Orbis 2: A Hierarchical World Model for Driving

Sudhanshu Mittal* **Arian Mousakhan*** **Silvio Galesso***

Karim Farid **Johannes Dienert** **Rajat Sahay** **Thomas Brox**

University of Freiburg, Germany

Abstract

Current world models operate at a single level of abstraction, with most prioritizing perceptual fidelity while lacking the spatial reasoning and semantic understanding required for real-world downstream tasks. We present a hierarchical driving world model that factorizes future prediction across two levels operating at distinct temporal and abstraction scales: a high-level predictor that forecasts coarse scene structure over extended temporal horizons, and a low-level generator that produces detailed predictions conditioned on the high-level output. This decomposition yields high perceptual fidelity while also capturing strong spatial and semantic representations. We further show that pretraining with a diffusion forcing objective yields substantially richer internal representations than the standard teacher forcing objective, while teacher forcing — predicting only the next frame from clean context — produces more stable autoregressive rollouts. We therefore introduce a generic two-stage training paradigm that pretrains the model with diffusion forcing and fine-tunes with teacher forcing, combining the representational benefits of the former with the rollout stability of the latter. Our approach achieves state-of-the-art results across the standard suite of driving world model evaluations on established benchmarks, including long-horizon generation fidelity, steering responsiveness evaluated on counterfactual scenarios, and internal representation quality. Project page with code, demo, checkpoints and qualitative results: https://lmb-freiburg.github.io/orbis2.github.io

1 Introduction

World models [15, 17, 31] are learned dynamical systems that approximate how an environment evolves in response to an agent's actions, providing internal predictions of transition dynamics that support planning, simulation, and policy learning [49]. In high-dimensional, complex domains such as autonomous driving, world models are commonly formulated in a learned latent space where an encoder maps raw observations to compact latent representations and a predictor propagates these states forward, conditioned on the agent's actions. Current implementations of this paradigm are limited in two key respects. First, the predictor operates at a single temporal scale [3, 2]: rollouts are generated frame-by-frame, accumulating error over time and limiting the model's ability to explicitly reason about long-horizon scene evolution. Second, the latent space is typically shaped by a reconstruction objective that preserves all fine visual details such as texture and appearance, even though long-horizon dynamics are primarily governed by coarse, slowly varying semantic properties, making such representations suboptimal for long-horizon forecasting. This aligns with a well-established principle in cognitive science: humans plan over distant goals using simplified,

*Main Contributors. {mittal,mousakha,galessos}@cs.uni-freiburg.de

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

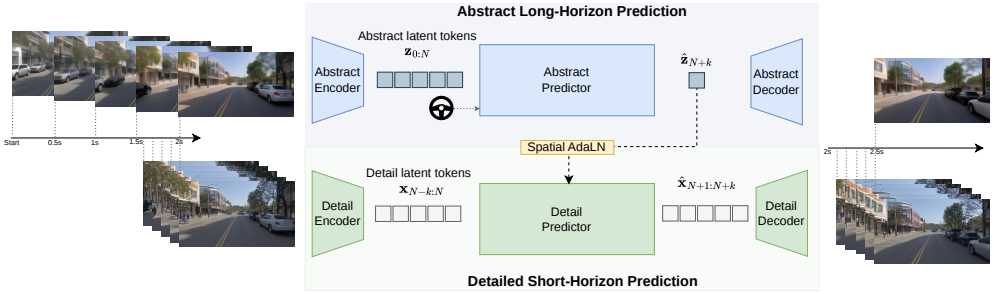


Figure 1: **Orbis 2: a hierarchical driving world model.** The *abstract predictor* operates over a long temporal context to forecast a future state in latent space, capturing abstract scene dynamics over long horizons, and enabling steering control. The *detail predictor* is conditioned on this abstract prediction and generates fine-grained short-horizon frames, enabling high-fidelity local prediction grounded in long-range temporal context.

abstract representations and refine details only as actions approach execution [51, 22]. Building on this principle, we propose a setup that decomposes future prediction across two levels of abstraction. A high-level predictor forecasts long-horizon dynamics in an abstract latent space, producing subgoals that condition a low-level predictor. The latter operates on a finer-grained latent space, specializing in detailed, short-horizon prediction. Figure 1 provides an overview of our hierarchical framework.

For the high-level predictor, we require a latent space that strongly captures spatial structure and semantic understanding, enabling long-horizon temporal estimation. We therefore utilize a compressed DINO [40] representation as the target for our high-level predictor. Since directly applying flow matching [34] in the high-dimensional space is challenging [61, 30], we introduce a lightweight projection that compresses DINOv2 features into a lower-dimensional latent space, which stabilizes training and improves long-horizon rollout quality, while retaining the representation properties of DINO. The goal of the low-level predictor is then to capture fine-grained details and precise motion dynamics, conditioned on the high-level prediction. This is naturally suited to pixel-reconstruction based latent spaces such as VAEs [29], which preserve the fine-grained visual detail needed for short-horizon, high-fidelity generation.

To implement the two predictors, we rely on flow matching [34], which was shown to be particularly effective for long-horizon prediction in driving world models [36]. As the major objective of world models is to generalize to unseen scenarios observed previously only in different contexts, an important indicator of the quality of a world model is the quality of its learned internal representations. Driving world models have typically been evaluated with FVD [12, 1, 36], which measures the visual fidelity of generated frames. The high-level predictor operating in an abstract latent space, however, is not expected to reproduce fine-grained appearance, even though it may capture the scene structure that matters for long-horizon prediction. Thus, we additionally evaluate the quality of the internal representations of both predictors via linear probing on semantic segmentation and depth estimation, which directly measures how well the learned latent space encodes scene structure and geometry. In the proposed hierarchical world model, both predictors can play to their advantages: the abstract predictor learns higher-quality structural representations and dynamics, whereas the low-level predictor can satisfy the requirements of high-fidelity, short-horizon roll-outs with strong FVD scores. By probing the internal representations of the model, we find that the standard teacher forcing recipe — conditioning on clean ground-truth context to predict the target frame — fails to learn strong internal representations, as the model never observes out-of-distribution context during training. Pretraining with a *diffusion forcing* objective [6], which independently corrupts and denoises all frames in the sequence, produces substantially richer internal representations. We summarize our contributions as follows:

A hierarchical driving world model. We introduce a two-level world model that decomposes future prediction into a high-level predictor operating in an abstract latent space and a low-level predictor producing high-fidelity frames.

A two-stage training paradigm for predictive representation learning. We propose pretraining with a diffusion forcing objective followed by short fine-tuning with teacher forcing. This combina-

tion enables self-supervised representation learning directly from the temporal prediction loss and substantially improves representation quality over only next frame prediction objective.

State-of-the-art results on driving world modeling. As shown in Fig. 2, our model achieves state-of-the-art performance on video-based driving world modeling across complementary axes: long-horizon perceptual fidelity (FVD) and representation quality (linear probing for semantic segmentation and depth). Our model also outperforms prior methods in steering responsiveness on counterfactual scenarios.

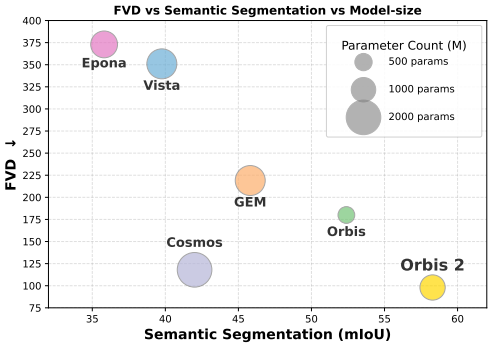


Figure 2: Comparison of driving world models over model scale (parameters), FVD (6s rollouts) and representation quality based on probing for semantic segmentation. FVD evaluated on Waymo and segmentation on Cityscapes.

