin Liu[4] **Yadan Luo**[5] **Shaoqing Xu**[6] **Lei Yang**[7] **Ziying Song**[8,7*]

[1]Beijing Institute of Technology
[2]Institute of Automation, Chinese Academy of Sciences
[3]Beihang University [4]Beijing Jiaotong University
[5]The University of Queensland [6]University of Macau
[7]Nanyang Technological University
[8]School of Artificial Intelligence (School of Software), Yanshan University

[†]Equal contribution [*]Corresponding author

Abstract

Existing latent world models for autonomous driving have opened a promising path toward future-aware driving intelligence. However, they typically treat future latent states as prediction targets or auxiliary signals, rather than directly conditioning trajectory planning. This can entangle current and future features in latent space. In this work, we propose DriveFuture, a future-aware latent world modeling framework for autonomous driving that explicitly learns planning-oriented foresight by conditioning the current latent state modeling process on future world states. Specifically, during training, the model first predicts future latent world states from the current latent state and ego action, and then refines the prediction against the ground-truth future latent state via cross-attention. The resulting future-aware latent serves as an explicit condition for a diffusion-based trajectory planner. During inference, DriveFuture conditions on the predicted future latent state instead of the ground-truth future state. DriveFuture achieves SOTA performance on the public NAVSIM benchmarks [1], reaching **55.5** EPDMS on NAVSIM-v2 *navhard*, **89.9** EPDMS on NAVSIM-v2 *navtest*, and **90.7** PDMS on NAVSIM-v1 *navtest*, respectively. These results suggest that the key to latent world modeling lies not merely in simulating future states, but more importantly in conditioning current decision-making on future states. Notably, as of April 2026, DriveFuture ranks **1st** on the NAVSIM-v2 *navhard* leaderboard and achieves SOTA performance on NAVSIM-v1 *navtest*. The code is publicly available at `https://github.com/damonxzhou/DriveFuture`.

1 Introduction

In recent years, world models have gradually emerged as a central direction in autonomous driving and are widely regarded as one of the most promising technical paradigms for achieving higher-level driving intelligence [2–6]. Unlike conventional approaches [7–16], which directly map the current state to driving actions, world models [2, 17] learn the latent dynamics of the driving environment. This allows the system to anticipate future outcomes under different actions and make more informed planning decisions. By reasoning about how the world may evolve before selecting an action, this paradigm shifts autonomous driving from reactive control toward future-aware decision-making. As a result, world models are evolving from an auxiliary modeling tool into a key foundation for scalable and generalizable autonomous driving.

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

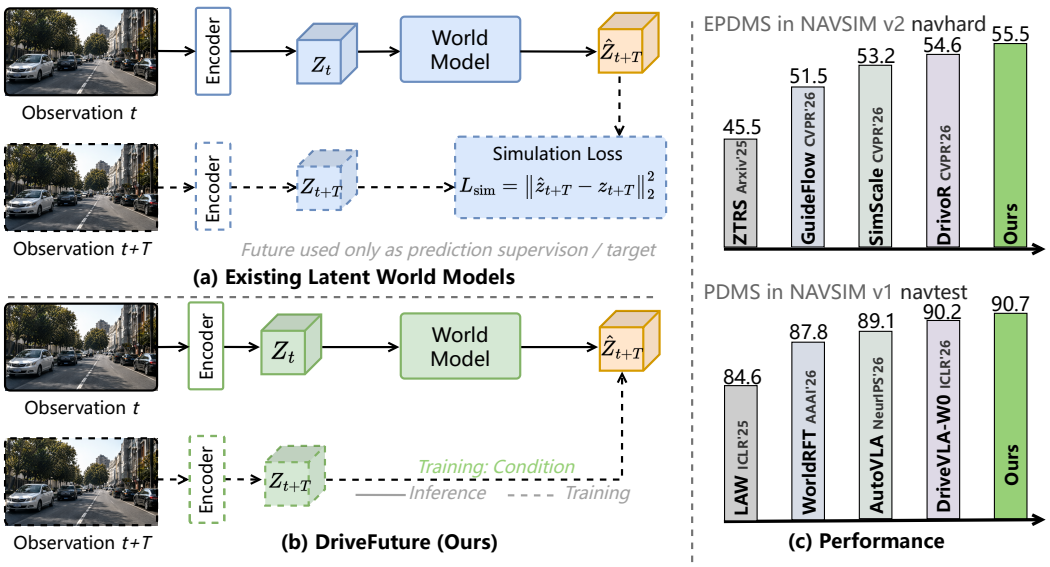


Figure 1: Motivation of **DriveFuture**. (a) Existing latent world models [17–19, 23, 20–22] primarily simulate future latent states and use them as prediction targets or supervision signals, without explicitly shaping the current representation for planning. (b) DriveFuture uses future latent states as direct conditions for the planning process. It adopts GT future states during training and predicted future states during inference, enabling future-aware trajectory planning. (c) DriveFuture achieves SOTA performance on the public leaderboards of NAVSIM-v2 *navhard* and NAVSIM-v1 *navtest*.

Among existing world-modeling paradigms, latent world models are particularly promising for autonomous driving for three reasons. First, they learn dynamics in compact latent spaces, avoiding costly pixel-level generation. Second, latent representations can abstract away low-level visual details and focus on planning-relevant scene structure, interactions, and action consequences. Third, their temporally structured representations are naturally suited for long-horizon planning. This line of work has rapidly evolved from latent future prediction [17] to intention-aware latent planning [18], planning-oriented representation refinement with reinforcement fine-tuning [19], and more recent unifications with VLA/planners and policy scaling [20–23]. These developments indicate that latent world models are no longer merely an efficient substitute for observation-space simulation, but are increasingly becoming a general representation substrate for scalable autonomous driving.

Although latent world models [17–19, 23, 20–22] provide an efficient paradigm for future modeling in autonomous driving, existing methods primarily focus on *future state prediction* rather than *current decision representation*. Specifically, as shown in Fig. 1, they typically treat future latent states as prediction targets or auxiliary signals, rather than using them as explicit conditions to shape the current decision-making process, which leads to entanglement between current and future features in latent space. As a result, these models are better at *simulating the future* than at learning planning-oriented decision representations, thereby limiting the ability of future information to guide current decision-making.

We argue that the role of future information in autonomous driving is not merely to predict how future scenes will evolve, but more importantly to endow the system with a decision-making perspective akin to that in *The Terminator*, namely, *bringing future knowledge back to the present*: using awareness of future consequences at the current moment to reshape the understanding of the present scene and the resulting decision. An ideal autonomous driving system, therefore, should not be merely a future predictor, but should instead allow future states to act as conditions that influence the current decision-making process in reverse, thereby enabling more foresighted planning. Inspired by this observation, we propose **DriveFuture**, a future-aware latent world modeling framework for autonomous driving. By conditioning the current decision-making process on future world states, DriveFuture explicitly learns planning-oriented *foresight*. Unlike prior approaches that treat future latent states solely as prediction targets, DriveFuture regards them as structured conditions for shaping the current decision-making process.

