# LEARN THE OPERATOR, NOT THE FUTURE: STATE-ANCHORED DISTILLATION OF WORLD ACTION MODEL DYNAMICS FOR AUTONOMOUS DRIVING

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

## ABSTRACT

World action models(WAM) offer end-to-end driving a glimpse of the future, yet directly feeding WAM-predicted features into a planner suffers from feature misalignment, error accumulation in closed-loop rollouts, and high runtime cost. Moreover, generic video WAMs model how scenes evolve but are not explicitly aligned with driving behavior. We propose state-anchored transition transfer, which turns the WAM from a runtime feature provider into a training-time source of transferable transition knowledge. A compact transition operator learns to transform current scene features toward future WAM representations and modulates the planner through a lightweight FiLM-style mechanism, while planning remains anchored to fresh observations. Importantly, the approach requires no future-frame observations during training and relies solely on the pretrained WAM to provide future representations, while the WAM is completely removed at inference. This keeps hallucinated futures off the control path and eliminates the computational cost of WAM deployment. To further align the learned transitions with driving behavior, we use a rule-supervised scorer trained with a metric simulator to guide transition learning. On the NAVSIM-v2 navhard benchmark, our method improves the baseline EPDMS by *13.7%* (from 48.3 to 54.9), and under zero-shot closed-loop evaluation on HUGSIM it further gains *25.4%* in driving score, achieving state-of-the-art performance with negligible inference overhead.

## 1 INTRODUCTION

Driving decisions are, at their core, decisions about the future: whether to commit to a gap, to yield, or to overtake depends not on how the scene looks now, but on how it will evolve over the next few seconds. The value of such foresight is well documented in end-to-end (E2E) driving—adding future modeling to a planner, whether as an auxiliary prediction objective (Li et al., 2025c;d), as anticipated scene representations (Zhang et al., 2025), or as imagined rollouts (Jia et al., 2026), consistently improves planning performance. Meanwhile, large-scale pretrained world action models(WAM) and world action models(WAM) (Wan Team, 2025; Bardes et al., 2025) have become the richest available source of exactly this knowledge: trained on internet-scale video, they internalize how scenes typically unfold at a scale no driving dataset can match, and video-prediction supervision has been shown to amplify the data efficiency of planners (Li et al., 2025d). The challenge, therefore, lies in how to effectively harness such rich future knowledge and translate it into improved driving decisions.

The dominant answer is to materialize the future as features (Figure 1). Three recipes recur. The first learns to predict future scene features and supervises the prediction with features encoded from observed future frames (Li et al., 2025c; Hu et al., 2022; Li et al., 2025d); the predicted future is then either consumed by the planning head (Zhang et al., 2025) or kept as a representation-shaping objective. The second recruits a large pretrained WM or WAM as a future provider, generating future images or latent states from the current observation and feeding them onward to the planner or supervising current features (Wang et al., 2023; Zheng et al., 2025; Jia et al., 2026). The third distills such future generation into an adapter attached to the planner, which produces future features on the fly and passes both current and future features to the downstream decoder (Zhang et al., 2025).



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

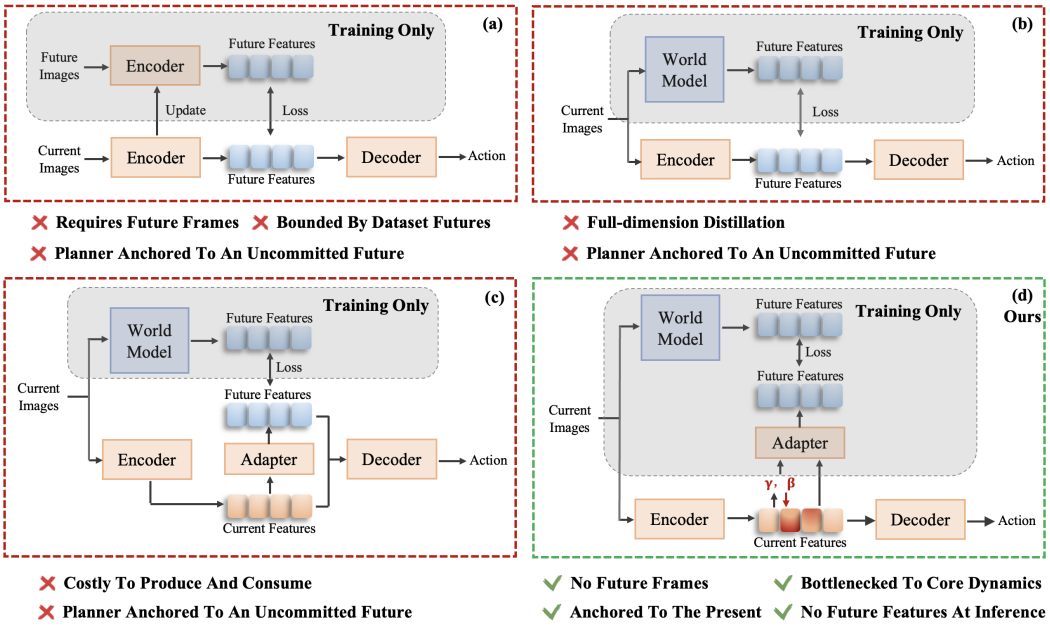


Figure 1: **How the future enters the planner. (a)** Future-frame self-supervision: features encoded from observed future frames supervise future-feature prediction, and the predicted features feed the decoder. **(b)** WAM-supervised prediction: a pretrained world action model replaces the future-frame teacher, active only during training. **(c)** Dual-stream consumption: an adapter produces future features on the fly, and both current and future features feed the decoder. **(d)** Ours (state-anchored transition transfer): a frozen world action model, training only, distills transition factors that modulate the current scene features; the decoder never sees a future-frame feature. Gray shading marks training-only components. $\times$ lists the paradigm defects, ✓ the corresponding properties of ours.

For all their differences, the three recipes rest on the same assumption: the value of the future can only be realized by producing future features and passing them on.

This assumption is the source of structural problems. **(i)** A teacher that replays observed futures caps what can be learned at the dataset's own future distribution, it transfers no dynamics knowledge beyond what the logs already contain, and in practice, requires future-frame data for training. **(ii)** Feeding predicted futures directly to the decoder puts uncalibrated, hallucination-prone content on the control path: predicted features are expressed in a future egocentric frame that the planner has not yet committed to, and no signal certifies how far the prediction, or its frame, has drifted from reality.**(iii)** Future features are expensive to both produce and consume: synthesizing features faithful enough to act on demands heavy generative capacity and iterative refinement, and consuming them inflates the token stream and the load of every downstream module. **(iv)**Generic WAM are blind to driving physics—they capture how scenes plausibly evolve, not how they should evolve under the rules and privileges of driving (Kang et al., 2025).