2 Related Work

World models for autonomous driving aim to predict future scene evolution conditioned on past observations and ego actions. A first line of work models this prediction in *discrete token space*: GAIA-1 [23], DrivingWorld [24], and DrivingGPT [7] employ autoregressive transformers over quantized visual tokens, while Cosmos [1] scales this paradigm to large-scale video pretraining. A second line operates in *continuous latent space*, typically with diffusion [21, 26] or flow-matching [34] objectives: Vista [12], DriveDreamer [55], GenAD [56], MagicDrive-V2 [11], Epona [58], and Orbis [36] demonstrate high-fidelity multi-frame generation across diverse driving scenarios. Orbis directly compares these two paradigms under a hybrid tokenizer and finds that continuous representations are less brittle and more effective for long-horizon prediction than discrete tokens. Despite differing in their representation choices, all of these models share a single level of abstraction in their predictor.

2.1 Hierarchical planners

In reinforcement learning, hierarchical approaches have been a popular paradigm to learn both high-level skills and detailed control [50, 32]. Director [16] learns a two-level hierarchy where a high-level planner selects subgoals in latent space and a low-level policy executes them. Many recent vision-language-action models can also be regarded as hierarchical planners, with language-based planning covering the abstract, strategic level, whereas a fine-grained policy covers low-level control. For example, Hi Robot [46] interprets complex instructions with a VLM to determine the next step, which conditions a low-level action policy. HiLAM [27] discovers latent skills in an unsupervised fashion by modeling long-term temporal structure, then uses them to condition short-horizon prediction.

2.2 World models with abstract latent spaces

THICK [14] learns multi-scale temporal abstractions in the context of hierarchical model-based RL. This is closest to our work, as it includes multiple abstractions on the side of the predictor. DINO-WM [62] does not implement a hierarchy, but uses DINO features as latents for its predictor, which is similar to our abstract-level predictor. In concurrent work, Zhang et al. [59] use predictors at two levels for hierarchical planning, but both predictors share the same latent space. They demonstrate that planning with this hierarchy of predictors substantially improves the success rate compared to various flat models.

3 Method

3.1 Preliminary: Flow Matching

We adopt the flow matching objective [34] to model the distribution of future frames $\mathbf{x}_N$ in latent space. A noisy interpolant is constructed as $\mathbf{x}_N^\tau = (1-\tau)\mathbf{x}_N + \tau\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\tau \in [0, 1]$, defining a linear path from data to noise. A velocity network $\mathbf{v}(\mathbf{x}_N^\tau;\, \mathbf{x}_{0:N-1})$, conditioned on context

frames $\mathbf{x}_{0:N-1}$, is trained to regress the ground-truth velocity field:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau,\,\epsilon}\big[\|\mathbf{v}(\mathbf{x}_N^\tau;\,\mathbf{x}_{0:N-1}) - (\epsilon - \mathbf{x}_N)\|^2\big]. \tag{1}$$

During inference, the predicted frame is generated by integrating the learned velocity field from $\tau{=}1$ to $\tau{=}0$, initialized from $\mathbf{x}_N^1 \sim \mathcal{N}(0, I)$.

Teacher forcing and diffusion forcing: A central design choice is how the velocity network conditions on context frames during training. Under **teacher forcing**, context frames are always provided as clean ground-truth observations and the velocity is predicted only for the last frame, as in eq. (1). In contrast, **diffusion forcing** [6] independently corrupts each frame $\mathbf{x}_i$ in the full sequence $\mathbf{x}_{0:N}$ with an independent noise level $\tau_i \sim [0, 1]$, and trains the model to predict the velocity for *all* frames jointly:

$$\mathcal{L}_{\text{DF}} = \mathbb{E}_{\tau_{0:N},\,\epsilon_{0:N}}\big[\|\mathbf{v}(\mathbf{x}_{0:N}^{\tau_{0:N}}) - (\epsilon_{0:N} - \mathbf{x}_{0:N})\|^2\big]. \tag{2}$$

While teacher forcing has emerged as a common training objective for video world models [36, 1, 58, 53], we find that it yields weaker internal representations, as the model is never exposed to corrupted context during training. We demonstrate that diffusion forcing leads to stronger internal representations.

3.2 Hierarchical World Model

We propose a two-level hierarchical world model that decomposes future prediction along both abstraction and temporal axes. The high-level predictor $\mathcal{F}_H$ operates at a coarse temporal resolution in an abstract latent space, capturing long-horizon dynamics. The low-level predictor $\mathcal{F}_L$ operates at a finer temporal resolution in a pixel-aligned latent space, making detailed short-horizon prediction conditioned on the high-level prediction. An overview of the architecture is shown in Figure 1.

Tokenizer latent representations: Let o denote an observed driving frame. For each frame o, we extract two complementary representations. The fine-grained latent $\mathbf{x} = \mathcal{E}_L(o)$ is produced by a tokenizer encoder $\mathcal{E}_L$ trained with a reconstruction objective [45, 10]. This latent preserves pixel-aligned details and is used by the low-level predictor. In parallel, we learn an abstract encoder $\mathcal{E}_H$ that produces a high-level latent representation, $\mathbf{z} = \mathcal{E}_H(o)$. Unlike the detail tokenizer, $\mathbf{z}$ is optimized to capture semantic and structural information useful for long-horizon prediction. We train $\mathcal{E}_H$ by aligning its representation with a frozen DINO embedding [40] through a learned projection head $\mathcal{P}$. We add a weak reconstruction regularizer through a lightweight decoder $\mathcal{D}_H$, weighted by $\lambda_{\text{rec}} = 0.1$, so that the abstract latent retains task-relevant visual information without degrading its semantic alignment. Formally, given an observation o and its frozen DINO embedding $\mathcal{E}_{\text{DINO}}(o)$, we train $\mathcal{E}_H$, $\mathcal{P}$, and $\mathcal{D}_H$ with:

$$\mathcal{L}_{\text{abs}} \;=\; \underbrace{\big\|\mathcal{P}\big(\mathcal{E}_H(o)\big) - \mathcal{E}_{\text{DINO}}(o)\big\|^2}_{\text{DINO alignment}} + \lambda_{\text{rec}}\,\underbrace{\big\|\mathcal{D}_H\big(\mathcal{E}_H(o)\big) - o\big\|^2}_{\text{reconstruction regularizer}}, \tag{3}$$

The resulting abstract latent $\mathbf{z}$ is only 16-dimensional latent space, providing a compact DINO-aligned representation that is well suited to the flow-matching objective of the high-level predictor.

Abstract long-horizon predictor: The high-level predictor $\mathcal{F}_H$ models future dynamics in the abstract latent space at a coarse temporal resolution. Let $\mathbf{z}_i$ denote the abstract latent of the frame at time step i, and let $\mathbf{Z}_N^k = \{\mathbf{z}_0, \mathbf{z}_k, \mathbf{z}_{2k}, \ldots, \mathbf{z}_N\}$ denote the strided context of past abstract latents sampled every k frames. Conditioned on this context and the current ego action a_N, $\mathcal{F}_H$ predicts the abstract latent k steps into the future:

$$\hat{\mathbf{z}}_{N+k} \;=\; \mathcal{F}_H\big(\mathbf{Z}_N^k,\, a_N\big). \tag{4}$$

Detailed short-horizon predictor: The low-level predictor $\mathcal{F}_L$ operates at fine temporal resolution and is responsible for generating the k intermediate fine-grained frames between $\mathbf{z}_N$ and $\hat{\mathbf{z}}_{N+k}$. Let $\mathbf{x}_i$ denote the fine-grained latent at time step i. Conditioned on its own short high-frequency context window of m frames and the abstract subgoal $\hat{\mathbf{z}}_{N+k}$, $\mathcal{F}_L$ predicts the next k frames:

$$\hat{\mathbf{x}}_{N+1:N+k} = \mathcal{F}_L\big(\mathbf{x}_{N-m:N},\, \hat{\mathbf{z}}_{N+k}\big). \tag{5}$$

During training, the abstract high-level predictor is pre-trained using the diffusion forcing loss eq. (2) and fine-tuned by teacher forcing loss eq. (1). The detailed low-level predictor is trained directly with a teacher-forcing objective conditioned on the high-level predictor output.

