EVOWORLD: EVOLVING GAUSSIANS FOR 4D OCCUPANCY FORECASTING AND MOTION PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Future 3D semantic occupancy forecasting and motion planning are central to autonomous driving, as they require models to reason about how surrounding scenes evolve and how the ego vehicle should act. Vision-centric world models must infer this evolution from camera observations, yet many existing approaches discretize scenes into latent embeddings, volumetric features, or quantized tokens and forecast future states through fixed-step autoregressive generation. This limits its temporal flexibility and makes the predicted scene evolution difficult to interpret. We propose *EvoWorld*, a vision-centric, non-autoregressive occupancy world model that predicts a compact set of 4D semantic Gaussian primitives from multi-view camera histories to represent the entire forecast horizon. For each primitive, EvoWorld directly predicts conditional spatial geometry, temporal support, and planar velocity, specifying its shape, motion, and temporal contribution at any queried time. Conditioning and splatting these primitives yields semantic occupancy at arbitrary timestamps, with independent temporal queries that can be decoded in parallel. Primitive velocity is further decomposed into shared ego-induced scene motion and object-level dynamic motion, and the same Gaussian representation supports open-loop motion planning. Extensive experiments on Occ3D-nuScenes and Occ3D-Waymo demonstrate improved vision-centric future occupancy forecasting and strong motion planning performance, with further analyses showing accurate primitive motion and consistent object association.

1 INTRODUCTION

Autonomous driving requires models to perceive the surrounding scene, forecast its future evolution, and plan ego trajectories accordingly. To support end-to-end driving from camera observations (Wu et al., 2022), many methods use Bird's-Eye View (BEV) representations to organize scene information for perception, prediction, and planning (Jiang et al., 2023; Hu et al., 2023; 2022). However, BEV often compresses the vertical dimension, making it less suitable for fine-grained 3D scene understanding. Semantic occupancy addresses this limitation by representing 3D space with dense geometric and semantic labels, and has shown strong potential for planning (Tong et al., 2023). Building on this representation, *4D occupancy forecasting* (Zheng et al., 2024a) further predicts future semantic occupancy from historical observations, providing a predictive world model for autonomous driving.

Existing vision-centric occupancy world models (Zheng et al., 2024a; Xu et al., 2025; Yan et al., 2025; Li et al., 2025; Dang et al., 2026; Liao et al., 2025b) typically encode the observed 3D scene into latent states, such as discrete tokens (Figure 1-(a)), dense volumetric features (Figure 1-(b)), or sparse point queries, and then forecast future states, often sequentially in an autoregressive manner, before decoding them into occupancy. *Despite promising progress, existing occupancy world models still face three limitations in commonly used forecasting pipelines. First*, several methods rely on multi-stage pipelines with separate perception, embedding, and forecasting modules, making end-to-end optimization more complex. *Second*, the encode–predict–decode design models future dynamics in latent space, which can make geometric and motion attributes difficult to interpret. *Third*, autoregressive forecasting restricts prediction to predefined time steps and requires all intermediate future states to be generated sequentially before reaching a target horizon; thus, farther-time queries require multiple dependent rollouts, increasing inference latency and computation. These limitations reduce both pipeline simplicity and temporal flexibility for downstream tasks.

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.

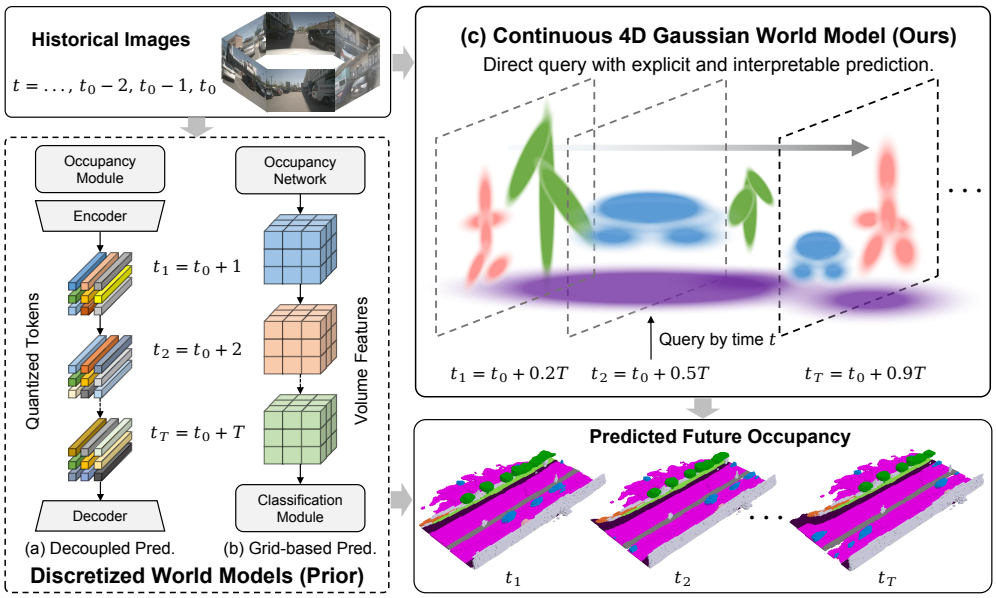


Figure 1: Comparison between *discretized* occupancy world models and our *continuous* 4D Gaussian world model. Existing methods typically forecast future occupancy through (a) discrete tokens or (b) volumetric features at fixed timestamps, often requiring sequential rollout. Our **EvoWorld** shown in (c) represents the scene as evolving 4D semantic Gaussian primitives that can be directly queried at arbitrary timestamps and splatted into future semantic occupancy volumes, enabling explicit, interpretable, and non-autoregressive scene forecasting.

To address these limitations, we introduce *EvoWorld*, a continuous, non-autoregressive occupancy world model that constructs a set of evolving 4D semantic Gaussian primitives from historical multi-view camera observations. The same primitives represent the entire unseen horizon and support direct queries at arbitrary timestamps, distinguishing EvoWorld from models that predict Gaussian states at discrete future timesteps (Zheng et al., 2024b; Zhu et al., 2026) or model scene dynamics in a tokenized space (Feng et al., 2025). As shown in Figure 1-(c), each primitive is grounded in space and time and carries explicit geometric, semantic, temporal, and motion attributes. Our reparameterization separately predicts spatial geometry, temporal support, and velocity, giving *separate control over geometry and motion* and enabling a planar-motion prior while retaining full 3D spatial geometry. To model future scenes in the moving ego frame, EvoWorld incorporates ego-vehicle state context and decomposes primitive velocity into shared ego-induced scene motion and object-level dynamic motion. Learning the shared component once for the scene allows per-primitive motion predictions to focus on object-level residuals. Conditioning and splatting these primitives at *arbitrary* timestamps allows future occupancy to be decoded independently and in parallel. The same set of 4D Gaussian primitives supports planning through a single ego-conditioned query and an MLP, with Gaussian anchors providing explicit spatiotemporal context before occupancy decoding.

By combining these designs, EvoWorld outperforms OccWorld (Zheng et al., 2024a), PreWorld (Li et al., 2025), and SparseWorld (Dang et al., 2026) in vision-centric 4D occupancy forecasting on Occ3D-nuScenes (Tian et al., 2023). On Occ3D-Waymo (Tian et al., 2023), it improves forecasting over SparseWorld at every evaluated horizon. These results show that an explicit Gaussian world can recover predictive 3D scene structure from camera observations while supporting competitive open-loop planning. Continuous querying is also supported by implicit occupancy fields (Agro et al., 2024; Diehl et al., 2025; Ljungbergh et al., 2026); EvoWorld instead predicts a finite set of semantic Gaussians for the unseen future horizon, with explicit geometry, temporal support, and separately learned scene and object motion. Our contributions are:

- We introduce *EvoWorld*, a vision-centric, non-autoregressive occupancy world model based on evolving 4D semantic Gaussians that can be trained end-to-end.
- We reparameterize 4D Gaussians to separately control spatial geometry, temporal support, and planar velocity, with velocity decomposed into ego-induced and object-level components.

- We design a simple yet effective planning head that predicts ego trajectories from the same 4D Gaussian primitives used for occupancy forecasting.
- Experiments on Occ3D-nuScenes and Occ3D-Waymo (Tian et al., 2023) demonstrate improved semantic occupancy forecasting and strong open-loop planning performance, with accurate primitive motion and consistent object association.

2 Related Work

Gaussian scene representations. The GaussianFormer series (Huang et al., 2024; 2025) predicts current occupancy with semantic Gaussians, and GaussianWorld (Zuo et al., 2025) extends them to streaming scene modeling. GaussianOcc (Gan et al., 2025) and GaussianFlowOcc (Boeder et al., 2025) use Gaussian rendering to learn occupancy without dense 3D voxel labels. GaussianAD (Zheng et al., 2024b) predicts Gaussian displacements at fixed future timestamps for forecasting and planning; DLWM (Zhu et al., 2026) pretrains Gaussian dynamics through latent prediction and forecasts through autoregressive updates. EvoWorld predicts *one shared set of 4D semantic Gaussian primitives* that supports occupancy queries at arbitrary future timestamps. Appendix E discusses dynamic Gaussian representations for scene reconstruction and novel view synthesis.

Occupancy world models. Existing methods forecast future occupancy using volumetric features, discrete tokens, or sparse queries (Zheng et al., 2024a; Li et al., 2025; Yan et al., 2025; Dang et al., 2026). GWM (Feng et al., 2025) incorporates Gaussian priors into a tokenized forecasting representation, while SparseWorld (Dang et al., 2026) evolves dynamic sparse queries. SparseWorld-TC (Du et al., 2026) studies trajectory-conditioned occupancy forecasting. EvoWorld jointly predicts occupancy and ego motion from one Gaussian state, with direct time conditioning and explicit scene and object motion (Appendix E).

