OCCSIM: MULTI-KILOMETER DRIVING SIMULATION WITH LONG-HORIZON OCCUPANCY WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-driven driving simulation relies on pre-recorded logs or spatial priors such as HD maps, so the environments it can produce are capped by the scale of existing annotated datasets. To break this bottleneck, we present OccSim, **the first 3D occupancy world model driven simulator that requires neither HD maps nor log-derived layouts**. OccSim obviates the requirement for long continuous logs or HD maps; it is conditioned only on **a single initial frame** and a **sequence of future ego-actions**. It can stably generate over **3000** continuous frames, an $>80\times$ improvement in stable generation length over previous SoTA occupancy world models, and fuses them into maps spanning over 9 kilometers. Further qualitative stress tests show that it can roll out beyond **110K frames**. OccSim couples W-DiT, a static occupancy world model that builds rigid transformations into its architecture for ultra-long-horizon rollout, with a Layout Generator that populates the synthesized topology with reactive agents. With these designs, OccSim can synthesize massive, diverse simulation streams. Extensive experiments demonstrate its downstream utility: data collected directly from OccSim can pre-train 4D semantic occupancy forecasting models to achieve up to 67% zero-shot performance on unseen data, outperforming previous asset-based simulator by 11%. When the collected data is scaled up to $5\times$ its original size, zero-shot performance increases to 74%. More videos and details are available at this anonymous page.

1 INTRODUCTION

Simulation is indispensable for the advancement of Level 4 autonomous driving systems (Caesar et al., 2021; Dauner et al., 2024; Codevilla et al., 2018), yet every existing route to a simulation environment is capped by human effort: log replay yields authentic data streams but cannot react to ego behaviour; geometry-based simulators such as CARLA (Dosovitskiy et al., 2017) are interactive but suffer a large sim-to-real gap in sensor realism and diversity (Yang et al., 2023; Zhang et al., 2025); and recent data-driven simulators (Yan et al., 2025; Zhou et al., 2024) remain conditioned on HD maps or BEV layouts, so their coverage is bounded by annotated or collected maps.

We argue that a scalable data-driven simulator must satisfy four desiderata. **D1. Sensor realism & semantic richness**: match the distribution of real sensor data while inherently producing dense semantic labels, removing manual annotation from the loop. **D2. Controllable & plausible agents**: infer initial traffic states from the environment rather than replaying logs, and expose agent control as a replaceable interface accepting anything from microscopic heuristics to learned forecasters. **D3. Global spatial persistence**: maintain a consistent static world rather than the memoryless hallucination of standard generative models. **D4. Log-independence**: generate novel environments open-endedly, so coverage is not limited by the scale of collected data.

To satisfy these desiderata jointly, we present **OccSim**. To our knowledge, OccSim is the first occupancy world model driven simulator that can produce multi-kilometer, globally consistent environments without any HD map or BEV layout as input (Fig. 1); it conditions solely on a single static frame and a sequence of future ego actions, and is trained entirely on publicly-available data.

OccSim decouples static and dynamic voxels into two synergistic generative tasks, which are also decoupled in time: a multi-kilometer rollout of the world model assembles the large-scale static scene once and then freezes it, while dynamic traffic is generated and controlled online on top of it. For static generation, the core difficulty is sustaining stable, diverse and high-quality output over

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

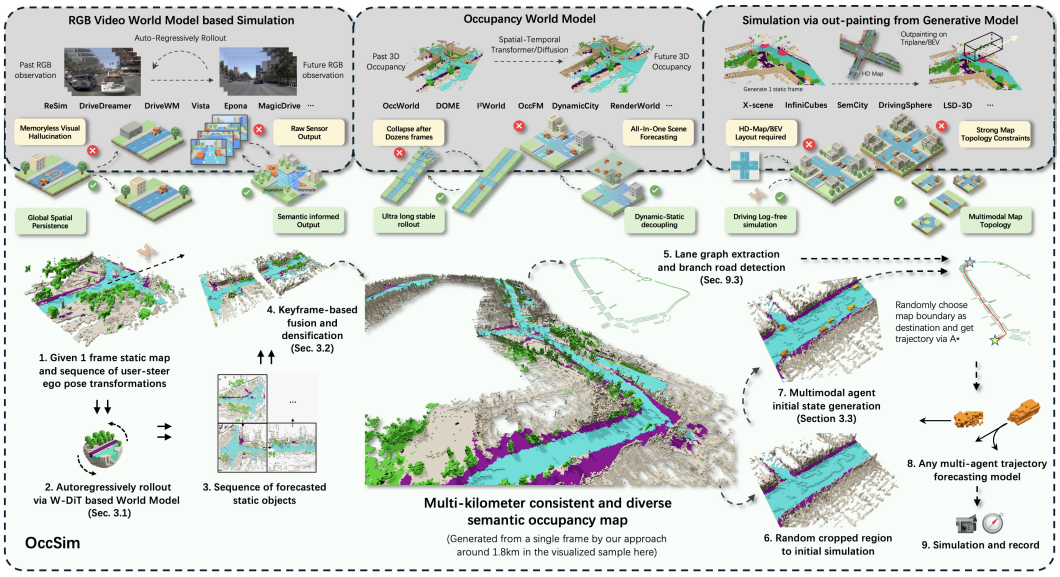


Figure 1: Comparison of OccSim and workflow. Given a single frame and a sequence of future actions, the proposed W-DiT builds a multi-kilometer consistent occupancy map for autonomous driving simulation, *without requiring HD maps or driving logs as input*. The agent initial poses are generated via the layout generator and then forward-simulated. OccSim is compatible with multi-agent forward simulation methods as plug-and-play modules; in our experiments, we use an IDM variant (Kesting et al., 2010).

long horizons. Given future actions, SoTA occupancy world models (Shi et al., 2025; Liao et al., 2025) remain stable for fewer than 50 frames (Fig. 4), which, at 2 Hz and $\sim$3 m of ego motion per frame, corresponds to a spatial span of at most 150 m. This is enough for a control decision, but two orders of magnitude short of a multi-block map ($\geq 2\,\mathrm{km}^2$). Outpainting-based methods require human-labelled HD maps (Lu et al., 2025) or trade stability for diversity (Lee et al., 2024), leaving topologically fragmented maps (Fig. 15). Dynamic generation poses a different challenge: agents must be plausibly placed and moved through the generated occupancy without a log to imitate.

We attribute this short horizon to the architecture of current stochastic 4D occupancy world models, which largely inherit design from video generation: 3D occupancy latents are stacked along the temporal dimension and aggregated by DiTs over temporal and spatial dimensions (Gu et al., 2024; Ma et al., 2024). However, most of the observed region at $t{+}1$ is a known rigid re-projection of t, yet this mapping is left to the network, so every step re-synthesizes established content instead of reusing it, spreading the generative burden over the whole latent field and letting errors accumulate quickly. Although several works on RGB video-based world models explicitly supply this transformation (Jin et al., 2025; Zhu et al., 2026), it remains unexplored for stochastic occupancy world models.

Based on this observation, we propose "W-DiT", an occupancy world model that sustains diverse rollouts of over 3,000 frames under complex trajectories (Tab. 1, 10km), an >80$\times$ improvement in stable rollout length over the previous SoTA model (Shi et al., 2025). The core mechanism of W-DiT is a pipeline designed for effective conditioning on the previous frame: W-DiT explicitly utilizes 3D geometric priors through the proposed dual-head injection mechanism. Beyond these, we further introduce an SNR-weighted occupancy perception loss that supervises the prediction in decoded occupancy space, which we find necessary to keep rollout quality stable over thousands of frames. Orthogonally, we show that a norm-invariant spherical semantic readout in the occupancy VAE reshapes the latent manifold in a way that any occupancy world model can benefit from, and which we find essential for stability.

Single-frame outputs are then consolidated into a persistent environment: a keyframe-based fusion scheme and a graph-search-based lane topology extraction algorithm yield a large-scale road network that agents can drive on. Populating it without historical logs requires a generative treatment of traffic layout, so rather than placing agents by hand-tuned spawning rules, we train a layout generator with latent flow matching to produce initial traffic states conditioned on the local road structure; agent motion is then rolled out by a replaceable control module, for which we use a 2D-IDM variant.

In summary, our contributions can be summarized as follows:

1. **A simulation pipeline that requires no logs or maps as input.** We propose a new pipeline for data-driven occupancy-based simulation: an occupancy world model generates the large-scale static scene, a layout generator infers initial agent states directly from the generated semantic occupancy, and agent motion is produced by a replaceable control module, instantiated here with a 2D-IDM variant, and compatible with any modern trajectory forecasting model.
2. **An occupancy world model stable over ultra-long horizons.** Using the same training data, our W-DiT-based backbone optimized with the proposed objective achieves a stable rollout length over **80**$\times$ longer than before. This makes W-DiT, to our knowledge, the first model to roll out over **3,000 frames** of static occupancy and form complex road topologies without relying on HD maps in any form. Under user-steered trajectories, further stress tests qualitatively show stable and diverse rollout beyond **110K frames** (Fig. 16), building a 100km+ city-scale occupancy map.
3. **An evaluation protocol for log-free environment generation.** Beyond the logged horizon no paired ground truth exists, and each individual metric admits a trivial optimum. We therefore report fidelity, diversity and road-network connectivity jointly at matched spatial scales, reference them against an explicit collapse baseline, and ground them in the sim-to-real transfer of a downstream forecasting task, an external judge the generator cannot optimise against. Under this protocol, forecasting models trained purely on OccSim data reach 67% of in-domain IoU (vs. 60% for CARLA). These numbers increase to 74% on the $5\times$ data scale, 22.1% better than CARLA.

2 Related Work

2.1 Long-horizon RGB world models

Driven by the abundance of video data, a large body of work builds driving world models directly on RGB signals. GAIA-1 (Hu et al., 2023) autoregressively generates minute-long clips with a multi-modal transformer under language and action conditioning, and GAIA-2 (Russell et al., 2025) extends this to multi-view generation by replacing the deterministic backbone with flow matching, while Cosmos-Drive-Dreams (Ren et al., 2025) finetunes a video foundation model (Agarwal et al., 2025) for driving scene synthesis. Others improve fidelity and horizon through training recipes and dynamic priors (Gao et al., 2024) or move to purely autoregressive and AR-diffusion formulations (Hu et al., 2024; Zhang et al., 2025), a paradigm since extended to unified world models that perform understanding, planning and generation in one VLM backbone (Zhou et al., 2025; Xiong et al., 2026).

2.2 Occupancy world models in autonomous driving

Occupancy representations, typically derived from voxelized and densified LiDAR, instead make 3D geometry explicit. Early works pair discrete encoding (Van Den Oord et al., 2017) with deterministic autoregressive transformers (Zheng et al., 2024; Wei et al., 2024), the latter additionally using an LLM. Generative formulations now dominate: DOME (Gu et al., 2024) combines a continuous VAE with a controllable diffusion model, COME (Shi et al., 2025) adds scene-centric forecasting control via a post-trained ControlNet (Zhang et al., 2023), DynamicCity (Bian et al., 2024) adopts hex-plane decomposition, and OccFM (Liu et al., 2025) uses a flow-matching multi-scale DiT (Peebles & Xie, 2023). UniScene (Li et al., 2025) instead conditions on BEV layouts to generate multi-sensor data. All stack latents temporally and jointly aggregate space-time, reporting only tens-of-frames horizons.

2.3 Data-driven simulation with generative models

Conventional physics-based simulators such as CARLA (Dosovitskiy et al., 2017) support interaction but suffer from domain discrepancy, motivating simulators built on real data. Nocturne (Vinitsky et al., 2022) replays Waymo trajectories, while SLEDGE (Chitta et al., 2024) and Scenario Dreamer (Rowe et al., 2025) synthesize scenarios by generatively modelling vectorized road layouts and agents. On the observation side, ReSim (Yang et al., 2025a) and DriveArena (Yang et al., 2025b) render RGB observations from HD maps for closed-loop policy evaluation, and 3D Gaussian Splatting (Chen et al., 2024; Zhou et al., 2024; Xiao et al., 2025) reconstructs static backgrounds with high realism from logged scenes. Closest to us, SemCity (Lee et al., 2024) and InfiniCube (Lu et al., 2025) extend

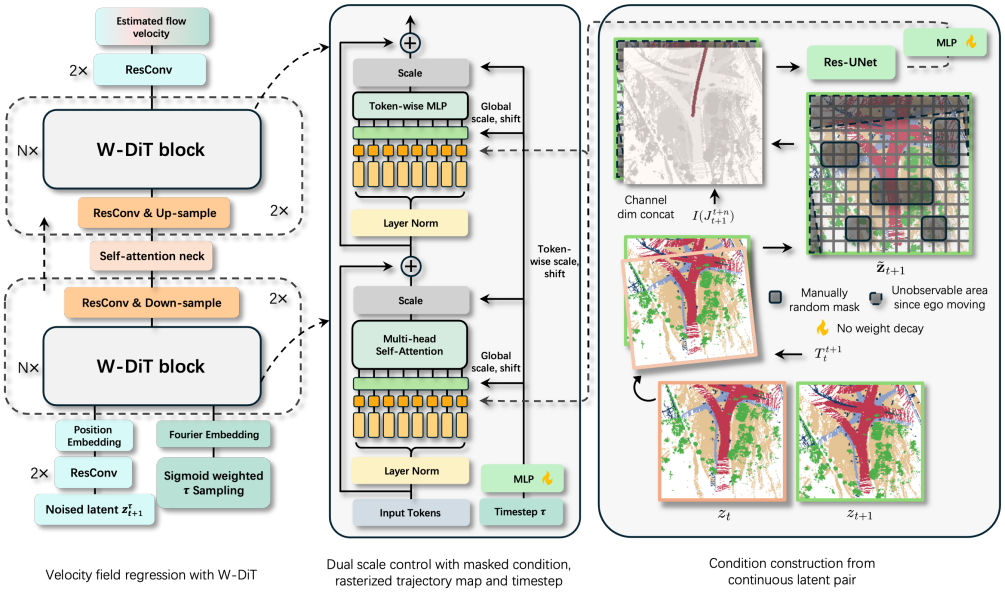


Figure 2: Illustration of structure to generate road map of $t+1$ from condition at timestep t. Here, t represents the sequence frame index, and τ denotes the probability flow timestep. This token-wise scale and shift condition injection method allows precise control of the rigid transformation during long-horizon generation. $I(\cdot)$ stands for rasterization process of given future trajectory J^{t+n}_{t+1}. The latent representations z_t and z_{t+1} ($\in \mathbb{R}^{H\times W\times C_z}$) at bottom right corner are visualized as decoded BEV occupancy maps for intuitive illustration.

occupancy spatially by outpainting (Lugmayr et al., 2022) with triplane diffusion, and X-Scene (Yang et al., 2026) generates BEV layouts from text prompts. Each of these still draws its environment from an external source: recorded log, HD map, or layout to be outpainted. Therefore, the environments available for simulation remain bounded by annotation effort rather than by generation.

3 Long-horizon Occupancy World Models

OccSim organizes simulation into two loops on different timescales. The *generation loop* runs once: W-DiT rolls out static occupancy along a trajectory and fuses the frames into a single frozen global map. The *simulation loop* then runs online at every timestep: the ego and background agents re-plan from local states, querying rather than regenerating the static environment. Sec. 3.1 presents W-DiT and its optimization, Sec. 3.2 fuses frames into loop-closing city-scale maps, and Sec. 3.3 shows the layout generator that populates them.

3.1 Long-horizon stable generation with W-DiT

We find that a geometry-aware alternative to the standard 2+1D aggregation yields substantial improvements in long-term stability. Formally, using an occupancy VAE (Liu et al., 2025) (consisting of spatial compressor $\mathcal{E}$ and decoder $\mathcal{D}$), we can map the occupancy frames $\mathcal{O}_t, \mathcal{O}_{t+1}$ to their respective latent representations $\mathbf{z}_t, \mathbf{z}_{t+1} \in \mathbb{R}^{H\times W\times C_z}$, where t denotes time index. Our aim is to model the joint predictive distribution over an extended future horizon: given an initial latent state $\mathbf{z}_0$ and a sequence of future ego-trajectories $J_{0:K+\sigma}$, our goal is to auto-regressively estimate:

$$P(\mathbf{z}_{1:K}|\mathbf{z}_0, J_{0:K+\sigma}) = \prod_{i=1}^{K} P_\theta(\mathbf{z}_i|\mathbf{z}_{i-1}, J_{i-1:i+\sigma-1}), \tag{1}$$

where K is the forecast horizon and $K+\sigma$ the trajectory length. For $K \geq 10^2$, the per-step residual of a learned re-projection compounds into structural drift. The W-DiT block (Fig. 2) instead conditions each step on a deterministic rigid transformation derived from J, warping the latent so that only the newly revealed region is generated. Specifically, assuming planar motion on the XY-plane

(zero vertical velocity and roll/pitch), the inter-frame transformation is $\mathbf{T}_t^{t+1} = \exp(\widehat{\xi}_t \Delta t)$, where $\widehat{\xi}_t \in \mathbb{R}^{4\times4}$ is the matrix form of the twist induced by the action $a_t = [v_{x,t}, v_{y,t}, \omega_t]^\top$, with planar velocity $\mathbf{v} = [v_{x,t}, v_{y,t}, 0]^\top$ and yaw rate $\boldsymbol{\omega} = [0, 0, \omega_t]^\top$.

