Aligned Delta-Triplane Transformers as Occupancy World Models

Haoran Xu[1], **Peixi Peng**[1,3]*, **Shijun Guo**[4], **Guang Tan**[2]*,
Yiqian Chang[5], **Yisen Zhao**[1], **Shuaixian Wang**[6], and **Yonghong Tian**[1,3]

[1]Peking University [2]Shenzhen Campus of Sun Yat-sen University
[3]Peng Cheng Laboratory [4]Xidian University
[5]Shenzhen Campus of Harbin Institute of Technology
[6]Shenzhen Polytechnic University
{xuhr.sz, pxpeng}@pku.edu.cn tanguang@mail.sysu.edu.cn

Abstract

Occupancy World Models (OWMs) aim to predict future 3D occupancy scenes from historical observations and future ego motions, providing a useful world simulation tool for autonomous driving. Existing methods usually rely on large networks to implicitly align multi-frame historical states and predict future occupancy in a full-state manner, which is costly and redundant because most scene regions remain unchanged over short time intervals. In this paper, we propose *Aligned Delta-Triplane Transformer* (ADTT), a compact 4D OWM that explicitly aligns historical triplanes with ego-motion compensation and predicts only future scene changes. Based on the aligned triplane prior, a query-conditioned Transformer predicts residual changes, where ego-motion queries guide controllable future motion and learnable external queries capture scene changes beyond ego motion. The predicted changes are added to the aligned prior and decoded into future occupancy scenes. Experiments on Occ3D-nus show that ADTT achieves state-of-the-art forecasting performance with fewer parameters and higher FPS. Our code is publicly available online: `https://github.com/kyoran/ADTT`.

1 Introduction

Recently, Occupancy World Models (OWMs) [49, 36, 39, 45, 9, 18, 29, 23, 47] have emerged as a promising framework for modeling dynamic 3D scenes. Instead of representing the environment with 2D images [37] or bird's-eye-view (BEV) features [21], OWMs describe real-world scenes with structured 3D semantic voxels and predict how these scenes evolve over time under different ego actions. This makes OWMs a useful tool for simulating future physical worlds, and provides a scalable way to generate training data, including rare corner cases, for autonomous driving and indoor navigation.

In designing OWMs, there are two tasks need to be solved: (i) removing historical ego-motion

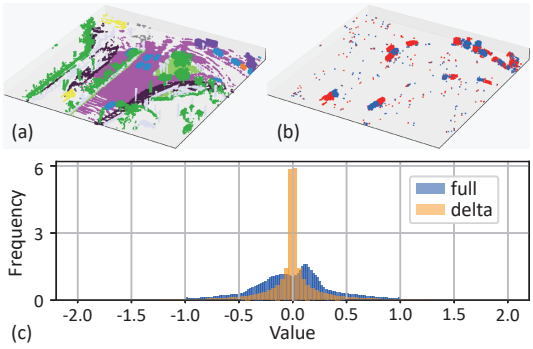

Figure 1: (a-b) Comparison between the full occupancy and occupancy changes (Δ) of a scene. In the change map, red indicates newly appeared voxels, while blue denotes disappeared voxels. (c) Distribution of latent-space feature values for full occupancy versus occupancy changes.

*Corresponding authors.

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

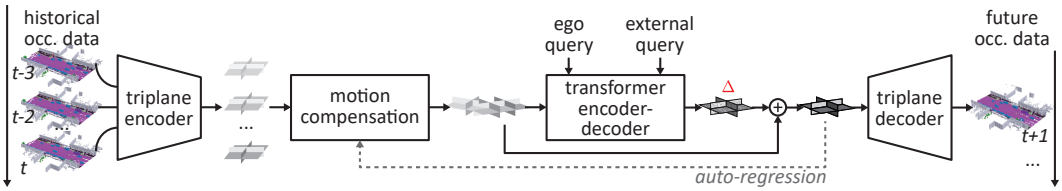

Figure 2: ADTT follows a simple yet effective procedure: it first represents occupancy scenes with pretrained triplanes, then explicitly aligns historical triplanes through historical ego-motion compensation, and finally employs a Transformer to predict triplane changes (Δ) based on the aligned triplane features, future ego-motion query, and external learnable query.

effects to better model surrounding object motion; and (ii) using future ego motions to generate controllable future occupancy scenes.

For the first task, the observed occupancy changes usually come from two sources: ego-motion and the motion of surrounding objects. If the scene variation introduced by ego-motion can be removed, the remaining changes mainly come from moving objects, making future prediction much easier. This process is usually referred to as motion compensation. Most existing OWMs perform this step implicitly. They first compress high-dimensional 3D occupancy data into latent xy-plane features, and then rely on neural networks to learn the alignment across multiple frames. Such an implicit formulation often requires large model capacity to fit the geometric transformation between frames. Recent works such as I^2-World [23] and GenieDrive [47] introduce dedicated ego-motion-related attention mechanisms to ease the learning of temporal correspondence. However, their alignment is mainly learned through network attention rather than explicit 3D geometric transformation.

We provide an intuitive example in Figure 1. Figure 1(a) shows the full occupancy state in Occ3D-nus benchmark [2, 33], while Figure 1(b) shows the occupancy changes between two adjacent frames after compensating the previous frame to the next timestep. It is clear that occupancy changes are inherently sparse and key elements of interest are more distinct in the change map. We further compare their latent-space distributions in Figure 1(c), where the x axis denotes a scalar feature value. Full occupancy states exhibit a scattered distribution, while occupancy change states are much more tightly concentrated around zero, which reduces variance and is easier to learn.

For the second task, future ego-motion should be used as a condition to guide scene prediction. Existing OWMs typically feed ego-motion signals into large Transformers or diffusion models to directly predict the full future occupancy state [45, 39]. However, as suggested by Figure 1, full-state prediction is often redundant, since most regions remain unchanged after motion compensation. This has motivated a compensation-then-prediction paradigm, where the model first removes ego-motion effects and then predicts future occupancy changes. Although this reduces the prediction burden, current methods [23, 47] mainly use future ego-motion as the only condition for prediction. Such signals specify how the ego coordinate frame moves and which areas may enter the future view, but they do not provide enough guidance for the uncertain content that should appear in the future scene. As a result, the predicted future scenes may have limited temporal consistency.

Based on these observations, we propose *Aligned Delta-Triplane Transformer* (ADTT), a compact 4D OWM that explicitly aligns historical triplanes and predicts sparse future changes in the triplane latent space, as shown in Figure 2. Specifically, we first pretrain a triplane encoder-decoder to obtain compact and geometry-aware occupancy representations. Then, using ego poses recorded at different timesteps, we perform ego-motion compensation to warp historical triplanes into the target prediction frame. This gives the predictor a stable geometric prior, so that it only needs to model residual changes that cannot be explained by ego motion. To predict these residuals, we design a query-conditioned Transformer. Ego-motion queries inject deterministic future motion guidance, while learnable external queries provide additional flexibility for modeling scene changes that are not explicitly controlled by ego motion. The predicted delta triplanes are added to the motion-aligned prior and decoded into future occupancy scenes. In summary, our contributions are three-fold:

• We propose ADTT, a compact 4D autoregressive OWM that aligns historical triplanes with ego-motion compensation and forecasts sparse future scene changes.

• Within ADTT, we introduce two key modules: triplane motion compensation, which aligns historical triplanes into the target frame, and query-conditioned delta prediction, which predicts residual changes from the aligned triplane using ego-motion queries for controllable future guidance and learnable external queries for scene dynamics beyond ego motion.

• Extensive experiments on Occ3D-nus [2, 33] validate the state-of-the-art (SOTA) performance of ADTT in short- and long-term forecasting, cross-dataset generalization, and real-time inference.

2 Related Works

3D occupancy reconstruction. 3D occupancy reconstruction methods [34, 33, 40, 24, 27] aim to represent the environment as a 3D voxel grid where each voxel encodes both geometric and semantic information. The field was pioneered by SSCNet [31] for indoor scenes using depth sensors, and later extended to outdoor camera-based settings by MonoScene [3]. These approaches primarily utilize multi-camera RGB images [20, 12] or LiDAR point clouds [4, 42] to infer voxel-wise occupancy and semantics in an agent-centric coordinate system. A core challenge lies in building accurate correlations between raw sensor data and 3D voxel space. Once an accurate occupancy reconstruction is achieved, further learning of the temporal dynamics of scene changes becomes necessary [19].

4D occupancy prediction. To anticipate future scene changes, various 4D occupancy prediction methods have been proposed to capture the temporal dynamics of scene evolution. Some [25, 28, 16] predict future sensor-level data, which is subsequently voxelized into occupancy results, while others [26, 6, 44, 43] directly forecast occupancy outcomes from historical observations. These methods mainly aim to reduce spatio-temporal prediction errors, but they generally do not consider controllable forecasting conditioned on ego motion.

World models for autonomous driving. World models [11, 10] aim to compress high-dimensional scene representations to capture the temporal dynamics of scene transitions, facilitating both future scene predictions and motion planning for the agent. In autonomous driving, existing models [15, 48] typically map surrounding traffic participants to the BEV perspective to predict instance-level tracklets or directly use diffusion models [41, 38, 37, 8, 14] to generate pixel-level future driving views. Recent world models [39, 45, 49, 46, 9] further exploit structured 3D occupancy data for future scene prediction. However, most of them still compress occupancy scenes into BEV latent features and rely on Transformers or diffusion models to implicitly align multi-frame historical context, which often requires large model capacity to forecast the full future scene. In contrast, our method explicitly aligns historical features and predicts sparse scene changes, both in the triplane latent space [47, 12].

3 Methodology

3.1 Problem Formulation

Let $S^t \in \mathbb{R}^{H\times W\times L}$ denote the semantic occupancy scene at timestep t in the ego-centric coordinate frame, where H, W, and L denote the height, width, and length of the voxel space. Given a sequence of historical occupancy observations and ego motions, the model uses them as predefined scene context and predicts future occupancy scenes under given future ego motions.

For clarity, we denote the historical occupancy context as

$$\mathcal{S}^t_{\text{hist}} = \{S^{t-\tau_p+1}, \cdots, S^t\}, \tag{1}$$

and the ego motions associated with the observed and future intervals as

$$\mathcal{M}^t_{\text{hist}} = \{m^{t-\tau_p+2}, \cdots, m^t\}, \quad \mathcal{M}^t_{\text{fut}} = \{m^{t+1}, \cdots, m^{t+\tau_f}\}. \tag{2}$$

Here, each ego motion m^i describes the ego transformation from timestep $i-1$ to timestep i, such as translation and rotation. The historical occupancy sequence provides scene context, including static structures, visible objects, and recent scene evolution. The historical ego motions are used to align observations captured under different ego-centric frames, while future ego motions serve as conditions for controllable future generation.