In this work, we take a different stance: a planner does not need the future itself, but the transition operator that produces it. The future is high-dimensional and error-prone; the change from present to future is low-dimensional, compressible, and cheap to carry. We propose state-anchored transition transfer, which demotes the WAM from a runtime feature provider to a training-time dynamics teacher. A compact transition operator is distilled, under a frozen pretrained world action model, to reproduce the teacher's future representations from current scene features. An information bottleneck of a few low-dimensional tokens forces the operator to keep only the core dynamics of the scene, and conditioning on ego status factors out the redundancy part of scene change induced by self-motion, spending the remaining capacity on how other agents and the scene itself evolve. The operator never materializes a future; instead, it uses a lightweight feature-wise transformation to modulate the current scene tokens, keeping the planner grounded in real observations while allowing the estimated dynamics to refine, rather than replace, their representations. Finally, to cure

the WAM's blindness to driving physics, the planner's own rule-supervised scorer—trained against a metric simulator—injects its driving knowledge back into the learned transitions, aligning them with driving rules rather than video plausibility. After training, the world action model is discarded entirely, and inference carries only a negligible operator.

Our method lifts the EPDMS on the challenging navhard benchmark (Kirby et al., 2026; Cao et al., 2025) from 48.3 to 54.9 without any simulation data, surpassing all published methods to our knowledge. The entire +6.6 gain is obtained at training time, the transition operator adds negligible inference overhead.

Our contributions are summarized as follows:

- **Paradigm.** We identify the structural flaws of materializing the future as planner inputs, and propose state-anchored transition transfer: the world action model serves as a training-time dynamics teacher and is discarded at deployment, leaving zero world-model cost at inference and no hallucinated future on the control path.
- **WAM-guided scene modulation.** An ego-motion-factorized, information-bottlenecked operator learns dynamics from a frozen video world action model and modulates the observed scene features for planning.
- **Completing WAM knowledge for driving.** We feed planning-relevant signals from a metric-simulator-supervised scorer back into the transition operator, complementing world-model-derived dynamics with driving knowledge.
- **State-of-the-art performance.** We achieve state-of-the-art results across multiple driving benchmarks, including navhard, with ablations validating each component.

## 2 Related Work

**End-to-end planning.** End-to-end planners map raw sensor observations directly to future trajectories or controls, replacing hand-crafted modular pipelines with a single differentiable network. UniAD (Hu et al., 2023b) demonstrated planning-oriented joint training over intermediate perception and prediction tasks, and VAD (Jiang et al., 2023) improved efficiency with vectorized scene representations; later systems simplify the stack further by predicting trajectories directly from sensor streams, through transformer-based sensor fusion (Chitta et al., 2022), diffusion policies (Liao et al., 2025), and iterative proposal refinement (Guo et al., 2025). A complementary line instead scores a large vocabulary of trajectory candidates (Chen et al., 2024), strengthened by rule-distilled sub-scores (Li et al., 2024), generalized vocabulary scoring (Li et al., 2025g), test-time training (Sima et al., 2025), and iterative selection (Yao et al., 2025). DrivoR (Kirby et al., 2026) compresses each camera view into a handful of register tokens and disentangles trajectory generation from scoring. Progress on these methods has been driven by (pseudo-)closed-loop evaluation protocols (Dauner et al., 2024; Cao et al., 2025; Zhou et al., 2024), which reward anticipation of scene evolution rather than single-frame fitting; its scene tokens, however, are purely static and single-frame. We endow them with transform dynamics at no additional inference cost.

**World action models for future features.** One family turns future modeling itself into supervision: the planner predicts future features and matches them against features encoded from *observed* future frames (Li et al., 2025c; Hu et al., 2022; Li et al., 2025d), a teacher that is bounded by the dataset's own future distribution. Another family recruits generative world action models to produce the future outright, spanning video (Hu et al., 2023a; Zhao et al., 2024), occupancy (Zheng et al., 2023; Li et al., 2025a), and LiDAR (Zyrianov et al., 2025). How the produced future reaches the planner varies: Drive-WM (Wang et al., 2023) feeds predicted multi-view images into the planner, latent variants forecast future states to assist decision-making (Zheng et al., 2025), what-if variants sample action-conditioned futures to evaluate candidate trajectories (Jia et al., 2026), and SeerDrive (Zhang et al., 2025) both supervises and consumes predicted future scenes, attending to current and future features separately under iterative world-model–planner interaction. Common to all is that the future is materialized as features on the planner's path—expensive to synthesize, and uncalibrated when consumed. Closest to our stance, Latent-WAM (Wang et al., 2026b) keeps its world action model training-only, yet its external teacher is a geometric foundation model and its dynamics are self-supervised from driving data itself. In contrast, we distill a frozen generative world action

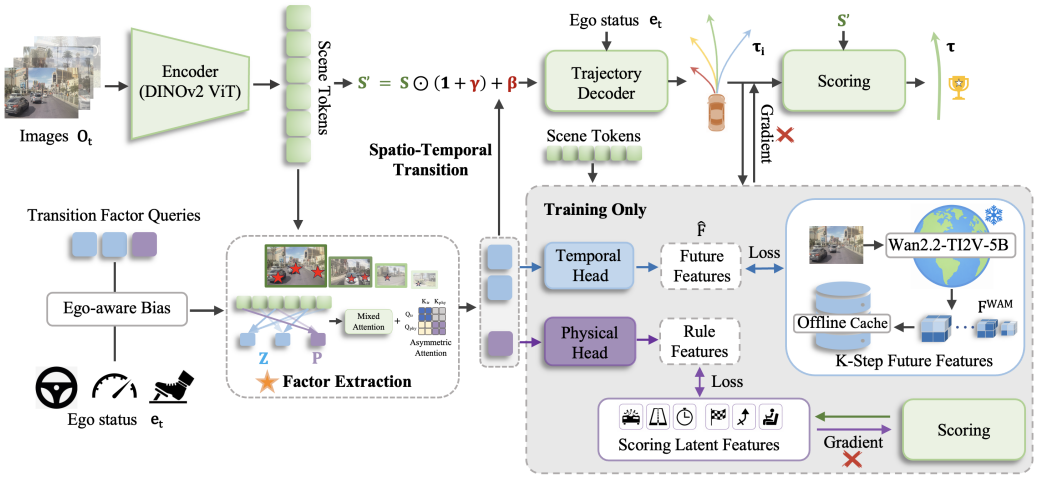


Figure 2: **Overview of state-anchored transition transfer.** During training, a frozen world model supervises future-feature prediction from semantic transition factors, while a rule-supervised scorer provides driving knowledge to physics tokens. At inference, the ego-conditioned factors modulate current scene tokens, which the trajectory decoder and scoring use to select the final trajectory. Components marked "Training Only" are discarded at inference.

model into a transition operator that modulates current features: no future frames are required, no future tokens are consumed, and the world action model is discarded entirely at deployment.

**Rule-supervised planning.** Rule-based metric simulators offer supervision that human demonstrations alone cannot provide, and the field has exploited them at two levels. At the level of scoring objectives, VADv2 (Chen et al., 2024) classifies trajectory candidates by proximity to the human trajectory, Hydra-MDP (Li et al., 2024) introduced multi-target distillation of rule-based sub-scores into a trajectory scorer, and follow-ups scale the vocabulary (Li et al., 2025g) or refine selection at test time (Sima et al., 2025) and through iterative refinement (Yao et al., 2025). At the level of training signals, scorers can be learned from rewards alone without any imitation (Li et al., 2025f), and metric rewards can be distilled on synthetically perturbed scenarios to improve robustness (Tian et al., 2025). In all of these, rule knowledge shapes output heads or training labels. We invert the direction: the scorer's knowledge is injected back into the perception-side transition operator, so that driving physics shapes what the planner sees, not only what it outputs.

