CLOSING THE LOOP ON OPEN-LOOP DRIVING DATA: REINFORCEMENT LEARNING OF ACTION POLICIES WITH CONTROLLABLE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Policies trained from recorded driving logs rarely observe how their own decisions change future observations, limiting their ability to recover once they depart from the recorded trajectory. We present DriveVerse, an autoregressive multi-view world model that turns logged scenes into interactive environments for policy learning. At each decision step, a structured simulation backend executes part of the policy plan and constructs view-aligned map and motion conditions; DriveVerse then generates synchronized camera observations for the next decision. These post-action observations enable recurrent policy interaction beyond the recorded ego trajectory. Shared cross-view attention, causal training, and self-rollout distillation support recurrent generation, while online diagnostics monitor rollout reliability. We optimize the resulting rollouts with T-GRPO, combining plan-level and executed-trajectory feedback across vision-language-action, world-action, and end-to-end policies through their native action representations. Experiments show that DriveVerse outperforms the strongest baseline, lowering front-view FVD by 22% and improving all reported object- and lane-geometry metrics and temporal consistency. Interactive training with DriveVerse improves all three policy families (VLA, WAM, E2E) over their SFT and RL baselines on both navtest and navhard. The VLA reaches 92.8 and 52.3 EPDMS, respectively, improving on its GRPO baseline by 1.7 and 5.2 points.

1 INTRODUCTION

Driving policies must recover from states caused by their own actions, whereas supervision from expert logs mainly exposes them to states visited by human drivers. Once a policy leaves the recorded trajectory, its next observation changes, and the necessary correction may be absent from the training data. Interactive training closes this gap by letting it act and replan from the resulting states.

Supporting such interaction for a vision-based policy requires both state transitions and the corresponding visual observations. Reconstruction-based simulators provide geometrically grounded novel views through scene-specific fitting (Tian et al., 2026), but applying them across a large dataset requires reconstructing each scene separately. A learned world model instead uses one observation generator across scenes, conditioned on visual history, maps, and structured scene states. Training on generated observations, however, introduces a stricter requirement than evaluating on them, because a policy learns from whatever those observations contain, including their errors.

Therefore, generated observations must follow the executed ego motion, remain synchronized across cameras, and preserve road geometry when reused as policy history. Image history may also miss distant or occluded road features. Figure 1 shows such a case: an image-conditioned rollout loses a junction whose layout remains available in the bird's-eye-view (BEV) map.

These requirements also become harder to maintain over repeated interaction. Feeding generated observations back into the model can cause visual and geometric errors to accumulate, which dataset-level metrics may not reveal. In addition, a local plan reward may not capture delayed outcomes such as route completion or infractions. Policy learning therefore requires both reliable autoregressive generation and feedback across policy decisions.

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.

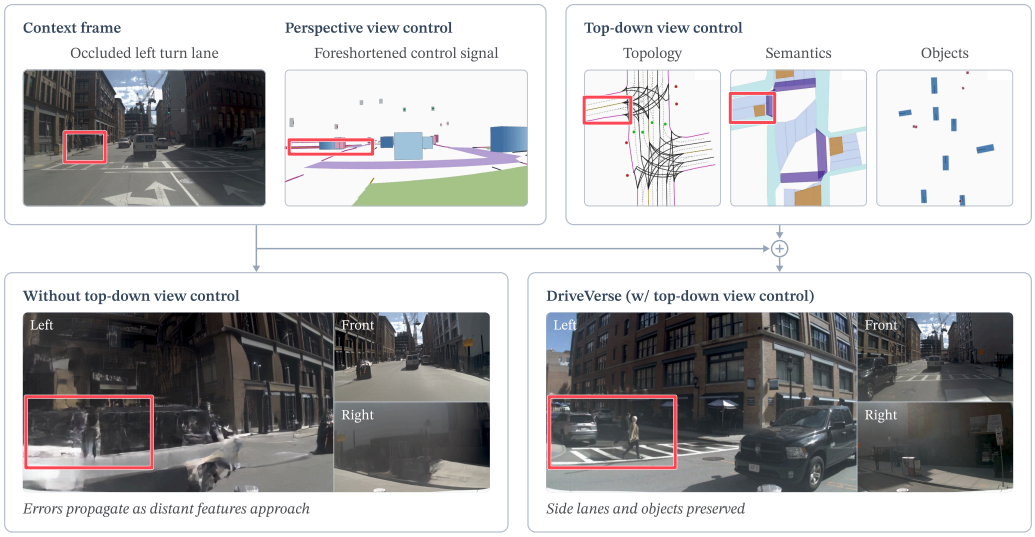


Figure 1: Long-context generation with a driving world model. An initially occluded side street is lost under perspective control alone as the ego approaches, while adding BEV conditioning preserves its layout. Outlined regions identify the same street across the conditioning inputs and generated views.

We introduce DriveVerse, an autoregressive multi-view world model that generates observations for an interactive policy-training loop (Figure 2): a structured simulation backend executes the leading segment of each plan and updates the scene, and DriveVerse renders the resulting synchronized camera views for replanning. Each component addresses one challenge. Map rasters and BEV tokens supply road structure that image history cannot; training on the model's own rollouts limits error accumulation; online checks against the structured state stop unreliable rollouts before they reach the policy; and a trajectory-level driving reward, optimized with T-GRPO (Chen et al., 2025), credits decisions for their delayed outcomes. Our contributions are threefold:

- We develop a highly controllable multi-view world model with innovative map conditioning from ego-view and top-down perspectives, ego-motion conditioning, shared cross-view attention, and few-step autoregressive generation. We train it efficiently using a four-stage teacher–student pipeline, reducing front-view FVD by 22% relative to the state-of-the-art baseline.
- We build an interactive simulation loop around a shared, deterministically updated world state. The policy supplies actions, the world model generates observations from the resulting state for replanning, and online diagnostics monitor observation reliability throughout recurrent rollout.
- We develop an online reinforcement learning pipeline enabled by DriveVerse and optimized with T-GRPO for three driving-policy families: vision-language-action (VLA), world-action-model (WAM), and end-to-end (E2E). Our training improves all three families over their SFT and RL baselines, with the RL-fine-tuned VLA policy achieving EPDMS scores among the best reported on the NAVSIM v2 benchmarks.

2 Methodology

2.1 Overall Interaction Framework

We couple the driving policy and observation generator through a shared world state maintained by a deterministic simulation backend (Figure 2). Driving logs and annotations initialize a world state s_0 containing ego and surrounding-agent states, road geometry, and traffic signals. This structured layer provides the ground-truth scene description within the simulated environment. At decision k, the policy receives camera views, ego state, and navigation input, denoted collectively by o_k, and proposes a plan using its observation history H_k. The backend executes the leading segment a_k of this plan and updates the world state to s_{k+1}. A scene renderer then constructs motion and map conditions c_k along the executed segment, and the world model generates the corresponding synchronized camera observations:

$$s_{k+1} = \mathcal{T}(s_k, a_k), \qquad c_k = \mathcal{R}(s_k, a_k, s_{k+1}), \qquad \hat{o}_{k+1} \sim p_\phi(\cdot \mid H_k, c_k). \tag{1}$$

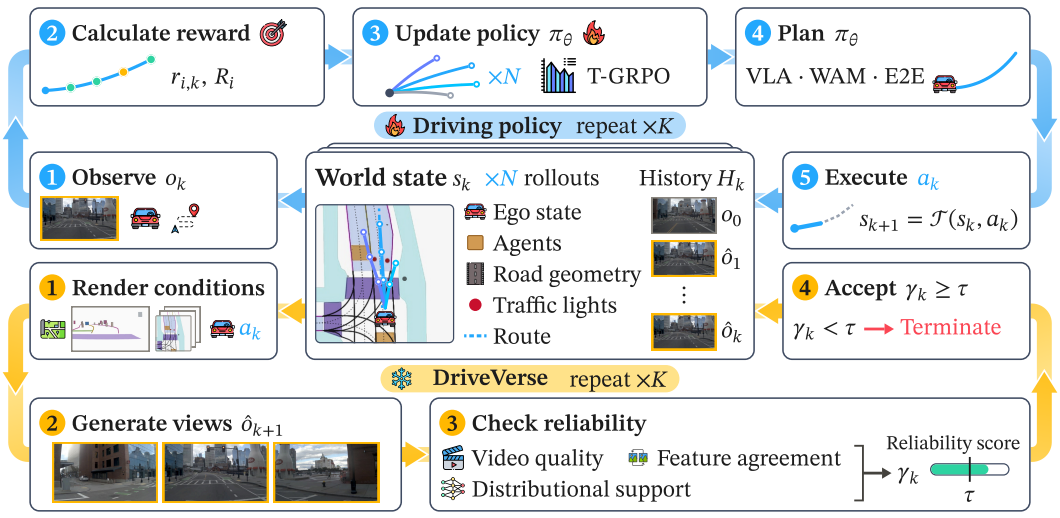


Figure 2: Closed-loop interaction between a driving policy and DriveVerse. The upper (blue) loop executes policy actions in a structured world state and computes driving rewards. The lower (yellow) loop generates synchronized views and checks their reliability before adding them to the observation history. A score $\gamma_k < \tau$ terminates interaction before the next policy decision; rewards retain the executed trajectory up to termination.

where $\mathcal{T}$ is the simulator's state-transition function, $\mathcal{R}$ is the scene renderer that constructs map and ego-motion conditions, and p_ϕ denotes the world model DriveVerse. The generated observations extend the policy history for the next decision only if they pass the online diagnostics in Section 2.3. Only the executed segment advances the persistent state; each new decision replaces the remaining plan. The policy receives sensor history, ego state, and navigation input, while structured future states used to score candidate plans remain internal to the simulator.

Sections 2.2–2.4 describe world-model training, rollout diagnostics, and policy optimization.

