OCCWAM: 4D OCCUPANCY-AUGMENTED WORLD-ACTION MODELING FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

World-Action Models (WAMs) offer a promising paradigm for end-to-end autonomous driving by jointly modeling future observations and driving actions. However, image- and video-based world representations do not explicitly encode occupied and free space in 3D, requiring action models to infer these spatial constraints from visual features. To address this limitation, we propose OccWAM, the first World-Action Model to explicitly incorporate 4D occupancy modeling. Within a unified flow-matching framework, OccWAM jointly generates future visual observations, temporally evolving 3D occupancy, and ego trajectories using video, occupancy, and action experts. The occupancy generation task provides explicit supervision for volumetric scene structure, while cross-modal token attention within a Mixture-of-Transformers architecture enables the action expert to integrate spatial information from the occupancy expert. To focus interactions on motion-relevant regions while limiting the cost of processing dense occupancy tokens, we introduce Trajectory-Centric Occupancy Routing (TCOR). A lightweight scene-conditioned network scores predefined trajectory anchors and selects the top three, and TCOR routes occupancy tokens in their neighborhoods to the action expert, concentrating occupancy–action interactions on regions relevant to prospective ego motion. Experiments on nuScenes and NAVSIM demonstrate that OccWAM achieves an average trajectory L2 error of $0.36\,\mathrm{m}$ and a collision rate of 0.12% on nuScenes, and a PDMS of 91.1 on NAVSIM. Furthermore, TCOR reduces the number of occupancy tokens participating in cross-expert interaction by 90% while maintaining planning accuracy.

1 INTRODUCTION

Vision-Language-Action (VLA) models have shown strong potential for end-to-end autonomous driving(Hu et al., 2023; Jiang et al., 2023; Tian et al., 2024; Wu et al., 2022; Li et al., 2026; Zhang et al., 2025; Zhou et al., 2025b; Li et al., 2025b; Zhou et al., 2025a) by connecting visual understanding with action generation. However, many VLA policies learn to infer driving actions from observed context without explicitly modeling how the surrounding environment evolves. Anticipating such evolution is essential for planning ahead in dynamic driving scenarios. World-Action Models (WAMs)(Ignat Georgiev & Garg, 2024; Shi et al., 2026; Zhao et al., 2026; Xia et al., 2025) address this limitation by jointly learning future world evolution and ego actions, allowing predictive world modeling to inform driving policies. Beyond learning to predict the future, a central question is how to represent the predicted world so that its information can effectively support action generation.

In video-based WAMs, future scene evolution is represented through 2D images or visual latents, and the action expert interacts with video tokens to exploit their semantic and dynamic information, as illustrated in Fig 1(a). While these representations capture rich visual content, they do not explicitly encode the volumetric spatial structure that constrains ego motion. Image changes reflect the combined effects of object motion, ego-viewpoint changes, and appearance variations, leaving spatial geometry entangled with visual factors such as texture and illumination. Consequently, interactions with video tokens do not directly expose where obstacles occupy space, how the surrounding free space is distributed, or how these spatial states evolve. The action expert must recover these

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.

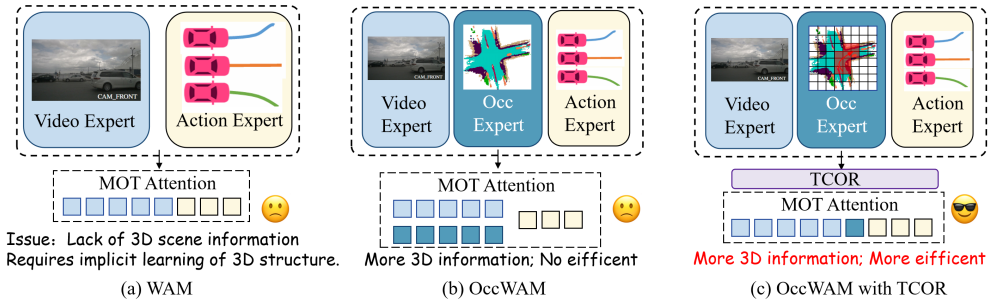


Figure 1: (a) Vanilla WAM lacks explicit 3D information. (b) Adding an Occupancy Expert adds 3D geometry but over dense tokens. (c) TCOR selects trajectory-relevant tokens for efficient 3D-aware world–action modeling.

properties from visual features and determine their implications for future ego motion. This reliance on implicit geometric inference places additional demands on action learning and may introduce spatial ambiguity into trajectory prediction.

To provide a more direct geometric basis for action generation, we argue that visual world modeling should be complemented by an explicit representation of the driving space. Volumetric occupancy offers such a representation by organizing the environment into a regular 3D grid with occupied and free states and associated semantic labels. Representing occupancy and ego trajectories in a common coordinate frame establishes a spatial basis for relating obstacle extents to potential ego paths and assessing collision risks. Extending this representation over time further captures how occupied and free regions evolve, providing geometric context for future motion rather than only the current scene. Combining visual world predictions with 4D occupancy therefore offers a way to preserve rich visual semantics and dynamics while making evolving spatial structure explicitly available to the action model.

Building on this insight, we propose OccWAM, the first World-Action Model to explicitly incorporate 4D occupancy modeling. OccWAM extends joint world–action generation to three complementary modalities including video, occupancy, and ego trajectories through dedicated experts built on Diffusion Transformers. These experts are trained under a joint flow-matching objective and interact within a unified Mixture-of-Transformers (MoT) architecture. The occupancy generation task provides explicit supervision for spatiotemporal scene structure, while cross-expert attention enables the action expert to integrate this geometric information with visual semantics and dynamics. Incorporating dense occupancy, however, substantially increases the number of tokens available for cross-expert interaction. Since the relevance of spatial information depends on prospective ego motion, candidate trajectories provide a natural reference for selecting which occupancy regions to expose to the action expert. We therefore introduce Trajectory-Centric Occupancy Routing (TCOR) within OccWAM to replace full occupancy–action interaction with trajectory-guided token selection, as illustrated in Fig. 1(b–c). A lightweight scene-conditioned network scores predefined trajectory anchors and selects the top three. TCOR then routes occupancy tokens in their neighborhoods to the action expert. Together, the three-expert architecture and trajectory-guided routing enable OccWAM to incorporate explicit 4D spatial information into action generation while limiting the interaction cost of dense occupancy.

We evaluate OccWAM on nuScenes and NAVSIM. On nuScenes, it achieves an average trajectory L2 error of 0.36 m and a collision rate of 0.12%. On NAVSIM, it achieves a PDMS of 91.1. Furthermore, TCOR reduces the number of occupancy tokens participating in cross-expert interactions by 90% while maintaining planning accuracy. These results demonstrate the planning performance of OccWAM and the effectiveness of trajectory-guided routing in reducing occupancy interaction overhead. Our main contributions are summarized as follows:

1. We propose OccWAM, the first World-Action Model for autonomous driving to jointly generate future video, 4D occupancy, and ego trajectories. Its three-expert architecture integrates visual world modeling with explicit spatiotemporal occupancy modeling to provide complementary information for action generation.

