SPATIOTEMPORALLY CONSISTENT SPARSE WORLD MODEL FOR 4D OCCUPANCY PREDICTION AND PLANNING IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

4D semantic occupancy forecasting provides a scene-level representation for anticipating future geometry and semantics in autonomous driving. Sparse occupancy world models enable efficient long-horizon forecasting by compressing a scene into compact latent queries and rolling them forward in time. However, existing sparse world models suffer from **spatiotemporally inconsistent semantic rollouts**. This problem arises because semantic information evolved in latent query space is not explicitly aligned with the rendered 4D occupancy structure formed by refined points and their temporal evolution. To address this limitation, we present **STC-World**, a spatiotemporally consistent sparse world model that explicitly aligns latent semantics with the rendered 4D occupancy structure. For spatial semantic alignment, we propose a **coordinate-conditioned semantic decoder** that incorporates normalized point coordinates and query-relative offsets into semantic decoding, producing location-specific corrections to point-level logits. For temporal semantic consistency, we introduce **adjacent-horizon temporal self-distillation** to constrain semantic evolution during future rollout. The distillation objective uses corrected semantics at each horizon as a stop-gradient teacher for the next horizon over inherited query indices, with class-aware consistency terms for static and dynamic categories. STC-World therefore maintains semantic correspondence throughout spatial refinement and temporal rollout while retaining a sparse latent representation. On Occ3D-nuScenes, STC-World achieves a state-of-the-art average mIoU of 13.47% over the 1–3 s forecasting horizons. It attains a 0.20 m average planning L2 error and reduces the average collision rate from 0.29% to 0.09%, a 69.0% relative reduction, while running at 2.56 FPS with 391.2 ms mean latency. Project page: `https://stc-world.github.io/`.

1 INTRODUCTION

Safe autonomous driving depends on anticipating how the surrounding scene will evolve before a vehicle commits to its future motion. Semantic occupancy represents both scene geometry and category at the voxel level, including irregular structures that are not well summarized by bounding boxes (Huang et al., 2023; Wei et al., 2023; Tian et al., 2023; Chen et al., 2026). Extending this representation through time yields 4D occupancy forecasting, which predicts static layout, dynamic actors, and their interactions at future horizons (Ma et al., 2024a; Liao et al., 2025). Compared with object-centric prediction alone, future occupancy provides a scene-level interface for collision checking, drivable-space reasoning, and downstream planning.

Recent work has progressed from reconstructing the current 3D scene to learning occupancy world models that roll latent scene states into the future (Zheng et al., 2024; Li et al., 2025; Yang et al., 2025; Dang et al., 2026). Latent rollout provides a compact mechanism for summarizing history, forecasting future states, and supporting planning without explicitly tracking every object. Sparse query models further reduce computation by assigning a compact latent state to a set of predicted spatial supports (Tang et al., 2024; Dang et al., 2026). However, this abstraction also exposes a central limitation. **Semantic rollouts can become spatiotemporally inconsistent because semantics evolved in latent query space are not explicitly aligned with the rendered 4D occupancy structure.** A query-level semantic state may no longer match the refined points that define its spatial

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.

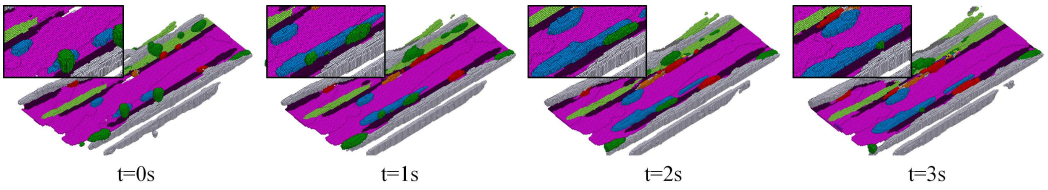

Figure 1: Representative spatiotemporal semantic inconsistency in an occupancy-world-model rollout, visualized with PreWorld predictions (Li et al., 2025). From left to right, the forecasts advance over successive horizons. The global region retains a plausible global layout, while local semantics detach from refined geometry and drift over time.

support, particularly around object boundaries and mixed local regions. As the same sparse state is rolled into future horizons, independently supervised predictions can drift along its query lineage. Figure 1 visualizes this failure pattern. The global road layout remains plausible, yet local vehicle and surface semantics detach from the refined geometry and vary inconsistently over time. This correspondence is especially important in sparse world models because one compact semantic state controls multiple refined points across successive horizons.

To address this limitation, we propose **STC-World**, a spatiotemporally consistent sparse world model for 4D occupancy prediction and planning. STC-World retains the efficiency of sparse query-based world modeling, while making semantic prediction explicitly aware of where and when each sparse point is evaluated. For spatial semantic alignment, we introduce a coordinate-conditioned semantic decoder that refines each point-level logit using the query feature, the normalized refined 3D coordinate, and the point offset from the center of its query support. The decoder predicts a residual correction to the base semantic logits, so semantic refinement is tied to the rendered point location without altering the recurrent sparse rollout state. For temporal semantic alignment, we introduce adjacent-horizon self-distillation over corrected point logits. The corrected semantics at one forecast horizon serve as a stop-gradient teacher for the next horizon over inherited query and point-slot indices, with class-aware consistency terms for static and dynamic categories. This index continuity is used as a local rollout prior rather than as physical object correspondence.

We evaluate STC-World on camera-based 4D occupancy forecasting and downstream planning on Occ3D-nuScenes. STC-World achieves a camera-based state-of-the-art 13.47% average mIoU. It also reaches 0.20 m average L2 error in planning and reduces the average collision rate from the second-best 0.29% to 0.09%, a relative reduction of 69.0%. The model retains the computational profile, running at 2.56 FPS with 391.2 ms mean latency. Our contributions are as follows:

- We identify spatiotemporally inconsistent semantic rollout as a central limitation of sparse occupancy world models and propose **STC-World**, which aligns latent semantics with the refined spatial support and temporal evolution of rendered 4D occupancy.
- We introduce **coordinate-conditioned semantic alignment**, which combines query features with refined coordinates and query-relative offsets to predict point-specific logit residuals, coupling semantic decoding with rendered occupancy geometry.
- We develop **adjacent-horizon temporal self-distillation**, which regularizes corrected semantics over inherited query indices and applies class-aware objectives to static and dynamic categories without adding inference-time computation.
- Extensive experiments on **Occ3D-nuScenes** show that STC-World achieves a camera-based state-of-the-art 13.47% average mIoU under the reported protocol and reduces the average planning collision rate by 69.0% relative to the second-best result.

2 Related Work

2.1 Camera-Based Semantic Occupancy Prediction

Camera-based semantic occupancy prediction extends image perception from visible surfaces and object boxes to a complete 3D field. Early camera-only methods lift 2D evidence into dense vol-

umetric representations. MonoScene connects 2D and 3D networks through line-of-sight feature projection, TPVFormer represents the scene on three orthogonal planes, and OccFormer processes dense volumes through local and global transformer paths (Cao & de Charette, 2022; Huang et al., 2023; Zhang et al., 2023). VoxFormer initializes sparse visible and occupied voxel queries before completing a dense scene, while SurroundOcc aggregates multi-camera features into a multi-scale 3D volume (Li et al., 2023; Wei et al., 2023). Occ3D provides a large-scale benchmark and visibility-aware protocol for autonomous-driving occupancy (Tian et al., 2023).

Subsequent work has reduced either representation cost or dependence on dense 3D supervision. RenderOcc trains occupancy fields with semantic and depth rendering losses from 2D observations, and SelfOcc learns 3D occupancy from video through differentiable rendering (Pan et al., 2024; Huang et al., 2024a). COTR compresses dense occupancy into a compact latent representation, whereas SparseOcc and OSP formulate occupancy with sparse latent instances or points of interest (Ma et al., 2024b; Tang et al., 2024; Shi et al., 2024). GaussianFormer and GaussianFormer-2 replace regular voxel features with adaptive semantic Gaussians, providing another sparse representation of occupancy geometry (Huang et al., 2024b; 2025). These methods reconstruct current scenes. STC-World instead studies sparse semantic attachment over future horizons.

2.2 4D Occupancy Forecasting and World Models

4D occupancy forecasting extends semantic occupancy from a single scene state to a sequence of future states. Cam4DOcc establishes a camera-only benchmark and predicts future occupancy together with scene flow, while OccProphet improves forecasting efficiency through a lightweight observer–forecaster–refiner framework (Ma et al., 2024a; Chen et al., 2025). OccWorld tokenizes 3D occupancy and applies a spatial-temporal generative transformer to predict future scene and ego tokens (Zheng et al., 2024). RenderWorld derives self-supervised 3D labels through Gaussian rendering, while DFIT-OccWorld separates static pose transformation from dynamic voxel flow and adds image-assisted training (Yan et al., 2025; Zhang et al., 2024). GaussianWorld explicitly evolves a sparse Gaussian scene representation for streaming occupancy prediction (Zuo et al., 2025). Drive-OccWorld conditions a vision-centric world model on candidate actions and connects future occupancy to trajectory selection, while PreWorld uses rendering-based pre-training to exploit additional 2D supervision (Yang et al., 2025; Li et al., 2025). In parallel, planning-oriented systems such as UniAD and VAD integrate perception, prediction, and planning through unified query or vectorized scene representations (Hu et al., 2023; Jiang et al., 2023).