2.2 Controllable Multi-View World Model

DriveVerse generates synchronized multi-view future camera frames from visual history and structured driving conditions (Figure 3). We initialize it from the Cosmos 3 generation expert (NVIDIA, 2026a), retaining the pretrained video and action interfaces and shared multimodal attention.

Motion and scene conditioning. Ego-motion conditioning alone provides limited control over road geometry and object placement in generated frames. We therefore execute the policy's action in the structured simulation backend and derive three complementary conditions from the resulting ego trajectory and scene states: relative ego motion, perspective HD-map rasters, and top-down BEV annotations. The motion encoding specifies how the ego moves, while the rendered maps and annotations explicitly constrain road structure and object locations.

For relative ego-motion conditioning, we convert the executed ego trajectory into frame-to-frame relative poses in the ego-view camera coordinate system, following Cosmos 3's pretrained action encoding. Each relative pose is represented by its translation and the first two columns of its rotation matrix, then embedded into an action token using the pretrained linear projection. This conversion accommodates policies with different native action representations.

For each camera, the renderer projects road geometry, traffic signals, and agent boxes along the executed ego motion. We encode maps with the video's causal VAE, preserving alignment between map and video latent grids. For camera v, latent timestamp t, and spatial patch p, the input is

$$h^v_{t,p} = W_{\text{vid}} z^v_{\sigma_t,p} + W_{\text{map}} m^v_{t,p} + e_v + e_{\sigma_t}, \tag{2}$$

where $z^v_{\sigma_t,p}$ is the video latent at noise level σ_t, $m^v_{t,p}$ is the clean perspective-map latent, and e_v and e_{σ_t} identify the camera and noise level. The zero-initialized map projection introduces a residual at existing visual-token positions, preserving the pretrained input representation at initialization. We do not add diffusion noise to map conditions or observed visual context.

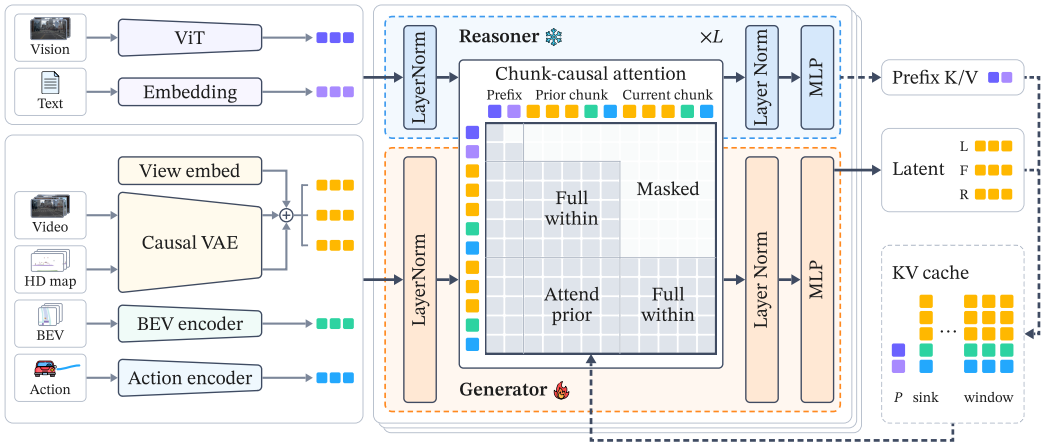


Figure 3: DriveVerse integrates multimodal conditioning through shared attention to generate synchronized camera views. Chunk-causal attention and cached history support long-context generation as the policy replans.

Perspective projection can compress lateral road structure and omit geometry outside the camera frustum. We therefore additionally encode top-down BEV annotations of road topology, semantic areas, and objects as tokens b_t. These tokens expose the surrounding layout to the generator, complementing the dense camera-aligned residual in Equation 2. The perspective maps and BEV tokens are derived from the same structured state and aligned with the same ego pose.

Chunk-causal generation. After observing each generated segment, the policy replans, so future executed actions and scene conditions are unknown. We therefore use chunk-causal generation, conditioning each chunk on available history and its aligned motion and scene conditions.

The sequence begins with a conditioning prefix P containing visual-history features and embedded text tokens. Generation tokens are grouped by latent timestamp:

$$\mathcal{C}_t = [h_t^{\mathrm{F}}, h_t^{\mathrm{L}}, h_t^{\mathrm{R}}, b_t, a_{\mathcal{I}_t}], \tag{3}$$

where h_t^v collects the spatial tokens of view v and $\mathcal{I}_t$ indexes the motion increments corresponding to that VAE chunk. Shared multimodal attention couples the camera views, BEV representation, and actions. In the causal model, generation tokens attend within their synchronized chunk and to preceding chunks, while the autoregressive conditioning subsequence retains its causal mask. The same temporal boundary across modalities prevents future information from entering through another token stream. This reuses pretrained attention without a separate cross-view attention stack.

Bidirectional and causal training. Our four-stage teacher–student pipeline combines bidirectional fine-tuning, diffusion forcing, consistency distillation, and distribution matching distillation for robust few-step autoregressive generation (Figure 4); optimization details are in Appendix A.2.

Bidirectional teacher fine-tuning. We adapt the pretrained generator to driving with full-sequence attention and a noise level shared across timestamps and views. Joint training of the conditioning modules and generation weights learns multi-view scene structure. Progressive increases in resolution subsequently refine visual detail. For clean latents z_t^v and Gaussian noise ϵ_t^v, the rectified-flow objective (Liu et al., 2022) is

$$\mathcal{L}_{\mathrm{RF}} = \mathbb{E}\sum_{t,v} w_{t,v} \left\| u_\phi(z_{\sigma_t}^v, \sigma_t; c) - (\epsilon_t^v - z_t^v) \right\|_2^2, \quad z_{\sigma_t}^v = (1-\sigma_t) z_t^v + \sigma_t \epsilon_t^v, \tag{4}$$

where c denotes the conditioning inputs, u_ϕ predicts latent velocity, and $w_{t,v}$ weights valid targets.

Diffusion forcing. We adapt the pretrained generator using the chunk-causal mask and Diffusion Forcing (Chen et al., 2024). Noise levels are sampled independently across timestamps and shared across synchronized views. A randomly selected prefix remains clean. The same flow objective thus trains the model to predict from histories with different noise levels and context lengths, matching the temporal information available during incremental generation.

Consistency distillation. We distill the causal model into a few-step generator by matching the student's clean prediction to an exponential-moving-average target evaluated after a teacher ODE

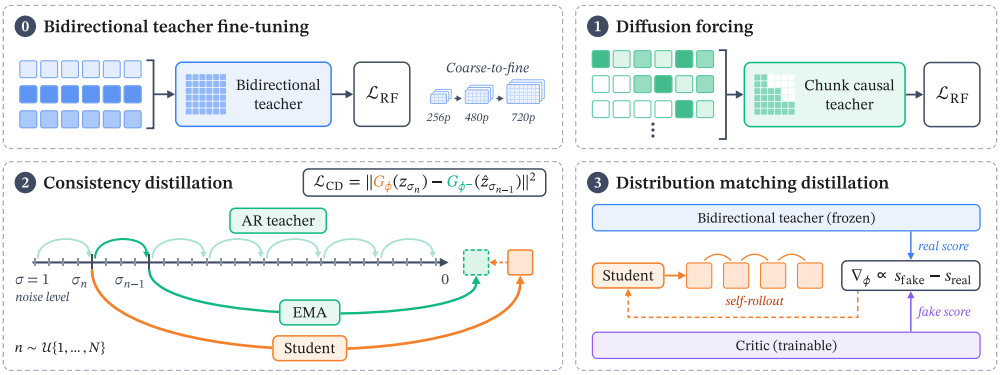


Figure 4: Bidirectional teacher fine-tuning adapts the world model to driving with full-sequence attention; diffusion forcing trains chunk-causal generation. Consistency distillation enables few-step sampling, while distribution matching distillation improves robustness during autoregressive rollout.

step (Zhao et al., 2026a). Both predictions condition on the same preceding context. Using a causal teacher aligns the teacher and student's information access; local consistency supervision learns the denoising map without requiring stored full teacher trajectories.

Distribution matching distillation. The student generates its own autoregressive contexts and is refined using scores from a frozen bidirectional teacher and a trainable critic (Zhu et al., 2026; Zhao et al., 2026a). Conditioning on its own outputs, the causal student learns to match the teacher distribution, addressing the gap between recorded training and generated rollout histories. The last two stages reduce repeated denoising computation and improve robustness to recurrent generation.

2.3 Online Rollout and Diagnostics

We check each generated segment for consistency with the simulated world state before returning it to the policy (Figure 2). The diagnostics use generated frames, accepted history, structured scene conditions, and a reference bank of real driving data; no ground-truth images at counterfactual poses are required. Scores lie in $[0, 1]$, with larger values indicating greater reliability.

Diagnostic signals. We combine three complementary checks. Video quality q_k^{vid} measures perceptual drift with MUSIQ and temporal consistency with frame interpolation, following VBench-I2V (Ke et al., 2021; Huang et al., 2024). Feature agreement q_k^{obj} and q_k^{lane} compares YOLOv12 and CLRerNet detections with visible objects and lanes projected from the structured state (Tian et al., 2025; Honda & Uchida, 2024), combining coverage and localization accuracy. Distributional support q_k^{sup} measures proximity to real state–action pairs using Cosmos Embed features and relative ego motion (NVIDIA, 2025; Yu et al., 2026). Appendix A.4 defines each score.