2. We introduce Trajectory-Centric Occupancy Routing (TCOR), which uses scene-conditioned trajectory anchors to select occupancy tokens for interaction with the action expert, focusing geometric information exchange on regions associated with prospective ego motion.

3. We evaluate OccWAM on nuScenes and NAVSIM, demonstrating strong planning performance across both benchmarks. We further show that TCOR preserves planning accuracy while largely reducing the number of occupancy tokens involved in cross-expert interactions.

2 RELATED WORK

2.1 WORLD-ACTION MODELS FOR AUTONOMOUS DRIVING

Video-based World–Action Models (WAMs) extend video generative modeling to policy learning, using predictive representations of scene dynamics to support action generation. Recent advances in robotic manipulation have demonstrated the potential of large-scale video pretraining for learning control policies, motivating its extension to autonomous driving through coupled visual forecasting and trajectory planning. PWM(Ignat Georgiev & Garg, 2024) couples action-free future visual forecasting with trajectory planning in an autoregressive framework, using anticipated scene evolution to inform driving decisions. DriveLaW(Xia et al., 2025) conditions a diffusion-based trajectory planner on latent representations from a video generator, providing predictive visual context for ego-motion planning. DriveWAM(Shi et al., 2026) adapts a pretrained video diffusion transformer through joint flow matching, coupling future video prediction with ego-trajectory generation so that planning can draw on learned visual dynamics. SimWAM(Zhao et al., 2026) uses video prediction during training to strengthen shared observation representations for trajectory planning, while allowing its action expert to predict driving trajectories without generating future video at inference. These methods have shown promising planning performance. However, their world representations primarily rely on images or visual latents, without explicit representations of 3D geometry and spatial constraints. This may reduce planning reliability when future trajectories depend on accurately assessing obstacle locations and available driving space.

2.2 OCCUPANCY-CENTRIC WORLD MODELING AND PLANNING

Occupancy-centric world models represent scene evolution through temporally changing 3D occupancy fields, providing explicit geometric and semantic information for planning. OccWorld(Zheng et al., 2023) tokenizes occupancy scenes and autoregressively predicts future scene and ego-motion tokens. OccLLaMA(Wei et al., 2024) integrates occupancy, language, and action into a unified autoregressive model, supporting both 4D occupancy forecasting and motion planning. DriveOccWorld(Yang et al., 2024) incorporates action conditions into occupancy and flow forecasting and uses an occupancy-based cost function to select trajectories. These studies demonstrate that occupancy and its predicted evolution can provide useful spatial information for driving decisions. However, in these approaches, occupancy serves as the primary predictive scene representation, and future video generation is not jointly modeled with occupancy evolution and ego planning. OccWAM extends this direction by jointly modeling future video, 4D occupancy, and ego trajectories through video, occupancy, and action experts under a unified flow-matching framework. This enables the action expert to combine visual semantics and dynamics with explicit spatiotemporal geometry for trajectory prediction.

3 METHOD

Fig. 2 illustrates the overall pipeline of OccWAM. Given the current image, the model first predicts the current 3D occupancy. The image and occupancy are encoded by their respective VAEs, while an action encoder embeds the ego state to provide conditioning information for trajectory generation. Future video, occupancy, and trajectory representations are initialized from noise. Before cross-expert interaction, TCOR selects a compact set of occupancy tokens relevant to candidate ego trajectories. Specifically, the Trajectory Anchor Selector combines global scene features with occupancy features along predefined anchors to score the anchors and predict their offsets. Local Occupancy Routing then selects occupancy tokens around the top three refined anchors. The video tokens, selected occupancy tokens, and action tokens interact through joint attention within a MoT

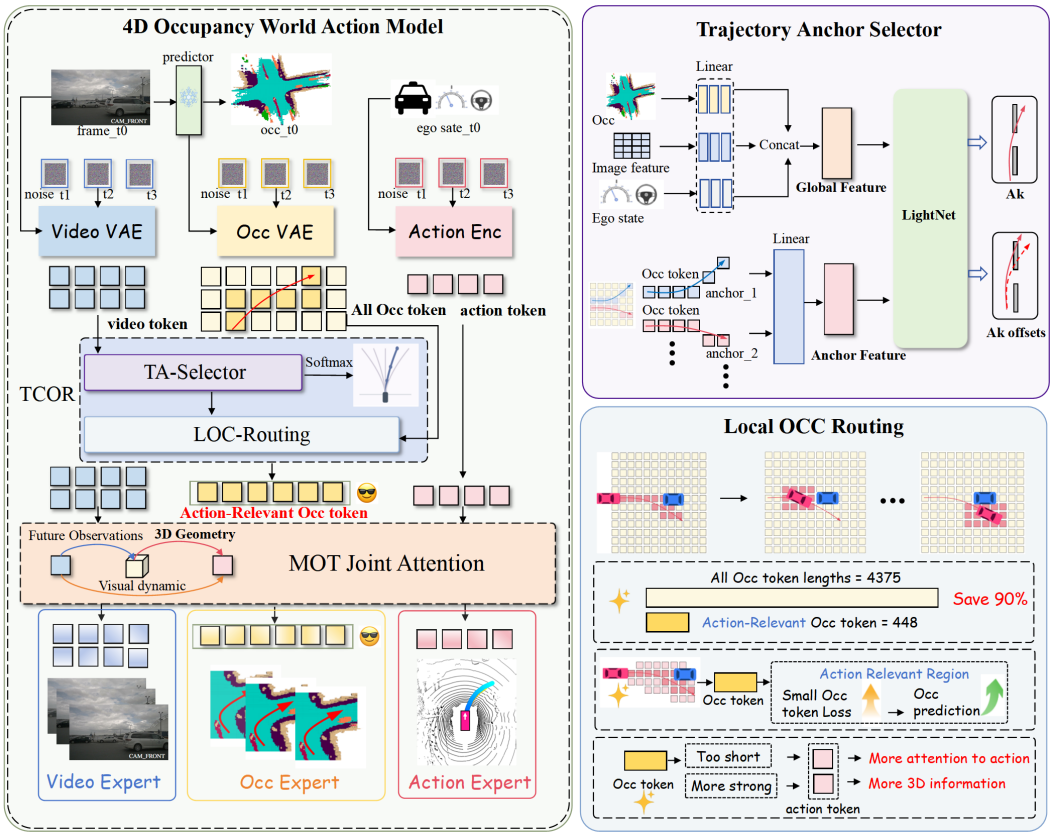


Figure 2: Overview of OccWAM. Video, occupancy, and action experts jointly generate future video, 4D occupancy, and ego trajectories through MoT attention. TCOR selects trajectory anchors and routes nearby occupancy tokens for cross-expert interaction, focusing on motion-relevant geometry while reducing interaction cost.

framework, enabling the three experts to jointly denoise future representations. Throughout this process, the Action Expert integrates visual dynamics with local spatiotemporal geometry to generate the planned ego trajectory.