SparseWorld introduces dynamic sparse queries for perception, future occupancy rollout, and planning (Dang et al., 2026). Its efficiency makes sparse query states an attractive substrate for a world model, but query-space evolution does not by itself enforce correspondence between semantic logits and the refined points used for rendering. STC-World addresses this representational gap. It couples point semantics to refined coordinates and regularizes their evolution over inherited query indices, without replacing the sparse rollout or densifying its latent state.

2.3 Knowledge Distillation and Temporal Consistency

Knowledge distillation transfers predictive distributions from a teacher to a student, commonly to improve a compact model (Hinton et al., 2015). Self-distillation removes the need for a separate teacher by transferring information between generations or internal stages of the same model (Furlanello et al., 2018; Zhang et al., 2019). These methods usually relate different networks, training generations, or depths that predict the same target.

Our use of self-distillation has a different temporal role. The teacher and student are point-level semantic predictions at adjacent forecast horizons within one sparse rollout. Pairing follows inherited query and point-slot indices and is filtered by a validity mask, rather than inferred as physical object correspondence. Static categories are matched as distributions, whereas dynamic categories use a weaker logit constraint, with no added inference-time module.

Forecasting also changes the distillation target. Adjacent horizons need not share identical geometry, so temporal regularization should preserve semantics only over valid inherited indices. This links self-distillation to sparse state transitions rather than to a separate teacher network.

3 METHODOLOGY

3.1 PRELIMINARIES

3.1.1 4D SEMANTIC OCCUPANCY FORECASTING

Let $\mathcal{I}_{-H+1:0}$ denote synchronized images from N_c cameras over H observed frames, and let $\mathcal{E}_{-H+1:0}$ contain the corresponding ego poses and states. We use $\tau \in \{0, \dots, T\}$ for prediction horizon, with $\tau = 0$ denoting the current frame and $t_\tau = t_0 + \tau\Delta t$. The semantic occupancy target $\mathbf{Y}_\tau \in \{1, \dots, C, \emptyset\}^{X \times Y \times Z}$ is expressed in the current-frame LiDAR coordinate system. The forecasting model estimates the following conditional distribution

$$p_\theta(\mathbf{Y}_{0:T} \mid \mathcal{I}_{-H+1:0}, \mathcal{E}_{-H+1:0}),$$

and produces $\hat{\mathbf{Y}}_{0:T}$ for occupancy evaluation and downstream planning. The task requires semantic labels to remain attached to their 3D support while that support evolves across future horizons.

3.1.2 SPARSE WORLD MODELS

Sparse world models replace dense latent voxel features with a compact state

$$\mathcal{Z}_\tau = \{(\mathbf{q}_{\tau,i}, \mathbf{P}_{\tau,i}, \mathbf{S}_{\tau,i})\}_{i=1}^{N_q}, \qquad \mathbf{P}_{\tau,i} = \{\mathbf{p}_{\tau,i,j}\}_{j=1}^{R},$$

where $\mathbf{q}_{\tau,i}$ is a query feature, $\mathbf{p}_{\tau,i,j} \in [0,1]^3$ is a refined point normalized within the current LiDAR range, and $\mathbf{S}_{\tau,i}$ contains its semantic logits. A shared query feature summarizes the visual and temporal evidence for several refined points. This compression commonly induces the approximation

$$p_\theta(c_{\tau,i,j} \mid \mathbf{q}_{\tau,i}, \mathbf{p}_{\tau,i,j}) \approx p_\theta(c_{\tau,i,j} \mid \mathbf{q}_{\tau,i}).$$

The refined coordinate then affects semantics only indirectly through the query that generated it, which can be insufficient when one query spans multiple local semantic patterns.

3.1.3 SPATIOTEMPORAL SEMANTIC INCONSISTENCY IN LATENT ROLLOUT

Let $C_\tau(\mathbf{x})$ denote the semantic class supported at location $\mathbf{x}$ and horizon τ. For a persistent scene element with local motion map $T_{\tau\to\tau+1}$, semantic anchoring requires

$$C_{\tau+1}(T_{\tau\to\tau+1}(\mathbf{x})) = C_\tau(\mathbf{x})$$

whenever the support remains valid and its semantic identity does not change. Sparse rollout can violate this condition spatially because rendering evaluates logits at refined points that the semantic decoder does not explicitly observe. It can also violate the condition temporally because independent occupancy supervision at each horizon does not constrain how semantics evolve along an inherited latent index. The resulting inconsistency concerns the attachment between latent semantics and rendered 4D support, rather than object tracking. Appendix A.1.1 gives the full derivation and states the scope of the internal lineage assumption.

3.2 OVERVIEW OF THE PROPOSED FRAMEWORK

Figure 2 summarizes the proposed spatiotemporally consistent sparse world model. Historical camera views are geometrically aligned using the observed ego poses $\mathcal{E}_{-H+1:0}$ and encoded into the current sparse state $\mathcal{Z}_0$. The transition model then conditions on the observable current ego state $\mathbf{e}_0$ to produce future states $\mathcal{Z}_{1:T}$, whose normalized point coordinates share the current-frame LiDAR reference used by the occupancy targets. Corrected point semantics are rendered into 4D semantic occupancy, and the planning branch consumes the predicted scene representation. Future occupancy $\mathbf{Y}_{1:T}$ and ego trajectory $\mathbf{u}_{1:T}$ are training targets rather than rollout inputs.

The design targets the two sources of semantic inconsistency discussed in the preliminaries. Spatially, the semantic label attached to a refined point should depend on the point location that will be rendered into the final occupancy grid. Temporally, predictions associated with an inherited query and point slot should not change arbitrarily between adjacent horizons. We therefore add two alignment mechanisms to the sparse rollout. The coordinate-conditioned semantic decoder maps the query feature and refined coordinates to a residual $\Delta\mathbf{S}$, and the corrected semantics $\hat{\mathbf{S}}$ enter rendering and temporal self-distillation. Adjacent-horizon self-distillation uses $\mathrm{sg}(\hat{\mathbf{S}}_\tau)$ as the teacher for $\hat{\mathbf{S}}_{\tau+1}$ under a validity mask, providing an internal rollout prior rather than physical object tracking.

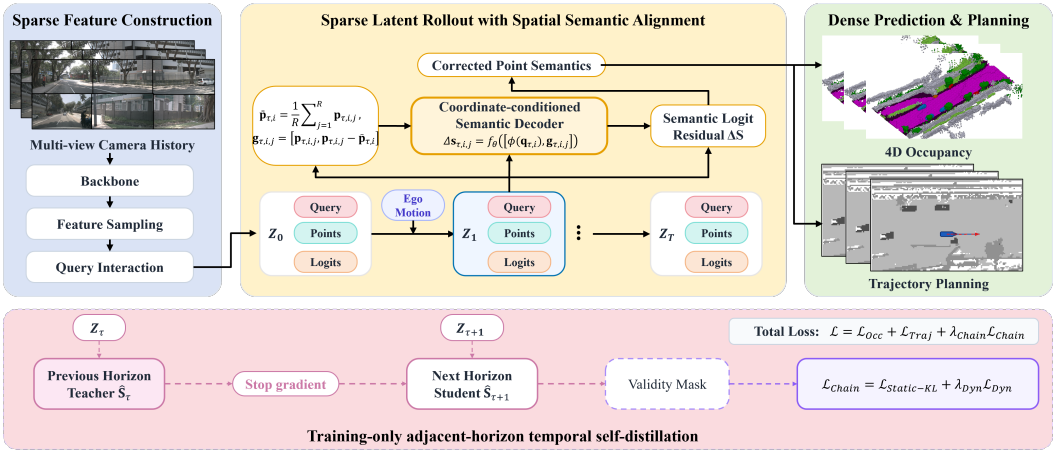

Figure 2: Overall architecture. Multi-view camera history and the current observable ego state are encoded into sparse world states $\mathcal{Z}_{0:T}$, where each state contains query features, refined points, and semantic logits. The coordinate-conditioned semantic decoder predicts a residual $\Delta\mathbf{S}$ from query features and point-level coordinate descriptors, producing corrected point semantics $\hat{\mathbf{S}}$ for dense 4D occupancy prediction. During training, adjacent-horizon self-distillation uses $\mathrm{sg}(\hat{\mathbf{S}}_\tau)$ as the teacher for $\hat{\mathbf{S}}_{\tau+1}$ over inherited query indices valid at both horizons, for $\tau = 1, \ldots, T-1$. The current-to-first-forecast transition is not distilled, and the internal lineage is not an object-level correspondence.

