DriveDreamer-Policy: A Geometry-Grounded World-Action Model for Unified Generation and Planning

Yang Zhou[1,2] **Xiaofeng Wang**[3] **Hao Shao**[4] **Letian Wang**[2]
Guosheng Zhao[3] **Jiangnan Shao**[1] **Jiagang Zhu**[1] **Tingdong Yu**[1]
Zheng Zhu[1,*] **Guan Huang**[1] **Steven L. Waslander**[2]

[1] GigaAI [2] University of Toronto [3] Tsinghua University [4] CUHK MMLab
Project Website: https://drivedreamer-policy.github.io/

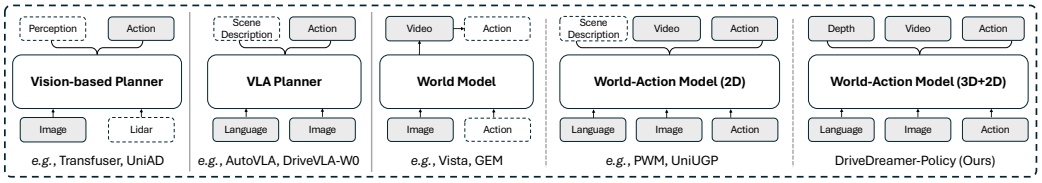


Figure 1: Comparison of our DriveDreamer-Policy with existing models. Items with dashed lines are optional. Vision-based and VLA planners directly map observations to actions without explicitly predicting the future world. World models generate future observations but often rely on external action signals. Recent world–action models unify future world generation and planning, but typically operate on image/video representations. DriveDreamer-Policy explicitly generates depth alongside video and actions, enabling geometry-grounded imagination and planning within a unified model.

Abstract

Recently, world-action models (WAM) have emerged to bridge vision-language-action (VLA) models and world models, unifying their reasoning and instruction-following capabilities and spatio-temporal world modeling. However, existing WAM approaches often focus on modeling 2D appearance or latent representations, with limited geometric grounding, an essential element for embodied systems operating in the physical world. We present DriveDreamer-Policy, a unified driving world-action model that integrates depth generation, future video generation, and motion planning within a single modular architecture. The model employs a large language model to process language instructions, multi-view images, and actions, followed by three lightweight generators that produce depth, future video, and actions. By learning a geometry-aware world representation and using it to guide both future prediction and planning within a unified framework, the proposed model produces more coherent imagined futures and more informed driving actions, while maintaining modularity and controllable cost. Experiments on the Navsim v1 and v2 benchmarks demonstrate that DriveDreamer-Policy achieves strong performance on both planning and world generation tasks. In particular, our model reaches 89.2 PDMS on Navsim v1, 88.7 EPDMS on `navtest`, and 30.7 EPDMS on `navhard` in Navsim v2, outperforming existing world-model-based approaches while producing higher-quality future video and depth predictions. Ablation studies further show that depth learning improves video imagination and planning robustness.

* Corresponding Author

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

1 Introduction

Autonomous driving systems have undergone a paradigm shift from handcrafted modular stacks [51] to end-to-end learning systems [6, 16], and more recently to vision-language-action (VLA) models built on large language models (LLMs) [4, 62, 14, 32] that offer richer common-sense knowledge, instruction following, and reasoning capabilities [74, 20, 77, 24]. Despite these advances, most VLA planners primarily optimize action outputs and do not explicitly model how the future world may evolve under alternative actions, which limits interpretability and can reduce reliability in rare or safety-critical situations where forward-looking reasoning about occlusions and hidden hazards is essential. In parallel, driving world models have emerged as a powerful direction for learning spatio-temporal traffic dynamics directly from large-scale sensor logs [13, 47, 15, 39, 21, 5, 76, 70, 41]. By forecasting future observations (*e.g.*, videos, BEV features, or other state representations), they enable scalable simulation, rare-event evaluation, and controllable synthetic data generation. Motivated by the complementary strengths of these two lines, recent world-action models aim to unify future world generation with motion prediction or planning [65, 56, 37, 71], thereby bridging imagination and decision-making in a single framework.

However, in many existing world-action models, the world component is still implemented as image/video prediction or latent rollouts without explicit geometric grounding, and the benefits of imagined futures for action prediction can be constrained by representation mismatch or tightly coupled module designs. As a result, the generated world may be visually plausible but not maximally informative for planning, and the planner may not consistently benefit from structured safety cues such as geometric layout and free-space constraints.

A key observation motivating this work (Fig. 1) is that autonomous driving is fundamentally a 4D physical process: 3D geometry evolves over time. Consequently, an actionable world model should not only synthesize appearance, but also preserve geometric structure that is essential for occlusion reasoning, distance estimation, and physically consistent motion. Depth-centric modeling is particularly attractive here: depth is compact, directly tied to geometry, and can serve as an explicit representation that constrains future image/video generation and informs planning decisions. Moreover, recent progress in depth foundation models [60, 29, 45, 44, 57] suggests that high-fidelity depth can be produced off-the-shelf, without collecting extra data or training a depth estimator from scratch. These developments suggest an opportunity to more effectively drive world-action models: explicitly generating depth representations and studying how this benefits both future video generation and motion planning within a unified architecture.

To this end, we propose **DriveDreamer-Policy**, a unified driving world-action model that jointly generates 1) a depth-based 3D geometric representation of the current scene, 2) action-conditioned future videos, and 3) future trajectories for planning. The system is built on a large language model for perception and reasoning, producing a compact set of world embeddings and action embeddings. These embeddings serve as conditions for the multimodal generator: a pixel-space depth generator, a latent-space video generator, and an action generator. Importantly, we impose a structured causal attention mask across query groups in a depth$\rightarrow$video$\rightarrow$action manner: video queries may consume depth context, and action queries may consume both depth and video context. This yields a simple, single-pass information flow, while enabling video imagination to benefit from 3D understanding and allow planning to leverage both 3D structure and predicted future-world context.

Our contributions are threefold. 1) We introduce DriveDreamer-Policy, a unified, modular world-action architecture for autonomous driving that combines an LLM with generative experts connected through a fixed-size query interface, enabling practical compute control. 2) We incorporate an explicit 3D depth generation module and utilize a causal 3D$\rightarrow$2D$\rightarrow$1D conditioning pathway, allowing geometry to benefit future video generation and motion planning and supporting multimodal data generation/simulation. (3) We conduct comprehensive experiments and analyses on Navsim [10, 7], evaluating both planning performance and world generation quality, achieving state-of-the-art compared to existing world-based models. For instance, our model achieves an EPDMS score of 88.7 (+2.6 over the previous method) and an FVD of 53.59 (-32.36 over the previous method), improving both planning performance and video generation quality. We also perform targeted ablations to quantify how depth and future imagination jointly contribute to planning robustness.

2 Related Works

2.1 Driving World-Action Models

Driving-focused generative models use sensor data, such as images, to generate future videos, thereby enabling scalable data synthesis and simulation [54, 13, 15, 1, 39, 27, 5, 42, 69, 40, 38, 68, 50]. Recently, driving world action models that unify generation and planning have emerged as an active frontier. Epona [65] introduces an autoregressive diffusion world model that decouples causal temporal latents from per-step diffusion generation to support long-horizon video rollout and trajectory planning. ReSim [59] trains a diffusion-transformer world simulator on real logs plus simulator non-expert behaviors to improve action-following reliability and adds Video2Reward for reward estimation. DriveVLA-W0 [24] combats the VLA supervision deficit by adding future-image world modeling and using a lightweight MoE action expert for lower latency. PWM [71] treats a unified autoregressive transformer as a Policy World Model that performs action-free future forecasting and collaborative state-action prediction to benefit planning. DriveLaW [56] unifies planning and generation by feeding video-generator latents into a diffusion trajectory planner to align imagined futures with control. OmniNWM [21] jointly generates panoramic RGB, semantics, depth, and 3D occupancy, conditions trajectories via Plucker ray-maps, and derives intrinsic occupancy-based dense rewards. UniUGP [37] unifies understanding, video generation, and trajectory planning by integrating a pretrained VLM with a video generator through hybrid experts. Similar to existing world-action models, DriveDreamer-Policy uses an LLM to model driving-world knowledge, serving as the perception module. To incorporate multimodal generators, fixed-size latent queries are used as cross-attention keys, enabling generative experts to jointly predict depth, video, and action.