Continuous occupancy representations. Implicit occupancy fields support continuous spatial and temporal querying (Agro et al., 2024; Diehl et al., 2025). QueryOcc (Lilja et al., 2026) uses adjacent-frame 4D query supervision, and GASP (Ljungbergh et al., 2026) studies unified representation learning for forecasting and ego-trajectory prediction. Compared with implicit representations, EvoWorld exposes geometry, temporal support, and velocity as Gaussian attributes, combining flexible temporal querying with interpretable scene evolution and motion planning. Appendix E details the differences in input modalities, prediction targets, and temporal representations.

3 Methodology

3.1 Preliminary

For a vehicle at timestamp t, vision-centric *3D occupancy prediction* takes N synchronized camera images $\mathbf{S}_t = \{I_t^{(n)}\}_{n=1}^{N}$ as input and estimates a semantic occupancy volume $\mathbf{Y}_t \in [0,1]^{X\times Y\times Z\times C}$, where (X, Y, Z) denotes the spatial resolution of the discretized 3D volume, and C is the number of semantic categories. Vision-centric *4D occupancy forecasting* extends 3D occupancy prediction from current scene reconstruction to future scene evolution. For *motion planning*, the model predicts future ego waypoints in the ground plane over the forecasting horizon. We evaluate planning in the open-loop setting.

3.2 Decoupled 4D Gaussian Reparameterization

To obtain a continuous and interpretable representation of future scene evolution, we represent the entire forecasting horizon with one set of 4D semantic Gaussian primitives. Each primitive carries explicit spatial geometry, temporal support, semantic attributes, and motion dynamics. Time-conditioned primitives are sliced and splatted into occupancy volumes, enabling direct queries at arbitrary future timestamps without sequential rollout.

The 4D parameterization determines how the model learns primitive geometry and motion. Full 4D rotation–scale parameterizations (Yang et al., 2024; Duan et al., 2024) form a joint space–time covariance $\mathbf{\Sigma}_{4D}$ through 4D rotation and scaling, as detailed in Appendix A.1. We decompose the joint covariance into a spatial covariance block $\mathbf{\Sigma}_{ss}$, a space–time cross-covariance $\mathbf{\Sigma}_{st}$, and a

temporal variance σ_t^2. Conditioning this 4D Gaussian on a timestamp t yields a 3D Gaussian with time-invariant spatial covariance $\mathbf{\Sigma}_{s|t}$ and a center that moves at constant velocity $\mathbf{v}$, given by

$$\mathbf{v} = \frac{\mathbf{\Sigma}_{st}}{\sigma_t^2}, \qquad \mathbf{\Sigma}_{s|t} = \mathbf{\Sigma}_{ss} - \frac{\mathbf{\Sigma}_{st}\mathbf{\Sigma}_{st}^\top}{\sigma_t^2} \quad . \tag{1}$$

The full 4D rotation–scale parameterization therefore couples conditional geometry and velocity through its rotation and scale parameters. EvoWorld instead *directly* predicts $\mathbf{\Sigma}_{s|t}$, σ_t, and $\mathbf{v}$, giving the model separate control over geometry and motion. This enables planar velocity constraints and decomposition into shared scene motion and object-level residuals.

Figure 2 illustrates this separation. The reparameterized time-conditioned primitive is expressed as:

$$\mathbf{x} \mid t \sim \mathcal{N}\big(\boldsymbol{\mu}_s + \mathbf{v}(t - \mu_t),\ \mathbf{\Sigma}_{s|t}\big) \quad . \tag{2}$$

The spatial mean $\boldsymbol{\mu}_s$ denotes the primitive center at the reference time μ_t. The vector $\mathbf{v}$ describes how the primitive center moves as the query time changes. The conditional covariance $\mathbf{\Sigma}_{s|t}$ specifies the spatial extent and orientation of the sliced 3D Gaussian, and is shared across queried timestamps. The temporal variance σ_t^2 determines the temporal support of the primitive, *i.e.*, how strongly it contributes when the queried timestamp t moves away from μ_t through a Gaussian temporal weight.

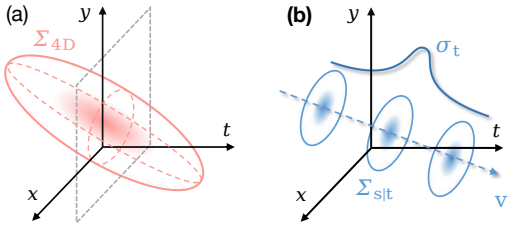


Figure 2: Unlike a full 4D covariance shown in (a) that couples space and time, our structured formulation shown in (b) decouples spatial covariance, temporal support, and motion for interpretable continuous occupancy forecasting. For visualization, we only show the x-y plane and omit the z-axis.

The conditional model corresponds to a joint 4D Gaussian with spatial covariance $\mathbf{\Sigma}_{ss} = \mathbf{\Sigma}_{s|t} + \sigma_t^2 \mathbf{v}\mathbf{v}^\top$ and space-time cross-covariance $\mathbf{\Sigma}_{st} = \sigma_t^2 \mathbf{v}$. This construction maintains a valid joint Gaussian while directly exposing the geometry and motion used for forecasting. Appendix A.2 derives the corresponding joint covariance and verifies its positive definiteness.

We parameterize $\mathbf{\Sigma}_{s|t}$ using the 3D rotation-scale decomposition in GaussianFormer (Huang et al., 2024): $\mathbf{\Sigma}_{s|t} = \mathbf{R}\,\mathrm{diag}(s_x^2, s_y^2, s_z^2)\mathbf{R}^\top$, where $\mathbf{R}$ controls spatial orientation and (s_x, s_y, s_z) control spatial extent. The rotation acts only on spatial coordinates, without mixing space and time. With geometry, velocity, and temporal scale parameterized separately, the network can learn to adjust a primitive's velocity without changing its conditional shape or temporal support, or refine its spatial extent without changing its center trajectory.

Scene dynamics in autonomous driving are predominantly planar. We therefore constrain primitive velocity to the ground plane:

$$\mathbf{v} = \begin{bmatrix} v_x & v_y & 0 \end{bmatrix}^\top \quad , \tag{3}$$

This prior concentrates motion prediction on the ground plane and removes vertical space–time correlation while preserving the full 3D conditional covariance $\mathbf{\Sigma}_{s|t}$. Appendix A.4 quantifies the reduction in parameters per Gaussian, while Appendix C.5 evaluates the planar-motion prior.

At inference time, we condition each 4D primitive on a requested timestamp t using Equation 2. Its temporal support is measured by:

$$w(t) = \exp\left(-\frac{(t - \mu_t)^2}{2\sigma_t^2}\right) \quad , \tag{4}$$

which modulates the opacity o to form the effective splatting weight $\tilde{o}(t) = o \cdot w(t)$. The resulting time-conditioned Gaussians are splatted with $\tilde{o}(t)$ and semantic logits $\mathbf{c}$ to produce the semantic occupancy prediction at timestamp t. Conditioning and splatting at requested timestamps can run independently and in parallel, reusing the Gaussian world without autoregressive rollout.

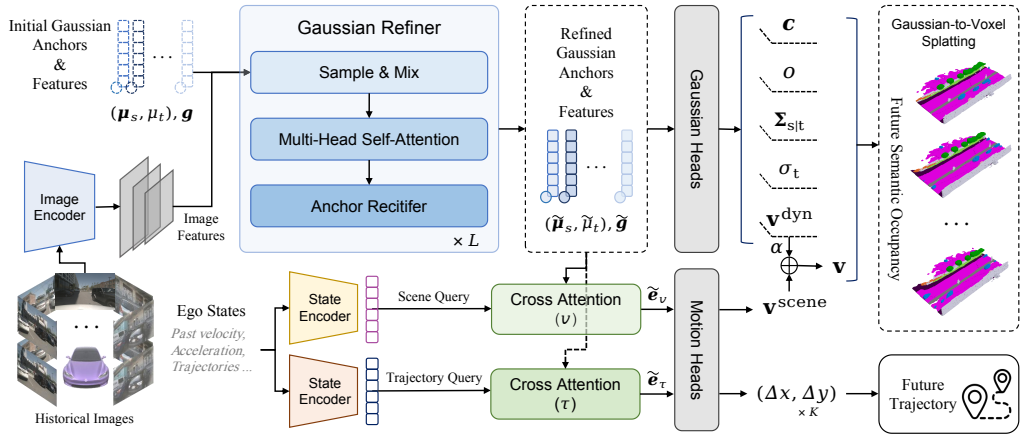


Figure 3: Overall pipeline of EvoWorld. Historical multi-view images and ego states are encoded to refine spatiotemporal Gaussian anchors and features. The refined Gaussians are decoded into semantic, geometric, temporal, and motion attributes, then sliced and splatted into future semantic occupancy volumes, while separate ego-conditioned queries support Gaussian velocity decomposition and future motion trajectory prediction.