3.3 Sparse 4D World Representation and Future Rollout

At horizon τ, we write the sparse world state as

$$\mathcal{Z}_\tau = \{(\mathbf{q}_{\tau,i}, \mathbf{P}_{\tau,i}, \mathbf{S}_{\tau,i})\}_{i=1}^{N_q}, \qquad \mathbf{P}_{\tau,i} = \{\mathbf{p}_{\tau,i,j}\}_{j=1}^{R}.$$

Multi-view feature sampling, query interaction, and point refinement construct $\mathcal{Z}_0$. The future states are produced by an ego-conditioned transition,

$$\mathcal{Z}_{\tau+1} = G_\theta(\mathcal{Z}_\tau, \mathbf{e}_0, \tau), \qquad \tau = 0, \ldots, T-1,$$

where $\mathbf{e}_0$ is the current observable ego state and τ selects the horizon embedding. Historical ego poses align the camera history before $\mathcal{Z}_0$ is constructed; future ground-truth ego poses and trajectories are not rollout inputs. After semantic correction, a deterministic renderer voxelizes the refined points and aggregates their corrected logits,

$$\hat{\mathbf{Y}}_\tau = \mathcal{R}(\mathbf{P}_\tau, \hat{\mathbf{S}}_\tau).$$

The renderer applies class-specific confidence filtering, resolves multiple point contributions by class-wise maximum aggregation, and marks unassigned voxels as empty. Appendix A.1.2 specifies point refinement and rollout, while Appendix A.1.3 gives the complete thresholding, conflict-resolution, and morphological post-processing procedure.

3.4 Coordinate-conditioned Semantic Alignment

Point-level coordinate conditioning. The sparse state separates a query feature from the refined points generated by that query. This separation is efficient, but it creates a possible mismatch. The query feature $\mathbf{q}_{\tau,i}$ summarizes a local scene component, whereas the final occupancy label is assigned at the refined point $\mathbf{p}_{\tau,i,j}$ and later at its receiving voxel. We encode each refined point with the following descriptor:

$$\bar{\mathbf{p}}_{\tau,i} = \frac{1}{R}\sum_{j=1}^{R} \mathbf{p}_{\tau,i,j}, \qquad \mathbf{g}_{\tau,i,j} = \left[\mathbf{p}_{\tau,i,j}, \mathbf{p}_{\tau,i,j} - \bar{\mathbf{p}}_{\tau,i}\right].$$

This equation matches the coordinate box in Figure 2. The first term gives the normalized location of the point in the global occupancy range. The second term gives its local offset inside the query

support. Together, they describe where the semantic logit will be rendered and where the point lies relative to its sibling points.

Residual semantic logit correction. The coordinate-conditioned semantic decoder receives the coordinate descriptor and a projected query feature,

$$\mathbf{z}_{\tau,i} = \phi(\mathbf{q}_{\tau,i}).$$

For each point, it predicts a class-logit residual,

$$\Delta\mathbf{s}_{\tau,i,j} = f_\theta\left([\phi(\mathbf{q}_{\tau,i}), \mathbf{g}_{\tau,i,j}]\right).$$

The corrected point semantics are

$$\hat{\mathbf{s}}_{\tau,i,j} = \mathbf{s}_{\tau,i,j} + \Delta\mathbf{s}_{\tau,i,j}, \qquad \hat{\mathbf{S}}_\tau = \{\hat{\mathbf{s}}_{\tau,i,j}\}_{i,j}.$$

A zero-initialized final layer preserves base logits and learns coordinate corrections.

Integration into current and future predictions. The decoder is applied after point refinement for the current state and for each future state in the rollout,

$$\hat{\mathbf{S}}_\tau = \mathbf{S}_\tau + \Delta\mathbf{S}_\tau, \qquad \tau = 0, \ldots, T.$$

The corrected logits $\hat{\mathbf{S}}_\tau$ replace $\mathbf{S}_\tau$ in the occupancy losses, renderer $\mathcal{R}$, and temporal self-distillation. This adds coordinate awareness without changing recurrent query dynamics.

3.5 Adjacent-horizon Temporal Self-distillation

The rollout preserves inherited query and point-slot indices. For adjacent forecast states, corrected semantics at τ provide a detached teacher for the student at $\tau + 1$:

$$\mathbf{s}^{\text{tea}}_{\tau,i,j} = \text{sg}(\hat{\mathbf{s}}_{\tau,i,j}), \qquad \mathbf{s}^{\text{stu}}_{\tau+1,i,j} = \hat{\mathbf{s}}_{\tau+1,i,j},$$

where $\text{sg}(\cdot)$ stops gradients through the teacher. The loss pairs only inherited indices that are valid at both horizons; newly appended queries are excluded. This lineage is a computational continuity prior, not physical object correspondence.

Static surfaces are matched with a temperature-scaled Bernoulli KL term, whereas actor-like categories use a weaker logit-distance term to accommodate motion and deformation. We denote these terms by $\mathcal{L}^{\text{static-KL}}_\tau$ and $\mathcal{L}^{\text{dyn}}_\tau$, respectively. The chain objective is

$$\mathcal{L}_{\text{chain}} = \frac{1}{T-1}\sum_{\tau=1}^{T-1} \alpha_\tau \left(\mathcal{L}^{\text{static-KL}}_\tau + \lambda_{\text{dyn}}\mathcal{L}^{\text{dyn}}_\tau\right).$$

The sum starts at $\tau = 1$, so distillation excludes $\mathcal{Z}_0 \to \mathcal{Z}_1$ and covers only adjacent forecast states. We use $\alpha_\tau = 0.9^{\tau-1}$, $\lambda_{\text{dyn}} = 0.1$, a temperature annealed from 4.0 to 1.0, and an overall chain-loss weight of 0.75. Complete loss, class, mask, and validity details appear in Appendix A.1.4.

3.6 Training Objective

The total loss shown in Figure 2 groups all occupancy-related sparse supervision into $\mathcal{L}_{\text{occ}}$ and combines it with trajectory and chain-distillation losses,

$$\mathcal{L} = \mathcal{L}_{\text{occ}} + \mathcal{L}_{\text{traj}} + \lambda_{\text{chain}}\mathcal{L}_{\text{chain}}.$$

The occupancy term includes current and future supervision,

$$\mathcal{L}_{\text{occ}} = \mathcal{L}_{\text{cur}} + \mathcal{L}_{\text{future}}, \qquad \mathcal{L}_{\text{future}} = \sum_{\tau=1}^{T}\mathcal{L}^\tau_{\text{occ}}.$$

Each $\mathcal{L}^\tau_{\text{occ}}$ contains the sparse semantic classification and point refinement losses used to train the logits and refined point coordinates before rendering.

The trajectory branch is trained with an L2 loss over future ego states,

$$\mathcal{L}_{\text{traj}} = \sum_{\tau=1}^{T}\|\hat{\mathbf{u}}_\tau - \mathbf{u}_\tau\|_2^2,$$

where $\hat{\mathbf{u}}_\tau$ and $\mathbf{u}_\tau$ are the predicted and ground-truth trajectory states. The future trajectory $\mathbf{u}_{1:T}$ is a training target and is not provided during inference. The coordinate decoder is supervised by $\mathcal{L}_{\text{occ}}$ through rendered $\hat{\mathbf{S}}_\tau$, while $\mathcal{L}_{\text{chain}}$ aligns these logits across adjacent horizons. The objective therefore couples coordinate-conditioned semantics with temporal consistency.

Table 1: 4D occupancy forecasting on Occ3D-nuScenes. All models use camera input; OccLLaMA-F additionally uses action and language. Baseline numbers are taken from the corresponding publications, while STC-World is evaluated with our implementation. The best result in each column is shown in **bold**, and the second-best result is underlined.

Method	Aux. Sup.	mIoU ↑				IoU ↑			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
OccWorld-S	None	0.28	0.26	0.24	0.26	5.05	5.01	4.95	5.00
RenderWorld	None	2.83	2.55	2.37	2.58	14.61	13.61	12.98	13.73
OccWorld-T	Semantic LiDAR	4.68	3.36	2.63	3.56	9.32	8.23	7.47	8.34
OccWorld-D	3D Occ	11.55	8.10	6.22	8.62	18.90	16.26	14.43	16.53
OccLLaMA-F	3D Occ, Language	10.34	8.66	6.98	8.66	**25.81**	**23.19**	19.97	**22.99**
PreWorld	3D Occ	11.69	8.72	6.77	9.06	23.01	20.79	18.84	20.88
PreWorld	2D & 3D Occ	12.27	9.24	7.15	9.55	23.62	21.76	19.63	21.62
DFIT-OccWorld	3D Occ	13.38	10.16	7.96	10.50	19.18	16.85	15.02	17.02
SparseWorld	3D Occ	14.93	13.15	11.51	13.20	22.96	22.10	21.05	22.03
STC-World (Ours)	3D Occ	**15.20**	**13.44**	**11.77**	**13.47**	23.11	22.23	**21.15**	22.16

4 Experiments

4.1 Experimental Details

Datasets. We conduct all experiments on Occ3D-nuScenes (Tian et al., 2023), which augments the nuScenes benchmark (Caesar et al., 2020) with visibility-aware semantic occupancy annotations. Following the official split, we use 700 scenes for training and 150 scenes for validation. Each scene lasts approximately 20 seconds and is sampled every 0.5 seconds. The evaluation region spans $[-40, 40]$ m along the x- and y-axes and $[-1, 5.4]$ m along the z-axis. It is discretized into a $200 \times 200 \times 16$ grid with a voxel size of 0.4 m. The labels contain 17 occupied semantic categories and free space. The model observes the current frame and four preceding frames over two seconds, then predicts six future states across a three-second horizon.