2.2 Driving Vision-Language-Action Models

The use of VLMs in autonomous driving has gradually evolved from scene interpretation to direct action generation. Early works such as DriveGPT4 [58] primarily used LLMs/VLMs to generate scene descriptions and high-level maneuver suggestions, thereby improving interpretability. To bridge language understanding and low-level control, a line of modular language-to-action frameworks [74, 2] introduced multi-stage pipelines that pass intermediate textual commands between modules; however, these non-differentiable interfaces hinder end-to-end optimization by blocking gradient backpropagation across perception, reasoning, and control. More recent end-to-end VLA methods [61, 64, 49] adopt unified architectures that directly map sensor observations to trajectories. Within this paradigm, DriveMoE [61] introduces a Mixture-of-Experts (MoE) design with an action decoder attached after a VLM, ReCogDrive [26] couples a VLM with a diffusion planner trained by imitation and reinforcement learning to better align semantic reasoning and control, and AutoVLA [77] discretizes trajectories into action primitives so that a single autoregressive model can jointly learn adaptive reasoning and planning. DriveVLA-W0 [24] scales driving VLA learning by adding dense world modeling (future image prediction) as supervision and introducing a lightweight MoE action expert to reduce inference cost and improve latency. Our method is similar to this end-to-end VLA direction and further extends it with a unified world-action modeling design. DriveDreamer-Policy follows driving VLA models but replaces image-token world prediction with a diffusion video generation head, enabling controllable action-conditioned video generation. We further add a depth-based 3D representation to resolve missing-geometry grounding. While prior works [67, 72] have explored depth supervision for VLAs, they treat depth as an independent auxiliary task (*e.g.*, disentangled modeling). In contrast, our method utilizes depth as a shared representation to benefit both video generation and planning in a structured information flow.

2.3 3D Representation and Learning in Autonomous Driving

Early planners such as UniAD [16] and VAD [19] incorporate 3D representation and learning into the planning pipeline and often use 3D perception (e.g., 3D detection, occupancy, or mapping) as auxiliary tasks that implicitly require depth. GaussianAD [73] represents the scene with 3D semantic Gaussians and plans with a future-forecasting objective; DriveTransformer [18] lets agent, map, and planning queries interact directly with raw sensor features; HiP-AD [48] uses multi-granularity planning queries and trajectory geometry to retrieve image features; UniFuture [27] is a 4D driving world model that jointly predicts future RGB and depth by mapping both modalities into a shared latent space; and ORION [12] combines an LLM for reasoning with a generative planner by aligning

the reasoning and action spaces. Our work differs in how and why geometry is used. First, these methods use 3D representations mainly for perception and planning, whereas our depth is a shared representation that simultaneously constrains future video generation and informs planning. Second, most of them are planners without future generation, while we jointly produce depth, future video, and actions. In particular, although UniFuture entangles depth and video at the UNet latent level, we use an LLM with modular experts organized by a causal depth-video-action query interface. ORION also couples an LLM with a generative planner, it does not explicitly model geometry or generate future appearance, whereas we introduce depth learning to ground both generation and planning.

3 Methodology

We propose DriveDreamer-Policy, a unified driving world–action model that couples a large language model with lightweight generative experts to jointly support 1) 3D world representation, 2) 2D world generation, and 3) motion planning. We introduce world understanding with large language models and world-action prediction using generative experts in Sec. 3.1 and Sec. 3.2, then provide training details in Sec. 3.3.

3.1 Preliminaries

Flow Matching. We use conditional flow matching [33, 35, 31, 30] as the training principle for our generative experts when predicting continuous targets. Let $x \in \mathbb{R}^d$ denote a target variable, and let c denote conditioning information. Flow matching learns a time-dependent velocity field $v_\theta(x_t, t \mid c)$ that transports a simple base distribution toward the data distribution along a predefined path. We adopt the standard linear interpolation path between a data sample $x_0 \sim p_{\text{data}}$ and a noise sample $x_1 \sim p_{\text{noise}}$:

$$x_t = (1-t)\, x_0 + t\, x_1, \quad t \sim \mathcal{U}(0,1), \tag{1}$$

with the target velocity given by $\dot{x}_t = x_1 - x_0$. The objective is a regression loss on the velocity:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x_0, x_1, t}\left[\| v_\theta(x_t, t \mid c) - (x_1 - x_0) \|_2^2 \right]. \tag{2}$$

At inference, we sample $x_1 \sim p_{\text{noise}}$ and integrate the induced ODE backward from $t{=}1$ to $t{=}0$ to obtain a sample consistent with the conditioning c.

3.2 DriveDreamer-Policy

Our overall pipeline is illustrated in Fig. 2. Multi-view images, language instructions, and the action are first encoded as tokens and processed by an LLM, together with a compact set of learned world and action queries. The resulting world embeddings and action embeddings serve as a geometry-aware interface that conditions three modular experts: a depth generator, a video generator, and an action generator. The LLM is responsible for multimodal understanding and producing compact state representations, while the experts generate modality-specific outputs (depth, video, and action, respectively), all mediated by a fixed-size query bottleneck. This design is motivated by the complementary strengths of the two components: the LLM provides stable semantics and strong contextual reasoning, whereas generative experts better capture multi-modality and uncertainty in long-horizon prediction. As a result, the model can operate in multiple modes: planning-only (enabling only the action expert), imagination-enabled planning (running action plus depth/video generation when needed), or full generation for offline simulation and data synthesis.

3.2.1 World Understanding

Input Processing. At each decision step, the model takes as input a natural-language instruction and synchronized multi-view RGB observations. We also provide the current action as context to the LLM, thereby contributing to both world modeling and planning. We tokenize the inputs into three streams. First, the instruction is converted into standard text tokens using the LLM's tokenizer. Second, each camera view is encoded by the vision encoder into a sequence of visual patch tokens. Third, the action context is embedded into a set of action tokens using a lightweight action encoder. Finally, we append three fixed-size groups of learnable query tokens—depth queries, video queries, and action queries—in that order. This design yields a stable, compact interface: the backbone always consumes the same set of query slots, and downstream heads can read out the corresponding query embeddings to produce depth maps, future videos, and future actions.

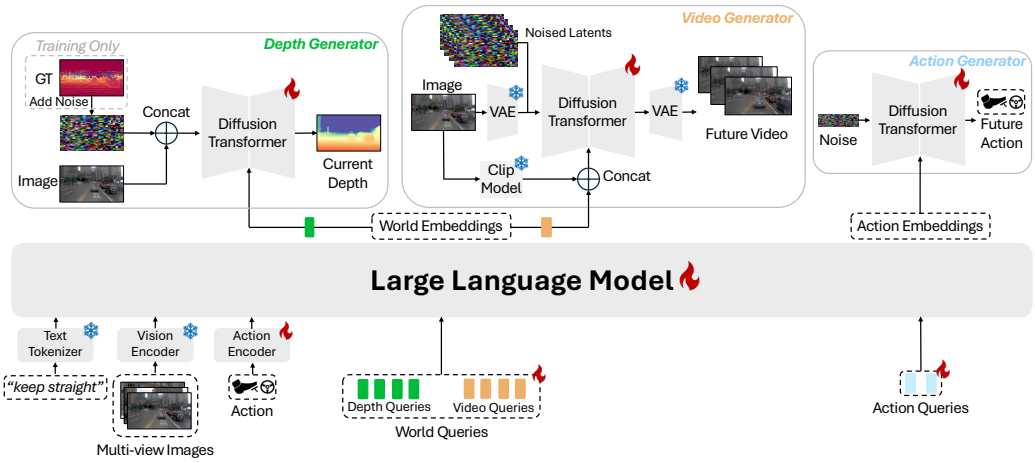


Figure 2: Overview of our DriveDreamer-Policy pipeline. The large language model takes the language instruction, multi-view images and current action, along with a set of learnable queries as inputs to reason and generate world and action embeddings. The generated embeddings are then passed into our three generative expert models as cross-attention conditions to generate depth, future images, and future action.