Rollout gating. We use the reliability score, $\gamma_k = \min\{q_k^{\text{vid}}, q_k^{\text{obj}}, q_k^{\text{lane}}, q_k^{\text{sup}}\}$, so acceptance requires every applicable criterion to meet the threshold. Before interaction, we filter initial clips using generations conditioned on recorded ego motion, with a reliability threshold of 0.80. Within this retained pool, online generation uses $\tau = 0.74$, calibrated on validation trajectories with manual inspection near the acceptance boundary (Appendix A.5). Scales and tolerances are also fixed using validation data. If $\gamma_k \geq \tau$, the segment extends the accepted history; otherwise, it is rejected and the rollout terminates before decision $k+1$. The executed state transition remains part of the trajectory; driving rewards use this structured-state prefix, as described next.

2.4 Trajectory-Wise Policy Optimization

We optimize policy rollouts in DriveVerse using T-GRPO (Chen et al., 2025), with driving rewards and each policy's native action interface. Its step- and trajectory-group advantages combine local planning feedback with outcomes across successive decisions (Appendix C.1). We sample $N = 8$ rollouts per initial scene, replanning every $0.5\,\text{s}$ for up to eight decisions over $4\,\text{s}$. Each executed trajectory τ_i concatenates the segments committed to the simulator, with $K_i \leq K = 8$ decisions.

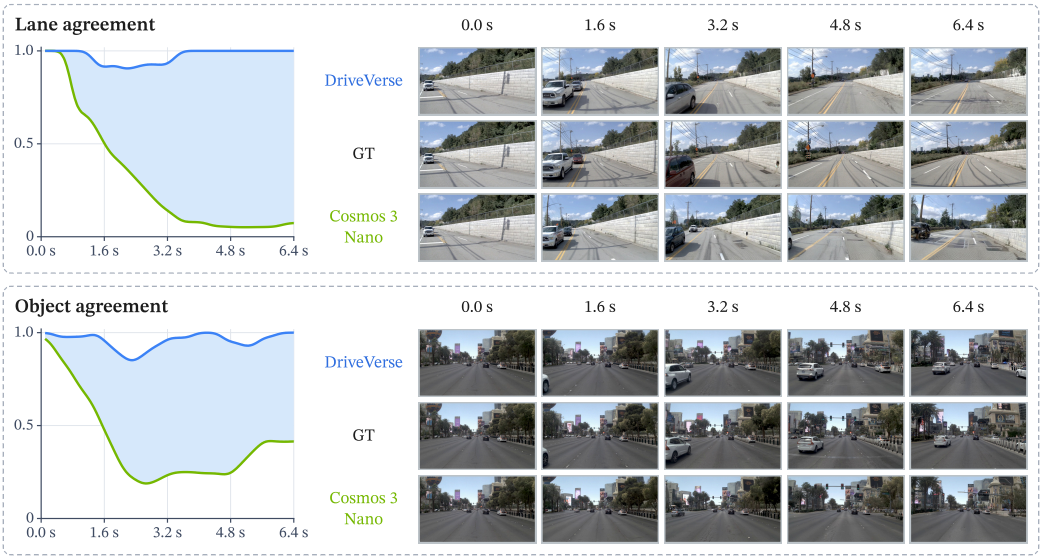


Figure 5: Online feature agreement over two driving sequences. DriveVerse preserves lane geometry (top) and surrounding traffic (bottom) more consistently than Cosmos 3 Nano. Agreement curves capture localized deviations as well as sustained scene drift. Real recordings (GT) provide a visual reference; the diagnostics compare generated features with structured scene conditions.

Plan and execution rewards. The local reward $r_{i,k}$ is the NAVSIM v2 EPDMS of the proposed plan, scored from the current structured state. Its extended comfort (EC) term compares successive plans over their overlapping horizon, supervising consistency along the policy's own rollout. We also score τ_i from the initial scene using the same PDM backend, combining completion, infractions, and driving quality following HUGSIM and Bench2Drive (Zhou et al., 2024; Jia et al., 2024):

$$\mathrm{CLS}_i = \mathrm{RC}_i\,\mathrm{IP}_i\,\mathrm{DQ}_i \in [0,1], \qquad r_i^{\mathrm{cl}} = \lambda(\mathrm{CLS}_i - 1), \qquad R_i = \frac{K}{K_i}\sum_{k=0}^{K_i-1} r_{i,k} + r_i^{\mathrm{cl}}. \tag{5}$$

Here, RC_i is the fraction of the scored duration before the first critical driving failure, IP_i multiplicatively penalizes infractions and stalling, and DQ_i combines progress, time-to-collision compliance, lane keeping, and comfort. The closed-loop term lies in $[-\lambda, 0]$; λ starts at 6 and is re-estimated during training to match its within-group standard deviation to that of the local-return term. The factor K/K_i scales local returns to eight decisions and equals one for complete rollouts. Appendix C.2 gives the full definitions. Driving failures do not terminate rollouts. A diagnostic rejection stops only the affected instance. For scoring, we retain its executed prefix and complete the horizon using its latest full plan in the structured backend. This planned tail adds no policy decisions; the diagnostic stop itself carries no penalty, and other group members continue independently.

Policy interfaces. VLA RL updates the VLM through reasoning- and trajectory-token log-probabilities. A fixed token decoder supplies the scored and executed plans; the flow-matching expert is absent from RL rollout and update forward passes. WAM uses transition log-probabilities along its trajectory-denoising chain; E2E uses categorical log-probabilities over trajectory candidates. All three share driving rewards and the executed-motion interface, with the world model fixed during policy optimization (Appendix B.1).

3 Experiments

Dataset and training. We use nuPlan (Caesar et al., 2021) driving logs with synchronized cameras, machine-labeled scene annotations, and HD maps. Anchored at NAVSIM (Dauner et al., 2024) timestamps, our adaptation corpus contains 89,668 training and 1,771 validation clips from approximately 200 hours of driving; evaluation uses 1,000 uniformly sampled test clips. We sample the front, front-left, and front-right cameras at 10 Hz and crop frames to 1280×720. Each clip pairs 21 observed frames with 64 future frames, extended to 200 for long-horizon DMD. Section 2.2 specifies the training stages; Appendix A gives preprocessing and optimization details.

Table 1: Offline comparison on video quality and feature agreement on 1,000 clips from NAVSIM's navtest. FVD reports front / left / right views or front view only based on model capability. Other reference baselines retain their published inference configuration. Feature agreement is measured using frozen BEVFormer (Li et al., 2022) for 3D detection with LET-AP and LET-APL (Hung et al., 2024) and LATR (Luo et al., 2023) for lane detection (LATR-F1 and far-range lateral RMSE, LATR-x-err.).

Model	Generation quality			Feature Agreement			
	FVD ↓	PSNR ↑	Sampson ↓	LET-AP ↑	LET-APL ↑	LATR-F1 ↑	LATR-x-err. ↓
SVD img2vid-xt (2023)	476.14	14.89	2.92	0.129	0.073	0.266	0.897
GEM (2024)	91.79	18.78	2.28	0.133	0.080	0.176	0.940
DrivingWorld (2024)	327.63	16.38	2.54	0.175	0.102	0.372	0.694
Vista (conditioned) (2024)	169.95	18.77	2.75	0.152	0.088	0.319	0.785
Epona (2025)	78.51	20.57	2.46	0.206	0.120	0.473	0.606
Wan2.2-TI2V-5B (2025)	237.27	15.11	2.61	0.150	0.087	0.293	0.799
Cosmos 3 Super (2026a)	37.22	21.08	2.03	0.292	0.183	0.554	0.438
Cosmos 3 Nano (2026a)	39.30	20.86	2.08	0.279	0.173	0.536	0.461
OmniDreams (2026b)	22.7 / 21.7 / 30.0	27.50	2.01	0.397	0.253	0.873	0.316
DriveVerse (ours)	**17.7 / 19.5 / 28.2**	**28.35**	**1.86**	**0.407**	**0.261**	**0.892**	**0.306**

Table 2: Ablations on conditioning modalities.

Module	FVD ↓	LET-AP ↑	LATR-F1 ↑
Module removal			
w/o map residual	27.6	0.389	0.818
w/o BEV tokens	27.8	0.387	0.819
w/o ego-motion tokens	27.5	0.389	0.820
Module replacement			
Dedicated cross-view attention	21.4	0.411	0.835
AdaLN view embedding	29.01	0.384	0.811

Table 3: Training-stage ablations. The teacher and causal models start from Cosmos 3 Nano; (+) continues from the preceding stage.

Stage	Steps	FVD ↓	LET-AP ↑	LATR-F1 ↑
Bidirectional teacher	35	24.6	0.390	0.820
Causal diffusion forcing	35	20.1	0.402	0.829
+ Consistency distillation	4	20.3	0.400	0.827
+ Long-horizon DMD	4	17.7	0.407	0.892

Generation and monitoring metrics. We generate 6.4 s continuations under recorded motion and compare them with held-out futures using FVD, PSNR, and temporal Sampson error. Frozen BEVFormer (Li et al., 2022) and LATR (Luo et al., 2023) measure object and lane agreement through LET-AP, LET-APL, LATR-F1, and LATR-x-err. All methods use the same evaluation models and fixed random seeds (Appendix A.3).

Online diagnostics assess segments against the simulated state and available history (Section 2.3). Figure 5 pairs object and lane agreement with generated sequences and real recordings for reference.

Policy learning and evaluation. For each selected policy, we use the strongest available open-source RL implementation: NVIDIA's GRPO recipe for Alpamayo-R1 (NVIDIA et al., 2025), SimWAM's released FlowGRPO checkpoint (Zhao et al., 2026b; SimWAM Authors, 2026), and WorldEngine's HydraMDP RL checkpoint (Li et al., 2024; 2026). SFT denotes supervised initialization. Baselines follow the published training recipes and all policies use official NAVSIM v2 evaluation (Appendix B.1). Training and checkpoint selection use training and validation logs. Online rollouts follow a common route for 4 s with 0.5 s replanning. We report EPDMS (0–100) on `navtest` and two-stage `navhard`, combining original and pre-generated follow-up observations with official proximity weighting (Cao et al., 2025).

