

DreamCube: 3D Panorama Generation via Multi-plane Synchronization

Yukun Huang¹, Yanning Zhou², Jianan Wang³, Kaiyi Huang¹, Xihui Liu^{1†}

¹The University of Hong Kong ²Tencent ³Astribot

<https://yukun-huang.github.io/DreamCube/>

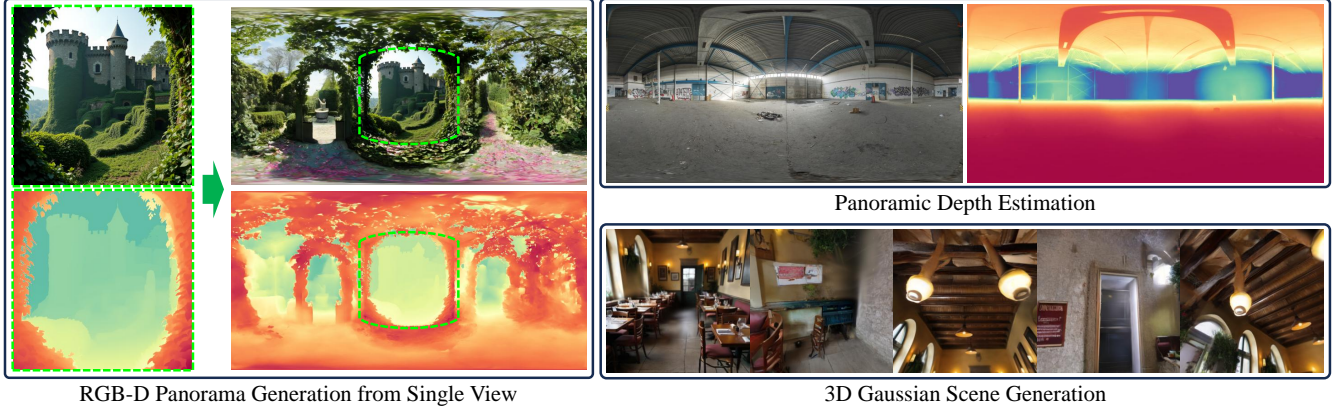


Figure 1. In this work, we introduce **Multi-plane Synchronization** to generalize 2D diffusion models to multi-plane omnidirectional representations (*i.e.*, cubemaps), and **DreamCube** for RGB-D cubemap generation. The proposed approaches can be applied to different tasks including RGB-D panorama generation, panorama depth estimation, and 3D scene generation.

Abstract

3D panorama synthesis is a promising yet challenging task that demands high-quality and diverse visual appearance and geometry of the generated omnidirectional content. Existing methods leverage rich image priors from pre-trained 2D foundation models to circumvent the scarcity of 3D panoramic data, but the incompatibility between 3D panoramas and 2D single views limits their effectiveness. In this work, we demonstrate that by applying multi-plane synchronization to the operators from 2D foundation models, their capabilities can be seamlessly extended to the omnidirectional domain. Based on this design, we further introduce DreamCube, a multi-plane RGB-D diffusion model for 3D panorama generation, which maximizes the reuse of 2D foundation model priors to achieve diverse appearances and accurate geometry while maintaining multi-view consistency. Extensive experiments demonstrate the effectiveness of our approach in panoramic image generation, panoramic depth estimation, and 3D scene generation.

1. Introduction

Humans live in a fully immersive 3D environment. Simulating this immersive experience is crucial for applications such as virtual reality, gaming, and robotics [47, 58, 65]. As a fundamental technology for building 3D world, omnidirectional content synthesis aims to generate visual content that covers a full $360^\circ \times 180^\circ$ field of view, encompassing both appearance and geometry. Despite this necessity, modern generative models [8, 28, 41] require large amounts of training data, yet the scale of currently available omnidirectional assets remains relatively small compared to conventional perspective images.

Leveraging the rich image priors from pre-trained 2D diffusion models [41], previous works [1, 9, 21, 30, 51–53, 60, 62] explore repurposing diffusion-based image generators to create 360° panoramas, which circumvents the problem of insufficient panoramic data. While most of these works [1, 9, 52, 53, 60, 62] adopt equirectangular projection (ERP) for panoramas for simplicity, this approach introduces significant challenges. The ERP causes severe spatial distortions near the poles, resulting in a pixel distribution fundamentally different from that of perspective images on

[†] Corresponding author.