3.3 EvoWorld: Occupancy Forecasting and Motion Planning

Figure 3 illustrates the overall EvoWorld pipeline. Each Gaussian primitive is represented by a spatiotemporal anchor and a latent feature. For the q-th primitive, we define:

$$\mathbf{a}_q = (\boldsymbol{\mu}_{s,q}, \mu_{t,q}) \ , \qquad \mathbf{g}_q \in \mathbb{R}^D \ , \tag{5}$$

where $\boldsymbol{\mu}_{s,q}$ and $\mu_{t,q}$ denote its spatial center and temporal anchor, respectively. The anchor provides the geometric state for projection and refinement, while the feature $\mathbf{g}_q$ encodes the information used to predict Gaussian attributes, *i.e.*, semantics, opacity, scale, rotation, motion, and temporal support.

Gaussian Refiner. To align Gaussian primitives with image observations, stacked refinement blocks aggregate image features, exchange global context, and rectify anchors before attribute decoding. First, each Gaussian primitive aggregates multi-view, multi-scale image features through adaptive spatiotemporal feature sampling and adaptive mixing (Liu et al., 2023). Next, we use M learnable latent features $\mathbf{f}_B \in \mathbb{R}^{M\times D}$, with $M \ll Q$, to model global interactions among the Q Gaussian primitive features $\mathbf{f}_G = \{\mathbf{g}_q\}_{q=1}^{Q} \in \mathbb{R}^{Q\times D}$. We use inducing-feature attention (Lee et al., 2019):

$$\mathbf{f}_Z = \mathrm{MHAttn}(\mathbf{f}_B, \mathbf{f}_G, \mathbf{f}_G) \ , \qquad \tilde{\mathbf{f}}_G = \{\tilde{\mathbf{g}}_q\}_{q=1}^{Q} = \mathrm{MHAttn}(\mathbf{f}_G, \mathbf{f}_Z, \mathbf{f}_Z) \ . \tag{6}$$

Finally, a residual rectifier updates each anchor with predicted spatial and temporal offsets. Its update rule is given in Appendix B.1.

Gaussian Heads. Let $\tilde{\mathbf{g}}_q$ denote the final refined feature of the q-th Gaussian primitive. We decode it into the attributes needed for continuous-time slicing and occupancy splatting, including semantic logits, opacity, spatial scale, rotation, object-level dynamic velocity, and temporal support:

$$\begin{aligned} &\mathbf{c}_q = \mathrm{MLP}_c(\tilde{\mathbf{g}}_q) \ , && o_q = \sigma(\mathrm{MLP}_o(\tilde{\mathbf{g}}_q)) \ , && \mathbf{s}_q = \mathrm{MLP}_s(\tilde{\mathbf{g}}_q) \ , \\ &\mathbf{r}_q = \mathrm{norm}(\mathrm{MLP}_r(\tilde{\mathbf{g}}_q)) \ , && \mathbf{v}_q^{\text{dyn}} = \mathrm{MLP}_{v_d}(\tilde{\mathbf{g}}_q) \ , && \sigma_{t,q} = \mathrm{MLP}_t(\tilde{\mathbf{g}}_q) \ . \end{aligned} \tag{7}$$

Here $\mathbf{s}_q$ and $\mathbf{r}_q$ parameterize the conditional spatial covariance $\boldsymbol{\Sigma}_{s|t,q}$, while $\sigma_{t,q}$ determines the primitive's temporal support.

Gaussian Velocity Decomposition. Explicit velocity prediction enables *scene and object motion decomposition* with independently predicted conditional geometry. The original full 4D rotation–scale parameterization encodes their combined effect through space–time covariance. We separate each primitive's velocity into shared ego-induced scene motion $\mathbf{v}^{\text{scene}}$ and object-level motion $\mathbf{v}_q^{\text{dyn}}$. The former captures common apparent motion from ego translation, while the latter captures independent object motion. Let $\mathbf{e}$ denote the ego state. As shown at the bottom of Figure 3, we use a state

encoder to project $\mathbf{e}$ into a scene query $\mathbf{q}_v = \psi_v(\mathbf{e})$, which attends to the refined Gaussian primitive features $\tilde{\mathbf{f}}_G$. The motion head MLP_v predicts one scene velocity shared by all primitives, allowing the per-primitive head to focus on object-level residuals. The negative sign expresses apparent scene motion opposite to ego translation:

$$\tilde{\mathbf{e}}_v = \mathrm{CrossAttn}_v\left(\mathbf{q}_v, \tilde{\mathbf{f}}_G, \tilde{\mathbf{f}}_G\right) \quad , \qquad \mathbf{v}^{\mathrm{scene}} = -\mathrm{MLP}_v\left(\tilde{\mathbf{e}}_v\right) \quad . \tag{8}$$

For primitives associated with dynamic semantic classes, we add the object-level dynamic velocity predicted by the Gaussian head, modulated by the primitive's dynamic probability:

$$\mathbf{v}_q = \mathbf{v}^{\mathrm{scene}} + \alpha_q \mathbf{v}_q^{\mathrm{dyn}} \quad , \tag{9}$$

with $\alpha_q \in [0, 1]$ estimated from the semantic distribution. Static primitives have small α_q and follow the ego-induced scene motion, while dynamic primitives have larger α_q and move independently. These explicit velocities also yield voxel flow through occupancy-weighted aggregation at any queried timestamp (Appendix C.3).

Motion Planning. For planning, a state encoder maps the ego state to one trajectory query $\mathbf{q}_\tau = \psi_\tau(\mathbf{e})$. The query attends to the refined Gaussian features with their 4D anchors, and an MLP predicts future ego displacements directly from this state, before occupancy splatting. We stop planning gradients to both the Gaussian features $\tilde{\mathbf{f}}_G$ and their anchors:

$$\tilde{\mathbf{e}}_\tau = \mathrm{CrossAttn}_{\tau,\mathrm{pos}}\left(\mathbf{q}_\tau, \mathrm{sg}(\tilde{\mathbf{f}}_G), \mathrm{sg}(\tilde{\mathbf{f}}_G)\right) \quad , \qquad \hat{\boldsymbol{\tau}}_{1:K} = \mathrm{MLP}_\tau\left(\tilde{\mathbf{e}}_\tau\right) \quad . \tag{10}$$

Here positional encodings of the detached refined 4D anchors are included in the attention keys, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The anchors associate each attended feature with a scene location and time, giving the simple readout explicit spatiotemporal context. Planning therefore uses the occupancy-trained Gaussian state directly and avoids future occupancy decoding in its computation path. The output $\hat{\boldsymbol{\tau}}_{1:K} = \{\hat{\boldsymbol{\tau}}_k\}_{k=1}^{K}$ denotes the planned future ego trajectory, where each $\hat{\boldsymbol{\tau}}_k \in \mathbb{R}^2$ represents the incremental planar displacement $(\Delta x_k, \Delta y_k)$ at step k. Accumulating these displacements gives the ego waypoints used for planning evaluation.

Voxelization and Supervision. At a query timestamp t, each 4D Gaussian is sliced into a conditional 3D Gaussian, and the resulting primitives are splatted into a semantic occupancy volume using local aggregation, following GaussianFormer-2 (Huang et al., 2025).

We supervise semantic occupancy with ground-truth semantics at each training timestamp using semantic cross-entropy and Lovasz loss (Berman et al., 2018). Motion planning uses mean squared error between predicted and ground-truth incremental ego displacement. The final objective is $\mathcal{L} = \lambda_{\mathrm{ce}}\mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{lov}}\mathcal{L}_{\mathrm{lov}} + \lambda_{\mathrm{plan}}\mathcal{L}_{\mathrm{plan}}$, with $\lambda_{\mathrm{plan}} = 1$. Appendix B.2 gives the loss definitions.

4 Experiments

4.1 Experimental Settings

We evaluate semantic occupancy forecasting and open-loop planning on Occ3D-nuScenes and Occ3D-Waymo (Tian et al., 2023), which provide dense semantic occupancy annotations for nuScenes (Caesar et al., 2020) and Waymo (Sun et al., 2020). Each annotation covers a spatial range of $[-40, 40]\,\mathrm{m}$, $[-40, 40]\,\mathrm{m}$, and $[-1, 5.4]\,\mathrm{m}$ around the ego vehicle. The ground-truth occupancy is represented as a $200 \times 200 \times 16$ voxel grid with $0.4\,\mathrm{m}$ resolution. The evaluation uses 17 occupied classes and free space on nuScenes, and 10 occupied classes and free space on Waymo. Following common evaluation protocols, we report IoU and mIoU for 4D occupancy forecasting. IoU measures foreground-background occupancy overlap, while mIoU averages semantic IoU over semantic classes. For motion planning, we report L2 error and collision rate. Unless stated otherwise, EvoWorld uses 25,600 Gaussians. Appendix B.3 details training and dataset protocols and Appendix C.1 examines hyperparameter sensitivity.

Table 1: **4D occupancy forecasting performance on Occ3D-nuScenes and Occ3D-Waymo.** Bold and underlined values denote the best and second-best results within each nuScenes input group; best Waymo results are bold. LiDAR init denotes initialization from LiDAR.