Implementation Details. STC-World uses a ResNet-50 backbone (He et al., 2016), an FPN neck, and six surround-view images at 256×704 resolution. Training proceeds in three stages: 48 epochs for the base sparse world model, eight epochs with adjacent-horizon self-distillation, and 10 epochs of joint optimization after adding the coordinate-conditioned decoder. We use AdamW (Loshchilov & Hutter, 2019) with weight decay 0.01. Initial learning rates are 2×10^{-4} for the new decoder, 1×10^{-5} for existing non-backbone modules, and 1×10^{-6} for ResNet layer4; the chain-loss weight is 0.75. Historical ego poses align the camera history, while rollout receives only the current ego state $\mathbf{e}_0$. Future ego poses and trajectories are supervision and evaluation signals, never inference inputs. Architecture and optimization details appear in Appendices A.1.5 and A.2.1.

Evaluation Metrics. For occupancy forecasting, we report semantic mean intersection-over-union (mIoU) over the 17 occupied classes and class-agnostic occupied-space IoU at 1, 2, and 3 seconds. The average is the arithmetic mean across these three horizons. For motion planning, we report the Euclidean L2 displacement error in meters and the object collision rate in percent at the same horizons, together with their averages. Batch-one efficiency on a single A800 GPU includes throughput, latency, memory, parameters, and FLOPs.

4.2 Main Results

4D Occupancy Forecasting. Table 1 compares STC-World with camera-based occupancy world models under the same validation protocol. STC-World obtains 15.20, 13.44, and 11.77 mIoU at 1, 2, and 3 seconds, respectively, and establishes a camera-based state of the art under the reported comparison protocol with an average mIoU of 13.47. Relative to SparseWorld, the strongest sparse baseline, the gains are 0.27, 0.29, and 0.26 percentage points at the three horizons, yielding a 0.27-point improvement on average. The improvement is maintained at the longest horizon rather than being concentrated near the observed frames. For class-agnostic occupancy, STC-World raises the average IoU from 22.03 to 22.16 and achieves the best 3-second IoU. OccLLaMA-F remains strongest at 1 and 2 seconds and on average, but operates under a richer information regime: it uses a LLaMA-based autoregressive model to jointly process occupancy, language, and action

Table 2: Motion planning on Occ3D-nuScenes. All models are camera-based. Baseline numbers are taken from the corresponding publications, while STC-World is evaluated with our implementation. The best result in each column is shown in **bold**, and the second-best result is underlined.

Method	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
UniAD	Map, Box, Motion, Track, 3D Occ	0.48	0.96	1.65	1.03	**0.05**	0.17	0.71	0.31
VAD	Map& Box& Motion	0.54	1.15	1.98	1.22	0.10	0.24	0.96	0.43
OccNet	Map & Box & 3D Occ	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
OccWorld-D	3D Occ	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87
RenderWorld-V	3D Occ	0.48	1.30	2.67	1.48	0.14	0.55	2.23	0.97
OccLLaMA-F	3D Occ, Language	0.38	1.07	2.15	1.20	0.06	0.39	1.65	0.70
PreWorld	2D & 3D Occ	0.22	0.30	0.40	0.31	0.21	0.66	0.71	0.53
SparseWorld	3D Occ	0.19	0.25	0.36	0.27	0.11	0.29	0.46	0.29
STC-World (Ours)	3D Occ	**0.16**	**0.20**	**0.24**	**0.20**	0.08	**0.09**	**0.10**	**0.09**

Table 3: Batch-one inference efficiency. Best values are bold; ties after rounding are jointly marked.

Method	FPS ↑	Latency (ms) ↓	Allocated (GB) ↓	Reserved (GB) ↓	Params (M) ↓	FLOPs (G) ↓
PreWorld	0.59	1691.5	3.5	4.9	121.7	8331
SparseWorld	2.24	447.2	**2.7**	**3.1**	**76.0**	**421**
STC-World (Ours)	**2.56**	**391.2**	**2.7**	**3.1**	**76.0**	427

tokens (Wei et al., 2024). STC-World uses neither language input nor pretrained language-model knowledge. Its stronger semantic mIoU across all horizons and leading long-horizon IoU therefore demonstrate effective spatiotemporal alignment without external language priors.

Motion Planning. Table 2 reports end-to-end planning performance on Occ3D-nuScenes. STC-World achieves 0.16, 0.20, and 0.24 m L2 errors at 1, 2, and 3 s, reducing SparseWorld's average from 0.27 to 0.20 m. Collision rates fall from 0.11, 0.29, and 0.46% to 0.08, 0.09, and 0.10%, yielding 0.09% on average, 69.0% below 0.29%. UniAD and OccLLaMA-F have lower 1-second rates (0.05%/0.06%) but use richer supervision or priors: UniAD adds map, box, motion, tracking, and occupancy labels, whereas OccLLaMA-F adds language supervision and LLaMA pretraining (Hu et al., 2023; Wei et al., 2024). STC-World uses neither and achieves the best L2 at all horizons, plus the best collision rate at 2 and 3 s and overall. The final-stage joint optimization of planning parameters captures end-to-end interactions among occupancy, trajectory prediction, and planning over all three horizons.

4.3 Comparison Results

Efficiency. Table 3 compares batch-one inference efficiency. Relative to SparseWorld, STC-World increases throughput from 2.24 to 2.56 FPS, lowers latency from 447.2 to 391.2 ms, and adds only 1.4% FLOPs while retaining the same rounded parameter count and peak memory. It is also 4.3× faster than PreWorld with 19.5× fewer FLOPs. Temporal distillation has no inference cost.

4.4 Ablation Studies

Contribution of the two alignment mechanisms. Table 4 reports stage-wise comparisons of the two alignment mechanisms. The baseline obtains 13.18 mIoU; spatial alignment reaches 13.21, temporal alignment reaches 13.24, and their combination reaches 13.47. These rows are not a compute-matched, from-scratch 2×2 design. Appendix A.2.1 specifies their initialization, trainable parameters, budgets, and batch sizes.

Static and dynamic temporal consistency. Table 5 separates the class-aware chain terms. Static-only KL yields 13.18 mIoU, dynamic-logit consistency reaches 13.23, and their combination reaches 13.47 by balancing continuity against horizon-dependent changes. This complementary gain suggests that static and dynamic categories require different temporal regularization strategies, supporting the use of class-aware consistency rather than a uniform constraint.

Direct consistency evidence. Appendix A.2.4 directly evaluates the two forms of alignment targeted by STC-World. Joint alignment reduces pose-aligned semantic drift by 0.107, 0.032, and 0.030 percentage points at 1, 2, and 3 s, respectively, yielding a mean reduction of 0.056 points relative to

Table 4: Stage-wise alignment-component comparison. Training settings are given in Appendix A.2.1.

Spatial align.	Temporal distill.	$\text{mIoU}_{\text{avg}} \uparrow$
–	–	13.18
✓	–	13.21
–	✓	13.24
✓	✓	**13.47**

Table 5: Class-aware temporal-distillation ablation with fixed pairing and masks.

Static KL	Dynamic logits	$\text{mIoU}_{\text{avg}} \uparrow$
–	–	13.21
✓	–	13.18
–	✓	13.23
✓	✓	**13.47**

no alignment. Spatial alignment alone produces the highest boundary recall and geometry–semantic attachment, improving them by 0.066 and 0.030 points, respectively. These complementary trends connect the observed gains to the intended mechanisms. Coordinate conditioning strengthens the local attachment between semantics and refined geometry, while joint spatial and temporal alignment stabilizes scene-aligned semantic predictions throughout future rollout. Alongside the mIoU improvements, these diagnostics provide mechanism-specific evidence beyond final task accuracy.

4.5 Visualizations

Figure 3 visualizes 4D occupancy forecasting. PreWorld gradually loses local semantic structure as the forecast horizon increases, whereas STC-World retains occupancy support that is better aligned with the ground truth, including dynamic objects and their surrounding drivable regions. This behavior reflects coordinate-conditioned point correction and adjacent-horizon temporal regularization.

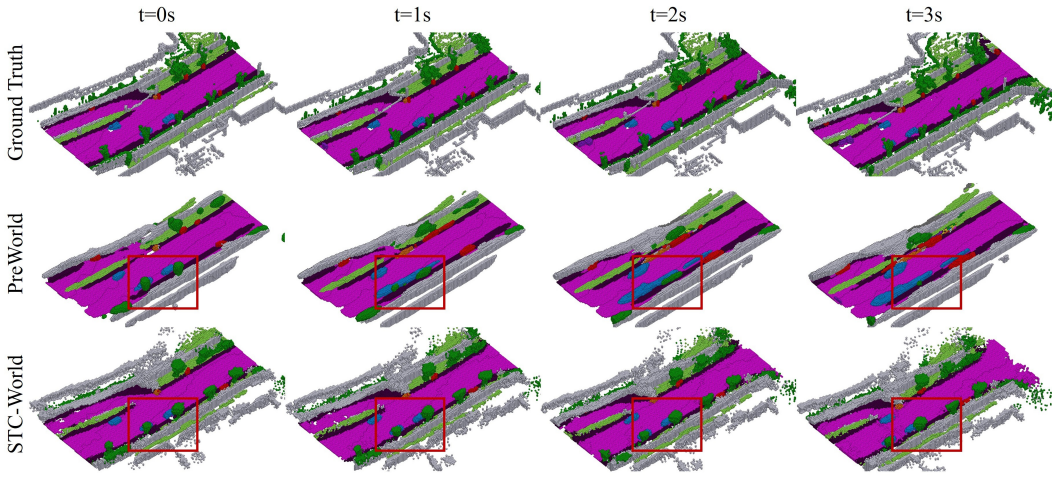

Figure 3: Qualitative 4D occupancy forecasting on Occ3D-nuScenes. Columns show the current frame and the 1, 2, and 3 s forecast horizons. From top to bottom, the rows present ground truth, PreWorld, and STC-World. Red boxes mark local regions where STC-World better preserves semantic occupancy and its spatial support over time.