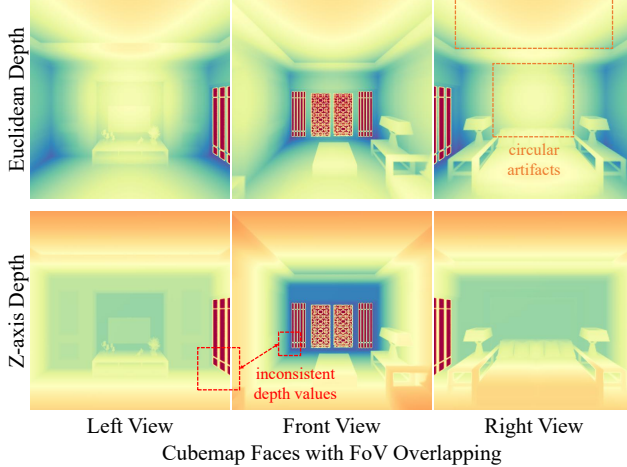


Figure 2. **Motivation.** Previous works [50, 55] on RGB-D panorama generation is based on equirectangular representations and only supports Euclidean depth instead of the more popular Z-depth. However, the distribution of Euclidean depth is quite different from that of RGB images (e.g., circles on flat surfaces as highlighted by orange dashed boxes), which hinders the use of pre-trained 2D diffusion priors. Multi-plane methods [21, 51] support Z-depth, but their adopted FoV overlapping techniques lead to depth ambiguity, as highlighted by red dashed boxes. Different from existing works, our approach supports multi-plane Z-depth generation without using FoV overlapping techniques.

which the pre-trained models were trained. These distortions not only affect visual quality but also limit the transferability of pre-trained knowledge, resulting in suboptimal generation quality particularly at the poles.

Another solution for panoramic synthesis is the multi-plane approach, which utilizes 2D diffusion to generate synchronized multi-perspective images. In general, multi-plane images are closer to the in-domain distribution of the pre-trained 2D diffusion models and can natively produce higher resolution panoramas. However, the separate generation of multiple planes leads to severe seam inconsistencies. To the best of our knowledge, all existing works [21, 51] on multi-plane panorama generation adopt field-of-view (FoV) overlapping to improve seam inconsistencies. Although effective to some extent, the overlapping planes actually cause computational redundancy and reduce the effective image resolution. Moreover, we empirically found that FoV-overlapping multi-planes exhibit significant incompatibilities in non-image domains, such as the latent space of LDM [41] and the Z-depth domain, as shown in Figure 2. These incompatibilities manifest as conflicting, non-unique z-depth values at overlapping regions, leading to the artifacts in the end. These limitations lead to a fundamental question: *Can seam inconsistencies be resolved without FoV overlapping?*

To address this question, we present the first thorough

Table 1. Comparisons of different panorama generation methods. Omni. refers to Omnidirectional ($360^\circ \times 180^\circ$).

Methods	Condition	Output	Resolution [†]	Omni.
Text2Light [4]	text	HDR	512×1024	✓
LDM3D-Pano [49]	text	RGB-D	512×1024	✓
OmniDreamer [1]	image	RGB	256×512	✓
Diffusion360 [9]	text/image	RGB	512×1024	✓
MVDiffusion [51]	text/text-image	RGB	$256 \times 256 \times 8^\ddagger$	×
PanFusion [62]	text/text-layout	RGB	512×1024	✓
PanoDiffusion [55]	image	RGB-D	256×512	✓
CubeDiff [21]	text-image	RGB	$491 \times 491 \times 6^\ddagger$	✓
DreamCube (Ours)	text-image	RGB-D	$512 \times 512 \times 6$	✓

[†]Base resolution without super-resolution or post-processing.

[‡]Calculated effective resolution accounting for overlapping fields of view.

analysis of 2D diffusion models to identify the causes of seam inconsistencies in multi-plane generation. Our analysis reveals that these inconsistencies are rooted in the translational inequivalence of certain neural operators in the omnidirectional image domain. Surprisingly, we find that by adapting these operators to be omnidirectionally translation-equivalent, existing 2D diffusion models can generate seam-consistent panoramic multi-planes without requiring fine-tuning or FoV overlapping. In this work, we refer to this adaptation as “multi-plane synchronization”.

Building on multi-plane synchronization, we introduce DreamCube, a novel framework for generating RGB-Depth (RGB-D) cube maps from a single view through joint panoramic appearance and geometry modeling. Inspired by previous work on diffusion-based depth estimation [10, 15, 22, 26], we repurpose the pre-trained 2D diffusion model for multi-plane image and depth generation. Unlike previous approaches [49, 55] to equirectangular RGB-D panorama generation, we adopt cube maps as panorama representation. This choice is significant because pixels in each plane of a cube map are uniformly distributed (due to perspective projection rather than equirectangular projection) and therefore align with the in-domain distribution of the pre-trained 2D diffusion models. Table 1 compares our approach with existing panorama generation methods across key dimensions. Together with multi-plane synchronization, our method maximally exploits pre-trained 2D diffusion models for joint modeling of panoramic appearance and geometry. The resulting RGB-D cubemaps can be easily lifted to 3D scene, effectively enabling single-view to omnidirectional 3D scene generation.

Our main contributions are as follows:

- We thoroughly analyze the operator incompatibilities of existing 2D diffusion models for panoramic multi-plane generation, and propose a multi-plane synchronization strategy that enables seam-consistent cubemap generation without fine-tuning or FoV overlapping.

- Based on multi-plane synchronization, we further introduce DreamCube, a masked RGB-D cubemap generator from single view for panoramic appearance and geometry generation.
- Extensive experiments demonstrate the effectiveness of our approach in RGB-D panorama generation, panoramic depth estimation, and 3D scene generation.