Formally, the OWM learns a mapping:

$$\hat{S}^{t+1:t+\tau_f} = \mathbf{\Phi}\left(\mathcal{S}^t_{\text{hist}}, \mathcal{M}^t_{\text{hist}}, \mathcal{M}^t_{\text{fut}}\right), \tag{3}$$

where $\hat{S}^{t+1:t+\tau_f}$ denotes the predicted future occupancy sequence.

Accordingly, as shown in Figure 3, we propose *Aligned Delta-Triplane Transformer* (ADTT), which aligns historical states and predicts sparse future changes in the aligned triplane latent space.

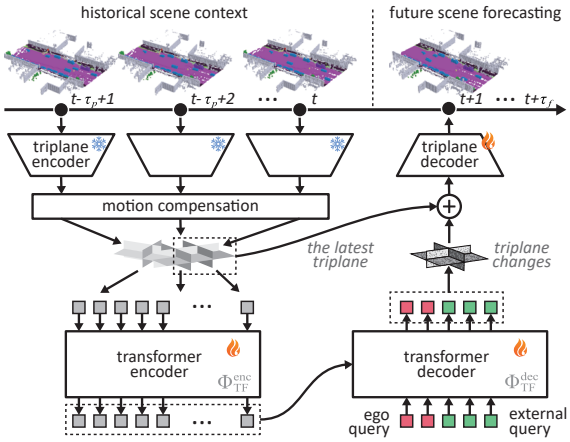

Figure 3: ADTT performs motion compensation and tokenization in the triplane latent space. The Transformer encoder then encodes historical context from triplane tokens, while the decoder predicts next-frame triplane changes using deterministic ego-motion queries and learnable external queries. These predicted changes are added to the latest triplane and finally decoded into future occupancy states.

3.2 Ego-Motion-Aligned Triplane Representation

Recent occupancy-related works [47, 12] have gradually shifted their latent space from BEV features to triplane representations, since triplanes provide a compact latent space while preserving more explicit spatial structure. Following this trend, we build our OWM on a pretrained triplane latent space. Different from prior methods that mainly use triplanes as compact scene encodings, we further exploit their geometric structure for ego-motion compensation, explicitly aligning historical triplanes across ego-centric frames before predicting sparse future changes.

As shown in Figure 4, the occupancy scene S^t is encoded by $\mathbf{\Phi}^{\text{enc}}_{\text{Tri}}$ and compressed into three orthogonal feature planes:

$$s^t = \{s^t_{xy}, s^t_{xz}, s^t_{yz}\}, \tag{4}$$

where

$$s^t_{xy} \in \mathbb{R}^{C\times H_{xy}\times W_{xy}}, \quad s^t_{xz} \in \mathbb{R}^{C\times H_{xz}\times W_{xz}}, \quad s^t_{yz} \in \mathbb{R}^{C\times H_{yz}\times W_{yz}}. \tag{5}$$

The three planes provide complementary projections of the 3D scene. To reconstruct the occupancy scene, each 3D point $pos = (x, y, z)$ queries its corresponding features from the xy, xz, and yz planes. The queried features are fused with positional encoding $\text{PE}(pos)$ and passed into $\mathbf{\Phi}^{\text{dec}}_{\text{Tri}}$ to predict the semantic occupancy label.

We pretrain $\mathbf{\Phi}^{\text{enc}}_{\text{Tri}}$ and $\mathbf{\Phi}^{\text{dec}}_{\text{Tri}}$ with occupancy reconstruction:

$$J_{\text{enc,dec}} = \mathbb{E}_{t\sim\mathcal{T}}\left[\mathcal{L}_{occ}\left(\mathbf{\Phi}^{\text{dec}}_{\text{Tri}}(\mathbf{\Phi}^{\text{enc}}_{\text{Tri}}(S^t)), S^t\right)\right], \tag{6}$$

where $\mathcal{T}$ denotes the set of timesteps in the training data, and $\mathcal{L}_{occ} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{lz}$ combines cross-entropy and Lovasz-softmax losses [1, 17]. After pretraining, $\mathbf{\Phi}_{\text{enc}}$ provides compact and geometry-aware triplane states for future prediction.

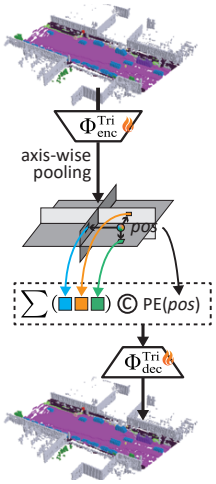

Figure 4: Pretrained triplane for OWM.

We then perform ego-motion compensation using the known ego poses at different timesteps. These poses describe the trajectory of the ego vehicle and are recorded by the vehicle localization system, such as odometry and IMU [2]. Inspired by spatial transformation and differentiable warping [13, 50], we design an ego-motion compensation operation in the triplane latent space.

For each timestep k, we denote its ego pose in the global coordinate frame as $(\mathbf{R}_k, \mathbf{T}_k)$, where $\mathbf{R}_k$ and $\mathbf{T}_k$ transform a point from the ego coordinate frame at timestep k to the global coordinate frame:

$$\mathbf{p}^k_{\text{global}} = \mathbf{R}_k\mathbf{p}^k_{\text{local}} + \mathbf{T}_k. \tag{7}$$

Here, $\mathbf{p}^k_{\text{local}}$ denotes a point in the ego coordinate frame at timestep k, and $\mathbf{p}^k_{\text{global}}$ denotes the same point in the global coordinate frame.

Given a historical timestep i and a target prediction timestep j, our goal is to align the historical triplane s^i to the target ego coordinate frame at timestep j. For each location on the target plane, we

first regard it as a 3D point $\mathbf{p}^j_{\text{local}}$ in the target ego coordinate frame, and then find where this point should sample from in the historical triplane. Specifically, we first map $\mathbf{p}^j_{\text{local}}$ to the global coordinate frame using the target ego pose, and then map it back to the historical ego coordinate frame:

$$\mathbf{p}^i_{\text{local}} = \mathbf{R}_i^\top \left(\mathbf{R}_j \mathbf{p}^j_{\text{local}} + \mathbf{T}_j - \mathbf{T}_i \right). \tag{8}$$

This transformation gives the corresponding source-frame location for each target-frame point. In practice, if the projected location lies inside the valid range of the historical plane, we copy the nearest source feature to the target frame. Otherwise, the location is filled with zeros, indicating that this region has no valid historical correspondence and should be inferred by the prediction module.

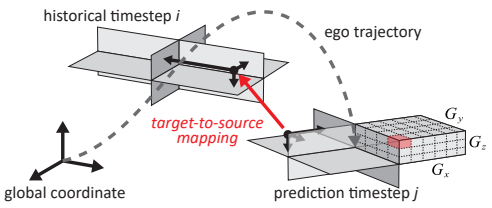

Figure 5: Motion compensation in the triplane latent space.

We denote the motion-compensated triplane as:

$$\bar{s}^j = \{\bar{s}^j_{xy}, \bar{s}^j_{xz}, \bar{s}^j_{yz}\}, \tag{9}$$

which provides an aligned geometric prior for predicting the target scene at timestep j.

3.3 Query-Conditioned Delta Prediction

Given the motion-aligned triplane representation, ADTT predicts the triplane changes Δs_{t+1} with a Transformer encoder-decoder [35, 5]. The encoder builds a spatio-temporal memory from historical triplanes, while the decoder predicts future state changes using ego queries and external queries.

Triplane tokenization. We first convert the motion-compensated triplane $\bar{s}^j$ into a low-resolution 3D latent token grid, allowing the Transformer to model the scene in a unified and compact spatial structure. Specifically, each plane is projected and downsampled by a lightweight 2D convolutional tokenizer, denoted as ϕ_{xy}, ϕ_{xz}, and ϕ_{yz}, with D_{down} output channels. As shown in Figure 5, the three plane features are then broadcast to the same 3D grid of size $G_x \times G_y \times G_z$ and fused. For each 3D grid location pos, the fused feature is computed as:

$$v^t(pos) = \text{Concat}_{uv \in \{xy,xz,yz\}} \left[\text{Broad}_{uv} \left(\phi_{uv}(\bar{s}^t_{uv}) \right) (pos) \right], \tag{10}$$

where $\text{Broad}_{uv}(\cdot)$ denotes broadcasting a plane feature to the 3D latent grid along its missing axis.

The fused feature is then projected to the Transformer token dimension and combined with 3D sinusoidal positional encoding:

$$z^j(pos) = \text{LN}\left(\text{FC}(v^j(pos))\right) + \text{PE}(pos), \tag{11}$$

where $\text{FC}(\cdot)$ denotes a fully connected layer that maps the feature to D_{token} dimensions, $\text{LN}(\cdot)$ denotes layer normalization, and $\text{PE}(pos)$ is the 3D spatial positional encoding.

Historical memory encoding. For each prediction step, we use the most recent τ_p motion-compensated triplane states as temporal context. After tokenization, the historical token sequences are concatenated along the temporal dimension:

$$Z^t_{\text{hist}} = [z^{t-\tau_p+1}; \cdots; z^t]. \tag{12}$$

We add a learnable temporal embedding to distinguish different history frames and feed the sequence into a Transformer encoder:

$$\mathcal{M}^t = \mathbf{\Phi}^{\text{enc}}_{\text{TF}} \left(Z^t_{\text{hist}} + e_{\text{time}} \right), \tag{13}$$

where $\mathcal{M}^t$ denotes the encoded spatio-temporal memory. Through self-attention, the encoder allows tokens from different spatial locations and historical timesteps to interact, producing a memory bank that summarizes recent scene context.

Ego-motion and external queries. The decoder queries serve as conditions for future scene prediction. Different from methods that only rely on ego motion, we construct the decoder query from two complementary sources. First, the future ego motion m^{t+1} is encoded by an MLP and expanded into N_{ego} ego-motion queries:

$$Q^{\text{ego}}_{t+1} = f_{\text{ego}}(m^{t+1}) \in \mathbb{R}^{N_{\text{ego}} \times D_{\text{token}}}. \tag{14}$$

These queries inject explicit information about the future ego transformation. Second, we introduce N_{ext} learnable external queries $Q^{\text{ext}} \in \mathbb{R}^{N_{\text{ext}} \times D_{\text{token}}}$. Unlike ego-motion queries, these queries are not tied to a specific motion input. They provide additional learnable slots for modeling residual scene changes that are not fully determined by ego motion, such as local appearance changes, occlusion variations, and newly observed regions.