Embeddings Generation. All tokens are processed by the LLM to produce contextualized hidden states. We impose a causal ordering across query groups: depth queries come first, video queries can attend to depth context, and action queries can attend to both depth and video context. Concretely, within the same time step the attention pattern satisfies depth queries→video queries → action queries. This structured mask provides a clean, single-pass information flow without extra synchronization or iterative refinement across branches.

3.2.2 World and Action Prediction

Depth Generator. We generate a monocular depth map as an explicit 3D representation for the world action model. Depth is not only a geometric output for visualization: it provides a compact representation that is directly useful for downstream video imagination (*e.g.*, occlusions and object boundaries) and action planning (*e.g.*, free space and distance-to-collision cues). Our model predicts depth with a generative objective rather than a purely deterministic regression head, thereby better capturing the inherent ambiguity of monocular depth and preserving sharp depth discontinuities. Generating depth directly in pixel-space is practical here because depth has much lower dimensionality than RGB video and preserves boundary fidelity without requiring an additional learned codec.

As shown in the upper-left of Fig. 2, our depth generator is a pixel-space diffusion transformer trained with a standard flow-matching objective. During training, we sample a continuous flow time and corrupt the ground-truth depth with noise; the denoiser takes as input the concatenation of the noisy depth and the corresponding RGB image, and predicts the denoising update. To ground generation in global scene semantics, we condition the depth denoiser on the LLM world depth embeddings through cross-attention: the depth-query world embedding acts as a compact global representation (keys/values) that guides the diffusion transformer to maintain global structural consistency. This makes depth a queryable modality in DriveDreamer-Policy: the predicted depth can be generated on demand with depth embeddings, and the depth embeddings serve as an upstream geometric feature that later query groups (video/action) can attend to.

Video Generator. For future video generation, we employ a text-image-to-video diffusion transformer [43, 53] (see the upper-middle of Fig. 2). Given the current RGB images, we first encode them into a compact latent representation using a VAE and initialize a sequence of noisy video latents for the target horizon. Instead of conditioning the diffusion model on text embedding (as in standard text-to-video pipelines), we condition it on the LLM world video embeddings produced by the video query tokens. These world embedding tokens summarize language intent, multi-view perception, and action context, and extrinsically incorporate upstream geometric cues from the depth queries.

The video denoiser attends to the world embeddings at each transformer block via cross-attention. To preserve appearance, identity, and camera content, we additionally extract a lightweight visual condition from the current image frame using a CLIP [46] model and inject it into the denoiser as an explicit conditioning signal, concatenated with the world video embeddings as shown in our pipeline. This design keeps the generator tightly grounded in the current scene and commanded maneuver, while enabling controllable, action-aware video generation.

Action Generator. As presented in the upper-right of Fig. 2, the action generator is implemented as a standalone diffusion transformer that maps a noise trajectory to a feasible future action sequence. It is conditioned on the action embedding produced by the LLM from the action query tokens, which aggregates instruction semantics, multi-view observations, and upstream geometric and imagination cues. This conditioning is injected via cross-attention, thereby keeping the action head lightweight while still leveraging rich scene context. Since the action generator does not depend on explicit depth and video generation, it can be activated independently for planning, while implicitly benefiting from predicted future world context. We parameterize each trajectory state by position and heading using a continuous representation [75], namely $(x, y, \cos\theta, \sin\theta)$, which avoids angular wrap-around and encourages smooth turn dynamics.

3.3 Training Details

Depth Normalization. We normalize depth to a stable range before training the depth generator. Given a depth map, we first apply a log transform and then compute per-map percentiles to normalize to the range [-0.5, 0.5]. During inference, we invert the transform to recover metric or relative depth as needed.

Model Initialization and Adaptation. For the large language model, we use Qwen3-VL-2B [4] to process and understand multimodal inputs. For the depth generator, we initialize our model from PPD [57]. For the video generator, we initialize the model from Wan-2.1-T2V-1.3B [53] and adapt it to the image-to-video task. For both depth and video generation, we fine-tune at a spatial resolution of 144×256 to reduce computational and memory costs. The video training horizon is 9 frames.

Training Objective and Optimization. We train all components in a single stage with a joint multi-task loss:

$$\mathcal{L} = \lambda_d \mathcal{L}_d + \lambda_v \mathcal{L}_v + \lambda_a \mathcal{L}_a, \tag{3}$$

where $\mathcal{L}_d$ is the loss for depth prediction, $\mathcal{L}_v$ is the loss for video prediction, and $\mathcal{L}_a$ is the loss for trajectory prediction. We use $\lambda_d = 0.1$ and set the remaining hyperparameters to 1.0 by default. The depth label used in training is obtained from an off-the-shelf depth foundation model, Depth Anything 3 (DA3) [29]. The detailed study of using DA3 for depth labels, rather than sensor-level ground truth (e.g., lidar-derived depth), is presented in Appendix E.

4 Experiments

4.1 Experimental Setup

We train and evaluate on the Navsim benchmark [10, 7], using the `navtrain` (100k samples) and `navtest` (12k samples) splits at 2 Hz. Planning is measured by the predictive driver model score (PDMS) on Navsim v1 and the extended PDMS (EPDMS) on Navsim v2. For world generation, we report depth accuracy of the absolute relative error (AbsRel) and threshold accuracy (δ) with video quality of Learned Perceptual Image Patch Similarity (LPIPS) [66], Peak Signal-to-Noise Ratio (PSNR) [17], and Fréchet Video Distance (FVD) [52]. We compare against three families of baselines: vision-based end-to-end planners, VLA planners, and world-model-based planners, all under their official Navsim results. We train in a single stage for 100k steps on 8 NVIDIA H20 GPUs with AdamW [36] (learning rate 1×10^{-5}, batch size 32), using 64 depth, 64 video, and 8 action query tokens. No additional datasets or extra pre-training beyond the initialized backbones are used. See Appendix A for all details of datasets and planning metrics, world generation metrics, baselines, and implementation details.

4.2 Quantitative Results

Planning Performance Comparison with Other Methods. As shown in Table 1, Table 2 and Table 3, we first report planning performance compared with other methods on the `navtest` set of Navsim v1 using PDMS and `navtest` and `navhard` sets of Navsim v2 using EPDMS. For a fair and meaningful comparison, we consider published methods. We note that AutoVLA and RecogDrive additionally employ RL post-training with PDMS as the reward, and we report their IL results for a more comparable evaluation setup. Specifically, we categorize comparisons against state-of-the-art vision-based end-to-end planners, vision-language-action planners, and world-model-based models. Our DriveDreamer-Policy outperforms all considered methods and achieves scores of 89.2 PDMS and 88.7 EPDMS on `navtest` in Navsim v1 and v2 and 30.7 EPDMS on `navhard` in Navsim v2, respectively. Moreover, DriveDreamer-Policy achieves strong performance on specific planning-critic sub-scores, indicating our method's control of planning trajectories (*e.g.*, a DAC of 97.1 and an EP of 83.5 on Navsim v1, a DDC of 99.5 and an LK of 97.6 on `navtest` in Navsim v2, and an EP of 85.1 of stage 1 and 89.8 in stage 2 on `navhard` in Navsim v2).