5 Conclusion

We presented STC-World, a sparse world model that addresses spatiotemporally inconsistent semantic rollout in 4D occupancy forecasting. Coordinate-conditioned semantic alignment ties point-level logits to their refined spatial support, while adjacent-horizon temporal self-distillation regularizes semantic evolution over inherited query indices. The two mechanisms improve average mIoU to 13.47% on Occ3D-nuScenes, and the complete system yields 0.20 m average L2 error and 0.09% average collision rate in downstream planning. STC-World retains the efficiency of sparse modeling at 2.56 FPS. The temporal pairing is an internal continuity prior rather than physical object tracking, which remains a direction for stronger motion-aware correspondence.

AI USE STATEMENT

We used generative AI tools for language polishing and grammar checking. All reported measurements were produced by the executable training and evaluation pipelines described in this work rather than generated as text by an AI system. The authors made the final research decisions, verified the code and experimental outputs, reviewed all AI-assisted material, and take full responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

This work uses only the publicly available Occ3D-nuScenes benchmark and its associated nuScenes data in accordance with their licenses and terms of use. It involves no new data collection, human-subject study, or collection of personally identifiable information. All experiments are conducted as offline research evaluations and do not control a deployed vehicle or make real-world driving decisions.

REPRODUCIBILITY STATEMENT

The model formulation and training objectives are provided in Section 3, while the dataset, implementation details, and evaluation metrics are described in Section 4.1. The main comparisons and ablations are reported in Sections 4.2–4.4, and the stage-wise optimization protocol is specified in Appendix A.2.1. Source code, configurations, model checkpoints, and evaluation scripts will be released through the project page at `https://stc-world.github.io/`.

REFERENCES

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.

Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3991–4001, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Cao_MonoScene_Monocular_3D_Semantic_Scene_Completion_CVPR_2022_paper.html.

Junliang Chen, Huaiyuan Xu, Yi Wang, and Lap-Pui Chau. Occprophet: Pushing the efficiency frontier of camera-only 4d occupancy forecasting with an observer-forecaster-refiner framework. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/60b7841836d8f03484033050468fe6bf-Abstract-Conference.html.

Sitao Chen, Zhuangwei Zhuang, Hui Luo, Qingyao Wu, and Mingkui Tan. Geometry-aware spatio-temporal context modeling for 4d occupancy forecasting. *arXiv preprint arXiv:2608.15279*, 2026.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (5):3497–3505, 2026. doi: 10.1609/aaai.v40i5.37347.

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1607–1616. PMLR, 2018. URL https://proceedings.mlr.press/v80/furlanello18a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. doi: 10.48550/arXiv.1503.02531. URL https://arxiv.org/abs/1503.02531.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Hu_Planning-Oriented_Autonomous_Driving_CVPR_2023_paper.html.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9223–9232, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Huang_Tri-Perspective_View_for_Vision-Based_3D_Semantic_Occupancy_Prediction_CVPR_2023_paper.html.

Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19946–19956, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_SelfOcc_Self-Supervised_Vision-Based_3D_Occupancy_Prediction_CVPR_2024_paper.html.

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, 2024b. doi: 10.1007/978-3-031-73383-3_22.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27477–27486, 2025. doi: 10.1109/CVPR52734.2025.02559.

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. 8340–8350, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Jiang_VAD_Vectorized_Scene_Representation_for_Efficient_Autonomous_Driving_ICCV_2023_paper.html.

Xiang Li, Pengfei Li, Yupeng Zheng, Wei Sun, Yan Wang, and Yilun Chen. Semi-supervised vision-centric 3d occupancy world model for autonomous driving. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9c979c7a7bcc6a791c3b492697f97e1e-Abstract-Conference.html.

Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M. Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9087–9098, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Li_VoxFormer_Sparse_Voxel_Transformer_for_Camera-Based_3D_Semantic_Scene_Completion_CVPR_2023_paper.html.

Zhimin Liao, Ping Wei, Ruijie Zhang, Shuaijia Chen, Haoxuan Wang, and Ziyang Ren. I^2-World: Intra-inter tokenization for efficient dynamic 4d scene forecasting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25810–25819. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Junyi Ma, Xieyuanli Chen, Jiawei Huang, Jingyi Xu, Zhen Luo, Jintao Xu, Weihao Gu, Rui Ai, and Hesheng Wang. Cam4docc: Benchmark for camera-only 4d occupancy forecasting in autonomous driving applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21486–21495, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ma_Cam4DOcc_Benchmark_for_Camera-Only_4D_Occupancy_Forecasting_in_Autonomous_Driving_CVPR_2024_paper.html.

Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong Zhang, and Yuan Xie. COTR: Compact occupancy transformer for vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19936–19945, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ma_COTR_Compact_Occupancy_TRansformer_for_Vision-based_3D_Occupancy_Prediction_CVPR_2024_paper.html.

Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *2024 IEEE International Conference on Robotics and Automation*, pp. 12404–12411, 2024. doi: 10.1109/ICRA57147.2024.10611537.

Yiang Shi, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Xinggang Wang. Occupancy as set of points. In *European Conference on Computer Vision*, pp. 72–87, 2024. doi: 10.1007/978-3-031-73030-6_5.

Pin Tang, Zhongdao Wang, Guoqing Wang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma. Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15035–15044, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Tang_SparseOcc_Rethinking_Sparse_Latent_Representation_for_Vision-Based_Semantic_Occupancy_Prediction_CVPR_2024_paper.html.

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, 2023. doi: 10.52202/075280-2809.

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. URL https://arxiv.org/abs/2409.03272.

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. 21729–21740, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wei_SurroundOcc_Multi-camera_3D_Occupancy_Prediction_for_Autonomous_Driving_ICCV_2023_paper.html.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, and Yuexin Ma. Renderworld: World model with self-supervised 3d label. In *2025 IEEE International Conference on Robotics and Automation*, pp. 6063–6070, 2025. doi: 10.1109/ICRA55743.2025.11127609.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(9):9327–9335, 2025. doi: 10.1609/aaai.v39i9.33010.

Haiming Zhang, Ying Xue, Xu Yan, Jiacheng Zhang, Weichao Qiu, Dongfeng Bai, Bingbing Liu, Shuguang Cui, and Zhen Li. An efficient occupancy world model via decoupled dynamic flow and image-assisted training. *arXiv preprint arXiv:2412.13772*, 2024. doi: 10.48550/arXiv.2412.13772. URL https://arxiv.org/abs/2412.13772.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3713–3722, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Zhang_Be_Your_Own_Teacher_Improve_the_Performance_of_Convolutional_Neural_ICCV_2019_paper.html.

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. 9433–9443, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_OccFormer_Dual-path_Transformer_for_Vision-based_3D_Semantic_Occupancy_Prediction_ICCV_2023_paper.html.

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, 2024. doi: 10.1007/978-3-031-72624-8_4.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6772–6781, 2025. doi: 10.1109/CVPR52734.2025.00635.

SUPPLEMENTARY MATERIAL CONTENTS

Appendix A.1 Methodological and Implementation Details 15
A.1.1 Derivation of Spatiotemporal Semantic Inconsistency 15
A.1.2 Sparse Query Construction and Rollout 15
A.1.3 Sparse-to-Dense Occupancy Rendering 16
A.1.4 Complete Temporal Distillation Specification 16
A.1.5 Architecture and Optimization Details 17
A.2 Additional Experiments 17
A.2.1 Stage-wise Training and Optimization 17
A.2.2 Additional Design and Loss Ablations 18
A.2.3 Multi-seed Evaluation 18
A.2.4 Direct Evaluation of Semantic Consistency 19
A.2.5 Qualitative Motion Planning 20
A.2.6 Qualitative Reconstruction Success and Failure Cases 21
A.2.7 Limitations 22

A Appendix

A.1 Methodological and Implementation Details

A.1.1 Derivation of Spatiotemporal Semantic Inconsistency

The consistency condition in Section 3.1 concerns persistent spatial support rather than object identity. Let $C_\tau(\mathbf{x})$ denote the semantic class of a physical point or voxel location $\mathbf{x}$ at horizon τ. If a local motion map $T_{\tau\to\tau+1}$ carries the support to the next horizon, its semantic identity should satisfy

$$C_{\tau+1}(T_{\tau\to\tau+1}(\mathbf{x})) = C_\tau(\mathbf{x})$$

unless the support leaves the evaluated region, becomes occluded, or undergoes a true semantic change.

Sparse latent rollout can violate this condition spatially. The rendered label at voxel $\mathbf{v}$ is determined by the refined points assigned to that voxel,

$$\hat{Y}_\tau(\mathbf{v}) = \arg\max_c \max_{(i,j):\Pi(\mathbf{p}_{\tau,i,j})=\mathbf{v}} s_{\tau,i,j,c},$$

where $\Pi(\cdot)$ maps a point to a voxel. When logits are decoded primarily from $\mathbf{q}_{\tau,i}$, points with different refined coordinates can receive weakly differentiated semantics. Rendering then evaluates semantics at locations that were not explicit inputs to semantic decoding.