Applying $\mathbf{T}_t^{t+1}$, we forward-warp $\mathbf{z}_t$ to the viewpoint at $t+1$, obtaining $\hat{\mathbf{z}}_{t+1}$. Ego motion inevitably pushes some spatial regions out of the field of view or into occlusion, so we deterministically compute a binary visibility mask $\mathbf{M}_{\text{vis}} \in \{0,1\}^{H\times W}$ from the known geometry alone. However, the inter-frame displacement is small (3–4 m at $\Delta t = 0.5$ s), so $\mathbf{M}_{\text{vis}}$ alone leaves the task nearly degenerate: the model can satisfy it by copying the input, which hinders robust representation learning. Following MAE (He et al., 2022), we therefore add a random mask $\mathbf{M}_{\text{rand}} \sim \mathcal{B}(1-p)$ with masking ratio p, and condition on $\tilde{\mathbf{z}}_{t+1} = (\mathbf{M}_{\text{vis}} \odot \mathbf{M}_{\text{rand}}) \odot \hat{\mathbf{z}}_{t+1}$, where $\odot$ is the Hadamard product and the spatial masks broadcast across C_z. Next, instead of fusing the future action into τ directly (Gu et al., 2024; Shi et al., 2025), we rasterize it into a spatial signal aligned with index $t+1$. We represent the future actions as physical waypoints $P_t = \{p_{t+i} \in \mathbb{R}^2\}_{i=1}^{\sigma}$ on the BEV plane, anchored in the ego frame at $t+1$ by cumulatively applying $\mathbf{T}_{t+1}^{t+\sigma}$. Connecting them yields a polyline tracing the intended path onto the future road surface, which we rasterize to $l_{t+1} \in \{0,1\}^{H\times W\times 1}$ (denoted $I(\cdot)$ in Fig. 2). Since warping introduces interpolation error, we concatenate l_{t+1} with $\tilde{\mathbf{z}}_{t+1}$ and refine them with a U-Net before injection, yielding the conditioning tokens $\hat{f}_{t+1} \in \mathbb{R}^{B\times N\times C_w}$ with $N = H \times W$.

We then inject $\hat{f}_{t+1}$ by token-wise scaling and shifting. Global scale, shift and gate are regressed from the flow timestep τ as in DiT, while a token-wise projection regresses a spatial control signal from $\hat{f}_{t+1}$ (dual-head injection hereafter); the two are summed per token:

$$[\gamma_g, \beta_g, \alpha_g] = \text{MLP}_{\text{time}}(\tau), \quad [\Gamma, B, A] = \text{MLP}_{\text{cond}}(\hat{f}_{t+1}), \quad \gamma^{(i)} = \gamma_g + \Gamma^{(i)}, \tag{2}$$

and analogously for $\beta^{(i)}, \alpha^{(i)}$. On the main stream, $\mathbf{z}_{t+1}^{\tau} = (1-\tau)\epsilon + \tau \mathbf{z}_{t+1}$ is tokenized to $f_{t+1} \in \mathbb{R}^{B\times N\times C_w}$, and the MHSA block becomes $f'_{t+1} = f_{t+1} + \alpha_1 \odot \text{MHSA}(\gamma_1 \odot \text{LN}(f_{t+1}) + \beta_1)$, where $\gamma_1 = [\gamma^{(1)}, \ldots, \gamma^{(N)}]^\top \in \mathbb{R}^{N\times C_w}$ stacks the token-wise parameters, with β_1, α_1 defined analogously and the same scheme applied in the subsequent MLP layer. Each W-DiT block therefore modulates every spatial token by its own warped condition, and the network outputs the instantaneous velocity field $v_\theta(\mathbf{z}_{t+1}^{\tau}, \tau, c)$, where c denotes all conditions.

Following the success of flow models in high-quality generation with few sampling steps (Song et al., 2025; Zhang et al., 2025), we optimize W-DiT with a velocity regression objective. This objective is sound but not sufficient for long-horizon rollout: supervising only in the continuous latent space leaves errors that are small in latent norm yet consequential once decoded, which will be re-injected at every autoregressive step. Empirically, models trained with the velocity loss alone match the full objective on the first generated frame but degrade within 10 rollout frames (Tab. 4).

We therefore introduce an SNR-scaled perception loss. Our insight is to exploit the discrete, semantic nature of 3D occupancy: rather than confining supervision to the continuous latent manifold, we project the denoised prediction back into occupancy space, redirecting gradient pressure (from CE loss) onto the directions the decoder is sensitive to and anchoring it to a target that does not drift with the latent. The overall objective is

$$\mathcal{L}_{\text{total}}(\theta) = \mathbb{E}_{\tau,\epsilon,z_{t+1},\mathbf{c}}\left[\|v_\theta - (z_{t+1} - \epsilon)\|_2^2 + \lambda \tau^2 \frac{\mathbf{M}_{\text{mask}}}{|\mathbf{M}_{\text{mask}}|} \text{CE}\Big(\mathcal{O}_{t+1}, \mathcal{D}(z_{t+1}^{\tau} + (1-\tau)v_\theta)\Big)\right] \tag{3}$$

where $\mathbf{M}_{\text{mask}} = \mathbf{M}_{\text{vis}} \odot \mathbf{M}_{\text{rand}}$, λ balances the two terms, and $\hat{\mathbf{z}}_{t+1} = \mathbf{z}_{t+1}^{\tau} + (1-\tau)v_\theta$ is the one-step denoised latent recovered from the current velocity field without sampling, decoded by the frozen VAE $\mathcal{D}$ for a mask-region-only cross-entropy against $\mathcal{O}_{t+1}$. Crucially, the τ^2 weighting concentrates this supervision in the low-noise regime, where the one-step reconstruction is close enough to the target for the cross-entropy to carry usable geometric signal. We ablate the usage of perception loss and SNR weight in rows 4 and 5 of Tab. 4.

3.2 Multi-kilometer road map fusion from single frames

W-DiT outputs remain discrete, ego-centric local scenes, and naive aggregation such as temporal averaging over-smooths crucial high-frequency details, because the forecasted frames are not exactly temporally consistent. We therefore fuse them in two passes (Alg. 2 in Sec. 9): given local predictions

and their global poses, a minimally overlapping keyframe subset first establishes a sharp foundational map $\mathcal{M}_{\text{global}}$; then, after gravity alignment, the redundant non-keyframes inpaint the remaining spatial gaps via threshold-based voting and morphological cleaning.

Further expansion requires knowing where to extend. We skeletonize the road manifold to extract a topological graph and isolate candidate branch points, then reject collision-prone directions with a dual-probe filter combining topological and 3D semantic probes (Alg. 3, Sec. 9). Connecting two valid disjoint endpoints via a collision-free spline gives a trajectory along which we prompt W-DiT to autoregressively synthesize the missing occupancy, closing the loop and growing a single prompt frame into an arbitrarily extended road network.

3.3 Agent Generation and Control on Top of Static Road Map

Since our topologies are procedurally generated rather than replayed, populating $\mathcal{M}_{\text{global}}$ (whose lane graphs are extracted via Alg. 4) calls for a generative treatment of traffic layout. Formally, we model the agent layout conditioned on one frame of static environment, $P(\mathcal{A} \mid \mathbf{z}_t)$, where $\mathcal{A} = \{(x_i, y_i, s_i)\}_{i=1}^{N}$ collects vehicle centers and motion states $s_i \in \{\text{dynamic}, \text{static}\}$, and $\mathbf{z}_t$ is cropped from the fused map at a random pose. To make this tractable for a flow-based framework, we map the discrete set $\mathcal{A}$ to a continuous 2D heatmap $\mathcal{H}$, encoding static and dynamic vehicles as positive and negative Gaussian kernels, and train a compact DiT-S to learn $P(\mathcal{H} \mid \mathbf{z}_t)$ under spatial augmentations that prevent it from memorizing the layouts of a small dataset (Sec. 6.3).

At inference, $\mathcal{H}$ is discretized by NMS into vehicle instances, each projected onto the nearest lane for a kinematically feasible initial yaw (Fig. 13). Motion is then rolled out by a replaceable controller, instantiated with A* routing and a 2D-IDM variant (Kesting et al., 2010) driving agents toward randomly selected exits on the map boundary (Alg. 6). Since the interface consumes only lane graph and agent poses, any learned planner can be substituted without changing the environment.

4 Experiment

We organized our experiments to answer the following 3 questions: Q1. **Static Realism & Stability**: Can the proposed W-DiT structure generate highly realistic and spatiotemporally consistent static environments, especially over extended horizons? Q2. **Generation Diversity**: Does OccSim exhibit sufficient multimodality and diversity in long-horizon rollout? Q3. **Dynamic Fidelity & Downstream Utility**: Can OccSim be used to train temporal tasks like 4D semantic occupancy forecasting more effectively than data collected from CARLA?

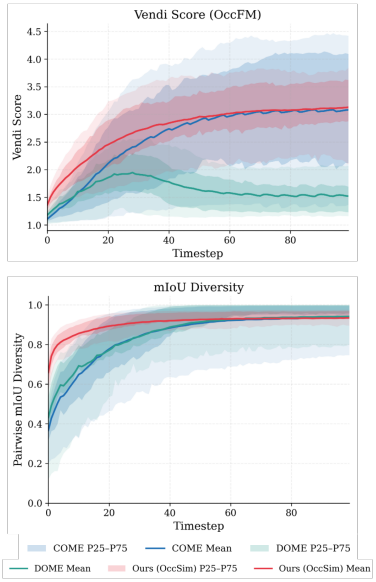


Figure 3: W-DiT's mean generation diversity over the first 100 timesteps (red) is highest. Solid lines and shaded bands denote the mean and the 25th–75th percentiles across initial scenes using 10 samples per initial scene.

4.1 Metrics Protocol Design

No single metric suffices for log-free generation: beyond the logged horizon no paired ground truth exists, so realism can only be measured against the marginal distribution, which a model repeatedly emitting one plausible scene would satisfy, while diversity scores are themselves inflated by structural breakdown. We therefore report fidelity, diversity and road-network connectivity jointly, reference them against an explicit collapse baseline, and ground the protocol in an external downstream task (Q3) that the generator cannot optimise against. Training and dataset details can be found in Sec. 6.1.

Static realism (Q1). Following Bian et al. (2024) and Yang et al. (2026), we measure 2D/3D-FID, KID and MMD (denoted Dis later) in two settings. *Conditional fidelity*, $\text{Dis}\big(P_\theta(\hat{\mathcal{O}}_t \mid \mathcal{O}_0, J), P_{\text{data}}(\mathcal{O}_t \mid \mathcal{O}_0, J)\big)$, scores structural accuracy against the strictly paired frame at each $t \in \{1, \ldots, K\}$ within the GT sequence. *Unconditional realism*, $\text{Dis}\big(P_\theta(\hat{\mathcal{O}}_t \mid \mathcal{O}_0, J), P_{\text{data}}(\mathcal{O})\big)$, compares generated samples

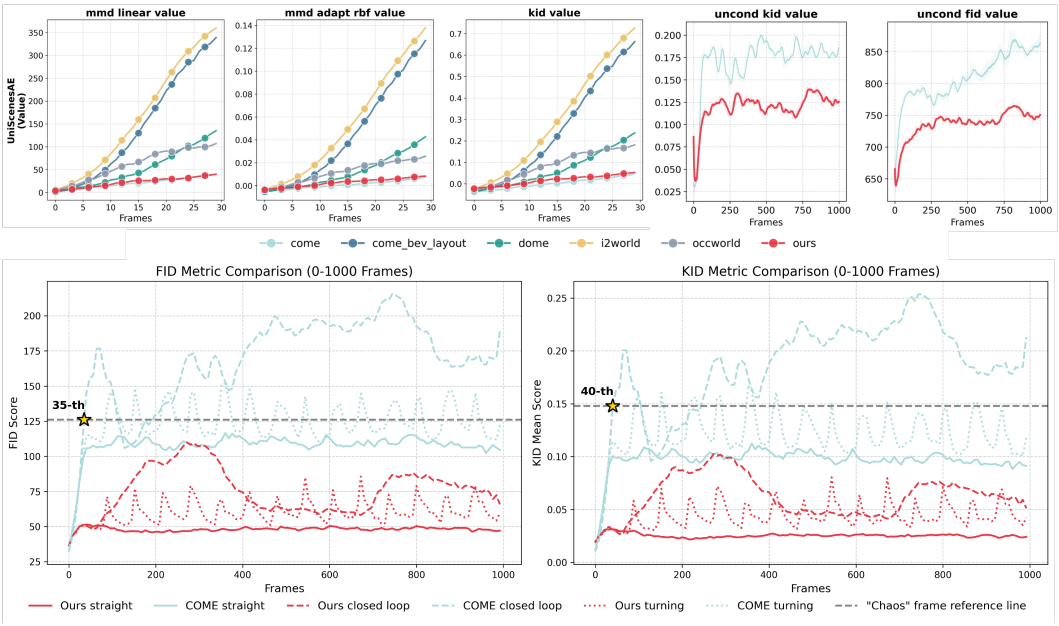


Figure 4: **Realism of generated static occupancy.** Our rollout matches the previous SoTA over the short paired horizon, but is the only one that stays stable at scale. *Top*: 3D evaluation in the UniScene AE latent space. The left three panels report conditional fidelity against the strictly paired GT frame over a 30-frame horizon; the right two report unconditional realism against the marginal distribution of the training set over a 1000-frame rollout. *Bottom*: 2D realism of the projected occupancy under three ego-action:straight, closed-loop and continuous turning. Compared against COME. The dashed line marks the "chaos" threshold characterizing scene collapse (Sec. 8.7); stars mark where COME first crosses it. Our rollout (red) stays strictly below the threshold under every action. The same 3D evaluation in the OccFM VAE latent space is reported in Sec. 7.3 and trajectory shapes in Fig. 12.

against the global marginal of the training set: beyond the logged horizon a rollout may legitimately diverge into a different valid structure, and only its residence on the data manifold can be asked of it.

Diversity (Q2). The Semantic IoU diversity score is the complement of the average pairwise IoU across stochastic rollouts from the same initial scene, measured in decoded semantic space; Vendi score (Friedman & Dieng, 2022) measures the effective sample size of the generated distribution in latent feature space. Both are reported at sample counts N, as Vendi score is upper-bounded by N.
Downstream transfer (Q3). We pre-train 4D semantic occupancy forecasting models on continuous ego-centric streams collected inside our simulation and evaluate them zero-shot on real data. This makes the simulator's fidelity to real traffic dynamics answerable by an external judge.

4.2 Main Results Comparisons

Static realism and stability. Fig. 4 evaluates realism in both settings of Sec. 4.1. Since fewer than 1000 GT samples exist per timestep t, we use the unbiased KID and MMD for conditional fidelity; the 1000-frame rollout continues past the end of the GT sequence along a straight trajectory at constant velocity. Our method attains the best long-horizon stability and structural realism in 3D, while the baselines accumulate error rapidly, and the same ordering holds in the OccFM VAE latent space (Sec. 7.3), indicating the result is not an artifact of a particular autoencoder. Compared with COME, the previous SoTA diffusion-based occupancy world model, our method on the curved and closed-loop trajectories still scores below COME on the easier straight-line case, while COME crosses the chaos threshold within 40 frames. COME also receives $\mathbf{T}_t^{t+1}$ through future poses, so the comparison involves no privileged information.

Diversity. Fig. 3 tracks pairwise mIoU diversity and Vendi scores over the first 100 timesteps. Our method rises rapidly at early timesteps, opening a clear margin over COME and DOME, and then holds a high, stable level as the rollout extends, achieving the highest Vendi score.

Table 1: Diversity and fidelity comparison with other non-world-model occupancy generation methods across distances. SemCity (Lee et al., 2024) extends map boundaries via unconditional diffusion outpainting, where quality degrades with extrapolation distance. UniScene (Li et al., 2025)† requires an aligned per-frame HD-Map to generate occupancy and is thus distance-independent; we treat it as a performance upper bound; results of condition-free version reported below. ⋆: SemCity is not officially trained on Occ3D dataset, we reproduced results with the official code. LCC: largest-connected-component ratio, measuring the road-network completeness of the stitched map. All experiments are run under **5 different seeds** and 150 scenes then averaged. 3 m per step used beyond the GT trajectory (10km ≈ 3300 frames). Details and visualization can be found in Sec. 6.1 and 8.4.