Action-conditioning: We first pre-train the hierarchical predictors unconditionally, without action information. We then fine-tune the world model to enable conditional generation on planar trajectories. The embedded steering vector is added to the τ embedding, and fed to the predictor via adaptive layer normalization layers [42, 41, 3]. We only condition the top-level predictor branch.

3.3 Diffusion-forcing pre-training for improved predictor representations

Diffusion forcing [6] was originally introduced to improve rollout length by mitigating exposure bias in sequential generation: models trained with teacher forcing observe clean ground-truth context during training, but must condition on their own prediction at inference time. This typically causes error to compound over long rollouts. We discover that diffusion forcing also substantially improves the internal representations learned by the predictor. Building on this finding, we propose using diffusion forcing as a pre-training stage followed by teacher-forcing fine-tuning for the predictor training. The fine-tuning stage aligns the predictor with the downstream future-prediction objective while retaining the representation quality gained during diffusion forcing.

We validate this effect through linear probes on semantic segmentation and depth estimation, and observe consistent gains across predictors operating in both detail and abstract tokenizer latent spaces, as well as across both probing tasks. This training strategy is independent of our hierarchical architecture and generalizes to any predictor level. This new training strategy also results in much stable long-horizon rollouts. In our model, we apply it to the abstract predictor $\mathcal{F}_H$, where it yields stronger high-level dynamics representations and better downstream world modeling performance.

4 Experiments

4.1 Datasets and Training Details

Table 1 provides an overview of the training data used in training both predictors. Both levels are trained for a single epoch over the full 5,890 hours of video. Both tokenizers are trained on a curated mix of 2.6 million frames sampled from BDD100K [57], OpenDV [56], Honda HAD [28], Honda HDD [43], ONCE [35], nuScenes [4], and nuPlan [5]. For steering-conditional generation, the model is fine-tuned on a 500h sub-set of the NVIDIA PhysicalAI AV dataset [39] using the dataset's IMU annotations to compute planar trajectory inputs. The video dataset [38] used in this study was acquired in partnership with NATIX. The dataset is partially publicly available. It is a large-scale, crowd-sourced collection of multi-camera driving data spanning the United States, Europe, and Japan, covering varied road conditions, traffic environments, and regional driving behaviors. For this work, we used a 2,374-hour subset of front-camera driving video data. In future work, we plan to extend this approach using multi-camera data.

The high-level predictor operates at 2Hz with a window of 6 frames (3 seconds), taking 5 frames as input and predicting 1 frame. Its predicted abstract latent serves as the subgoal that conditions the low-level predictor. The low-level predictor operates at 10 Hz with a window of 10 frames (1second), taking 5 frames as context and generating the next 5 frames. During training, the low-level predictor is conditioned on ground-truth abstract latents. At inference, it uses the predicted latents from the high-level predictor. Rollouts are performed autoregressively: the high-level predictor advances with a sliding window of 1 frame and the low-level predictor with a sliding window of 5 frames. Both predictor models are trained at a resolution of 512×288.

Table 1: Overview of the training datasets used for the world model.*partially public

	OpenDV [56]	BDD100k [57]	PhysicalAI-AV [39]	NATIX [38]*	Total
Hours (h)	1124	659	1633	2374	5890

4.2 Model design

For the first stage, we use the hybrid tokenizer proposed in [36], consisting of a ViT-based [9] image encoder with a vector-quantized bottleneck for regularization, trained with a VQGAN [10] objective.

The encoder contains 86M parameters, while the decoder is a CNN-based architecture with 91M parameters. In the hierarchical setup we train a separate tokenizer per predictor. The high-level one targets DINOv2B embedding reconstruction with a small pixel reconstruction weight $\lambda_{\text{rec}} = 0.1$; the low-level one follows the pixel-aligned VQGAN [10] objective.

Following [36], we extend the DiT block [41] into a Spatio-Temporal Transformer (ST-Transformer) backbone with 512M parameters. The low-level detail predictor adds 43M parameters through spatially aligned conditioning via adaLN, summing to 555M parameters. The full hierarchical predictor totals 1067M parameters. We maintain an exponential moving average (EMA) of the model weights with a decay of 0.9999 and use the EMA model at inference. For the model-internal design studies, we train at a lower resolution of 336×192 with a smaller 257M abstract predictor and 279M detail predictor.

4.3 Evaluation

Representation quality. We evaluate internal representations via linear probing on two downstream tasks: semantic segmentation on Cityscapes [8], reporting mIoU, and depth estimation on KITTI [13], reporting RMSE. Details about linear probing setup are included in Appendix A.8.

Predictive performance. We assess the visual quality and diversity of predicted rollouts using Fréchet Video Distance (FVD) [52], evaluated on three benchmarks: Waymo [48], nuPlan, and nuPlan-turns [36]. nuPlan-turns is a subset of nuPlan dataset focused on turning scenarios. 800 sequences are used for nuPlan benchmark, and 400 sequences for Waymo and nuPlan-turns benchmarks.

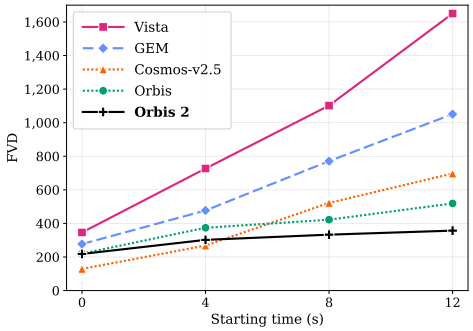


Figure 3: Performance (FVD) of long-horizon rollouts on nuPlan-turns over consecutive 4s windows. Models with good initial video quality can quickly degrade over time. FVD-slope captures this progressive fidelity loss.

Long-horizon stability. In a predictive setting, standard FVD can be confounded by shared conditioning frames at the beginning of each rollout. Methods that prioritize perceptual fidelity or are optimized for long-chunk generation can achieve strong short-horizon FVD without truly capturing semantics or physical structure. To address this, we adopt the chunked-FVD metric from [36], which measures FVD over consecutive 4-second windows. We additionally introduce **FVD-slope**, a scalar metric capturing the temporal degradation of video quality across consecutive chunks: FVD-slope $= \frac{\text{FVD}_T - \text{FVD}_1}{T-1}$, where FVD_t denotes the FVD of the t-th chunk and T is the total number of chunks. A lower FVD-slope indicates greater stability over long-horizon prediction.

Action controllability. Given the importance of action-conditional generation for world models, we evaluate their ability to follow given steering commands in the form of 2D trajectories. Beyond the original trajectories, we evaluate on counterfactual trajectories with scaled speed and yaw rate (under- and oversteering), which cannot be inferred from the context frames and therefore test whether the model genuinely responds to the steering input.

5 Results

5.1 Comparison to State-of-the-art

Long-horizon rollout performance The ability to perform long-horizon rollouts reflects the quality of the learned transition function - how well the world model maps past states to future states. We first evaluate this by comparing driving world models using FVD. Table 2 (first three columns) reports FVD over the 6-second horizon window. Cosmos-v2.5 (version: Predict-2B, post-trained, Video2World) achieves the lowest FVD scores for nuPlan benchmarks. However, it is $2\times$ larger than our model and is trained on over $3\times$ more driving data from an undisclosed data mix. Despite this,

Table 2: FVD over 6s rollouts on nuPlan, Waymo and nuPlan-turns datasets. FVD-slope is shown for 4 chunks of 4 seconds each. Linear probing results are shown for semantic segmentation on Cityscapes and depth estimation on KITTI. *Epona is trained only on the nuPlan and nuScenes datasets. The training mix of Cosmos is not public.