The final decoder query is obtained by concatenating the two query groups and adding a learnable future-step embedding:

$$Q_{t+1} = [Q_{t+1}^{\text{ego}}; Q^{\text{ext}}] + e_{t+1}^{\text{step}}. \tag{15}$$

The total number of queries is set to match the number of target latent tokens, where $N = N_{\text{ego}} + N_{\text{ext}} = G_x G_y G_z$. The Transformer decoder attends to the encoded memory and predicts:

$$\Delta Z^{t+1} = \mathbf{\Phi}_{\text{TF}}^{\text{dec}} \left(Q_{t+1}, \mathcal{M}^t \right), \tag{16}$$

where $\Delta Z^{t+1} \in \mathbb{R}^{N \times D_{\text{token}}}$ denotes the predicted delta token sequence.

The predicted delta tokens are reshaped back to the predefined 3D latent grid and then upsampled $\text{Up}(\cdot)$ to the triplane resolution. We denote this operation as:

$$\Delta s^{t+1} = \text{Up}\left(\text{Reshape}(\Delta Z^{t+1}) \right), \tag{17}$$

where $\text{Up}(\cdot)$ includes 3D feature refinement, interpolation, and projection back to the three triplane feature maps $\Delta s^{t+1} = \{\Delta s_{xy}^{t+1}, \Delta s_{xz}^{t+1}, \Delta s_{yz}^{t+1}\}$. The predicted delta triplane is added to the latest motion-aligned triplane prior to obtain the updated triplane:

$$\hat{s}^{t+1} = \overline{s}^t + \Delta s^{t+1}. \tag{18}$$

Finally, the updated triplane is decoded into the future occupancy scene:

$$\hat{S}^{t+1} = \mathbf{\Phi}_{\text{Tri}}^{\text{dec}} \left(\hat{s}^{t+1} \right). \tag{19}$$

The future prediction module is trained with both latent-space and occupancy-space supervision:

$$J_{\text{fut}} = \mathbb{E}_{t,k,uv} \left[\mathcal{L}_{\text{tri}} \left(\hat{s}_{t+k}^{uv}, s_{t+k}^{uv} \right) + \xi \mathcal{L}_{\text{occ}} \left(\hat{S}^{t+k}, S^{t+k} \right) \right], \tag{20}$$

where $\mathcal{L}_{\text{tri}}$ is the sum of L1 and L2 regression losses, and ξ balances triplane- and occupancy-level supervision. During training, we fix the pretrained encoder to keep the target triplanes stable and fine-tune the decoder to better reconstruct occupancy scenes from predicted latent states.

4 Experiments

4.1 Experimental settings

Dataset and evaluation metrics. We evaluate occupancy forecasting on the widely used nuScenes dataset [2] adopted by OWM methods [49, 36, 39, 45, 9, 18, 29, 23, 47] with occupancy annotations from Occ3D [33], referred to as Occ3D-nus. The dataset contains 1,000 scenes, with 700 scenes for training and 100 scenes for testing. For the main evaluation, we use the past four frames (2 seconds) to predict the next six frames (3 seconds), which measures short-term forecasting performance. We further evaluate two types of zero-shot inference ability without additional training. First, following GenieDrive [47], we directly evaluate the model trained on the 3-second setting for longer forecasting horizons. Second, following I^2-World [23], we use Occ3D-Waymo [32, 33] to evaluate cross-dataset generalization on 202 validation scenes. For all tasks, we report intersection over union (IoU) for occupied and unoccupied voxels, as well as mean IoU (mIoU) over 18 semantic classes.

Implementation details. At each timestep, the occupancy data S^t has a resolution of $16 \times 200 \times 200$. It covers a spatial range of $[-40, 40]$ meters along the X and Y axes and $[-1, 5.4]$ meters along the Z axis, with a voxel size of 0.4 meters. The pretrained triplane representation has a resolution of $16 \times 100 \times 100$ with 8 channels. During triplane tokenization, we use a 3D latent grid size of $(G_x, G_y, G_z) = (8, 8, 4)$, resulting in $N = G_x G_y G_z = 256$ latent tokens with token dimension $D_{\text{token}} = 128$. For each frame, the triplane tokens are first augmented with 3D sinusoidal positional encoding, and then with a learnable temporal embedding e_{time}. In the Transformer decoder, we set the numbers of ego-motion queries and learnable external queries to $N_{\text{ego}} = 64$ and $N_{\text{ext}} = 192$,

Table 1: Comparison on Occ3D-nus 4D occupancy forecasting at 1s, 2s, and 3s. The best and second-best results are shown in **bold** and underlined, respectively.

Models	Input	mIoU (%)↑				IoU (%)↑				FPS↑	Params↓
		1s	2s	3s	Avg.	1s	2s	3s	Avg.		
OccWorld [49]	Occ	25.78	15.14	10.51	17.14	34.63	25.07	20.18	26.63	18	72M
OccSora [36]	Occ	32.77	22.04	14.40	23.07	41.39	33.68	29.97	35.01	20	174M
OccLLaMA [39]	Occ&Text	25.05	19.49	15.26	19.93	34.56	28.53	24.41	29.17	-	> 7B
OccLLM [45]	Occ&Text	24.02	21.65	17.29	20.99	36.65	32.14	28.77	32.52	-	> 7B
DOME [9]	Occ	35.11	25.89	20.29	27.10	43.99	35.36	29.74	36.36	6.54	397M
UniScene [18]	Occ&Box&Map	35.37	29.59	25.08	31.76	38.34	32.70	29.09	34.84	1.72	69M
COME [29]	Occ	42.75	32.97	26.98	34.23	50.57	43.47	38.36	44.13	0.30	692M
I^2-World [23]	Occ	47.62	38.58	32.98	39.73	54.29	49.43	45.69	49.80	37	22M
GenieDrive [47]	Occ	50.47	41.47	35.83	42.59	56.87	51.46	**47.08**	51.80	41	3.47M
ADTT (Ours)	Occ	**55.88**	**45.47**	**39.27**	**46.87**	**59.15**	**51.89**	46.87	**52.63**	**46**	**2.30M**
OccWorld-V	Camera	10.97	15.16	10.19	7.53	20.68	24.81	20.09	17.14	3.77	132M
DOME-V	Camera	17.79	14.23	11.58	14.53	26.39	23.20	20.42	23.33	2.75	457M
I^2-World-V	Camera	21.67	18.78	16.47	18.97	30.55	28.76	26.99	28.77	4.21	82M
ADTT-V (Ours)	Camera	**24.15**	**20.37**	**17.63**	**20.71**	**33.62**	**31.05**	**28.49**	**31.05**	**6.16**	**62.30M**

Table 2: Comparison on the long-term occupancy forecasting at 4s, 5s, and 6s in Occ3D-nus.

Models	mIoU (%)↑				IoU (%)↑			
	4s	5s	6s	Avg.	4s	5s	6s	Avg.
OccWorld [49]	7.79	6.29	5.76	6.61	17.23	15.31	14.54	15.69
DOME [9]	23.50	19.37	17.27	20.05	13.84	10.14	8.15	10.71
UniScene [18]	16.44	16.41	15.61	16.15	14.31	14.33	14.15	14.26
I^2-World [23]	28.59	25.32	22.55	25.49	41.75	38.47	35.41	38.54
GenieDrive [47]	31.16	27.17	23.66	27.33	42.81	39.00	35.60	39.14
ADTT (Ours)	**35.53**	**31.32**	**27.42**	**31.42**	**43.95**	**40.12**	**37.79**	**40.62**

Table 3: Cross-dataset generalization performance in Occ3D-Waymo.

Models	Avg. mIoU	Avg. IoU
Copy&Paste	17.17	28.21
I^2-World [23]	36.38	52.36
ADTT (Ours)	**38.46**	**53.19**

respectively. The upsampling operator $\mathrm{Up}(\cdot)$ uses nearest-neighbor interpolation to recover the triplane resolution from the predicted latent tokens. Training is performed on 4 NVIDIA RTX A6000 GPUs, while testing is conducted on a single NVIDIA RTX 4090 GPU. Our code is available online[2].

4.2 Comparisons with the state-of-the-art

Quantitative comparisons. Table 1 reports the occupancy forecasting results over the next 3 seconds. With direct occupancy input, our method uses explicit motion compensation and sparse delta prediction, achieving the best overall performance with the highest FPS and the fewest parameters. The only exception is the 3s IoU, where our result is slightly lower. For the vision-input setting (-V), we follow I^2-World [23] and use the output of the vision-based occupancy prediction model STCOcc [22] as the input to the OWM. In this setting, the inference speed is mainly limited by the VON network. Nevertheless, our compact OWM still achieves the best efficiency and performance. Table 2 compares long-term forecasting results under the same setting as GenieDrive [47]. Our method shows better long-term consistency. Motion compensation provides a stable prior for future scenes, and the two decoder queries help model scene changes and newly observed regions from historical context. Finally, Table 3 shows the direct evaluation results on Occ3D-Waymo under the 2Hz setting. "Copy-paste" directly uses the current frame as the future prediction and serves as a lower-bound baseline. Our method achieves the best generalization performance.

Qualitative comparisons. As shown in Figure 6, GenieDrive misses the pedestrian in the first frame and the vehicle in the third frame, and these errors persist in later predictions. It also predicts the pedestrian motion inaccurately in the last frame. In contrast, our method consistently tracks the pedestrian and keeps the correct number of vehicles in the lower region. For the newly observed region at the bottom, our prediction is less smooth than that of GenieDrive because we do not use a generative diffusion module. Nevertheless, it still recovers a reasonably accurate drivable area.