Model	Dataset	HD-Map?	2D-FID ↓ / Mean Vendi Score at different physical distance ↑						LCC. Ratio ↑
			40m	400m	1km	2km	5km	10km	
SemCity⋆	Occ3d-NuScenes	✗	46.31/1.00	97.07/2.03	102.01/2.04	147.36/2.06	141.96/1.93	148.29/1.95	11.70%
UniScene†		✓	33.12/1.55						–
UniScene		✗	215.59/3.82						–
Ours		✗	36.48/1.24	50.00/2.05	50.89/2.22	46.40/2.27	48.60/2.29	47.13/2.35	92.90%

Comparison with outpainting. Tab. 1 compares against non-world-model occupancy generation. Temporal rollout faces error accumulation that spatial outpainting does not, yet on Occ3D-nuScenes W-DiT still surpasses the outpainting baseline in both fidelity and diversity, with the gap widening with distance. The LCC ratio shows the structural difference: SemCity's stitched maps remain fragmented where ours stay almost fully connected. The unconditional UniScene row illustrates why these axes must be read jointly: its Vendi score exceeds every other method, while its FID places it near collapse. The 10km in Tab. 1 is the limit we can assess quantitatively in our compute budget; qualitatively, stability holds an order of magnitude further, and Sec. 8.5 presents a 100km+ map, to our knowledge the largest reported in occupancy-based driving scene generation.

Downstream utility. We use 4D semantic forecasting as a proxy for the value of OccSim as a data generator. Unrolling 3,000 frames along a curved closed-loop trajectory gives a substantially larger map, on which we build the simulation environment via Sec. 3.2 and Sec. 3.3. In Tab. 3, OccWorld trained on OccSim data outperforms its CARLA-trained counterpart (Uni-C) by 14% relative average mIoU at matched volume and by 52% with 5× data, reaching 67% and 74% (1× / 5×) of the in-domain IoU averaged over 1-3s. The same ordering holds for the open-loop planning L2 of OccWorld's ego trajectory head, all using a non-learned IDM controller. Augmenting the full real Waymo logs with an extra 30% of OccSim data improves the average 1–3s IoU by a further 7% relative (Sec. 7.7), showing OccSim also complements real data.

Table 2: Agent motion realism under our log-free setting. We adopt the behaviour-simulation JSD protocol of Scenario Dreamer (Rowe et al., 2025) (Lin. Spd., Ang. Spd., Acc., Near. Dist.). Most of the discrepancy comes from IDM control and layout initialization rather than the generated occupancy maps. **GT vs. GT split** (real trajectories randomly split in half) serves as an empirical lower bound; **IDM-on-real-map** isolates the effect of our generated road topology by running the *same* IDM controller on real lane graphs from GT occupancy. Behav. and Occ. : Agent action behavior and occupancy used to extract lanes for IDM control.

Layout	Behav.	Occ.	Distributional JSD ($\times 10^{-2}$) ↓			
			Lin. Spd.	Ang. Spd.	Acc.	Near. Dist.
GT split (l.b.)	GT	GT	1.42	0.05	7.61	0.15
GT	IDM	GT	5.91	0.13	11.76	0.7
Layout Gen.	IDM	GT	8.13	0.35	17.9	1.7
Layout Gen. (Ours)	IDM	Gen.	8.76	0.35	18.6	1.4

4.3 Ablation Study

W-DiT design. Tab. 4 shows the three components are only effective jointly. Token-wise condition injection alone (row 3) does not stabilise the rollout and is in fact worse than the plain baseline at intermediate horizons; adding the perception loss on top of it (row 4) improves the 500-frame endpoint but still leaves quality degrading monotonically. Only when the perception loss is SNR-scaled (row 5) does the rollout become stable, halving the 500-frame FID relative to row 3 and keeping it within 12% of the first generated frame. This matches the role of the τ^2 weighting: it concentrates occupancy-space supervision in the low-noise regime, where the decoded prediction is close enough to the GT for the cross-entropy term to carry geometric signal.

Agent motion realism. Following Scenario Dreamer (Rowe et al., 2025), Tab. 2 reports the JSD between simulated and real agent motion, with a GT-vs-GT as an empirical lower bound. We

Table 3: The performance of different semantic forecasting models trained with occupancy collected from UniOcc-CARLA (Uni-C) or OccSim, tested on the UniOcc-nuScenes (Uni-N) validation set. At matched data volume, pre-training on OccSim outperforms CARLA on every metric and for both forecasters, and the margin keeps growing when OccSim data is scaled $5\times$. mIoU is measured over all background occupancy categories, and over the car category for foreground. †: we collected the same amount of data on OccSim as UniOcc provides for CARLA, for fairness. ⋆: uses $5\times$ the amount of data collected from OccSim to train the downstream models. Stoch.: whether the model is stochastic. Rows 1 and 5 are trained and tested on the same domain without domain shift, which is the upper bound. We omit the L2 error for OccFM here, as it is future-action conditioned.

Model	Train Data	mIoU ↑			IoU ↑			L2 (m) ↓			IoU_{veh}	Stoch.
		1s	2s	3s	1s	2s	3s	1s	2s	3s		
OccWorld	Uni-N	26.53	16.73	12.33	32.17	22.88	18.41	0.43	1.08	1.99	17.10	✗
OccWorld	Uni-C	11.79	8.35	6.75	18.31	14.09	12.03	0.72	2.31	4.05	6.58	✗
OccWorld†	OccSim	13.51	9.56	7.61	20.53	15.48	13.24	0.65	1.42	3.27	7.46	✗
OccWorld⋆	OccSim	19.54	12.31	9.08	23.87	16.92	13.47	0.61	1.38	2.91	12.41	✗
OccFM	Uni-N	36.28	25.10	19.47	41.22	30.90	25.02	–	–	–	28.79	✓
OccFM	Uni-C	6.99	5.34	4.70	8.63	6.50	5.71	–	–	–	6.42	✓
OccFM†	OccSim	15.93	11.76	9.26	18.31	14.10	11.75	–	–	–	13.28	✓
OccFM⋆	OccSim	25.32	18.96	13.10	29.19	21.08	13.93	–	–	–	15.82	✓

Table 4: W-DiT design ablation, reported as the UniScene-AE 3D unconditional realism FID score at the n-th generated frame. All three proposed components are essential for long-horizon stability. When not using dual-head injection, we adopt the classical spatiotemporal architecture in DOME.

Architecture	Percep. Loss	Dual-head injection	SNR scaled	1st	5th	10th	30th	50th	100th	500th
Spatiotemporal DiT	✗	✗	✗	663.89	712.56	789.91	1023.51	1281.99	1587.21	1612.40
	✓	✗	✓	661.79	710.63	801.11	1034.75	1342.72	1601.15	1607.97
W-DiT	✗	✓	✗	672.15	756.28	1013.12	1335.88	1295.42	1378.56	1526.95
	✓	✓	✗	696.13	833.17	946.23	1306.21	1415.59	1427.66	1440.57
	✓	✓	✓	665.23	630.34	645.96	687.90	698.87	712.65	744.71

decompose the remaining gap in three steps. Replacing real trajectories with IDM control on lane graphs extracted from GT occupancy (row 2) yields the largest increase. This step bundles two costs: the simplification of the IDM itself, and the representational cost of recovering topology from voxelized semantics rather than from a vectorized map; we analyze the latter in Sec. 7.4, where lane graphs extracted from GT semantic occupancy are compared against those rasterized from HD maps and the residual Chamfer discrepancy remains sub-voxel relative to the 0.4m grid. Substituting GT agent positions with our Layout Generator (row 3) adds a smaller increment, so initialization is a secondary error source. Most importantly, moving from real to generated occupancy (row 4) changes the JSD only marginally across all four behaviour statistics, and is in fact lower on nearest-distance: our map topology is therefore not the limiting factor, and the remaining gap is dominated by the representation-and-controller stage, i.e. exactly the component our pipeline exposes as replaceable.

Further analyses are reported in the appendix. Sec. 7.1 gives the end-to-end cost of building and then running an environment, and Sec. 7.2 shows the closed-loop collision rate under different agent counts and initialization schemes. Sec. 7.5 ablates the number of function evaluations (NFE) jointly with the single-step rollout stride. Sec. 7.6 analyses the semantic readout of our VAE, which we find crucial for long-horizon stability: as the design is orthogonal to the world model itself, other occupancy world models can also benefit at negligible cost.

5 Conclusion

We present OccSim, the first log-free occupancy world model-driven simulator, generating diverse, realistic rollouts spanning thousands of frames without any BEV layout or HD map as input. Simulators built on these generated topologies replicate driving dynamics with substantially higher fidelity than traditional simulators. As our framework is compatible with learned policies, future work could replace IDM with data-driven control and extend generation to high-fidelity multi-modal output.

REFERENCES

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.

Hengwei Bian, Lingdong Kong, Haozhe Xie, Liang Pan, Yu Qiao, and Ziwei Liu. Dynamiccity: Large-scale 4d occupancy generation from dynamic scenes. *arXiv preprint arXiv:2410.18084*, 2024.

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.

Qifeng Chen, Sheng Yang, Sicong Du, Tao Tang, Rengan Xie, Peng Chen, and Yuchi Huo. Lidar-gs: Real-time lidar re-simulation using gaussian splatting. *arXiv preprint arXiv:2410.05111*, 2024.

Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *European Conference on Computer Vision*, pp. 57–74. Springer, 2024.

Felipe Codevilla, Antonio M Lopez, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 236–251, 2018.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In A. D. Dragan, M. Hebert, and S. Levine (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16, Mountain View, CA, USA, 13–15 Nov 2017. PMLR. URL http://proceedings.mlr.press/v78/dosovitskiy17a.html.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

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

Songen Gu, Wei Yin, Bu Jin, Xiaoyang Guo, Junming Wang, Haodong Li, Qian Zhang, and Xiaoxiao Long. Dome: Taming diffusion model into high-fidelity controllable occupancy world model. *arXiv preprint arXiv:2410.10429*, 2024.

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*, pp. 16000–16009, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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.

Bu Jin, Weize Li, Baihan Yang, Zhenxin Zhu, Junpeng Jiang, Huan-ang Gao, Haiyang Sun, Kun Zhan, Hengtong Hu, Xueyang Zhang, et al. Posepilot: Steering camera pose for generative world models with self-supervised depth. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8051–8058. IEEE, 2025.

Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. SemCity: Semantic scene generation with triplane diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28337–28347, 2024.

Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11971–11981, 2025.

Zhimin Liao, Ping Wei, Ruijie Zhang, Shuaijia Chen, Haoxuan Wang, and Ziyang Ren. I2-world: Intra-inter tokenization for efficient dynamic 4d scene forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25810–25819, 2025.

Tianran Liu, Shengwen Zhao, and Nicholas Rhinehart. Towards foundational lidar world models with efficient latent flow matching. *arXiv preprint arXiv:2506.23434*, 2025.

Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27272–27283, 2025.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

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

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17207–17218, 2025.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Yining Shi, Kun Jiang, Qiang Meng, Ke Wang, Jiabao Wang, Wenchao Sun, Tuopu Wen, Mengmeng Yang, and Diange Yang. Come: Adding scene-centric forecasting control to occupancy world model. *arXiv preprint arXiv:2506.13260*, 2025.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, et al. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Eugene Vinitsky, Nathan Lichtlé, Xiaomeng Yang, Brandon Amos, and Jakob Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *Advances in Neural Information Processing Systems*, 35:3962–3974, 2022.

Yuping Wang, Xiangyu Huang, Xiaokang Sun, Mingxuan Yan, Shuo Xing, Zhengzhong Tu, and Jiachen Li. Uniocc: A unified benchmark for occupancy forecasting and prediction in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25560–25570, 2025.

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

Haihong Xiao, Jianan Zou, Yuxin Zhou, Ying He, and Wenxiong Kang. Splatco: Structure-view collaborative gaussian splatting for detail-preserving rendering of large-scale unbounded scenes. *arXiv preprint arXiv:2505.17951*, 2025.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. *arXiv preprint arXiv:2601.04453*, 2026.

Tianyi Yan, Dongming Wu, Wencheng Han, Junpeng Jiang, Xia Zhou, Kun Zhan, Cheng-zhong Xu, and Jianbing Shen. Drivingsphere: Building a high-fidelity 4d world for closed-loop simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27531–27541, 2025.

Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger, Xiangyu Yue, and Li Chen. Resim: Reliable world simulation for autonomous driving. *arXiv preprint arXiv:2506.09981*, 2025a.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, et al. Drivearena: A closed-loop generative simulation platform for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26933–26943, 2025b.

Yu Yang, Alan Liang, Jianbiao Mei, Yukai Ma, Yong Liu, and Gim Hee Lee. X-scene: Large-scale driving scene generation with high fidelity and flexible controllability. *Advances in Neural Information Processing Systems*, 38:104415–104451, 2026.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.

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. *arXiv preprint arXiv:2506.24113*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

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

Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27817–27827. IEEE, 2025.

Yiyao Zhu, Ying Xue, Haiming Zhang, Guangfeng Jiang, Wending Zhou, Xu Yan, Jiantao Gao, Yingjie Cai, Bingbing Liu, Zhen Li, et al. Dlwm: Dual latent world models enable holistic gaussian-centric pre-training in autonomous driving. *arXiv preprint arXiv:2604.00969*, 2026.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks requiring disclosure (i.e., generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results), as none of these apply to our use of AI in this work. Additionally, we used generative AI tools for editing the paper to improve readability and for creating and editing parts of the software code (tasks with recommended disclosure). We have reviewed all AI-assisted work: all AI-edited text was checked and revised by the authors for accuracy and tone, and all AI-generated or AI-assisted code was tested and verified for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly released datasets and does not involve any new data collection from human subjects. We have no conflicts of interest or sponsorship arrangements to disclose that could have influenced this research.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The full implementation details of our proposed method are described in Section 3. Training details, including hyperparameters and optimization settings, are provided in Appendix 6.1. Details of our data processing and simulation/auto-labeling pipeline are described in Appendix 6.5. Additional ablation studies and experimental settings that further support our results are included in Appendix 7. We will release the source code publicly upon acceptance of this paper to facilitate reproduction of our results.

In this appendix, we supplement the following materials to support the findings and conclusions drawn in the main body of this paper.

CONTENTS

6 Training and metric details **16**

6.1 Details of OccSim configuration 16

6.2 Training details of baseline reproduced 16

6.3 Augmentation in Layout generator 16

6.4 Additional metrics details 17

6.5 More simulation pipeline details 17

7 Additional ablation experiments **18**

7.1 Inference speed and condition comparison 18

7.2 Closed-loop interaction safety 19

7.3 3D fidelity evaluation under OccFM manifold 19

7.4 Validating the lane extraction heuristic against HD maps 20

7.5 More details in few steps sampling & single step rollout stride 21

7.6 Spherical readout: a norm-invariant semantic head 21

7.7 Complementing Real Logs with OccSim Data 23

8 Additional visualization **23**

8.1 Details of trajectory shapes used in Fig. 4 24

8.2 Agent initialization within generated static maps 24

8.3 Comparison with other Occ. World Models in long-horizon rollouts 25

8.4 Comparison with out-painting based methods 25

8.5 Ultra long rollout from W-DiT: 100+km city scale generation with real world trajectory. 26

8.6 Multi-modal diversity of W-DiT: fix-seed multi-time rollout results 27

8.7 Visualization of chaos baseline constructed 27

9 Algorithm details **28**

9.1 Heuristic for Single frame fusion. 28

9.2 Branch road detection and topology extraction. 30

9.3 Lane topology extraction. 30

9.4 2D-IDM based agent simulation engine. 32

10 Potential limitation and Future work **34**

6 TRAINING AND METRIC DETAILS

6.1 DETAILS OF OCCSIM CONFIGURATION

For training of OccSim, Occ3D-nuScenes (Tian et al., 2023) is used for all static realism comparisons to equalise training data volume; Considering the training cost, for performance reported in Fig. 4, we use the public released weight trained on original Occ3d NuSc dataset (with dynamic agent) for all conditional evaluation (First 3 figures on the top), where we can see a close performance between W-DiT and COME. To fairly compare W-DiT with the previous SoTA occupancy world model, we used the preprocessed static occupancy retrained COME in all unconditional experiments in Fig. 4.

For Tab. 3, OccSim is trained on UniOcc-Waymo (Wang et al., 2025), and downstream forecasting models trained on the data collected from it are evaluated on the UniOcc-nuScenes validation set. We adopt UniOcc labels rather than Occ3D for their unified voxel categorization; since no Occ3D-nuScenes label is used to train OccSim, no label leakage occurs at test time. The OccSim-collected data (†) matches UniOcc-Carla in volume, and $5\times$ ($\star$) denotes five times this volume collected from OccSim, with OccSim's own training data unchanged.