# 3 Method

## 3.1 Overview

Given multi-camera observations $O_t$ and the ego status $e_t$, the trajectory decoder produces candidates $\{\tau_i\}$, and the scorer selects the final trajectory $\tau$; each trajectory has 8 poses over a 4-second horizon. Prediction-as-input methods generate future features for the planner to consume at inference time. Our method instead uses a video WAM only during training to distill predictive information into a transition operator. At inference, the cameras yield scene features $S_t$ independently of $e_t$. The operator extracts semantic transition factors $Z_t$ and physics tokens $P_t$ from $S_t$, conditioning both pooling branches on $e_t$; these factors produce modulation parameters $(\gamma_t, \beta_t)$ that transform $S_t$ into $S'_t$. The decoder and scorer use $S'_t$ to generate the candidates and select $\tau$, respectively, without generating predicted future features. Figure 2 illustrates the inference path and the training-only teachers; the operator is defined in subsection 3.2.

We build on the register-token planner of Kirby et al. (2026), which encodes multi-camera observations into scene tokens and uses a decoder and scorer to plan trajectories. Its scene representation comes from the current observation alone. We augment it with a transition operator trained using a frozen WAM and the planner's rule-supervised scorer (subsection 3.2 and subsection 3.3).

### 3.2 WAM-GUIDED SCENE MODULATION

The transition operator summarizes the current scene in two compact groups. A set of $M{=}16$ learnable queries $Q$, augmented with the ego encoding as $Q_e = Q + \phi(e_t)$, pools the $N{=}64$ scene tokens into semantic transition factors; each pooled token then passes through a channel MLP with hidden width $64$. In parallel, four physics tokens $P_t$ are pooled through an independent bottleneck, with their own ego-conditioned queries and projections (subsection 3.3). The two groups interact through one mixed attention, $[Z'_t;\, P'_t] = \mathrm{MixedAttn}(Z_t, P_t)$, and the result modulates the scene,

$$Z_t = \mathrm{MLP}\big(\mathrm{Softmax}(Q_e(S_tW_k)^\top/\sqrt{d})\, S_tW_v\big), \tag{1}$$

$$\gamma_t, \beta_t = \mathrm{CA}(S_t,\, [Z'_t; P'_t]), \quad S'_t = S_t \odot (1+\gamma_t) + \beta_t. \tag{2}$$

The bottlenecks act on different axes: attention pooling reduces the semantic token count from $N{=}64$ to $M{=}16$, while the MLP passes each token through a width-$64$ hidden representation before projecting it back to $d{=}256$.

Part of the apparent future change in a driving scene is caused by the ego vehicle's own motion and is therefore predictable from $e_t$. We add the ego encoding to the pooling queries in both branches so the operator can recognize this predictable component when extracting factors from $S_t$. Under the token and channel bottlenecks, this encourages the semantic factors to avoid spending capacity on ego-induced changes and instead retain dynamics not explained by ego motion, such as the behavior of surrounding agents.

The head that predicts the future is an indirect constraint, not a product. For the factors to be worth carrying, they must be able to reconstruct the future in combination with the current scene; the head instantiates precisely this ability—$K{=}8$ temporal queries, derived from $Z'_t$ and augmented with temporal position encodings, cross-attend to the full scene features and are projected into the teacher's space. The teacher is a frozen Wan2.2-TI2V-5B WAM (Wan Team, 2025); its features are extracted once, offline,

$$F^{\mathrm{WAM}}_{t+k} = \psi\big(\mathrm{WAM}_k(x_t^{\mathrm{front}})\big), \quad k = 1, \ldots, K, \tag{3}$$

where $\mathrm{WAM}_k$ denotes the frozen WAM's output at future step $k$, and $\psi$ extracts the intermediate feature used as the distillation target. We precompute these targets from the current front-view image and align the head's predictions $\hat{F}_{t+k}$ with them using the cosine loss in subsection 3.4; no observed future frames are required.

The modulation in Equation 2 carries the distilled dynamics into planning: each scene token attends to the factors to obtain its own pair $(\gamma_t, \beta_t)$, and the resulting $S'_t$ serves as the key-value input to both the trajectory decoder and the scorer. The planner thus acts on observed scene features informed by predicted evolution, rather than on generated future tokens. The change to each scene token is $S'_t - S_t = \gamma_t \odot S_t + \beta_t$; when $\gamma_t = \beta_t = 0$, the original representation is recovered. Imperfect dynamics can adjust observation-based features but cannot replace them. The predicted features $\hat{F}_{t+k}$ are used only for distillation, and the prediction head is removed at inference.

### 3.3 DRIVING-KNOWLEDGE COMPLETION

The WAM captures plausible scene evolution but lacks explicit supervision on driving rules (Kang et al., 2025). The planner's scorer is trained against a rule-based metric simulator covering collisions, drivable-area compliance, and comfort. We use its representations to complement the WAM's dynamics prior with driving knowledge, transferring that knowledge to the physics tokens through reverse injection.

The two token groups serve different roles: semantic factors capture anticipated scene evolution, whereas physics tokens carry driving-rule knowledge. We connect them asymmetrically so the rule knowledge can use anticipated dynamics without being written back into the semantic factors. In this mixed attention, physics queries attend to both groups, while semantic queries attend only to semantic tokens; an attention mask enforces this one-way flow.

The physics tokens reach planning through the modulation: together with the semantic factors they form the key-value pairs of the FiLM cross-attention in Equation 2, so injected rule knowledge is written into $S'_t$—the very features that both the trajectory decoder and the scorer consume. It is fixed

in the objective by a reverse injection: for each proposal $\tau_i$, its embedding queries the physics tokens through a cross-attention head $h$, and the resulting feature is aligned—by cosine—with the scorer's own trajectory feature $\phi_s(\tau_i)$, detached as the teacher ($\mathcal{L}_{\text{inj}}$ in Equation 5). Rule knowledge thus flows backwards, from the output head where a metric simulator supervises it, into the perception-side factors that planning consumes; in all prior rule-supervised pipelines this knowledge shapes outputs only (section 2).