Our framework follows a simple future-aware latent world modeling pipeline. During training, DriveFuture first predicts future latent world states from the current latent state and ego action, and then uses the ground-truth (GT) future latent world state as a grounding signal for the predicted future latent, which then conditions the diffusion-based trajectory planner. This process enables the planner to leverage future semantics for more informed trajectory generation. During inference, GT future latent states are no longer available, and the model instead uses predicted future latent states as conditions, enabling closed-loop planning without future annotations. DriveFuture establishes a unified future-aware latent dynamics mechanism for both training and inference. We evaluate DriveFuture on the public *NAVSIM* benchmarks [1] and observe substantial performance gains. Specifically, DriveFuture achieves **55.5**, **89.9**, and **90.7** EPDMS on *NAVSIM-v2 navhard*, *NAVSIM-v2 navtest*, and *NAVSIM-v1 navtest*, respectively. These results show that future-aware latent world modeling can significantly improve planning quality in challenging closed-loop driving scenarios. In summary, the main contributions of this work are as follows:

- We identify a key limitation of existing latent world models for autonomous driving: they mainly focus on future-state simulation, while failing to fully exploit future states as conditions for current decision-making, which leads to current–future feature entanglement.
- We propose **DriveFuture**, a future-aware latent world modeling framework that conditions the current decision-making process on future world states to learn planning-oriented *foresight*. It adopts a unified training–inference paradigm, using GT future latent states as conditions during training and predicted future latent states during inference.
- We achieve SOTA results on the public NAVSIM benchmarks [1], demonstrating the effectiveness of future-conditioned latent world modeling. Notably, as of April 2026, DriveFuture ranks **1st** on the NAVSIM-v2 *navhard* leaderboard with **55.5** EPDMS, and achieves SOTA performance on NAVSIM-v1 *navtest* with **90.7** PDMS.

2 Related Work

2.1 World Models for Autonomous Driving

Recent research on world modeling for autonomous driving can be broadly grouped into several directions. The first line of work explicitly models future scene evolution in the observation space, including multi-view driving video generation and world simulation methods [24–32], which emphasize high-fidelity future rollout, controllable scene evolution, long-horizon generation, and reliable simulation for downstream planning. The second line models the evolution of 3D/4D world states in occupancy space [33–36], which characterize future worlds from the perspectives of 3D occupancy world modeling, vision-centric 4D occupancy forecasting, 4D occupancy generation, and unified occupancy-language-action modeling, respectively. The third line performs world modeling in more compact BEV or structured state spaces [37, 38], which support future evaluation and trajectory selection through a unified BEV latent space or a BEV world model. More recently, world models have also been increasingly integrated with planner- or VLA-oriented frameworks, as exemplified by LAW [17], World4Drive [18], WorldRFT [19], DriveWorld-VLA [20], DriveVLA-W0 [21], and DriveLaW [22], suggesting a clear trend toward tighter coupling between world modeling, trajectory planning, and scalable autonomous driving systems. Although these methods have demonstrated strong potential for future scene modeling and planning support, they typically involve high modeling complexity and are prone to error accumulation in observation-space or dense-space reconstruction.

For long-horizon planning, GraphWorld [39] combines ego-centric interaction graphs with world-state-conditioned planning, while MomWorld [40] jointly models future latent configuration and momentum and refines trajectories through momentum-conditioned flow matching.

2.2 End-to-End Autonomous Driving.

End-to-end autonomous driving (E2E-AD) methods map raw sensor observations directly to vehicle controls or planned trajectories. Early E2E-AD methods [41, 7, 42, 15, 43, 44, 9, 45, 13, 12, 46, 47, 16, 48, 49], mainly focus on improving scene representation, multi-modal fusion, and planning stability within unified end-to-end frameworks. MomAD [13] promotes temporal consistency through trajectory matching and historical query interactions, while MomADv2 [50] combines selective state-space memory with flow-matching residual refinement. FocalAD [12] emphasizes nearby

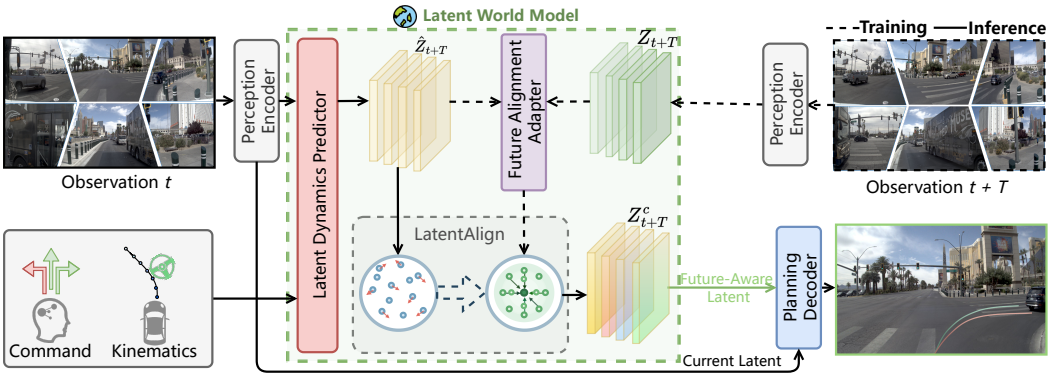


Figure 2: **Overview of DriveFuture.** Multi-view observations at time t are encoded by a shared Perception Encoder into a scene latent $\mathbf{Z}_t$. The *Latent Dynamics Predictor* conditions on $\mathbf{Z}_t$ and a tokenised trajectory intent to produce a predicted future latent $\hat{\mathbf{Z}}_{t+T}$. During training, the future observation at $t+T$ is encoded by the same Perception Encoder into $\mathbf{Z}_{t+T}$, the *Future Alignment Adapter* grounds $\hat{\mathbf{Z}}_{t+T}$ via cross-attention against $\mathbf{Z}_{t+T}$, yielding the future-aware latent $\mathbf{Z}^c_{t+T}$. *LatentAlign* anneals the planning condition from $\mathbf{Z}^c_{t+T}$ towards $\hat{\mathbf{Z}}_{t+T}$ over training, closing the train–inference gap. At inference, the adapter is bypassed, the *Planning Decoder* consumes $\mathbf{Z}_t$ and $\hat{\mathbf{Z}}_{t+T}$ as dual conditioning contexts to iteratively denoise the ego trajectory.

agents through ego-centric graph interactions and locally weighted motion supervision. More recent approaches increasingly introduce generative planning mechanisms to better capture multi-modal futures and long-horizon behaviors. Representative examples include DiffusionDrive [8], DIVER [51], GoalFlow [10], GuideFlow [11], and GTRS [52], which improve trajectory diversity, scoring, controllability, or reasoning ability through diffusion, flow matching, reinforcement learning, or stronger trajectory evaluation. These advances suggest that strong E2E-AD performance increasingly depends on reasoning over possible future evolutions rather than solely reacting to the current scene. However, in most existing systems, future information is introduced only at the output level, e.g., by generating multiple candidate trajectories or scoring future outcomes after the current representation has already been formed. As a result, they improve trajectory generation or selection, but do not fundamentally reshape how the current latent state itself is learned for planning. Complementary to planning, GraphBEV++ [53] improves multi-modal BEV representations through neighborhood-aware local alignment and global alignment tailored to LSS-based and query-based features. DriveFuture complements these advances by leveraging future latent states as direct conditions for trajectory planning, explicitly aligning the decision-making process with downstream objectives.