The two components OccSim's: W-DiT and the Layout Generator, was conducted separately as follows. For W-DiT, we trained for 200 epochs on 4 Blackwell RTX Pro 6000 GPUs with a batch size of 32. Training on the Nuscenes dataset took approximately 100 GPU hours. For the Layout Generator, we trained for 500 epochs on 8 A100 80G GPUs, taking approximately 160 GPU hours. All experiments used AdamW with cosine annealing, which decayed the learning rate from 3.2e-5 to 3.2e-6.

In Eq. 3, we set $\lambda = 2$, with 20% probability to drop condition during training for classifier-free guidance (Ho & Salimans, 2022) in sampling stage. When generating $\mathbf{M}_{\text{rand}}$, we control the maximum of overall masked region $\mathbf{M}_{\text{mask}}$ between 10% to 50% of the full latent size. τ in Eq. 3 is sampled from sigmoid($\mathcal{N}(0, I)$). In all W-DiT-based rollout experiments described in the main text, we used NFE=10, CFG=1.1, and a roll-out of 3m per step. We further analyze these hyperparameters in Sec. 7.5.

6.2 TRAINING DETAILS OF BASELINE REPRODUCED

For SemCity (Lee et al., 2024) reproduced, we first retrain the triplane autoencoder in existing codes on preprocessed Occ3d-Nusc datasets (filter all of dynamic agents) for 30 epochs, the rIOU/rMIOU is 86.07%/76.7% for Occ3d-Nusc. Then the triplane diffusion was trained with these triplane representations for 50000 steps and 70000 steps for Occ3d-Nusc and Occ3d-Waymo, with batch size 18, initial learning rate 5×10^{-4} linearly annealed to 0, EMA rate 0.9999. Final training losses were 0.126–0.129 (vs. ~ 0.78 at initialization) for both datasets. All other architectural hyperparameters (embedding dim, MLP depth, triplane channel width, DDPM schedule) match the original SemCity configuration.

6.3 AUGMENTATION IN LAYOUT GENERATOR

Theoretically, the distribution of valid vehicle behaviors given a specific map representation, denoted $P(\mathcal{H}|\mathbf{z}_t)$, is inherently highly multimodal, as various driving intentions can perfectly align with the same static environment. However, direct optimization on a finite training dataset $\mathcal{D} = \{(\mathbf{z}_t^{(i)}, \mathcal{H}^{(i)})\}_{i=1}^{N}$ poses a severe risk of overfitting, as the number of samples is only 28K to train the DiT-S. In practice, for any specific scene $\mathbf{z}_t^{(i)}$ in the log, we only observe a single deterministic realization of the vehicles' behavior $\mathcal{H}^{(i)}$. Consequently, without proper regularization, model tend to memorize these exact map-action pairs, effectively collapsing the learned conditional distribution into a Dirac delta function, $P_\theta(\mathcal{H}|\mathbf{z}_t^{(i)}) \approx \delta(\mathcal{H} - \mathcal{H}^{(i)})$.

To prevent the model from memorizing spurious correlations between specific voxel arrangements and deterministic position, we introduce the data augmentation in layout generator training. By applying paired spatial transformations $\mathcal{T}$ to both the condition and the target, i.e., $(\mathcal{T}(\mathbf{z}_t), \mathcal{T}(\mathcal{H}))$, we artificially expand the support of the empirical distribution. Specifically, we introduce a two-stage data augmentation pipeline comprising agent-level perturbations and global spatial transformations.

First, to manage scene density and prevent the model from overfitting to overly crowded environments, we cap the maximum number of vehicles per frame at 10; if a frame contains $N > 10$ vehicles, we uniformly sample a subset of exactly 10 vehicles without replacement, ensuring each vehicle has an equal retention probability of $10/N$. For each retained vehicle, we project it onto a 2D spatial heatmap $\mathcal{H} \in \mathbb{R}^{200\times200\times1}$ as a Gaussian distribution. Specifically, the Gaussian is centered at the vehicle's spatial coordinates with a kernel radius of 3. To simulate positional uncertainty and enrich the local behavioral manifold, we independently perturb the center of each Gaussian, randomly shifting it within a local 5×5 neighborhood restricted to the drivable surface area.

Following this localized perturbation, we apply rigid global transformations to the entire augmented scene (including both the occupancy map and the constructed heatmap $\mathcal{H}$). We uniformly sample and apply discrete rotations $\mathcal{R} \in \{90°, 180°, 270°\}$ and axis-aligned reflections (flipping along the x- or y-axis). Crucially, identical spatial transformations are applied to the corresponding latent representations to maintain spatial alignment with $\mathcal{H}$. While such reflections fundamentally invert the handedness of the traffic rules (effectively synthesizing valid left-hand traffic scenarios from right-hand data), this joint transformation ensures that all local topological constraints and relative vehicle dynamics remain physically coherent. This composite strategy effectively artificially expands the empirical distribution, forcing the network to learn robust, viewpoint-invariant geometric representations.

6.4 Additional metrics details

To evaluate the diversity of the generated stochastic rollouts, we employ two distinct metrics: Pairwise mIoU Diversity in the decoded semantic space, and the Vendi Score (Friedman & Dieng, 2022) in the latent feature space.

For the decoded semantic space, we generate N independent sequences over $\mathbb{T}$ frames using different random seeds. At any given timestep t, we compute the mean Intersection over Union (mIoU) across all semantic categories for every unique pair of generated occupancy grids. The Pairwise mIoU Diversity at time t is then defined as the complement of the average pairwise similarity:

$$D_t = 1 - \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \text{mIoU}_t^{(i,j)}$$

where $\mathcal{P} = \{(i, j) : 1 \le i < j \le N\}$ denotes the set of all unordered seed pairs. The rollout-level diversity is obtained by averaging D_t over time.

To capture structural diversity in the latent space, we report the Vendi Score, which measures the effective sample size of the generated distributions. First, we extract frame-level features $\{x_i\}_{i=1}^N$ using pretrained encoders (i.e., the OccFM VAE encoder and the UniScenes AE encoder). We then compute a cosine similarity kernel matrix $\mathcal{Q}$, where its elements are given by $\mathcal{Q}_{ij} = (1 + \tilde{x}_i^\top \tilde{x}_j)/2$, with $\tilde{x}$ being the ℓ_2-normalized feature. The Vendi Score is calculated as the exponential of the von Neumann entropy of the normalized kernel matrix $\hat{\mathcal{Q}} = \mathcal{Q}/\text{tr}(\mathcal{Q})$:

$$V = \exp\left(-\sum_\ell \lambda_\ell \log \lambda_\ell\right)$$

where $\{\lambda_\ell\}$ are the non-negative eigenvalues of $\hat{\mathcal{Q}}$.

Intuitively, the Vendi Score quantifies the effective number of distinct modes within a given set of generations. By evaluating the entropy of the similarity eigenspectrum, it properly penalizes highly correlated samples. A higher Vendi Score indicates that the generated occupancy features are widely distributed across the latent space, reflecting true structural and geometric diversity rather than minor variations of a single collapsed mode.

6.5 More simulation pipeline details

In Tab. 3, we quantitatively demonstrate an important downstream application of our proposed method: providing data for training of downstream occupancy world model. Here, in Fig. 5, we show the detailed process of how ego vehicle was designed to run in our simulation environment.

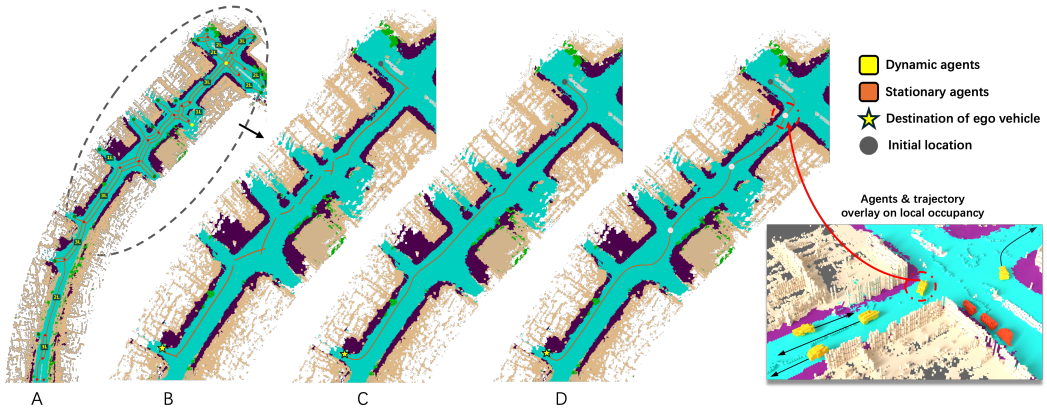


Figure 5: **The simulation pipeline after static occupancy fusion.** **A**: Part of the lane graph $\mathcal{G}$ extracted from the fused global semantic occupancy $\mathcal{M}_{\text{global}}$, shown overlaid on it. **B**: All agents, include ego, are instantiated by the same procedure (Layout generator in 3.3); the ego is distinguished only in that observations are rendered from its pose and itself is not rendered in collected data. Grey dots mark sampled initial poses; each agent independently draws a destination from all possible branch road boundary, and routes to it via A* over the extracted lane graph. **C**: The routed lane segments are connected and B-Spline smoothed into an initial reference route π_{traj}. **D**: The realized trajectory after closed-loop 2D-IDM execution. Pink dots mark lane switches triggered by agent–agent interaction (Sec. 9.4); because these depend on the joint state of all agents, the realized trajectories differ across rollouts even when the reference routes coincide. No route or trajectory at any stage is replayed from a recorded log.

Specifically, when simulating the motion behavior of agents (vehicles), we do not distinguish between ego and non-ego vehicles, we just collect the training data with the ego as the center. Therefore, in another words, all agents in the simulation are equivalent and we simply selected a car at random to record the data. Considering the irregularities in the generated occupancy, a closed-loop episode in our setting lasts around 1 min (Tab. 7), whereas Carla can run indefinitely in autopilot mode.

It is worth noting that the simulation data generated by OccSim exhibits a high degree of diversity and non-deterministic trajectories:

- W-DiT, which generates static scenes, is multi-modal, which allow us synthesize a large number of occupancy frames with complex road networks.
- The multi-modality of layout generator: as the ego vehicle moves forward, agents ahead are generated and rendered into the scene in real time, and each generation of a reasonable position is multimodal.
- Each agent's destination is a randomly selected point on the map's boundary, which means that traffic flow behavior can vary even within the same map.
- The initial velocity and acceleration of each agent also vary slightly, and when multiple agents move under IDM control, their interactions (avoidance maneuvers) with other agents are completely different in multi rollouts.

7 Additional Ablation Experiments

7.1 Inference Speed and Condition Comparison

In the previous section, we demonstrated a comparison between our method and prior approaches in terms of realism. However, it is worth noting that our method also achieves improvements in real-time performance compared to COME. In Tab. 5, we compare the time and FLOPS required for our method to generate a single frame. Our work generate 1 frame per inference, but still more efficient than COME, which generate 6 frames together per inference step.

Table 5: Comparison of efficiency and computational cost among different methods. When evaluating the time taken by our method, we distribute the post-processing time evenly across each frame of the sequence generation. Each result was computed independently using an RTX4090. DOME and COME generate 6 frames together, ours is 1 single frame per step, for fair comparison, we keep measure the GFLOPS / inference time needed per single frame.

Method	WM	Input	#Params	FPS	$\frac{\text{GFLOPS}}{\text{frame}}$
DOME	✓	4 frames sem. occ.&traj	444.07M	5.48	8891.98
COME	✓	4 frames sem. occ.&traj	204.53M	2.33	41333.17
InfiniCubes	×	1 frames sem. occ.	831.75M	0.02	990927.28
OccSim (Ours)	✓	1 frames sem. occ. & traj	181.24M	2.58	15570.18

Tab. 5 measures the per-frame cost of the world model in isolation. In Tab. 6 we instead break the full pipeline into its stages, which is what determines the cost of actually using OccSim, since the generation loop and the simulation loop are paid at very different rates. Building one environment from a 1000-step action sequence and an initial frame—encoding the first frame, 1000 W-DiT steps, decoding, and heuristic lane extraction—takes about 656s, and is paid once per environment rather than per simulated frame. Once the map is built, simulation inside the pre-computed area costs only IDM planning time (7.1ms per step); every 40m of ego motion additionally triggers a frontal-view layout recomputation, which encodes the front area, samples a layout and re-plans (92.68ms). No pre-compilation or reduced precision is used, so these numbers are an upper bound on the achievable latency.

Table 6: Breakdown of inference speed per frame (ms). All results are tested on $1\times$ RTX4090 with float32 precision. No further accelerations, such as pre-compilation, are applied here.

	Encoder	**W-DiT**	**Decoder**	**Layout Gen.**	**IDM Plan. & Rend.**
Latency	16.88	386.4	9.49	68.7	7.1

7.2 Closed-loop interaction safety

Because agents are initialized generatively rather than replayed, an initialization that is merely plausible in appearance can still place vehicles in configurations that are unrecoverable once the controller starts. We therefore measure the collision rate of the closed-loop simulation as a function of horizon and agent density, comparing our Layout Generator against uniform random placement on the drivable surface. As shown in Tab. 7, collisions under our initialization are nearly absent in the first 5s (at most 0.11% across all densities, and zero with up to 4 agents), and stay below 0.8% even at 60s with 7 agents. At matched agent count, random placement is $3.8\times$ worse at 3s and $2.1\times$ worse at 5s, and its 20s rate (0.91%) already exceeds our 60s rate (0.79%). The relative gap is largest at short horizons and narrows as the rollout proceeds ($1.7\times$ at 10s, $1.3\times$ at 60s), which localizes the benefit of the Layout Generator at initialization: random placement frequently yields geometrically inadmissible initial states, e.g., vehicles placed too close to a leading agent, and these conflicts surface within the first seconds. Collisions that accrue later stem from agent–agent interactions under the shared IDM controller and affect both schemes alike, consistent with Tab. 2, where the controller rather than the generated map accounts for most of the remaining discrepancy. For the downstream data collection of Tab. 3 we cap the scene at 6 agents and retain 10s collision-free clips.

7.3 3D fidelity evaluation under OccFM manifold

The 3D realism evaluation in the main paper (Fig. 4) is computed in the UniScene AE latent space. Since both FID-style metrics and the Vendi score are defined through a feature extractor, a single latent space leaves open whether the ranking reflects generation quality or a property of that particular autoencoder. We therefore repeat the same protocol—identical rollouts, identical baselines, identical metrics—in the OccFM VAE latent space (Liu et al., 2025), which differs in compression rate, training objective and semantic readout.

Table 7: Closed-loop Interaction Safety Evaluation (Collision Rate %). Each setting tested using 7500 samples in 150 maps. We use max 6 agents to collect 10s non-collision video clips for downstream training.

Init. method	Max agents	3s	5s	10s	20s	60s
Ours	4	0.00	0.00	0.12	0.45	0.49
	5	0.00	0.03	0.23	0.42	0.61
	6	0.02	0.11	0.38	0.44	0.77
	7	0.05	0.11	0.41	0.52	0.79
Random	7	0.19	0.23	0.71	0.91	1.00

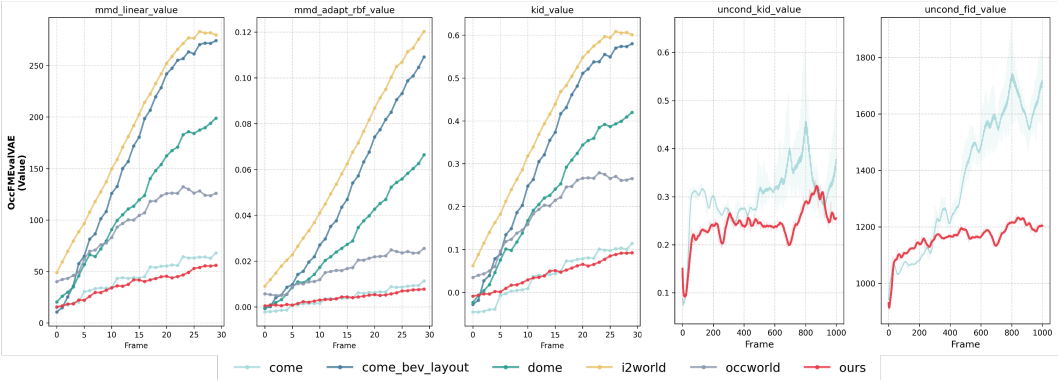


Figure 6: **3D realism evaluation in the OccFM VAE latent space.** The same protocol as the top row of Fig. 4, with features extracted by the OccFM VAE encoder instead of the UniScene AE. The left three panels report conditional fidelity (MMD with two kernels, KID) against the strictly paired GT frame over a 30-frame horizon; the right two report unconditional realism (KID, FID) against the marginal distribution of the training set over a 1000-frame rollout. Absolute values are not comparable to Fig. 4 as the feature distribution differs; the ranking of methods is.