### 3.4 Training Objective

The full objective combines the planner's original imitation and scoring losses with two distillation terms:

$$\mathcal{L}_{\text{dis}} = \tfrac{1}{K} \sum_{k=1}^{K} \Big(1 - \cos\big(\hat{F}_{t+k},\, F_{t+k}^{\text{WAM}}\big)\Big), \tag{4}$$

$$\mathcal{L}_{\text{inj}} = 1 - \cos\big(h(\tau_i, P'_t),\, \text{sg}[\phi_s(\tau_i)]\big). \tag{5}$$

We optimize their weighted sum, $\mathcal{L} = \mathcal{L}_{\text{traj}} + \mathcal{L}_{\text{score}} + \lambda_d \mathcal{L}_{\text{dis}} + \lambda_p \mathcal{L}_{\text{inj}}$. Here, $h(\tau_i, P'_t)$ is the physics-conditioned feature obtained by using the embedding of proposal $\tau_i$ to query the physics tokens, and $\text{sg}[\cdot]$ stops gradients through the scorer feature $\phi_s(\tau_i)$ in $\mathcal{L}_{\text{inj}}$. The losses $\mathcal{L}_{\text{traj}}$ and $\mathcal{L}_{\text{score}}$ are the winner-takes-all $L_1$ imitation loss and the per-sub-score binary cross-entropy against the metric simulator, respectively, both inherited from the base planner (Kirby et al., 2026). Training never touches a future frame and never loads the world model; at inference, the world model and the distillation head are discarded, and the planner carries only the two pooling branches, the mixed attention, and the FiLM modulation—a negligible addition to the ViT backbon

## 4 Experiments

### 4.1 Experimental Setup

We evaluate on three complementary benchmarks. **NAVSIM-v2 navhard** (Cao et al., 2025) is a two-stage pseudo-closed-loop protocol: stage one scores original real-world scenarios, stage two re-scores scenarios perturbed towards out-of-distribution states, and the aggregate EPDMS spans nine sub-metrics over both stages—making it the benchmark most sensitive to how well a planner anticipates scene evolution. **NAVSIM-v1 navtest** (Dauner et al., 2024) scores open-loop planning on real-world scenarios with the PDMS. **HUGSIM** (Zhou et al., 2024) is a photorealistic closed-loop simulator built on 3D Gaussian-splatting reconstructions of real logs; we evaluate zero-shot, i.e., trained on real data only and never on rendered frames, using Road Completion (RC) and the HUGSIM Driving Score (HD-Score) across four difficulty levels.

Our implementation builds on the DrivoR planner (Kirby et al., 2026): four cameras encoded by a DINOv2 ViT-S with LoRA of rank 32 and 16 register tokens per camera, 64 proposals refined over 4 stages, and $d$=256. The transition operator uses $M$=16 semantic factors with a width-64 channel bottleneck and four physics tokens, with $\lambda_d$=1 and $\lambda_p$=0.5. Teacher features are extracted once from the frozen Wan2.2-TI2V-5B for all training scenes; the WAM is never loaded during training, and training never touches a future frame. We train on navtrain ($\sim$85k scenes) for 20 epochs with a batch size of 16 per GPU on 8 GPUs, using AdamW with learning rate $2\times10^{-4}$ and a warmup-cosine schedule. Training uses only real NAVSIM scenes throughout—no synthetic, rendered, or simulated data is involved at any stage.

### 4.2 Main Results

On navhard (Table 1), our method reaches 54.9 EPDMS, improving the DrivoR base by 13.7%, and it is the only learning-based method in the comparison that surpasses PDM-Closed, which consumes ground-truth symbolic states. Notably, the largest sub-metric gain appears in ego progress (90.7 vs. 72.6 at stage one, 85.3 vs. 70.0 at stage two): anticipating how the scene unfolds lets the planner commit to progress where a purely reactive policy hesitates. This is precisely the behavior the transition operator is trained to enable—and unlike methods that obtain it by feeding predicted futures to the decoder, ours pays nothing at inference.

On navtest (Table 2), our method achieves 93.8 PDMS with camera input only, ahead of every listed planner including its own DrivoR base (93.7), with the strongest competitor at 92.5. The

Table 1: **Performance on the NAVSIM-v2 navhard two-stage benchmark** using the extended predictive driver model score (EPDMS). NC: no at-fault collision. DAC: drivable area compliance. DDC: driving direction compliance. TLC: time-to-lane-crossing. EP: ego progress. TTC: time-to-collision. LK: lane keeping. HC: hard comfort. EC: extended comfort. †: trained on a dataset $10\times$ larger than navtrain. GTRS-A/GTRS-D refer to GTRS-Aug/GTRS-Dense.

| Method | Stage 1 ↑ | | | | | | | | | Stage 2 ↑ | | | | | | | | | EPDMS ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NC | DAC | DDC | TLC | EP | TTC | LK | HC | EC | NC | DAC | DDC | TLC | EP | TTC | LK | HC | EC | |
| PDM-Closed (Dauner et al., 2023) | 94.4 | 98.8 | 100 | 99.5 | 100 | 93.5 | 99.3 | 97.7 | 36.0 | 88.1 | 90.6 | 96.3 | 98.5 | 100 | 83.1 | 73.7 | 91.5 | 25.4 | 51.3 |
| Constant Velocity (Dauner et al., 2024) | 88.8 | 42.8 | 70.6 | 99.3 | 77.5 | 87.3 | 78.6 | 97.1 | 60.4 | 83.2 | 59.1 | 76.5 | 98.0 | 71.3 | 81.1 | 47.9 | 97.1 | 61.9 | 10.9 |
| Ego Status MLP (Dauner et al., 2024) | 93.2 | 55.7 | 86.6 | 99.3 | 81.2 | 92.2 | 83.5 | 97.5 | 77.7 | 77.2 | 51.9 | 74.4 | 98.2 | 77.1 | 75.0 | 40.8 | 97.8 | 79.8 | 12.7 |
| LTF (Chitta et al., 2022) | 96.2 | 79.6 | 99.1 | 99.6 | 84.1 | 95.1 | 94.2 | 97.6 | 79.1 | 77.8 | 70.2 | 84.3 | 98.1 | 85.1 | 75.7 | 45.4 | 95.8 | 76.0 | 23.1 |
| GTRS-D (Li et al., 2025g) | 98.9 | 96.2 | 99.4 | 99.3 | 72.9 | 98.9 | 95.1 | 96.9 | 39.1 | 91.2 | 89.4 | 94.4 | 98.8 | 69.5 | 90.0 | 54.3 | 94.0 | 48.7 | 45.0 |
| GTRS-A (Li et al., 2025g) | 98.9 | 95.1 | 99.1 | 99.6 | 76.2 | 99.1 | 94.9 | 97.6 | 54.2 | 88.1 | 88.8 | 89.3 | 98.9 | 98.9 | 85.9 | 53.7 | 96.8 | 56.9 | 45.4 |
| DriveSuprim (Yao et al., 2025) | 98.7 | 98.0 | 99.1 | 99.8 | 75.9 | 98.7 | 94.7 | 97.6 | 49.8 | 89.5 | 89.6 | 92.9 | 98.5 | 78.9 | 86.4 | 55.3 | 96.5 | 52.7 | 44.7 |
| ZTRS (Li et al., 2025f) | 98.9 | 97.6 | 100 | 100 | 66.7 | 98.9 | 96.2 | 96.7 | 44.0 | 91.1 | 90.4 | 95.8 | 99.0 | 63.6 | 89.8 | 60.4 | 97.6 | 66.1 | 48.1 |
| RAP-DINO† (Feng et al., 2026) | 97.1 | 94.4 | 98.8 | 99.8 | 83.9 | 96.9 | 94.7 | 96.4 | 66.2 | 83.2 | 83.9 | 87.4 | 98.0 | 86.9 | 80.4 | 52.3 | 95.2 | 52.4 | 39.6 |
| R2SE† (Liu et al., 2026a) | 98.8 | 96.2 | 98.7 | 99.8 | 75.7 | 99.5 | 96.0 | 96.2 | 40.8 | 90.8 | 87.1 | 91.8 | 98.6 | 76.4 | 89.7 | 52.3 | 91.5 | 36.4 | 43.8 |
| GuideFlow† (Liu et al., 2026b) | 97.8 | 97.1 | 100 | 100 | 81.4 | 98.5 | 91.4 | 92.8 | 34.2 | 87.3 | 92.3 | 98.0 | 96.9 | 75.8 | 85.5 | 59.3 | 95.4 | 53.5 | 46.7 |
| DrivoR (Kirby et al., 2026) | 98.8 | 95.1 | 98.9 | 100 | 72.6 | 98.7 | 94.0 | 97.6 | 73.3 | 90.2 | 88.4 | 91.9 | 98.6 | 70.0 | 88.0 | 50.1 | 98.5 | 76.2 | 48.3 |
| Gigapixel-DrivoR (Rowe et al., 2026) | 99.4 | 95.8 | 99.4 | 99.8 | 68.1 | 99.6 | 91.8 | 97.6 | 49.8 | 93.7 | 92.9 | 96.0 | 98.9 | 62.6 | 90.7 | 60.2 | 98.2 | 58.8 | 50.1 |
| **Ours** | 99.6 | 96.4 | 99.6 | 100 | 90.7 | 99.1 | 93.1 | 97.6 | 70.2 | 90.5 | 88.4 | 93.6 | 98.5 | 85.3 | 87.9 | 53.3 | 98.9 | 75.9 | **54.9** |