4.3 Ablation study

[2] https://github.com/kyoran/ADTT

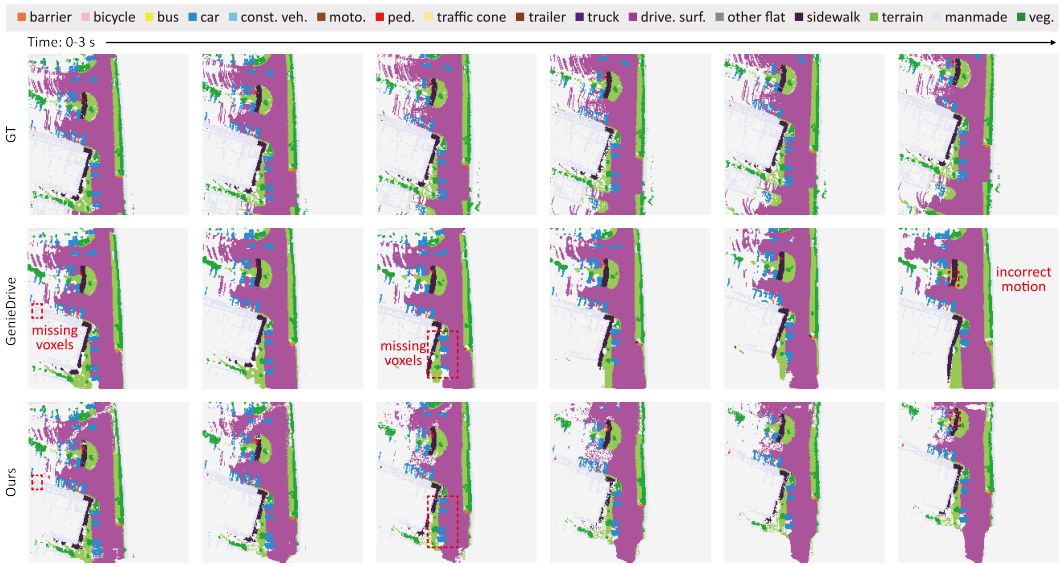

Figure 6: Occupancy forecasting comparison on the Occ3D-nus val set (zoom in for details).

Effect of key designs in ADTT.

Table 4 shows the ablation study of our core modules. M0 is the full model, and the other variants are built by removing key components from M0. M1 removes the xz and yz planes and uses only the xy plane (i.e., BEV). Without height information, it cannot provide accurate 3D predictions. In M2, the triplane and the transformer are trained fully end-to-end. This makes the prediction model rely on unstable triplane features, leading to limited performance. M3 uses only motion compensation without the transformer. Since objects move in the scene, motion compensation causes overlaps. Without the transformer, the model cannot understand scene dynamics or predict future changes, and it cannot infer newly appeared regions, resulting in poor performance. M4 removes motion compensation, and M5 directly predicts the full future scene. Both show clear performance drops. With limited model capacity, misaligned triplane features or direct full-scene prediction makes the learning problem harder. In M6, the triplane decoder is fixed when training the transformer, which leads to a slight performance drop.

Table 4: Ablation on key designs in ADTT.

Idx. Models	Avg. mIoU	Avg. IoU	FPS
M0 ADTT	**46.87**	**52.63**	46
M1 w/o triplane representation	41.16	47.85	**52**
M2 w/o pretraining triplane	35.43	40.08	46
M3 w/o transformer	28.69	34.36	59
M4 w/o motion compensation	31.87	37.48	43
M5 w/o delta prediction	36.65	45.50	46
M6 w/o finetuning tri. decoder	45.13	51.42	46

Table 5: Ablation on hyperparameters.

$G_{\{x,y,z\}}$	D_{token}	depth	Avg. mIoU	Avg. IoU	FPS
8,8,4	32	1	43.94	51.77	**48**
8,8,4	128	2	46.87	52.63	46
12,12,3	128	2	46.91	53.11	43
12,12,3	128	8	47.05	52.49	39
16,16,6	256	8	**49.13**	**54.89**	32

Table 6: Ablation on motion compensation.

Compensation strategy	Avg. mIoU	Avg. IoU	FPS
C1.1 Occ.-space warp	44.99	50.61	44
C1.2 Occ.-space learnable	43.07	48.36	42
C2.1 Tri.-space warp	46.87	52.63	46
C2.2 Tri.-space learnable	45.62	52.71	44

Effect of hyperparameters. Table 5 shows the impact of tokenization grid size (G_x, G_y, G_z), token dimension D_{token}, and transformer depth. Larger values improve occupancy forecasting by reducing information loss during tokenization and increasing model capacity, but lower the FPS. The gray row denotes our trade-off version.

Effect of motion compensation. Table 6 compares different motion compensation strategies. All variants use the same model and training settings, and only differ in the compensation strategy. Methods marked with "warp" use the explicit compensation operation in Section 3.2. "Occ.-space" and "Tri.-space" denote the compensation in the occupancy space and the triplane space, respectively. "learnable" replaces explicit warping with a learnable alignment module, i.e., Mutual Control Attention

Table 7: Ablation on ego and external query counts.

Idx.	N_{ego}	N_{ext}	Avg. mIoU↑	Avg. IoU↑
Q1	0	256	40.14	50.28
Q2	64	192	**46.87**	**52.63**
Q3	128	128	40.53	45.35
Q4	192	64	41.85	50.39
Q5	256	0	41.72	46.37

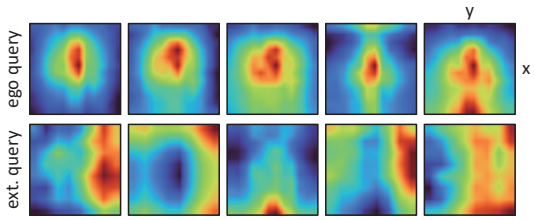

Figure 7: Visualization of ego queries and external queries on all prediction timesteps from Occ3D-nus val set, with five queries per type.

in I^2-World [23]. C1.1<C2.1 and C1.2<C2.2 show that warping discrete occupancy labels can introduce boundary artifacts, holes, and label discontinuities. Triplane features are smoother and more compact, making alignment more stable. The learnable module adds flexibility, but it leads to less stable performance and lower FPS. We therefore use C2.1 as our trade-off setting.

Effect of query counts. Ego queries provide explicit control signals that describe future ego motion, while external queries model scene dynamics outside direct control. When either type is insufficient, performance drops. In Q1, without ego queries, the model relies mainly on historical context, leading to weaker controllability. In Q5, without external queries, the model depends mostly on ego motion and cannot fully capture interactions in the scene. Increasing the number of ego queries (from Q1 to Q5) does not consistently improve performance, as they are derived from the same control signal and become redundant. Q2 achieves the best balance between ego guidance and scene modeling. Figure 7 shows the statistical behavior of different queries. For each query, we aggregate its activation over all samples in the Occ3D-nus val set on the xy plane and then normalize it to $[0, 1]$. The resulting maps indicate which regions each query mainly contributes to when decoding future state changes. Ego queries exhibit similar patterns and concentrate around the ego vehicle, while external queries cover surrounding areas. This indicates that ego queries mainly encode local control, and a moderate number is sufficient, which explains the strong performance of Q2.

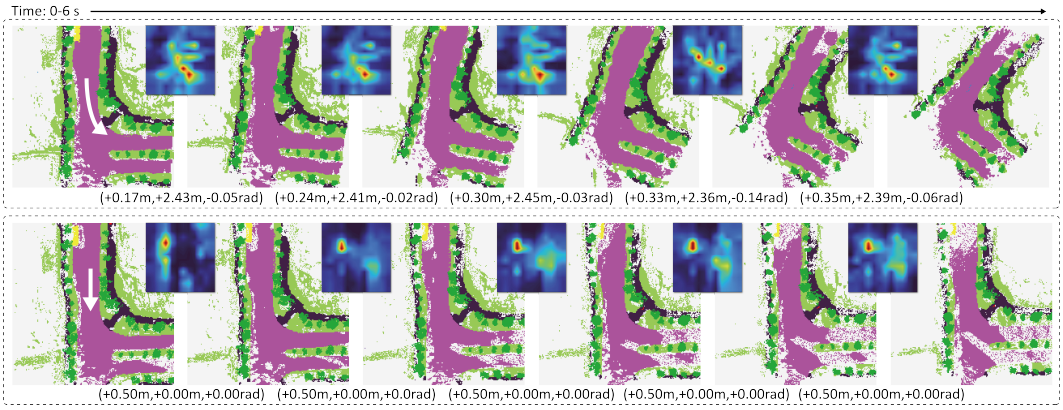

Figure 8: Controllable future occupancy prediction under different ego-motion inputs.

Effect of controllable ego query. The learned external queries are fixed after training, while the ego queries are generated from variable ego-motion inputs, including displacement along the x axis, displacement along the y axis, and the yaw angle in radians. Figure 8 shows two examples with the same initial scene but different ego-motion inputs over the next 6 seconds. In each example, adjacent frames show the specific ego-motion control signal and the visualization of one corresponding ego query. The upper example shows the vehicle turning left, while the lower example shows the vehicle going straight. The ego query mainly attends to the region around the ego vehicle and changes consistently with the given control signal, demonstrating the controllability of our method.

5 Conclusion

This paper introduces ADTT, a compact and efficient 4D OWM that performs both motion compensation and scene-change prediction in the triplane latent space. ADTT achieves SOTA performance in short- and long-term forecasting, while delivering faster inference than existing methods.

Acknowledgments and Disclosure of Funding

The study was funded by the Shenzhen Science and Technology Program under contracts No. KQTD20240729102051063 and No. JCYJ20241202130025030, the Postdoctoral Fellowship Program (Grade B) of China Postdoctoral Science Foundation under contract No. GZB20260408, the National Natural Science Foundation of China under contracts No. U25B6003, No. 62422602, No. 62372010, No. 62425101, No. 62332002, and the Major Key Project of Pengcheng Laboratory (2024B0101010003). Computing support was provided by Pengcheng Cloudbrain.

References

[1] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018.

[2] 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*, pages 11621–11631, 2020.

[3] 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*, pages 3991–4001, 2022.

[4] Anh-Quan Cao, Angela Dai, and Raoul De Charette. Pasco: Urban 3d panoptic scene completion with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14554–14564, 2024.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[6] Junliang Chen, Huaiyuan Xu, Yi Wang, and Lap-Pui Chau. Occprophet: Pushing efficiency frontier of camera-only 4d occupancy forecasting with observer-forecaster-refiner framework. In *ICLR*, 2025.

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

[8] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2025.

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

[10] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

[11] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[12] 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*, pages 9223–9232, 2023.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[14] Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. Adriver-i: A general world model for autonomous driving. *arXiv preprint arXiv:2311.13549*, 2023.

[15] 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*, pages 8340–8350, 2023.

[16] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1116–1124, 2023.

[17] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28337–28347, 2024.

[18] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *Proceedings of the computer vision and pattern recognition conference*, pages 11971–11981, 2025.

[19] Jinke Li, Xiao He, Chonghua Zhou, Xiaoqiang Cheng, Yang Wen, and Dan Zhang. Viewformer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. In *European Conference on Computer Vision*, pages 90–106. Springer, 2024.