3.1 Joint Modeling of Video, 4D Occupancy, and Action

Problem Formulation. Given current camera observations I_0, the current-frame occupancy O_0, and driving conditions c, our goal is to predict the future ego trajectory

$$Y = f_\theta(I_0, O_0, c) = \{y_h\}_{h=1}^{H}, \qquad y_h \in \mathbb{R}^2, \tag{1}$$

where c includes ego state, navigation information, and a high-level driving command, and f_θ denotes our model. During training, future visual observations and future occupancy are used solely as supervision targets; at inference they are neither provided nor required, and trajectory coordinates are expressed in the current ego coordinate system.

Three Modality-Specific experts. OccWAM comprises a Video Expert, an Occupancy Expert, and an Action Expert. Visual observations and occupancy are encoded into latent representations and tokenized, while trajectory coordinates are mapped to action tokens. Let X_v, X_o, and X_a denote the resulting token sequences; each expert maintains its own input projection, denoising timestep embedding, Transformer parameters, and output head. During flow-matching training, a noise level $t \sim U(0,1)$ is sampled, and for each modality $m \in \{v, o, a\}$ the target tokens are interpolated between their clean values z_m and unit Gaussian noise ϵ_m as

$$z_m^t = (1-t)\epsilon_m + t\, z_m, \qquad u_m = z_m - \epsilon_m, \tag{2}$$

while the conditioning tokens (current I_0, $\hat{O}_0$, c) are kept clean and never noised. Each expert is trained to regress the velocity u_m by minimizing

$$\mathcal{L}_m = \mathbb{E}_{t,\epsilon_m} \left\| v_{\theta_m}(z_m^t, t; \text{context}) - u_m \right\|_2^2 . \tag{3}$$

At each Transformer layer, the experts exchange information through mixture-of-transformers (Liang et al., 2025) attention. We next describe how each expert is instantiated.

Video Expert. The Video Expert is trained with the objective of generating future-frame driving images. Its network consists of a video VAE and a video DiT: the video VAE reuses the VAE from Wan2.2-5B, while the video DiT is initialized from the Wan2.2-5B(Wan et al., 2025) video model. Front-view images are projected by the video VAE into a compact set of latent tokens, which, together with the input high-level semantic text and the navigation command, serve as conditioning. Under a flow-matching(Lipman et al., 2023) objective, the expert predicts the latent features of future-frame images, which can be decoded by the VAE decoder back into future-frame images.

Occupancy Expert. The Occupancy Expert is trained with the objective of generating future-frame occupancy. Its network consists of an occupancy VAE and an occupancy DiT: the occupancy VAE reuses the VAE from DOME(Gu et al., 2024), while the occupancy DiT shares the same architecture as the video DiT but is not initialized from the Wan video model; instead, it is trained from scratch. The current-frame predicted occupancy is likewise projected by the occupancy VAE into a compact set of latent tokens, which interact with the encoded text information through cross-attention. Under a flow-matching objective(Lipman et al., 2023), the expert predicts the latent features of future-frame occupancy, which can be decoded by the occupancy VAE decoder back into future-frame occupancy.

Action Expert. The Action Expert is trained to produce future ego trajectories. Rather than a standalone policy, it is a lightweight DiT (hidden size d_a=1024) that participates in the same joint flow-matching denoising as the other experts: the raw ego state is first embedded by a small MLP, and future waypoints are mapped into action tokens. At every denoising step, action generation is coupled with the evolving video and occupancy streams, so that the planned trajectory is conditioned on both the predicted visual dynamics and the geometric constraints. The denoised action tokens are decoded by the action head into the waypoints Y.

3.2 Trajectory-Centric Occupancy Routing

Processing the full occupancy sequence introduces substantial overhead into MoT. We therefore predict a set of trajectory hypotheses from current observations and use them to determine which occupancy tokens participate in subsequent world-action modeling.

Trajectory Anchor Bank. Each training trajectory is a sequence of H=6 future 2D positions in the current ego coordinate frame, expressed in meters. We group trajectories into three navigation commands, namely left turn, right turn, and straight, and within each command build an anchor bank of M=12 anchors via offline K-means clustering, yielding 36 prototypes, as detailed in App. A.2. Anchor coordinates are stored as fixed buffers and never updated by gradients. At runtime the command identifies the applicable subset, and online hypothesis prediction uses only current observations and driving conditions, without access to future ground-truth trajectories or noisy future action tokens.

Trajectory Anchor Selector. For each anchor k and its h-th waypoint $a_{k,h}$, we aggregate features from the predicted current-frame occupancy $\hat{O}_0$ around $a_{k,h}$ using a precomputed Gaussian kernel normalized over the full occupancy grid. This yields a route feature $r_{k,h}$. We combine $r_{k,h}$ with the waypoint coordinates, global occupancy and visual features, and driving conditions to form a candidate feature $u_{k,h}$, as detailed in App. A.3.

Two lightweight heads score and refine each anchor. The score head averages its candidate features across waypoints and produces an anchor logit:

$$s_k = f_{\text{cls}}\left(\frac{1}{H} \sum_{h=1}^{H} u_{k,h} \right) . \tag{4}$$

Table 1: Comparison of planning performance on the NAVSIM v1 navtest split. NC, DAC, EP, TTC, and C denote no-at-fault collision, drivable-area compliance, ego progress, time-to-collision, and driving comfort, respectively. In the sensor column, C and L denote camera and LiDAR inputs. Higher is better for all metrics. The best results among learning-based methods are highlighted in bold.

Method	Reference	Sensors	NC↑	DAC↑	EP↑	TTC↑	C↑	PDMS↑
Human Agent	–	–	100.0	100.0	87.5	100.0	99.9	94.8
Traditional E2E planners								
UniAD(Hu et al., 2023)	CVPR'23	6×C	97.8	91.9	78.8	92.9	**100.0**	83.4
WorldRFT(Yang et al., 2025)	AAAI'26	3×C	97.8	96.8	81.7	94.0	**100.0**	87.8
SeerDrive(Zhang et al., 2025)	NeurIPS'25	3×C+L	98.4	97.0	83.2	94.9	99.9	88.9
VLM-based planners								
AutoVLA(Zhou et al., 2025b)	NeurIPS'25	3×C	98.4	95.6	81.9	**98.0**	99.9	89.1
ReCogDrive(Li et al., 2025b)	ICLR'26	1×C	97.9	97.3	**87.3**	94.9	**100.0**	90.8
DriveVLA-W0(Zhou et al., 2025a)	ICLR'26	1×C	98.7	**99.1**	83.3	95.3	99.3	90.2
World-model-based planners								
PWM(Ignat Georgiev & Garg, 2024)	NeurIPS'25	1×C	98.6	95.9	81.8	95.4	**100.0**	88.1
DriveLaw(Xia et al., 2025)	CVPR'26	1×C	**99.0**	97.1	81.3	96.7	**100.0**	89.1
DriveDreamer-Policy(Zhou et al., 2026)	arXiv'26	3×C	98.4	97.1	83.5	95.1	**100.0**	89.2
DriveWAM(Shi et al., 2026)	arXiv'26	1×C	98.3	98.1	84.3	95.2	**100.0**	90.1
SimWAM(no RL)(Zhao et al., 2026)	arXiv'26	1×C	98.7	98.0	83.9	95.9	100.0	90.3
OccWAM **(ours)**	–	1×C	98.7	98.6	84.3	96.5	**100.0**	**91.1**