World Performance Comparison with Other Methods. We also compare with other generative world-based methods. For video quality, we compare with the existing method PWM [71]. To the best of our knowledge, only PWM was open-sourced and conducted video quality evaluation on Navsim at the time of this paper's submission. Since PWM supports only single-view (front-view) generation, we evaluate the single-view quality for a fair and convenient comparison. We follow the metrics used in PWM and report the results in Table 4(a). Our method shows a larger improvement than PWM by a substantial margin (*e.g.*, improvement of 32.36 on FVD). At Navsim's 2 Hz sampling rate, the generation horizon of 9 frames covers a 4-second future horizon, which aligns with the standard planning evaluation window. Our strong FVD score confirms temporal consistency within this horizon. For depth accuracy, we compare against PPD because our depth generator is initialized from it, depth prediction serves more as a representation in our method, and there is no wide-scale depth prediction on Navsim. Specifically, we evaluate two variants: 1) zero-shot and 2) fine-tune PPD on Navsim. As shown in Table 4(b), DriveDreamer-Policy achieves lower depth error. We attribute the improvement to LLM conditioning: the depth denoiser is guided by world depth embeddings, resolving locally ambiguous regions and producing more consistent geometry than image cues alone.

Ablations on World Learning for Planning. To isolate the contribution of each modality, we evaluate four variants under identical training budgets: 1) action-only: the simplest VLA version without any world modeling, 2) depth+action: generates depth and future action, 3) video+action: generates video and future action, and 4) depth+video+action: our full model. We report improvements in PDMS for each variant. The results are shown in Table 5. All world training strategies effectively improve planning performance compared to training from scratch. However, single-world training (depth or video) yields smaller improvement, likely due to incomplete world information. Joint world training, leveraging both depth geometry and video temporal evolution, achieves the largest performance gains by enabling the model to learn more generalizable and robust features.

To further ablate our modular generator design, we conducted an additional experiment: world-conditioned action generation. We inject detached final-layer features from both the depth and video generators (using unnoised inputs during training) into the action generator via cross-attention, enabling it to leverage fine-grained geometric and temporal cues from the world generators. The results are presented in the last row of Table 5. By allowing information fusion between generators, planning achieves a PDMS of 89.9. However, the inference cost is at least three times that of modular generators, since all three generators must be fully activated. Our modular world learning yields a planning improvement of 1.2 PDMS, representing a trade-off between performance and inference cost. We restrict information exchange to the LLM's query interface to decouple world generation from planning, supporting flexible application and usage (*e.g.*, planning-only as an action model, or full generation as a world model).

4.3 Ablation Studies

Ablations on Depth Learning for Video Generation. The impact of depth learning on future video generation is evaluated by comparing two variants under the same training data and compute budget: 1) video-only, where the video generator is conditioned on the backbone features without depth joint learning, serves as an approximation of the disentangled modeling in [67]; 2) depth+video, where

Table 1: Comparison with state-of-the-art methods on the `navtest` of Navsim v1 [10] benchmark. The best result outlines as bold. "*" indicates results with imitation learning (IL). Methods fall into three categories: vision-based E2E models, VLA-based models, and world-model-based models.

Methods	Venue	Sensors	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Human	–	–	100.0	100.0	100.0	99.9	87.5	94.8
Vision-Based End-to-End Methods								
TransFuser [9]	TPAMI'23	3×C + L	97.7	92.8	92.8	**100.0**	79.2	84.0
UniAD [16]	CVPR'23	6×C	97.8	91.9	92.9	**100.0**	78.8	83.4
PARA-Drive [55]	CVPR'24	6×C	97.9	92.4	93.0	99.8	79.3	84.0
DiffusionDrive [28]	CVPR'25	3×C + L	98.2	96.2	94.7	**100.0**	82.2	88.1
Vision-Language-Action Methods								
AutoVLA* [77]	NeurIPS'25	3×C	96.9	94.4	88.0	99.9	75.8	80.5
Recogdrive* [26]	ICLR'26	3×C	98.1	94.7	94.2	**100.0**	80.9	86.5
DriveVLA-W0 [24]	ICLR'26	1×C	98.7	96.2	95.5	**100.0**	82.2	88.4
World-Model-Based Methods								
LAW [23]	ICLR'25	1×C	96.4	95.4	88.7	99.9	81.7	84.6
DrivingGPT [8]	ICCV'25	1×C	**98.9**	90.7	94.9	95.6	79.7	82.4
WoTE [25]	ICCV'25	3×C + L	98.5	96.8	94.4	99.9	81.9	88.3
Epona [65]	ICCV'25	3×C	97.9	95.1	93.8	99.9	80.4	86.2
FSDrive [64]	NeurIPS'25	3×C	98.2	93.8	93.3	99.9	80.1	85.1
PWM [71]	NeurIPS'25	1×C	98.6	95.9	95.4	**100.0**	81.8	88.1
DriveDreamer-Policy (Ours)	–	3×C	98.4	**97.1**	95.1	**100.0**	**83.5**	**89.2**

Table 2: Comparison with state-of-the-art methods on the `navtest` of Navsim v2 [7].

Methods	Venue	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Vision-Based End-to-End Methods											
TransFuser [9]	TPAMI'23	96.9	89.9	97.8	99.7	87.1	95.4	92.7	**98.3**	87.2	76.7
DiffusionDrive [28]	CVPR'25	98.2	95.9	99.4	99.8	87.5	97.3	96.8	**98.3**	87.7	84.5
Drivesuprim [63]	AAAI'26	97.5	96.5	99.4	99.6	**88.4**	96.6	95.5	**98.3**	77.0	83.1
ARTEMIS [11]	RAL'26	98.3	95.1	98.6	99.8	81.5	97.4	96.5	**98.3**	**89.1**	83.1
Vision-Language-Action Methods											
DriveVLA-W0 [24]	ICLR'26	**98.5**	**99.1**	98.0	99.7	86.4	**98.1**	93.2	97.9	58.9	86.1
World-Model-Based Methods											
DriveDreamer-Policy (Ours)	–	98.4	97.1	**99.5**	**99.9**	87.9	97.7	**97.6**	**98.3**	79.4	**88.7**

Table 3: Comparison with state-of-the-art methods on `navhard` of Navsim v2.

Methods	Venue	Stage	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Vision-Based End-to-End Methods												
GuideFlow [34]	CVPR'26	S1	96.6	80.5	96.3	99.3	82.3	94.9	91.5	97.7	67.8	27.1
		S2	**87.3**	**76.7**	**88.8**	**99.2**	84.3	**85.1**	**49.7**	93.1	44.5	
DiffusionDrive [28]	CVPR'25	S1	**96.8**	86.0	98.8	99.3	84.0	**95.8**	**96.7**	97.6	**79.6**	27.5
		S2	80.1	72.8	84.4	98.4	85.9	76.6	46.4	96.3	**72.8**	
Vision-Language-Action Methods												
DriveVLA-W0 [24]	ICLR'26	S1	**96.8**	83.3	99.0	99.6	84.6	95.3	96.4	97.6	78.2	24.4
		S2	76.8	64.3	79.9	98.3	89.2	75.0	46.8	95.8	53.1	
SGDrive [22]	CVPR'26	S1	95.8	**87.6**	97.8	**99.8**	84.4	94.7	92.9	**97.8**	28.9	25.5
		S2	79.4	65.4	79.1	98.9	88.9	75.3	42.7	96.4	29.6	
ReCogDrive [26]	ICLR'26	S1	96.4	78.9	98.7	**99.8**	82.6	95.6	94.4	97.6	74.2	25.7
		S2	80.2	65.0	82.4	98.7	85.2	76.9	43.8	**96.6**	71.8	
World-Model-Based Methods												
DriveDreamer-Policy (Ours)	–	S1	96.3	87.3	**99.7**	**99.8**	**85.1**	95.1	96.2	97.6	66.2	**30.7**
		S2	80.3	72.7	85.3	98.9	**89.8**	76.1	47.0	96.3	58.3	

Table 4: World generation performance on Navsim. For video generation, we compare against existing generative world-model methods trained and evaluated on Navsim. For depth generation, we report comparisons among our model variants. Our method achieves higher quality for depth and video.

(a) Video performance comparison.