Method	Input	Aux. Sup.	mIoU ↑				IoU ↑			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
Occ3D-nuScenes: camera input										
OccWorld-T (Zheng et al., 2024a)	Camera	LiDAR	4.68	3.36	2.63	3.56	9.32	8.23	7.47	8.34
OccWorld-D	Camera	3D Occ	11.55	8.10	6.22	8.62	18.90	16.26	14.43	16.53
PreWorld (Li et al., 2025)	Camera	3D Occ	11.69	8.72	6.77	9.06	23.01	20.79	18.84	20.88
+ Pre-training	Camera	2D & 3D Occ	12.27	9.24	7.15	9.55	23.62	21.76	19.63	21.62
SparseWorld (Dang et al., 2026)	Camera	3D Occ	14.93	13.15	**11.51**	13.20	22.96	22.10	21.05	22.03
EvoWorld-25,600	Camera	3D Occ	**15.71**	**13.62**	11.46	**13.60**	**25.56**	**24.03**	**21.86**	**23.82**
Occ3D-nuScenes: additional inputs or LiDAR initialization										
OccLLaMA-F (Wei et al., 2024)	Camera+Action+Language	3D Occ	10.34	8.66	6.98	8.66	25.81	23.19	19.97	22.99
GWM (Feng et al., 2025)	Camera+LiDAR	3D Occ	11.63	10.07	8.17	10.12	26.22	24.97	22.13	24.60
EvoWorld-25,600	Camera, with LiDAR init	3D Occ	**18.66**	**15.93**	**13.02**	**15.87**	**31.56**	**28.94**	**25.42**	**28.64**
Occ3D-Waymo										
SparseWorld	Camera	3D Occ	17.38	14.96	13.15	15.16	38.82	36.17	33.83	36.27
EvoWorld-12,800	Camera	3D Occ	**17.71**	**15.83**	**13.98**	**15.84**	**42.39**	**41.08**	**39.77**	**41.08**

Table 2: **Motion planning performance on Occ3D-nuScenes and Occ3D-Waymo.** We use bold and underlined numbers to denote the best and second-best results within each dataset, respectively.

Method	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
Occ3D-nuScenes									
VAD-Tiny (Jiang et al., 2023)	Map & Box & Motion	0.46	0.76	1.12	0.78	0.21	0.35	0.58	0.38
VAD-Base	Map & Box & Motion	0.41	0.70	1.05	0.72	**0.07**	0.17	0.41	0.22
OccWorld-D (Zheng et al., 2024a)	3D Occ	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87
PreWorld (Li et al., 2025)	3D Occ	0.22	0.31	0.41	0.31	0.36	0.52	0.73	0.54
+ Pre-training	2D & 3D Occ	0.22	0.30	0.40	0.31	0.21	0.66	0.71	0.53
SparseWorld (Dang et al., 2026)	3D Occ	**0.19**	**0.25**	0.36	**0.27**	0.11	0.29	0.46	0.29
EvoWorld-25,600	3D Occ	**0.19**	0.26	**0.35**	**0.27**	0.09	**0.14**	**0.26**	**0.16**
Occ3D-Waymo									
SparseWorld	3D Occ	**0.12**	**0.41**	**1.02**	**0.52**	**0.00**	0.21	0.45	0.22
EvoWorld-12,800	3D Occ	0.14	0.45	1.07	0.55	0.02	**0.12**	**0.41**	**0.18**

4.2 Results and Analysis

4D Occupancy Forecasting. Among the camera-input methods in Table 1, EvoWorld achieves the strongest average mIoU and IoU. Its IoU advantage over SparseWorld across all horizons indicates that the explicit Gaussian state preserves occupied scene structure over time, while the stronger average mIoU shows that this continuity does not compromise semantic prediction. At 3 seconds on nuScenes, EvoWorld and SparseWorld have similar mIoU (11.46 and 11.51), while EvoWorld retains higher IoU (21.86 versus 21.05). Appendix C.4 gives per-class results with 12,800 primitives. On Waymo, EvoWorld improves both metrics at every horizon, and its IoU margin grows from 3.57 points at 1 second to 5.94 points at 3 seconds.

Figure 4 compares future occupancy predictions from 1s to 3s. SparseWorld (Dang et al., 2026) captures the global layout but exhibits boundary artifacts and scattered occupied voxels near object surfaces. In contrast, EvoWorld's explicit continuous Gaussians provide smoother spatial support and more consistent temporal evolution, with interpretable geometry and semantics.

Figure 5 illustrates the temporal flexibility of our continuous formulation. Unlike autoregressive methods with predefined 0.5s output intervals, EvoWorld can be directly queried at arbitrary timestamps without sequential intermediate generation. The predictions evolve smoothly and preserve the global layout, enabling intermediate future occupancy outputs beyond fixed timestamps.

LiDAR Initialization. The additional-input section of Table 1 examines the effect of incorporating modalities beyond camera images. LiDAR-based initialization significantly improves the IoU and mIoU of EvoWorld, showing that stronger geometric priors can improve Gaussian placement and lead to more accurate future occupancy forecasting. Its higher forecasting accuracy than GWM further shows that the Gaussian representation can effectively use additional geometric cues.

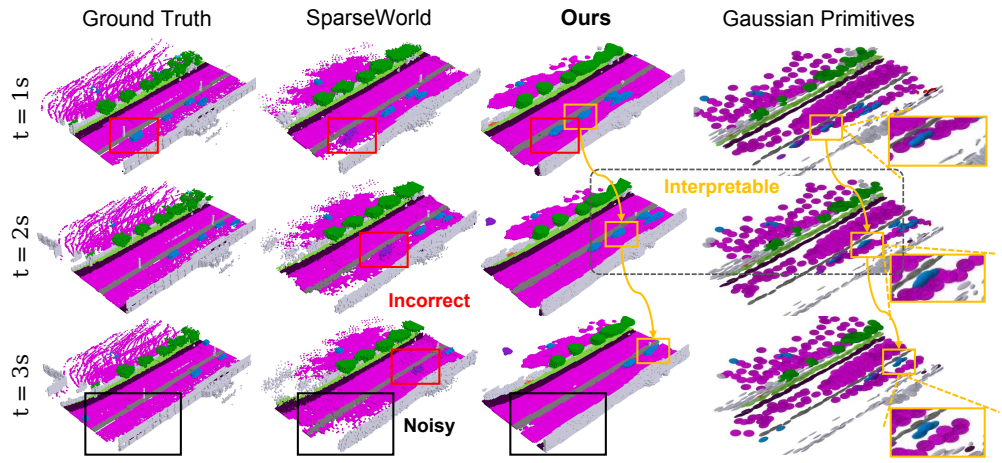


Figure 4: Qualitative comparison of future occupancy forecasting at 1–3 seconds. Compared with SparseWorld (Dang et al., 2026), EvoWorld produces cleaner and more accurate occupancy predictions over time. The right column shows the corresponding Gaussian primitives, where the highlighted blue car primitives remain identifiable, making the predicted motion directly inspectable.

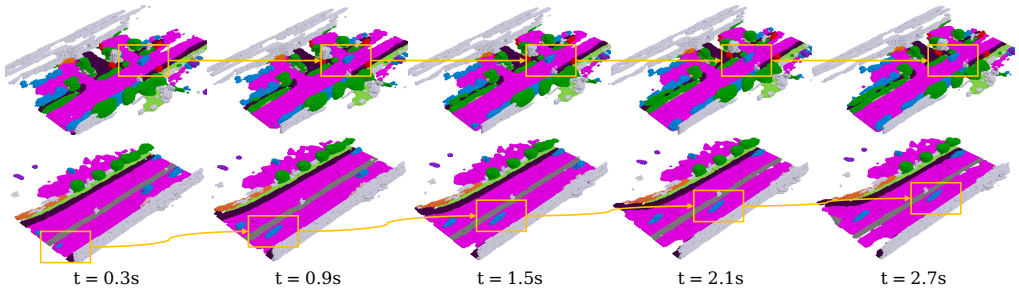


Figure 5: Arbitrary-timestamp occupancy forecasting of two scenes with EvoWorld. The continuous 4D Gaussian world can be queried at any intermediate timestamp without sequential rollout, producing smooth scene evolution while preserving the global layout.

Motion Planning. Table 2 shows that EvoWorld achieves the lowest average open-loop collision rate on both benchmarks with trajectory error comparable to SparseWorld. The planner's only scene input is the occupancy-trained Gaussian state, whose features are tied to 4D anchors through the attention keys. Stop-gradient at both features and anchors prevents planning supervision from updating the Gaussian state. These results demonstrate that the learned world representation supports competitive planning through a simple anchored readout before future occupancy decoding.

4.3 Ablation Study

Gaussian Parameterization and Motion Decomposition. Table 3 compares EvoWorld with single-velocity and full-4D-covariance variants. The single-velocity variant must explain ego-induced apparent motion and independent object motion together; its weaker results support learning these components separately. Full 4D covariance further degrades occupancy and motion accuracy, showing that direct conditional geometry and velocity offer a better learning interface than coupled covariance terms.

Gaussian Primitive Motion Accuracy. To assess how the parameterization affects predicted motion, we evaluate displacement error on tracked dynamic objects. For each object's box at $t = 0$, we take the median predicted planar velocity $\hat{\mathbf{v}}$ of class-matched Gaussian primitives. With GT centers $\mathbf{c}_0$ and $\mathbf{c}_t$, displacement end-point error is

$$\mathrm{EPE}(t) = \left\| \hat{\mathbf{v}}t - (\mathbf{c}_t - \mathbf{c}_0) \right\|_2 . \tag{11}$$

Table 3: **Core design ablations on nuScenes.** All variants use 25,600 Gaussians. Single planar velocity removes motion decomposition; full 4D covariance uses the original parameterization. Forecasting and planning scores are horizon averages. GT tracks are used only to evaluate EPE.

	Forecasting ↑		Planning ↓		Object-level EPE (m) ↓		
Variant	IoU	mIoU	L2 (m)	Col. (%)	1 s	2 s	3 s
EvoWorld	**23.82**	**13.60**	**0.27**	**0.16**	**1.448**	**2.463**	**3.447**
Single planar velocity	23.23	12.88	0.37	0.24	1.537	2.828	4.200
Full 4D covariance	21.83	11.42	**0.27**	0.20	3.824	7.110	9.896

Table 4: **Temporal consistency of primitive-to-object association.** Propagated primitive centers are assigned to class-matched GT boxes. Higher values indicate stronger consistency.