3.1 Controllable Video Generation

Offline comparison. DriveVerse leads every reported generation and feature-agreement metric in Table 1. All models use the shared test set with their supported conditioning and published inference configurations. Front-view FVD falls from OmniDreams' 22.7 to 17.7, a 22% reduction, alongside improved left- and right-view FVD. Higher PSNR, lower Sampson error, and gains in all four feature-agreement metrics establish improved visual quality, temporal consistency, and object and lane agreement under recorded motion.

Conditioning and architecture. Removing map residuals, BEV tokens, or ego-motion tokens worsens FVD, LET-AP, and LATR-F1 (Table 2), supporting their complementary roles. AdaLN view embeddings degrade all three metrics. Shared attention improves FVD from 21.4 to 17.7 and LATR-F1 from 0.835 to 0.892 over dedicated cross-view attention, with LET-AP of 0.407 versus 0.411. Nano jointly adapts about 0.8M added map and view parameters with pretrained generation weights.

Table 4: NAVSIM v2 policy performance. DriveVerse with T-GRPO improves all three policy families over SFT and their open-source RL baselines. VLA GRPO and WAM FlowGRPO use logged observations; WorldEngine RL uses interactive 3D Gaussian Splatting simulation. †: evaluated using released checkpoints.

Method	Training setup	navtest v2 EPDMS					navhard v2 EPDMS									
							Stage 1					Stage 2				
		NC	DAC	EP	EC	EPDMS	NC	DAC	EP	EC	EPDMS	NC	DAC	EP	EC	EPDMS	EPDMS
VLA base planning policy																	
Alpamayo-R1 (NVIDIA et al., 2025)	SFT	98.9	97.0	87.0	84.5	89.3	97.6	93.1	83.3	77.8	81.7	84.6	76.4	84.6	68.9	45.7	38.1
	GRPO	99.3	98.4	90.0	82.0	91.1	97.6	93.8	81.3	76.9	82.0	89.4	84.7	83.0	63.2	56.8	47.1
	DriveVerse RL (T-GRPO)	99.4	98.5	93.4	86.0	**92.8**	98.0	95.8	80.9	85.8	87.6	90.2	84.7	81.9	68.6	59.6	**52.3**
WAM planning policy																	
SimWAM (Zhao et al., 2026b)	SFT†	98.6	98.1	87.4	84.7	90.3	97.8	92.0	83.8	72.4	80.6	81.8	78.4	86.3	68.8	46.0	37.3
	FlowGRPO†	98.3	98.5	90.1	11.0	82.1	95.6	95.3	86.6	11.1	74.0	81.9	79.4	89.9	9.7	41.0	31.2
	DriveVerse RL (T-GRPO)	99.0	98.8	90.8	87.0	92.0	98.4	94.7	85.8	79.8	84.9	85.4	82.7	88.3	75.3	52.8	44.8
End-to-end planning policy																	
HydraMDP (Li et al., 2024)	SFT†	99.0	97.9	90.3	77.6	88.8	96.4	92.9	84.3	57.3	74.0	85.2	81.7	90.0	51.2	44.4	32.7
	WorldEngine RL†	98.6	97.5	89.6	63.7	85.2	96.2	90.4	81.1	39.1	67.9	89.1	82.2	87.0	35.7	46.9	33.0
	DriveVerse RL (T-GRPO)	99.2	98.6	92.1	84.0	91.4	97.2	94.1	88.1	71.8	80.5	89.8	85.0	90.4	66.4	54.2	43.6

Training stages. Table 3 compares the bidirectional teacher, causal diffusion forcing, and successive distillation stages. Causal diffusion forcing improves all three metrics over this teacher. Consistency distillation reduces denoising from 35 to four steps per chunk with little change in quality; subsequent long-horizon distribution matching distillation (DMD) improves FVD from 20.3 to 17.7 and raises both feature-agreement scores. The final row is the model in Table 1. Causal generation and distillation thus support incremental interaction with less repeated denoising computation.

Online diagnostic behavior. Figure 5 shows that DriveVerse maintains the specified road layout and traffic over the illustrated horizon. Cosmos 3 Nano's lane and object scores fall as the generated scene drifts. Transient dips in DriveVerse's curves identify localized deviations, complementing clip-level metrics with time-resolved feature agreement.

Recurrent rollout reliability. With the frozen VLA SFT policy and final DMD-trained world model, we sample eight rollouts per retained test clip, replanning every $0.5\,\mathrm{s}$. Initial filtering at 0.80 retains 65.3% of test clips; approximately 87.2% of all attempted rollouts from this pool complete $4\,\mathrm{s}$ with every segment passing the validation-calibrated threshold of 0.74. The corresponding `navtrain` rates are 71.3% and 90.3%. Checkpoints and diagnostic parameters are fixed during evaluation. Appendix A.5 details calibration and $20\,\mathrm{s}$ generation under recorded motion.

3.2 Policy Improvement Through Online Interaction

Table 5: Interaction and advantage ablations of Alpamayo-R1 (2025). Both benchmarks report EPDMS.

Training setup	navtest ↑	navhard ↑
Logged-observation baseline		
GRPO	91.1	47.1
Online interaction with DriveVerse		
Step advantage only	92.1	49.4
Trajectory advantage only	91.8	50.0
T-GRPO (DriveVerse RL)	92.8	52.3

Interactive training with DriveVerse, using T-GRPO, improves all three policy families over SFT and their strongest available open-source RL baselines on both benchmarks (Table 4). The VLA achieves 92.8 on `navtest` and 52.3 on `navhard`, gains of 1.7 and 5.2 points over GRPO and 3.5 and 14.2 points over SFT. These EPDMS scores rank among the best reported on the NAVSIM v2 benchmarks (Appendix D).

For the VLA on `navtest`, ego progress rises from 90.0 to 93.4 and EC from 82.0 to 86.0 relative to GRPO, while collision avoidance and drivable-area compliance also improve. On `navhard`, its first- and second-stage scores rise from 82.0 to 87.6 and from 56.8 to 59.6, respectively. The gains therefore extend across progress, comfort, and performance in follow-up states.

Interaction and advantage ablations. Table 5 matches VLA initialization, scenes, sample counts, and update budgets. DriveVerse RL variants replan from generated observations and use the step advantage, trajectory advantage, or both (T-GRPO). Each single-advantage variant improves on logged-observation GRPO. Combining both yields the strongest scores, exceeding step-only training

by 0.7 points on `navtest` and 2.9 on `navhard`. These gains support both observation feedback and trajectory-level credit assignment.

The WAM results show the value of successive planning feedback. Released FlowGRPO optimizes PDM rewards for alternative plans from each logged observation (SimWAM Authors, 2026). DriveVerse supplies post-action observations for EC supervision between successive plans. The WAM reaches 92.0 and 44.8 EPDMS on `navtest` and `navhard`, exceeding SFT by 1.7 and 7.5 points.

The E2E policy improves over WorldEngine's 3D Gaussian Splatting-based interactive post-training (Li et al., 2026), reaching 91.4 and 43.6 EPDMS. All three families also improve their second-stage `navhard` scores over their RL baselines (Table 6), demonstrating the shared framework's effectiveness across native action representations.

4 Related work

Reconstructed and generative driving simulation. FreeVS (Wang et al., 2025b) conditions a shared generator on projected colored point clouds, retaining geometric priors without per-scene model optimization. Its off-trajectory evaluation distinguishes realistic appearance from agreement with the underlying geometry. Our approach uses calibrated camera views, map conditions, and executed ego motion to support repeated policy interaction with a shared observation generator.

Controllable multi-view generation. Glad (Xie et al., 2025a) generates driving video sequentially using propagated latents and layout conditions. X-Drive (Xie et al., 2025b) jointly generates camera and LiDAR observations with geometric cross-modality conditioning, while OmniDreams (NVIDIA, 2026b) supports autoregressive multi-view simulation. We reuse pretrained attention and action interfaces for driving adaptation and integrate the resulting observation model into online policy optimization, where the policy receives new observations after executing actions.

Autoregressive diffusion. Diffusion Forcing (Chen et al., 2024) trains causal sequence models with independently sampled noise levels across timestamps. Causal Forcing and Causal Forcing++ (Zhu et al., 2026; Zhao et al., 2026a) develop few-step generators trained on model-generated contexts. We build on these methods to generate synchronized camera views under map and motion control.

World models for policy learning. LAW (Li et al., 2025b) improves planning representations through future latent prediction, and AdaWM (Wang et al., 2025a) adapts models and policies according to estimated mismatch. Our world model supplies RGB observations after executed actions, enabling the policy to replan from its own visited states. We adopt the step and trajectory grouping of T-GRPO (Chen et al., 2025), with likelihood models suited to all three policy families.

5 Discussion and conclusion

We presented DriveVerse, a highly controllable multi-view world model that turns driving logs into interactive environments for policy learning. Ego-view and top-down map conditioning, ego-motion inputs, shared cross-view attention, and a four-stage teacher–student pipeline support efficient autoregressive generation. A shared, deterministically updated world state links policy actions to generated observations, with online diagnostics monitoring rollout reliability. Using T-GRPO, the resulting online RL pipeline combines plan-level and trajectory-level feedback and improves VLA, WAM, and E2E policies over their SFT and RL baselines. The fine-tuned VLA achieves 92.8 EPDMS on `navtest` and 52.3 on `navhard`, among the best reported NAVSIM v2 scores.

Limitations. Beyond camera recordings, DriveVerse requires aligned HD maps and machine-labeled scene annotations to construct conditioning inputs and evaluate driving rewards. World-model controllability is bounded by the scenes and motion conditions it can reliably reproduce, so initial filtering and online diagnostics select the data and rollout segments used for policy training.

AI USE STATEMENT

We used ChatGPT and Claude to revise manuscript text for clarity and grammar, identify relevant literature, and check descriptions of existing methods against their published implementations. These assistants were not used to originate the research ideas, generate experimental datasets, or prove mathematical claims. The authors retain responsibility for the accuracy and integrity of the manuscript and all accompanying materials.