2. Related Work

2D panorama generation. Panorama image generation aims to create 360-degree panoramas from text prompts or partial images. Early works [4, 36] utilized GAN [11] for panorama generation, while recent diffusion models have enabled more sophisticated panoramic synthesis. Current approaches either iteratively outpaint 360° panoramas from narrow field-of-view images [2, 30, 33] or directly generate complete panoramas [51–53, 55, 60, 62]. Many methods employ equirectangular projection (ERP) to represent 360°×180° views, but face challenges with geometric distortions, particularly in polar regions. PanFusion [62] addresses this through a dual-branch diffusion model combining panorama and perspective domains with specialized attention mechanisms, while DiffPano [60] introduces a spherical epipolar-aware attention module for consistency. Various denoising and decoding strategies [52, 53, 55] have been proposed to reduce stitching artifacts. To avoid polar distortion and leverage pretrained perspective image diffusion models, alternative approaches [21, 51] generate multiple perspective views. CubeDiff [21] introduces a cubemap representation that projects the 360° view onto six faces of a cube, effectively eliminating severe distortions while maintaining consistency. Yet these multi-plane approaches use FoV overlapping to reduce seam artifacts, which introduces computational redundancy and limits effective resolution.

Generative depth estimation. Various methods explore diffusion models for depth estimation [6, 20, 44, 45, 63]. Marigold [22] converts pretrained Latent Diffusion Models into image-conditioned depth estimators. However, directly applying these depth estimation methods to panoramic images presents challenges: the domain gap between ERP and perspective image representations degrades performance. Alternatively, estimating depth for multiple perspective views independently requires an additional alignment step to ensure consistency. DAC [12] attempts to jointly learn depth in both ERP and perspective spaces within a unified framework, but its effectiveness remains limited.

Joint appearance and geometry modeling. Another line of works jointly models appearance and geometry [7, 29, 57]. GeoWizard [10] extends Stable Diffusion with a geometry switcher and scene distribution decoupler to jointly predict depth and normals. Orchid [26] trains a VAE for a new joint latent space incorporating depth and normals. This joint modeling strategy has also demonstrated

effectiveness in specific domains, such as human-centric scenarios [19, 24, 34]. Following this trend, we jointly model appearance and depth information for panorama generation to enhance scene structure understanding and obtain high-quality depth maps for subsequent 3D scene creation. Unlike previous RGB-D panorama generation methods [49, 55], our approach operates directly on perspective images, better leveraging prior knowledge from pretrained models while jointly modeling appearance and depth for high-quality panoramic scene creation.

3. Multi-plane Synchronization

3.1. Preliminary: Panorama Representations

Panorama representations refer to methods of storing 360-degree images or videos, allowing viewers to look in all directions from a single viewpoint. Equirectangular and cube map formats are the two primary formats due to their compatibility with existing image processing frameworks.

Equirectangular representations project the panoramic content onto a rectangular grid, where the horizontal axis represents the longitude and the vertical axis represents the latitude. This format is one of the most commonly used for storing and transmitting 360-degree images and videos due to its simplicity and compatibility with standard image and video formats. Despite its ease of use, the equirectangular representation suffers from significant distortion, especially near the poles, where the pixels are stretched.

Cube map representations divide the panoramic scene into six square faces of a cube, each representing a 90-degree field of view. The primary advantage of the cube map is its uniform distribution of pixels across the entire field of view, which minimizes distortion compared to the spherical representation. However, it can introduce discontinuities at the edges where the cube faces meet, which may require additional processing to ensure seamless transitions. In our work, we adopt the cube map representation for its minimal distortion and compatibility with pre-trained 2D diffusion models.

3.2. Multi-plane Synchronization

Directly applying 2D diffusion models pre-trained on single-view images to multi-plane panoramic representations like cube maps faces a fundamental limitation: They generate each face independently with no inherent correlation, which leads to discontinuities at the seams of adjacent cube faces. To address this issue, we propose Multi-plane Synchronization, which achieves seamless panoramic generation without the need for fine-tuning or explicitly constructing overlapping regions [21, 51].

To present our multi-plane synchronization, we use U-Net-based diffusion models [41] and six-plane cube maps throughout this work. The principles established, however,