The offset head predicts a displacement for each waypoint:

$$\Delta a_{k,h} = \tanh\big(f_{\text{reg}}(u_{k,h})\big), \qquad \Delta a_{k,h} \in \mathbb{R}^2. \tag{5}$$

Its final layer is zero-initialized, so each refined anchor initially coincides with its original prototype. Let $\mathcal{K}_c$ denote the anchors associated with the current navigation command. We select the three highest-scoring anchors within $\mathcal{K}_c$ and refine them using their predicted offsets:

$$\{k_r\}_{r=1}^{3} = \text{TopK}_{k\in\mathcal{K}_c}(s_k, 3), \qquad \hat{A}_r = A_{k_r} + \Delta A_{k_r}, \quad r = 1, \ldots, 3. \tag{6}$$

The refined hypotheses $\{\hat{A}_r\}$ determine which occupancy tokens are retained at each timestamp, as described below. With TCOR enabled, the selected tokens are used for subsequent OCC Expert denoising, cross-expert attention, and local occupancy supervision. The selector is trained with auxiliary classification and regression losses (App. A.4); the final ego trajectory is generated by the Action Expert rather than selected directly from the anchor bank.

Local Occupancy Routing. The three refined hypotheses determine which occupancy tokens are processed at each timestamp. For the current frame, we rank tokens by their minimum distance to the three trajectory corridors; for future frame h, we rank them by their minimum distance to the three corresponding waypoints:

$$d_{h,i} = \begin{cases} \min\limits_{r} \text{dist}\Big(p_{0,i}, \mathcal{C}(\hat{A}_r)\Big), & h = 0, \\ \min\limits_{r} \left\|p_{h,i} - \hat{a}_{r,h}\right\|_2, & h = 1, \ldots, H, \end{cases} \tag{7}$$

where $p_{h,i}$ is the ground-plane center of token i and $\mathcal{C}(\hat{A}_r)$ includes the ego origin. We retain the 64 nearest tokens at each timestamp:

$$\mathcal{I}_h = \text{TopKSmallest}_i(d_{h,i}, 64). \tag{8}$$

With TCOR enabled, only these occupancy tokens participate in subsequent joint denoising; the occupancy loss supervises the selected future tokens. Further details are provided in App. A.5.

4 EXPERIMENTS

Datasets. We train and evaluate a separate model on NAVSIM(Dauner et al., 2024) and nuScenes(Caesar et al., 2020) without mixing their training samples. On NAVSIM, we follow the

Table 2: Planning performance on nuScenes. VLM, WAM, and OCC indicate the use of a pretrained vision-language model, video-action modeling, and explicit 3D occupancy, respectively. Lower is better. Best results are bold; "–" denotes unavailable results.

Method	VLM	WAM	OCC	L2 (m) ↓				Collision rate (%) ↓			
				1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
OccNet(Sima et al., 2023)	×	×	✓	1.29	2.13	2.99	2.13	0.21	0.59	1.37	0.72
UniAD(Hu et al., 2023)	×	×	×	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37
VAD(Jiang et al., 2023)	×	×	×	0.17	**0.24**	0.60	0.34	0.04	0.27	0.67	0.33
OccWorld(Zheng et al., 2023)	×	×	✓	0.39	0.73	1.18	0.77	0.11	0.19	0.67	0.32
PreWorld(Li et al., 2025a)	×	×	×	0.22	0.30	**0.40**	**0.31**	0.21	0.66	0.71	0.53
Percept-WAM(Han et al., 2025)	✓	✓	×	0.17	0.35	0.63	0.38	–	–	–	–
OccWAM (ours)	×	✓	✓	**0.16**	0.32	0.59	0.36	**0.00**	**0.07**	**0.28**	**0.12**

Table 3: Comparison with representative world models on 3D occupancy forecasting. "Avg." denotes the average forecasting performance at 1 s, 2 s, and 3 s. Higher is better. Best results are bold.

Method	Input	mIoU (%) ↑				IoU (%) ↑			
		1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
Copy&Paste	3D-Occ	14.91	10.54	8.52	11.33	24.47	19.77	17.31	20.52
OccWorld-D(Zheng et al., 2023)	Camera	11.55	8.10	6.22	8.62	18.90	16.26	14.43	16.53
OccWorld-O(Zheng et al., 2023)	3D-Occ	25.78	15.14	10.51	17.14	34.63	25.07	20.18	26.63
OccLLaMA-F(Wei et al., 2024)	Camera	10.34	8.66	6.98	8.66	25.81	23.19	19.97	22.99
OccLLaMA-O(Wei et al., 2024)	3D-Occ	25.05	19.49	15.26	19.93	34.56	28.53	24.41	29.17
OccWAM (ours)	Camera	**11.98**	**9.80**	**8.45**	**10.07**	19.89	18.51	17.25	18.55
OccWAM (ours)	3D-Occ	**29.91**	**21.66**	**16.49**	**22.68**	**39.17**	**31.86**	**27.07**	**32.70**

standard `navtrain`/`navtest` splits with roughly 100K and 12K samples at 2 Hz, and associate the samples with dense 3D occupancy annotations from OpenScene to supervise the OCC Expert; the input current-frame occupancy is obtained from a fine-tuned OccNet. On nuScenes, we use the official 700-train/150-val split at 2 Hz and the Occ3D dense semantic occupancy annotations(Tian et al., 2023) to supervise the OCC Expert; the input current-frame occupancy is obtained from the frozen FBOCC, on a voxel grid of $200\times200\times16$ with 17 semantic classes.

Evaluation Metrics. On NAVSIM, we evaluate planning performance using the PDMS for NAVSIM v1. PDMS jointly evaluates no-at-fault collision, drivable-area compliance, time-to-collision, ego progress, and driving comfort. On nuScenes, we evaluate action prediction using the L2 displacement error at 1, 2, and 3 seconds, together with their average. Lower values indicate better performance for trajectory errors.