Model	FVD↓			FVD-slope↓	Segmentation	Depth
	nuPlan	Waymo	nuPlan-turns	nuPlan-turns	mIoU↑	RMSE↓
Vista	289.95	351.42	353.27	434.67	39.76	4.884
GEM	348.36	218.61	318.73	258.00	45.81	4.373
Orbis	132.25	180.54	239.05	99.67	52.39	4.321
Epona*	251.81	373.03	123.60	93.89	35.81	5.720
Cosmos-v2.5*	**67.74**	116.12	**126.26**	189.34	43.20	6.384
Orbis 2	98.97	**98.01**	187.53	**46.45**	**58.29**	**4.157**

our model outperforms Cosmos-v2.5 on the Waymo benchmark. We further compare long-horizon rollout stability using chunked-FVD (see Figure 3) and FVD-slope (see Table 2), where our model achieves the best scores, demonstrating significantly greater stability than all baselines including Cosmos-v2.5. Epona, which is trained only on nuPlan and nuScenes datasets, performs well on nuPlan benchmarks but fails to generalize to the unseen Waymo. Orbis shows the next-best rollout stability. It also uses two abstraction levels, but in a single branch setting and at same temporal resolution. A direct comparison with the Orbis (v1) model is added in Appendix A.2. Please find qualitative examples of long-horizon prediction for different baselines on the project page.

Conditional trajectory evaluation To evaluate the responsiveness of our model to steering commands, we consider two settings. 1) **Original trajectories**: using the unaltered ground truth steering signal corresponding to each validation sample (as done in [12, 18, 36]) and 2) **Counterfactual trajectories**: obtained by altering the original odometry, to construct alternative steering events (e.g. under/oversteering). Specifically, we obtain four counterfactual trajectories per sample by scaling the original speeds and yaw rates by a factor 0.5 and 1.5. It is essential to evaluate steering in a counterfactual setting, because the original trajectories are partly leaked by the context frames, i.e. a model that cannot steer could still score well by relying on context only. We evaluate the models by estimating the trajectory of the generated videos using an external inverse dynamics model, as done in [12, 18, 36], specifically VGGT [54]. We then compute the average displacement error (ADE) between the estimated and ground truth trajectories.

Because VGGT trajectories are scale-ambiguous, we align each generated trajectory $\mathbf{x}$ to its real counterpart $\bar{\mathbf{x}}$ before computing ADE using the least-squares scale $s^* = \left(\sum_t \mathbf{x}_t^\top \bar{\mathbf{x}}_t\right) / \left(\sum_t \mathbf{x}_t^\top \mathbf{x}_t\right)$. This removes global scale error while preserving penalties for shape, direction, and curvature. We evaluate one original and four counterfactual modes on nuPlan-turns with six-second drives. Table 3 shows that our model is best on all cases. Qualitative examples are in Appendix A.6.

Table 3: Steering evaluation on original and counterfactual trajectories. ADE↓ in meters.

Model	Original	Counterfactual			
		speed		yaw rate	
		×0.5	×1.5	×0.5	×1.5
Orbis	2.30	1.39	3.27	1.31	3.69
Epona	1.40	1.76	2.10	1.81	2.42
Orbis 2	**1.24**	**0.22**	**1.70**	**0.59**	**2.38**

5.2 Strong predictor representations lead to stable world models

We evaluate the internal representation quality of driving world models by linear probing on semantic segmentation and depth estimation (Table 2). We observe a clear trend: models with stronger semantic and geometric understanding also achieve more stable long-horizon prediction, as measured by FVD-slope. Our model achieves the best results on both probing tasks and the long-horizon stability metric. Orbis (v1), which also learns strong representations, achieves the second-best long-horizon stability. Several baselines operate at significantly higher resolutions to maximize perceptual fidelity, yet our model is more stable at 512×288 resolution. This suggests that representation quality plays a complementary, and arguably more important role in world modeling than perceptual fidelity alone.

5.3 Comparison between hierarchical variants

Table 4 compares the design choices for the top and bottom predictors (illustrated in fig. 1), varying the level of latent abstraction, frame rate, and training mode. The first two rows establish flat (single-branch) baselines at different abstraction levels. The detail-level model (Exp. 1) achieves strong but progressively degrading FVD and learns weak internal representations, whereas the abstract-level model (Exp. 2) yields substantially richer representations and remarkably stable rollout quality over long horizons, despite its high absolute FVD. The hierarchical configurations (Exps. 3–5) all fix the bottom predictor to operate on detailed latents at 10 fps, and vary the top predictor. Exp. 3 shows the best hierarchical design, where the top predictor operates on high abstraction latents at a longer horizon of 2 fps: it achieves both strong representations and the best FVD-slope with a competitive absolute FVD performance. Exp. 4 has high FVD-slope: the subpar long-horizon rollouts of the top predictor in the detailed latent space cause the model to drift out-of-distribution. In Exp. 5, the top predictor operating in abstract space in high-temporal resolution leads to stopping behavior in rollouts. We attribute this to the abstract latent space is not detailed enough to capture small motion at 10 fps, which leads to loss in motion signal. All bottom predictors do not learn strong representations due to their detailed predictive task with strong conditioning. The best design is hierarchical with an abstract-latent top predictor at 2 fps (Exp. 3): it best balances overall (FVD) and long-horizon video fidelity (FVD-slope), with the strongest representations. Figure 4 qualitatively confirms this: Exp. 1 drifts, Exp. 2 is stable but lacks detail, Exp. 4 diverges out-of-distribution, and Exp. 5 progressively loses motion, whereas Exp. 3 remains stable and perceptually detailed.

Table 4: Hierarchy design study across abstractions and temporal resolutions. All experiments have comparable parameter counts. Exp. 3 is our final hierarchical setting. Top/Bottom refers to predictor positions in fig. 1. FVD and FVD-slope are reported on the nuPlan-turns benchmark.

Exp.	Model branch	Abstraction detail/abstract	Frame-rate fps	Sem. Seg. mIoU ↑	Depth RMSE ↓	FVD↓ 16s	FVD-slope↓ 4 x 4s
1.	Single-branch	detail	10	32.72	6.107	170.02	101.63
2.	Single-branch	abstract	10	47.88	4.624	490.04	48.38
3.	Top ↳ Bottom	abstract detail	2 10	49.39 31.97	4.498 6.396	209.15	47.87
4.	Top ↳ Bottom	detail detail	2 10	41.02 25.52	5.256 6.514	207.71	81.40
5.	Top ↳ Bottom	abstract detail	10 10	49.27 31.46	4.460 6.311	551.38	137.07

5.4 Training strategy: diffusion forcing + teacher forcing

We evaluate the effect of diffusion forcing pre-training followed by teacher forcing fine-tuning on the representation quality of the abstract predictor. Table 5 reports linear probing results for semantic segmentation and depth estimation. Diffusion forcing improves over teacher forcing alone on both tasks, and subsequent teacher forcing fine-tuning provides additional gains, yielding the best overall probing performance among our predictor variants. The final predictor outperforms the abstract tokenizer based on DINOv2-B on both probing tasks, improving mIoU from 57.19 to 58.29 and reducing RMSE from 4.574 to 4.157. The same trend holds for the predictor trained on pixel reconstruction-based tokenizer latents: diffusion forcing improves representation quality at every block of the predictor, on both semantic segmentation and depth estimation, rather than only at the output. The per-block probing curves are reported in Figure 7 in the Appendix.

DINOv2-Base shows stronger overall performance in segmentation probing, but our predictor achieves better performance on frequent driving classes such as road, car, sidewalk, and vegetation, while underperforming on rarer classes such as bus, train, and motorcycle (see Appendix A.1 for class-wise performance). Compressing DINOv2 features into the tokenizer reduces depth estimation performance, but our training strategy significantly narrows the gap to DINOv2. With an identical DPT [44] decoder trained on frozen features (Table 7), Orbis 2 slightly outperforms DINOv2-B and V-JEPA 2.1 [37] on single frames. The gap widens with temporal context, suggesting that the predictor's features integrate geometric cues across frames.



Figure 4: Qualitative examples showing behavior of different hierarchy variants discussed in Table 4. We observe that Exp-2 rollouts are stable but lacks details, Exp-4 drifts out-of-distribution and Exp-5 suffers from loss in motion.