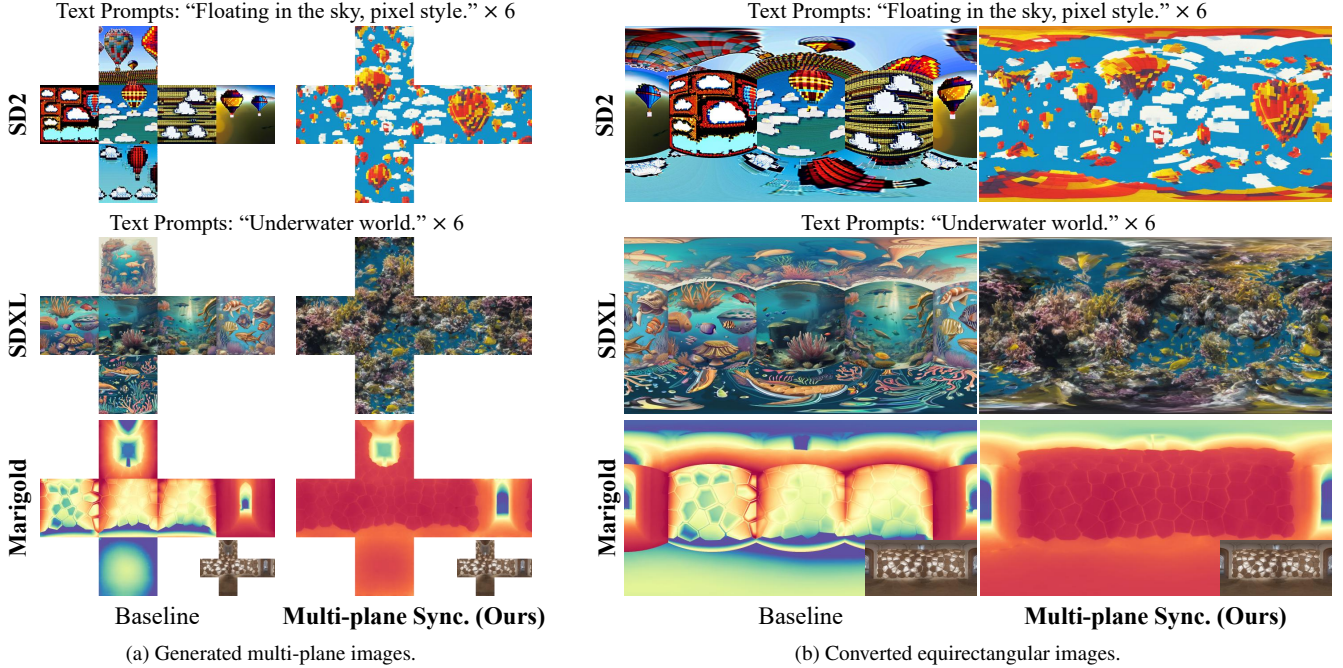


Figure 3. Results of **Multi-plane Synchronization** on pre-trained 2D diffusion models: SD2 [41], SDXL [39], and Marigold [22]. Our method enables 2D diffusion to generate multi-plane synchronized omnidirectional image representations without fine-tuning.

are architecture-agnostic and can be adapted to other diffusion frameworks, such as DiT [37] and alternative panorama representation with minimal modifications.

Analysis of spatial operators. Neural operators should ideally maintain translation-equivalence in omnidirectional representations, but standard operators (*e.g.*, self-attentions) in 2D diffusion models break this property. For instance, in standard 2D convolutions, boundary pixels of each cube face are padded with zeros instead of information from adjacent faces, resulting in discontinuities at cube map seams. Therefore, all operators of 2D diffusion involving the spatial domain must be adapted to ensure translation invariance in the omnidirectional domain.

Synchronizing spatial operators to cube maps. For U-Net-based diffusion models, we adapt three key operators from single-view ($H \times W$) to multi-plane domains ($M \times H \times W$, where $M = 6$ for cube maps): (1) synced attentions - reshaping tokens from $(BM) \times (HW) \times C$ to $B \times (MHW) \times C$, enabling attention to operate across all faces simultaneously; (2) synced 2D convolutions - replacing zero-padding with geometrically projected pixels from adjacent faces; and (3) synced group normalizations - calculating statistics globally across all planes rather than per-view independently.

Remarks. While individual adaptations of these operators exist in previous works-flattened attentions in [21, 46], projective resampling in [36], and tiled group normalizations in [16, 21], none provides a comprehensive so-

lution. Our proposed multi-plane synchronization offers a more complete and systematic adaptation of 2D diffusion models to support the omnidirectional image domain. As shown in Figure 3, our method can be applied to various pre-trained 2D diffusion models, such as Stable Diffusion [41] and Marigold [22], to achieve omnidirectional generation. These results demonstrate the effectiveness and potential of our method in omnidirectional vision tasks such as panorama generation and panoramic depth estimation.

4. Method

In this section, we introduce **DreamCube**, a masked RGB-D cube map diffusion model for joint panoramic appearance and geometry generation.

4.1. Generative Formulation

We conceptualize the generation of RGB-D cube map as the production of six colored images $x_c \in \mathbb{R}^{M \times H \times W \times 3}$ and their corresponding depth maps $x_d \in \mathbb{R}^{M \times H \times W}$, where $M = 6$ denotes the number of cubic views. Each of these views is associated with a specific face of the cube map, namely *front*, *right*, *back*, *left*, *up*, and *down*. Given single-view RGB-D images and multi-view texts as conditions, our model aims to generate RGB-D images of other cubic views through a synchronous diffusion denoising process.