3 Method

The central principle of DriveFuture is to leverage predicted future latents to condition and enhance trajectory planning. As shown in Fig. 2, our framework is built upon three core modules: the Latent Dynamics Predictor (Sec. 3.1), the Future Alignment Adapter (Sec. 3.2), and the Planning Decoder (Sec. 3.3). We further detail the Training (Sec. 3.4) and Inference (Sec. 3.5) pipelines.

3.1 Latent Dynamics Predictor

To provide the planning decoder with a compact, action-controllable future signal while avoiding the computational overhead of observation-space prediction, we instantiate a trajectory-conditioned latent predictor in a shared BEV latent space.

Trajectory-Conditioned Latent Prediction. Let ϕ_{enc} denote the shared BEV encoder, which maps the multi-view observation $\mathbf{I}_t$ and ego status $\mathbf{s}_t$ into a scene latent $\mathbf{Z}_t$ comprising N BEV tokens and one ego status token. Let ϕ_τ encode an absolute trajectory $\boldsymbol{\tau} = \{(x_k, y_k, \theta_k)\}_{k=1}^{T}$ into a token sequence $\mathbf{E}_\tau = \phi_\tau(\boldsymbol{\tau})$ via a normalised differential representation $(\Delta x, \Delta y, \sin\theta, \cos\theta)$, a linear projection, and a temporal positional embedding. A learnable future query bank $\mathbf{Q}_f \in \mathbb{R}^{K\times d}$ attends

to the joint context through a stack of transformer decoder layers f_ψ, producing

$$\hat{\mathbf{Z}}_{t+T} \;=\; f_\psi(\mathbf{Q}_f\,;\,[\mathbf{Z}_t \,\|\, \mathbf{E}_\tau])\,. \tag{1}$$

We refer to $\hat{\mathbf{Z}}_{t+T}$ as the *trajectory-conditioned latent prediction*: it describes the scene latent at horizon T under the hypothesis that the ego enacts $\boldsymbol{\tau}$, and serves as the substrate on which the Future Alignment Adapter and the Planning Decoder operate.

Conditioning Source Randomisation. The predictor in Eq. (1) requires a trajectory $\boldsymbol{\tau}$ that is unavailable at inference time, and classifier-free guidance (CFG) additionally demands calibrated outputs under absent or coarse trajectory inputs. Both requirements are addressed jointly by drawing the trajectory token sequence from three sources during training, with probabilities $\{p_{\mathrm{gt}}, p_{\mathrm{kin}}, p_\varnothing\}$:

$$\mathbf{E}_\tau \;\sim\; \begin{cases} \phi_\tau(\boldsymbol{\tau}^{\mathrm{gt}}), & \text{w.p. } p_{\mathrm{gt}}, \\ \phi_\tau(\boldsymbol{\tau}^{\mathrm{kin}}), & \text{w.p. } p_{\mathrm{kin}}, \\ \mathbf{E}^{\varnothing}, & \text{w.p. } p_\varnothing, \end{cases} \tag{2}$$

where $\mathbf{E}^{\varnothing}$ is a learned null token sequence and $\boldsymbol{\tau}^{\mathrm{kin}}$ is a coarse but stable kinematic surrogate produced by constant-acceleration extrapolation from the ego velocity (v_x, v_y) and acceleration (a_x, a_y):

$$x_k = v_x t_k + \tfrac{1}{2} a_x t_k^2, \quad y_k = v_y t_k + \tfrac{1}{2} a_y t_k^2, \quad \theta_k = \mathrm{atan2}(v_y + a_y t_k,\, v_x + a_x t_k)\,, \tag{3}$$

with $t_k = k\Delta t$. The three branches expose the predictor to fine, coarse, and unconditional regimes during training, so that the Tweedie-based and kinematic-based guidance terms used in Sec. 3.5 share a well-trained baseline.

3.2 Future Alignment Adapter

The trajectory diffusion loss reaches the latent dynamics predictor only through the planning decoder, which is a weak and indirect signal under which $\hat{\mathbf{Z}}_{t+T}$ is free to drift from the true future semantics in the early stages of training. The Future Alignment Adapter introduces a grounded but lightweight signal that anchors the predicted future latent to the actual future scene, while avoiding dense reconstruction targets that would tie the latent to appearance details unrelated to planning.

Cross-Attention Grounding. During training, the future-step observation $\mathbf{I}_{t+T}$ is passed through the same encoder ϕ_{enc} used for the current frame, with gradients stopped to prevent the future branch from acting as a shortcut, yielding $\mathbf{Z}_{t+T} = \mathrm{sg}(\phi_{\mathrm{enc}}(\mathbf{I}_{t+T}))$. We use $\mathbf{Z}_{t+T}$ as the empirical proxy for the GT future latent state. The adapter is a single multi-head cross-attention layer with $\hat{\mathbf{Z}}_{t+T}$ as queries and $\mathbf{Z}_{t+T}$ as keys/values:

$$\tilde{\mathbf{Z}}_{t+T} \;=\; \mathrm{MHA}\Big(\,\mathrm{LN}(\hat{\mathbf{Z}}_{t+T}),\;\; \mathbf{Z}_{t+T},\;\; \mathbf{Z}_{t+T}\,\Big)\,. \tag{4}$$

The query side preserves the K-token interface used by the Planning Decoder, so the adapter never enlarges the conditioning footprint regardless of the spatial resolution of the future BEV. Because the future observation is unavailable for a subset of the training samples (e.g., at sequence boundaries), let $\mathcal{S}_+$ denote the indices that admit a future frame, the planning condition is

$$\mathbf{Z}^{c,(i)}_{t+T} \;=\; \begin{cases} \tilde{\mathbf{Z}}^{(i)}_{t+T}, & i \in \mathcal{S}_+, \\ \hat{\mathbf{Z}}^{(i)}_{t+T}, & i \notin \mathcal{S}_+, \end{cases} \tag{5}$$

which falls back to the unrefined forecast on samples without future observations. At inference time, $\mathbf{Z}_{t+T}$ is unavailable and the adapter is bypassed, so $\mathbf{Z}^c_{t+T} = \hat{\mathbf{Z}}_{t+T}$.