Table 2: **Performance on the NAVSIM-v1 navtest benchmark** using the predictive driver model score (PDMS). NC: no at-fault collision. DAC: drivable area compliance. TTC: time-to-collision. Comf.: comfort. EP: ego progress.

| Method | NC ↑ | DAC ↑ | TTC ↑ | Comf. ↑ | EP ↑ | PDMS ↑ |
|---|---|---|---|---|---|---|
| Human driver (Dauner et al., 2024) | 100 | 100 | 100 | 99.9 | 87.5 | 94.8 |
| PDM-Closed (Dauner et al., 2023) | 94.6 | 99.8 | 89.9 | 86.9 | 99.9 | 89.1 |
| World4Drive (Zheng et al., 2025) | 97.4 | 94.3 | 92.8 | 100 | 79.9 | 85.1 |
| DiffusionDrive (Liao et al., 2025) | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| AutoVLA (Zhou et al., 2025) | 98.4 | 95.6 | 98.0 | 99.9 | 81.9 | 89.1 |
| DriveVLA-W0 (Li et al., 2025d) | 98.7 | 99.1 | 95.3 | 99.3 | 83.3 | 90.2 |
| ReCogDrive (Li et al., 2025e) | 97.9 | 97.3 | 94.9 | 100 | 87.3 | 90.8 |
| Hydra-MDP++ (Li et al., 2025b) | 98.6 | 98.6 | 95.1 | 100 | 85.7 | 91.0 |
| iPad (Guo et al., 2025) | 98.6 | 98.3 | 94.9 | 100 | 88.0 | 91.7 |
| KnowVal-iPad (Xia et al., 2026b) | 99.2 | 98.7 | 95.4 | 100 | 88.1 | 92.9 |
| WorldRFT (Yang et al., 2026) | 97.8 | 96.8 | 94.0 | 100 | 81.7 | 87.8 |
| MeanFuser (Wang et al., 2026a) | 98.6 | 97.0 | 95.0 | 100 | 82.8 | 89.0 |
| LaST-VLA (Luo et al., 2026) | 98.7 | 97.9 | 95.6 | 100 | 86.8 | 91.3 |
| VeteranAD (Zhang et al., 2026) | 99.1 | 98.3 | 96.1 | 100 | 83.1 | 90.2 |
| ResAD (Zheng et al., 2026) | 98.9 | 97.8 | 94.9 | 100 | 87.0 | 90.6 |
| DrivoR (Kirby et al., 2026) | 99.0 | 98.9 | 96.7 | 100 | 90.0 | 93.7 |
| NavDrive (Yin et al., 2026) | 100 | 98.1 | 99.2 | 100 | 85.4 | 92.5 |
| DriveWorld-VLA (Liu et al., 2026c) | 99.1 | 98.2 | 96.1 | 100 | 85.9 | 91.3 |
| DriveLaW (Xia et al., 2026a) | 99.0 | 97.1 | 96.7 | 100 | 81.3 | 89.1 |
| DriveFuture (Hong et al., 2026) | 98.8 | 99.1 | 95.4 | 100 | 84.2 | 90.7 |
| WA-JEPA (Wang et al., 2026c) | 99.5 | 98.3 | 97.7 | 100 | 85.0 | 91.8 |
| **Ours** | 98.8 | 98.8 | 95.8 | 100 | 91.5 | **93.8** |

edge over the base is markedly smaller than on navhard and HUGSIM, and the reason is the same: what a WAM contributes is open-world dynamics—priors about how scenes unfold beyond the training distribution—whereas navtest scores open-loop imitation on scenes drawn from that very distribution, anchored to the logged human trajectory, and its open-loop protocol is known to reflect closed-loop behavior only weakly (Cao et al., 2025). For such in-distribution scenes there is little beyond the present observation for an open-world prior to add; the identity prior of our modulation, in turn, keeps the base planner intact, so the operator yields a small but consistent margin here while reserving its full effect for the regimes that reward anticipation.

On HUGSIM (Table 3), where our NAVSIM-trained planner is evaluated zero-shot under photorealistic closed-loop rollouts, our method obtains 45.5 RC and the best HD-Score of 35.0—a $25.4\%$ gain over the DrivoR base and $21.1\%$ over the strongest competitor Latent-WAM. This setting tests planning under a render-domain shift and repeated closed-loop decisions. The results are consistent with the benefit of keeping planning anchored to current observations; representative rollouts across the four HUGSIM source datasets are shown in Appendix C.

Table 3: **Zero-shot closed-loop evaluation on HUGSIM** (Zhou et al., 2024), using Road Completion (RC) and HUGSIM Driving Score (HD-Score) across difficulty levels (E: easy, M: medium, H: hard, X: extreme). All methods are trained on real driving data only and evaluated zero-shot on 3DGS-rendered scenes.

| Method | RC ↑ | | | | | HD-Score ↑ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | E | M | H | X | **Avg.** | E | M | H | X | **Avg.** |
| UniAD (Hu et al., 2023b) | 58.6 | 41.2 | 40.4 | 26.0 | 40.6 | 48.7 | 29.5 | 27.3 | 14.3 | 28.9 |
| VAD (Jiang et al., 2023) | 38.7 | 27.0 | 25.5 | 23.0 | 27.9 | 24.3 | 9.9 | 10.4 | 8.2 | 12.3 |
| LTF (Chitta et al., 2022) | 68.4 | 40.7 | 36.9 | 25.5 | 41.4 | 52.8 | 24.6 | 19.8 | 8.1 | 24.8 |
| DrivoR (Kirby et al., 2026) | 78.6 | 42.7 | 29.3 | 30.3 | 43.4 | 68.9 | 24.6 | 13.5 | 14.3 | 27.9 |
| GTRS-Dense (Li et al., 2025g) | 64.2 | 50.0 | 20.7 | 22.3 | 38.0 | 55.5 | 39.0 | 11.7 | 14.3 | 28.6 |
| Latent-WAM (Wang et al., 2026b) | 84.2 | 42.5 | 30.6 | 35.5 | 45.9 | 72.5 | 24.0 | 12.2 | 18.1 | 28.9 |
| **Ours** | 79.8 | 56.3 | 27.9 | 20.0 | 45.5 | 72.5 | 44.5 | 17.6 | 9.3 | **35.0** |

### 4.3 Ablation Studies

Table 4: **Ablations on navhard (EPDMS).** (a) Module contributions; (b) semantic factor count $M$. Mask: ✓ one-directional, × bidirectional, — absent. $64/M$ is the scene-to-factor token ratio.