The temporal violation follows from horizon-wise supervision. The standard objective

$$\sum_{\tau=1}^{T} \mathcal{L}_{\mathrm{occ}}(\hat{\mathbf{Y}}_\tau, \mathbf{Y}_\tau)$$

constrains each marginal prediction but does not require

$$p_\theta(c_{\tau+1,i,j} \mid \mathcal{Z}_{\tau+1}) \approx p_\theta(c_{\tau,i,j} \mid \mathcal{Z}_\tau)$$

for point slots propagated under the same internal index. Plausible predictions can therefore be obtained independently at adjacent horizons while their semantic distributions drift. Index continuity remains an architectural relation, not evidence that a query tracks a physical object. We consequently use it as a local regularization prior and weaken the constraint for dynamic categories.

A.1.2 Sparse Query Construction and Rollout

The current sparse state is constructed from the camera history and observed ego poses,

$$\mathcal{Z}_0 = F_\theta(\mathcal{I}_{-H+1:0}, \mathcal{E}_{-H+1:0}).$$

The encoder samples multi-view and multi-frame features around 3D point hypotheses, mixes them into query states, performs query interaction, and refines the point support. At decoder layer ℓ, query i predicts point offsets and semantic logits,

$$\Delta\mathbf{p}^\ell_{\tau,i,j} = f_{\mathrm{reg}}(\mathbf{q}^\ell_{\tau,i})_j, \qquad \mathbf{s}^\ell_{\tau,i,j} = f_{\mathrm{cls}}(\mathbf{q}^\ell_{\tau,i})_j.$$

The corresponding point update is

$$\bar{\mathbf{p}}^\ell_{\tau,i} = \frac{1}{R_\ell}\sum_{j=1}^{R_\ell} \mathbf{p}^\ell_{\tau,i,j}, \qquad \mathbf{p}^{\ell+1}_{\tau,i,j} = \bar{\mathbf{p}}^\ell_{\tau,i} + \Delta\mathbf{p}^\ell_{\tau,i,j}.$$

This refinement allows one sparse query to represent a local geometric support rather than a single voxel.

For each inherited query, the future rollout combines its feature with the current ego state and a horizon embedding,

$$\tilde{\mathbf{q}}_{\tau+1,i} = \psi_{\mathrm{roll}}\left(\mathbf{q}_{\tau,i}, \mathbf{e}_0, \psi_{\mathrm{time}}(\tau)\right).$$

The updated feature predicts future refined points and semantic logits. Dynamic categories can additionally receive velocity-related displacement, whereas static regions are governed primarily by geometric refinement. Historical ego poses are used before rollout to align the observed camera history. The transition itself receives the current ego-state vector $\mathbf{e}_0$ and no future ground-truth ego pose, occupancy label, or trajectory.

A.1.3 SPARSE-TO-DENSE OCCUPANCY RENDERING

At each horizon, the renderer converts the 17 corrected class logits to independent sigmoid scores. If the highest-scoring category is manmade or vegetation, the score vector is attenuated by $\min(0.1/d_{\text{norm}}, 1)$, where d_{norm} is the normalized distance from the point to its query center. A point is retained only when its metric distance from the center is below 3 m and its maximum score exceeds the threshold of its predicted class. The threshold vector in occupancy-class order is

$$\boldsymbol{\theta} = (\underbrace{0.35, \ldots, 0.35}_{15 \text{ classes}}, 0.25, 0.30),$$

where the final entries correspond to manmade and vegetation. Retained points are decoded to metric coordinates, quantized with a 0.4 m voxel size, and discarded if their indices lie outside $[0, 200) \times [0, 200) \times [0, 16)$.

For voxel $\mathbf{v}$ and class c, the renderer keeps the maximum score among retained points assigned to that voxel,

$$\mathbf{o}_\tau(\mathbf{v}, c) = \max_{(i,j) \in \mathcal{K}_\tau : \Pi(\mathbf{p}_{\tau,i,j}) = \mathbf{v}} \tilde{s}_{\tau,i,j,c},$$

where $\mathcal{K}_\tau$ is the retained point set and $\tilde{s}$ is the score after optional attenuation. Empty score entries are initialized to zero. Because aggregation is class-wise, conflicting points can contribute different class maxima to one voxel; the final semantic label is selected from the largest aggregated score.

The score volume then undergoes a $3 \times 3 \times 3$ morphological closing implemented by max-pool dilation followed by min-pool erosion, with the original scores restored at high-confidence input voxels. A voxel is occupied if at least one post-processed class score exceeds its class threshold. Its label is the class with the largest score, while all remaining voxels receive label 17 for empty space.

A.1.4 COMPLETE TEMPORAL DISTILLATION SPECIFICATION

Let $N_0 = 720$ be the number of current queries and $\mathcal{I}_0 = \{1, \ldots, N_0\}$ their index set. The implementation preserves each inherited index, so $\ell_{\tau \to \tau+1}(i) = i$ for $i \in \mathcal{I}_0$. Point slot j is paired with the same slot at the next forecast horizon. Queries appended during rollout are excluded because they have no predecessor in $\mathcal{Z}_0$.

The Occ3D class order is

(others, barrier, bicycle, bus, car, construction vehicle, motorcycle, pedestrian,
traffic cone, trailer, truck, driveable surface, other flat, sidewalk, terrain, manmade, vegetation).

We use $\mathcal{C}_{\text{static}} = \{11, 12, 13, 14, 15, 16\}$ and $\mathcal{C}_{\text{dyn}} = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Class 0 is a heterogeneous catch-all category without a stable semantic identity. Class 1 represents thin barrier support on which an inherited point slot can move on or off during refinement. Both are excluded from chain distillation.

With distillation temperature T_d, the teacher and student probabilities are

$$p^{\text{tea}}_{\tau,i,j,c} = \sigma\left(\frac{s^{\text{tea}}_{\tau,i,j,c}}{T_d}\right), \qquad p^{\text{stu}}_{\tau+1,i,j,c} = \sigma\left(\frac{s^{\text{stu}}_{\tau+1,i,j,c}}{T_d}\right).$$

For the set Ω_τ of valid inherited point slots, the static loss is

$$\mathcal{L}^{\text{static-KL}}_\tau = \frac{1}{|\Omega_\tau|\,|\mathcal{C}_{\text{static}}|} \sum_{(i,j) \in \Omega_\tau} \sum_{c \in \mathcal{C}_{\text{static}}} T_d^2 \mathrm{KL}_{\mathrm{B}}\left(p^{\text{tea}}_{\tau,i,j,c} \middle\| p^{\text{stu}}_{\tau+1,i,j,c}\right).$$

Dynamic categories use a weaker logit-distance objective,

$$\mathcal{L}^{\text{dyn}}_\tau = \frac{1}{|\Omega_\tau|\,|\mathcal{C}_{\text{dyn}}|} \sum_{(i,j) \in \Omega_\tau} \sum_{c \in \mathcal{C}_{\text{dyn}}} \left\| s^{\text{stu}}_{\tau+1,i,j,c} - s^{\text{tea}}_{\tau,i,j,c} \right\|_2^2.$$

The training-only validity mask decodes each forecast point to metric coordinates, applies the ground-truth planar ego-trajectory offset used for supervision, and re-encodes the point into the normalized range. Denoting this coordinate by $\tilde{\mathbf{p}}_{\tau,i,j}$, the single-horizon rule is

$$m_{\tau,i,j} = \mathbb{1}[\tilde{p}^x_{\tau,i,j} \geq 0], \qquad \Omega_\tau = \{(i,j) \mid m_{\tau,i,j} m_{\tau+1,i,j} = 1\}.$$

The chain mask tests the lower normalized x boundary after the training-time transform; full 3D range filtering belongs to the renderer. Only the first 720 inherited queries are eligible. New-query counts at the six future horizons are $(60, 60, 60, 60, 40, 40)$. If a point is invalid at either horizon, its pair is removed. If no pair remains, the clamped normalization gives zero contribution for that horizon pair. The mask establishes joint validity but does not certify physical identity.

The chain loss is evaluated over $\mathcal{Z}_1 \to \mathcal{Z}_2, \ldots, \mathcal{Z}_{T-1} \to \mathcal{Z}_T$ and excludes $\mathcal{Z}_0 \to \mathcal{Z}_1$. We set $\alpha_\tau = 0.9^{\tau-1}$, $\lambda_{\text{dyn}} = 0.1$, anneal T_d from 4.0 to 1.0, and weight the complete chain objective by $\lambda_{\text{chain}} = 0.75$.

A.1.5 Architecture and Optimization Details

The sparse representation contains 720 current queries and adds $(60, 60, 60, 60, 40, 40)$ queries over the six future horizons. Six refinement stages expand each query to $(1, 4, 16, 24, 32, 48)$ points. The base sparse camera world model is trained end to end for 48 epochs with occupancy and trajectory supervision. Adjacent-horizon self-distillation is then trained for eight epochs, and its best validation checkpoint initializes the final stage. After attaching the coordinate-conditioned decoder, we jointly optimize it with the FPN, all existing non-ResNet perception, rollout, and planning parameters, and ResNet layer4 for 10 epochs. The ResNet stem and layers1–3 remain frozen, as do all batch-normalization affine parameters and running statistics.