Metric	EvoWorld	Full 4D
Primitives that retain object identity at adjacent timestamps	**93.84%**	78.90%
Primitives that retain object identity throughout the horizon	**86.91%**	62.93%

Table 5: **Accuracy and computational cost.** Measured on one H100 for 100 samples with 10 warm-ups. Each method decodes current timestamp plus six future queries.

Model	Params (M)	mIoU	Latency (ms)	Throughput (Hz)	Memory (GiB)
SparseWorld	76.03	13.20	158.95	6.29	**2.67**
EvoWorld-25,600	62.50	13.60	227.48	4.40	6.38
EvoWorld-12,800	**59.17**	**13.62**	**154.02**	**6.49**	4.40

EvoWorld has the lowest EPE at every horizon (Table 3), with a growing advantage at longer horizons. These results show that separating scene and object motion improves both tracked-object displacement and occupancy forecasting. Appendix C.3 compares predicted speed distributions with ground truth and visualizes derived voxel flow.

Gaussian Primitive Association with Objects. We propagate Gaussian centers and assign them to class-matched GT boxes to test whether their trajectories retain object identity. Adjacent-time retention is measured among primitives assigned at both timestamps. The high identity retention in Table 4 complements the displacement accuracy in Table 3: primitives follow object motion while maintaining their associations, giving the learned geometry and velocity a stable object-level interpretation over time.

Computational Efficiency. Reducing the representation from $25{,}600$ to $12{,}800$ primitives preserves forecasting accuracy while substantially lowering latency and memory use (Table 5). The smaller model uses 22.2% fewer parameters than SparseWorld and reaches comparable latency. Further reducing the count to $6{,}400$ degrades forecasting accuracy, indicating that computational savings depend on sufficient scene coverage (Appendix C.2). Occupancy queries at additional timestamps reuse the predicted Gaussian primitives, allowing downstream tasks to choose their temporal sampling density after world construction. With 12,800 primitives, decoding 31 timestamps across two H100 GPUs gives a $1.40\times$ overall inference speedup (Appendix D).

5 Conclusion

We present EvoWorld, a vision-centric, non-autoregressive occupancy world model based on continuous 4D Gaussian primitives. EvoWorld directly queries a structured Gaussian scene representation at arbitrary future timestamps. This design enables dense semantic occupancy forecasting, interpretable scene evolution, and ego vehicle motion planning within a unified end-to-end framework. Experiments on Occ3D-nuScenes and Occ3D-Waymo demonstrate improved average occupancy forecasting and competitive open-loop planning. The motion decomposition improves object-level displacement accuracy, and primitive-to-object association demonstrates consistent object identity over time. Reducing the representation by half to 12,800 Gaussians preserves forecasting performance while lowering memory footprint and inference time.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have not used generative AI tools for generate synthetic data sets, design or provide feedback on research methodology or experiments, assist with translation and help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper, summarize or analyse existing literature, sourcing/searching for information, identify relevant literature, propose a title or keywords for a research paper. We have reviewed all AI-assisted work. Any LLM-generated code, summaries, or draft text was reviewed and verified. 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 section is **recommended** and does not count toward the page limit.)

This work was conducted in accordance with the ICLR Code of Ethics. Our evaluation relies exclusively on public benchmark datasets, and does not involve human participants, private records, or personally identifiable information. We did not collect any new data, nor does the study make use of sensitive information. The primary contribution is methodological, with the goal of improving techniques for autonomous driving research. We have aimed to present the experimental findings accurately, clearly state the limitations of the proposed approach, and follow established standards for responsible research practice. We are not aware of any conflicts of interest or additional ethical issues specific to this study beyond those generally associated with autonomous driving research.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

To support reproducibility, we report the methodological and experimental details needed to reproduce our results. The main paper and appendix describe the proposed algorithms, assumptions, and derivations. We provide the model architectures, hyperparameter settings, training configurations and evaluation procedures used in all experiments. All datasets are publicly available. The source code will be released upon acceptance.

REFERENCES

Ben Agro, Quinlan Sykora, Sergio Casas, Thomas Gilles, and Raquel Urtasun. Uno: Unsupervised occupancy fields for perception and forecasting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14487–14496. IEEE, 2024.

Arnold Caleb Asiimwe and Carl Vondrick. 4d gaussian splatting as a learned dynamical system. *arXiv preprint arXiv:2512.19648*, 2025.

Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4413–4421. IEEE, 2018.

Simon Boeder, Fabian Gigengack, and Benjamin Risse. Gaussianflowocc: Sparse and weakly supervised occupancy estimation using gaussian splatting and temporal flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24943–24954. IEEE, 2025.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11618–11628. IEEE, 2020.

Cheng Chen, Hao Huang, and Saurabh Bagchi. Vision-only gaussian splatting for collaborative semantic occupancy prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2796–2804, 2026.

Dubing Chen, Jin Fang, Wencheng Han, Xinjing Cheng, Junbo Yin, Chengzhong Xu, Fahad Shahbaz Khan, and Jianbing Shen. Alocc: Adaptive lifting-based 3d semantic occupancy and cost volume-based flow predictions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4156–4166, 2025.

Chenxu Dang, Haiyan Liu, Jason Bao, Pei An, Xinyue Tang, An Pan, Jie Ma, Bingchuan Sun, and Yan Wang. Sparseworld: A flexible, adaptive, and efficient 4d occupancy world model powered by sparse and dynamic queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3497–3505, 2026.

Christopher Diehl, Quinlan Sykora, Ben Agro, Thomas Gilles, Sergio Casas, and Raquel Urtasun. Dio: Decomposable implicit 4d occupancy-flow world model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27456–27466. IEEE, 2025.

Jiayuan Du, Yiming Zhao, Zhenglong Guo, Yong Pan, Wenbo Hou, Zhihui Hao, Kun Zhan, and Qijun Chen. Sparseworld-tc: Trajectory-conditioned sparse occupancy world model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7425–7434, 2026.

Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Tuo Feng, Wenguan Wang, and Yi Yang. Gaussian-based world model: Gaussian priors for voxel-based occupancy prediction and future motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25239–25249. IEEE, 2025.

Wanshui Gan, Fang Liu, Hongbin Xu, Ningkai Mo, and Naoto Yokoya. Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28980–28990. IEEE, 2025.

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.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pp. 533–549. Springer, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17853–17862. IEEE, 2023.

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9223–9232. IEEE, 2023.

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In *European Conference on Computer Vision*, pp. 376–393. Springer, 2024.

Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and Jiwen Lu. Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27477–27486. IEEE, 2025.

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

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

Xiang Li, Pengfei Li, Yupeng Zheng, Wei Sun, Yan Wang, et al. Semi-supervised vision-centric 3d occupancy world model for autonomous driving. In *International Conference on Learning Representations*, volume 2025, pp. 62563–62580, 2025.

Zhimin Liao, Ping Wei, Shuaijia Chen, Haoxuan Wang, and Ziyang Ren. Stcocc: Sparse spatial-temporal cascade renovation for 3d occupancy and scene flow prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1516–1526. IEEE, 2025a.

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. IEEE, 2025b.

Adam Lilja, Ji Lan, Junsheng Fu, and Lars Hammarstrand. Queryocc: Query-based self-supervision for 3d semantic occupancy. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21399–21408, 2026.

Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18534–18544. IEEE, 2023.

William Ljungbergh, Adam Lilja, Adam Tonderski, Arvid Laveno Ling, Carl Lindström, Willem Verbeke, Junsheng Fu, Christoffer Petersson, Lars Hammarstrand, and Michael Felsberg. Gasp: Unifying geometric and semantic self-supervised pre-training for autonomous driving. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3077–3087. IEEE, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Yining Shi, Kun Jiang, Qiang Meng, Ke Wang, Jiabao Wang, Wenchao Sun, Tuopu Wen, et al. Come: Adding scene-centric forecasting control to occupancy world model. In *Advances in Neural Information Processing Systems*, volume 38, pp. 70–96, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2443–2451. IEEE, 2020.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64318–64330, 2023.

Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8372–8381. IEEE, 2023.

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. IEEE, 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.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21672–21683. IEEE, 2023.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6119–6132, 2022.

Tianshuo Xu, Hao Lu, Xu Yan, Yingjie Cai, Bingbing Liu, and Yingcong Chen. Occ-llm: Enhancing autonomous driving with occupancy-based large language models. In *IEEE International Conference on Robotics and Automation*, pp. 8434–8441. IEEE, 2025.

Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pp. 156–173. Springer, 2024.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. Renderworld: World model with self-supervised 3d label. In *IEEE International Conference on Robotics and Automation*, pp. 6063–6070. IEEE, 2025.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations*, volume 2024, pp. 9142–9159, 2024.

Zhangchen Ye, Tao Jiang, Chenfeng Xu, Yiming Li, and Hang Zhao. Cvt-occ: Cost volume temporal fusion for 3d occupancy prediction. In *European Conference on Computer Vision*, pp. 381–397. Springer, 2024.

Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9399–9409. IEEE, 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, 2024a.

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

Yiyao Zhu, Ying Xue, Haiming Zhang, Guangfeng Jiang, Wending Zhou, Xu Yan, Jiantao Gao, Yingjie Cai, Bingbing Liu, Zhen Li, and Shaojie Shen. Dlwm: Dual latent world models enable holistic gaussian-centric pre-training in autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 39713–39723, June 2026.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6772–6781. IEEE, 2025.