ETHICS STATEMENT

This work studies driving-policy learning in simulation using existing driving datasets. Structured collision, road-compliance, and comfort rewards guide policy optimization, and observation-reliability checks govern rollout collection. Evaluation follows the established NAVSIM protocols. Deployment in physical vehicles is subject to the applicable safety-validation and regulatory processes.

REPRODUCIBILITY STATEMENT

Section 2 describes the world-model architecture, training stages, rollout diagnostics, and policy optimization objectives. Section 3 specifies the data splits and preprocessing, pretrained checkpoints, evaluation metrics, and matched ablation settings. Additional implementation and experimental details are provided in the appendix.

REFERENCES

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. URL https://arxiv.org/abs/2311.15127.

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. *arXiv preprint arXiv:2106.11810*, 2021.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning*, 2025. URL https://arxiv.org/abs/2506.04218.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

Zengjue Chen, Runliang Niu, He Kong, Qi Wang, Qianli Xing, and Zipei Fan. TGRPO: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization. *arXiv preprint arXiv:2506.08440*, 2025.

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*, 2022.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. ARTEMIS: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 2025.

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. *arXiv preprint arXiv:2405.17398*, 2024. URL https://arxiv.org/abs/2405.17398.

Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro M. B. Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, Marco Cannici, Elie Aljalbout, Botao Ye, Xi Wang, Aram Davtyan, Mathieu Salzmann, Davide Scaramuzza, Marc Pollefeys, Paolo Favaro, and Alexandre Alahi. GEM: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. *arXiv preprint arXiv:2412.11198*, 2024. URL https://arxiv.org/abs/2412.11198.

Hiroto Honda and Yusuke Uchida. CLRerNet: Improving confidence of lane detection with LaneIoU. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. URL https://arxiv.org/abs/2305.08366.

Xiaotao Hu, Wei Yin, Mingkai Jia, Junyuan Deng, Xiaoyang Guo, Qian Zhang, Xiaoxiao Long, and Ping Tan. DrivingWorld: Constructing world model for autonomous driving via video GPT. *arXiv preprint arXiv:2412.19505*, 2024. URL https://arxiv.org/abs/2412.19505.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024. URL https://arxiv.org/abs/2411.13503.

Wei-Chih Hung, Vincent Casser, Henrik Kretzschmar, Jyh-Jing Hwang, and Dragomir Anguelov. Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8272–8279, 2024. doi: 10.1109/ICRA57147.2024.10609986.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.03877.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2108.05997.

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. *arXiv preprint arXiv:2601.05083*, 2026.

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

Tianyu Li, Li Chen, Caojun Wang, Haochen Liu, Kashyap Chitta, Zhenjie Yang, Yuhang Lu, Naisheng Ye, Yihang Qiu, Yufei Wang, Luoxi Zou, Jiaxin Peng, Jin Pan, Zhaoyu Su, Andrei Bursuc, Shengbo Eben Li, Andreas Geiger, Peng Su, and Hongyang Li. World engine: Towards the era of post-training for autonomous driving, 2026. URL https://arxiv.org/abs/2606.19836.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing End-to-End Autonomous Driving with Latent World Model. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6aa4967920e495e90aeeaa3acf18d019-Abstract-Conference.html.

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*, 2025c.

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, 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*, 2025d.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View representation from multi-camera images via spatiotemporal transformers, 2022. URL https://arxiv.org/abs/2203.17270.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yueru Luo, Chaoda Zheng, Xu Yan, Kun Tang, Chao Zheng, Shuguang Cui, and Zhen Li. LATR: 3d lane detection from monocular images with transformer, 2023. URL https://arxiv.org/abs/2308.04583.

NVIDIA. Cosmos-Embed1, 2025. URL https://research.nvidia.com/labs/dir/cosmos-embed1/.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026a. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

NVIDIA. NVIDIA OmniDreams: Real-time generative world model for closed-loop autonomous vehicle simulation. *arXiv preprint arXiv:2606.03159*, 2026b.

NVIDIA, Yan Wang, Wenjie Luo, et al. Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025. URL https://arxiv.org/abs/2511.00088.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. ExploreVLA: Dense world modeling and exploration for end-to-end autonomous driving. *arXiv preprint arXiv:2604.02714*, 2026.

SimWAM Authors. SimWAM: Official implementation and benchmark results, 2026. URL https://github.com/H-EmbodVis/SimWAM. NAVSIM v2 results in the official repository, accessed September 10, 2026.

Team Wan. Wan2.2-TI2V-5B. Official model release, 2025. URL https://huggingface.co/Wan-AI/Wan2.2-TI2V-5B.

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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2511.23369.

Yunjie Tian, Qixiang Ye, and David Doermann. YOLOv12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025. URL https://arxiv.org/abs/2502.12524.

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. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d4c745dcbaf8ef0d7e145754e31b1516-Abstract-Conference.html.

Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. FreeVS: Generative View Synthesis on Free Driving Trajectory. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/93b4d708976a1d9b1250c400e7fda811-Abstract-Conference.html.

Bin Xie, Yingfei Liu, Tiancai Wang, Jiale Cao, and Xiangyu Zhang. Glad: A Streaming Scene Generator for Autonomous Driving. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9a0ec010802c72b238e41490fa03e4d2-Abstract-Conference.html.

Yichen Xie, Chenfeng Xu, Chensheng Peng, Shuqi Zhao, Nhat Ho, Alexander T. Pham, Mingyu Ding, Masayoshi Tomizuka, and Wei Zhan. X-Drive: Cross-modality Consistent Multi-Sensor Data Synthesis for Driving Scenarios. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7ae38278240b94cc08928b01d552209c-Abstract-Conference.html.

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.

Hongzhan Yu, Chenghao Li, Ruipeng Zhang, Henrik Christensen, and Sicun Gao. Sensitivity shaping for latent modeling. *arXiv preprint arXiv:2606.14585*, 2026. URL https://arxiv.org/abs/2606.14585.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. *arXiv preprint arXiv:2506.24113*, 2025. URL https://arxiv.org/abs/2506.24113.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026a.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. SimWAM: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026b.

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. URL https://arxiv.org/abs/2412.01718.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. DiffusionDriveV2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

A World-model implementation and evaluation

A.1 Data preprocessing and caching

The splits and video configuration are specified in Section 3. We remove clips with incomplete side-camera coverage or failed map alignment, and match the test-set size used by OmniDreams (NVIDIA, 2026b). All stages use the fixed caption "Synchronized front-left, front, and front-right camera view of the ego vehicle driving on the road."

HD-map rasters are generated offline in ego-centric coordinates tied to the simulated pose. The renderer includes lane dividers, center lines, road boundaries, crosswalks, stop lines, traffic-light state, and dynamic-agent boxes, with a terrain-aware elevation field preserving map geometry on grades. Video and view-aligned map rasters are center-cropped and resized to 1280×720, then encoded by the shared video VAE to matching latent grids. We cache the video and HD-map latents and verify cache fingerprints against the camera set, map crop, temporal window, and resolution.

A.2 Optimization and inference

The bidirectional and causal stages train the generation expert, video projections, pretrained automotive action heads, zero-initialized HD-map projection, and view embeddings; the language-understanding expert remains frozen. We use bfloat16 FSDP over 64 GPUs, two clips per GPU, selective activation checkpointing, and a packed-token ceiling of 262,144. Optimization uses AdamW with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-6}$, weight decay 10^{-3}, and gradient clipping at 10. The causal stage uses a base learning rate of 1.2×10^{-4}, 100 warm-up iterations, and cosine decay over 2,100 iterations. The newly introduced map-projection and view parameters receive a $5\times$ learning-rate multiplier during the first epoch and the base rate thereafter. Noise levels are sampled uniformly before the resolution-dependent shift; at 720p we use shift 5, matching the deployed denoising schedule.

For consistency and distribution-matching distillation, the generator uses AdamW with learning rate 2×10^{-6}, $\beta = (0, 0.999)$, and weight decay 10^{-2}. The critic uses learning rate 1.2×10^{-6} with the same optimizer settings. Both schedules are flat. The EMA decay is 0.99, the distribution-matching score timestep is sampled uniformly and clamped to $[0.02, 0.98]$ after shift 5, and distillation runs for 300 generator iterations.

Unless noted otherwise, the bidirectional and undistilled causal models use 35 denoising steps; the distilled model uses four steps per chunk. Autoregressive inference pins the first latent as an attention sink and retains the five most recent latent chunks.

A.3 Generation evaluation protocol

We generate 6.4 s continuations of 1,000 held-out clips using fixed random seeds and compare them with recorded futures. Models with an action interface receive the recorded ego motion, with aligned scene conditions supplied where supported. FVD measures distributional video quality per view, PSNR measures framewise reconstruction, and temporal Sampson error measures two-frame geometric consistency. Frozen BEVFormer (Li et al., 2022) and LATR (Luo et al., 2023) assess feature agreement with ground-truth scene annotations: LET-AP and its longitudinal-affinity-weighted variant LET-APL (Hung et al., 2024) evaluate object detection, while LATR-F1 and far-range lateral RMSE (LATR-x-err.) evaluate lane geometry. We apply the same evaluation models across methods. Tables 1–3 use this protocol for baseline and ablation comparisons.

A.4 Online diagnostic definitions

Video quality. We adapt the image-quality and motion-smoothness measures in VBench-I2V (Huang et al., 2024). Let ΔQ_k be the mean absolute difference between the MUSIQ (Ke et al., 2021) scores of generated frames and the initial context frame of the corresponding view. Let $E_k^{\text{int}} \in [0, 1]$ be the mean absolute pixel error between generated intermediate frames and their reconstructions from a pretrained interpolation model, using normalized pixel intensities. Averaging over the current segment and camera views, we define

$$q_k^{\text{vid}} = \exp(-\Delta Q_k / b_Q)\big(1 - E_k^{\text{int}}\big), \tag{6}$$

Here, $b_Q > 0$ sets perceptual-quality tolerance; quality drift and temporal discontinuities lower q_k^{vid}.