At training time, we first encode the colored cube map x_c and the corresponding depth cube map x_d into latents

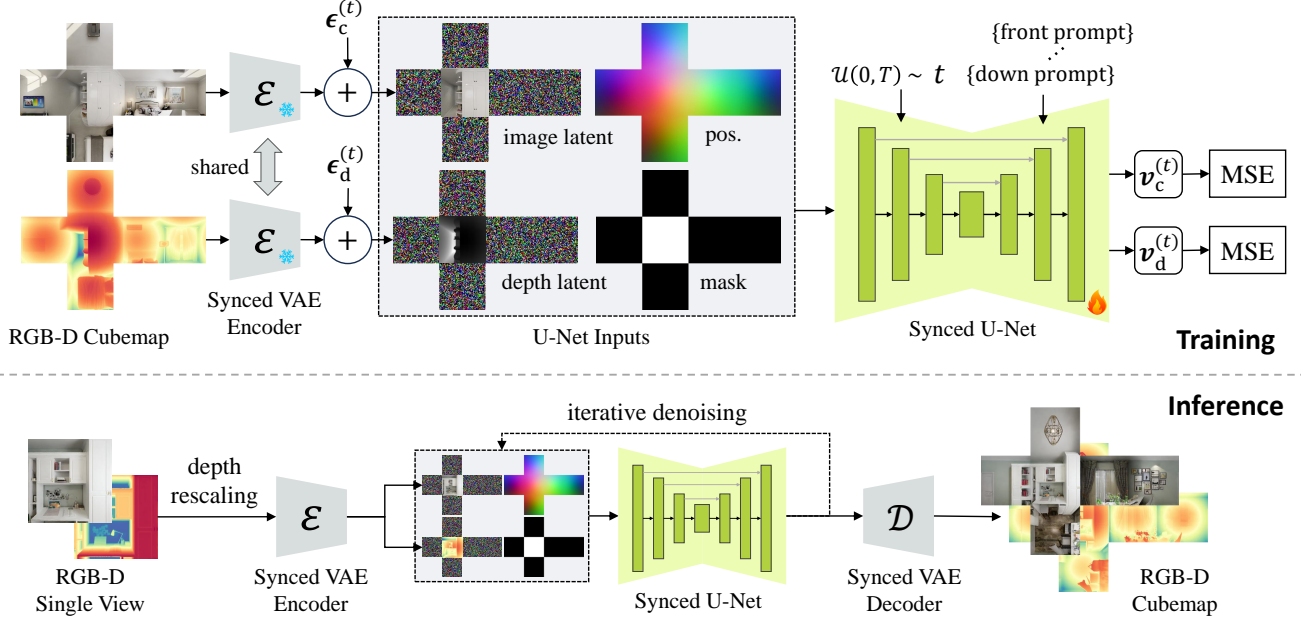


Figure 4. Training and inference framework of **DreamCube** for RGB-D cube map generation. At training time, RGB-D cube faces are encoded by synced VAE and injected masked Gaussian noises, obtaining image and depth latents. These latents are concatenated with positional encoding and mask as diffusion U-Net’s input. The entire U-Net is fine-tuned with v -objective [42] to learn to jointly denoise RGB and depth latents. At inference time, DreamCube receives single-view RGB-D images and multi-view texts as input and generates completed RGB-D cube map representations via iterative diffusion denoising and synced VAE decoding.

$z_c \in \mathbb{R}^{M \times H' \times W' \times 4}$ and $z_d \in \mathbb{R}^{M \times H' \times W' \times 4}$ using a pre-trained VAE [41]. In particular, the depth inputs x_d are broadcasted to 3-channels to match the configuration of the VAE. Then, the masked Gaussian noises $\epsilon_c^{(t)}$ and $\epsilon_d^{(t)}$ are sampled and injected into z_c and z_d , obtaining $z_c^{(t)}$ and $z_d^{(t)}$, respectively. Note that the conditional image latent (*w.l.o.g.*, we choose the front view as condition) are kept noise-free throughout the diffusion process. Finally, the image latents $z_c^{(t)}$ and depth latents $z_d^{(t)}$ are concatenated with the positional encoding and mask on the channel axis and fed into the diffusion U-Net. We use v -prediction [42] as the learning objective, and the training loss is given as follows:

$$\begin{aligned} \mathcal{L} = & \mathbb{E}_{x_c, \epsilon_c \sim \mathcal{N}(0, I), t \sim \mathcal{U}(T)} \|v_c^{(t)} - \hat{v}_c^{(t)}\|_2^2 \\ & + \mathbb{E}_{x_d, \epsilon_d \sim \mathcal{N}(0, I), t \sim \mathcal{U}(T)} \|v_d^{(t)} - \hat{v}_d^{(t)}\|_2^2, \end{aligned}$$

where $v_c^{(t)}$ and $v_d^{(t)}$ are predicted by the diffusion U-Net.

At inference time, the conditional image latents $c_c \in \mathbb{R}^{1 \times H' \times W' \times 4}$ and depth latents $c_d \in \mathbb{R}^{1 \times H' \times W' \times 4}$ are concatenated with Gaussian noises to get the initial noisy latents $z_c^{(T)}$ and $z_d^{(T)}$. These noisy latents will be iteratively denoised as the time step decreases from T to 1 to get noise-free cube map latents $z_c^{(0)}$ and $z_d^{(0)}$. The final results can be obtained by decoding these latents with VAE.

4.2. DreamCube: RGB-Depth Cube Diffusion

As illustrated in Figure 4, DreamCube adopts several designs to enable RGB-D cube map generation from a single view, which involves depth data processing, omnidirectional position encoding, and multi-plane synchronization.