**(a) Module contributions**

| Operator | Ego conditioning | Rule injection | Mask | **EPDMS ↑** |
|---|---|---|---|---|
| × | × | × | — | 48.3 |
| ✓ | × | × | — | 52.3 |
| ✓ | ✓ | × | — | 54.4 |
| ✓ | ✓ | ✓ | × | 54.1 |
| ✓ | ✓ | ✓ | ✓ | **54.9** |

**(b) Semantic bottleneck size**

| $M$ | $64/M$ | **EPDMS ↑** |
|---|---|---|
| 8 | 8.0× | 51.9 |
| 16 | 4.0× | **54.9** |
| 24 | 2.7× | 54.3 |
| 32 | 2.0× | 52.5 |
| 64 | 1.0× | 52.4 |

Table 4(a) isolates the operator, ego conditioning, and rule injection. The distilled operator lifts EPDMS from 48.3 to 52.3, accounting for most of the full $+6.6$ gain. Ego conditioning raises it to 54.4, consistent with ego-aware transition queries, though this score alone does not establish motion disentanglement. Bidirectional rule mixing scores 54.1; the one-directional mask reaches 54.9, supporting the asymmetric flow in subsection 3.3.

Table 4(b) varies only $M$: EPDMS rises from 51.9 at $M$=8 to 54.9 at $M$=16, then falls to 52.4 at $M$=64. This is consistent with a selective bottleneck: too few factors limit transition information, while $M$=64 removes the token-count reduction.

Appendix A compares two further controls that use the same WAM feature targets but pass predicted future tokens directly to the planner, either replacing or concatenating with the current scene tokens.

### 4.4 Qualitative Results

Figure 3 compares DrivoR (Kirby et al., 2026), GTRS-Dense (Li et al., 2025g), ZTRS (Li et al., 2025f), and ours on two navhard scenarios. Each row combines BEV with the left, front, and right camera views. In stage one, the recorded human trajectory provides a reference: ours follows the left turn closely, while DrivoR drifts toward the curb and the other planners choose earlier, conservative branches. In the perturbed stage-two case there is no human reference. The baselines yield shorter, hesitant paths, whereas ours completes the right turn in a smooth arc. This visual contrast is consistent with the ego-progress gains in Table 1.

## 5 Conclusion

We presented state-anchored transition transfer, a paradigm that turns a large generative WAM from a runtime feature provider into a training-time dynamics teacher. A transition operator compresses the scene's future into a handful of bottlenecked factors—factorized against ego motion so that capacity is spent on how the world evolves around the ego—and these factors modulate current

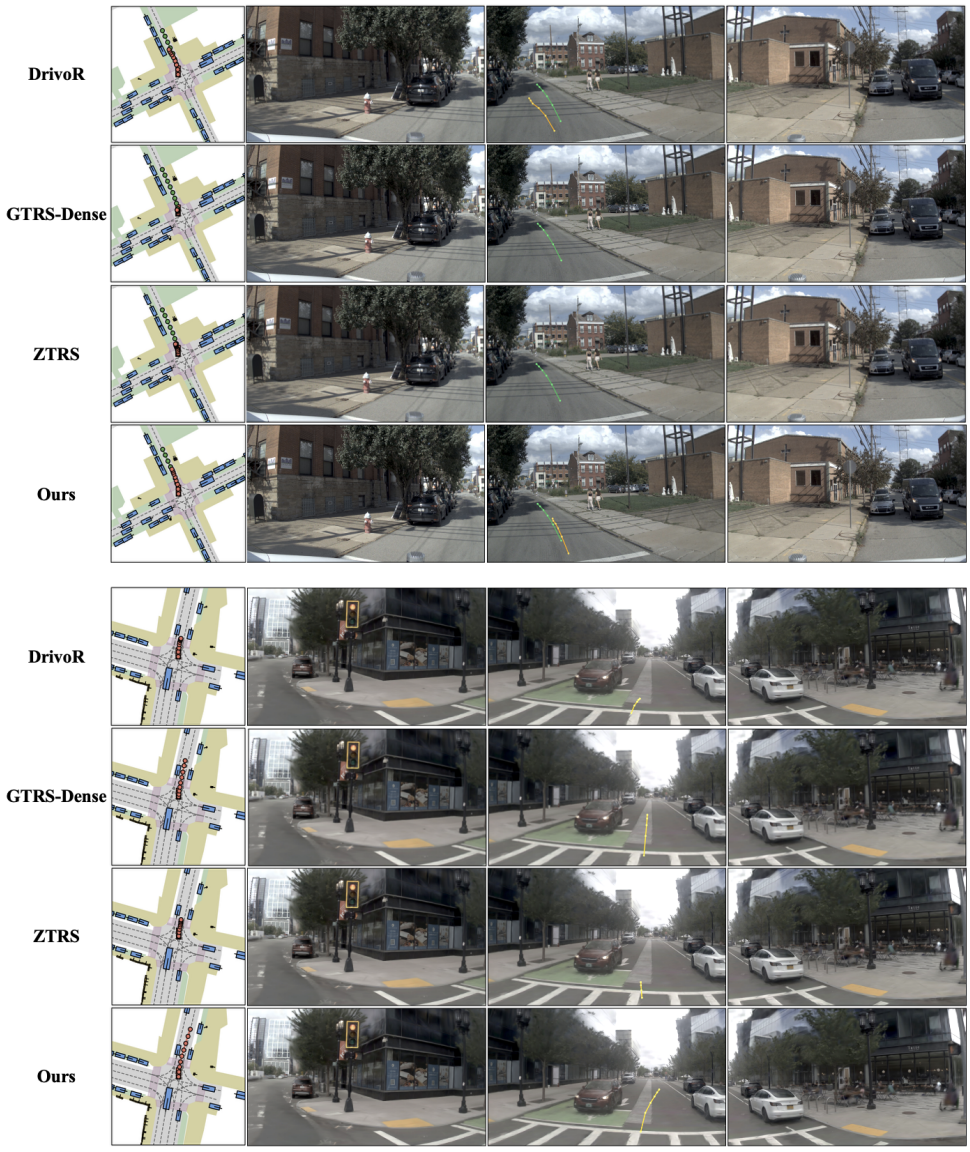


Figure 3: **Navhard qualitative comparison.** Top: stage one (green: human; yellow: prediction). Bottom: perturbed stage two (no reference). Each row: BEV and three views. Methods, top to bottom: DrivoR (Kirby et al., 2026), GTRS-Dense (Li et al., 2025g), ZTRS (Li et al., 2025f), and ours.

scene features instead of materializing futures; a rule-supervised scorer then injects driving physics back into the learned transitions along a strictly one-directional pathway. The resulting planner stays anchored to real observations, trains without a single future frame, and deploys with zero WAM cost, improving navhard EPDMS by 13.7% and closed-loop HUGSIM HD-Score by 25.4% over its base planner.