LatentAlign. To bridge the train–inference gap caused by the unavailability of the grounded future latent $\mathbf{Z}^c_{t+T}$ at inference time, we propose LatentAlign, a sigmoid-based annealing schedule that smoothly transitions the planning condition from the grounded future latent to the self-predicted forecast over the course of training:

$$\alpha(e) \;=\; 1 \;-\; \sigma(\beta\,(e - e_0))\,, \qquad \tilde{\mathbf{Z}}^c_{t+T} \;=\; \alpha(e)\,\mathbf{Z}^c_{t+T} \;+\; \big(1 - \alpha(e)\big)\,\hat{\mathbf{Z}}_{t+T}, \tag{6}$$

where $\sigma(\cdot)$ is the logistic function. Early in training ($\alpha \to 1$), the decoder receives grounded future semantics for stable learning, while late in training ($\alpha \to 0$), only the self-predicted forecast is supplied, thereby matching the inference regime.

3.3 Planning Decoder

Trajectory planning under closed-loop driving is inherently multi-modal: a deterministic regression head averages distinct intents into a single feasible-but-uninformative output. We therefore realise the planner as a conditional diffusion model so that multi-modal trajectory distributions emerge naturally, and route the future condition into every denoising step so that future semantics directly shape action generation rather than entering only as a post-hoc score.

Future-Conditioned Diffusion Transformer. The planning action is parameterised in the differential trajectory space $\mathbf{a} = (\Delta x, \Delta y, \sin\theta, \cos\theta) \in \mathbb{R}^{T\times 4}$. With a denoising diffusion probabilistic model (DDPM) forward process,

$$\mathbf{a}_s \;=\; \sqrt{\bar{\alpha}_s}\,\mathbf{a}_0 \;+\; \sqrt{1-\bar{\alpha}_s}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}), \tag{7}$$

the noise predictor ϵ_θ is a transformer in the diffusion-transformer (DiT) family that consumes a noisy action token, a timestep embedding $\mathbf{e}_s$, and two conditioning contexts, namely the scene context $\mathbf{C}_{\text{scene}} = [\mathbf{e}_s \,\|\, \mathbf{Z}_t]$ and the future context $\mathbf{Z}^c_{t+T}$:

$$\hat{\boldsymbol{\epsilon}} \;=\; \epsilon_\theta\big(\mathbf{a}_s,\; s,\; \mathbf{C}_{\text{scene}},\; \mathbf{Z}^c_{t+T}\big)\,. \tag{8}$$

Each block first cross-attends to $\mathbf{C}_{\text{scene}}$ and then to $\mathbf{Z}^c_{t+T}$, so that environmental geometry is consumed before the future-semantic correction is applied. To preserve a stable warm-start from a planner pretrained without future conditioning, the output projection of the future cross-attention is initialised to zero, making the freshly added module behave as an identity at the start of training and gradually learn to exploit $\mathbf{Z}^c_{t+T}$ as optimisation proceeds.

3.4 Training

Trajectory Diffusion Objective. The primary supervision is the standard DDPM noise-prediction loss applied with the future-aware predictor in Eq. (8):

$$\mathcal{L}_{\text{plan}} \;=\; \mathbb{E}_{s,\boldsymbol{\epsilon},\mathbf{a}_0}\left\|\boldsymbol{\epsilon} - \epsilon_\theta\Big(\mathbf{a}_s, s, \mathbf{C}_{\text{scene}}, \tilde{\mathbf{Z}}^c_{t+T}\Big)\right\|_2^2, \tag{9}$$

where $\tilde{\mathbf{Z}}^c_{t+T}$ is the annealed future condition defined in Eq. (6). To prevent the BEV encoder from collapsing into representations that are useful only for trajectory regression, we retain a semantic auxiliary that decodes BEV tokens into a class-wise occupancy map and supervises it against the rasterised semantic ground truth $\mathbf{Y}^{\text{bev}}$,

$$\mathcal{L}_{\text{bev}} \;=\; \text{CE}\Big(\hat{\mathbf{Y}}^{\text{bev}},\, \mathbf{Y}^{\text{bev}}\Big)\,. \tag{10}$$

The total objective is

$$\mathcal{L} \;=\; \lambda_{\text{plan}}\,\mathcal{L}_{\text{plan}} \;+\; \lambda_{\text{bev}}\,\mathcal{L}_{\text{bev}}. \tag{11}$$

3.5 Inference

At inference, the future observation $\mathbf{I}_{t+T}$ is unavailable. The Future Alignment Adapter is therefore bypassed, and the planning condition reduces directly to $\mathbf{Z}^c_{t+T} = \hat{\mathbf{Z}}_{t+T}$. This introduces a circular dependency: computing $\hat{\mathbf{Z}}_{t+T}$ via Eq. (1) requires a trajectory intent $\mathbf{E}_\tau$, which is itself the output being denoised. Progressive Foresight Guidance (PFG) resolves this dependency by supplying two phase-adaptive surrogate trajectory intents throughout the denoising process. The kinematic extrapolation $\boldsymbol{\tau}^{\text{kin}}$ from Eq. (3) is noise-free but geometrically coarse. A self-consistent estimate $\boldsymbol{\tau}^{\text{tw}}$ is recovered from the current noisy sample via Tweedie's formula,

$$\hat{\mathbf{a}}_0^{(s)} \;=\; \frac{\mathbf{a}_s \;-\; \sqrt{1-\bar{\alpha}_s}\,\hat{\boldsymbol{\epsilon}}_\varnothing}{\sqrt{\bar{\alpha}_s}}, \qquad \boldsymbol{\tau}^{\text{tw}} \;=\; \text{cumsum}\Big(\hat{\mathbf{a}}_0^{(s)}\Big)\,, \tag{12}$$

is reliable only once the noise level decreases. Let $r = s/(S-1) \in [0,1]$ index denoising progress. Three classifier-free guidance branches condition the Latent Dynamics Predictor on different trajectory intents:

$$\mathbf{Z}^{c,\varnothing}_{t+T} = f_\psi(\mathbf{Z}_t;\, \mathbf{E}^{\varnothing}), \quad \mathbf{Z}^{c,\text{kin}}_{t+T} = f_\psi(\mathbf{Z}_t;\, \phi_\tau(\boldsymbol{\tau}^{\text{kin}})), \quad \mathbf{Z}^{c,\text{tw}}_{t+T} = f_\psi(\mathbf{Z}_t;\, \phi_\tau(\boldsymbol{\tau}^{\text{tw}})), \tag{13}$$

Table 1: Comparison with SOTA methods on the **NAVSIM-v2 navhard** split [1].