4.1 Main Results

Planning Performance on NAVSIM. Table 1 compares OccWAM with representative end-to-end, VLM-based, and world-model-based planners on the NAVSIM v1 `navtest` split. Using predicted current-frame occupancy and TCOR, OccWAM achieves a PDMS of 91.1, the highest among the learning-based methods listed in the table. This exceeds ReCogDrive (90.8) by 0.3 points and SimWAM (no RL; 90.3) by 0.8 points. OccWAM scores 98.7 in no-at-fault collision, 98.6 in drivable-area compliance, 84.3 in ego progress, 96.5 in time-to-collision, and 100.0 in driving comfort. These results show strong performance across the safety, progress, and comfort components of the NAVSIM metric.

Trajectory Prediction on NuScenes. Table 2 reports trajectory L2 error and collision rate on nuScenes. Using predicted current-frame occupancy and TCOR, OccWAM obtains L2 errors of 0.16, 0.32, and 0.59 m at 1, 2, and 3 s, respectively, with an average of 0.36 m. Its average error is lower than that of Percept-WAM (Han et al., 2025) (0.38 m), without using a pretrained VLM. OccWAM obtains collision rates of 0.00%, 0.07%, and 0.28% at the three horizons, yielding an

Table 4: Ablation on the OCC Expert contribution. All variants share the same Action Expert and training protocol, evaluated on NAVSIM `navtest` and nuScenes validation. The "Video" and "OCC" columns denote introducing the Video Expert and the OCC Expert, respectively. Higher is better for PDMS; lower is better for L2 and collision rate.

	World Experts		NAVSIM v1	nuScenes				
Variant	Video	OCC	PDMS ↑	1 s ↓	2 s ↓	3 s ↓	Avg. L2 ↓	Avg. Col. ↓
Action-only	×	×	87.2	0.29	0.51	0.76	0.52	0.34
Video–Action WAM	✓	×	89.9	0.19	0.37	0.65	0.41	0.16
OccWAM	✓	✓	**91.1**	**0.16**	**0.32**	**0.59**	**0.36**	**0.12**

Table 5: Effect of the occupancy source on trajectory prediction, evaluated on nuScenes. The two rows are separately trained models differing only in occupancy source (predicted vs. ground-truth). L2 is reported in meters.

Occupancy input	1 s↓	2 s↓	3 s↓	Avg.↓
Predicted OCC	0.16	0.32	0.59	0.36
Ground-truth OCC	**0.15**	**0.28**	**0.44**	**0.29**

Table 6: Results on nuScenes, comparing Full OCC, Random selection, and TCOR in terms of the tokens per frame, the ratio w.r.t. Full OCC, the average L2 error (m), and the end-to-end latency (ms).

Strategy	Tokens	Ratio	Avg. L2	latentcy(ms)
Full OCC	625	100%	0.35	4701
Random	64	10.2%	0.45	2834
TCOR	64	10.2%	**0.36**	**2841**

average of 0.12%. Together, these results support the planning performance of the TCOR variant under the reported evaluation protocol.

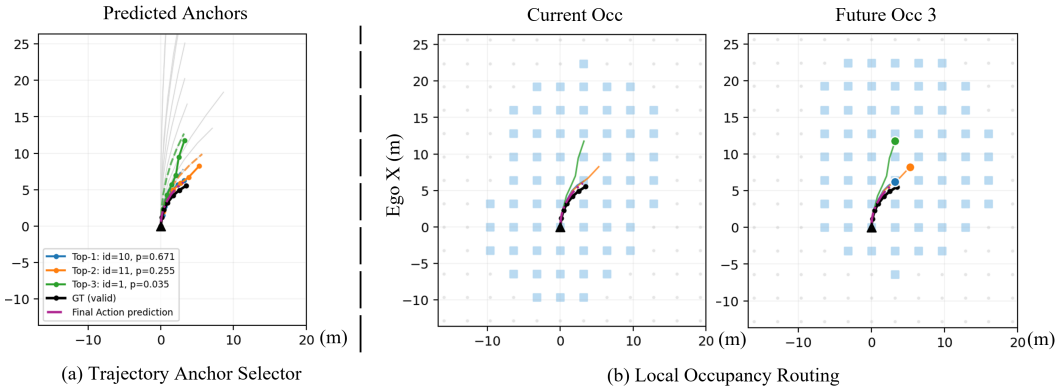
4D Occupancy Forecasting. Table 3 evaluates future occupancy forecasting on the Occ3D annotations of nuScenes. We use the Full OCC variant with TCOR disabled for this experiment. With predicted current-frame occupancy, OccWAM obtains mIoU scores of 11.98%, 9.80%, and 8.45% at 1, 2, and 3 s, respectively, averaging 10.07%. When ground-truth current-frame occupancy is provided, the corresponding scores rise to 29.91%, 21.66%, and 16.49%, with a reported average of 22.68%. Compared with OccLLaMA-O (Wei et al., 2024), the latter setting improves mIoU by 4.86, 2.17, and 1.23 percentage points at the three horizons, respectively. These results demonstrate the capability of OccWAM to forecast the temporal evolution of 3D occupancy.

4.2 Ablation Studie

Effect of the OCC Expert. Table 4 compares the Action-only, Video–Action, and complete OccWAM variants on NAVSIM and nuScenes. Adding the Video Expert increases NAVSIM PDMS from 87.2 to 89.9 and reduces the average nuScenes L2 error from 0.52 to 0.41 m. The complete model, which further includes the OCC Expert and TCOR, achieves a PDMS of 91.1, an average L2 error of 0.36 m, and a collision rate of 0.12%. These comparisons show that the occupancy-augmented model improves planning performance over the Video–Action baseline under the evaluated settings. Additional qualitative comparisons of predicted trajectories are provided in App. A.7 and Fig. 5.

Impact of Occupancy Quality. Table 5 compares two separately trained TCOR models that differ in their current-frame occupancy input: predictions from a frozen FBOCC model or ground-truth occupancy. Using ground-truth occupancy reduces the average nuScenes L2 error from 0.36 to 0.29 m. The errors at 1, 2, and 3 s decrease from 0.16/0.32/0.59 m to 0.15/0.28/0.44 m, respectively. The largest reduction occurs at 3 s, where the error decreases by 25.4%. These results indicate that OccWAM has not yet reached its performance ceiling: improving the accuracy of current-frame occupancy can further enhance its trajectory prediction performance.

Effectiveness of TCOR. Table 6 compares three occupancy-token selection strategies on nuScenes under the same training protocol. Full OCC processes all $25 \times 25 = 625$ tokens per frame.



(a) Trajectory Anchor Selector (b) Local Occupancy Routing

Figure 3: Qualitative visualization of TCOR components. (a) Trajectory Anchor Selector: dashed lines are the top-3 anchors selected from the bank, solid lines are their residual-refined results, and black is the ground-truth trajectory. (b) Local Occupancy Routing: as future waypoints evolve, the selected occupancy regions tightly follow the planned corridor, while background occupancy is suppressed.