Feature agreement. YOLOv12 (Tian et al., 2025) and CLRerNet (Honda & Uchida, 2024) extract objects and lane segments. For feature type $f \in \{\text{obj}, \text{lane}\}$, let G_k^f contain visible features projected from agent annotations or HD-map geometry within the sensing range. We match detections to these features at corresponding timestamps along the executed motion. For each match $g \in M_k^f \subseteq G_k^f$, the localization error d_g is one minus box IoU for objects or mean point-to-curve distance for lanes; matches require $d_g \le r_f$. We combine feature coverage and localization as

$$q_k^f = \frac{1}{|G_k^f|} \sum_{g \in M_k^f} \left[1 - \frac{(d_g - \epsilon_f)_+}{r_f - \epsilon_f} \right]_+, \qquad 0 \le \epsilon_f < r_f, \tag{7}$$

where $(x)_+ = \max(x, 0)$ and ϵ_f accommodates detector and projection error. Missing features contribute zero, while matched features receive full credit within this tolerance and decreasing credit as their locations deviate. Aggregation omits feature types with no expected visible instances.

Distributional support. We encode accepted visual history with Cosmos Embed (NVIDIA, 2025), $e_k = E(H_k)$, and query a training-data reference bank $\mathcal{B} = \{(e_j^{\text{real}}, a_j^{\text{real}})\}_j$. Representing actions by relative ego motion, we measure local state–action support (Yu et al., 2026) by

$$d_k = \frac{1}{J} \sum_{j \in \mathcal{N}_J(e_k, a_k)} \left(\frac{\|e_k - e_j^{\text{real}}\|_2^2}{b_e^2} + \frac{\|a_k - a_j^{\text{real}}\|_2^2}{b_a^2} \right), \qquad q_k^{\text{sup}} = e^{-d_k}, \tag{8}$$

where $\mathcal{N}_J$ contains the J nearest bank entries and $b_e, b_a > 0$ balance the two representations.

A.5 Reliability calibration and recurrent evaluation

Initial clip filtering. After world-model training, we generate observations under recorded ego trajectories and apply a reliability threshold of 0.80 to select initial clips for policy interaction. This retains 71.3% of `navtrain` clips and 65.3% of the held-out test clips. For 20 s generations under recorded ego motion, the same 0.80 filter accepts 65.8% of `navtrain` clips. The filter selects scenes that the world model can reproduce consistently before introducing policy-induced deviations.

Validation calibration. We calibrate the online threshold on the 1,771-clip validation split using 11 trajectory variants following the `navhard` construction (Cao et al., 2025). Starting from 0.80, we manually inspect generated clips with reliability scores within ± 0.01 of candidate thresholds to assess their suitability for continued rollout and policy training. This procedure selects $\tau = 0.74$ for interaction within the initially filtered pool. The stricter initial filter establishes scene suitability under recorded motion, while the online threshold accommodates policy-induced motion within these scenes. The 11 variants are used for validation calibration; subsequent rollout evaluation follows actions sampled from the policy.

Full-rollout acceptance. Evaluation uses the T-GRPO rollout collection procedure with group size $N = 8$ and 0.5 s replanning intervals. Accepted generated observations become history for subsequent policy decisions and generation. Full-rollout acceptance requires completion of the entire evaluation horizon and $\gamma_k \ge 0.74$ for every generated segment. Each sampled rollout contributes separately to the denominator, including rollouts stopped by diagnostic rejection. Over 4 s ($K = 8$ decisions), the acceptance rates are 90.3% on filtered `navtrain` clips and approximately 87.2% on filtered test clips. The test evaluation uses the frozen VLA SFT checkpoint and final DMD-trained world model with fixed diagnostic parameters, and the collected test rollouts are reserved for evaluation. Initial clip retention and full-rollout acceptance use their respective denominators: all clips before filtering and all attempted rollouts from retained clips.

B Policy implementation details

This appendix supplements the policy settings in Section 3 and the optimization method in Section 2.4.

Shared execution interface. The three policy families retain their observation encoders and action distributions. Each decision executes the leading $\Delta t = 0.5\,\mathrm{s}$ of its ego plan; $K = 8$ decisions form a $T = 4\,\mathrm{s}$ trajectory. Executed poses become framewise camera-coordinate actions for world-model conditioning (Section 2.1). These simulator decisions are distinct from internal token or denoising steps. Only executed prefixes update the persistent state and visual cache; future states and reward annotations remain internal to the scoring backend.

B.1 Policy instantiations

VLA: Alpamayo-R1. Our VLA uses Alpamayo-R1 (NVIDIA et al., 2025) and the discrete-trajectory-token pathway in NVIDIA's official RL recipe.[1] Given camera observations, ego-motion history, and navigation input, the VLM autoregressively samples reasoning followed by discrete trajectory tokens. The flow-matching action expert is absent from both RL rollout and update forward passes.

The fixed decoder deterministically reconstructs the continuous ego plan from trajectory-token IDs and ego-motion history, without VLM hidden states or the reasoning KV cache. The local reward $r_{i,k}$ scores this decoded $4\,\mathrm{s}$ plan; its leading $0.5\,\mathrm{s}$ segment is executed. Successive segments form τ_i, with scoring after diagnostic truncation defined in Appendix C.2.

In Equation 11, $y_{i,k,\ell}$ is a generated reasoning or trajectory token, and $c_{i,k,\ell}$ contains the recorded input and sampled prefix $y_{i,k,<\ell}$. The ratio $\rho_{i,k,\ell}$ compares the current and collection VLMs' probabilities of the same token under that conditioning. All generated tokens receive the decision's fused advantage $A_{i,k}$. Updates hold observations and token IDs fixed and recompute VLM representations under the current parameters. No continuous action-expert likelihood or flow-sampling noise enters the objective. Reference regularization uses conditional KL divergence to the frozen pre-RL VLM.

WAM: SimWAM. SimWAM (Zhao et al., 2026b) co-trains a Wan2.2-initialized video expert and a separate action diffusion transformer using joint flow matching. Both future-video and action tokens attend to the current observation, while an isolated attention mask prevents them from attending to each other. The policy therefore predicts a trajectory without generating future frames. Its NAVSIM configuration uses one front-camera image, ego state, and a navigation command to predict eight poses over 4 s. The released implementation retains the frozen video expert for current-observation conditioning; it omits future-video generation during policy inference and RL.[2]

The WAM SFT entries use the released supervised checkpoint (SimWAM Authors, 2026). The published RL stage applies FlowGRPO: a stochastic differential equation replaces the deterministic flow sampler, enabling trajectory exploration and likelihood-based updates against the NAVSIM PDM reward. This baseline optimizes action-expert LoRA adapters while keeping the remaining parameters fixed, using groups of trajectories from difficult logged scenes. Each group contains alternative plans conditioned on the same logged observation and scored with the PDM reward. DriveVerse supplies post-action observations for successive planning decisions, enabling EC feedback along the policy's own rollout. The official SimWAM release reports NAVSIM v2 results for its supervised checkpoint; our FlowGRPO entries evaluate the released RL checkpoint under the NAVSIM v2 protocol.

For the online update, $y_{i,k,\ell}$ is the next trajectory-denoising state, and $c_{i,k,\ell}$ includes the observation, sampler state, and flow time. We use Gaussian transition log-probabilities, clipped and averaged over trajectory coordinates, while retaining the policy's native reference penalty. Flow steps ℓ occur within one planning decision and remain distinct from simulator decisions k.

E2E: HydraMDP. We use HydraMDP (Li et al., 2024) as the base model for our E2E results, using the WorldEngine release Li et al. (2026).[3] The model selects a trajectory from a fixed vocabulary conditioned on camera observations, ego state, and navigation commands. The E2E SFT and baseline RL entries use the released HydraMDP SFT and WorldEngine RL checkpoints, respectively. The released WorldEngine RL checkpoint uses interactive 3D Gaussian Splatting simulation for reinforcement post-training.

[1] NVIDIA Alpamayo RL recipe and implementation.
[2] https://github.com/H-EmbodVis/SimWAM
[3] https://github.com/OpenDriveLab/WorldEngine

For online T-GRPO, the policy samples from a softmax distribution over trajectory candidates and uses the selected candidate's log-probability in Equation 11. Here, $y_{i,k,1}$ is the selected candidate and $L_{i,k} = 1$, since there is one categorical choice per planning decision. The corresponding continuous trajectory is passed to the shared execution interface. The online update uses the same plan and executed-trajectory rewards as the other policy families while retaining the E2E policy's reference penalty.

Initialization and benchmark provenance. Within each family, SFT denotes the supervised initialization. For VLA and WAM, GRPO or FlowGRPO denotes refinement using logged observations. Baseline training follows the selected open-source recipes; DriveVerse RL (T-GRPO) denotes interactive training with generated post-action observations and the adopted T-GRPO update.

C Policy Optimization and Driving Rewards

We apply T-GRPO (Chen et al., 2025) to driving with step- and trajectory-group advantages and a clipped policy objective. This section specifies the update and driving rewards for the policy interfaces in Appendix B.1.