A Derivation of the Gaussian Representation

This section derives the conditional representation in Section 3.2, establishes its correspondence to a joint 4D Gaussian, and explains how direct attribute prediction separates geometry from motion.

A.1 Background on Joint 4D Gaussians

Analogous to 3D Gaussian primitives (Kerbl et al., 2023), each 4D primitive in dynamic scenes is defined over a 4D space-time coordinate $\mathbf{u} = [\mathbf{x}^\top, t]^\top \in \mathbb{R}^4$, with a 4D mean $\boldsymbol{\mu}_{4\text{D}} = [\boldsymbol{\mu}_s^\top, \mu_t]^\top$ and covariance matrix $\boldsymbol{\Sigma}_{4\text{D}} \in \mathbb{R}^{4\times4}$. The corresponding unnormalized Gaussian density is:

$$\mathcal{G}(\mathbf{u}) = \exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}_{4\text{D}})^\top \boldsymbol{\Sigma}_{4\text{D}}^{-1}(\mathbf{u}-\boldsymbol{\mu}_{4\text{D}})\right) \quad . \tag{12}$$

The covariance can be parameterized through a 4D rotation-scale decomposition:

$$\boldsymbol{\Sigma}_{4\text{D}} = \mathbf{R}_{4\text{D}}\mathbf{S}_{4\text{D}}\mathbf{S}_{4\text{D}}^\top\mathbf{R}_{4\text{D}}^\top \quad , \tag{13}$$

where $\mathbf{S}_{4\text{D}} = \mathrm{diag}(s_x, s_y, s_z, s_t)$ contains the spatial and temporal scales. The 4D rotation $\mathbf{R}_{4\text{D}}$ can be represented by two unit quaternions (Yang et al., 2024) or a 4D rotor (Duan et al., 2024). This parameterization jointly determines spatial shape and space-time correlation.

Formally, we decompose the 4D covariance into spatial and temporal components. Specifically, we write $\boldsymbol{\Sigma}_{4\text{D}} = \begin{bmatrix}\boldsymbol{\Sigma}_{ss} & \boldsymbol{\Sigma}_{st}\\ \boldsymbol{\Sigma}_{ts} & \sigma_t^2\end{bmatrix}$, where $\boldsymbol{\Sigma}_{ss} \in \mathbb{R}^{3\times3}$ describes the spatial covariance, $\sigma_t^2 \in \mathbb{R}$ denotes the temporal variance, and $\boldsymbol{\Sigma}_{st} \in \mathbb{R}^{3\times1}$ captures the space-time correlation. Under this formulation, a 4D Gaussian primitive can be sliced at any queried timestamp by considering the conditional distribution of space given time. Applying the standard Gaussian conditioning rule yields:

$$\mathbf{x} \mid t \sim \mathcal{N}\big(\boldsymbol{\mu}_s + \boldsymbol{\Sigma}_{st}\sigma_t^{-2}(t-\mu_t),\ \boldsymbol{\Sigma}_{ss} - \boldsymbol{\Sigma}_{st}\sigma_t^{-2}\boldsymbol{\Sigma}_{ts}\big) \quad . \tag{14}$$

The conditional covariance is independent of t, while the conditional mean varies linearly with time. Substituting the velocity and conditional covariance from Equation 1 recovers Equation 2.

A.2 Joint Covariance Construction

Although Equation 2 is expressed as a time-conditioned 3D Gaussian, it is still consistent with a valid joint 4D Gaussian. Specifically, given the learned conditional spatial covariance $\boldsymbol{\Sigma}_{s|t}$, motion coefficient $\mathbf{v}$, and temporal variance σ_t^2, we construct the corresponding joint space-time covariance:

$$\boldsymbol{\Sigma}_{4\text{D}} = \begin{bmatrix}\boldsymbol{\Sigma}_{s|t} + \sigma_t^2\mathbf{v}\mathbf{v}^\top & \sigma_t^2\mathbf{v}\\ \sigma_t^2\mathbf{v}^\top & \sigma_t^2\end{bmatrix} \quad . \tag{15}$$

Under this construction, the space-time cross-covariance is $\boldsymbol{\Sigma}_{st} = \sigma_t^2\mathbf{v}$, while the spatial covariance of the joint 4D Gaussian is $\boldsymbol{\Sigma}_{ss} = \boldsymbol{\Sigma}_{s|t} + \sigma_t^2\mathbf{v}\mathbf{v}^\top$. Substituting these terms into the Gaussian conditioning formula in Equation 14 recovers exactly the conditional model in Equation 2. This defines a valid joint 4D Gaussian with separately predicted conditional geometry, temporal support, and velocity. With unrestricted 3D velocity, the construction reparameterizes a general positive-definite joint covariance. Its positive definiteness follows from the factorization

$$\boldsymbol{\Sigma}_{4\text{D}} = \begin{bmatrix}\mathbf{I} & \mathbf{v}\\ \mathbf{0}^\top & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_{s|t} & \mathbf{0}\\ \mathbf{0}^\top & \sigma_t^2\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{v}^\top & 1\end{bmatrix}, \tag{16}$$

provided $\boldsymbol{\Sigma}_{s|t} \succ 0$ and $\sigma_t^2 > 0$. The planar constraint $v_z = 0$ removes vertical space-time correlation while preserving the full conditional spatial covariance.

A.3 Independent Control of Gaussian Attributes

The two parameterizations in Figure 2 differ in how the conditional attributes are predicted. Equation 1 shows how conditional shape depends on both spatial covariance and space–time correlation under the full covariance formulation. Directly predicting $\mathbf{\Sigma}_{s|t}$, σ_t, and $\mathbf{v}$ permits separate constraints on shape, temporal support, and motion.

For example, a conditional spatial scale of 1 m, temporal scale of 1 s, and velocity of $6.6\,\mathrm{m\,s^{-1}}$ correspond to $\Sigma_{xx} = 44.56\,\mathrm{m}^2$, $\Sigma_{xt} = 6.6\,\mathrm{m\,s}$, and $\Sigma_{tt} = 1\,\mathrm{s}^2$. Recovering the conditional variance requires the subtraction $44.56 - 43.56 = 1\,\mathrm{m}^2$. Errors in the predicted covariance terms therefore affect the recovered shape. Direct conditional prediction separates this shape estimate from the velocity estimate. The decoded velocity can change while the conditional spatial covariance remains fixed, and spatial shape can change while the center trajectory remains fixed. This separation gives occupancy supervision direct control over the attributes that determine scene evolution. The forecasting and motion gains in Table 3 support this parameterization.

A.4 Gaussian Attribute Counts and Degrees of Freedom

Let n count predicted scalar components and d count covariance degrees of freedom. For the two-quaternion full 4D parameterization (Yang et al., 2024), our conditional parameterization, and its planar restriction, respectively,

$$\begin{aligned}(n, d)_{\mathrm{4D}} &= (8 + 4,\ 6 + 4) = (12, 10),\\ (n, d)_{\mathrm{conditional}} &= (4 + 3 + 3 + 1,\ 3 + 3 + 3 + 1) = (11, 10),\\ (n, d)_{\mathrm{planar}} &= (4 + 3 + 2 + 1,\ 3 + 3 + 2 + 1) = (10, 9).\end{aligned} \tag{17}$$

Each unit quaternion contributes four scalar components and three rotational degrees of freedom. The conditional terms count spatial rotation, spatial scales, velocity, and temporal scale; setting $v_z = 0$ removes one motion component. For Q primitives, velocity decomposition adds two shared scene-velocity outputs to the per-primitive attributes:

$$n^{\mathrm{world}}_{\mathrm{EvoWorld}} = 10Q + 2, \qquad n^{\mathrm{world}}_{\mathrm{Full\ 4D}} = 12Q. \tag{18}$$

The conditional parameterization preserves the covariance capacity of a full 4D Gaussian with one fewer scalar output, while exposing spatial geometry, temporal support, and velocity as separately predicted attributes. The planar prior focuses motion learning on the ground plane while retaining full 3D spatial geometry. Sharing scene velocity requires only two additional outputs for the entire Gaussian world, allowing primitive heads to focus on object-level residuals. Table 3 shows the benefits for forecasting and object-motion accuracy, and Table 11 supports the planar-motion prior.

B Implementation Details

B.1 Primitive Refinement and Attribute Decoding

Each refinement block updates the primitive features and rectifies their spatiotemporal anchors. For primitive q, the rectifier predicts spatial and temporal offsets from the updated feature $\tilde{\mathbf{g}}_q$ and adds them to the current anchor:

$$(\Delta\boldsymbol{\mu}_{s,q}, \Delta\mu_{t,q}) = \mathrm{MLP}_{\mathrm{rect}}(\tilde{\mathbf{g}}_q)\ , \qquad \tilde{\mathbf{a}}_q = (\tilde{\boldsymbol{\mu}}_{s,q}, \tilde{\mu}_{t,q}) = \mathbf{a}_q + (\Delta\boldsymbol{\mu}_{s,q}, \Delta\mu_{t,q})\ . \tag{19}$$

After the final block, the heads in Equation 7 decode semantics, opacity, spatial geometry, temporal support, and object-level velocity. The normalized quaternion and spatial scales define $\mathbf{\Sigma}_{s|t,q}$, while $\sigma^2_{t,q}$ gives the temporal variance. The shared scene velocity in Equation 8 combines with the object-level velocities through Equation 9.