The random variant selects 64 tokens per frame without a learned selection rule, while TCOR uses the predicted trajectory anchors to select 64 tokens near prospective ego motion. Both 64-token variants are trained under the same protocol as the Full OCC model. Random selection increases the average planning L2 error from 0.35 to 0.45 m, showing that reducing the token count without regard to spatial relevance degrades accuracy. TCOR achieves an average L2 error of 0.36 m, only 0.01 m above Full OCC, while reducing the number of processed occupancy tokens by 89.8%. The measured latency decreases from 4701 to 2841 ms, a 39.6% reduction. These results show that trajectory-guided selection retains most of the planning performance of Full OCC while reducing occupancy processing and measured inference time.

To analyze the two components of TCOR, Trajectory Anchor Selector and Local Occupancy Routing, we first visualize the anchor prediction results and anchor offset results of the Trajectory Anchor Selector, as shown in Fig. 3(a). The figure illustrates how the Trajectory Anchor Selector predicts and refines candidate anchors. The blue, orange, and green dashed lines denote the top three candidate trajectories classified and selected by the Trajectory Anchor Selector from the candidate anchors. These trajectories still deviate to some extent from the ground-truth trajectory shown in black. Therefore, the Trajectory Anchor Selector predicts anchor offsets to refine the three candidate trajectories, yielding the blue, orange, and green solid trajectories. The refined trajectories are visibly closer to the ground-truth trajectory, indicating that the Trajectory Anchor Selector improves the accuracy of the candidate trajectories. Furthermore, we visualize the OCC tokens selected by Local Occupancy Routing, as shown in Fig. 3(b). This module selects occupancy regions around the three refined trajectories and the predicted waypoints. As the waypoints evolve over future time steps, the selected regions move accordingly and remain concentrated around the corresponding waypoint at each time step. This enables the action representation to interact with occupancy information relevant to that time step. By reducing attention to irrelevant background regions and lowering computational cost, these results demonstrate the effectiveness of Local Occupancy Routing.

5 Conclusion

We presented OccWAM, a 4D occupancy-augmented World-Action Model for autonomous driving. By combining visual world modeling with explicit occupancy prediction, OccWAM provides complementary information for ego-trajectory generation. Experiments on NAVSIM and nuScenes demonstrate strong planning performance, including a low collision rate on nuScenes. Moreover, TCOR reduces the number of processed occupancy tokens by approximately 90% while maintaining accuracy close to the full-token variant. These findings support the value of incorporating spatial structure into world-action modeling and focusing model interactions on regions relevant to future ego motion.

AI USE STATEMENT

We used generative AI tools to assist with searching for and identifying relevant literature and to improve the grammar, clarity, and readability of the final manuscript. The authors examined the original sources and verified the references rather than relying on AI-generated summaries or citations. The research method, experimental implementation, data organization, analysis of experimental results, and initial manuscript were completed by the authors. All AI-assisted edits were reviewed by the authors to ensure that they preserved the intended technical meaning. The authors take full responsibility for the accuracy of the reported results, the claims, and the final content of this paper.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.01164. URL http://dx.doi.org/10.1109/cvpr42600.2020.01164.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking, 2024. URL https://arxiv.org/abs/2406.15349.

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, 2024. URL https://arxiv.org/abs/2410.10429.

Jianhua Han, Meng Tian, Jiangtong Zhu, Fan He, Huixin Zhang, Sitong Guo, Dechang Zhu, Hao Tang, Pei Xu, Yuze Guo, Minzhe Niu, Haojie Zhu, Qichao Dong, Xuechao Yan, Siyuan Dong, Lu Hou, Qingqiu Huang, Xiaosong Jia, and Hang Xu. Percept-wam: Perception-enhanced world-awareness-action model for robust end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2511.19221.

Yikang Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xihan Zhu, Zhenyu Chai, Xianming Du, Dengjie Lin, Kai Wang, Yuhua Shi, Yun Wang, Yu Qiao, Lizhao Wang, Jiang Liu, Yu Guo, Hang Yan, Xingang Wang, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Nicklas Hansen Ignat Georgiev, Varun Giridha and Animesh Garg. Pwm: Policy learning with large world models, 2024.

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. *ICCV*, 2023.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving, 2026. URL https://arxiv.org/abs/2601.05640.

Xiang Li, Pengfei Li, Yupeng Zheng, Wei Sun, Yan Wang, and Yilun Chen. Semi-supervised vision-centric 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2502.07309*, 2025a.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025b.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models, 2025. URL https://arxiv.org/abs/2411.04996.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. Drivewam: Video generative priors enable scalable world-action modeling for autonomous driving. *arXiv preprint arXiv:2605.28544*, 2026.

Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. 2023.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. Apr 2023.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models, 2024. URL https://arxiv.org/abs/2402.12289.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 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, 2024. URL https://arxiv.org/abs/2409.03272.

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 *NeurIPS*, 2022.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Hangjun Ye, Wenyu Liu, et al. Drivelaw: Unifying planning and video generation in a latent driving world. *arXiv preprint arXiv:2512.23421*, 2025.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving, 2025. URL https://arxiv.org/abs/2512.19133.

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. *arXiv preprint arXiv:2408.14197*, 2024.

Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *NeurIPS*, 2025.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. Simwam: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2311.16038*, 2023.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model, 2025a. URL https://arxiv.org/abs/2503.23463.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, and Steven L. Waslander. Drivedreamer-policy: A geometry-grounded world-action model for unified generation and planning, 2026. URL https://arxiv.org/abs/2604.01765.

Zewei Zhou, Tianhui Cai, Yun Zhao, Seth Z.and Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

A APPENDIX

A.1 MIXTURE-OF-TRANSFORMERS ATTENTION MASK

At each MoT layer, the modality-specific queries, keys, and values are projected to a shared attention dimension. For expert $m \in \{v, o, a\}$, the attention output is

$$U_m = \mathrm{Softmax}\left(\frac{Q_m K_{\mathrm{all}}^{\top}}{\sqrt{d}} + M_m\right) V_{\mathrm{all}}, \tag{9}$$

where K_{all} and V_{all} concatenate the keys and values of the three experts, and M_m is an additive attention mask. In the TCOR variant, the occupancy part of K_{all} and V_{all} contains only the selected occupancy tokens.

An entry of M_m is 0 for an allowed query–key pair and $-\infty$ otherwise. Within each modality, tokens attend bidirectionally. Across modalities, Video queries attend only to Video keys; Occupancy queries may attend to Video, Occupancy, and Action keys; and Action queries may attend to all three modalities. For temporal visibility, clean first-frame conditioning tokens cannot attend to any future tokens. Future tokens may attend to the clean first frame and bidirectionally to tokens at all future timestamps. Thus, future frames are jointly denoised rather than generated autoregressively frame by frame.