Depth data processing. Panoramic depth data usually uses the Euclidean distance metric, the depth distribution of which is significantly different from the RGB image distribution (*e.g.*, the circle on the flat wall as shown in Figure 2), which hinders the joint modeling of RGB images and depth maps. Therefore, unlike previous works [49, 55], DreamCube models the Z-distance, which is more consistent with the diffusion image priors, rather than the Euclidean distance.

Additionally, DreamCube estimates the depth maps of other cubic views based on the conditional view. Even though the conditional depth values lie in the diffusion’s in-domain range $[-1.0, +1.0]$, the generated depths of other views may exceed this range, leading to performance degradation. Inspired by recent work on depth inpainting [35], we adopt a depth rescaling strategy. Specifically, we rescale the conditional depth to the range of $[-s, +s]$ before input, where the real number s is less than 1. This strategy creates an additional margin for the depth generation of other views, thus avoiding out-of-domain depth values.

Omnidirectional positional encoding. To ensure geo-

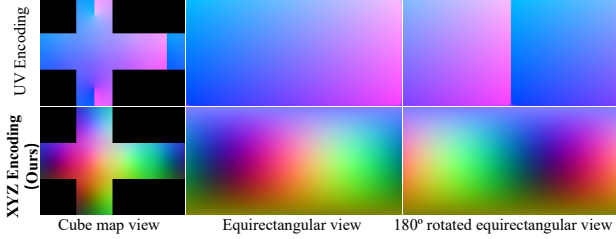


Figure 5. Comparison between UV encoding and XYZ encoding for omnidirectional image representations.

metric consistency and coherent object relationships across generated cube faces, we improve the spatial awareness of LDM [41] by integrating positional encodings derived from the 3D geometry of the cube. Specifically, for each point on a cube face, we project its coordinates onto the unit sphere with (x, y, z) values normalized to $[-1, 1]$. These normalized coordinates are then appended as three additional channels to LDM’s latent representations. This encoding strategy encodes spatial information for each latent patch relative to its cube face while ensuring smooth omnidirectional transitions - addressing limitations of the UV-based encoding proposed by CubeDiff [21] (see comparisons in Fig. 5).

Multi-plane synchronized diffusion. As described in Sec. 3.2, the proposed multi-plane synchronization can improve 2D diffusion models to handle omnidirectional image representations. Specifically, all self-attentions, 2D convolutions, and group norms in the diffusion U-Net and VAE are changed to multi-plane synchronized operators. This strategy enables DreamCube to model multi-plane omnidirectional representations, significantly alleviating the tricky issues of discontinuous seams and inconsistent color tones in cube map generation.

4.3. Building 3D Scene from RGB-D Cubemap

The generated RGB-D panoramas contain the direction and distance of each pixel, so a colored 3D point cloud can be obtained by projecting all pixels into 3D space. We can further convert the point cloud into different 3D representations, such as 3D meshes and 3D Gaussians [23]. Note that these conversions can be achieved either by differentiable optimization or by handcrafted algorithms. We choose handcrafted algorithms for fast 3D scene reconstruction in seconds from RGB-D panoramas. Specifically, for 3D mesh reconstruction, we use the obtained point cloud as vertices, and the vertex colors are assigned by the RGB values of the corresponding pixels. The connections between vertices can be extracted based on the adjacency relationship of image pixels. For 3D Gaussians, the position and color of each Gaussian point can be directly assigned from the colored point cloud, while other properties are inferred

using a method similar to WonderWorld [61].

It is worth mentioning that different panoramic projections affect the quality of the reconstructed 3D scene. For equirectangular-based RGB-D panoramas, due to the significant geometric distortion at the poles, the reconstructed 3D point cloud will be unevenly distributed and particularly dense at the top and bottom poles. In contrast, the distribution of 3D points from RGB-D cubemap is more uniform and regular, as shown in Figure 9.

5. Experiment

5.1. Implementation Details

We implement both Multi-plane Synchronization and DreamCube using PyTorch. For DreamCube, we utilize Stable Diffusion v2 [41] as pre-trained backbone. At training time, we adopt the DDPM noise scheduler [18] with 1000 timesteps. We use a batch size of 4 for training, where the resolution of RGB images and depth maps is 512×512 . Random rotation and flipping are used to expand the amount and diversity of panorama training data. The depth rescaling parameter s is randomly sampled from a uniform distribution in $[0.2, 1.0]$ during training. We froze the VAE and fine-tuned the diffusion U-Net for 10 epochs. We use the AdamW optimizer with a learning rate of $2 \cdot 10^{-5}$. The entire training process took approximately two days on four Nvidia L40S GPUs. At inference time, we adopt the DDIM noise scheduler [48] with 50 sampling steps. The depth rescaling parameter s is fixed to 0.6 for inference.

5.2. Datasets

We conduct experiments in two distinct data settings to comprehensively evaluate the performance of our method across various scenarios:

Indoor setting. To ensure fair comparison with prior work, we first evaluate our approach on the Structured3D dataset [64], which provides equirectangular indoor RGB-D panorama with a 512×1024 resolution. Following the same experimental protocol as PanoDiffusion [55], we utilize their exact data split consisting of 16,930 training, 2,116 validation, and 2,117 test instances. Besides, the SUN360 dataset [56] is also used for out-of-domain evaluation.