B.2 Training Losses

At each training timestamp, ground-truth semantics supervise occupancy through semantic cross-entropy and Lovasz loss (Berman et al., 2018):

$$\mathcal{L}_{\text{occ}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{lov}}\mathcal{L}_{\text{lov}} \quad . \tag{20}$$

For motion planning, $\hat{\boldsymbol{\tau}}$ and $\boldsymbol{\tau}$ denote predicted and ground-truth sequences of incremental planar ego displacements. We supervise these increments using mean squared error:

$$\mathcal{L}_{\text{plan}} = \text{MSE}\left(\hat{\boldsymbol{\tau}}, \boldsymbol{\tau}\right) \quad . \tag{21}$$

For evaluation, the cumulative sum $\hat{\mathbf{p}}_k = \sum_{j=1}^{k} \hat{\boldsymbol{\tau}}_j$ gives the ego waypoint at step k; the same accumulation is applied to ground truth.

The final training objective is $\mathcal{L} = \mathcal{L}_{\text{occ}} + \lambda_{\text{plan}}\mathcal{L}_{\text{plan}}$ where we set λ_{plan} to 1.0 in our experiments.

B.3 Training and Dataset Protocols

On nuScenes, the $200 \times 200 \times 16$ occupancy grid covers $[-40, 40]$ m along both ground-plane axes and $[-1, 5.4]$ m vertically, with 17 occupied classes and free space. The model uses 2 seconds of history to forecast the next 3 seconds. ResNet-50 extracts image features, while a three-layer MLP encodes ego states; three Gaussian refinement blocks then process features of dimension 256 through 1,280 inducing features. Gaussian anchors are randomly initialized in the camera-only setting, and the backbone initialization matches SparseWorld.

For Occ3D-Waymo (Tian et al., 2023), both methods use the same unified forecasting protocol (Wang et al., 2025), with the original labels remapped to 10 occupied classes and free space. Both are trained on Waymo data and evaluated on all 6,049 valid forecasting windows. EvoWorld uses 12,800 Gaussians and 20 training epochs, while SparseWorld uses its 64-epoch schedule. These settings underlie the Waymo comparisons in Tables 1 and 2.

We train EvoWorld end-to-end with AdamW (Loshchilov & Hutter, 2019) for 20 epochs on two H100 GPUs, with one sample per GPU. The initial learning rate is 2×10^{-4} with cosine decay, a backbone multiplier of 0.1, weight decay of 10^{-2}, and gradient clipping at 5. We weight semantic cross-entropy by 6 and both Lovasz and planning losses by 1. Appendix C.1 examines alternative occupancy-loss weights.

C Additional Experimental Analysis

C.1 Hyperparameter Sensitivity

To complement the design ablations in the main text, we examine the sensitivity of EvoWorld to supervision hyperparameters. Table 6 compares four configurations with different CE and free-class weights. The setting with CE weight 6 and free-class weight 0.3 achieves the best performance, reaching 23.82 IoU and 13.60 mIoU. Both smaller and larger weights lead to lower performance, indicating that moderate weighting provides the best balance for occupancy supervision and supporting our selected configuration.

Table 6: Sensitivity to supervision hyperparameters. Both CE and free-class weights are varied across configurations.

CE weight	Free-class weight	IoU	mIoU
5	0.2	23.02	13.28
6	0.3	**23.82**	**13.60**
7	0.4	22.40	13.48
10	0.5	22.26	13.30

C.2 Primitive Count

Table 7 studies the effect of the number of Gaussian primitives. Reducing the number from $25{,}600$ to $12{,}800$ preserves similar forecasting performance, with slightly higher IoU and mIoU, while

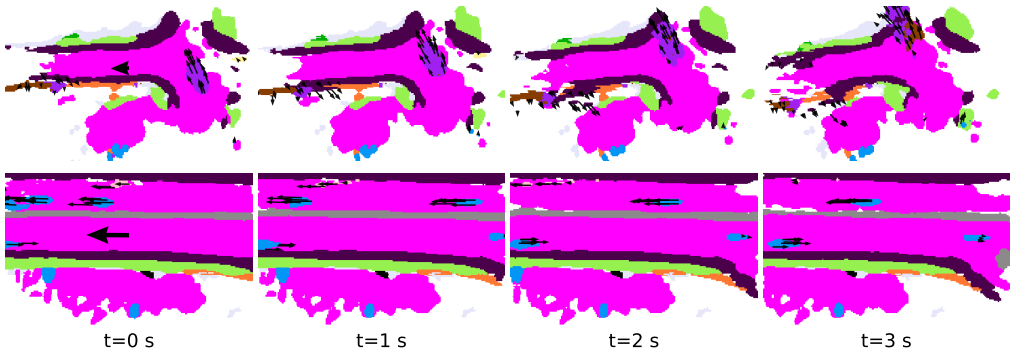


Figure 6: Derived voxel flow for two scenes. Colors show predicted semantic occupancy; small arrows show object motion above 2 m/s, with at most one arrow per 3.2 m block. The large arrow at $t = 0$ shows scene motion.

planning performance changes only marginally. However, further reducing the representation to 6,400 Gaussians leads to a clear drop in occupancy forecasting accuracy and higher planning error, indicating insufficient scene coverage with too few primitives.

Table 7: **Gaussian-count ablation on nuScenes.** Metrics are averaged across three horizons.

Gaussians	IoU ↑	mIoU ↑	L2 (m) ↓	Collision (%) ↓
6,400	20.91	12.12	0.29	0.24
12,800	**23.99**	**13.62**	0.28	0.18
25,600	23.82	13.60	**0.27**	**0.16**

C.3 Object Speed and Voxel Flow

The displacement errors in Table 3 are complemented by the speed distributions in Table 8. As in the main evaluation, each dynamic object receives the median planar velocity of class-matched primitives inside its initial GT box. For the speed comparison, we retain objects with at least one assigned primitive from both EvoWorld and the full 4D variant, yielding 103,630 object–future-time comparisons across forecast horizons. The mean and upper speed percentiles from EvoWorld are closer to GT, consistent with its lower displacement error.

Table 8: Dynamic-object planar-speed distributions in m s^{-1}.

Planar speed (m s^{-1})	Mean	P50	P90	P95	P99
GT	4.961	4.769	9.868	11.072	17.764
EvoWorld	4.637	4.256	9.484	10.479	16.230
Full 4D	1.882	0.619	6.268	7.736	9.962

Derived voxel flow. Let $w_q(\mathbf{x}, t)$ be the occupancy contribution of primitive q, including its density and effective opacity. The resulting planar voxel-flow readout is

$$\mathbf{F}(\mathbf{x}, t) = \frac{\sum_q w_q(\mathbf{x}, t)\mathbf{v}_q}{\sum_q w_q(\mathbf{x}, t) + \epsilon}. \tag{22}$$

Here $\epsilon > 0$ stabilizes the denominator in regions with little Gaussian support. This readout converts primitive velocities into an occupancy-weighted flow field at any queried timestamp. For the visualization, we apply the same weighted readout separately to the scene and object terms of Equation 9.

Figure 6 visualizes the flow derived from EvoWorld's Gaussian primitives at $t = 0, 1, 2, 3$ s. Each row shows one scene: the large arrow at $t = 0$ indicates scene motion, while the smaller arrows show

object motion above $2\,\mathrm{m/s}$. The changing occupancy and flow remain spatially aligned across the queried times. These examples show how EvoWorld's explicit Gaussian state provides both future semantic occupancy and a motion field that can be queried directly.

C.4 Per-Class Metrics

Table 9 compares the 12,800-primitive variant with SparseWorld. At 3 seconds, improvements on bicycles, motorcycles, pedestrians, and trailers coexist with weaker performance on buses, cars, construction vehicles, and trucks. Overall mIoU at this horizon is 11.52 versus 11.51.

Table 9: Per-class semantic IoU at 1 and 3 seconds on nuScenes. Dynamic class names are bold. EvoWorld uses 12,800 primitives.

Class	SparseWorld 1 s	EvoWorld 1 s	SparseWorld 3 s	EvoWorld 3 s
Others	3.99	**4.71**	3.61	**4.17**
Barrier	18.47	**18.50**	**15.02**	14.21
Bicycle	6.57	**8.36**	3.89	**4.42**
Bus	**21.96**	21.05	**12.20**	11.01
Car	**21.66**	20.82	**15.13**	13.72
Construction vehicle	12.56	**12.99**	**10.99**	9.63
Motorcycle	8.31	**9.03**	4.56	**5.79**
Pedestrian	7.98	**8.14**	3.21	**4.47**
Traffic cone	6.52	**7.16**	3.95	**4.75**
Trailer	**9.91**	8.85	7.06	**7.15**
Truck	**18.58**	17.59	**12.64**	11.53
Driveable surface	30.76	**37.31**	27.47	**31.03**
Other flat	20.09	**21.45**	17.21	**17.44**
Sidewalk	19.04	**21.66**	16.27	**17.56**
Terrain	18.12	**19.50**	**15.66**	15.33
Manmade	**12.15**	11.95	**11.03**	9.47
Vegetation	**17.60**	17.45	**16.06**	14.19
mIoU	14.93	**15.68**	11.51	**11.52**

C.5 Inclined Scenes and Vertical Motion

To examine the planar-motion assumption, we partition the 4,219 nuScenes validation samples by ego-pose inclination, used as a proxy for road grade. Samples whose forward axis reaches at least 2° inclination within 0–3 seconds form the inclined group; this threshold corresponds to approximately 3.5% grade. Table 10 compares the resulting 568 inclined samples with the 3,651 flatter samples.