Fig. 6 shows the ranking is preserved throughout. Under conditional fidelity, our method holds the lowest MMD and KID across the 30-frame horizon, and the baselines separate in the same order as in the UniScene space, with I2-World and COME-BEV-layout accumulating error fastest. Under unconditional realism, our rollout stays flat over 1000 frames while COME drifts upward monotonically. The absolute values are not comparable across the two latent spaces—FID in particular is offset by roughly an order of magnitude, since it is measured against a different feature distribution—but the relative ordering, the stability of our curves and the divergence of the baselines all transfer, indicating the conclusions of Sec. 4.2 are not an artifact of the chosen autoencoder.

7.4 Validating the lane extraction heuristic against HD maps

In design of OccSim, we use the heuristic detailed described in Sec. 9.3, which will later be used as trajectory segments for agents. Here we use the real HD-Map to measure the effectiveness of this approch.

As shown in Tab. 8, we first rasterize the HD-map drivable-area polygon and feed it to the same heuristic in Alg. 4, which establishes a lower bound on the error that our lane construction incurs regardless of the input quality. We attribute this residual mainly to two properties intrinsic to the construction: the absence of explicit turning connectors at intersections, and the fixed-width assumption of our orthogonal offsetting, which cannot accommodate roads of non-uniform lane width. Substituting GT semantic occupancy for the polygon raises the Chamfer error to 0.74 m, with precision and recall of 0.79 and 0.82. The additional degradation is within one voxel of the 0.4,m grid, and we attribute it to voxel quantization together with the absence of a separate parking-area label in the occupancy taxonomy, which causes parking lanes to be treated as drivable surface.

Table 8: **Geometric validation of lane topology extraction against HD maps.** Both rows run the identical extraction pipeline on the same spatial extent and are evaluated against the same nuScenes HD-map centerlines; they differ only in the input drivable mask. The first row uses a rasterized HD-map polygon and thus lower-bounds the error attributable to our lane construction alone; the excess of the second row is the net cost of operating on fused occupancy.

Drivable mask	Error source	Chamfer (m) ↓	P@1.0m ↑	R@1.0m ↑
HD-map polygon	Definitional (l.b.)	0.51	0.87	0.90
GT semantic occupancy	+ occupancy & fusion	0.74	0.79	0.82

7.5 More details in few steps sampling & single step rollout stride

For NFE step, as shown in Fig. 7a, NFE=10 get the best and stablest results in terms of fidelity. For the rollout stride, we use straight trajectory in Fig. 12 to test fidelity of rollouts under different single step stride, from 3m to 30m. As shown in figure 8, we find that 3 meters per step achieve the best results.

We do not attributes this to out-of-distribution(OOD) in update size: During training, $\mathbf{M}_{\text{rand}}$ is composed of one to three random rectangular blocks whose spatial extent spans roughly 8–56 m (overall 10%-50% of the full scene); a 30 m forward band is therefore not OOD in size, and the model is in principle capable of completing an unknown region at that scale in a single pass.

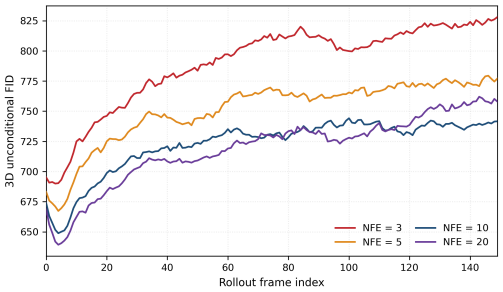


Figure 7: **Effect of sampling steps (NFE) on rollout quality.** Per-frame 3D unconditional FID against the training-set marginal, over 150-frame rollouts on the UniScenes AE latent (3 m per step, cfg $= 1.1$). Fidelity does not improve beyond NFE=10, which we adopt as the operating point.

We instead attribute the degradation to the autoregressive structure itself: since W-DiT re-synthesizes the complete frame at every step, a given spatial location is refined $\lceil \text{FOV}/\Delta d \rceil$ times over its visible lifetime, and this count falls as the stride Δd grows (FOV: Field of View, $80m \times 80m$ here). Prior outpainting-based methods like InfiniCube (Lu et al., 2025) obtain an **equivalent effect** through repeated intra-block resampling (Lugmayr et al., 2022), at the cost of a $64\times$ larger per-frame compute budget (Tab. 5); our stride offers a cheaper form of the same fidelity–throughput trade-off, and the ablation shows that the returns are already saturated at 3–5 m.

7.6 Spherical readout: a norm-invariant semantic head

In Tab. 4, we analyzed several factors that influence performance in W-DiT, including the loss function and architectural design. However, we found that the special designed representation readout method is also essential for stable, long-term rollouts. Specifically, when training the VAE for occupancy latent compression, we revise readout method from standard matmul to sphere decode, as shown in Alg. 1.

As shown in Fig. 9 and Tab. 9, the sphere readout has almost the same reconstruction performance as original matmul-based method, but significantly help the rollout stability. Furthermore, when we applied the sphere readout only during testing to VAE representations trained using the matmul readout, as shown by the blue line in Fig. 9, the results were actually worse. This experiment decoupled the decoder design itself from its effect on constraining the latent manifold during VAE training, demonstrating that the sphere readout implicitly reshapes the geometry of the latent manifold, and that this constraint ensures stability.

Treating each generated latent frame $z \in \mathbb{R}^{50\times50\times16}$ as 2,500 tokens of length 16, the latents from approximately 28,000 samples in the training set form a 16-dimensional space comprising approximately 7×10^7 tokens. In this space, we evaluate angular(which isolates directional drift)

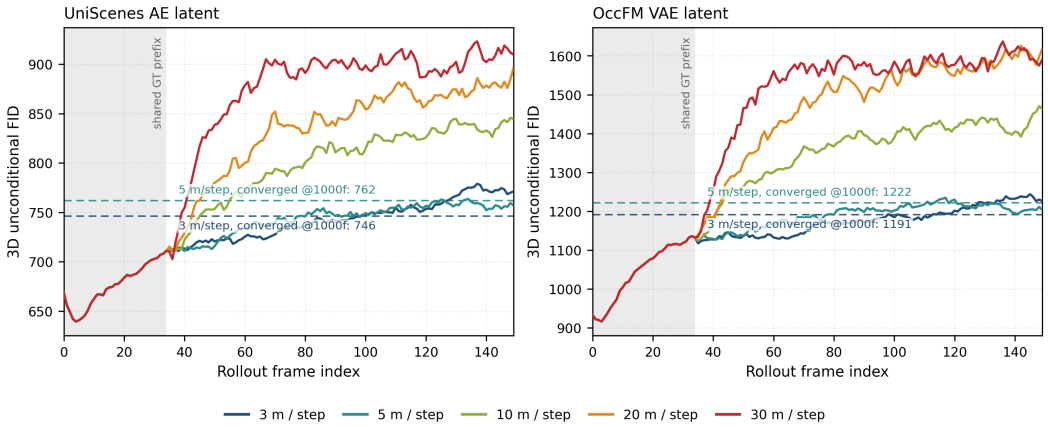


Figure 8: Fidelity of generated frames over time under different moving steps. We use the ground truth trajectories and poses at first 35 frames, concatenated with straight trajectory moving with different stride in single step. We use 3m / step for all of the experiments mentioned in the main text.

Table 9: Single-frame VAE encode–decode reconstruction on Occ3D-nuScenes (C18). Sphere decoding has almost the same IoU and mIoU.

VAE Decode Method	IoU	mIoU
Matmul	74.66	85.99
Sphere	**75.15**	**86.27**

Table 10: Geometric analyze of the semantic template $\mathbf{W}$ before (Matmul) and after (Sphere) ℓ_2 normalization. CV.: coefficient of variation. PC.: principal component. MACS: Mean Absolute Cosine Similarity.

Template statistic	Matmul	Sphere
Avg. length	2.801	1.000
Length CV. (↓)	0.4385	**0**
Top-3 PC. ratio	58.73%	49.89%
MACS. (↓)	0.2324	**0.2207**

and radial distance to reflect the OOD of the latents under VAEs with different readout schemes. For rollout frame $z_i^{(t)}$ at frame t, the angular distance is its k-th nearest-neighbor $\pi_k(i)$ distance in ground-truth latent z_{gt} on the unit sphere, defined below, here k=10. Radial distance is the angular distance without unit sphere constrain. Each is a median over the 2,500 tokens per frame, then across sequences (band: 25–75th pct.).

$$d_t^{\text{ang}} = \underset{i}{\text{median}} \left\| \hat{z}_i^{(t)} - \hat{z}_{\text{gt},\,\pi_k(i)} \right\|_2, \qquad \hat{z} = z/\|z\|_2.$$

The results are shown in Fig. 10. Under the matmul VAE, the raw kNN distance inflates by $\sim 2.2\times$over the rollout while the angular distance stays essentially flat ($\sim 0.99\times$); the sphere VAE remains stable in both. Since raw and angular kNN differ only in whether the token norm is retained, this shows the matmul latent drifts *purely radially*(norm inflation), not by directional change. Although the absolute distances of the two sub-figures are not directly comparable (they live in different latent spaces), the within-space trends consistently indicate that the sphere readout stabilizes the rollout by constraining the norm geometry of the latent space, mirroring the static norm imbalance in Tab. 10. Further analysis in Tab. 10 show that normalization removes inter-class norm imbalance (length CV $\to 0$) in template $\mathbf{W}$ while leaving inter-class *directions* almost untouched (MACS nearly unchanged), and flattens the singular-value spectrum toward isotropy.

Finally, we further verify that the sphere readout is not specific to our architecture: revising OccFM(Liu et al., 2025)'s VAE with the sphere readout from the matmul based one improves its downstream forecasting mIoU by several points, indicating the gain comes from the geometric regularity it imposes on the latent space rather than any model-specific design. As shown in Fig. 11,

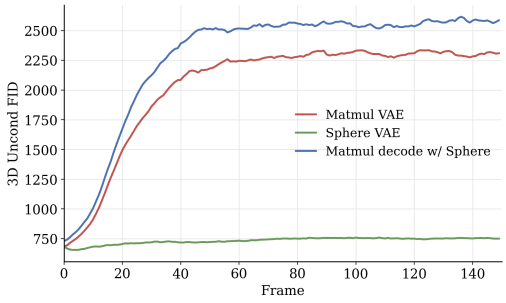


Figure 9: Long-horizon 3D unconditional FID in the UniScenesAE latent space. Standard matmul decoding degrades rapidly as latent energy decays over rollout, while norm-invariant sphere decoding stays stable across all 150 frames.

Algorithm 1 Norm-Invariant Sphere Decoding

Require: Decoded feature map $\mathbf{F} \in \mathbb{R}^{B\times H\times W\times Z\times C}$, semantic template $\mathbf{W} \in \mathbb{R}^{C\times K}$, scale s
Ensure: Class logits $\mathbf{L} \in \mathbb{R}^{B\times K\times H\times W\times Z}$

if method $=$ `sphere` **then**
  $\hat{\mathbf{F}} \leftarrow \mathbf{F}/\|\mathbf{F}\|_2$ ▷ normalize over C (dim $= -1$)
  $\hat{\mathbf{W}} \leftarrow \mathbf{W}/\|\mathbf{W}\|_2$ ▷ normalize over C (dim $= 1$)
  $\mathbf{S} \leftarrow \hat{\mathbf{F}}\,\hat{\mathbf{W}}$ ▷ cosine similarity
  $\mathbf{S} \leftarrow s \cdot \mathbf{S}$ ▷ fixed temperature s
else
  $\mathbf{S} \leftarrow \mathbf{F}\,\mathbf{W}$ ▷ standard matmul; logit $\propto \|\mathbf{F}\|$
$\mathbf{L} \leftarrow \text{permute}(\mathbf{S})$
return $\mathbf{L}$

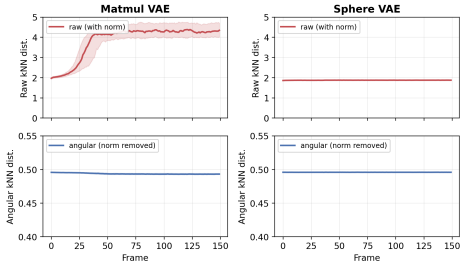


Figure 10: Per-token latent OOD within each VAE's own space: raw kNN (with norm) inflates only under matmul, while angular kNN (norm removed) stays flat for both.

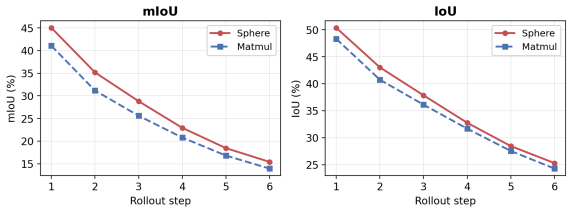


Figure 11: Downstream forecasting mIoU / IoU across rollout steps. Retraining the VAE with the sphere readout improves both metrics over the matmul baseline with almost no extra cost.

the sphere readout thus offers an almost free improvement (+2.74% absolutely, +11% relatively): no extra parameters cost, no reconstruction trade-off, yet a consistent downstream gain.

7.7 Complementing Real Logs with OccSim Data

Tab. 11 compares OccWorld trained on real UniOcc-Waymo logs (Uni-W), on OccSim data alone, and on their mixture, all evaluated zero-shot on UniOcc-nuScenes. Trained alone, OccSim data remains below real logs, which is expected: real logs are the realism upper bound of the data OccSim learns from. It nonetheless outperforms the asset-based CARLA data by a large margin. The value of OccSim lies in what logs cannot offer: a log is a fixed recording, whereas OccSim generates new road topologies and traffic layouts on demand and supports closed-loop interaction. Adding OccSim data amounting to 30% of Uni-W improves the average IoU over 1–3s by 6.8% relative (21.95 to 23.44) and the average mIoU by 4.1% (19.13 to 19.91), with gains at every horizon except 3s mIoU, which decreases slightly (15.80 to 15.08). OccSim data thus complements real logs without additional annotation. As in Tab. 3, all metrics here are computed open-loop against logged futures (see limitation 5 in Sec. 10).

8 Additional Visualization

We provide more visualizations for details of our work here. More information like dynamic traffic simulated can be found in the video in the supplementary materials.

Table 11: OccWorld trained on real UniOcc-Waymo logs (Uni-W), OccSim data (5× volume, as in Tab. 3), their mixture (OccSim data amounting to 30% of Uni-W), and CARLA (Uni-C), evaluated zero-shot on the UniOcc-nuScenes validation set.

Train Data	mIoU ↑			IoU ↑		
	1s	2s	3s	1s	2s	3s
Uni-W	23.47	18.11	15.80	27.35	20.86	17.63
Uni-W + OccSim	24.95	19.71	15.08	29.73	22.11	18.49
OccSim	19.54	12.31	9.08	23.87	16.92	13.47
CARLA	11.79	8.35	6.75	18.31	14.09	12.03

8.1 Details of trajectory shapes used in Fig. 4

To systematically evaluate the robustness of our model in long-horizon generation across diverse driving behaviors, we configure three distinct spatial trajectory profiles: Loop, Straight, and Turn. As illustrated in Fig. 12, the trajectory generation process conditioned on is consist of the initial ground-truth (GT) trajectory (highlighted in blue, encompassing frames 1–40) and extended trajectory with 3 modes (red lines).

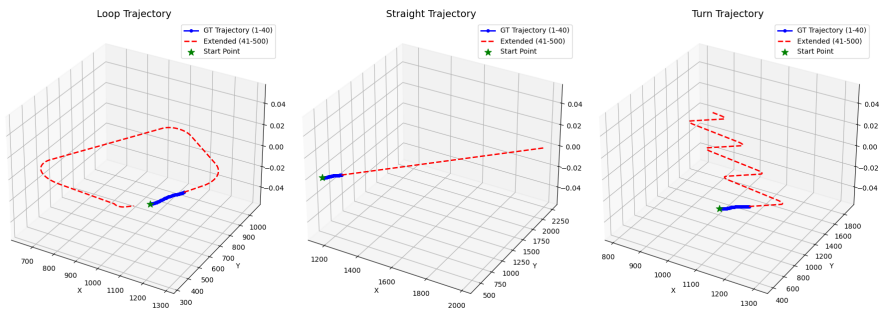


Figure 12: Visualization of trajectory shapes (Loop, Straight, and Turn) evaluated in Fig. 4. The green star indicates the starting point. The blue solid line denotes the initial ground-truth (GT) trajectory snippet (frames 1–40) used as the condition prompt, while the red dashed line represents the extended trajectory explicitly designed to guide the long-horizon generation. For visual clarity the displayed trajectories are truncated at 500 frames, whereas our actual synthesis spans up to 3000 frames.