General setting. To further evaluate our model’s generalization capabilities across diverse environments, we construct a more comprehensive dataset by combining multiple publicly available sources, including Structured3D [64], Pano360 [25], Polyhaven [40], Humus [38], HDRI-Skies [13] and iHDRI [14]. This combined dataset encompasses a broad spectrum of both indoor and outdoor environments, resulting in more than 30,000 panoramic instances. This general setting allows us to evaluate the robustness of our approach across a wider range of scenarios.

Data processing pipeline. All panorama data needs to

be processed into a unified format, including RGB cube maps, depth cube maps, and image captions for each cube face. For datasets not originally in cubemap format, we apply standard perspective projection to produce cube maps. Next, we adopt BLIP-2 [31] to obtain image captions of all cube faces. While the Structured3D dataset includes depth data, the other datasets only contain RGB data. To annotate the depth of these panoramas, we build a high-resolution panorama depth estimation pipeline by connecting the existing panorama depth estimation work Depth Anywhere [54] and the image-guided depth up-sampling work PromptDA [32], which supports panoramic depth estimation at 4K resolution. We use this pipeline to perform depth estimation on equirectangular-based panoramas and then project the obtained depth panoramas into cube maps.

5.3. RGB-D Panorama Generation

Evaluating RGB-D panorama generation from a single view presents unique challenges due to the absence of standardized benchmarks. We train and test our method on the standard split of the Structured3D [64] dataset following another RGB-D panorama generation work, PanoDiffusion [55]. To comprehensively evaluate the capabilities of our method, we evaluate the RGB panorama quality and depth panorama accuracy separately tailored to each modality’s characteristics.

Evaluation protocol for RGB panorama generation. We evaluate our method on both in-domain Structured3D [64] and out-of-domain SUN360 [56] datasets. SUN360 consists of around 1000 panorama images including both indoor and outdoor scenes. We use Fréchet Inception Distance (FID) [17] and Inception Score (IS) [43] to evaluate the visual quality of generated panorama images.

Evaluation protocol for depth panorama generation. Since no ground-truth depths exist for generated panoramas, we propose a reference-based evaluation protocol. We first project our generated RGB-D panoramas into multiple perspective views at randomly sampled viewpoints. For each projected RGB image, we obtain a reference depth map using Depth Anything v2 [59], a state-of-the-art monocular depth estimator. This provides pseudo ground-truth depth for each perspective view. We then compare projected depth maps against these reference depths using standard metrics: δ -1.25, AbsREL, RMSE and MAE, following the implementation in [5].

Quantitative results for RGB panorama generation.

We compare our approach with state-of-the-art panorama generation methods including OmniDreamer [1], LDM3D-Pano [50], Diffusion360 [9], MVDiffusion [51], PanoDiffusion [55], and PanFusion [62]. The comparison results are reported in Table 2. Our method performs competitively on both in-domain and out-of-domain datasets with the best overall ranking. Note that for SUN360, we only

Table 2. **Quantitative results on RGB panorama generation** compared with state-of-the-art methods, evaluated on both in-domain Structured3D and out-of-domain SUN360 datasets.

Methods	Structured3D		SUN360		Avg. Rank
	FID ↓	IS ↑	FID ↓	IS ↑	
OmniDreamer [†] [1]	97.46	3.35	128.17	2.29	7.0
LDM3D-Pano [50]	32.57	<u>6.13</u>	72.67	4.86	<u>3.3</u>
Diffusion360 [†] [9]	26.23	4.85	63.03	4.21	3.5
MVDiffusion [51]	35.99	5.00	67.22	4.33	4.0
PanoDiffusion [55]	<u>16.20</u>	4.04	80.02	3.91	4.8
PanFusion [62]	44.86	6.18	84.25	<u>4.98</u>	3.8
DreamCube (Ours)	12.58	5.50	<u>66.56</u>	5.35	1.8

[†]Including SUN360 as training data.

Table 3. **Quantitative results on depth panorama generation** compared with RGB-D panorama generation methods: LDM3D-Pano [50], PanoDiffusion [55], and panoramic depth estimation method: Depth Any Camera (DAC) [12].

Methods	δ -1.25 ↑	AbsRel ↓	RMSE ↓	MAE ↓
LDM3D-Pano [50]	0.655	0.199	0.323	0.267
PanoDiffusion [55]	0.695	0.160	0.301	0.255
DAC [12]	0.751	0.139	0.266	0.220
DreamCube (Ours)	0.787	0.133	0.256	0.211

use it for out-of-domain evaluation while Diffusion360 and OmniDreamer use it for training. Nonetheless, our method significantly outperforms both methods on inception score.

Quantitative results for depth panorama generation.

We compare our approach with RGB-D panorama generation methods: LDM3D-Pano [50], PanoDiffusion [55], and panoramic depth estimation method: Depth Any Camera (DAC) [12], and the results are reported in Table 3. Our method outperforms the competing methods consistently on all metrics, demonstrating the superiority of the proposed joint RGB-D cube map generation in obtaining accurate geometry compared to equirectangular-based RGB-D generation and depth estimation methods.