Method	Backbone	Stage	NC↑	DAC↑	DDC↑	TL↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
E2E-based Methods												
TransFuser [41]	ResNet-34	Stage 1	96.2	79.5	99.1	99.5	84.1	95.1	94.2	97.5	79.1	23.1
		Stage 2	77.7	70.2	84.2	98.0	85.1	75.6	45.4	95.7	75.9	
DiffusionDrive [8]	ResNet-34	Stage 1	96.0	79.7	97.4	99.5	81.3	93.1	90.8	96.8	73.8	24.2
		Stage 2	82.1	72.2	88.5	98.7	85.1	78.8	49.2	89.3	71.2	
GuideFlow [11]	ResNet-34	Stage 1	96.6	80.5	96.3	99.3	82.3	94.9	91.5	97.7	67.8	27.1
		Stage 2	87.3	76.7	88.8	99.2	84.3	85.1	49.7	93.1	44.5	
Senna-E2E [57]	ResNet-50	Stage 1	95.6	86.0	98.9	99.6	83.9	95.1	95.3	97.6	75.6	27.2
		Stage 2	78.6	74.8	84.8	98.2	88.2	75.7	46.9	96.0	65.8	
DriveSuprim [58]	V2-99	Stage 1	98.9	95.1	99.2	99.6	76.1	99.1	94.7	97.6	54.2	42.1
		Stage 2	87.9	88.8	89.6	98.8	80.3	86.0	53.5	97.1	56.1	
ZTRS [59]	V2-99	Stage 1	98.9	97.6	100.0	100.0	66.7	98.9	96.2	96.7	44.0	48.1
		Stage 2	91.1	90.4	95.8	99.0	63.6	89.8	60.4	97.6	66.1	
GTRS-E [52]	V2-99+EVA-ViT-L+ViT-L	Stage 1	98.9	99.3	99.8	99.8	75.2	98.4	96.0	97.6	51.6	49.4
		Stage 2	92.3	93.3	94.6	99.2	73.1	91.2	53.9	96.7	56.8	
SimScale [60]	V2-99	Stage 1	99.6	99.1	99.9	100.0	69.6	99.6	95.8	95.6	28.4	53.2
		Stage 2	94.5	94.2	95.8	99.2	75.8	92.8	60.1	96.1	43.2	
DrivoR [61]	ViT-S	Stage 1	99.1	98.2	99.3	99.8	75.4	98.7	94.9	97.6	70.2	54.6
		Stage 2	92.3	91.6	97.3	99.1	75.7	90.6	56.1	98.4	44.7	
VLA-based Methods												
SpanVLA [62]	Qwen2.5-VL-3B	Stage 1	98.4	94.3	97.8	99.9	85.7	97.2	94.2	97.6	72.1	40.1
		Stage 2	86.9	84.3	87.1	98.2	85.5	82.7	62.3	96.8	67.4	
DiffVLA [63]	V2-99 + ViT-L/14	Stage 1	95.7	99.2	100.0	100.0	85.9	96.4	97.1	95.0	84.2	45.0
		Stage 2	81.2	88.8	94.6	99.0	86.0	76.4	59.8	98.6	80.4	
World-Model-based Methods												
MindDrive [14]	ResNet-34	Stage 1	96.1	86.0	98.8	99.3	83.3	95.6	94.4	97.6	74.7	30.9
		Stage 2	82.6	79.1	86.4	98.0	85.3	79.4	49.2	96.5	71.0	
World4Drive [18]	ResNet-34	Stage 1	97.3	89.1	97.6	99.7	60.5	96.8	87.7	93.1	60.0	34.9
		Stage 2	91.4	82.0	91.0	98.5	53.1	90.6	52.3	93.3	62.8	
DriveFuture	V2-99	Stage 1	99.8	99.8	100	99.6	85.7	99.8	98.7	97.6	66.2	**55.5**
		Stage 2	90.6	87.5	94.1	99.1	84.6	88.8	58.3	93.5	45.6	

yielding three noise predictions $\hat{\boldsymbol{\epsilon}}_{\varnothing}, \hat{\boldsymbol{\epsilon}}_{\mathrm{kin}}, \hat{\boldsymbol{\epsilon}}_{\mathrm{tw}}$ at each step. The guided noise estimate is a phase-dependent mixture

$$\hat{\boldsymbol{\epsilon}} \;=\; \hat{\boldsymbol{\epsilon}}_{\varnothing} \;+\; w_{\mathrm{kin}}(r)(\hat{\boldsymbol{\epsilon}}_{\mathrm{kin}} - \hat{\boldsymbol{\epsilon}}_{\varnothing}) \;+\; w_{\mathrm{tw}}(r)(\hat{\boldsymbol{\epsilon}}_{\mathrm{tw}} - \hat{\boldsymbol{\epsilon}}_{\varnothing})\,, \tag{14}$$

with cosine schedules governed by a shared envelope template

$$w(r;\rho,\nu,w^{\max}) \;=\; w^{\max}\cos\!\Big(\tfrac{\pi r}{2\rho}\Big)\mathbf{1}_{[r<\rho]} \;+\; \tfrac{w^{\max}}{2}\Big[1-\cos\!\Big(\tfrac{\pi(r-\nu)}{1-\nu}\Big)\Big]\mathbf{1}_{[r\geq\nu]}, \tag{15}$$

where w_{kin} uses the decay form and w_{tw} uses the rise form. The two curves overlap on $(r_{\mathrm{start}}, r_{\mathrm{fade}})$, producing a smooth handover from the inertial prior to the self-consistent surrogate as the trajectory crystallises. Combined with the LatentAlign annealing of Sec. 3.4, the planning condition remains a future-aware latent at both training and inference, anchored to real future evidence in training and to phase-adaptive self-consistent surrogates at inference.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate DriveFuture on the public *NAVSIM* benchmark [1], which is built on OpenScene [54] and nuPlan [55] logs for lightweight planning evaluation. We report results on *NAVSIM-v1 navtest* [1], *NAVSIM-v2 navtest* [56], and *NAVSIM-v2 navhard* [56]. *navtest* measures general planning performance, while *navhard* emphasizes safety-critical and long-tail scenarios. Following the official protocol, we use PDMS for *NAVSIM-v1* and EPDMS for *NAVSIM-v2*. EPDMS extends PDMS with additional rule- and comfort-related metrics, including driving-direction compliance, traffic-light compliance, lane keeping, and extended comfort. For *NAVSIM-v2 navhard*, we adopt the official two-stage evaluation, where Stage 2 re-evaluates the planner with synthesized future observations around the Stage-1 endpoint. All results are reported as percentages unless otherwise specified.

4.2 Implementation Details

Table 2: Comparison with SOTA methods on the **NAVSIM-v2 navtest** split [1]. EPDMS* denotes results computed with the original NAVSIM-v2 evaluation code before the human-behavior filtering fix, while EPDMS denotes the corrected official implementation.