A.2 TRAJECTORY ANCHOR BANK CONSTRUCTION

Each trajectory is a sequence of 6 future 2D positions in the current ego frame. Command labels are derived from the lateral coordinate of the last valid position: left turn ($y_H \geq 2\,\mathrm{m}$), right turn ($y_H \leq -2\,\mathrm{m}$), and straight (otherwise); trajectories without a valid future point are assigned to the straight group. Within each command, anchors are built by offline K-means clustering: each trajectory is flattened into a 12-dimensional vector and assigned by

$$D_{\mathrm{cluster}}(Y, A_k) = \sum_{h=1}^{H} \|y_h - a_{k,h}\|_2^2,$$

using K-means++ initialization, at most 300 iterations, convergence when the largest center shift is below 10^{-5}, and random seeds $0 +$ command_id; only trajectories with all six steps valid are collected, and validation data are used solely for oracle metrics, never for fitting centers. Each command yields $M{=}12$ anchors, giving 36 prototypes. Anchor coordinates are stored as fixed buffers and never updated by gradients.

A.3 SELECTOR NETWORK AND SPATIAL AGGREGATION

The selector encodes current-frame occupancy ($\mathrm{LN}(1024) \to \mathrm{Linear}(1024, 256) \to \mathrm{SiLU}$), video, ego state ($\mathrm{LN}(4096) \to \mathrm{Linear}(4096, 256) \to \mathrm{SiLU}$; raw 13-dim ego input via $\mathrm{Linear}(13, 4096)$), and text features (masked mean pooling of valid text tokens) into a shared 256-dim space; anchor coordinates are encoded by $\mathrm{Linear}(2, 256) \to \mathrm{SiLU} \to \mathrm{Linear}(256, 256)$. The spatial aggregation kernel has width $\sigma_r{=}4.8\,\mathrm{m}$ and is normalized over the full 625 occupancy grids (no radius truncation); weights are precomputed from the original anchors and are not resampled from predicted offsets. The global condition is

$$g = \mathrm{MLP}_{\mathrm{cond}}\big([f_{\mathrm{ego}}, f_{\mathrm{text}}, \mathrm{Mean}(f_{\mathrm{occ}}), \mathrm{Mean}(f_{\mathrm{video}})]\big),$$

and the per-point candidate feature is

$$u_{k,h} = \mathrm{MLP}_{\mathrm{cand}}\big(r_{k,h} + \mathrm{MLP}_{\mathrm{anchor}}(a_{k,h}) + g\big).$$

The score head pools the six candidate features and the offset head predicts a bounded displacement with a zero-initialized last layer.

A.4 Selector Supervision

Let $\mathcal{K}_c$ be the anchors allowed by the current command. Classification uses ADE-based soft labels with temperature τ=1.0,

$$q_k = \frac{\exp(-d_k/\tau)}{\sum_{j\in\mathcal{K}_c}\exp(-d_j/\tau)}, \quad \mathcal{L}_{\mathrm{cls}} = -\sum_{k\in\mathcal{K}_c} q_k \log p_k,$$

where $d_k = \frac{1}{H}\sum_h \|a_{k,h} - y_h\|_2$ is the ADE to the original anchor; disallowed commands are masked (distance 10^4, logit -10^4). The nearest anchor is thus not a unique positive class but the one with the largest target probability. Regression matches the original anchor with the smallest ADE,

$$k^* = \arg\min_{k\in\mathcal{K}_c} d_k, \qquad \mathcal{L}_{\mathrm{reg}} = \frac{1}{12}\sum_{h=1}^{6}\sum_{d\in\{x,y\}} \mathrm{SmoothL1}_{\beta=1}\big(a_{k^*,h,d} + \Delta a_{k^*,h,d} - y_{h,d}\big),$$

which equivalently supervises the residual $Y - A_{k^*}$; regression is not restricted to whether k^* is among the selected top-3. Both losses are computed on all six future steps (the implementation does not mask invalid future steps).

A.5 Local Occupancy Routing Details

All waypoint coordinates and occupancy-token centers used for routing are expressed in the current ego coordinate frame. For the current frame, each trajectory corridor is the polyline connecting the ego origin and the waypoints of a refined anchor. The function $\mathrm{dist}(p, \mathcal{C})$ returns the minimum ground-plane distance from p to that polyline. For future frame h, the distance is computed to the closest of the three refined waypoints at that frame. Taking the minimum over hypotheses assigns one routing distance to each occupancy token; the 64 smallest distances determine $\mathcal{I}_h$. No additional union or duplicate-removal step is required.

The selected current-frame occupancy tokens are kept clean as conditioning inputs. At each future frame, the Occupancy Expert processes and denoises only the tokens indexed by $\mathcal{I}_h$, and the occupancy flow-matching loss is computed only on these tokens. Tokens outside the selected set do not participate in subsequent OCC processing or receive occupancy supervision in the TCOR variant. The discrete routing indices are detached, so the loss propagates through gathered feature values but not through the index-selection operation. The selector is supervised by the classification and regression objectives in App. A.4.

A.6 Implementation Details

We independently train OccWAM on NAVSIM and nuScenes for 100 epochs, using a total batch size of 8 on eight NVIDIA A800 GPUs. We adopt the AdamW optimizer with an initial learning rate of 2×10^{-4}, a weight decay of 0.01, and a cosine annealing learning-rate schedule with 1000 warm-up steps. The loss weights for the Video, OCC, and Action Experts are all set to 1.0. TCOR predicts the top three candidate trajectory anchors and retains 64 occupancy tokens for interaction with the Action Expert. All ablation variants use the same training data, parameter initialization, optimization strategy, and inference configuration to ensure a fair comparison.