C.1 Trajectory-wise Optimization

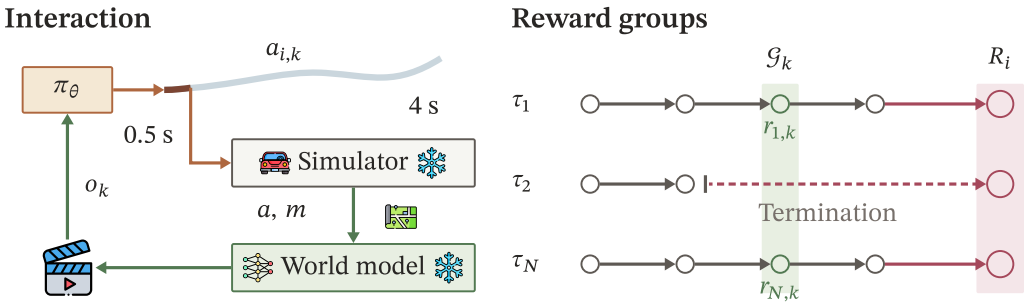


Figure 6: Driving interaction and reward grouping for the adopted T-GRPO update (Chen et al., 2025). The simulator executes 0.5 s before the policy replans; eight such steps form a 4 s rollout. Local rewards are grouped at corresponding decisions, and returns R_i are grouped across rollouts from the same initial scene. The illustrated termination denotes a diagnostic rejection: only that rollout stops, its executed prefix remains usable without a diagnostic penalty, and other instances continue. Driving infractions do not terminate collection.

Sampling and normalization. Within each initial-scene group, the N instances share initial observations, ego state, route, and background-traffic initialization; actions are sampled independently under π_{old}. Rollout τ_i contains $K_i \le K$ decisions, with $K_i = K$ unless online diagnostics truncate that instance. Driving failures leave collection running, so group members have equal lengths in the absence of diagnostic rejection. Let $\mathcal{G}_k = \{i : k < K_i\}$ contain the instances with an executed decision at time k. Using the local rewards $r_{i,k}$ and total returns R_i from Equation 5, the step and trajectory advantages are

$$A^{\text{step}}_{i,k} = \frac{r_{i,k} - \text{mean}_{j\in\mathcal{G}_k} r_{j,k}}{\text{std}_{j\in\mathcal{G}_k} r_{j,k} + \varepsilon_{\text{n}}}, \qquad A^{\text{traj}}_{i} = \frac{R_i - \text{mean}^{N}_{j=1} R_j}{\text{std}^{N}_{j=1} R_j + \varepsilon_{\text{n}}}. \tag{9}$$

Here, std is the sample standard deviation and $\varepsilon_{\text{n}} > 0$ provides numerical stability. Singleton and constant-reward groups receive zero advantage. Normalization is performed separately for each initial scene. Step groups align elapsed decision times; their members generally occupy different states after the initial decision, and terminated instances contribute no padded actions to these groups or to the policy loss. If $\mathcal{N}$ subtracts the group mean and normalizes by the group standard deviation, the fused advantage is

$$A_{i,k} = \alpha_{\text{s}} \mathcal{N}_{\mathcal{G}_k}(r_{i,k}) + \alpha_\tau \mathcal{N}_{\{1,\dots,N\}}(R_i), \qquad \alpha_{\text{s}} + \alpha_\tau = 1. \tag{10}$$

The weights are nonnegative. The trajectory advantage assigns the same full-horizon feedback to each valid decision in a rollout.

Incomplete action-chunk sequences. Every valid decision supplies a full $4\,\mathrm{s}$ plan. Step normalization uses $\mathcal{G}_k$, omitting missing decisions from rewards and likelihood terms. Appendix C.2 defines the full-horizon scoring trajectory and return. The policy loss averages over each rollout's K_i valid decisions, while other group members continue independently.

Clipped policy objective. Denote the policy-specific log-probabilities by $\log q_\theta(y_{i,k,\ell} \mid c_{i,k,\ell})$, where $y_{i,k,\ell}$ is a token, next denoising state, or trajectory candidate and $c_{i,k,\ell}$ is its conditioning. Our T-GRPO update uses $\rho_{i,k,\ell} = q_\theta(y_{i,k,\ell} \mid c_{i,k,\ell})/q_{\text{old}}(y_{i,k,\ell} \mid c_{i,k,\ell})$, with q_{old} fixed during each update. We maximize

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_i \frac{1}{K_i}\sum_k \frac{1}{L_{i,k}}\sum_\ell \min(\rho_{i,k,\ell}A_{i,k}, \mathrm{clip}(\rho_{i,k,\ell}, 1-\epsilon, 1+\epsilon)A_{i,k})\right] - \beta\Omega(\theta), \quad (11)$$

where $L_{i,k}$ is the number of update terms, ϵ is the clipping parameter, and β weights reference regularization. For the VLA, Ω is conditional KL divergence to the frozen pre-RL policy; the other policies retain their native reference penalties. Averaging over samples, decisions, and rollouts prevents length alone from determining an update's weight. Observations, rewards, and group statistics remain fixed, and the collection policy is refreshed between rounds.

The sums run over rollouts $i = 1, \ldots, N$, valid decisions $k = 0, \ldots, K_i - 1$, and likelihood terms $\ell = 1, \ldots, L_{i,k}$. The expectation is over initial scenes and rollouts collected with the sampling policy. All policy updates hold the observation generator and simulation backend fixed.

C.2 Driving rewards

EPDMS as the step-group reward. The local reward scores the full proposed plan in a separate scoring rollout initialized at the visited state, without advancing the persistent episode. With NAVSIM v2 subscores normalized to $[0, 1]$, we use

$$r_{i,k} = \left(\prod_{m\in\{\mathrm{NC,DAC,DDC,TLC}\}} m_{i,k}\right) \frac{5\mathrm{EP}_{i,k} + 5\mathrm{TTC}_{i,k} + 2\mathrm{LK}_{i,k} + 2\mathrm{HC}_{i,k} + 2\mathrm{EC}_{i,k}}{16}. \quad (12)$$

The multiplicative terms measure collision avoidance, drivable-area, driving-direction, and traffic-light compliance. The weighted terms measure progress, time-to-collision compliance, lane keeping, history comfort, and extended comfort. We retain NAVSIM's subscore thresholds and progress normalization against its reference planner, recomputed from the visited state. Thus, local progress is judged against the opportunities in that state rather than against a different instance's physical displacement.

HC includes the executed ego history at the start of the proposed plan. EC compares plans predicted at successive observations over their common future timestamps after alignment in the global frame, applying bounds to the root-mean-square differences in acceleration, jerk, yaw rate, and yaw acceleration. NAVSIM's one-stage evaluator pairs predictions at adjacent original logged observations to compute this term offline.[4] In our training loop, the later plan uses observations generated by DriveVerse after executing the earlier plan's leading segment. This supplies EC feedback on states reached by the policy's own actions. We assign this pairwise term to the later decision, starting from the second decision. At the first decision, we omit EC and renormalize the remaining weights by 14. During closed-loop training, we use unfiltered geometric subscores at states visited by the policy. Benchmark evaluation retains the official human-violation filtering and aggregation rules. These local plan scores supply $r_{i,k}$ in Equation 10; the final two-stage `navhard` score uses the benchmark aggregation.

Closed-loop score. The executed trajectory τ_i starts at the initial state and concatenates the $0.5\,\mathrm{s}$ prefixes committed at successive decisions. A complete rollout has $K = 8$ decisions and duration $T = 4\,\mathrm{s}$. We score it against the scene at the initial time with the same PDM backend used for local rewards. If online diagnostics truncate collection, we retain the executed prefix and complete the scoring horizon using the latest full plan, as specified below. Write $\bar{\tau}_i$ for this $4\,\mathrm{s}$ scoring trajectory; $\bar{\tau}_i = \tau_i$ for a complete rollout. Motivated by completion and safety scoring in HUGSIM

[4] NAVSIM one-stage scoring implementation (commit `0a380a9`).

and Bench2Drive (Zhou et al., 2024; Jia et al., 2024), we use $\mathrm{CLS}_i = \mathrm{RC}_i\mathrm{IP}_i\mathrm{DQ}_i$. The following definitions and coefficients specify our reward.

Completion. Let t_i^{crit} be the elapsed time of the first critical driving failure in $\bar{\tau}_i$: an at-fault collision, a drivable-area violation, or entry into oncoming traffic. Set $t_i^{\mathrm{crit}} = +\infty$ if no such failure occurs. Define

$$t_i^{\mathrm{fail}} = \min\{t_i^{\mathrm{crit}}, T\}, \qquad \mathrm{RC}_i = \frac{t_i^{\mathrm{fail}}}{T}, \qquad T = 4\,\mathrm{s}. \tag{13}$$

Thus, completion measures the fraction of the fixed horizon before a critical failure. A driving failure does not terminate simulation: all eight decisions are collected unless online diagnostics reject a generated segment. Infractions and driving quality are evaluated over the entire scoring trajectory, including motion after a driving failure.

Infraction penalty and stalling. Each infraction type contributes a binary indicator, applied once regardless of repeated occurrences:

$$\mathrm{IP}_i = 0.40^{C_i^{\mathrm{agent}}}\, 0.55^{C_i^{\mathrm{static}}}\, 0.65^{O_i}\, 0.70^{W_i}\, 0.70^{V_i}\, 0.30^{S_i}. \tag{14}$$

Here, C_i^{agent} indicates an at-fault collision with an agent, C_i^{static} a collision with a static object, O_i a drivable-area violation, W_i wrong-direction driving, V_i a red-light violation, and S_i a stall. Let $p(\tau)$ denote centerline progress and τ_i^{ref} the PDM-Closed reference (Dauner et al., 2023) generated from the same initial state over the same scoring interval. We declare a stall when

$$S_i = \mathbf{1}\left[p(\bar{\tau}_i) < 0.4\, p(\tau_i^{\mathrm{ref}})\right]. \tag{15}$$

This penalty discourages a policy from avoiding driving infractions by stopping when the reference makes progress.