[20] 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*, pages 9087–9098, 2023.

[21] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27137–27146, 2025.

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

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

[24] Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16317–16328, 2024.

[25] Fan Lu, Guang Chen, Zhijun Li, Lijun Zhang, Yinlong Liu, Sanqing Qu, and Alois Knoll. Monet: Motion-based point cloud prediction network. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 13794–13804, 2021.

[26] 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*, pages 21486–21495, 2024.

[27] Nicola Marinello, Simen Cassiman, Jonas Heylen, Marc Proesmans, and Luc Van Gool. Camera-only 3d panoptic scene completion for autonomous driving through differentiable object shapes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2520–2529, 2025.

[28] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022.

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

[30] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[31] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

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

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

[34] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024.

[37] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024.

[38] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[39] 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.

[40] 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*, pages 21729–21740, 2023.

[41] Zehuan Wu, Jingcheng Ni, Xiaodong Wang, Yuxin Guo, Rui Chen, Lewei Lu, Jifeng Dai, and Yuwen Xiong. Holodrive: Holistic 2d-3d multi-modal street scene generation for autonomous driving. *arXiv preprint arXiv:2412.01407*, 2024.

[42] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17642–17651, 2023.

[43] Haoran Xu, Peixi Peng, Xinyi Zhang, Guang Tan, Yaokun Li, Shuaixian Wang, and Luntong Li. Exploiting continuous motion clues for vision-based occupancy prediction. In *AAAI*, 2025.

[44] Jingyi Xu, Xieyuanli Chen, Junyi Ma, Jiawei Huang, Jintao Xu, Yue Wang, and Ling Pei. Spatiotemporal decoupling for efficient vision-based occupancy forecasting. *arXiv preprint arXiv:2411.14169*, 2024.

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

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

[47] Zhenya Yang, Zhe Liu, Yuxiang Lu, Liping Hou, Chenxuan Miao, Siyi Peng, Bailan Feng, Xiang Bai, and Hengshuang Zhao. Geniedrive: Towards physics-aware driving world model with 4d occupancy guided video generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

[48] Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. Bevworld: A multimodal world model for autonomous driving via unified bev latent space. *arXiv preprint arXiv:2407.05679*, 2024.

[49] 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*, pages 55–72. Springer, 2024.

[50] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

Appendix for "Aligned Delta-Triplane Transformers as Occupancy World Models"

This appendix provides implementation details (Section A), supplementary experiments (Section B), limitations and future directions (Section C), and broader impacts (Section D).

A Additional implementation details

A.1 Network architecture

Figure 9 and Figure 10 show the detailed structures of the encoder and decoder used to obtain triplane representations, respectively. The Transformer-based prediction module follows the official implementation in [35, 5].

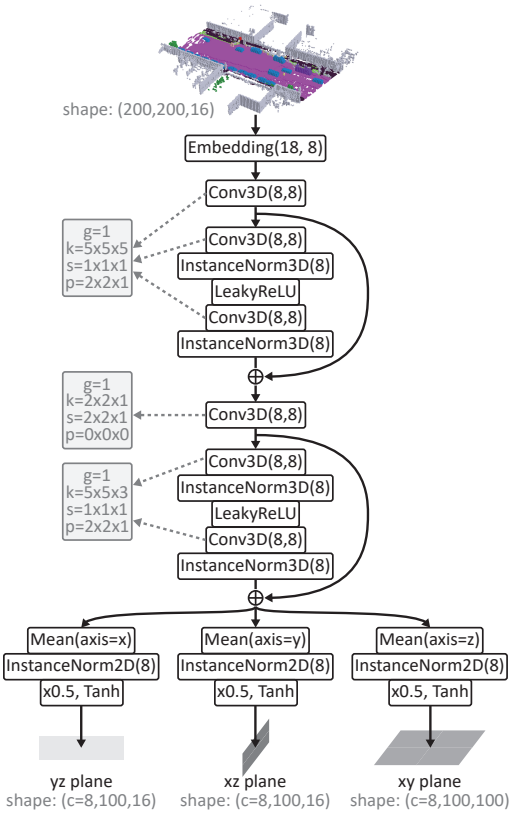

Figure 9: Detailed network structure of $\Phi_{\mathrm{Tri}}^{\mathrm{enc}}$.

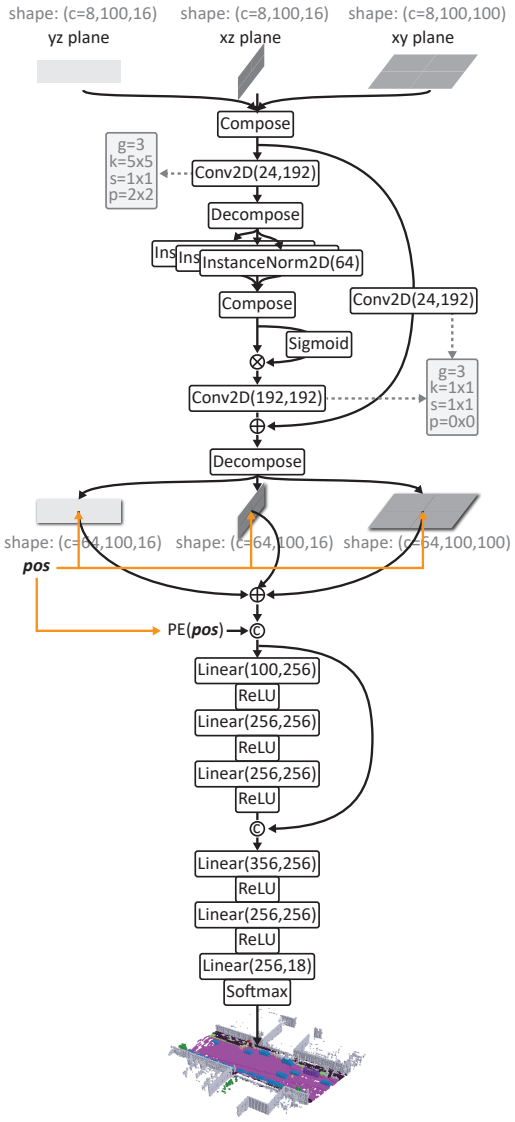

Figure 10: Detailed network structure of $\Phi_{\mathrm{Tri}}^{\mathrm{dec}}$.

A.2 Hyperparameter settings

Table 8 summarizes the default configuration, consistent with the implementation settings in the main text and the default row in Table 5. The occupancy grid has 200×200 horizontal locations and 16 vertical bins. The $16 \times 200 \times 200$ array shape in the implementation paragraph lists the vertical dimension first, whereas $(H, W, L) = (200, 200, 16)$ lists the horizontal dimensions first. Likewise, the $16 \times 100 \times 100$ spatial resolution associated with the triplanes corresponds to three separate feature maps of shapes $C \times 100 \times 100$, $C \times 16 \times 100$, and $C \times 16 \times 100$.

Table 8: Hyperparameters of the original occupancy-input ADTT model.

Hyperparameter	Value	Description
H, W, L	200, 200, 16	Dimensions of the occupancy data
τ_p	4	Number of input past frames
τ_f	6	Number of predicted future frames
C, H_{xy}, W_{xy}	8, 100, 100	Dimensions of the xy plane in the triplane representation
C, H_{xz}, W_{xz}	8, 16, 100	Dimensions of the xz plane in the triplane representation
C, H_{yz}, W_{yz}	8, 16, 100	Dimensions of the yz plane in the triplane representation
(G_x, G_y, G_z)	(8,8,4)	The tokenization grid size in the triplane representation
D_{token}	128	Dimension of the triplane token
$N_{\text{ego}}, N_{\text{ext}}$	64, 192	Numbers of ego-motion and external queries
-	4	Number of transformer heads
-	2	Depth of transformer encoder and decoder
ξ	1	Trade-off factor for $\mathcal{L}_{occ}$ in J_{fut}
$\text{AdamW}(\beta_1, \beta_2)$	0.9, 0.999	Optimizer used for training ADTT
-	5e-4	Initial learning rate
-	35	The maximum allowed gradient
-	0.01	Weight regularization factor
-	10	Batch size for training $J_{\text{enc,dec}}$
-	1	Batch size for training J_{fut}

A.3 Details of Triplane Motion Compensation

This section provides more details about the motion compensation operation used in Section 3.2. Given a historical source timestep i and a target timestep j, our goal is to warp the historical triplane $s^i = \{s^i_{xy}, s^i_{xz}, s^i_{yz}\}$ into the target ego coordinate frame. We denote the compensated triplane as

$$\overline{s}^{j \leftarrow i} = \{\overline{s}^{j \leftarrow i}_{xy}, \overline{s}^{j \leftarrow i}_{xz}, \overline{s}^{j \leftarrow i}_{yz}\}. \tag{21}$$

The compensation is implemented by pose-based backward warping. For each location on a target plane, we first map it to a metric 3D point $\mathbf{p}^j_{\text{local}}$ in the target ego coordinate system. Using the target ego pose $(\mathbf{R}_j, \mathbf{T}_j)$ and the source ego pose $(\mathbf{R}_i, \mathbf{T}_i)$, this point is transformed to the source ego frame by

$$\mathbf{p}^i_{\text{local}} = \mathbf{R}_i^\top \left(\mathbf{R}_j \mathbf{p}^j_{\text{local}} + \mathbf{T}_j - \mathbf{T}_i \right). \tag{22}$$

This gives the corresponding source-frame location from which the target feature should be sampled. The same procedure is applied separately to the three triplane planes, i.e., the horizontal xy plane and the two vertical xz and yz planes. For each plane, the relevant two coordinates are used to obtain the source sampling position.

In practice, the source sampling coordinates are normalized to the valid grid range of the corresponding source plane. We then apply grid-based feature sampling. In our implementation, we use nearest-neighbor sampling. Therefore, if the projected source location is valid, the target feature is directly copied from the nearest source grid position:

$$\overline{s}^{j \leftarrow i}_{uv}(\mathbf{q}^j_{uv}) = s^i_{uv}\left(\text{NN}(\mathbf{q}^i_{uv})\right), \quad uv \in \{xy, xz, yz\}, \tag{23}$$

where $\mathbf{q}^j_{uv}$ denotes a target plane location, $\mathbf{q}^i_{uv}$ is its projected source-plane location, and $\text{NN}(\cdot)$ denotes nearest-neighbor indexing. This operation selects existing source feature vectors without blending neighboring cells; the feature values themselves remain continuous latent representations.