8.2 Agent initialization within generated static maps

Fig. 13 shows initial agent states sampled by the Layout Generator inside maps produced by W-DiT, i.e. with no HD map or recorded log anywhere in the pipeline. Two properties are worth noting. First, the placements respect the road structure the generator is conditioned on: vehicles fall within drivable regions and align with the local lane direction after projection onto the nearest lane, including at intersections where several headings are admissible. Second, the conditional distribution is genuinely multimodal rather than collapsed onto a single mode. The third and fourth panels crop the same revisited location and differ only in the sampling seed, yet yield different plausible traffic configurations—the behaviour the augmentations of Sec. 6.3 are designed to preserve, since each logged scene is observed with only one realization of agent states. Stationary and moving vehicles are also separated: the parked vehicles in the second panel are placed along the roadside rather than in the travel lane, which follows from encoding the two motion states as Gaussian kernels of opposite sign in $\mathcal{H}$.

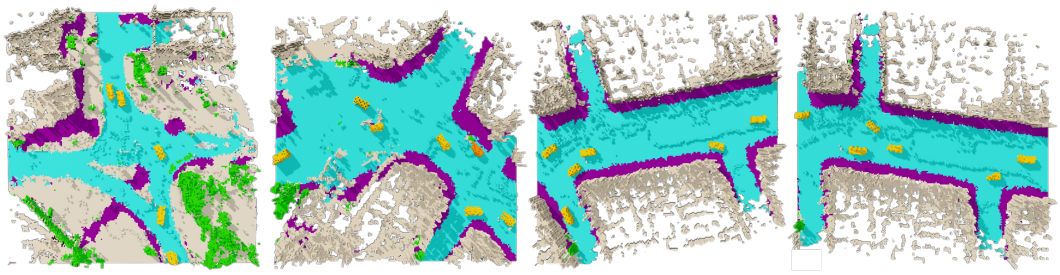


Figure 13: **Agent initialization within generated static maps.** Each panel displays a 200x200 voxel crop centered on the (unrendered) ego-vehicle from $\mathcal{M}_{\text{global}}$. The first two panels show distinct generated scenes, the third and fourth panels demonstrate the model's multimodality: two different plausible layouts generated for the same revisited location. Orange vehicles (second panel) are stationary.

8.3 Comparison with other Occ. World Models in long-horizon rollouts

From Fig. 14, we can clearly observe that our proposed W-DiT significantly extends the duration of stable rollout. Even after 3000 frames (1500s, approximately 25 minutes with 2Hz setting in Occ3d-NuSc), the generated occupancy remains stable. The top half of Fig. 14 shows the output when the loop trajectory from Fig. 12 is used as input, while the bottom half shows the output when the straight-line trajectory from Fig. 12 is used as input.

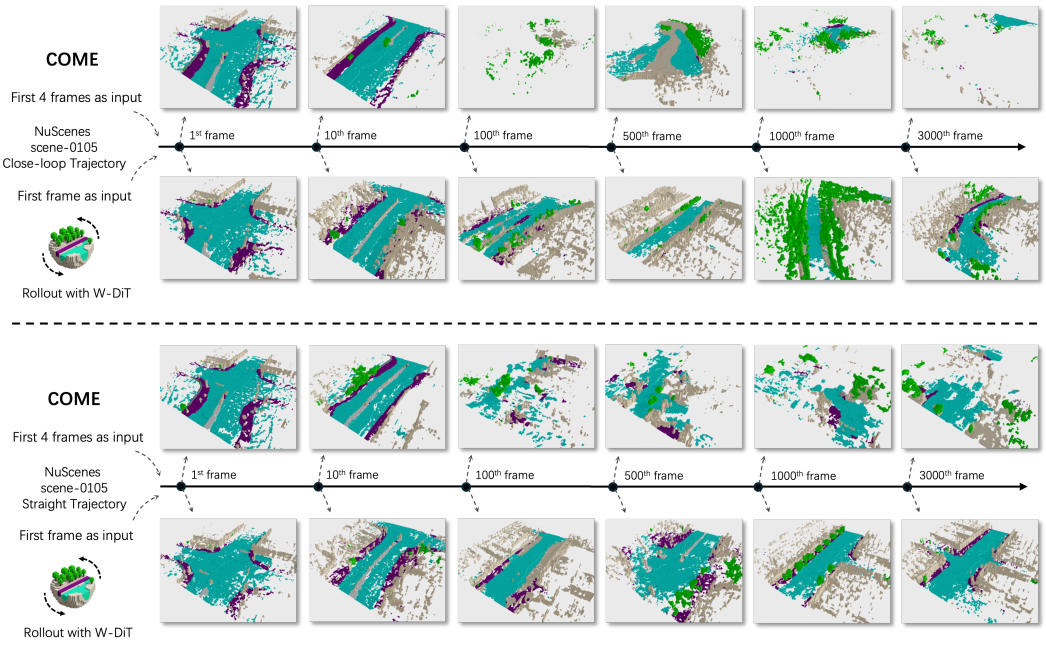


Figure 14: Qualitative comparison of two rollouts for two models: previous SOTA occupancy world model (first and third rows), and our model (second and fourth rows) at different horizons. Ours exhibits long-horizon stability under different trajectory input.

8.4 Comparison with out-painting based methods

World models are indeed one of the technologies used to generate boundary-free driving scenes; however, outpainting-based methods can achieve the same in the context of static occupancy. Therefore, in this section, we aim to qualitatively demonstrate that our proposed method outperforms previous outpainting-based approaches. In Fig. 15, we compare our model against SemCity (Lee et al., 2024)

and InfiniCube (Lu et al., 2025), both of which generate future scenes via outpainting. Notably, InfiniCube additionally requires an HD-map aligned with the region to be generated.

As shown in the Fig. 15, the outpainting-based InfiniCube not only requires extra HD-map annotations but also suffers from severely limited diversity, exhibiting only minor variations in the foreground across different segments. In contrast, among the HD-map-free approaches, SemCity's results are markedly inferior to ours in realism, diversity, and continuity.

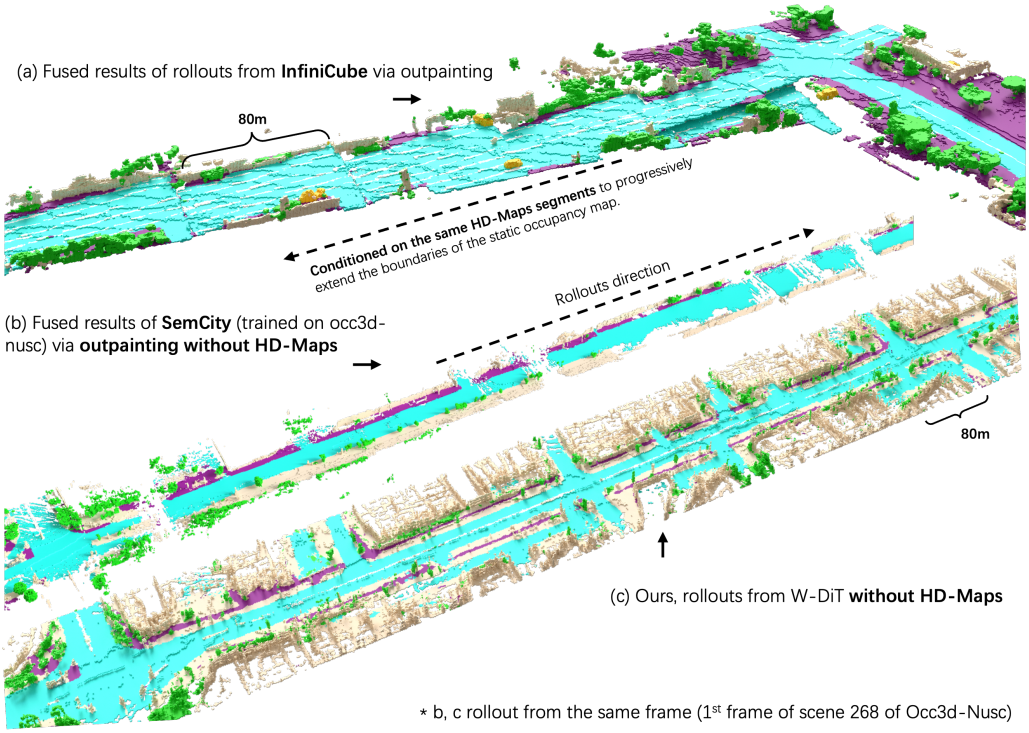


Figure 15: Visualization of static occupancy generated from InfiniCube, SemCity and our approach (from top to down). For InfiniCube, we reuse the boundary HD-map to continuously extend the map along a straight line. Likewise, for SemCity and our method, we extend the scene in the same direction.

8.5 Ultra long rollout from W-DiT: 100+km city scale generation with real world trajectory.

In Fig. 4, we evaluate the fidelity of model rollout, our proposed method remained stable over a time scale of 1,000 frames. Furthermore, we used a 3,000-frame rollout to construct a static map and collected data based on this map to conduct the experiments presented in Tab. 3. Here we push the length of generation to 100x longer, build a city-scale occupancy in Fig. 16, to further show the stability of W-DiT.

By manually concatenating all real-world trajectories from the occ3d-waymo training, we obtained driving trajectories from real drivers totaling over 100 km in length (shown at middle of Fig. 16). With this as the user-steered input of our model trained purely on occ3d-nuscenes, we further expanded the rollout to 110K frames, resulting in city-level occupancy data spanning over 100 km.

Specifically, after fully generating these approximately 110K frames, we fused them using Alg. 2 and presented the fusion results for a portion of the subsequences. Furthermore, we randomly visualized 10 frames from this sequence. As can be seen, driven by a large number of real-world trajectories, our method is able to generate highly diverse and realistic road structures, which provides a foundation for subsequent data collection and training in downstream tasks. As far as we know, this is the largest scale of driving scenario generation.

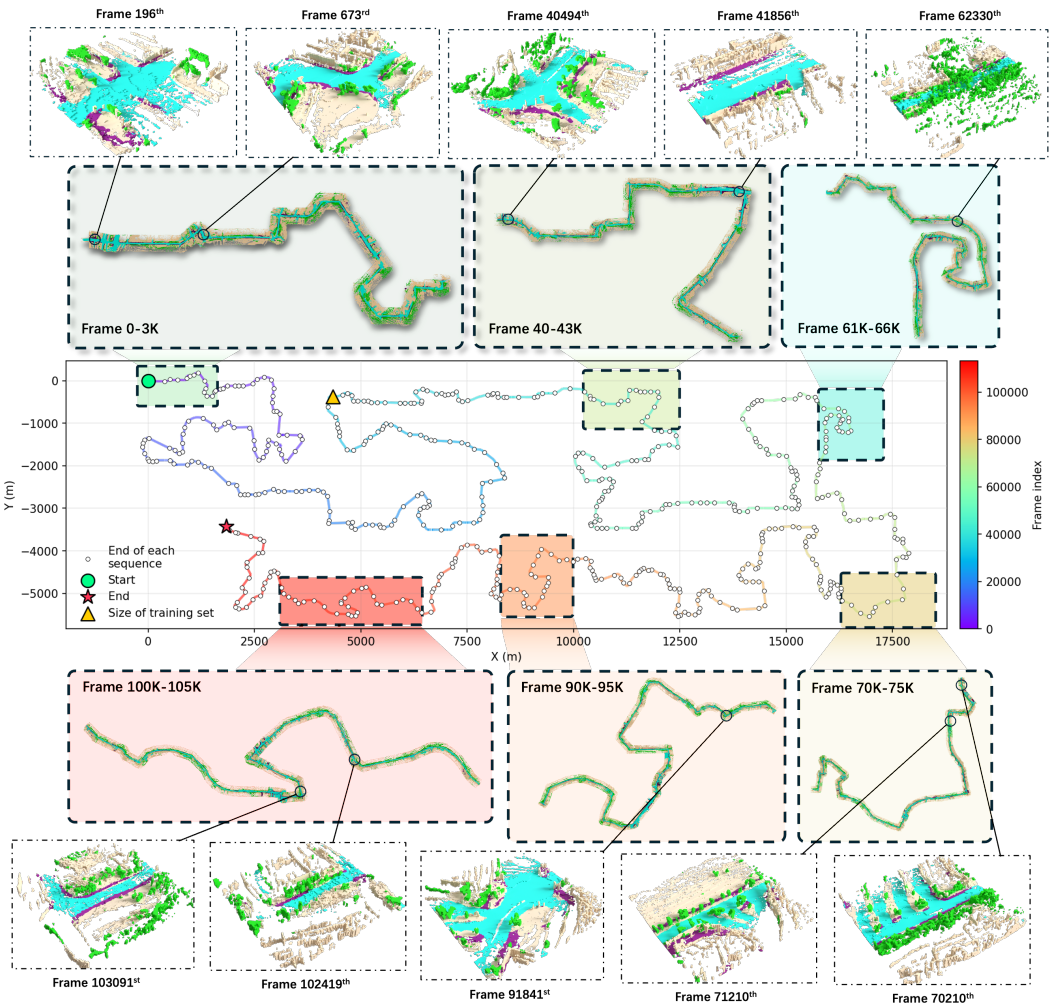


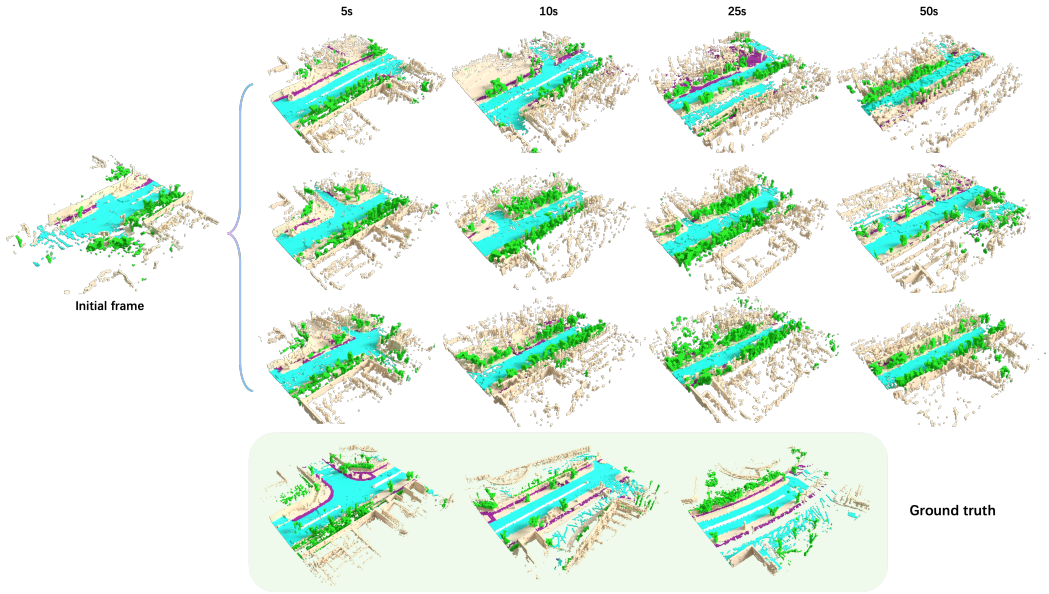
Figure 16: The full 100+ km trajectory (113321 frames) concatenated from Occ3D-Waymo. The connecting angle between every two pose sequences is set manually, and the full trajectory is smoothed with a B-spline after concatenation. This experiment uses the raw step size from real Waymo trajectories (approximately 0.9 m per frame).

8.6 Multi-modal diversity of W-DiT: fix-seed multi-time rollout results

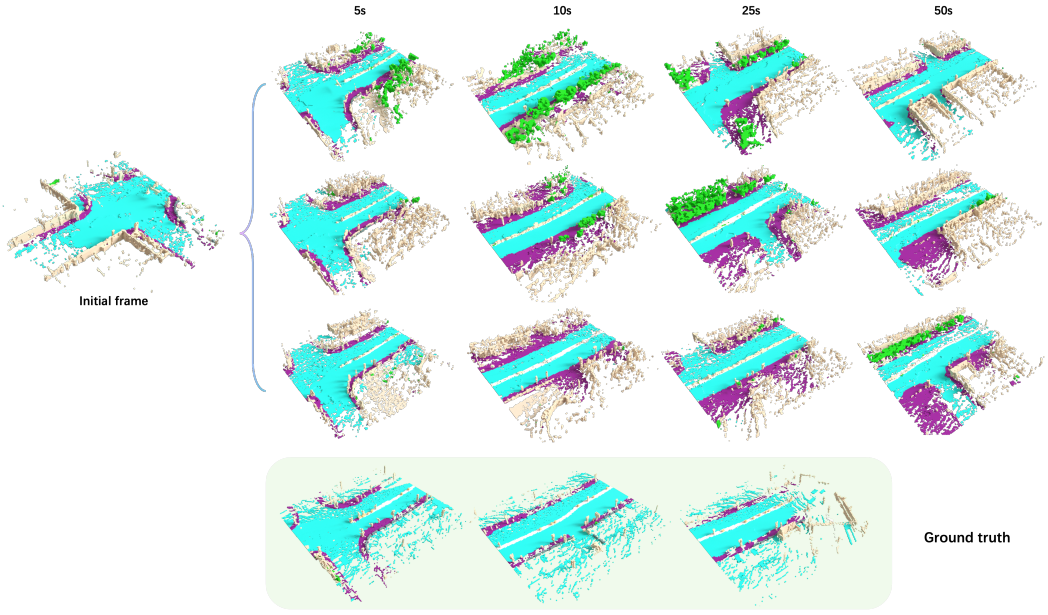
In the main text, we quantitatively demonstrate the SoTA diversity of our W-DiT-based model rollouts using the Vendi Score. Here, in Figure 17, we qualitatively illustrate our model's ability to capture the multi-modality of the future when only the input noise (seed) differs. After 50s (100 frames, we use 6m/s speed to generate trajectory) rollout, the branch road topology, vegetation / man-made presents obvious difference.