Methods	Venue	LPIPS↓	PSNR↑	FVD↓
PWM [71]	NeurIPS'25	0.23	**21.57**	85.95
DriveDreamer-Policy (Ours)	–	**0.20**	21.05	**53.59**

(b) Depth performance comparison.

Methods	Venue	AbsRel↓	δ_1↑	δ_2↑	δ_3↑
PPD [57]	NeurIPS'25	18.5	80.4	94.0	97.2
PPD-Finetuned	NeurIPS'25	9.3	91.4	98.3	**99.5**
DriveDreamer-Policy (Ours)	–	**8.1**	**92.8**	**98.6**	**99.5**

depth is trained jointly and the video queries are causally conditioned on depth queries. Video quality is reported in Table 6. Joint learning with depth improves video generation accuracy, indicating that depth provides an effective 3D representation for coherent future prediction.

Table 5: Ablations on world learning for planning. All world learning strategies effectively improve planning performance compared to training from scratch. "*" indicates an experiment where world-generator features are additionally fused into the action generator. For details, see Sec. 4.2.

Strategy	World Representation		Planning Metrics					
	Depth	Video	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Without World Learning	✗	✗	98.0	96.3	94.4	**100.0**	82.5	88.0
With World Learning	✔	✗	98.1	96.7	94.9	**100.0**	82.8	88.5
	✗	✔	98.1	97.0	95.0	**100.0**	83.1	88.9
	✔	✔	**98.4**	**97.1**	**95.1**	**100.0**	**83.5**	**89.2**
World Condition*	✔	✔	98.4	97.4	95.4	100.0	84.2	89.9

Table 6: Ablations on depth learning for video generation. Using depth as a prior in joint learning improves video generation accuracy.

Strategy	World Representation	Video Metrics		
	Depth	LPIPS↓	PSNR↑	FVD↓
Without Depth Learning	✗	0.22	19.89	65.82
With Depth Learning	✔	**0.20**	**21.05**	**53.59**

Table 7: Ablations on depth modeling. Learnable depth query with depth as the model's output achieves stronger planning performance compared to depth as the model input.

Depth Source	Planning Metrics					
	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Frozen DA3 depth as image input	97.9	96.5	94.4	**100.0**	82.5	88.2
Frozen DA3 feature via adapter	97.7	96.0	94.4	99.9	82.1	87.6
Learnable depth query + generator (Ours)	**98.1**	**96.7**	**94.9**	**100.0**	**82.8**	**88.5**

Ablations on Depth Modeling. We explored two training settings of injecting frozen DA3 depth: 1) we render the DA3 depth as an image and feed it as an additional image input to the LLM; 2) we extract the DA3's 128-channel decoder features before the depth prediction layer and inject them through a lightweight pooling + linear adapter, at the position of the depth query in the LLM input. The planning results on Navsim are shown in Table 7. Both variants underperformed our design of a learnable depth query with an attached depth generator that learns the DA3 depth. Injecting frozen DA3 depth as an external input degrades planning. We attribute this to a mismatch: frozen DA3 depth is purely geometric and decoupled from the LLM's scene understanding. As an isolated input, it does not align with, and can interfere with, the LLM's representation. In contrast, our learnable depth query makes depth an internal representation coupled with driving-scene semantics, which is more useful for downstream planning. On inference cost, our modular design discards the depth generator in planning-only mode, so it adds only the lightweight world-query computation and no depth forward pass at inference. Both frozen DA3 variants instead require running DA3 at inference, and the depth-as-image variant additionally increases the token count substantially, raising cost. Our design is therefore both stronger in planning and cheaper at inference.

Ablation on Different LLMs and Action Modeling. As presented in Table 8, our variant method without world learning (VLA version) achieves competitive planning performance on Navsim, with a PDMS of 88.0. To better understand what drives the competitive performance of our base VLA architecture, we conduct further ablations. Our VLA version includes an action encoder for the LLM and uses action queries for the action generator (DiT). To understand 1) the effectiveness of different LLMs and 2) the mechanism of action modeling, we conduct four variants of our VLA model: 1) ours with Qwen2-VL-2B; 2) Qwen3-VL-2B with MLP action decoder; 3) ours without the action encoder; 4) ours without the action generator, replaced with an MLP action decoder. The results are listed in Table 8. For different LLMs, we show that Qwen2-VL-2B with our action modeling achieves slightly lower performance than Qwen3-VL-2B (PDMS of 87.6 vs. 88.0), due to weaker model capabilities. Both LLMs with 2B parameters perform well on Navsim. For action modeling, we find that combining an action encoder with the LLM and the DiT generator yields the best performance.

Table 8: Ablations on different LLMs and action modeling. LLMs with identical parameters achieve comparable results, whereas action modeling improves planning performance by a larger margin.

LLM	Action Modeling		Planning Metrics					
	With Action Encoder	Action Generator	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
Qwen2-VL-2B	✔	DiT	97.8	96.0	94.3	99.9	82.1	87.6
Qwen3-VL-2B	✗	MLP	97.9	95.1	94.0	**100.0**	80.2	86.2
Qwen3-VL-2B	✗	DiT	97.6	95.6	93.2	99.9	81.9	86.8
Qwen3-VL-2B	✔	MLP	**98.0**	95.4	94.0	99.9	81.8	87.1
Qwen3-VL-2B	✔	DiT	**98.0**	**96.3**	**94.4**	**100.0**	**82.5**	**88.0**

Table 9: Ablations on the number of queries. More query tokens provide higher-capacity slots for storing relevant context, thereby enhancing both generation and planning.

Number of Queries			Metrics											
Depth	Video	Action	Depth				Video			Action					
			AbsRel↓	δ_1↑	δ_2↑	δ_3↑	LPIPS↓	PSNR↑	FVD↓	NC↑	DAC↑	TTC↑	C↑	EP↑	PDMS↑
32	32	4	9.7	90.2	97.9	99.4	**0.20**	20.67	57.97	98.2	97.0	95.0	**100.0**	83.2	88.9
64	64	8	**8.1**	**92.8**	**98.6**	**99.5**	**0.20**	**21.05**	**53.59**	**98.4**	**97.1**	**95.1**	**100.0**	**83.5**	**89.2**

Directly fine-tuning the LLM with an MLP action decoder leads to relatively poor planning (*e.g.*, a PDMS of 86.2). These results demonstrate the importance of our action modeling in LLMs for action learning, resulting in a 1.8 improvement in PDMS from 86.2 to 88.0. Notably, from the MLP baseline (86.2) to our full model (89.2), the total improvement is 3.0 PDMS, of which action modeling contributes 1.8 and world learning contributes the remaining 1.2. The latter gain is on top of an optimized VLA pipeline, where marginal improvements are harder to achieve.

Ablations on Number of Queries. We ablate the query budget to study the trade-off between accuracy. Specifically, we compare the default setting (64 depth + 64 video + 8 action query tokens) against smaller budgets (32 depth + 32 video + 4 action query tokens). The results, presented in Table 9, indicate that increasing the query budget generally improves both planning and world-generation performance, as more query tokens provide higher-capacity slots to store geometry, appearance, and action-relevant context.

4.4 Qualitative Results

We present some visualization results of world generation and motion planning in Appendix B.1, overlaying predicted trajectories on BEV renderings and showing the corresponding generated depth maps and future frames. These results demonstrate the model's spatially stable generation ability and well-aligned planning trajectories. We also show some additional results to show the effectiveness of world learning in benefiting planning in Appendix B.2, categorized into four distinct ablation variants: 1) action-only, 2) depth-action, 3) video-action, and 4) depth-video-action. When combined with world learning, the model consistently produces safer and more human-like trajectories.

5 Conclusion

We present DriveDreamer-Policy, a unified driving world–action model that jointly performs depth generation, future video imagination, and motion planning within a single framework. The model combines a large language model with modular generative experts connected through a compact query interface, enabling flexible operating modes for both planning and world generation. By introducing depth as an explicit geometric representation and organizing information flow in a depth→video→action manner, the model allows planning to leverage both scene geometry and predicted future dynamics. Experiments on Navsim demonstrate strong performance across planning and world generation tasks, and ablations show that depth and video provide complementary cues that improve planning robustness.