For locations whose projected source coordinates fall outside the valid range, we treat them as out-of-bound regions. These locations are filled with zeros:

$$\overline{s}_{uv}^{j\leftarrow i}(\mathbf{q}_{uv}^{j}) = \mathbf{0}, \quad \text{if } \mathbf{q}_{uv}^{i} \text{ is out of range.} \tag{24}$$

We use zero padding. We also compute a binary validity mask

$$\mathcal{V}_{uv}^{j\leftarrow i}(\mathbf{q}_{uv}^{j}) = \mathbb{I}\left[\mathbf{q}_{uv}^{i} \in \Omega_{uv}\right], \tag{25}$$

where Ω_{uv} denotes the valid coordinate range of plane uv. A target location is marked as valid if both projected source coordinates lie within the legal range; otherwise it is marked as invalid. This mask records the geometric support of the source plane. The occupancy-grid coverage partition used for evaluation is described separately in Section B.5.

The resulting operation aligns historical triplanes to the target timestep through pose-based backward warping. It uses nearest-neighbor sampling for valid source locations and zero filling for out-of-bound locations. As the prediction horizon becomes longer, more target regions may move outside the support of historical triplanes. The subsequent delta prediction module infers scene content in these unsupported regions.

A.4 Hardware and runtime scope

Training uses four NVIDIA RTX A6000 GPUs, and testing uses one NVIDIA GeForce RTX 4090 GPU, matching the implementation settings in the main text. The reported 46 FPS refers to the occupancy-input evaluation; the original frozen-STCOcc camera-input pipeline runs at 6.16 FPS.

A.5 State-autoregressive rollout and action conditioning

Equation (3) summarizes the sequence-level input–output relationship. ADTT predicts future states autoregressively. To make the source and target coordinate frames explicit, we use the two-index notation $\overline{s}^{j\leftarrow i}$ for state i aligned to frame j. In the main-paper update, Eq. (18), $\overline{s}^{t}$ is the shorthand for the latest state after alignment to the next prediction frame, i.e., $\overline{s}^{t+1\leftarrow t}$ in the notation used here. Let t be the last observed timestep and define

$$\widetilde{s}^{i} = \begin{cases} \Phi_{\mathrm{Tri}}^{\mathrm{enc}}(S^{i}), & i \le t, \\ \widehat{s}^{i}, & i > t. \end{cases} \tag{26}$$

At rollout step $k \ge t$, the model uses the most recent τ_p states, aligns them to the target ego frame, and predicts the next residual. Writing $\mathcal{W}_{k+1\leftarrow i}$ for the pose-based alignment operator, the update is

$$\overline{s}^{k+1\leftarrow i} = \mathcal{W}_{k+1\leftarrow i}(\widetilde{s}^{i}), \qquad i = k-\tau_p+1, \ldots, k, \tag{27}$$

$$\widehat{s}^{k+1} = \overline{s}^{k+1\leftarrow k} + F_{\Delta}\Big(\{\overline{s}^{k+1\leftarrow i}\}_{i=k-\tau_p+1}^{k}, m^{k+1}\Big), \tag{28}$$

$$\widehat{S}^{k+1} = \Phi_{\mathrm{Tri}}^{\mathrm{dec}}(\widehat{s}^{k+1}). \tag{29}$$

Here, the residual term is precisely Δs^{k+1} in Eq. (17). The shorthand F_{Δ} comprises tokenization, the memory encoder in Eq. (13), the query-conditioned Transformer decoder in Eq. (16), and the triplane upsampling module. During evaluation, each predicted triplane updates the context window for the next prediction.

In the standard forecasting setting, future ego motions are supplied from the recorded trajectory. This setting is therefore *state-autoregressive but action-open-loop*. In the planner-in-the-loop experiment, the ego-motion conditions are instead generated by a planner.

A.6 Supervision and camera-input protocols

For the original camera-input ("-V") evaluation, occupancy forecasters are trained with ground-truth occupancy inputs and occupancy supervision. At evaluation time, the outputs of a frozen vision-based occupancy network (VON), STCOcc [22], replace the ground-truth historical occupancy. Neither the VON nor the forecaster is jointly fine-tuned in this original setting, following the evaluation setup of I^2-World [23]. Thus, the original "-V" experiment measures the transfer of an occupancy-trained forecaster to predicted occupancy inputs.

As specified after Eq. (20), standard training freezes the pretrained triplane encoder, trains the forecasting module, and fine-tunes the triplane decoder using the latent and occupancy losses. The decoder-only adaptation below is an additional camera-domain fine-tuning stage.

We additionally examine stronger VONs and two adaptation regimes. In decoder-only fine-tuning, only the triplane decoder is updated. In full-parameter fine-tuning, the camera-to-forecasting pipeline is fine-tuned end to end.

A.7 Explanation of Figure 1(c)

This section provides the construction and interpretation of the latent-space comparison in Figure 1(c). Panels (a) and (b) show a full occupancy scene and the occupancy changes after ego-motion compensation, respectively; red indicates newly appeared voxels and blue indicates disappeared voxels in the change map. Panel (c) compares the magnitudes of complete future-state features and motion-aligned residual features in the same pretrained triplane space.

Construction of the two representations. For this diagnostic, two complete consecutive occupancy states are encoded with the same pretrained and frozen triplane encoder:

$$s^t = \mathbf{\Phi}^{\text{enc}}_{\text{Tri}}(S^t), \qquad s^{t+1} = \mathbf{\Phi}^{\text{enc}}_{\text{Tri}}(S^{t+1}), \tag{30}$$

$$\bar{s}^{t+1\leftarrow t} = \mathcal{W}_{t+1\leftarrow t}(s^t), \tag{31}$$

$$\Delta s^{t+1}_{\text{target}} = s^{t+1} - \bar{s}^{t+1\leftarrow t}. \tag{32}$$

Here, $\mathcal{W}_{t+1\leftarrow t}$ is the ego-motion compensation operation described in Section 3.2 of the main paper. The residual is the difference between complete scene representations after alignment to the same target coordinate frame.

The ground-truth frame pair is used to analyze the target representations. Training follows Eq. (20), which supervises the updated triplane $\widehat{s}^{t+1}$ and the decoded occupancy $\widehat{S}^{t+1}$.

Location-wise statistic. At a spatial location $pos = (x, y, z)$, we retrieve a C-dimensional feature from each plane, using the same plane-coordinate association as in the main-paper triplane representation. Let $\mathcal{R}_{uv}(s_{uv}, pos)$ denote retrieval from the corresponding uv-plane coordinates. The joint full-state and residual features are

$$\mathbf{f}_{\text{full}}(pos) = \text{Concat}_{uv\in\{xy,xz,yz\}}\, \mathcal{R}_{uv}(s^{t+1}_{uv}, pos), \tag{33}$$

$$\mathbf{f}_{\Delta}(pos) = \text{Concat}_{uv\in\{xy,xz,yz\}}\, \mathcal{R}_{uv}(\Delta s^{t+1}_{uv,\text{target}}, pos), \tag{34}$$

where both vectors have $3C$ components. The root-mean-square (RMS) magnitude of a vector $\mathbf{f} \in \mathbb{R}^{3C}$ is

$$\text{RMS}(\mathbf{f}) = \sqrt{\frac{1}{3C}\sum_{c=1}^{3C} f_c^2}. \tag{35}$$

The histogram aggregates normalized location-wise RMS magnitudes over the validation set. Its scalar statistic is a nonnegative joint-feature magnitude.

Interpretation and supporting evidence. The residual-feature magnitudes are more concentrated near zero, consistent with the fact that alignment accounts for much of the coordinate change between adjacent frames.

The direct learning evidence is the controlled M0/M5 comparison in Table 4. With the same encoder, forecasting architecture, model capacity, and training settings, aligned-delta prediction improves average mIoU from 36.65 to 46.87 and average IoU from 45.50 to 52.63. These results support the benefit of predicting residual changes relative to an aligned prior. Table 10 summarizes the corresponding ablations, including the motion-compensation-only variant.

B Additional experimental results

B.1 Camera-input comparisons and adaptation

Table 9 extends the camera-input comparisons with COME [29], stronger VON inputs, and SparseWorld-TC. For frozen-VON ADTT, changing the VON from BEVDet to STCOcc and ViewFormer [19] improves average mIoU from 18.99 to 20.71 and 26.89, respectively. Decoder-only fine-tuning further raises the ViewFormer result to 28.52, while full-parameter fine-tuning achieves 32.17. These results demonstrate the importance of both input quality and adaptation to perception errors.

SparseWorld-TC [7] is trained end to end from images; its unstarred variants use ResNet-50 and its starred variants use a pretrained DINOv3-Base backbone [30].

Table 9: Additional camera-input forecasting results on Occ3D-nus. All scores are average mIoU (%). Decoder FT denotes additional camera-domain fine-tuning of only the ADTT triplane decoder; full FT fine-tunes the complete pipeline. A dash indicates that a separate VON is not specified for an end-to-end architecture.

VON / front end	Forecaster	Protocol	Avg. mIoU
BEVDet	COME	Reference comparison	22.26
STCOcc	COME	Reference comparison	23.07
ViewFormer	COME	Reference comparison	24.32
BEVDet	ADTT	Frozen VON	18.99
STCOcc	ADTT	Frozen VON	20.71
ViewFormer	ADTT	Frozen VON	26.89
BEVDet	ADTT	Decoder FT	23.41
ViewFormer	ADTT	Decoder FT	28.52
—	SparseWorld-TC-Small	End-to-end	25.60
—	SparseWorld-TC-Large	End-to-end	26.42
—	SparseWorld-TC-Small*	End-to-end	27.71
—	SparseWorld-TC-Large*	End-to-end	29.89
ViewFormer	ADTT	Full FT	32.17

B.2 Direct evidence for residual prediction

Table 10 isolates the role of residual prediction using the main-paper ablations. With the same encoder, forecasting architecture, model capacity, and training settings, residual prediction improves average mIoU by 10.22 percentage points and average IoU by 7.13 points over full-state prediction. The compensation-only variant also performs substantially worse, indicating that propagating historical geometry alone is insufficient. These comparisons support the empirical benefit of combining an aligned prior with learned residual dynamics.

Table 10: Controlled evidence for aligned-delta forecasting, using the corresponding variants in Table 4. Scores are percentages.