8.7 Visualization of chaos baseline constructed

As shown in Fig. 18, the constructed "chaos" baseline primarily comprises severely fragmented road segments, topologically disconnected layouts, and large-scale empty regions. Consequently, as we mentioned in Fig. 4, any model yielding FID/KID scores above this baseline threshold can be deemed completely incapable of synthesizing large-scale, valid road networks.



(a) Scene 0039



(b) Scene 0105

Figure 17: Visualization of W-DiT's long-horizon rollouts under different random seeds: given the same input frame (the first frame of scene-039 and scene-105 from Occ3D-nuScenes) and an identical straight-line trajectory. The three rows correspond to seeds 999/2026/3407, respectively. We also show the comparison with GT sequences at 5s, 10s and 20s here, to further illustrate that our method generate diverse future map, instead of just memorize the ground truth.

9 Algorithm Details

9.1 Heuristic for Single Frame Fusion.

After we obtained the raw forecast observation $\{\mathcal{O}_i\}_i^K$ from the W-DiT based world model, we than use the following algorithm 2 to fuse all frames to $\mathcal{M}_{\text{global}}$.

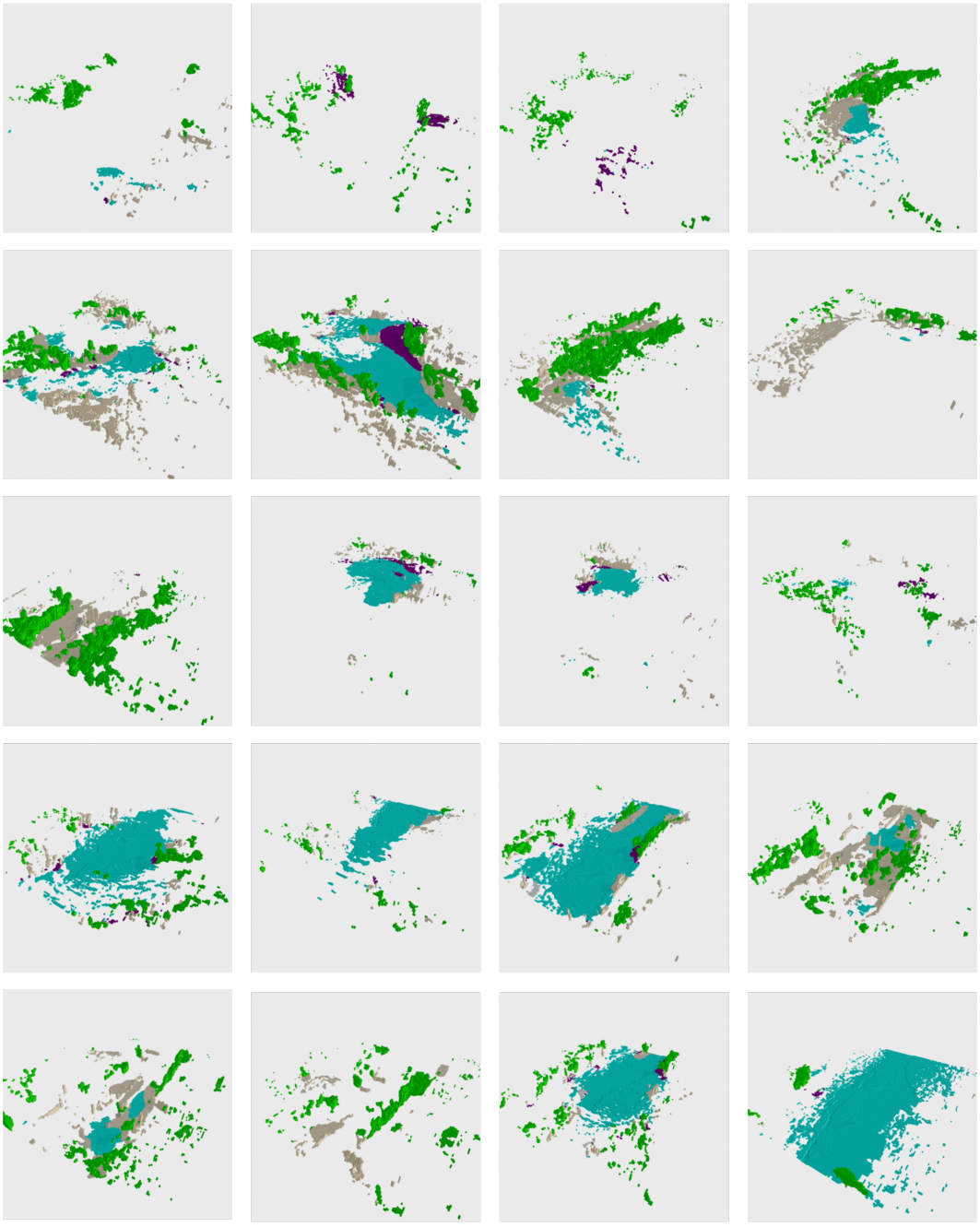


Figure 18: Visualization of frames from the "chaos" baseline. We randomly visualize 20 frames from the 1000 samples pool. The semantic categories represented by colors here are consistent with those in Fig. 13.

The pipeline consists of four primary phases. First, to manage computational overhead and redundancy, we subsample the generated sequences into keyframes based on a spatial distance threshold, projecting them onto a unified SE(2) bird's-eye-view (BEV) plane. Second, during the initial fusion pass, we introduce a deterministic column sinking operation. This forces all predicted ground-level voxels to align with the absolute zero-elevation plane ($z = 0$), effectively rectifying pitch and roll artifacts or elevation drifts accumulated over long-horizon rollouts. Third, to address occlusions and local generation artifacts, we leverage the temporal redundancy of non-keyframes; by applying a spatial majority voting mechanism, we robustly inpaint the remaining gaps on the $z = 0$ plane.

Algorithm 2 Two-pass Keyframe-based Occupancy Fusion

Require: Local predictions $\{\hat{\mathcal{O}}_t\}_{t=1}^{T}$, Ego-poses $\{\mathbf{T}_t\}_{t=1}^{T}$, max step $d_{\max}$, vote threshold τ, number of semantic categories C.
Ensure: Fused global occupancy map $\mathcal{M}_{\text{global}} \in \{0, 1, \dots, C\}^{X \times Y \times Z}$.

// Phase 1: SE(2) Projection & Keyframe Selection
$\mathbf{T}_t(x, y) \leftarrow \text{Proj}_{\text{SE(2)}}(\mathbf{T}_t), \quad \forall t \in \{1 \dots T\}$ ▷ Align to BEV plane
$\mathcal{K} \leftarrow \{1\}, \quad \mathbf{p}_{\text{last}} \leftarrow \mathbf{T}_1(x, y)$
for $t = 2$ **to** T **do**
  if $\|\mathbf{T}_t(x, y) - \mathbf{p}_{\text{last}}\|_2 > d_{\max}$ **then**
    $\mathcal{K} \leftarrow \mathcal{K} \cup \{t\}, \quad \mathbf{p}_{\text{last}} \leftarrow \mathbf{T}_t(x, y)$
// Phase 2: Pass 1 - Keyframe Sinking & Fusion
Initialize $\mathcal{M}_{\text{global}}(\mathbf{v}) \leftarrow \varnothing, \quad \forall \mathbf{v} \in \mathbb{Z}^{X \times Y \times Z}$ ▷ $\varnothing$ denotes unassigned.
for $k \in \mathcal{K}$ **do**
  $\mathcal{O}'_k \leftarrow \mathcal{W}(\hat{\mathcal{O}}_k, \mathbf{T}_k)$ ▷ Spatial warping to global coordination
  ▷ Assign key-frame voxels from local map to global map.
  $\mathcal{M}_{\text{global}}(\mathbf{v}) \leftarrow \mathcal{O}'_k(\mathbf{v}), \quad \forall \mathbf{v} \text{ s.t. } \mathcal{M}_{\text{global}}(\mathbf{v}) = \varnothing$
$\Delta z(x, y) \leftarrow \min\{z \mid \mathcal{M}_{\text{global}}(x, y, z) \in \mathcal{C}_{\text{ground}}\}$
$\mathcal{M}_{\text{global}}(x, y, z) \leftarrow \mathcal{M}_{\text{global}}(x, y, z + \Delta z)$ ▷ Column sinking to lowest ground
$\mathcal{M}_{\text{global}}(\mathbf{v}) \leftarrow \text{Mode}(\mathcal{N}_{3\times3}(\mathbf{v}))$ for $\mathbf{v}$ where $\mathcal{M}_{\text{global}}(\mathbf{v}) = \varnothing, z = 0$
// Phase 3: Pass 2 - Non-Keyframe Voting Inpainting
Initialize vote tensor $\mathcal{V} \in \mathbb{N}^{X \times Y \times Z \times C} \leftarrow \mathbf{0}$
for $t \notin \mathcal{K}$ **do**
  $\mathcal{O}'_t \leftarrow \mathcal{W}(\hat{\mathcal{O}}_t, \mathbf{T}_t)$
  $\mathcal{V}(\mathbf{v}, c) \leftarrow \mathcal{V}(\mathbf{v}, c) + \mathbb{I}[\mathcal{O}'_t(\mathbf{v}) = c], \quad \forall \mathbf{v} \text{ s.t. } \mathcal{M}_{\text{global}}(\mathbf{v}) = \varnothing$
$\mathcal{M}_{\text{global}}(\mathbf{v}) \leftarrow \arg\max_c \mathcal{V}(\mathbf{v}, c)$ if $\max_c \mathcal{V}(\mathbf{v}, c) \geq \tau$
// Phase 4: Morphological Refinement
$\mathcal{M}_{\text{global}} \leftarrow \text{CloseOp}(\mathcal{M}_{\text{global}}, c_{\text{sidewalk}})$ ▷ Binary closing on sidewalk
$\mathcal{M}_{\text{global}} \leftarrow \text{FilterAreaOp}(\mathcal{M}_{\text{global}}, c_{\text{road}}, A < 2\text{m}^2)$ ▷ Remove isolated noise
return $\mathcal{M}_{\text{global}}$

Finally, a morphological refinement stage—comprising binary closing and connected-component area filtering—is applied to enforce spatial continuity on sidewalks and systematically remove isolated noise clusters (e.g., artifacts smaller than 2m^2). This pipeline ensures a connected and smooth road topology, which is strictly required for downstream 2D multi-agent simulations.

9.2 Branch road detection and topology extraction.

Then, to bridge the gap between discrete occupancy grids $\mathcal{M}_{\text{global}}$ and vectorized road networks required for downstream multi-agent simulation, we propose a robust topology extraction and graph refinement algorithm. We first extract the 2D drivable surface mask, apply connected-component filtering, and perform strict single-pixel skeletonization to initialize the graph structure. To mitigate rasterization artifacts, we systematically eliminate spurious triangular loops by removing the longest edges within 3-cliques. Subsequently, a rigorous graph cleaning protocol is executed: we iteratively prune short, noisy spurs and introduce a node contraction mechanism to merge spatially dense junction clusters into unified super-nodes, accurately reflecting complex real-world intersections. Finally, to precisely isolate valid map ingress and egress points, we design a novel Dual-Probe filtering mechanism. A topology probe identifies and discards artificial endpoints caused by internal fragmentation, while a semantic probe leverages the 3D elevation profile of the occupancy map to detect physical occlusions or obstacles along the projected path. Ultimately, this pipeline translates the generated voxel map into a high-fidelity, deadlock-free topological graph, providing a reliable foundation for downstream closed-loop planning and control.

9.3 Lane topology extraction.

After we have $\mathcal{G}$, $\mathcal{V}_{\text{valid}}$, we use the retained graph information to create lane graphs. As shown in Alg. 4, based on the extracted skeleton, i.e., $E \in \mathcal{G}$, we introduce a two-phase lane topology

Algorithm 3 Branch point detection / topology extraction

Require: Fused global map $\mathcal{M}_{\text{global}} \in \{0, \dots, C\}^{X \times Y \times Z}$, lane width w_{lane}, pruning threshold τ_{prune}.
$\mathcal{V}_{\text{valid}} \leftarrow \varnothing$ for valid branch node set in the graph.
Ensure: Vectorized road graph $\mathcal{G} = (V, E)$, valid endpoints $\mathbf{P}_{\text{valid}}$.

// Phase 1: Binary Masking & Skeletonization
$\mathcal{B}(x, y) \leftarrow \mathbb{I}[\mathcal{M}_{\text{global}}(x, y, 0) = c_{\text{road}}], \quad \forall (x, y) \in \mathbb{Z}^2$
$\mathcal{S} \leftarrow \text{Skeletonize}(\mathcal{B})$
$\mathcal{S}(x+1, y+1) \leftarrow 0, \quad \forall (x, y) \text{ s.t. } \sum_{i=0}^{1} \sum_{j=0}^{1} \mathcal{S}(x+i, y+j) = 4$
// Phase 2: Graph Initialization & Artifact Removal
$V \leftarrow \{\mathbf{v} \in \mathbb{Z}^2 \mid \mathcal{S}(\mathbf{v}) = 1\}$
$E \leftarrow \{(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in V, \|\mathbf{u} - \mathbf{v}\|_\infty = 1\}$, with weight $w(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2$
Initialize graph $\mathcal{G} \leftarrow (V, E)$
for each 3-clique (triangle) $K_3 \in \mathcal{G}$ **do**
    $e_{\max} \leftarrow \arg\max_{e \in K_3} w(e)$
    $E \leftarrow E \setminus \{e_{\max}\}$ ▷ Break loops to expose leaf nodes
// Phase 3: Iterative Spur Pruning & Junction Contraction
repeat
    $\mathcal{V}_{\text{leaf}} \leftarrow \{v \in V \mid \deg(v) = 1\}$
    Find branch path $P_{l \to j}$ from $l \in \mathcal{V}_{\text{leaf}}$ to nearest junction j ($\deg(j) > 2$)
    if $\sum_{e \in P} w(e) < \tau_{\text{prune}}$ **then**
        $V \leftarrow V \setminus (P_{l \to j} \setminus \{j\})$ ▷ Remove short spurs
until no spurs are removed
$\mathcal{V}_{\text{junc}} \leftarrow \{v \in V \mid \deg(v) > 2\}$
for $\mathbf{u}, \mathbf{v} \in \mathcal{V}_{\text{junc}}$ s.t. $\|\mathbf{u} - \mathbf{v}\|_2 < 2 \cdot w_{\text{lane}}$ **do**
    $\mathcal{G} \leftarrow \text{ContractNodes}(\mathcal{G}, \mathbf{u}, \mathbf{v})$ ▷ Merge clustered junctions
// Phase 4: Dual-Probe Endpoint Filtering
$\mathcal{O}_{\text{obstacle}} \leftarrow (\mathcal{M}_{\text{global}}(x, y, z \geq 1) \notin \{c_{\text{road}}, c_{\text{sidewalk}}, \varnothing\})$ ▷ Get 3D obstacle occ above ground layer
$\mathcal{V}_{\text{end}} \leftarrow \{v \in V \mid \deg(v) = 1\}$
for $v \in \mathcal{V}_{\text{end}}$ **do**
    Let $\vec{d}$ be the outward normalized direction vector from parent junction to v.
    Topology Probe: $p_{\text{topo}} \leftarrow v + (1.5 \cdot w_{\text{lane}})\vec{d}$
    Semantic Probe: Generate 15m $\times$ 3.6m bounding box B_{sem} along $\vec{d}$ from v.
    if $\mathcal{B}(p_{\text{topo}}) == 0$ **and** $\sum_{\mathbf{p} \in B_{\text{sem}}} \mathbb{I}[\mathcal{O}_{\text{3D}}(\mathbf{p})] < \tau_{\text{obs}}$ **then**
        $\mathcal{V}_{\text{valid}} \leftarrow \mathcal{V}_{\text{valid}} \cup \{v\}$ ▷ Endpoint is neither internal nor blocked
return $\mathcal{G}$, $\mathcal{V}_{\text{valid}}$

extraction algorithm (Algorithm 3). In the first phase, we convert the raw skeletal segments into kinematically smooth reference centerlines using 3rd-degree B-Spline interpolation with a fixed spatial resolution Δs_{step}. To reconstruct multi-lane structures, we generate explicit parallel lanes via orthogonal offsetting along the normal vectors of the reference trajectories. To guarantee physical feasibility, we strictly enforce a topological boundary constraint: the interior of any newly generated lane must be fully bounded by the drivable surface mask $\mathcal{B}$.