Acknowledgments and Disclosure of Funding

Part of the work was done and supported during Yang Zhou's internship at GigaAI.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1933–1943, 2025.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[4] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025.

[5] Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loick Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, Renaud Marlet, Alexandre Boulch, Mickael Chen, Éloi Zablocki, Andrei Bursuc, Eduardo Valle, and Matthieu Cord. Vavim and vavam: Autonomous driving through video generative modeling, 2025.

[6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.

[7] 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, 2025.

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

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

[10] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking, 2024.

[11] 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*, 11(1):226–233, 2026.

[12] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24823–24834. IEEE, 2025.

[13] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

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

[15] 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, 2024.

[16] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving, 2023.

[17] Q Huynh-Thu and M Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.*, 44(13):800, 2008.

[18] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2025, pages 67227–67243, 2025.

[19] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8306–8316. IEEE, 2023.

[20] 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.

[21] Bohan Li, Zhuang Ma, Dalong Du, Baorui Peng, Zhujin Liang, Zhenqiang Liu, Chao Ma, Yueming Jin, Hao Zhao, Wenjun Zeng, and Xin Jin. Omninwm: Omniscient driving navigation world models, 2025.

[22] Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. *arXiv preprint arXiv:2601.05640*, 2026.

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

[24] 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, 2025.

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

[26] 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, 2025.

[27] Dingkang Liang, Dingyuan Zhang, Xin Zhou, Sifan Tu, Tianrui Feng, Xiaofan Li, Yumeng Zhang, Mingyang Du, Xiao Tan, and Xiang Bai. Seeing the future, perceiving the future: A unified driving world model for future generation and perception, 2025.

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

[29] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025.

[30] Yexiong Lin, Jia Shi, Shanshan Ye, Wanyu Wang, Yu Yao, and Tongliang Liu. Subflow: Sub-mode conditioned flow matching for diverse one-step generation. *arXiv preprint arXiv:2604.12273*, 2026.

[31] Yexiong Lin, Yu Yao, Yang Zhou, and Tongliang Liu. Beyond optimal transport: Model-aligned coupling for flow matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3955–3964, 2026.

[32] Yexiong Lin, Shanshan Ye, Yu Yao, Zhen Fang, Bo Han, and Tongliang Liu. Squidagent: Parallelize wisely, coordinate efficiently. *arXiv preprint arXiv:2610.08647*, 2026.

[33] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[34] 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*, pages 3719–3728, 2026.

[35] 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.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[37] Hao Lu, Ziyang Liu, Guangfeng Jiang, Yuanfei Luo, Sheng Chen, Yangang Zhang, and Ying-Cong Chen. Uniugp: Unifying understanding, generation, and planing for end-to-end autonomous driving, 2025.

[38] Yuhang Lu, Yichen Yao, Jiadong Tu, Jiangnan Shao, Yuexin Ma, and Xinge Zhu. Can lvlms obtain a driver's license? a benchmark towards reliable agi for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5838–5846, 2025.

[39] Arian Mousakhan, Sudhanshu Mittal, Silvio Galesso, Karim Farid, and Thomas Brox. Orbis: Overcoming challenges of long-horizon prediction in driving world models, 2025.

[40] Chaojun Ni, Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Wenkang Qin, Guan Huang, Chen Liu, Yuyin Chen, Yida Wang, Xueyang Zhang, et al. Recondreamer: Crafting world models for driving scene reconstruction via online restoration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1560–1569. IEEE, 2025.

[41] Chaojun Ni, Xinyu Zhou, Yukun Zhou, Jingyu Liu, Xiaofeng Wang, Zheng Zhu, Yang Wang, Qiuping Deng, Yun Ye, Hao Li, et al. Gigaworld-policy: An efficient action-centered world–action model. In *European Conference on Computer Vision*, pages 315–333. Springer, 2026.

[42] NVIDIA, :, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao

Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2025.

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.

[44] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2354–2367, March 2026.

[45] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation, 2024.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[47] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving, 2025.

[48] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 25605–25615. IEEE, 2025.

[49] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025.

[50] GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025.

[51] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, September 2006.

[52] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric challenges, 2019.

[53] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025.

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

[55] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15449–15458, 2024.

[56] Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Drivelaw:unifying planning and video generation in a latent driving world, 2025.

[57] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Sida Peng, and Xin Yang. Pixel-perfect depth with semantics-prompted diffusion transformers, 2025.

[58] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

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

[60] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2, 2024.

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

[62] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[63] 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, 2025.

[64] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving, 2025.

[65] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[66] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[67] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge, 2025.

[68] Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Xueyang Zhang, Yida Wang, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, et al. Drivedreamer4d: World models are effective data machines for 4d driving scene representation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12015–12026. IEEE, 2025.

[69] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10412–10420, 2025.

[70] Guosheng Zhao, Yaozeng Wang, Xiaofeng Wang, Zheng Zhu, Yu Tingdong, Guan Huang, Yongchen Zai, Ji Jiao, Changliang Xue, Xiaole Wang, et al. Unidrivedreamer: A single-stage multimodal world model for autonomous driving. In *European Conference on Computer Vision*, pages 360–377. Springer, 2026.

[71] Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction, 2025.

[72] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model, 2024.

[73] Wenzhao Zheng, Junjie Wu, Yao Zheng, Sicheng Zuo, Zixun Xie, Longchao Yang, Yong Pan, Zhihui Hao, Peng Jia, Xianpeng Lang, et al. Gaussianad: Gaussian-centric end-to-end autonomous driving. *arXiv preprint arXiv:2412.10371*, 2024.

[74] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model, 2025.

[75] Yang Zhou, Hao Shao, Letian Wang, Steven L Waslander, Hongsheng Li, and Yu Liu. Smartrefine: A scenario-adaptive refinement framework for efficient motion prediction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15281–15290. IEEE, 2024.

[76] Yang Zhou, Hao Shao, Letian Wang, Zhuofan Zong, Hongsheng Li, and Steven Waslander. Drivinggen: A comprehensive benchmark for generative video world models in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pages 103502–103524, 2026.

[77] 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.

A Experimental Setup

Datasets and Planning Metrics. We train and evaluate our method on the Navsim benchmark [10, 7], which is derived from real-world driving logs and provides synchronized surround-view sensory inputs for end-to-end planning evaluation. Following the standard Navsim protocol, we train on the `navtrain` split and evaluate on the `navtest` split, which contain 100k and 12k data samples sampled at 2Hz, respectively. We also evaluate on the NAVSIM v2 navhard split using two-stage pseudo-simulation, with 450 original observations in Stage 1 and 5,462 synthetic observations in Stage 2. Navsim evaluates closed-loop planning performance using the predictive driver model score (PDMS) on v1 and the extended PDMS (EPDMS) on v2. PDMS aggregates multiple safety and quality terms, including no-at-fault collision, drivable-area compliance, time-to-collision, ego progress, and comfort; EPDMS further includes direction and traffic-light compliance, as well as lane-keeping and comfort. For Navsim v2, we align with the common practice in recent methods [28, 24] and evaluate EPDMS on the `navtest` split for fair and convenient comparison.

World Generation Metrics. In addition to planning, we evaluate our generative experts. Video is evaluated on Navsim using the recorded future RGB frames as ground truth. Depth is evaluated using dense depth targets provided by DA3, which are also used for training. We report Absolute Relative Error (AbsRel) to quantify relative depth differences and threshold accuracy (δ) to measure the proportion of accurate predictions within a specified relative error. Higher δ values and lower AbsRel values indicate better depth estimation performance. For video evaluation, we follow [71] and report perceptual quality and temporal consistency of predicted future frames using Learned Perceptual Image Patch Similarity (LPIPS) [66], Peak Signal-to-Noise Ratio (PSNR) [17] and Fréchet Video Distance (FVD) [52].