Method	NC↑	DAC↑	DDC↑	TL↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS* ↑	EPDMS↑
E2E-based Methods											
TransFuser [41]	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7	–
DiffusionDrive [8]	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	–	84.5
Hydra-MDP++ [64]	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4	–
DriveSuprim [58]	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1	–
ARTEMIS [65]	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	98.3	83.1	–
DiffusionDriveV2 [66]	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	85.5	87.5
VLA-based Methods											
DriveWorld-VLA [20]	98.6	99.1	99.6	99.8	87.4	97.9	97.0	97.8	78.6	–	86.8
DriveVLA-W0 [21]	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	–	86.1
Recogdrive [16]	98.3	95.2	98.3	99.8	87.1	97.5	96.6	99.5	86.5	–	83.6
World-Model-based Methods											
Latent-WAM [67]	98.1	97.3	99.6	99.8	87.7	97.3	97.6	98.1	87.3	–	89.3
DriveFuture	**98.8**	**99.1**	**99.6**	**99.9**	86.6	**98.4**	96.4	98.3	74.8	**86.4**	**89.9**

Table 3: Comparison with SOTA methods on the **NAVSIM-v1 navtest** split [1].

Method	NC↑	DAC↑	TTC↑	Conf.↑	EP↑	PDMS↑
Human	100	100	100	99.9	87.5	94.8
Constant Velocity	69.9	58.8	49.3	100	49.3	21.6
E2E-based Methods						
VADv2 [15]	97.2	89.1	91.6	100	76.0	80.9
TransFuser [41]	97.7	92.8	92.8	100	79.2	84.0
UniAD [7]	97.8	91.9	92.9	100	78.8	83.4
PARA-Drive [68]	97.9	92.4	93.0	99.8	79.3	84.0
DRAMA [69]	98.0	93.1	94.8	100	80.1	85.5
GoalFlow [10]	98.3	93.8	94.3	100	79.8	85.7
Hydra-MDP [45]	98.3	96.0	94.6	100	78.7	86.5
ARTEMIS [65]	98.3	95.1	94.3	100	81.4	87.0
DiffusionDrive [8]	98.2	96.2	94.7	100	82.2	88.1
DIVER [51]	98.5	96.5	94.9	100	82.6	88.3
DriveSuprim [58]	97.8	97.3	93.6	100	86.7	89.9
GoalFlow [10]	98.4	98.3	94.6	100	85.0	90.3
VLA-based Methods						
AutoVLA [70]	98.4	95.6	98.0	99.9	81.9	89.1
Recogdrive [16]	98.2	97.8	95.2	99.8	83.5	89.6
DriveVLA-W0 [21]	98.7	99.1	95.3	99.3	83.3	90.2
DriveWorld-VLA [20]	99.1	98.2	96.1	100	85.9	91.3
World-Model-based Methods						
LAW [17]	96.4	95.4	88.7	99.9	81.7	84.6
World4Drive [18]	97.4	94.3	92.8	100	79.9	85.1
WoTE [38]	98.5	96.8	94.9	99.9	81.9	88.3
WorldRFT [19]	97.8	96.8	94.0	100	81.7	87.8
DriveLaW [22]	99.0	97.1	96.7	100	81.3	89.1
DriveFuture	98.8	**99.1**	95.4	**100**	**84.2**	**90.7**

DriveFuture is built on a TransFuser-based [71] planning pipeline, trained on the NavTrain split for 100 epochs with Adam at a learning rate of 10^{-4}, and uses a GTRS-Dense scorer [52] to select multi-modal trajectories. We use two temporal frames, front and rear camera streams, image resolution 2048×512, and a BEV token grid with 16×64 anchors. The planner predicts an 8-step trajectory over a 4 second horizon with a 0.5 second interval. The diffusion planning head contains 5 transformer layers and samples 100 trajectory proposals by default. Training is conducted on 8 NVIDIA 5090 GPUs with mixed bf16 precision, gradient clipping of 1.0, and loss weights $\lambda_{\text{plan}} = \lambda_{\text{bev}} = 10$. More implementation details are provided in the Supplementary Material.

4.3 Main Results

NAVSIM-v2 navhard. As shown in Table 1, DriveFuture achieves the best overall result with **55.5** EPDMS, outperforming DrivoR [61] (54.6 EPDMS), DiffVLA [63] (45.0 EPDMS), and the strongest prior world-model method World4Drive [18] (34.9 EPDMS). Notably, DriveFuture demonstrates consistent improvements across compliance and safety-related metrics, including 99.1 DAC and 95.4 TTC. These results indicate that conditioning the current latent state on future world states not only improves overall performance but also enhances safety-critical behavior under the challenging two-stage evaluation.

NAVSIM-v2 navtest. As shown in Table 2, DriveFuture achieves the best overall result of **89.9** corrected EPDMS. It surpasses the E2E-AD method DiffusionDriveV2 [66] with 87.5, the VLA method DriveWorld-VLA [20] with 86.8, and the world-model method Latent-WAM [67] with 89.3. These gains indicate that DriveFuture improves planning robustness under the stricter NAVSIM-v2 protocol by explicitly conditioning current representations on future latent states.

NAVSIM-v1 navtest. As shown in Table 3, DriveFuture achieves **90.7** PDMS. It surpasses GoalFlow [10] with 90.3 and DriveLaW [22] with 89.1, while remaining competitive with the best VLA method DriveWorld-VLA [20] at 91.3. This demonstrates that future-conditioned latent modeling better aligns world representations with planning requirements.

Table 4: Unified ablation study on future supervision and PFG heuristic guidance on NAVSIM-v2 navhard [56]. A checkmark denotes the option is enabled in that configuration. FF denotes whether future frames are considered during training, Impl. denotes the proposed implicit future constraint, MSE denotes direct mean-squared-error supervision, KS denotes kinematics-based heuristic guidance, and GT denotes GT trajectory guidance. All ablations are conducted without GTRS-Dense scorer [52].

Future Supervision			Heuristic Guidance		EPDMS	NC	DAC	TTC	LK	HC	EC
FF	Impl	MSE	KS	GT							
			✓	✓	30.9	81.8	72.6	79.7	46.4	95.7	71.7
✓		✓	✓	✓	32.1	81.7	77.0	78.4	49.5	96.7	69.9
✓	✓				32.0	81.1	75.2	78.1	47.6	97.0	75.9
✓	✓		✓	✓	**34.6**	**82.3**	**78.8**	**79.6**	**47.6**	**97.0**	**75.9**

4.4 Ablation Study

Effect of Future Supervision. As shown in Table 4, future-aware conditioning improves robustness, raising EPDMS from 30.9 to 34.6, with clear gains in drivable-area compliance and lane keeping. Direct future-latent MSE supervision is less effective, achieving 32.1 EPDMS, suggesting that future information is better used as an implicit planning condition than as a feature-level regression target.