## AI USE STATEMENT

We used generative AI tools to identify candidate related work. The authors read the original sources and independently verified all cited references.

### Ethics statement

This work uses existing research datasets and does not involve new collection of personal data or human-subject studies. We are not aware of any privacy concerns arising from this work, and the authors declare no conflicts of interest.

### Reproducibility statement

We provide the method, implementation details, and experimental settings in the main paper, with additional ablations in the appendix. We will release the code upon acceptance to facilitate reproducibility.

## References

Adrien Bardes et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

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.

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.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2022.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, 2024.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning, 2026.

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.

Yufeng Hong, Xiaotian Zhou, Yingyan Li, Xiangpo Zhou, Lin Liu, Yadan Luo, Shaoqing Xu, Lei Yang, and Ziying Song. Drivefuture: Future-aware latent world models for autonomous driving. *arXiv preprint arXiv:2605.09701*, 2026.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. In *NeurIPS*, 2022.

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*, 2023a.

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 *CVPR*, 2023b.

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

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 *ICCV*, 2023.

Bingyi Kang, Zhiyuan Xu, Shibo Yi, et al. How far is video generation from world model: A physical law perspective. In *ICML*, 2025.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung VU, and Matthieu Cord. Driving on registers, 2026. URL https://arxiv.org/abs/2601.05083.

Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, Shuchang Zhou, Li Zhang, Xiaojuan Qi, Hao Zhao, Mu Yang, Wenjun Zeng, and Xin Jin. Uniscene: Unified occupancy-centric driving scene generation, 2025a. URL https://arxiv.org/abs/2412.05435.

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, 2025b.

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, 2025c. URL https://arxiv.org/abs/2406.08481.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. Drivevla-w0: World models amplify data scaling law in autonomous driving, 2025d. URL https://arxiv.org/abs/2510.12796.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving, 2025e.

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.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, et al. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. *arXiv preprint arXiv:2510.24108*, 2025f.

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*, 2025g.

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 *CVPR*, pp. 12037–12047, 2025.

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *TPAMI*, 2023.

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*, 48(5): 5774–5792, 2026a. doi: 10.1109/TPAMI.2026.3653866.

Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, Junqiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3719–3728, June 2026b.

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*, 2026c.

Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, et al. Last-vla: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

Luke Rowe, Roger Girgis, Rodrigue de Schaetzen, Daphne Cornelisse, Alaap Grandhi, Felix Heide, Eugene Vinitsky, Christopher Pal, and Liam Paull. Scaling self-play for end-to-end driving, 2026.

Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Tieniu Tan, Long Chen, and Hongyang Li. Simscale: Learning to drive via real-world simulation at scale. *arXiv preprint arXiv:2511.23369*, 2025.

Wan Team. Wan2.2: Scaling intra-model continual pre-training for video generation. *arXiv preprint arXiv:2503.20314*, 2025.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, Zhongpu Xia, and Qichao Zhang. Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17884–17893, June 2026a.

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

Xinlin Wang, Yujiao Xiang, Yuheng Zhou, Jingqi Wang, Minqing Huang, Jiajie Huang, Dongxu Wei, Tingguang Zhou, Xiyang Wang, Gong Chen, et al. Wa-jepa: Rethinking the video jepa paradigm for world-action modeling in autonomous driving. *arXiv preprint arXiv:2608.20974*, 2026c.

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, 2023. URL https://arxiv.org/abs/2311.17918.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026a.

Zhongyu Xia, Wenhao Chen, Yongtao Wang, and Ming-Hsuan Yang. Knowval: A knowledge-augmented and value-guided autonomous driving system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3740–3749, June 2026b.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *ITSC*, 2021.

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 *AAAI*, volume 40, pp. 11649–11657, 2026.

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.

Hongbo Yin, Daxin Tian, Jianshan Zhou, Xuting Duan, Dezong Zhao, and Dongpu Cao. Navdrive: Safety-enhanced end-to-end autonomous driving with navigation-guided diffusion policy. *IEEE Transactions on Intelligent Transportation Systems*, 27(5):5957–5971, 2026. doi: 10.1109/TITS.2026.3650852.

Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *NeurIPS*, 2025.

Bozhou Zhang, Jingyu Li, Nan Song, and Li Zhang. Perception in plan: Coupled perception and planning for end-to-end autonomous driving. In *AAAI*, volume 40, pp. 12376–12384, 2026.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation, 2024. URL https://arxiv.org/abs/2403.06845.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving, 2023. URL https://arxiv.org/abs/2311.16038.

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 *ICCV*, 2025.

Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3729–3739, June 2026.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: a real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv preprint arXiv:2412.01718*, 2024.

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, 2025.

Vlas Zyrianov, Henry Che, Zhijian Liu, and Shenlong Wang. Lidardm: Generative lidar simulation in a generated world, 2025. URL https://arxiv.org/abs/2404.02903.

## A Future-Feature Consumption Controls

To test the future-only and current-plus-future decoder inputs illustrated in Figure 1, we use the same DrivoR backbone and cached WAM future-feature targets to train two direct-future controls. Both use a temporal adapter to predict $64$ tokens at each of $8$ future steps ($512$ tokens in total); the mean feature at each step is aligned with the corresponding WAM target. *Future Replace* uses these predicted tokens in place of the $64$ current scene tokens as input to the decoder and scorer. *Future Concat* appends them to the current scene tokens, giving the decoder and scorer $576$ tokens. In our method, future prediction is only a training constraint: at inference, the decoder and scorer consume the $64$ current scene tokens after transition-based modulation.

Table 5: **Future-feature consumption controls on NAVSIM-v2 navhard.** Future Replace and Future Concat use the same cached WM targets as ours but feed their predicted future tokens directly to the decoder and scorer. EPDMS is the aggregate two-stage score. NC, DAC, DDC, TLC, EP, TTC, LK, HC, and EC follow the definitions in Table 1.

| Method | Stage | NC | DAC | DDC | TLC | EP | TTC | LK | HC | EC | **EPDMS ↑** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | Stage 1 | 98.8 | 95.1 | 98.9 | 100 | 72.6 | 98.7 | 94.0 | 97.6 | 73.3 | 48.3 |
| | Stage 2 | 90.2 | 88.4 | 91.9 | 98.6 | 70.0 | 88.0 | 50.1 | 98.5 | 76.2 | |
| Future Replace | Stage 1 | 98.4 | 92.7 | 98.8 | 99.8 | 90.2 | 98.4 | 91.5 | 97.6 | 67.1 | 51.6 |
| | Stage 2 | 91.7 | 86.0 | 91.5 | 99.2 | 81.5 | 90.4 | 51.2 | 99.0 | 77.2 | |
| Future Concat | Stage 1 | 99.1 | 93.8 | 99.2 | 100 | 90.4 | 99.3 | 92.4 | 97.6 | 71.6 | 51.1 |
| | Stage 2 | 88.6 | 86.2 | 92.3 | 97.9 | 83.6 | 87.1 | 55.3 | 98.3 | 76.3 | |
| **Ours** | Stage 1 | 99.6 | 96.4 | 99.6 | 100 | 90.7 | 99.1 | 93.1 | 97.6 | 70.2 | **54.9** |
| | Stage 2 | 90.5 | 88.4 | 93.6 | 98.5 | 85.3 | 87.9 | 53.3 | 98.9 | 75.9 | |