Baselines. We compare against strong Navsim baselines spanning three families. 1) vision-based end-to-end planners that utilize vision models and map sensor inputs to trajectories, including TransFuser [9], UniAD [16], PARA-Drive [55], DiffusionDrive [28], Drivesuperim [63], ARTEMIS [11] and GuideFlow [34]. 2) Vision-language-action planners that use large language models and predict trajectories as tokens or by a diffusion-based expert, including approaches: DriveVLA-W0 [24], AutoVLA [77], Recogdrive [26] and SGDrive [22]. 3) World-model-based planners that integrate foresight into planning, including LaW [23], DrivingGPT [8], WoTE [25], Epona [65], FSDrive [64] and PWM [71]. All baselines are reported under their official Navsim performance.

Implementation Details. We implement the action encoder as a 2-layer MLP with layer normalization [3]. We train DriveDreamer-Policy in a single stage for 100k steps with a batch size of 32 on 8 NVIDIA H20 GPUs, using the AdamW optimizer [36] with a learning rate of 1×10^{-5}. Unless otherwise stated, all experiments use the same query configuration (64 depth-query tokens, 64 video-query tokens, and 8 action-query tokens). We use Navsim training data, without additional datasets or extra pre-training beyond the initialized backbones.

B Qualitative Results

B.1 Visualizations of World Generation and Motion Planning

We visualize representative Navsim scenarios in Fig. 3 by overlaying predicted trajectories on BEV renderings and by showing the corresponding generated depth maps and future frames. These examples highlight how depth-conditioned imagination reduces common failure modes (*e.g.*, short-horizon collision risk) and improves interpretability by exposing the model's predicted geometry and future scene evolution.

B.2 Visualizations of World Learning for Planning

We present additional visualization results to demonstrate the effectiveness of world learning in benefiting planning. Fig. 4 presents qualitative comparisons that demonstrate how learned world representations improve planning. Each row visualizes a representative Navsim scenario, and each column corresponds to an ablation variant: Action-Only, Depth-Action, Video-Action, and Depth-Video-Action. Across the three cases, adding world learning consistently produces safer and more

human-like trajectories. In the top-row example, action-only, depth-, and video-planning tend to lead to conflicting traffic, whereas full depth-video-conditioned planning maintains a clearer safety margin. In the middle row example, world learning helps recover the correct maneuver early, reducing lateral deviation and preventing late, abrupt turns. In the bottom-row example, the full model yields trajectories that more closely match the expert path, suggesting that geometry cues (depth) and future appearance/dynamics cues (video) provide complementary guidance for action prediction.

C Inference Efficiency Discussion

In planning-only mode, DriveDreamer-Policy activates only the 2B-parameter LLM and the lightweight action DiT, discarding the depth and video generators entirely. Compared to DriveVLA-W0, which relies on a 7B LLM, our model has a significantly smaller inference footprint while achieving stronger planning performance (89.2 vs. 88.4 PDMS on Navsim v1 and 88.7 vs. 86.1 EPDMS on Navsim v2). This advantage stems from our world-learning design: depth and video generation during training enrich the LLM's representations with geometric and temporal cues, enabling a compact backbone to achieve competitive results at lower inference cost. When the full generation mode is activated for world simulation or data synthesis, the additional depth and video generators increase inference cost by at least $3\times$.

The three generative experts are all DiTs: the action DiT has 538M parameters, the depth DiT has 506M, and the video DiT has 1564M. In planning-only mode, only the 2B LLM and the 538M action DiT are activated, while the depth and video DiTs (2070M in total) are discarded. In planning-only mode, end-to-end planning latency is about 260ms on a single H20 GPU (including the LLM forward pass and action generation), using 10 sampling steps for the action DiT. As for the Training cost, full three-modality training takes about 3 to 4 days on 8 NVIDIA H20 GPUs.

D Limitations and Future Work

Reinforcement Post-Training. As mentioned in Sec. 4, the two VLA-based methods (AutoVLA and RecogDrive) considered include post-training with reinforcement learning (RL). They achieve a PDMS of 89.1 and 90.8 by RL on Navsim, respectively. These methods use reinforcement learning with PDMS as the reward signal, auxiliary training on VQA/reasoning, and multiple external datasets, thereby incurring significantly higher training costs. In our proposed method, reinforcement learning is not included and marked as a limitation for two reasons: 1) difficulty in jointly optimizing multimodal output, and 2) high cost for training video models. With three generators to consider, post-training convergence is difficult with different reward signals. Furthermore, the cost (*e.g.*, GPU memory and training time) of video post-training is substantial. We believe RL post-training is an interesting and promising direction for world-action models, which requires both world generation and action improvement.

Generator Initialization. Our depth and video generators are initialized from PPD and Wan-2.1, both of which are general-purpose models with no driving-specific knowledge and therefore require adaptation during joint training. We use loss weighting (Eq. 3) to balance the multiple objectives. The two generators differ in their sensitivity to initialization. The depth generator is relatively small and can be trained from scratch on Navsim. The video generator, however, is much larger and requires large-scale video pretraining to converge, partly due to the VAE reconstruction bottleneck, which makes from-scratch training infeasible under our compute budget. Different initializations may also require re-tuning loss weights, which we leave to future work. A natural concern is whether strong pretrained generators might make the LLM backbone redundant for learning driving-relevant features. Our results suggest otherwise: the jointly trained depth generator outperforms fine-tuned PPD alone (Table 4b), showing that the LLM's world embeddings contribute to scene understanding that the pretrained generator does not capture by itself.

Larger Query Budgets. In Table 9, performance improves when using more query tokens. We leave the exploration of larger query budgets (*e.g.*, 128+) to future work due to limited compute resources, and the current 64-token configuration already achieves strong results. More query tokens enhance planning performance but increase training and inference costs, and diminishing returns may be observed beyond certain levels.

Multi-View Stereo. Our depth generator produces depth for three views and does incorporate multi-view information implicitly: the three-view RGB inputs are all visible to the depth query within the LLM, so the generated depth is conditioned on multi-view context rather than a single image. However, the proposed method does not perform explicit multi-view stereo. Explicit multi-view stereo (e.g., cross-view geometric matching) is a promising direction to better utilize 3D information.

Dynamic Reasoning of Other Agents. Our input uses multi-view images at the current timestamp. We note that future motion is inherently multi-modal, and this ambiguity is a property of the prediction task itself rather than a consequence of omitting history; even with history, the future may remain multi-modal. Adding history frames is a natural extension our framework can accommodate, and we leave it to future work to further improve future dynamic reasoning.

E Additional Experiments of DA3 Depth Labels

E.1 Motivation and Effect for Choosing DA3 Depth

We use an off-the-shelf depth foundation model (DA3) mainly for generality and scalability, as also discussed in our introduction: strong depth foundation models make dense depth readily available on data without ground-truth depth, such as large-scale web driving videos or synthetic data. LiDAR provides a valuable independent reference, but Navsim LiDAR is sparse and unevenly distributed, which makes it less suitable as a dense training target, so we train on dense pseudo-labels.

Both training and evaluation use DA3 pseudo-labels, the AbsRel/δ numbers in Table 4(b) reflect consistency with DA3 rather than absolute physical accuracy. We do not intend to claim sensor-level geometric precision. As stated in our paper, depth serves primarily as a representation that constrains video generation and informs planning, so its value is demonstrated by the downstream gains in Tables 5 and 6 rather than by the depth accuracy numbers themselves.

We trained an early experiment of the depth+action variant using LiDAR-projected ground-truth depth as the training label, instead of DA3 pseudo-labels, under the same setting. The planning result is essentially the same:

Depth label for training	PDMS↑
DA3 pseudo-label (Table 5)	88.5
LiDAR-projected GT depth	88.6

Since replacing DA3 pseudo-labels with independent LiDAR-derived depth leaves planning essentially unchanged, the downstream benefit does not depend on the absolute accuracy of the depth labels. This is consistent with our view that depth acts as a geometric representation rather than a precise output. This comparison focuses on planning, as the video generator is expensive to train and was not included in this early study.