Effect of Heuristic Guidance . Table 4 evaluates the training sources used by PFG. When the kinematic branch and GT-guided training are removed, the model still produces reasonable proposals, but the final EPDMS drops from 34.6 to 32.0 and the largest deficits appear in Stage-2 DAC and TTC. This indicates that dual-source guidance is not only an inference heuristic: it also improves the quality of the learned future-conditioned latent during training and leads to more stable behavior in hard interactive rollouts.

Sensitivity of Hyper-parameters. Table 5 shows that moderate hyper-parameter settings are consistently preferred. The best EPDMS is achieved with $t_f = 1.5$, $q_s = 16$, and $e_o = 0.83$ in their respective groups. Smaller or larger values reduce robustness, indicating that effective planning requires a balanced future horizon, query budget, and guidance strength.

Table 5: Sensitivity analysis of three hyper-parameters on NAVSIM-v2 navhard [56]. We report the effect of future horizon t_f, query size q_s, and initial guidance parameter e_o on final EPDMS. The best result in each group is highlighted in bold. Here t_f denotes the t_f-th future frame, q_s denotes the number of queries in the world model, and e_o denotes the inflection point in the annealing schedule. All ablations are conducted without GTRS-Dense scorer [52].

t_f	EPDMS	q_s	EPDMS	e_o	EPDMS
0.5	30.2	4	28.2	0.75	29.2
1.0	31.2	**16**	**34.6**	**0.83**	**34.6**
1.5	**34.6**	64	33.7	0.95	28.9

4.5 Qualitative Analysis

We analyze representative challenging cases from the NAVSIM-v2 *navhard* split, as shown in Fig. 3. Compared with World4Drive[18], DriveFuture performs better in common failure cases, including collision, inefficient, and braking. This shows that conditioning current decision representations on future states is more effective than using future states merely as prediction targets.

5 Conclusion

In this work, we propose **DriveFuture**, a future-aware latent world modeling framework for autonomous driving. Unlike existing methods that primarily simulate future states, DriveFuture uses future world states as explicit conditions to shape current latent representations for planning. It consists of a Latent Dynamics Predictor for compact future latent prediction, and a Future Alignment Adapter for extracting planning-relevant information from GT future observations. DriveFuture achieves SOTA performance on public NAVSIM-v1/v2 benchmarks, validating the effectiveness of future-conditioned latent world modeling for autonomous driving.

Limitation and Future Work. DriveFuture depends on predicted future latents during inference, so inaccurate future prediction may affect planning in uncertain scenarios. It also mainly focuses on near-future conditioning. In the future, we will explore uncertainty-aware and longer-horizon future modeling to improve robustness and generalization.

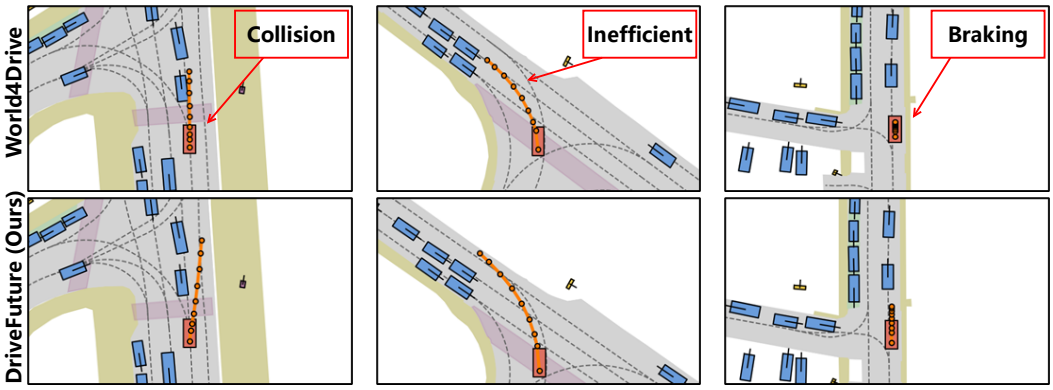


Figure 3: Visual comparison between latent world model World4Drive [18] and our DriveFuture across multiple driving scenarios from NAVSIM-v2 navhard [1].

Acknowledgments and Disclosure of Funding

We declare that no external funding was received for this work. The authors declare no competing financial interests related to this work.

References

[1] 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, 2024. URL `https://arxiv.org/abs/2406.15349`.

[2] Feiyang Jia, Caiyan Jia, Ziying Song, Zhicheng Bao, Lin Liu, Shaoqing Xu, Yan Gong, Lei Yang, Xinyu Zhang, Bin Sun, et al. Progressive robustness-aware world models in autonomous driving: A review and outlook. *Authorea Preprints*, 2025.

[3] Tuo Feng, Wenguan Wang, and Yi Yang. A survey of world models for autonomous driving. *arXiv preprint arXiv:2501.11260*, 2025.

[4] Sifan Tu, Xin Zhou, Dingkang Liang, Xingyu Jiang, Yumeng Zhang, Xiaofan Li, and Xiang Bai. The role of world models in shaping autonomous driving: A comprehensive survey. *arXiv preprint arXiv:2502.10498*, 2025.

[5] Ziying Song, Lin Liu, Feiyang Jia, Yadan Luo, Caiyan Jia, Guoxin Zhang, Lei Yang, and Li Wang. Robustness-aware 3d object detection in autonomous driving: A review and outlook. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):15407–15436, 2024.

[6] Yuechen Luo, Qimao Chen, Fang Li, Shaoqing Xu, Jaxin Liu, Ziying Song, Zhi-xin Yang, and Fuxi Wen. Unleashing vla potentials in autonomous driving via explicit learning from failures. *arXiv preprint arXiv:2603.01063*, 2026.

[7] 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 (CVPR)*, pages 17853–17862, June 2023.

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

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

[10] 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 Computer Vision and Pattern Recognition Conference*, pages 1602–1611, 2025.

[11] Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, JunQiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yandan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. *arXiv preprint arXiv:2511.18729*, 2025.

[12] Bin Sun, Boao Zhang, Jiayi Lu, Xinjie Feng, Jiachen Shang, Rui Cao, Mengchao Zheng, Chuanye Wang, Shichun Yang, Yaoguang Cao, and Ziying Song. FocalAD: Local motion planning for end-to-end autonomous driving. *Automotive Innovation*, 2026. doi: 10.1007/s42154-025-00465-6.

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

[14] Bin Suna, Yaoguang Caob, Yan Wanga, Rui Wanga, Jiachen Shanga, Xiejie Fenga, Jiayi Lu, Jia Shi, Shichun Yang, Xiaoyu Yane, et al. Minddrive: An all-in-one framework bridging world models and vision-language model for end-to-end autonomous driving. *arXiv preprint arXiv:2512.04441*, 2025.