Table 5 reports the full navhard breakdown. Both direct-future variants improve over the DrivoR base (48.3 EPDMS), reaching 51.6 with replacement and 51.1 with concatenation. Our state-anchored variant reaches 54.9, exceeding them by 3.3 and 3.8 points, respectively. These results favor keeping WAM knowledge in a transition-based modulation of observed scene features in this setting. Because the variants also differ in the number of tokens consumed downstream, this comparison evaluates the complete feature-consumption designs rather than isolating a single architectural factor.

## B Loss-Weight Sensitivity

We vary the WAM feature-alignment weight $\lambda_d$ and the scorer-to-physics injection weight $\lambda_p$ one at a time on NAVSIM-v2 navhard, keeping the other at the default value in subsection 3.4.

Table 6: **Loss-weight sensitivity on NAVSIM-v2 navhard.** One weight is varied at a time from the default $(\lambda_d, \lambda_p) = (1.0, 0.5)$.

| $\lambda_d$ | $\lambda_p$ | EPDMS ↑ |
|---|---|---|
| 0.5 | 0.5 | 51.0 |
| 1.0 | 0.5 | **54.9** |
| 1.5 | 0.5 | 53.4 |
| 2.0 | 0.5 | 53.2 |
| 1.0 | 0.25 | 54.3 |
| 1.0 | 1.0 | 53.7 |

As shown in Table 6, the default $(\lambda_d, \lambda_p) = (1.0, 0.5)$ reaches 54.9 EPDMS, while $\lambda_d = 1.5$ yields 53.4. Reducing $\lambda_d$ to 0.5 lowers EPDMS to 51.0; the other tested weights also do not improve on the default.

## C ZERO-SHOT HUGSIM ROLLOUTS

The main text reports aggregate HUGSIM scores in Table 3. Here we show one representative zero-shot closed-loop rollout for each of the four source datasets: KITTI-360 (Figure 4), nuScenes (Figure 5), Pandaset (Figure 6), and Waymo (Figure 7). All use the NAVSIM-trained planner without training on rendered data. These examples provide visual context for the aggregate scores rather than a separate quantitative comparison.



Figure 4: **KITTI-360 (Liao et al., 2023).** The ego vehicle navigates a turn and avoids nearby vehicles, with yellow dots marking the predicted trajectory.



Figure 5: **nuScenes (Caesar et al., 2020).** The ego vehicle overtakes another vehicle and handles oncoming traffic, with yellow dots marking the predicted trajectory.



Figure 6: **Pandaset (Xiao et al., 2021).** The ego vehicle overtakes a lead vehicle, with yellow dots marking the predicted trajectory.

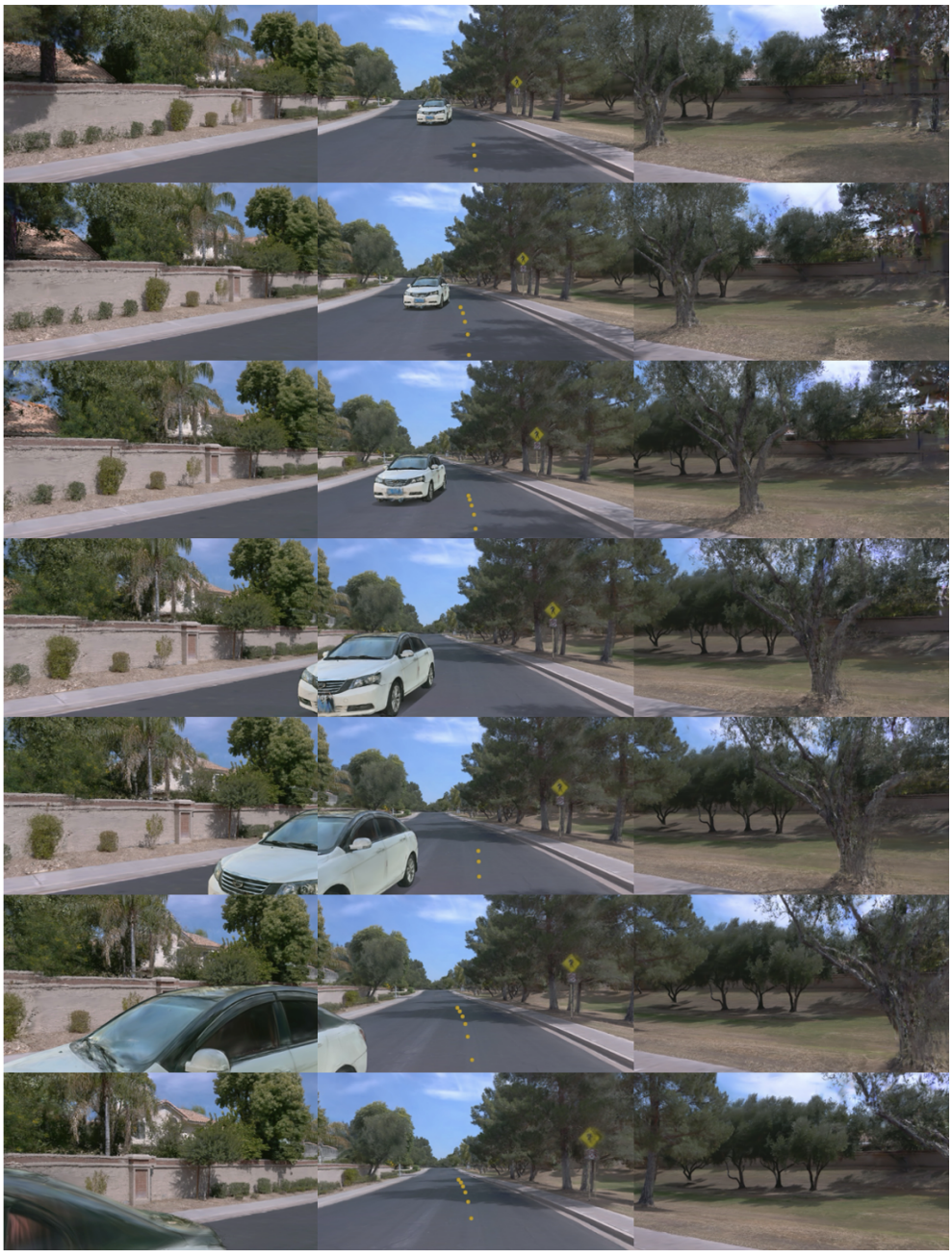

Figure 7: **Waymo (Sun et al., 2020).** The ego vehicle avoids an oncoming vehicle, with yellow dots marking the predicted trajectory.