Variant	Avg. mIoU	Avg. IoU
M3: motion compensation without Transformer	28.69	34.36
M5: full-state prediction	36.65	45.50
M0: aligned-delta prediction	**46.87**	**52.63**

B.3 Movable and static classes over the forecasting horizon

For supplementary group-wise analysis, we adapt the General Movable Object (GMO) and General Static Object (GSO) grouping of Cam4DOcc [26] to the Occ3D-nus labels. GMO comprises others, bicycle, bus, car, construction vehicle, motorcycle, pedestrian, trailer, and truck. GSO comprises

barrier, traffic cone, driveable surface, other flat, sidewalk, terrain, manmade, and vegetation. Each group score averages the per-class IoUs within that group. Free space is not included in either group.

GMO groups potentially movable semantic categories. This group-wise analysis complements whole-scene forecasting metrics.

Tables 11 and 12 show the horizon-wise breakdown. ADTT achieves higher group-average scores at 1, 2, and 3 seconds for both groups, with GMO improvements of 6.24, 4.90, and 4.45 percentage points. Both methods lose more accuracy on movable categories as the horizon increases. At the class level, GenieDrive achieves higher IoU for traffic cones, pedestrians, and vegetation at all three horizons, while ADTT achieves higher group averages through gains in other categories.

Table 11: Static-class IoU and GSO-mIoU (%) at each forecasting horizon.

Horizon	Method	GSO	Barrier	T. cone	Dri. surf.	Other flat	Sidewalk	Terrain	Manmade	Veg.
1s	GenieDrive	56.36	59.67	45.76	70.68	60.54	60.68	58.68	48.81	46.07
1s	ADTT	60.83	66.46	35.26	77.49	71.11	72.50	68.69	50.98	44.22
2s	GenieDrive	49.90	50.93	35.75	65.74	54.62	54.93	52.80	43.09	41.37
2s	ADTT	52.87	55.28	27.96	73.36	63.39	64.56	61.85	40.44	36.18
3s	GenieDrive	44.82	44.24	28.97	61.93	50.03	50.32	47.88	38.22	36.99
3s	ADTT	47.12	50.15	24.62	71.79	57.02	58.61	50.79	33.10	30.94

Table 12: Movable-class IoU and GMO-mIoU (%) at each forecasting horizon.

Horizon	Method	GMO	Others	Bicycle	Bus	Car	Const. veh.	Motor.	Ped.	Trailer	Truck
1s	GenieDrive	45.23	52.12	40.24	49.72	49.31	50.34	33.73	31.95	48.01	51.61
1s	ADTT	51.47	58.25	36.73	68.23	51.22	65.03	40.20	24.54	56.28	62.77
2s	GenieDrive	33.98	45.57	27.85	34.41	36.37	43.21	21.68	19.02	38.26	39.46
2s	ADTT	38.88	46.01	25.81	48.03	41.09	49.88	36.05	15.88	41.17	46.05
3s	GenieDrive	27.83	40.55	22.09	25.43	29.60	37.86	17.33	13.05	32.09	32.47
3s	ADTT	32.28	44.24	22.84	35.47	33.77	41.23	27.69	12.48	35.34	37.51

Table 13 reports the relative decline between consecutive horizons, calculated from the displayed group scores as $(q_{\text{earlier}} - q_{\text{later}})/q_{\text{earlier}}$ for each group score q. ADTT exhibits a slightly slower relative decline for GMO, but a slightly faster relative decline for GSO, while maintaining higher absolute scores in both groups.

Table 13: Relative decrease in group mIoU between consecutive horizons (%, lower is better).

	GSO		GMO	
Transition	GenieDrive	ADTT	GenieDrive	ADTT
1s $\rightarrow$ 2s	11.5	13.1	24.9	24.5
2s $\rightarrow$ 3s	10.2	10.9	18.1	17.0

B.4 External queries and movable-object forecasting

The learned external queries attend to historical memory and produce scene-specific responses for semantic occupancy changes, including object motion, occlusion patterns, and newly revealed geometry. Table 14 compares query allocations under a fixed total budget of 256. Allocating 64 queries to ego motion and 192 to external changes achieves the highest overall mIoU and GMO-mIoU among these variants. With 128 or 64 external queries, GMO-mIoU decreases from 42.18 to 31.82 or 31.06, respectively, while GSO-mIoU varies between 50.32 and 53.98.

B.5 Cumulative travel distance and historical coverage

We further evaluate forecasting performance by the ego vehicle's cumulative travel distance from the initial timestamp. The evaluated distances range from 0 to 39.71 m and are partitioned into $[0, 4)$,

Table 14: Group-level results for the query-allocation ablation. The total query count is 256. Scores are percentages.

Variant	N_{ego}	N_{ext}	Avg. mIoU	GMO-mIoU	GSO-mIoU
Q2	64	192	46.87	42.18	52.14
Q3	128	128	40.53	31.82	50.32
Q4	192	64	41.85	31.06	53.98

$[4, 8)$, $[8, 16)$, and $[16, 40]$ m. Tables 15 and 16 report the class-wise and group-wise results in each interval.

ADTT achieves higher GSO- and GMO-mIoU in all four distance intervals. The group scores of both methods decrease as cumulative travel distance increases, while ADTT maintains its advantage in both static and movable categories.

Table 15: Static-class IoU and GSO-mIoU (%) by cumulative ego travel distance.

Distance (m)	Method	GSO	Barrier	T. cone	Dri. surf.	Other flat	Sidewalk	Terrain	Manmade	Veg.
[0, 4)	GenieDrive	61.25	66.57	53.88	73.82	65.74	66.03	62.72	52.65	48.55
[0, 4)	ADTT	64.67	70.59	42.22	79.69	73.39	73.52	74.54	54.92	48.49
[4, 8)	GenieDrive	53.19	56.18	40.10	67.43	56.54	57.58	56.15	46.63	44.91
[4, 8)	ADTT	57.14	62.81	35.71	76.11	66.50	66.41	63.76	46.18	39.65
[8, 16)	GenieDrive	48.93	50.12	33.56	64.76	53.13	54.01	52.04	42.39	41.42
[8, 16)	ADTT	51.03	56.72	28.84	72.38	58.37	61.95	54.08	40.87	35.07
[16, 40]	GenieDrive	43.61	43.65	26.43	62.46	50.04	48.55	47.07	35.96	34.67
[16, 40]	ADTT	45.03	48.90	22.05	70.25	52.83	50.87	51.85	30.72	32.81

Table 16: Movable-class IoU and GMO-mIoU (%) by cumulative ego travel distance.

Distance (m)	Method	GMO	Others	Bicycle	Bus	Car	Const. veh.	Motor.	Ped.	Trailer	Truck
[0, 4)	GenieDrive	46.39	53.92	37.31	47.13	50.56	54.07	34.25	34.60	53.48	52.23
[0, 4)	ADTT	51.36	57.43	35.81	58.89	52.81	68.61	41.58	28.04	58.53	60.62
[4, 8)	GenieDrive	40.60	48.44	39.79	44.20	43.83	47.61	31.36	24.58	40.46	45.09
[4, 8)	ADTT	44.65	52.85	35.04	54.16	49.12	52.09	40.71	18.43	43.79	55.73
[8, 16)	GenieDrive	33.73	44.88	32.08	35.00	36.59	41.53	22.87	18.72	33.73	38.18
[8, 16)	ADTT	38.15	49.02	31.58	42.71	43.22	48.77	30.55	15.57	36.23	45.76
[16, 40]	GenieDrive	26.27	37.96	20.61	24.86	29.43	34.04	15.97	14.08	28.26	31.22
[16, 40]	ADTT	30.80	42.34	21.87	33.75	31.59	39.68	25.09	12.83	31.95	38.11

We further partition the target occupancy grid according to historical geometric coverage. A historical occupancy-grid support mask is warped into the target coordinate frame using the same rigid transformation. Target voxels covered by the warped mask are denoted "observed," and the remaining target voxels are denoted "newly revealed."

Table 17 separates the two regions within each semantic group. ADTT outperforms GenieDrive in observed GSO and observed GMO regions for every distance bin. It also achieves higher newly revealed GSO scores in all four bins, with gains of 1.81 and 0.56 percentage points in the first and last bins, respectively. For newly revealed GMO, ADTT achieves higher scores in $[0, 4)$ and $[8, 16)$ m, while GenieDrive achieves higher scores in $[4, 8)$ m (39.46 versus 38.59) and $[16, 40]$ m (25.86 versus 23.19).

B.6 Planner-in-the-loop evaluation

We additionally evaluate autoregressive forecasting with planner-generated ego actions. A model predictive control (MPC) planner evaluates candidate ego-motion trajectories through latent-space rollouts and selects the trajectory with the lowest planning cost. GenieDrive uses the same MPC

Table 17: Group mIoU (%) partitioned by historical grid coverage and cumulative ego travel distance. "New" denotes newly revealed regions.

Distance (m)	Method	Observed GSO	New GSO	Observed GMO	New GMO
[0, 4)	GenieDrive	70.47	54.33	54.32	42.23
[0, 4)	ADTT	78.51	56.14	59.63	43.89
[4, 8)	GenieDrive	56.40	52.12	42.67	39.46
[4, 8)	ADTT	69.92	55.47	49.72	38.59
[8, 16)	GenieDrive	50.61	48.43	35.16	33.01
[8, 16)	ADTT	56.74	49.60	40.12	33.76
[16, 40]	GenieDrive	43.12	43.59	28.13	25.86
[16, 40]	ADTT	46.42	44.15	33.76	23.19

planner for the comparison. Table 18 reports trajectory L2 error, and Table 19 compares forecasting under recorded and planner-generated actions.

Table 18: Trajectory L2 error (m, lower is better) in the planner-in-the-loop evaluation.

Method	1s	2s	3s	Avg.
Drive-OccWorld + Planner	**0.32**	0.75	1.49	0.85
GenieDrive + Planner	0.35	0.89	1.64	0.96
ADTT + Planner	**0.32**	**0.65**	**1.28**	**0.75**

Table 19: Forecasting mIoU (%) under recorded ground-truth (GT) and planner-generated actions.

Model	Action source	1s	2s	3s	Avg.
GenieDrive	GT	50.47	41.47	35.83	42.59
GenieDrive	Planner	51.58	41.84	36.91	43.44
ADTT	GT	55.88	45.47	39.27	46.87
ADTT	Planner	55.67	46.19	39.47	47.11

ADTT achieves the lowest average trajectory L2 error of 0.75 m, compared with 0.85 m for Drive-OccWorld and 0.96 m for GenieDrive. Using planner-generated actions changes ADTT's average forecasting mIoU from 46.87 to 47.11, showing effective autoregressive forecasting under planned ego motions.