In the second phase, we address the geometric overlaps that inevitably occur at complex intersections where multiple segments converge. We formulate this as a spatial intersection resolution problem. For each candidate lane, we identify conflict points that fall within a minimal distance threshold ϵ of any other independent lane. We then sever the trajectories at these overlapping junctions and exclusively retain the longest continuous sub-trajectory. Finally, a secondary B-Spline interpolation is applied to eliminate any geometric artifacts or non-differentiable kinks introduced by the splitting process. This rigorous pipeline yields a highly precise, non-overlapping lane network, providing a kinematically reliable foundation for multi-agent closed-loop simulation.

Algorithm 4 Lane Topology Extraction

Require: Extracted segments $\mathcal{S}$ from $\mathcal{G}$, drivable mask $\mathcal{B}$, lane width w_{lane}, distance threshold ϵ, resampling step Δs_{step}.
Ensure: Vectorized explicit lanes $\mathcal{L}_{\text{final}}$.

// Phase 1: B-Spline Smoothing & Parallel Offsetting
$\mathcal{L}_{\text{cand}} \leftarrow \varnothing$
for each $S \in \mathcal{S}$ s.t. $|S| \geq 10$ **do**
  $\mathbf{P}_c \leftarrow \text{BSpline}(S_{\text{path}}, \deg = 3, \Delta s_{\text{step}})$
  $\mathbf{N} \leftarrow \text{NormalVectors}(\mathbf{P}_c)$, $n \leftarrow \lfloor S_{\text{width}}/w_{\text{lane}} \rfloor - 1$
  $\mathbf{O} \leftarrow \left\{ \left(i - \frac{n-1}{2}\right) w_{\text{lane}} \;\middle|\; i \in \{0, \ldots, n-1\} \right\}$
  for $o \in \mathbf{O}$ **do**
    $\mathbf{P}_l \leftarrow \mathbf{P}_c + o \cdot \mathbf{N}$ ▷ Generate parallel lane curves
    if $\mathbf{P}_l^\circ \subset \mathcal{B}$ **then** ▷ $\mathbf{P}_l^\circ$ denotes the interior of the trajectory
      $\mathcal{L}_{\text{cand}} \leftarrow \mathcal{L}_{\text{cand}} \cup \{\mathbf{P}_l\}$
// Phase 2: Spatial Intersection Resolution & Resmoothing
for each $L \in \mathcal{L}_{\text{cand}}$ **do**
  $\mathcal{C}_L \leftarrow \{\mathbf{p} \in L \mid \exists L' \in \mathcal{L}_{\text{cand}} \setminus \{L\}, \text{dist}(\mathbf{p}, L') < \epsilon\}$ ▷ Identify cross-lane overlaps
  $L \leftarrow \arg\max_{L_{\text{sub}} \subseteq (L \setminus \mathcal{C}_L)} |L_{\text{sub}}|$ ▷ Retain the longest continuous sub-trajectory
  $L \leftarrow \text{BSpline}(L, \deg = 3, \Delta s_{\text{step}})$ ▷ Smooth artifacts caused by splitting
$\mathcal{L}_{\text{final}} \leftarrow \{L \in \mathcal{L}_{\text{cand}} \mid |L| \geq 5\}$ ▷ Remove all lanes short than 5 pixels
return $\mathcal{L}_{\text{final}}$

Algorithm 5 Spatially-conditioned agent generation and routing initialization

Require: Query anchor pose $\mathbf{T} \in SE(2)$ with spatial coordinate $\mathbf{p}_\mathbf{T}$, boolean flag b_{ego} indicating whether to explicitly initialize the ego vehicle at $\mathbf{T}$, mean/standard of initial velocity μ_v, σ_v, $\mathcal{M}_{\text{global}}$ as global map, $\mathcal{V}_{\text{valid}}$ from Alg. 3, Agents asset pool $\mathcal{X}$.

Crop local occupancy $\mathcal{O}_\mathbf{T}$ from $\mathcal{M}_{\text{global}}$ by pose $\mathbf{T}$.
$\mathcal{A}_{\text{new}} \leftarrow \text{LayoutGenerator}(\mathcal{O}_\mathbf{T}) \cup \left(\{\mathbf{p}_\mathbf{T}\} \text{ if } b_{\text{ego}} \text{ else } \emptyset\right)$
for each agent $a \in \mathcal{A}_{\text{new}}$ parameterized by pose $\mathbf{p}_a$ **do**
  Initialize velocity $v_a \leftarrow v_{\text{des}} \sim \mathcal{N}(\mu_v, \sigma_v)$
  $\mathbf{p}_{\text{target}} \sim \text{Uniform}(\mathcal{V}_{\text{valid}})$ ▷ Random valid endpoint
  $\S_a \sim \text{Uniform}(\mathcal{X})$ ▷ Random agent asset
  $\pi_{\text{traj}} \leftarrow \text{A*}(\mathcal{L}_{\text{final}}, \mathbf{p}_a, \mathbf{p}_{\text{target}})$ ▷ Shortest valid path
  $\mathbf{h}_a \leftarrow \text{Tangent}(\pi_{\text{traj}}, \mathbf{p}_a)$ ▷ Generate initial headings from trajectory
  $a \leftarrow (\mathbf{p}_a, \mathbf{h}_a, v_a, \pi_{\text{traj}}, \S_a)$ ▷ Assign initial states to agent
return $\mathcal{A}_{\text{new}}$

9.4 2D-IDM BASED AGENT SIMULATION ENGINE.

With all geometric information obtained, we introduce a 2D-IDM based simulation in Alg. 6. The algorithm is structurally divided into two phases: global pre-computation (Phase 1) and dynamic step-by-step execution (Phase 2).

Phase 1 employs a proactive spatial pre-computation strategy. Rather than solely initializing traffic around the starting ego-pose ($\mathbf{T}_0^{\text{ego}}$), the engine seamlessly injects background agents at forward and backward anchor poses ($\mathbf{T}_{\text{fwd}}$, $\mathbf{T}_{\text{bwd}}$) extending up to a predefined horizon d_{pre}. The unified SpawnAgents procedure handles this layout generation, simultaneously sampling valid endpoints, predicting spatial coordinates, and assigning initial shortest paths via A* routing.

In the simulation part, algorithm 6 Line 20 defines the spatial identification of candidate leading vehicles ($\mathcal{C}_a$). Instead of relying on heuristic bounding box checks, an agent a filters surrounding vehicles using a normalized vector dot product, $\left(\frac{(\mathbf{p}_{a'} - \mathbf{p}_a) \cdot \mathbf{h}_a}{\|\mathbf{p}_{a'} - \mathbf{p}_a\|_2} > 0.5\right)$, to ensure the target is strictly within a 60° forward field-of-view cone. This is coupled with a trajectory distance constraint ($\text{Dist}(\mathbf{p}_{a'}, \pi_{\text{traj}}) < d_{\text{lat}}$) to confirm the target is actively blocking the planned route.

Furthermore, algorithm 6 Line 23 explicitly shows the lane-changing logic. An agent initiates a Bezier-smoothed lane transition and A* re-routing only when the longitudinal distance falls below a critical threshold ($s < d_{\text{lc}}$) under two specific risks: either the ego agent is approaching a slower leader ($\Delta v > 0$), or it encounters a wrong-way/head-on vehicle. The latter scenario is elegantly captured by the negative dot product of their respective heading unit vectors ($\mathbf{h}_a \cdot \mathbf{h}_{a_{\text{lead}}} < -0.5$).

Finally, the kinematic states are updated via IDM, and the dynamic 3D assets are sequentially rendered onto the cropped static background using a spatial overwrite operator ($\circledast$) to handle semantic Z-buffering (Step 2.3), ultimately producing a seamless sequence of dynamic local environments.

Algorithm 6 A* routing + 2D-IDM based control engine pseudo codes

Require: A pool of global maps $\mathbb{M}$ generated, IDM parameters, rolling horizon threshold d_{roll}, simulation horizon ϕ, Lane topology $\mathcal{L}_{\text{final}}$, Δt for Inter-frame time interval, FOV($\mathbf{T}$): The set of all spatial coordinates within the perception range of the ego vehicle at pose $\mathbf{T}$, Agents asset pool $\mathcal{X}$, d_{lc} for distance threshold of lane change. SpawnAgents: method generate initial state of agents, details in Alg. 5.

Ensure: Rendered sequence of dynamic local environments $\{\mathcal{M}_{\text{local}}^{(t)}\}_{t=1}^{\phi}$.

// Phase 1: Global Initialization & Pre-computation
Randomly select a global map $\mathcal{M}_{\text{global}} \sim \text{Uniform}(\mathbb{M})$.
Get ego-poses $\{\mathbf{T}_t\}_{t=1}^{T}$ used to build $\mathcal{M}_{\text{global}}$ in Alg. 2.
Randomly select initial ego pose $\mathbf{T}_0^{\text{ego}} \sim \text{Uniform}(\{\mathbf{T}_t\}_{t=1}^{T})$.
Find $\mathbf{T}_{\text{fwd}}$ and $\mathbf{T}_{\text{bwd}}$ at distance $\pm d_{\text{pre}}$ along $\{\mathbf{T}_t\}_{t=1}^{T}$ from $\mathbf{T}_0^{\text{ego}}$.
$\mathcal{A} \leftarrow \text{SpawnAgents}(\mathbf{T}_0^{\text{ego}}, \text{True}) \cup \text{SpawnAgents}(\mathbf{T}_{\text{fwd}}) \cup \text{SpawnAgents}(\mathbf{T}_{\text{bwd}})$
// Phase 2: Closed-Loop Simulation
$\Delta d_{\text{ego}} \leftarrow 0, \quad \mathcal{S}_{\text{out}} \leftarrow \emptyset$ ▷ Init movement tracker and empty sequence for frames
for $t = 1 \dots \phi$ **do**
    // Step 2.1: Rolling Horizon Update
    $\Delta d_{\text{ego}} \leftarrow \Delta d_{\text{ego}} + v_{\text{ego}}^{(t)} \Delta t$
    if $\Delta d_{\text{ego}} \geq d_{\text{roll}}$ **then**
        $\mathcal{A} \leftarrow \{a \in \mathcal{A} \mid \mathbf{p}_a \in \text{FOV}(\mathbf{T}_t^{\text{ego}})\}$ ▷ Remove out-of-view agents
        $\mathbf{T}_{\text{fwd}} \leftarrow$ Pose ahead of $\mathbf{T}_t^{\text{ego}}$ by d_{pre}, $\mathbf{T}_{\text{bwd}} \leftarrow$ Pose behind of $\mathbf{T}_t^{\text{ego}}$ by d_{pre}
        $\mathcal{A} \leftarrow \mathcal{A} \cup \text{SpawnAgents}(\mathbf{T}_{\text{fwd}}) \cup \text{SpawnAgents}(\mathbf{T}_{\text{bwd}})$
        $\Delta d_{\text{ego}} \leftarrow 0$
    // Step 2.2: IDM Kinematics & Route Execution
    for each agent $a \in \mathcal{A}$ parameterized by state $(\mathbf{p}_a, \mathbf{h}_a, v_a, \pi_{\text{traj}}, \mathbf{p}_{\text{target}})$ **do**
        Let $\mathbf{h}_a$ denote the heading unit vector of agent a.
        $\mathcal{C}_a \leftarrow \left\{ a' \in \mathcal{A} \setminus \{a\} \;\middle|\; \frac{(\mathbf{p}_{a'} - \mathbf{p}_a) \cdot \mathbf{h}_a}{\|\mathbf{p}_{a'} - \mathbf{p}_a\|_2} > 0.5 \wedge \text{Dist}(\mathbf{p}_{a'}, \pi_{\text{traj}}) < d_{\text{lat}} \right\}$
        $a_{\text{lead}} \leftarrow \arg\min_{a' \in \mathcal{C}_a} \|\mathbf{p}_{a'} - \mathbf{p}_a\|_2$ ▷ Select out the agent ahead of current a
        $s \leftarrow \|\mathbf{p}_{a_{\text{lead}}} - \mathbf{p}_a\|_2, \Delta v \leftarrow v_a - v_{\text{lead}}$ ▷ adjusted if head-on
        if $s < d_{\text{lc}}$ **and** $(\Delta v > 0$ **or** $\mathbf{h}_a \cdot \mathbf{h}_{a_{\text{lead}}} < -0.5)$ **then**
            $\mathbf{p}_{\text{adj}} \leftarrow \text{SearchParallelLane}(\mathbf{p}_a, \mathcal{L}_{\text{final}})$
            if $\mathbf{p}_{\text{adj}} \neq \emptyset$ **then**
                $\pi_{\text{bezier}} \leftarrow \text{BezierCurve}(\mathbf{p}_a, \mathbf{p}_{\text{adj}})$ ▷ Generate smooth transition
                $\pi_{\text{new}} \leftarrow \text{A-Star}(\mathcal{L}_{\text{final}}, \mathbf{p}_{\text{adj}}, \mathbf{p}_{\text{target}})$ ▷ Re-route from adjacent lane
                $\pi_{\text{traj}} \leftarrow \pi_{\text{bezier}} \oplus \pi_{\text{new}}$ ▷ Concatenate trajectories
                $a_{\text{acc}} \leftarrow \text{IDM}(v_a, v_{\text{des}}, \Delta v, s)$ ▷ Compute acceleration
                $\mathbf{p}_a \leftarrow \text{Advance}(\mathbf{p}_a, \pi_{\text{traj}}, v_a \Delta t)$ ▷ Move along trajectory
    // Step 2.3: Rendering
    $\mathcal{A}_{\text{vis}} \leftarrow \{a \in \mathcal{A} \mid \mathbf{p}_a \in \text{FOV}(\mathbf{T}_t^{\text{ego}})\}$ ▷ Filter visible agents in current view
    $\mathcal{O}_{\mathbf{T}}^{(t)} \leftarrow \text{Crop}(\mathcal{M}_{\text{global}}, \mathbf{T}_t^{\text{ego}})$ ▷ Extract static local background
    $\mathcal{X}^{(t)} \leftarrow \bigcup_{a \in \mathcal{A}_{\text{vis}}} \Phi_{\text{rigid}}(\S_a, \mathbf{p}_a, \mathbf{h}_a)$ ▷ Transform and aggregate dynamic assets
    $\mathcal{M}_{\text{local}}^{(t)} \leftarrow \mathcal{O}_{\mathbf{T}}^{(t)} \circledast \mathcal{X}^{(t)}$ ▷ Overlay foreground onto static background
    $\mathcal{S}_{\text{out}} \leftarrow \mathcal{S}_{\text{out}} \cup \{\mathcal{M}_{\text{local}}^{(t)}\}$ ▷ Append rendered frame to sequence
return $\mathcal{S}_{\text{out}}$

10 POTENTIAL LIMITATION AND FUTURE WORK

Although OccSim provide the first implementation of using occupancy world model in simulation, we still observe the following limitation in design and evaluation period of OccSim:

- **Heuristic based fusion and lane graph extraction**: Although the robustness of our multi-frame fusion/lane graph extraction algorithm has been qualitatively demonstrated in numerous practical applications, given the occasional instability in the quality of the raw occupancy frames output by the model, we still seek an end-to-end fusion approach rather than frame-by-frame fusion.
- **Latent-space warping.** W-DiT warps in latent space, assuming the VAE is approximately equivariant to planar rigid transforms. The U-Net refinement absorbs the resulting interpolation error in practice, but warping directly in occupancy space or learning an equivariant latent would remove this approximation.
- **Limited training data**: Compared to image data, semantic occupancy data requires manual semantic annotation, which limits the total amount of available training data to fewer than 100,000 frames. This volume is significantly smaller than the amount of available RGB video data, and it also restricts our ability to explore the upper limits of the model's performance.
- **IDM based agent behavior simulation**: In this paper, we use IDM, a classic traffic flow control model to simulate the behavior of vehicles. Although this approach has proven effective in downstream tasks, we still hope to find more effective ways to enable this aspect of control to learn from the data, or even integrate it into the process of generating static occupancy.
- **Open-loop evaluation.** Existing closed-loop driving benchmarks expose either raw sensor inputs (e.g., NAVSIM) or vectorized map and object states (e.g., nuPlan, Waymax), and provide no semantic occupancy interface, so occupancy-input models cannot be evaluated there in closed loop. All our downstream metrics (mIoU, IoU and planning L2) are therefore computed open-loop against logged futures; closed-loop evaluation of learned policies inside OccSim is left for future work.