Driving quality. We evaluate progress, time-to-collision compliance, lane keeping, and comfort on $\bar{\tau}_i$:

$$\mathrm{DQ}_i = \frac{3\mathrm{EP}_i + 4\mathrm{TTC}_i + 2\mathrm{LK}_i + 2\mathrm{COM}_i}{11}. \tag{16}$$

All four subscores lie in $[0, 1]$. Ego progress EP_i is measured relative to τ_i^{ref} and clipped to $[0, 1]$, using the PDM scorer's progress normalization. The other terms measure time-to-collision compliance, lane keeping, and comfort under the same scoring backend. Comfort is evaluated continuously across replanning boundaries, including the join to a plan-based tail when diagnostics truncate collection.

Reward scaling. The closed-loop reward is $r_i^{\mathrm{cl}} = \lambda(\mathrm{CLS}_i - 1) \in [-\lambda, 0]$. To avoid assigning smaller local returns solely to shorter collections, we scale each rollout's mean observed local reward to the common K-decision horizon:

$$U_i = \frac{K}{K_i} \sum_{k=0}^{K_i-1} r_{i,k}, \qquad R_i = U_i + r_i^{\mathrm{cl}}, \qquad 1 \le K_i \le K = 8. \tag{17}$$

For complete rollouts, U_i sums the eight local rewards; for incomplete rollouts, it scales the observed mean to the same horizon. We initialize $\lambda = 6$ and re-estimate it to equalize the within-scene-group standard deviations of the return terms:

$$\lambda = \frac{\mathrm{std}_{i=1}^{N}(U_i)}{\mathrm{std}_{i=1}^{N}(\mathrm{CLS}_i)}. \tag{18}$$

Re-estimation requires a nonzero denominator; groups with no closed-loop score variation do not provide a new scale estimate. The scale is shared by the group and held fixed during its policy update.

Full-horizon scoring after diagnostic termination. Diagnostic rejection after decision k stops instance i with $K_i = k + 1$ and $T_i = K_i \Delta t$, retaining its executed prefix and local rewards. The latest 4 s plan, sampled before rejection, completes $[T_i, T]$ in a separate scoring rollout. Denote this continuation from the last executed state by $\hat{\tau}_i^{\text{tail}}$. The PDM backend scores

$$\bar{\tau}_i = \begin{cases} \tau_i, & K_i = K, \\ \tau_i \oplus \hat{\tau}_i^{\text{tail}}, & K_i < K, \end{cases} \qquad \text{duration}(\bar{\tau}_i) = T, \tag{19}$$

where $\oplus$ appends planned motion to the executed prefix up to $t_0 + T$ on the scene clock. This scoring-only continuation adds no policy decisions, generated observations, or persistent-state transitions. Likelihoods use collected decisions; all incurred infractions remain. Every score component and the PDM-Closed reference share the 4 s horizon.

Tolerance of incomplete collection. Diagnostic rejection adds no penalty and is neither a driving infraction nor a stall. Equation 17 gives $U_i = Kr$ for constant observed reward r and any K_i. Equal observed reward means and closed-loop scores yield equal returns regardless of collection length. The scoring tail adds no likelihood terms or step-group samples, and group members stop independently.

D Additional Evaluations

Tables 6 and 7 report full component scores on `navhard` and `navtest` for our policies and published planners, using each released policy's sensor, training, and inference configuration.

Table 6: Full EPDMS comparisons on NAVSIM v2 `navhard`. S1/S2 denote evaluation stages and S. the stage score. For SimScale co-trained LTF, GTRS, and DiffusionDrive (Tian et al., 2026), ∗/† denote pseudo-expert/reward-only supervision on simulated data. PDM-Closed uses symbolic ground truth. Published scores retain metric versions before and after the official fix (Kirby et al., 2026); – denotes unreported scores.

Method	Stage	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	S. ↑	EPDMS ↑
Policy baselines and interactive RL with DriveVerse												
Alpamayo-R1 (2025) SFT	S1	97.6	93.1	99.3	99.6	83.3	96.2	97.1	97.6	77.8	81.7	38.1
	S2	84.6	76.4	87.5	98.7	84.6	80.6	48.7	97.4	68.9	45.7	
Alpamayo-R1 (2025) GRPO	S1	97.6	93.8	99.7	99.6	81.3	97.3	96.4	97.8	76.9	82.0	47.1
	S2	89.4	84.7	91.4	98.9	83.0	86.0	53.2	96.7	63.2	56.8	
Alpamayo-R1 (2025) DriveVerse RL (T-GRPO)	S1	98.0	95.8	99.6	99.9	80.9	98.0	97.1	97.8	85.8	87.6	**52.3**
	S2	90.2	84.7	91.1	98.9	81.9	86.8	53.3	96.4	68.6	59.6	
SimWAM (2026b) SFT	S1	97.8	92.0	99.6	99.6	83.8	96.2	98.0	97.8	72.4	80.6	37.3
	S2	81.8	78.4	87.3	98.4	86.3	78.6	49.1	96.3	68.8	46.0	
SimWAM (2026b) FlowGRPO	S1	95.6	95.3	99.1	99.3	86.6	94.9	92.4	96.7	11.1	74.0	31.2
	S2	81.9	79.4	83.9	98.3	89.9	77.9	47.4	94.2	9.7	41.0	
SimWAM (2026b) DriveVerse RL (T-GRPO)	S1	98.4	94.7	99.7	99.8	85.8	97.1	97.8	97.9	79.8	84.9	44.8
	S2	85.4	82.7	89.0	98.8	88.3	82.0	54.2	96.6	75.3	52.8	
HydraMDP (2024) SFT	S1	96.4	92.9	95.2	99.3	84.3	95.6	93.6	97.1	57.3	74.0	32.7
	S2	85.2	81.7	83.0	97.8	90.0	80.9	49.6	95.0	51.2	44.4	
HydraMDP (2024) WorldEngine RL	S1	96.2	90.4	93.7	99.3	81.1	96.0	91.8	94.4	39.1	67.9	33.0
	S2	89.1	82.2	82.7	98.1	87.0	84.7	50.6	93.0	35.7	46.9	
HydraMDP (2024) DriveVerse RL (T-GRPO)	S1	97.2	94.1	95.7	99.5	88.1	96.8	95.2	97.4	71.8	80.5	43.6
	S2	89.8	85.0	89.2	98.2	90.4	85.8	55.1	96.2	66.4	54.2	
Published baselines												
PDM-Closed (2023)	S1	94.4	98.8	99.9	99.5	99.9	93.5	99.3	87.7	36.0	–	51.3
	S2	88.1	90.6	96.3	98.5	99.9	83.1	73.7	91.5	25.4	–	
LTF* (2022)	S1	96.1	85.3	99.4	99.3	84.7	94.7	93.6	97.6	77.3	66.3	30.2
	S2	85.5	66.9	91.6	99.1	93.0	81.1	58.3	95.1	42.9	44.8	
GTRS-Dense (R34)† (2025d)	S1	97.6	96.4	99.3	99.9	75.7	97.8	93.3	97.3	32.9	72.4	46.9
	S2	94.3	92.7	95.1	99.5	80.2	91.5	56.2	90.6	28.3	63.4	
GTRS-Dense (V2-99)† (2025d)	S1	99.3	97.6	99.4	99.9	71.5	99.6	96.0	95.8	30.7	72.5	48.0
	S2	94.9	94.3	94.5	99.3	79.0	92.6	57.8	93.4	30.9	65.4	
DrivoR (2026)	S1	98.8	95.1	98.9	99.9	72.6	98.7	94.0	97.6	73.3	–	48.3
	S2	90.2	88.4	91.9	98.6	70.0	88.0	50.1	98.5	76.2	–	
DiffusionDrive* (2026)	S1	97.4	88.7	99.3	99.3	82.8	96.9	98.0	97.3	59.6	67.5	32.6
	S2	86.4	72.1	92.9	98.5	92.1	80.6	60.8	95.4	31.9	46.8	

Table 7: Full EPDMS comparisons on NAVSIM `navtest`. TransFuser, Hydra-MDP++, and DriveSuprim use the ResNet34 evaluations of Yao et al. (2025); DiffusionDrive and the ARTEMIS metric-column mapping follow Li et al. (2025c). Unreported scores are shown as –.

Method	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Policy baselines and interactive RL with DriveVerse										
Alpamayo-R1 (2025) SFT	98.9	97.0	99.7	99.8	87.0	98.5	96.5	98.3	84.5	89.3
Alpamayo-R1 (2025) GRPO	99.3	98.4	99.3	99.8	90.0	98.8	96.2	98.3	82.0	91.1
Alpamayo-R1 (2025) DriveVerse RL (T-GRPO)	99.4	98.5	99.3	99.8	93.4	98.8	96.4	98.3	86.0	**92.8**
SimWAM (2026b) SFT	98.6	98.1	99.7	99.9	87.4	98.1	97.9	98.4	84.7	90.3
SimWAM (2026b) FlowGRPO	98.3	98.5	99.5	99.6	90.1	97.5	91.0	95.4	11.0	82.1
SimWAM (2026b) DriveVerse RL (T-GRPO)	99.0	98.8	99.8	99.9	90.8	98.7	98.2	98.4	87.0	92.0
HydraMDP (2024) SFT	99.0	97.9	98.2	99.6	90.3	98.5	94.2	98.0	77.6	88.8
HydraMDP (2024) WorldEngine RL	98.6	97.5	97.0	99.4	89.6	97.9	93.1	96.9	63.7	85.2
HydraMDP (2024) DriveVerse RL (T-GRPO)	99.2	98.6	99.0	99.8	92.1	98.8	96.5	98.2	84.0	91.4
Other published baselines										
TransFuser (2022)	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
Hydra-MDP++ (2025a)	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim (2025)	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS (2025)	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	–	83.1
DriveVLA-W0 (2025c)	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
ExploreVLA (2026)	98.8	96.2	99.6	99.8	87.1	98.2	97.8	98.3	86.8	88.8
DiffusionDrive (2025)	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
DiffusionDriveV2 (2025)	97.7	96.6	99.2	99.8	88.9	97.2	96.0	97.8	91.0	85.5