Table 5: Ablation of the proposed training strategy. All metrics are reported for the abstract branch including tokenizer and predictor. Full-scale setting used.

Model	Segmentation mIoU ↑ (val loss↓)	Depth RMSE ↓
Abstract Tokenizer	57.19 (0.265)	4.574
Teacher forcing (TF) only	55.01 (0.270)	4.355
Diffusion forcing (DF) only	58.03 (0.254)	4.274
DF + TF (Orbis 2)	58.29 (0.250)	4.157
DINOv2-Base	60.01 (0.284)	3.890

Table 6: Inference speed and VRAM requirements across models, measured on a H100 GPU.

Method	FPS↑	VRAM ↓
Cosmos v2.5	0.66	22
Vista	0.58	86
GEM	0.44	45
Epona	0.4	17
Orbis	0.70	24
Orbis 2	4.37	19
Orbis 2 Distilled	17.40	19

Ablation study To isolate the contribution of the hierarchical architecture from the proposed training strategy, we conduct an ablation comparing two hierarchical setups with identical model size, compute, and data, differing only in how the abstract predictor is trained: one with teacher forcing (TF) only, and one with our proposed diffusion-forcing + teacher-forcing (DF+TF) strategy (the bottom/detail predictor uses teacher forcing in both). Results on representation quality follow the same trend as Table 5, where DF+TF performs better than TF-only. The long-horizon stability with TF-only training is significantly worse compared to DF+TF training. The TF-only hierarchical model produces FVD-slope 107.21, compared to FVD-slope 47.87 with DF+TF training. Results are summarized in Table 8.

Table 8: Isolating the training strategy from the hierarchical architecture. Top/Bottom refers to position of predictors in Table 1. DF+TF improves representation quality, consistent with Table 5, and substantially improves long-horizon stability. FVD and FVD-slope are reported on the nuPlan-turns.

Model branch	Training mode	Sem. Seg. mIoU↑	Depth RMSE↓	FVD↓ 16s	FVD-slope↓ 4 x 4s
Top ↳Bottom	TF TF	48.01	4.606	239.46	107.21
Top ↳Bottom	DF+TF TF	**49.39**	**4.490**	**209.15**	**47.87**

Table 7: Monocular depth on a drive-disjoint KITTI split from frozen backbones, each with an identical, freshly trained DPT decoder. Six frames: target plus five past frames at 2 Hz. DINOv2 is evaluated on single frames only. RMSE in metres; best per block in bold. Full protocol and further baselines in Appendix A.8.

		Single frame			Six frames @ 2 Hz		
Backbone	Params	AbsRel↓	RMSE↓	δ_1 ↑	AbsRel↓	RMSE↓	δ_1 ↑
DINOv2-B	87M	0.067	3.035	0.950	–	–	–
V-JEPA 2.1 ViT-g	1013M	0.066	3.043	0.951	0.069	3.067	0.949
Orbis 2 (Abstract pred. only)	599M	**0.065**	**3.016**	**0.953**	**0.060**	**2.791**	**0.966**

5.5 More analysis

Model efficiency. Our final 1.07B-parameter model is trained in under 8k H100-GPU-hours, substantially less compute than comparable driving world models. GAIA-1's 6.5B world model required 23k A100-hours, Epona used 16k A100-hours for a 2.5B model, and Orbis used 8k A100-hours to train a 469M-parameter model, less than half our size. Our model achieves the fastest inference at 4.37 FPS while requiring only 19 GB VRAM, at least $5\times$ faster than all baselines, shown in Table 6. Further, we apply teacher-forced causal consistency distillation following Causal Forcing++ [60], building on consistency distillation [47]. Using consistency distillation for the abstract predictor and heun sampler [26, 20] for the detail predictor, inference latency improves from 4.37 to 17.40 FPS with marginal drop in performance.

Abstract latent design. Two choices define our abstract latent space: whether to compress DINO at all, and how far. A natural alternative is to predict directly in the high-dimensional DINO space, as proposed in Representation Autoencoders (RAE) [61], which could in principle replace our compressed latent with raw DINO features; and given compression, the latent width trades semantic capacity against how easily flow matching can model the space. We find that predicting in full DINO space is prohibitive for autoregressive rollouts, with frame quality degrading progressively over the horizon, whereas our compressed latent retains the semantic structure of DINO at a dimensionality that flow matching models stably. Within the compressed space, representation quality improves monotonically with width, but long-horizon stability is best at 16 dimensions and degrades on either side of it, so 16-dim gives the best overall trade-off. The qualitative comparison and the dimensionality study supporting both conclusions are reported in Appendix A.3 and Appendix A.5.

6 Conclusion

We demonstrated that the proposed hierarchical approach makes effective use of abstract and detailed latent spaces, yielding a world model that combines high-fidelity short-term predictions with stability over long horizons. The model is steerable through action conditioning. We further demonstrated a strong correlation between representation quality and rollout stability, and showed that a diffusion forcing pre-training strategy consistently strengthens the learned representations.

Limitations. Although diffusion forcing improves predictor representations, the reason for this improvement is not fully studied in this work. We hypothesize that independent noise levels across frames act as implicit data augmentation, while different noise patterns encourage the model to learn from related tasks such as denoising, interpolation, and forecasting. We do not study these factors separately in this work, and leave a more detailed analysis of noise schedules and corruption patterns to future work. Our hierarchy relies on separate abstract and fine-grained latent spaces, each optimized with a different objective. While this separation clarifies the role of each level, it limits the interaction between abstraction levels. Improving the connection between the two spaces is an important direction for future work.

Societal Impact. At this stage, the work remains an early research contribution, and any practical societal effects will depend on substantial future progress in model design, scaling, validation, and system integration.

Acknowledgment

This work was funded by the German Federal Ministry for Economic Affairs and Energy within the project “NXT GEN AI METHODS" (19A23014R). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS [25] and JUPITER at Jülich Supercomputing Centre (JSC). This project was selected as a winner of Gauss AI Compute Competition, organized by the Gauss Centre for Supercomputing (GCS) and Jülich Supercomputing Centre (JSC). The project was accomplished under GCS compute grants - *multiscale-wm* and *nxtaim-1*. Further local compute resources were funded through the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant numbers 417962828, 539134284, through EFRE (FEIH_2698644), and the state of Baden-Württemberg.



We thank NATIX for providing access to their large-scale, crowd-sourced, multi-camera driving dataset [38] in support of this open-source research. We thank Marcel Aach for his technical support and assistance with the GCS HPC cluster. We also thank Muhammad Ali for improving the model latency for the Hugging Face demo.

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] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[3] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[6] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[7] 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.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[11] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28135–28144, October 2025.

[12] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A Generalizable Driving World Model with High Fidelity and Versatile Controllability. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 91560–91596. Curran Associates, Inc., 2024.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V. Butz. Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

[16] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

[17] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[18] Mariam Hassan, Sebastian Stapf, Ahmad Rahimi, Pedro Rezende, Yasaman Haghighi, David Brüggemann, Isinsu Katircioglu, Lin Zhang, Xiaoran Chen, Suman Saha, et al. Gem: A generalizable ego-vision multimodal world model for fine-grained ego-motion, object dynamics, and scene composition control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22404–22415, 2025.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[20] Karl Heun et al. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45(23-38):7, 1900.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Mark K Ho, David Abel, Carlos G Correa, Michael L Littman, Jonathan D Cohen, and Thomas L Griffiths. People construct simplified mental representations to plan. *Nature*, 606(7912):129–136, 2022.

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

[24] Xiaotao Hu, Wei Yin, Mingkai Jia, Junyuan Deng, Xiaoyang Guo, Qian Zhang, Xiaoxiao Long, and Ping Tan. Drivingworld: Constructing world model for autonomous driving via video gpt. *arXiv preprint arXiv:2412.19505*, 2024.

[25] Jülich Supercomputing Centre. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities*, 7(A138), 2021.

[26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[27] Hanjung Kim, Lerrel Pinto, and Seon Joo Kim. Hierarchical latent action model. *arXiv preprint arXiv:2603.05815*, 2026.