[15] 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.

[16] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025.

[17] 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. *arXiv preprint arXiv:2406.08481*, 2024.

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

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

[20] Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

[21] Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025.

[22] Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. *arXiv preprint arXiv:2512.23421*, 2025.

[23] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15522–15533, 2024.

[24] 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.

[25] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024.

[26] 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. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

[27] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27220–27230, 2025.

[28] 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.

[29] Zhuoran Yang and Yanyong Zhang. Consisdrive: Identity-preserving driving world models for video generation by instance mask. *arXiv preprint arXiv:2602.03213*, 2026.

[30] 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.

[31] Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. Adriver-i: A general world model for autonomous driving. *arXiv preprint arXiv:2311.13549*, 2023.

[32] 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. *arXiv preprint arXiv:2501.13072*, 2025.

[33] 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*, pages 55–72. Springer, 2024.

[34] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9327–9335, 2025.

[35] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024.

[36] Julong Wei, Shanshuai Yuan, Pengfei Li, Qingda Hu, Zhongxue Gan, and Wenchao Ding. Occllama: An occupancy-language-action generative world model for autonomous driving. *arXiv preprint arXiv:2409.03272*, 2024.

[37] Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiaofan Li, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. Bevworld: A multimodal world simulator for autonomous driving via scene-level bev latents. *arXiv preprint arXiv:2407.05679*, 2024.

[38] 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.

[39] Ziying Song, Caiyan Jia, Lin Liu, Lei Yang, Shengkai Zhang, Feiyang Jia, Fengda Zhao, Peiliang Wu, Shaoqing Xu, Chen Lv, and Yadan Luo. GraphWorld: Long-horizon planning with world models for end-to-end autonomous driving. *arXiv preprint arXiv:2606.16274*, 2026.

[40] Ziying Song, Shengkai Zhang, Lei Yang, Haozhuang Chi, Yuchen Liu, Jiangtao Su, Lin Liu, Ziyang Liu, and Chen Lv. MomWorld: Momentum-aware latent world model for long-horizon autonomous driving. *arXiv preprint arXiv:2609.33737*, 2026.

[41] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

[42] 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.

[43] Dongyang Xu, Haokun Li, Qingfan Wang, Ziying Song, Lei Chen, and Hanming Deng. M2da: Multi-modal fusion transformer incorporating driver attention for autonomous driving. *arXiv preprint arXiv:2403.12552*, 2024.

[44] Mingzhe Guo, Zhipeng Zhang, Yuan He, Ke Wang, Liping Jing, and Haibin Ling. End-to-end autonomous driving without costly modularization and 3d manual annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[45] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[46] Lin Liu, Caiyan Jia, Ziying Song, Hongyu Pan, Bencheng Liao, Wenchao Sun, Yongchang Zhang, Lei Yang, Yandan Luo, et al. Fully unified motion planning for end-to-end autonomous driving. *arXiv preprint arXiv:2504.12667*, 2025.

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

[48] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6854–6863, 2025.

[49] Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[50] Ziying Song, Shengkai Zhang, Lin Liu, Peiliang Wu, Lei Yang, Dongyang Xu, Bin Sun, Li Wang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. MomADv2: Reliable temporal memory for end-to-end autonomous driving. *arXiv preprint arXiv:2608.23405*, 2026.

[51] Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *arXiv preprint arXiv:2507.04049*, 2025.

[52] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025.

[53] Ziying Song, Caiyan Jia, Lin Liu, Shaoqing Xu, Lei Yang, and Yadan Luo. GraphBEV++: Multi-modal feature alignment for autonomous driving. *arXiv preprint arXiv:2606.16354*, 2026.

[54] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

[55] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles, 2022. URL https://arxiv.org/abs/2106.11810.

[56] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

[57] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

[58] 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. *arXiv preprint arXiv:2506.06659*, 2025.

[59] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. *arXiv preprint arXiv:2510.24108*, 2025.

[60] H. Tian, T. Li, H. Liu, J. Yang, Y. Qiu, G. Li, J. Wang, Y. Gao, Z. Zhang, et al. Simscale: Learning to drive via real-world simulation at scale. *arXiv preprint arXiv:2511.23369*, 2025.

[61] Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Eloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. *arXiv preprint arXiv:2601.05083*, 2026.

[62] Zewei Zhou, Ruining Yang, Xuewei Qi, Yiluan Guo, Sherry X. Chen, Tao Feng, Kateryna Pistunova, Yishan Shen, et al. Spanvla: Efficient action bridging and learning from negative-recovery samples for vision-language-action model. *arXiv preprint arXiv:2604.19710*, 2026.

[63] Anqing Jiang, Yu Gao, Zhigang Sun, Yiru Wang, Jijun Wang, Jinghao Chai, Qian Cao, Yuweng Heng, Hao Jiang, Zongzheng Zhang, Xianda Guo, Hao Sun, and Hao Zhao. Diffvla: Vision-language guided diffusion planning for autonomous driving. *arXiv preprint arXiv:2505.19381*, 2025.

[64] Kailin Li, Zhenxin Li, Shiyi Lan, Y. Xie, Z. Zhang, J. Liu, Z. Wu, Z. Yu, and Jose M. Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv e-prints*, 2025.

[65] Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, Yanjun Huang, et al. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *arXiv preprint arXiv:2504.19580*, 2025.

[66] Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, et al. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

[67] Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, et al. Latent-wam: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026.

[68] Xinshuo Weng et al. PARA-Drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[69] C. Yuan, Z. Zhang, J. Sun, S. Sun, Z. Huang, C. D. W. Lee, D. Li, Y. Han, A. Wong, K. P. Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. In *International Symposium on Robotics Research*, 2024.

[70] Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

[71] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving, 2022. URL `https://arxiv.org/abs/2205.15997`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 primary focus in both the abstract and the introduction revolves around512the description of the key contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We have discussed the limitations of our method in the conclusion and the Supplementary Material.

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: [Yes]

   Justification: We have provided figures in the main text to substantiate the correctness of the theoretical part.

   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: We have provided all parameter settings and the corresponding experimental results under various parameter conditions in the main text.

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: We will release all code and checkpoints for training and inference upon acceptance.

   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: In the paper, we have clarified that we have adopted a consistent training and testing methodology, referring to mainstream approaches.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We have provided details on the statistical significance of the experiments.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: In the 'Implementation' section, we have specified the computational resources required.

   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: The research conducted in the paper conforms, in every respect, 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: [Yes]

    Justification: In the 'Introduction' and 'Conclusion', we have outlined the positive impact of our work.

    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: [No]

Justification: The submitted paper poses no such risks

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: This paper utilizes open-source datasets, all of which have been properly licensed.

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: We provide additional materials in the Supplementary Material.

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: The paper does not involve crowdsourcing nor 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 paper does not involve crowdsourcing nor research with human subjects.

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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.