AdamW uses weight decay 0.01 and initial learning rates of 2×10^{-4}, 1×10^{-5}, and 1×10^{-6} for the new decoder, existing non-backbone modules, and layer4, respectively. Final training uses eight NVIDIA A800 GPUs with 12 samples per GPU, no gradient accumulation, a global batch size of 96, and seed 17. Refined coordinates are normalized in the current LiDAR range, and future occupancy annotations are transformed to the current-frame reference system. Future ego poses and trajectories are used only for supervision, training-time coordinate transforms, validity masking, and planning evaluation.

A.2 Additional Experiments

A.2.1 Stage-wise Training and Optimization

Base sparse-world initialization. We first train the sparse world model end to end for 48 epochs under occupancy and trajectory supervision. This stage jointly learns the ResNet–FPN visual encoder, sparse query construction, point refinement, ego-conditioned future rollout, sparse-to-dense occupancy rendering, and the planning head. The resulting checkpoint provides the common initialization for the subsequent temporal and spatial alignment stages. We load it without parameter-key remapping to preserve the one-to-one correspondence between the pretrained parameters and the STC-World architecture.

Temporal alignment stage. Starting from the base model, we optimize adjacent-horizon temporal self-distillation over the five forecast-to-forecast pairs formed by the six future horizons for at most eight epochs with $\lambda_{\text{chain}} = 0.75$; the current-to-first-forecast transition is excluded. The previous forecast horizon provides a stop-gradient teacher for the next horizon over inherited query indices, while the validity mask removes slots that are invalid at either horizon. In this stage, the ResNet stem and layer1 are frozen; ResNet layers2–4, the FPN, and all non-ResNet modules, including the planning head, are trainable. Batch-normalization running statistics are fixed. The effective global batch is 96, formed from 12 samples on each of 8 GPUs with four-step gradient accumulation.

Joint spatiotemporal alignment stage. We attach the zero-initialized coordinate-conditioned semantic decoder and jointly optimize it with the FPN, all existing non-ResNet perception, rollout, and planning parameters, and ResNet layer4. The ResNet stem and layers1–3 are frozen throughout this stage. Batch-normalization affine parameters and running statistics are also fixed, and gradient checkpointing is disabled. We train for 10 epochs with a global batch size of 96, obtained from 12 samples on each of 8 GPUs without gradient accumulation. AdamW uses weight decay 0.01, gradients are clipped to a maximum norm of 5, and the first 500 optimizer steps use linear warm-up. Table 6 reports the stage-wise learning rates. Unless stated otherwise, the main experiments use seed 17 and the same six-horizon forecasting configuration.

Table 6: Learning-rate schedule for joint STC-World training.

Parameter group	Epochs 1–4	Epochs 5–8	Epochs 9–10
Coordinate-conditioned decoder	2×10^{-4}	1×10^{-4}	5×10^{-5}
Existing non-backbone modules	1×10^{-5}	1×10^{-5}	5×10^{-6}
ResNet layer4	1×10^{-6}	1×10^{-6}	5×10^{-7}

Component-comparison budgets. Table 7 discloses the optimization budget behind Table 4. "No alignment" and "spatial only" disable the chain loss during their 10-epoch final adaptation but start from the common temporal checkpoint; "temporal only" is that checkpoint before final adaptation. The comparison is therefore stage-wise rather than a compute-matched training-from-scratch factorial. All final-stage runs use the same trainable parameter groups described above. The batch-size difference reflects the recorded runs and is reported explicitly rather than treated as a controlled variable.

Table 7: Optimization provenance for the component comparison. "Global batch" includes gradient accumulation.

Reported model	Initialization	Epochs	Trainable scope	Global batch
Base sparse model	–	48	End-to-end sparse world model and planner	96
Temporal only	Base sparse model	8	ResNet layers2–4, FPN, all non-ResNet modules	96
No alignment	Temporal checkpoint	10	ResNet layer4, FPN, all non-ResNet modules	96
Spatial only	Temporal checkpoint	10	ResNet layer4, FPN, all non-ResNet modules	96
STC-World	Temporal checkpoint	10	ResNet layer4, FPN, all non-ResNet modules	96

All non-ResNet groups in Table 7 include the sparse perception, rollout, occupancy, trajectory, and planning parameters. Consequently, the planning results in Table 2 are end-to-end system comparisons; a frozen-head causal attribution is outside the evidence provided here.

A.2.2 Additional Design and Loss Ablations

Where and how to condition semantics. Tables 8 and 9 examine the design of spatial semantic alignment. Correcting post-refinement point logits reaches 13.47 mIoU, whereas a query-only residual and pre-refinement coordinates obtain 13.25 and 13.22. The useful signal therefore comes from the final point support used for occupancy rendering rather than from an additional query-level semantic branch. At the post-refinement stage, absolute or relative coordinates alone yield 13.18–13.19, and removing the query feature gives 13.20. Fourier encoding, additive fusion, and FiLM reach at most 13.23. Concatenating query features with absolute coordinates and query-relative offsets through an MLP performs best.

Table 8: Spatial stage used by the semantic residual. All variants retain the query feature; conditioning on post-refinement points performs best.

Coordinate stage	Residual target	$\text{mIoU}_{\text{avg}} \uparrow$
None	Query logits	13.25
Pre-refine	Point logits before refinement	13.22
Post-refine	Point logits after refinement	**13.47**

Loss-function ablation. Table 10 removes one supervision group at a time. Eliminating initial point localization, current semantic classification, current point refinement, or trajectory regression reduces average mIoU to 13.21–13.23. Removing future semantic classification reduces mIoU to 3.62, while removing future point refinement yields 2.01. Direct future geometric and semantic supervision is therefore essential; chain distillation regularizes adjacent horizons but does not replace their occupancy targets.

A.2.3 Multi-seed Evaluation

We repeat the complete training and full-validation evaluation with 5 random seeds. Each checkpoint is evaluated on all validation samples, and every entry in Table 11 averages the 1, 2, and 3 s horizons.

Table 9: Coordinate representation and query fusion at the post-refinement stage. Absolute and relative coordinates with MLP concatenation give the strongest result.

Coordinate input	Query	Fusion	$\text{mIoU}_{\text{avg}} \uparrow$
Absolute	✓	Concat. MLP	13.18
Relative	✓	Concat. MLP	13.19
Absolute + relative	–	Geometry-only MLP	13.20
Fourier (absolute + relative)	✓	Concat. MLP	13.22
Absolute + relative	✓	Addition	13.19
Absolute + relative	✓	FiLM	13.23
Absolute + relative	✓	Concat. MLP	**13.47**

Table 10: Loss-function ablation of STC-World. Each row removes one supervision group from the full objective.

Removed loss	$\text{mIoU}_{\text{avg}} \uparrow$
None (full objective)	**13.47**
Initial point localization	13.22
Current semantic classification	13.23
Current point refinement	13.23
Future semantic classification	3.62
Future point refinement	2.01
Ego-trajectory regression	13.21

All 5 completed runs are shown. For a metric x, these statistics are

$$\bar{x} = \frac{1}{5}\sum_{i=1}^{5} x_i, \qquad s^2 = \frac{1}{4}\sum_{i=1}^{5}(x_i - \bar{x})^2. \tag{1}$$

Table 11: Full-validation results across 5 random seeds.

Seed	mIoU_{avg} (%)	IoU_{avg} (%)	L2_{avg} (m)	$\text{Collision}_{\text{avg}}$ (%)
17	13.4700	22.1633	0.2000	0.0902
23	13.4367	22.2100	0.1985	0.0810
41	13.4467	22.1933	0.1982	0.0882
73	13.4267	22.2367	0.1990	0.0731
101	13.4267	22.1800	0.1985	0.0909

The five-seed mean and sample variance are 13.4414 and 0.000318 for mIoU_{avg}, 22.1967 and 7.9444×10^{-4} for IoU_{avg}, 0.19883 m and 5.3182×10^{-7} m^2 for L2_{avg}, and 0.08467% and 5.7350×10^{-5} $\%^2$ for the average collision rate.

A.2.4 Direct Evaluation of Semantic Consistency

We directly examine the two forms of semantic alignment targeted by STC-World on the same 4,219 validation samples. The comparison uses four existing checkpoints corresponding to neither mechanism, spatial alignment only, temporal distillation only, and their joint application; no model is retrained for this analysis. Geometry–semantic attachment (GSA) measures semantic accuracy where both prediction and ground truth are occupied, while boundary macro recall focuses on occupied ground-truth voxels adjacent to free space or another semantic class. Dense semantic drift measures the predicted-label change rate on pose-aligned voxel correspondences whose ground-truth class remains unchanged. We construct these correspondences by transforming voxel centers between LiDAR frames with the nuScenes ego poses, so the metric evaluates scene-aligned predictions rather than treating query indices as physical object tracks.

Table 12: Direct semantic-consistency diagnostics on the full validation set. All values are percentages. Higher is better for boundary recall and GSA. Lower is better for drift.