Table 10: Forecasting on inclined and flatter nuScenes samples.

Group	Samples	mIoU ↑				IoU ↑			
		1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Inclined	568	13.89	11.53	9.08	11.50	21.15	19.57	17.07	19.26
Flatter	3,651	15.94	13.90	11.79	13.88	26.35	24.83	22.73	24.64
All	4,219	15.71	13.62	11.46	13.60	25.56	24.03	21.86	23.82

Inclined scenes are harder already at 0 s, but their mIoU drops from 15.54 to 9.08 over 3 s (6.46 points), compared with 17.45 to 11.79 for flatter scenes (5.66 points). Thus, only 0.80 points of additional degradation accumulate during forecasting, supporting the robustness of the planar-motion prior on sloped roads.

We further compare EvoWorld with no vertical velocity and bounded vertical velocity using 12,800 primitives. Table 11 shows that the planar-motion prior outperforms adding bounded vertical velocity on the validation set. One possible explanation is that v_z introduces an additional degree of freedom whose learning depends on the chosen bound, while driving motion is predominantly planar.

Constraining center motion to the ground plane does not make the representation two-dimensional: each Gaussian retains a learned position and vertical spatial extent, which can cover moderate height variation without moving its center vertically.

Table 11: Comparison of planar and bounded vertical motion.

v_z setting	IoU	mIoU
Planar, $v_z = 0$	23.99	13.62
Bounded vertical motion, $\lvert v_z \rvert \leq 0.2\,\mathrm{m/s}$	23.44	13.32

Allowing unbounded v_z destabilized training and made Gaussian centers more likely to leave the scene bounds. We therefore use the better-performing planar prior and leave robust vertical-motion modeling to future work.

D Temporal Query Scaling and Parallel Decoding

Table 12 extends the efficiency comparison in Table 5 by varying the number of temporal queries. We measure the complete forward pass on one H100 with FP32 and batch size 1, using 100 consecutive validation samples after 10 warm-ups. Timings exclude data loading and metric accumulation. Here K counts additional timestamps beyond the current-frame output, giving $K + 1$ decoded volumes; $K = 6$ matches the seven-output workload in the main table. Reusing the Gaussian state limits the additional cost of denser temporal queries. With 12,800 primitives, increasing from one to seven outputs raises latency from 115.80 to 154.02 ms, adding 38.22 ms for six queries; 31 outputs take 307.17 ms in total. This computation depends on the number of requested outputs: each timestamp, including a direct 3-second query, uses the same conditioning and splatting operations.

Table 12: Timestamp-count scaling. K denotes additional timestamps beyond current frame.

	25,600 Gaussians		12,800 Gaussians	
Future timestamps K	Latency (ms)	Throughput (Hz)	Latency (ms)	Throughput (Hz)
0	177.23	5.64	115.80	8.64
3	202.94	4.93	134.88	7.41
6	227.48	4.40	154.02	6.49
12	278.48	3.59	192.52	5.19
24	378.02	2.65	269.25	3.71
30	427.71	2.34	307.17	3.26

Two-GPU timestamp parallelism. Each timestamp is decoded directly from the shared Gaussian state, allowing conditioning and splatting to be distributed across two H100 GPUs. Tables 13 and 14 report complete forward-pass latency and the isolated decoding stage, respectively. With 12,800 primitives, the complete forward-pass speedup increases from $1.13\times$ for seven outputs to $1.40\times$ for 31 outputs; conditioning and splatting reach $1.78\times$ for 31 outputs. Denser temporal sampling increases the parallelizable decoding workload while reusing the same predicted world.

Table 13: Two-H100 full-forward timestamp parallelism.

Model	K	2-GPU latency	2-GPU throughput	Speedup
EvoWorld-12,800	6	136.82 ms	**7.31 Hz**	1.13x
EvoWorld-12,800	30	220.17 ms	4.54 Hz	1.40x
EvoWorld-25,600	6	204.35 ms	4.89 Hz	1.11x
EvoWorld-25,600	30	309.05 ms	3.24 Hz	1.38x

Table 14: Two-H100 time conditioning and Gaussian-to-voxel splatting. These timings exclude the remaining forward-pass stages.

Model	K	1-GPU splatting	2-GPU splatting	Speedup
EvoWorld-12,800	6	45.61 ms	28.41 ms	1.61x
EvoWorld-12,800	30	198.73 ms	111.73 ms	1.78x
EvoWorld-25,600	6	58.77 ms	35.65 ms	1.65x
EvoWorld-25,600	30	258.60 ms	139.94 ms	1.85x

E Extended Related Work

3D occupancy prediction. 3D occupancy prediction (Huang et al., 2023) has become an important scene representation for autonomous driving because it provides dense geometric and semantic understanding beyond object-level detection. In the vision-centric setting, a large body of work predicts semantic occupancy from multi-view cameras using BEV-, plane-, or volume-based representations (Wei et al., 2023; Huang et al., 2023; Zhang et al., 2023; Ye et al., 2024; Liao et al., 2025a; Chen et al., 2025). Along a different direction, Gaussian-based methods replace dense discretization with sparse scene primitives. The GaussianFormer series (Huang et al., 2024; 2025) shows that semantic Gaussians can serve as compact occupancy primitives. GaussianWorld (Zuo et al., 2025) extends this idea to streaming scene modeling, while GaussianOcc (Gan et al., 2025) and GaussianFlowOcc (Boeder et al., 2025) show that Gaussian representations naturally support rendering- and projection-based occupancy learning. Recent work further extends Gaussian occupancy representations to collaborative perception (Chen et al., 2026). EvoWorld uses semantic Gaussian primitives to learn a structured 4D Gaussian world representation for continuous future scene evolution.

Gaussian splatting for dynamic scene representation. Gaussian splatting has become a strong explicit scene representation for real-time novel view synthesis, starting from 3D Gaussian Splatting (Kerbl et al., 2023) and extending to dynamic settings through time-varying positions, deformation fields, or 4D Gaussian parameterizations (Luiten et al., 2024; Yang et al., 2024; Duan et al., 2024). EvoGS (Asiimwe & Vondrick, 2025) further learns a continuous-time dynamical field for temporal extrapolation. These methods target photorealistic reconstruction and rendering across viewpoints and times. Recent urban-scene extensions such as Street Gaussians (Yan et al., 2024) improve reconstruction quality in driving scenes, and focus on rendering observed scenes. EvoWorld uses explicit, continuous Gaussian primitives to forecast future semantic occupancy from camera history. It predicts one set of 4D semantic Gaussian primitives for the forecast horizon, parameterizing their spatial geometry and temporal support directly while decomposing planar velocity into shared scene and object-level motion.

Occupancy world models for autonomous driving. Occupancy world models extend 3D occupancy prediction to future scene evolution through two main approaches: predicting future occupancy from continuous latent features or sparse spatial queries (Yang et al., 2025; Li et al., 2025; Dang et al., 2026; Gu et al., 2024; Shi et al., 2025); or encoding occupancy into discrete latent tokens and predicting future tokens before decoding them back to occupancy volumes (Zheng et al., 2024a; Liao et al., 2025b; Wei et al., 2024; Xu et al., 2025; Yan et al., 2025). These methods commonly model future dynamics in latent spaces. Autoregressive variants predict future states at fixed steps, requiring intermediate state generation to reach a target horizon. GWM introduces Gaussian priors into occupancy forecasting and predicts future scenes in a tokenized state space (Feng et al., 2025). SparseWorld-TC (Du et al., 2026) studies trajectory-conditioned occupancy forecasting and variants without trajectory conditioning. EvoWorld represents driving scenes with structured 4D semantic Gaussian primitives. Querying and splatting their conditional 3D Gaussians directly produces occupancy forecasts at arbitrary timestamps.

GaussianAD (Zheng et al., 2024b) predicts a separate displacement for each Gaussian at each fixed future timestamp and transforms the resulting states using the planned ego trajectory. DLWM (Zhu et al., 2026) pretrains Gaussian-flow and ego-planning latent models separately, then forecasts occupancy through autoregressive Gaussian propagation and refinement. EvoWorld predicts one set of 4D Gaussian primitives to represent the entire horizon, which provides direct timestamp queries and an anchored motion planning readout before occupancy splatting.

Continuous occupancy fields and forecasting. UnO (Agro et al., 2024) and DIO (Diehl et al., 2025) demonstrate the benefits of continuous implicit space–time fields, including semantic transfer and decomposable occupancy-flow. QueryOcc (Lilja et al., 2026) learns a time-queryable decoder through adjacent-frame 4D supervision and evaluates current-frame semantic occupancy. GASP (Ljungbergh et al., 2026) uses a LiDAR encoder with geometric, semantic-feature, and ego-occupancy supervision, then evaluates downstream forecasting and trajectory prediction. These methods establish continuous querying and unified representation learning. EvoWorld represents the forecast horizon with one explicit set of 4D semantic Gaussian primitives, exposing their geometry and temporal support while separating shared scene motion from object-level motion.

F Limitations and Future Work

First, EvoWorld represents scene evolution with a finite set of primitives, each assuming constant planar velocity and a time-invariant conditional shape. These assumptions limit its ability to model acceleration, vertical center motion, and shape deformation. Second, our initialization study compares random camera-only initialization and LiDAR-based initialization; depth-guided camera placement could improve coverage. Third, semantic predictions are restricted to the fixed training vocabulary, motivating language grounding for open-world forecasting. Finally, our planning evaluation measures open-loop trajectory error and collision rates. Closed-loop evaluation would further assess how the Gaussian world supports decisions during interaction.