Table 20 further evaluates alignment under both action sources. Explicit alignment improves accuracy at every horizon, with an average gain of 15.00 percentage points under GT actions and 15.28 points under planned actions. This demonstrates the benefit of alignment throughout autoregressive rollout.

B.7 Counterfactual action reconstruction

Occ3D-nus supplies one recorded future per historical scene. We assess control fidelity with the Counterfactual Action Reconstruction Error (CARE; lower is better). An inverse dynamics model (IDM) is pretrained on the Occ3D-nus training set to infer ego action from two consecutive occupancy frames, and is frozen during evaluation. For each prescribed action sequence, the world model generates future occupancy frames. The frozen IDM reconstructs the actions from consecutive generated frames, and the reconstructed actions are compared with the prescribed conditions. For a sequence of T evaluated transitions, CARE is defined as

$$\mathrm{CARE} = \frac{1}{T}\sum_{t=1}^{T}\left\| f_{\mathrm{IDM}}\left(\widehat{S}^{t-1}, \widehat{S}^{t}\right) - m_t \right\|_2, \tag{36}$$

where m_t denotes the input ego action (the ego-motion condition denoted by m^t in the main text), and $f_{\mathrm{IDM}}(\widehat{S}^{t-1}, \widehat{S}^{t})$ denotes the action reconstructed from two consecutive generated occupancy frames. A lower CARE indicates that the generated scene transition more faithfully follows the given ego action.

Table 20: Alignment ablation during autoregressive rollout. All scores are mIoU (%).

Action source	Alignment	1s	2s	3s	Avg.
GT	Without	37.81	31.65	26.15	31.87
Planner	Without	38.14	32.29	25.07	31.83
GT	With	55.88	45.47	39.27	46.87
Planner	With	55.67	46.19	39.47	47.11

For the initial controlled study, we use scenes 0108, 0331, and 0564. Each scene is evaluated with seven action-conditioned rollouts: straight, left-turn, and right-turn trajectories; three randomized trajectories around the recorded trajectory; and the recorded trajectory itself. Thus, each scene includes six counterfactual rollouts and one factual rollout. Table 21 gives the scene-level results.

Table 21: Initial CARE evaluation, averaged over seven action-conditioned rollouts per scene. Lower is better.

Scene	GenieDrive	ADTT
0108	0.1557	0.1320
0331	0.0947	0.0642
0564	0.0834	0.0792
Average	0.1112	0.0918

We expand the evaluation to 36 validation scenes containing four-way intersections, T-junctions, or roundabouts. Table 22 reports the mean and standard deviation. ADTT reduces mean CARE from 0.1213 to 0.0844, a relative reduction of approximately 30.4%.

Table 22: Expanded CARE evaluation on 36 validation scenes. Results are reported as mean $\pm$ standard deviation; lower is better.

Method	CARE
GenieDrive	0.1213 ± 0.0303
ADTT	$\mathbf{0.0844} \pm 0.0261$

B.8 Higher-frequency rollout with interpolated labels

To evaluate rollouts at a higher temporal rate, we construct 5-Hz pseudo-label sequences for ego motion and occupancy using nearest-neighbor interpolation of the 2-Hz annotations. Models trained at 2 Hz are evaluated at 5 Hz without architectural changes. Table 23 reports forecasting accuracy at physical horizons of 1, 2, and 3 seconds.

At 5 Hz, ADTT achieves an average mIoU of 44.18, compared with 37.07 for GenieDrive. Relative to the 2-Hz results, average mIoU decreases by 2.69 percentage points for ADTT and 5.52 points for GenieDrive, showing a smaller accuracy drop for ADTT under this resampling protocol.

B.9 Weight of occupancy-level supervision

In standard occupancy-input training, the frozen pretrained encoder maps ground-truth future occupancy to latent supervision targets. We examine the balance between latent and occupancy supervision by varying the weight ξ in Eq. (20), as shown in Table 24. The setting $\xi = 0.8$ achieves the highest average IoU of 53.11, while $\xi = 1.0$ achieves the highest average mIoU of 46.87. We use $\xi = 1.0$ as the default setting.

B.10 Additional visual examples

Figure 11 shows the results of ADTT at a T-junction, with representative external-query visualizations shown between two frames. The regions with large changes between frames clearly correspond to

Table 23: Forecasting at 2 Hz and 5 Hz using models trained at 2 Hz. The 5-Hz targets are pseudo-labels obtained through nearest-neighbor interpolation. All scores are mIoU (%).

Model	Rollout rate	1s	2s	3s	Avg.
GenieDrive	2 Hz	50.47	41.47	35.83	42.59
GenieDrive	5 Hz	45.32	37.14	28.75	37.07
ADTT	2 Hz	55.88	45.47	39.27	46.87
ADTT	5 Hz	53.59	43.81	35.14	44.18

Table 24: Effect of ξ on Occ3D-nus 4D occupancy forecasting in 3 seconds.

Idx.	ξ	Avg. mIoU↑	Avg. IoU↑
X0	0.1	45.19	52.07
X1	0.5	45.33	52.10
X2	0.8	46.51	**53.11**
X3	1.0	**46.87**	52.63

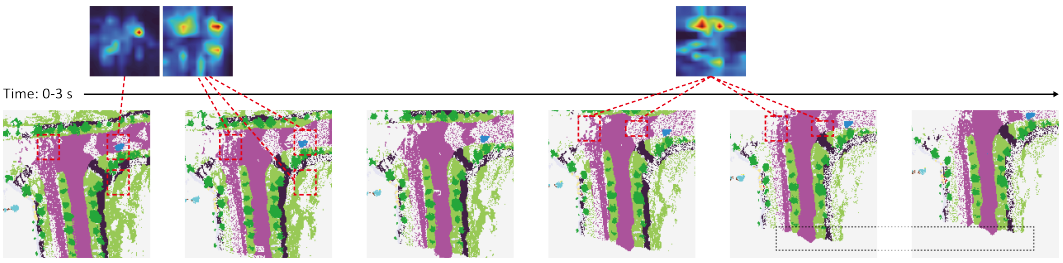

Figure 11: Occupancy forecasting results and representative external-query visualizations on the Occ3D-nus val set. Zoom in for details.

the activated external queries. Overall, our predictions preserve the T-junction structure well and maintain high consistency in the positions of obstacles and drivable areas. The newly appeared regions at the bottom receive relatively low attention from the external queries and exhibit lower prediction quality, as marked by the black dashed boxes.

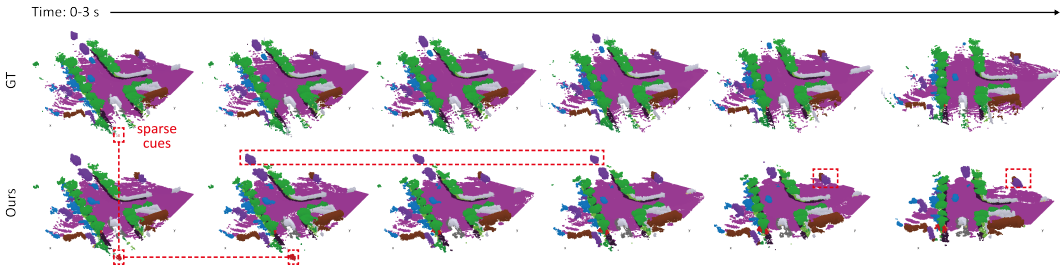

Figure 12: Detailed limitation analysis of ADTT on the Occ3D-nus val set. Zoom in for details.

As shown in Figure 12, the historical scene contains sparse pedestrian voxels near the scene boundary. A few pedestrian-class voxels persist in the predicted frames as the pedestrian moves out of view. As the ego vehicle moves forward, these voxels gradually disappear by the third frame.

We also observe an extra truck at the far end of the road in the first frame. Our model retains this truck in its predictions from the second to the fourth frame, before it gradually moves out of view in the fifth frame.

Finally, the red boxes in the last two frames illustrate denser predictions of distant truck geometry. Due to sparse LiDAR points at long range, the corresponding truck voxels in the occupancy labels are sparse, while our method predicts a denser truck shape.

C Limitations and future directions

Scope of the contribution. Cam4DOcc and OccProphet [26, 6] align historical voxel features for temporal aggregation before forecasting future occupancy. ADTT combines explicit motion alignment in a compact triplane representation with a lightweight, query-conditioned residual transition model.

Perception errors and unsupported regions. Camera-input forecasting remains sensitive to the quality of the perception front end. Stronger VONs and fine-tuning improve accuracy. Newly revealed geometry, small objects, and distant agents are important directions for further improvement, particularly in regions with limited historical evidence.

Deterministic prediction and multimodal futures. ADTT generates a deterministic future for each historical scene and ego-action sequence. Modeling multiple plausible behaviors under the same conditions is a promising extension. Stochastic latent variables or conditional generative modeling could be incorporated into the delta predictor to represent such uncertainty.

Appearance and semantic attributes. The current semantic occupancy representation describes scene geometry and category labels. Future work could extend it with color, texture, illumination, material properties, and traffic-light states. Appearance-relevant image features could be associated with 3D locations using depth and projected into additional triplane channels, with an auxiliary decoder predicting appearance attributes jointly with occupancy.

Temporal resolution and statistical coverage. Our higher-frequency study uses interpolated 5-Hz labels, and the expanded CARE evaluation covers 36 intersection-like scenes. Future evaluation can use directly annotated high-frequency sequences and a broader range of traffic scenarios.

Planning and physical interaction. ADTT integrates with an MPC planner through action-conditioned scene forecasts. Further work can extend this framework to longer planning horizons and interactive environments that model the responses of surrounding agents.

D Discussion on broader impacts

Occupancy world models support autonomous-driving research by forecasting scene geometry and semantics under different ego motions. ADTT's compact aligned-residual formulation reduces the computational cost of forecasting and supports efficient simulation and planning studies. Prediction errors involving small objects, distant agents, and newly revealed regions can propagate to downstream planning, while dataset biases can affect performance across traffic conditions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We claim the contributions of this work at the end of the Introduction, and elaborate on how we achieve them in the Methodology.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed model designs and experimental settings in the main text, and the code in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is provided in the github link, and the dataset used in this paper is publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The detailed experimental settings are presented in Section 4.1 and the supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use statistical significance in the evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources are described in Section 4.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: There are no ethical concerns involved in this work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide a discussion of broader impacts in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, and models), used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve IRB approvals or equivalent for research with human subjects.

    Guidelines:

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

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.