[28] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[30] Amandeep Kumar and Vishal M Patel. Learning on the manifold: Unlocking standard diffusion transformers with representation encoders. *arXiv preprint arXiv:2602.10099*, 2026.

[31] Yann LeCun and Courant. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. 2022.

[32] Alexander Li, Carlos Florensa, Ignasi Clavera, and Pieter Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2020.

[33] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024.

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[35] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021.

[36] Arian Mousakhan, Sudhanshu Mittal, Silvio Galesso, Karim Farid, and Thomas Brox. Overcoming challenges of long-horizon prediction in driving world models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[37] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

[38] NATIX. NATIX Multi-Camera Driving Dataset. Hugging Face Hub: https://huggingface.co/datasets/natix-network-org/natix-multi-camera-driving-dataset, 2026. Multi-camera driving dataset with telemetry metadata. NATIX website: https://www.natix.network/.

[39] NVIDIA. Physical ai autonomous vehicles dataset, 2025.

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[42] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[43] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168. IEEE, 2021.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[46] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

[47] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[48] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[49] Richard S Sutton. An adaptive network that constructs and uses and internal model of its world. *Cognition and Brain Theory*, 4(3):217–246, 1981.

[50] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

[51] Momchil S. Tomov, Samyukta Yagati, Agni Kumar, Wanqian Yang, and Samuel J. Gershman. Discovery of hierarchical representations for efficient planning. *bioRxiv*, 2018.

[52] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation, 2019.

[53] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

[54] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[55] 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.

[56] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Genad: Generalized predictive model for autonomous driving. *arXiv preprint arXiv:2403.09630*, 2024.

[57] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

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

[59] Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, et al. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026.

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

[61] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[62] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

A Appendix

A.1 Class-wise semantic segmentation probing results

In Table 9, we compare class-wise semantic segmentation probing performance for the tokenizer, predictor, and DINOv2-Base representations. DINOv2-Base achieves stronger overall segmentation performance, but our predictor performs better on frequent driving classes such as road, car, sidewalk, and vegetation, improving by 2–5 mIoU points. In contrast, it underperforms on rarer classes such as bus, train, rider, and motorcycle, with a gap of 5–20 mIoU points.

Table 9: Comparison of our tokenizer and abstract predictor model with DINOv2-base. Per-class IoU for semantic segmentation reported on Cityscapes. Δ columns show the predictor's change relative to our tokenizer and to DINOv2-B.

Class	Dino v2-B	Ours tokenizer	Ours predictor	Δ Predictor − Tokenizer	Δ Predictor − Dino v2-B
Road	94.88	95.3	95.9	+0.6	+1.02
Sidewalk	66.58	67.6	71.0	+3.4	+4.42
Building	82.98	84.4	84.8	+0.4	+1.82
Wall	48.28	47.7	47.9	+0.2	−0.38
Fence	45.84	44.8	44.9	+0.1	−0.94
Pole	17.54	16.7	18.5	+1.8	+0.96
Traffic Light	33.97	30.1	31.5	+1.4	−2.47
Traffic Sign	42.93	41.4	43.6	+2.2	+0.67
Vegetation	80.87	83.3	83.6	+0.3	+2.73
Terrain	53.05	53.5	53.4	−0.1	+0.35
Sky	82.11	86.6	87.0	+0.4	+4.89
Person	54.18	53.0	52.7	−0.3	−1.48
Rider	34.04	28.1	28.2	+0.1	−5.84
Car	84.27	85.0	86.4	+1.4	+2.13
Truck	70.25	67.0	67.1	+0.1	−3.15
Bus	77.80	63.4	69.5	+6.1	−8.30
Train	70.57	48.5	50.0	+1.5	−20.57
Motorcycle	45.48	37.4	38.1	+0.7	−7.38
Bicycle	54.57	52.9	53.1	+0.2	−1.47
mIoU	60.01	57.2	58.3	+1.1	−1.71

A.2 Single-branch, mixed abstraction

Table 10 shows the comparison between our model and a single branch model with mixed abstraction levels, combined via token factorization [33], as in [36]. Despite good FVD and representation quality, video fidelity degrades faster over time, as shown by the higher FVD-slope.

Table 10: Comparison of single-branch and hierarchical predictors.

Model	Parameters M	Sem. Seg. mIoU ↑	Depth RMSE ↓	FVD↓ 16s	FVD-slope↓ 4 × 4s
Single-branch, mixed	512	47.16	4.680	184.69	96.55
Hierarchical (ours)	257+279	49.39	4.498	209.15	47.87

A.3 Qualitative rollouts with full DINO space vs compressed DINO space

RAE [61] could in principle serve as the abstract predictor in our hierarchy by replacing our compressed latent with raw DINO features. However, its dimensionality is prohibitive for autoregressive rollouts, frame quality degrades progressively over the prediction horizon when the predictor operates

directly in full DINO space. Our compressed DINO latent retains the semantic structure of DINO at reduced dimensionality, which proves essential for stable long-horizon prediction with a flow matching objective.

Figure 5 shows qualitative results comparing 1-second rollouts generated using our compressed DINO latent targets versus the full 768-dim DINOv2 latent targets. Rollouts in the full-dimensional DINO space degrade rapidly, while the compressed DINO latent space is effectively modeled by the flow-matching objective, producing stable predictions throughout the horizon.

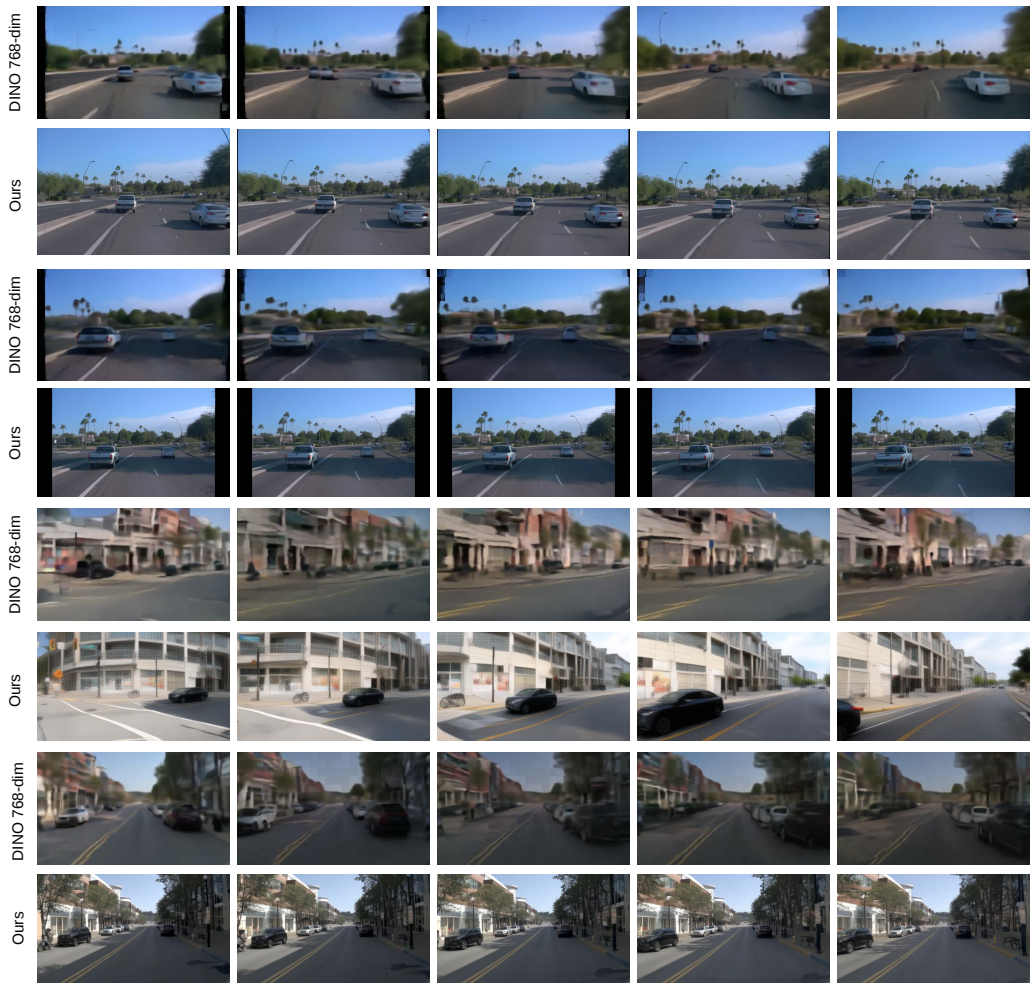


Figure 5: Comparison of raw DINO features versus our compressed DINO latent as the tokenizer for the abstract predictor. (Rollouts at 2hz)