E.2 The Quality of DA3 Depth Labels

To directly address the concern about pseudo-label bias, we evaluated both our predicted depth and DA3 pseudo-labels against LiDAR-derived depth on Navsim:

Depth source	AbsRel↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
DA3 pseudo-label	14.9	83.7	93.4	96.9
Ours	15.8	81.8	93.5	97.0

Our predicted depth is very close to the DA3 pseudo-labels when both are evaluated against LiDAR, indicating that our model faithfully learns the geometry of the depth foundation model without introducing additional systematic bias; the remaining gap to LiDAR reflects the inherent accuracy of the depth foundation model rather than our modeling.

These results indicate that the pseudo-label supervision from DA3 has limited effect on both the learned geometry and downstream planning.

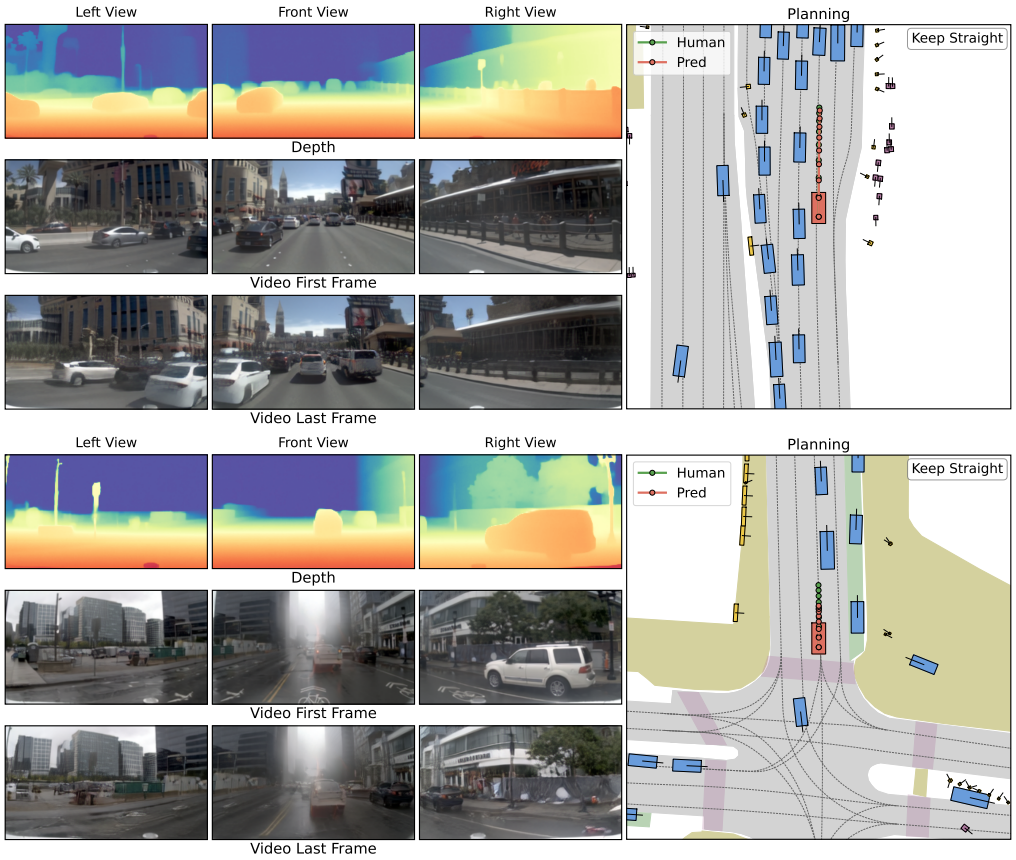


Figure 3: Visualization Results of our method. We show the generated depth, video, and actions, respectively. Depth is truncated to below 80 meters for better visualization. Our generation results remain spatially stable, and the planning performs well compared with human trajectories (*e.g.*, aligns with human trajectories (top) and slows down more effectively than human trajectories (bottom)).

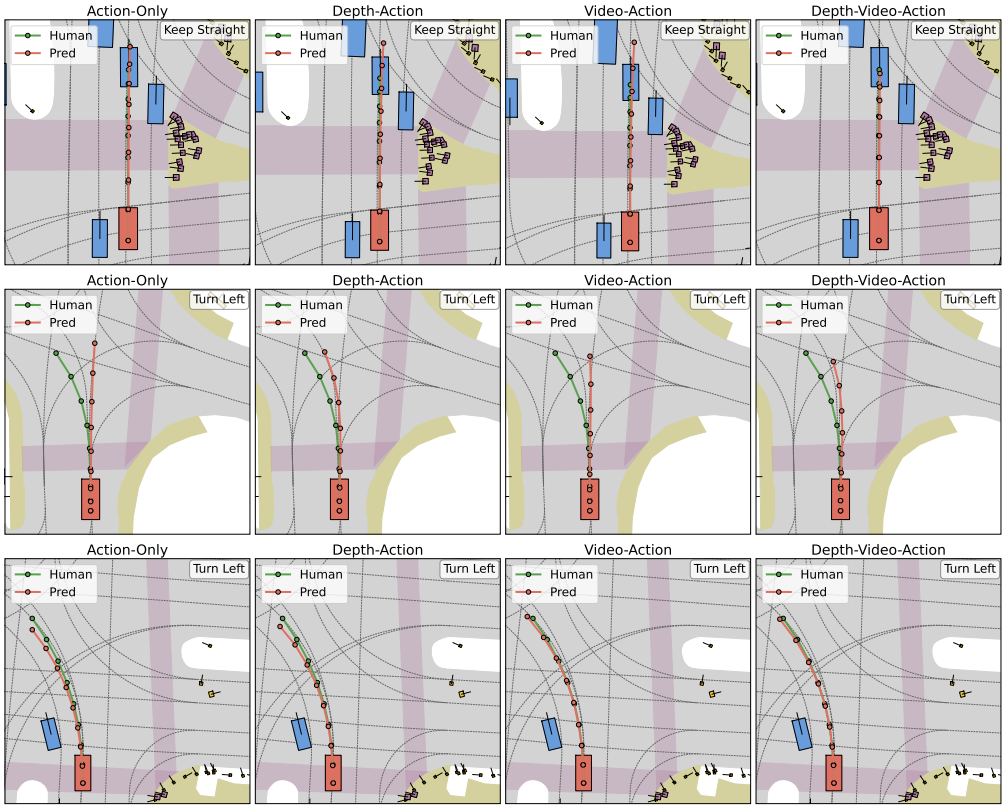


Figure 4: Visualization of world learning for planning. Columns compare Action-Only, Depth-Action, Video-Action, and Depth-Video-Action variants. Green denotes the human (expert) trajectory and red denotes the predicted trajectory. The three rows correspond to (top) avoiding potential collision by a slower trajectory, (middle) correcting an initially wrong maneuver, and (bottom) aligning more closely with the human trajectory. Depth and video provide complementary world cues that improve safety margins and trajectory consistency.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the contributions and scope of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: The limitations of the proposed method are discussed in the appendix.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: This paper does not include formal theoretical results or proofs.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide training details, plus code to reproduce our main results.

   Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: We have released our code and model weights on the project website.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All training and evaluation details are provided in the main paper and supplementary material.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the high computational cost of training and evaluating models, we do not report error bars. However, we follow standard evaluation protocols consistent with prior work.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the type and number of GPUs used in the main text.

   Guidelines:

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

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed and conform to the NeurIPS Code of Ethics in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper mainly discusses potential positive impacts: improving the generation (driving data/traffic simulation) and planning (safety) performance of autonomous driving systems.

    Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: Our work focuses on autonomous driving policy learning in simulation and does not pose significant risks for direct misuse.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We cite all datasets and codebases used in this work and respect their respective licenses.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The new assets (code and models) introduced in this paper have been released with proper documentation and license information.

Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve human subjects research and thus does not require IRB approval.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

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

    Guidelines:

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