Method	Boundary recall	GSA	Drift 0→1 s	Drift 0→2 s	Drift 0→3 s
No alignment	28.782	84.857	22.996	22.407	22.407
Spatial alignment only	28.848	84.887	22.994	22.462	22.464
Temporal distillation only	28.821	84.841	22.993	22.405	22.414
STC-World	28.790	84.829	22.889	22.375	22.377

The diagnostics reveal complementary effects. Spatial alignment alone gives the highest boundary recall and GSA, improving them over no alignment by 0.066 and 0.030 percentage points, respectively. STC-World attains the lowest dense semantic drift at every evaluated horizon. Relative to no alignment, the reductions are 0.107, 0.032, and 0.030 percentage points at 1, 2, and 3 s, with a mean reduction of 0.056 points. Temporal distillation without spatial alignment changes drift only marginally, whereas applying it to coordinate-aligned point semantics produces a consistent reduction across horizons. This pattern supports the role of the two modules as complementary alignment regularizers: coordinate conditioning improves local geometry–semantic attachment, and their joint use stabilizes scene-aligned semantics during rollout. Together with the higher final occupancy accuracy in Tables 1 and 4, these measurements show that spatiotemporal semantic alignment improves the final predictions rather than acting solely as an unconstrained capacity increase. The drift values remain close in absolute terms, so we interpret the temporal mechanism as a regularizer over scene-aligned semantics, not as physical object tracking.

A.2.5 Qualitative Motion Planning

Figure 4 visualizes downstream planning in straight-road and turning scenarios. The predicted ego trajectories progress smoothly through the forecast sequence and remain within the inferred drivable space. These examples complement the end-to-end L2 and collision results in Table 2. Because the planning head is jointly optimized, the figure illustrates the complete system rather than isolating occupancy under a frozen planning head.

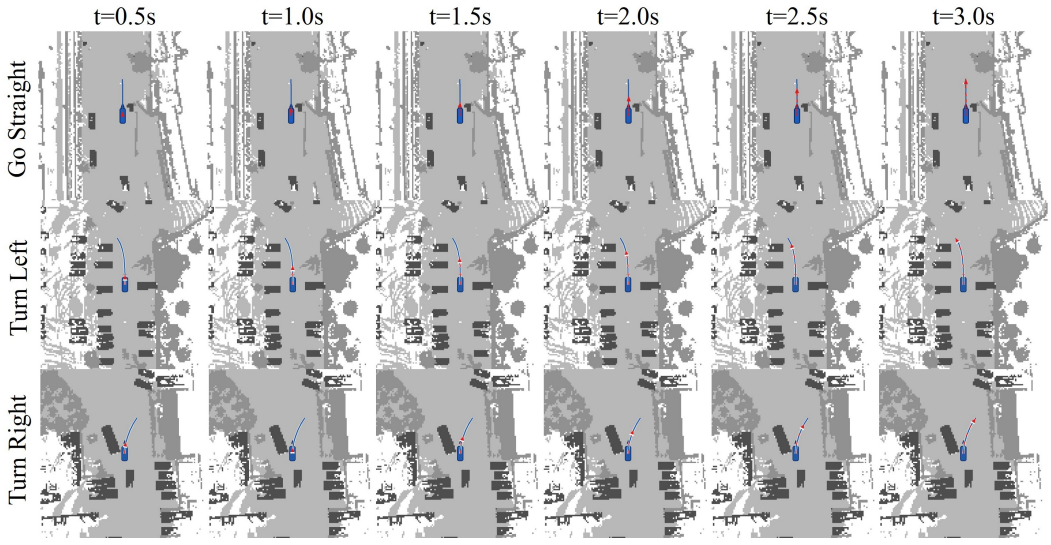

Figure 4: Qualitative motion-planning results on Occ3D-nuScenes. Rows show going straight, turning left, and turning right. Columns advance from 0.5 to 3.0 s at 0.5 s intervals, with the predicted ego trajectory overlaid on the occupancy map.

A.2.6 Qualitative Reconstruction Success and Failure Cases

Figure 5 presents a successful reconstruction in a structured road scene. STC-World preserves the longitudinal road layout, separates dynamic vehicles from the drivable surface, and retains the principal vegetation and man-made boundaries over the 3 s rollout. The spatial support remains coherent as the forecast advances, illustrating the behavior captured quantitatively by the occupancy and semantic-consistency evaluations.

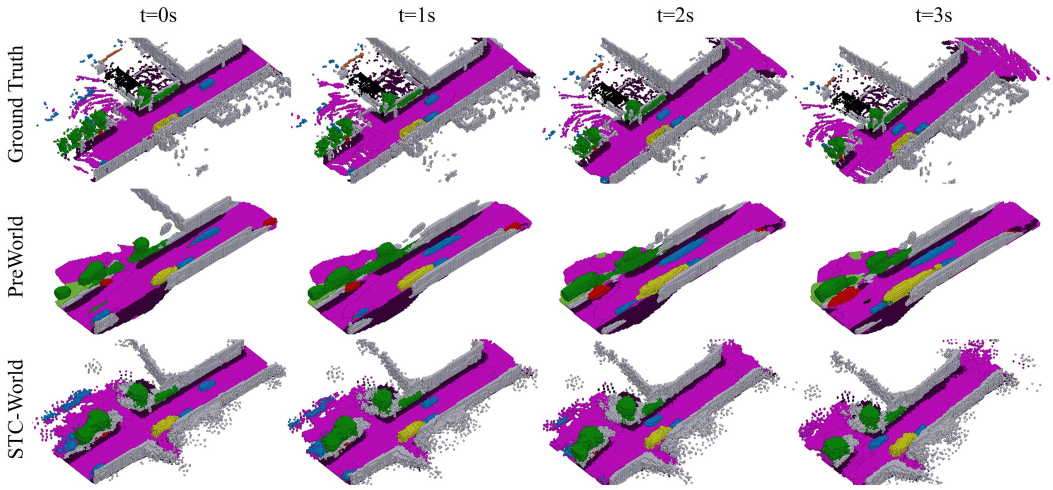

Figure 5: Successful 4D occupancy reconstruction. Columns show 0, 1, 2, and 3 s, and rows show ground truth, PreWorld, and STC-World from top to bottom. STC-World preserves the road topology, major semantic regions, and dynamic-object support throughout the rollout.

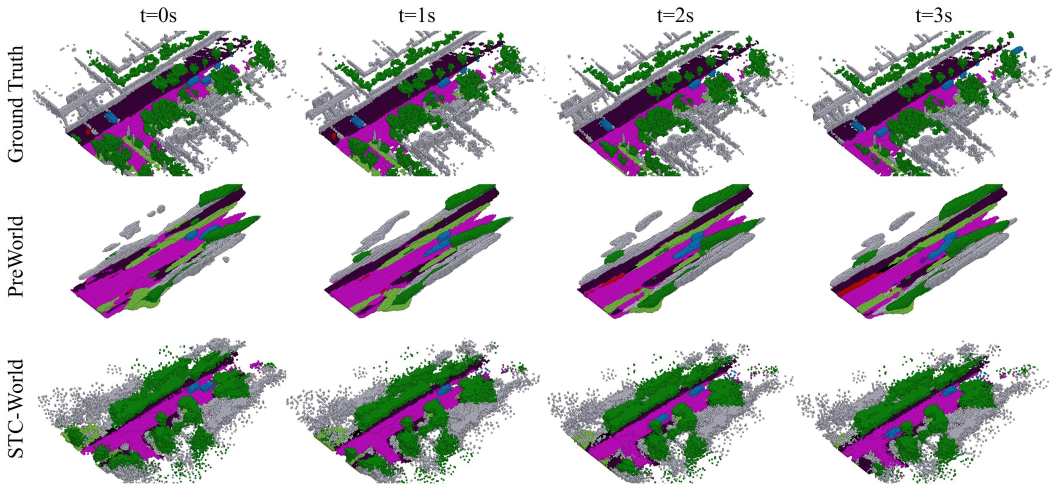

Figure 6: A common failure case in a dense, heavily occluded scene. Columns show 0, 1, 2, and 3 s, and rows show ground truth, PreWorld, and STC-World from top to bottom. STC-World recovers the coarse road layout but over-predicts occupied support and merges vegetation, drivable surface, and thin structural boundaries.

All models obtain low occupied-space overlap in Figure 6. STC-World predicts 60.2k–63.2k occupied voxels across the four horizons, compared with 32.8k–37.4k in the ground truth, yielding a 1.6–1.8$\times$ over-expansion. Vegetation and drivable surface spread into neighboring space, while thin structures, free-space gaps, and semantic boundaries are lost. In this dense, occluded scene, the fixed sparse budget appears to preserve coarse layout while under-resolving thin structures and over-aggregating frequent classes.

A.2.7 LIMITATIONS

STC-World uses inherited query and point-slot indices as a local continuity prior for adjacent-horizon distillation rather than explicitly tracking physical objects. Rapid motion, query reassignment, object entry or exit, and severe occlusion can therefore weaken the correspondence used by the temporal regularizer. The fixed sparse query budget also limits the resolution of thin structures and heavily occluded regions, as illustrated by the failure case above. In addition, the planning pathway is jointly optimized with the occupancy model, so the reported planning numbers characterize the complete end-to-end system. Future work can investigate motion-aware correspondence and adaptive allocation of sparse supports while retaining the efficiency of query-based rollout.