A.4 Qualitative examples for different hierarchy variants

In Figure 4, we present qualitative examples illustrating the failure modes of the hierarchical design variants discussed in Section 5.3. Experiment-1, which operates in single-branch mode at the detailed abstraction level, produces rollouts that drift away from the starting distribution. Experiment-2, a single-branch model at the abstract level, yields stable rollouts but fails to capture fine-grained motion and perceptual detail. Experiment-3 corresponds to our proposed hierarchical design, achieving stable rollouts with high perceptual quality. Experiment-4, which uses detailed latents for the top branch, diverges out-of-distribution rapidly, resulting in poor rollouts. Experiment-5, which operates abstract latents at high temporal frequency, fails to capture detailed motion, leading to motion degradation despite retaining high perceptual quality.

A.5 Choice of abstract latent dimensionality

Our choice of 16-dim for the abstract latent was initially guided by prior work (Orbis [36]), which uses a 16-dim DINO-aligned latent. To assess how sensitive performance is to this choice in our setup, we tested 8, 16 and 32-dim abstract latents at reduced scale, training the abstract predictor with our proposed strategy (DF+TF) and evaluating representation quality and FVD-slope over 32-second rollouts (4 chunks of 8s) at 2 fps. Results are summarized in Table 11. Representation quality improves monotonically with dimensionality from 8 to 32, but long-horizon stability measured with FVD-slope is best at 16-dim, degrading at both 8 and 32. The dimensionality therefore offers a trade-off between representation quality and long-horizon stability, with 16-dim giving the best stability.

Table 11: **Effect of abstract latent dimensionality** (DF+TF, reduced scale; FVD-slope over 32 s at 2 fps). Representation quality increases with dimensionality; stability is best at 16 dims.

compressed-DINOv2 dimensionality	Semantic Seg. mIoU ↑	Depth RMSE ↓	FVD-slope↓ 4 x 8s
8	48.42	4.547	66.54
16	49.39	4.498	**50.71**
32	**50.52**	**4.418**	62.49

A.6 Counterfactual steering

Evaluation details Evaluating with counterfactual steering inputs comes with additional challenges compared to the standard real-only steering evaluation done in [12, 18, 36], where the inverse dynamics model is applied to real and generated videos alike, compensating for scale ambiguities. Here, we compare the estimated generated trajectories with the actual input trajectories, but this exposes us to scale mismatches due to the fact that the output of VGGT is not in metric space. As a solution, for each pair of real and generated trajectories $(\bar{\mathbf{x}}, \mathbf{x})$ we compute an optimal scale parameter for the latter:

$$s^* = \frac{\sum_{t=1}^{T} \mathbf{x}_t^\top \bar{\mathbf{x}}_t}{\sum_{t=1}^{T} \mathbf{x}_t^\top \mathbf{x}_t} \tag{6}$$

and rescale it accordingly before computing ADE. This still penalizes shape, direction, and curvature, but it largely removes global trajectory-scale error.

Figure 6 shows an example of steering the model on counterfactual trajectories. It is important for a world model to be able to follow commands that deviate from the original context-bound trajectories, even if they lead to out-of-distribution or illegal maneuvers.

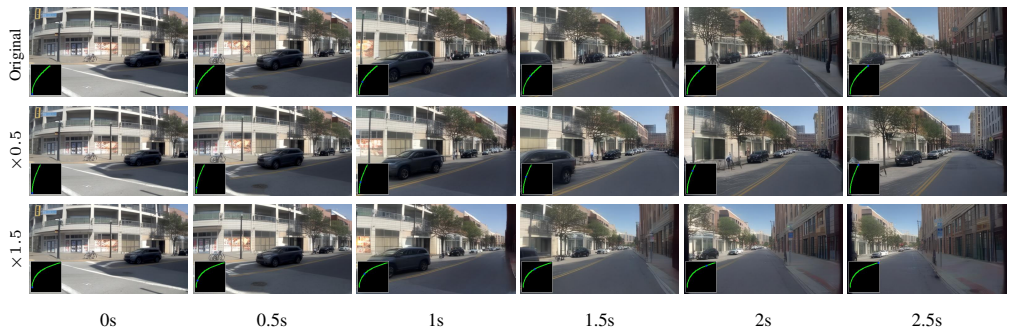


Figure 6: Example of counterfactual steering. Top row: the original steering trajectory, the model performs the turn as intended. Second row: the yaw rate was scaled by 0.5 (understeering), the model follows the command and executes a more gentle turn. Third row: the yaw rate was scaled by 1.5 (oversteering), the model follows the command and executes a sharper turn.

A.7 Model details

Tokenizer. We train two hybrid tokenizers with separate encoder and decoder branches for detail and abstract latents. The hybrid architecture follows the design proposed by Mousakhan et al. [36]. Both encoders are initialized from a MAE-pretrained [19] ViT with patch size 16×16, producing 768-dimensional embeddings that are quantized into 16-dimensional codes.

Abstract long-horizon predictor. This predictor generates semantic latents at 2 Hz autoregressively with *causal* temporal attention (`causal_time_attn=True`). It takes 5 context frames and predicts 1 future frame per rollout step.

Detailed short-horizon predictor. This predictor operates at 10 Hz and is conditioned on the abstract predictor's predicted latent. Temporal attention is *non-causal*. Conditioning is injected spatially per-frame via small non-zero-initialized linear projections and a learned frame-position MLP, ensuring the pretrained STDiT backbone is minimally perturbed at initialization. Each block injects the abstract predictor's signal through dense adaptive layer normalization (AdaLN), predicting per-token scale and shift parameters from the spatially aligned abstract latent.

Inference Protocol. At inference, generation proceeds in a cascaded fashion. The abstract predictor takes 5 context frames and generates 1 frame at 2 Hz using 15 NFE. Its predicted abstract latent serves as the subgoal that conditions the detail predictor. The detail predictor then takes 5 context frames and generates the next 5 frames at 10 Hz using 30 NFE, conditioned on the predicted abstract latent. During training, the detail predictor is conditioned on ground-truth abstract latents, whereas at inference it operates on the predicted latents from the abstract predictor. Rollouts are performed auto-regressively. The abstract predictor advances with a sliding window of 1 frame, operating over a 6-frame window (3 seconds) at 2 Hz. The detail predictor advances with a sliding window of 5 frames, operating over a 10-frame window (1 second) at 10 Hz.

A.8 Probing details

Semantic Segmentation To evaluate the quality of representations learned by the Stage-2 predictor model, we train a linear probe for semantic segmentation on the Cityscapes fine-annotation benchmark (19 classes) using the model's frozen intermediate features. Input images are normalized to $[-1, 1]$ and encoded through the frozen backbone. During probing, context is dropped (set to null) and the input image is tiled across all predicted-frame slots, with features extracted from the last frame slot. Intermediate patch-token features are extracted from the 4th transformer block from the end for the main model and from the 2nd block from the end for ablation models, then reshaped from the flat patch sequence into a 2D spatial feature map. The linear probe head is a single 1×1 LazyConv2d that maps this feature map to 19-class logits, followed by bilinear upsampling to match the ground-truth mask resolution. Only the probe parameters are trained, the backbone is kept entirely frozen throughout. Training uses AdamW (lr=1e-3, wd=1e-4) with cross-entropy loss for 100 epochs. Performance is reported as mean Intersection-over-Union (mIoU) on the Cityscapes validation set.