A.7 Additional Results

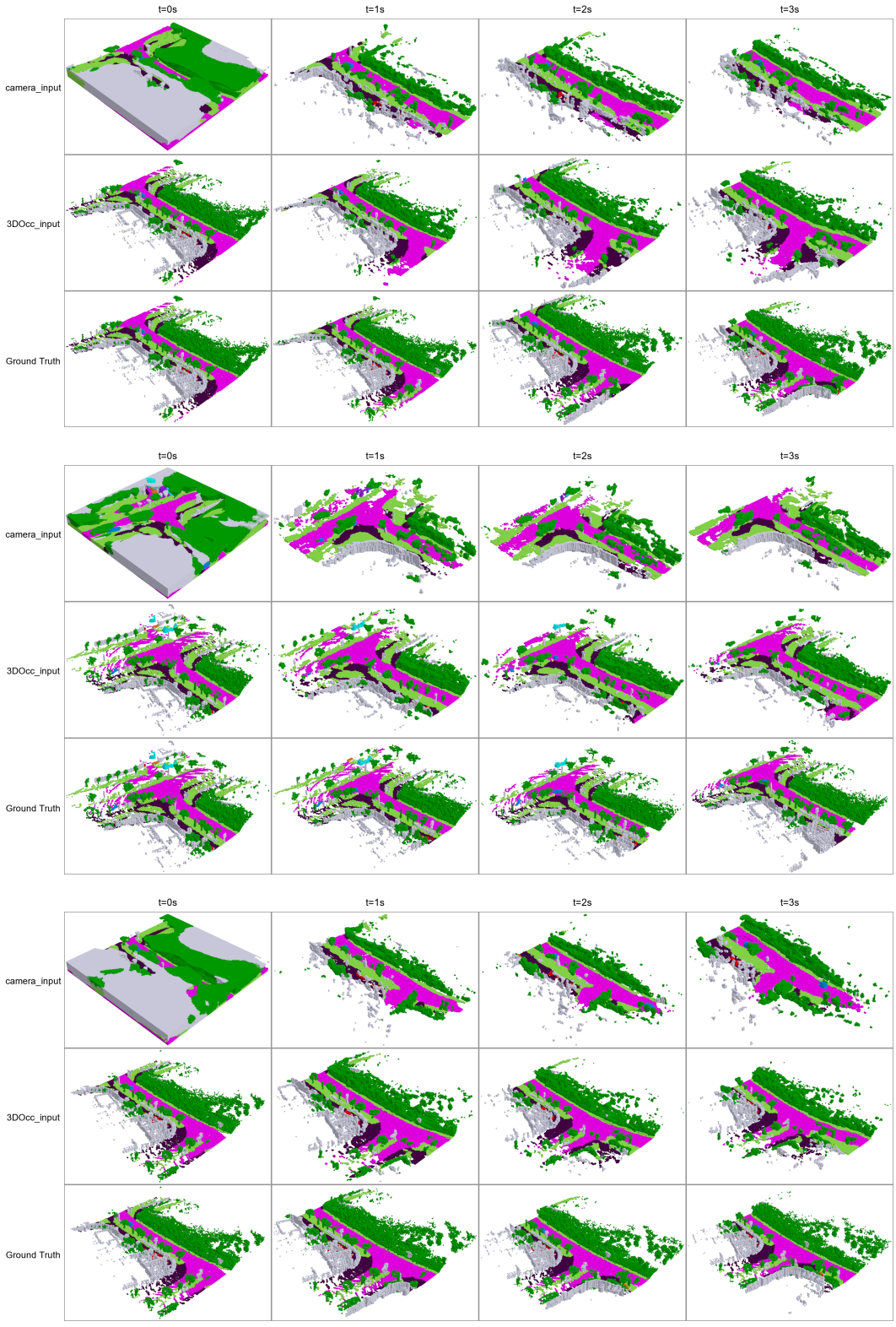


Figure 4: Visualization of 4D occupancy forecasting on nuScenes. Columns correspond to timesteps, from the current frame to future 1/2/3 s; colors denote semantic classes.

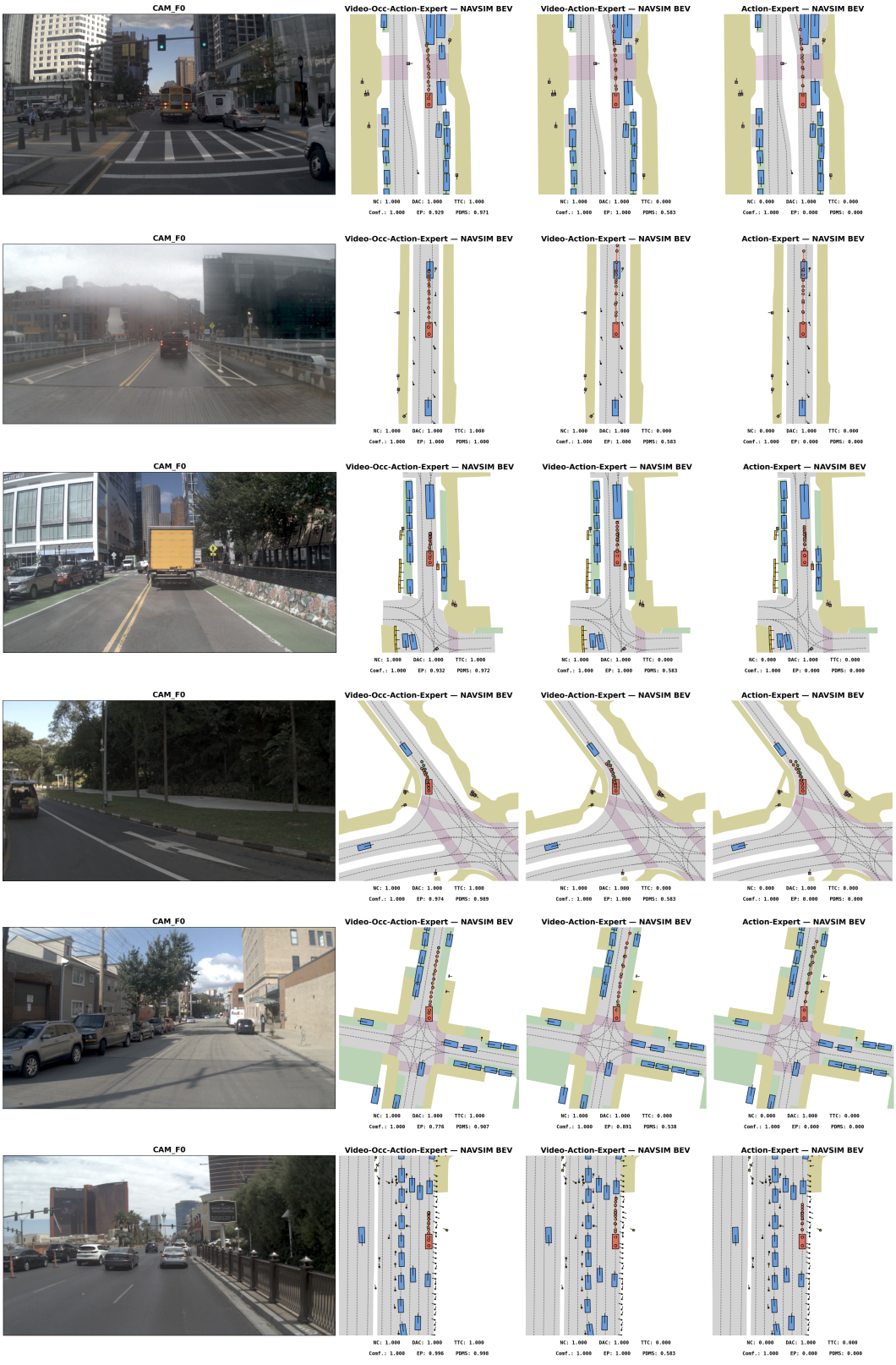


Figure 5: Visual comparison of trajectory prediction on NAVSIM. The three columns are the full OccWAM (video–occupancy–action), the video–action variant, and the action-only variant. Red lines are predicted trajectories and green lines are ground-truth trajectories.