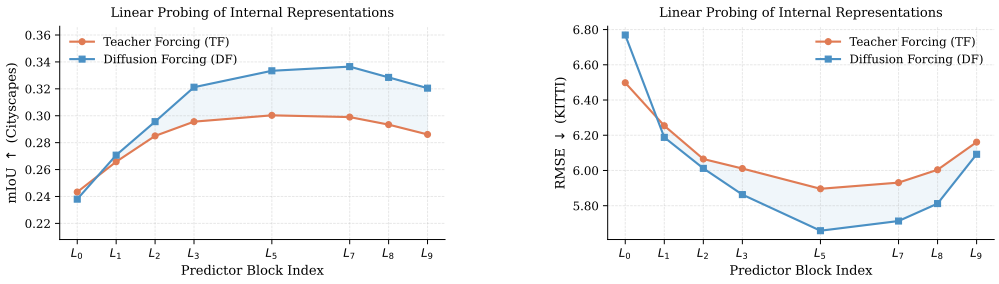
Depth Estimation (Linear). To probe the geometry-awareness of the Stage-2 predictor model, we train a linear depth probe on the KITTI Eigen split following the DINOv2 linear depth evaluation protocol. Input images are encoded through the frozen backbone; during probing, context is dropped (set to null) and the input image is passed as the sole target frame, with features extracted from the last frame slot. Intermediate patch-token features are extracted from the 4th transformer block from the end, then reshaped into a 2D spatial feature map. The linear probe head follows the DINOv2 Lin-1 design: a single 1×1 LazyConv2d producing logits over 256 uniformly-spaced depth bins covering [0, 80] m, preceded by a 4× bilinear upsample to recover spatial resolution, followed by a softmax and weighted sum over bin centers to yield a continuous depth prediction. At training time, ground-truth depth values are discretized into the same uniform bins and cross-entropy loss is minimized with AdamW (lr=1e-3, wd=1e-4) for 10 epochs; missing and out-of-range depth pixels are excluded from the loss. Only the probe parameters are trained; the backbone is kept entirely frozen. Performance is reported as RMSE on the KITTI Eigen validation split.

Table 12: Extended results for Table 7: monocular depth on the drive-disjoint KITTI split from frozen backbones with a retrained DPT decoder. Depth sup.: backbone pretrained with dense depth supervision. RMSE in metres; best per block in bold.

Backbone	Depth sup.	Params	DPT	Input	AbsRel↓	RMSE↓	δ_1 ↑
Single frame							
DINOv2-B		87M	retrained	1 frame	0.067	3.035	0.950
V-JEPA 2.1 ViT-g		1013M	retrained	1 frame	0.066	3.043	0.951
DA3-Large	✓	304M	retrained	1 frame	0.071	3.205	0.940
DA3-Giant	✓	1136M	retrained	1 frame	0.067	3.037	0.954
VGGT-1B	✓	909M	retrained	1 frame	**0.060**	**2.875**	**0.961**
Orbis 2 (ours)		599M	retrained	1 frame	0.065	3.016	0.953
Five past frames + target							
V-JEPA 2.1 ViT-g		1013M	retrained	6 @ 2 Hz	0.069	3.067	0.949
V-JEPA 2.1 ViT-g		1013M	retrained	6 @ 5 Hz	0.066	2.999	0.955
DA3-Large	✓	304M	retrained	6 @ 2 Hz	0.061	2.824	0.968
DA3-Giant	✓	1136M	retrained	6 @ 2 Hz	**0.049**	**2.516**	0.973
VGGT-1B	✓	909M	retrained	6 @ 2 Hz	0.053	2.573	**0.977**
Orbis 2 (ours)		599M	retrained	6 @ 2 Hz	0.060	2.791	0.966

Depth DPT probing protocol. We measure how much metric depth information a frozen backbone encodes by training an identical DPT decoder on top of each backbone, with the backbone weights kept fixed. KITTI is split by drive so that the training, validation and test drives are disjoint, and the decoder is trained on the training drives only. All methods are evaluated on the same 2,060 test images, namely those for which the five preceding frames exist, so single-frame and multi-frame results are directly comparable. Multi-frame inputs consist of the target frame plus five past frames sampled at 2 Hz or 5 Hz. The decoder adds 18.9–20.0M trainable parameters, depending on backbone width. The parameter counts in Table 12 cover only the frozen backbone; for Orbis 2 this is the tokenizer encoder (86M) plus the predictor (513M). DA3 and VGGT were pretrained with dense depth supervision (marked *Depth sup.*), whereas DINOv2, V-JEPA 2.1 and Orbis 2 were not. Following common practice, we evaluate on pixels with valid ground truth inside the Garg crop and within 0–80 m. Predictions are compared in metric scale, without median scaling. With $\hat{d}_i$ the predicted and d_i the ground-truth depth of pixel i, and N the number of valid pixels, we report the absolute relative error $\mathrm{AbsRel} = \frac{1}{N}\sum_i |\hat{d}_i - d_i|/d_i$, the root-mean-square error $\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_i (\hat{d}_i - d_i)^2}$ in metres, and the threshold accuracy δ_1, the fraction of pixels with $\max(\hat{d}_i/d_i,\, d_i/\hat{d}_i) < 1.25$.

Per-block representation quality. Figure 7 reports linear probing performance at each block of the predictor trained on detail latents, for the diffusion forcing and teacher forcing objectives discussed in 5.4. Diffusion forcing improves representation quality across the whole depth of the model on both probing tasks, not only in the final blocks.



(a) Semantic segmentation (mIoU ↑) on Cityscapes.

(b) Depth estimation (RMSE ↓) on KITTI.

Figure 7: Representation quality per predictor block via linear probing for the predictor trained on detail latents. Diffusion forcing consistently improves intermediate representations on both semantic segmentation and depth estimation.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract summarizes the problem addressed in this work and our key contributions. The introduction provides additional context and motivation, followed by a detailed summary of our contributions and findings.

   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: Limitations and open questions are discussed in Section 6 of the paper.

   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: The paper does not include theoretical results.

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: Sections 4.1, 4.2 and Appendix A.7 describe the data, architecture, training stages and inference protocol; Appendix A.8 describes the probing setup. Code and checkpoints are released at https://lmb-freiburg.github.io/orbis2.github.io.

   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: [No]

Justification: Code and checkpoints are released at https://lmb-freiburg.github.io/orbis2.github.io. Part of the training data (the NATIX dataset, 2,374 of 5,890 hours) is only partially public, so training cannot be fully reproduced from public data. The evaluation benchmarks (nuPlan, Waymo, Cityscapes, KITTI) are public, so all reported evaluations can be reproduced with the released checkpoints.

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: Training details (data, resolution, frame rates, context lengths, model sizes, EMA) are in Sections 4.1, 4.2; optimizer and training hyperparameters for both stages are in Appendix A.7; inference settings and probing hyperparameters are in Appendices A.7 and A.8.

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: Since training latent world models is computationally expensive, we cannot afford to train each model over multiple random seeds. To compensate, we train on large datasets and evaluate across multiple evaluation sets, which we believe yields a more reliable signal than single-seed comparisons on smaller benchmarks.

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: Section 5 reports training compute for the final model (under 8k H100 GPU-hours) and inference speed and memory on a single H100 (Table 6). Design studies use smaller models at 336×192 (Section 4.2), each trained in about [X] H100 GPU-hours.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work involves no human-subject research and uses existing datasets according to their terms; the NATIX data was acquired in partnership with NATIX.

   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: We discuss the broader social impact of our work, particularly how it can contribute to the development of reliable 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: The model is still in research area and is not ready for production or any use.

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 properly cite all prior works, including papers and code repositories, that we build upon in this paper.

Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We release code and checkpoints with documentation for training and evaluation at `https://lmb-freiburg.github.io/orbis2.github.io`.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not part of the method or experiments.

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.