Qualitative results. We provide visualization results of the generated RGB-D panorama, as shown in Figure 6. Compared with the equirectangular based RGB-D panorama generation methods: LDM3D-Pano [49] and PanoDiffusion [55], our method is able to obtain more detailed and accurate geometry. Moreover, our generated RGB panoramas have fewer artifacts than PanoDiffusion which uses the same training set as ours. We attribute these advantages to the fact that the cube map representations can better exploit the 2D diffusion’s image priors compared to the geometrically distorted equirectangular representations.

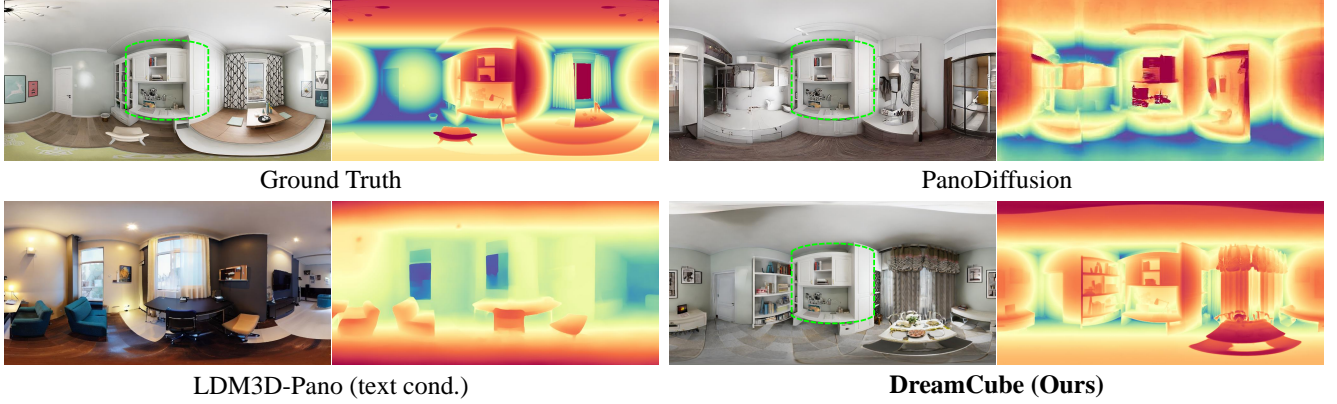


Figure 6. **Qualitative results of the proposed DreamCube** on RGB-D panorama generation compared with RGB-D panorama generation methods: LDM3D-Pano [50] and PanoDiffusion [55]. The **green** dashed boxes highlight the input image condition.

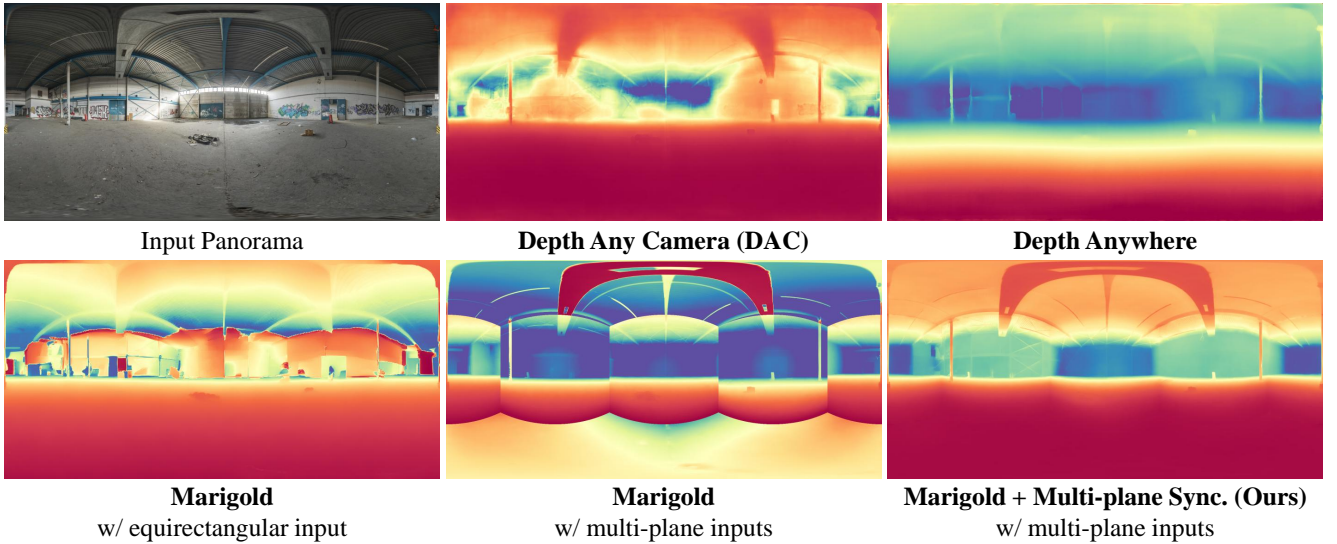


Figure 7. **Qualitative results of the proposed Multi-plane Synchronization** on panoramic depth estimation compared with recent panoramic depth estimation methods: Depth Any Camera (DAC) [12] and Depth Anywhere [54].

5.4. Panoramic Depth Estimation

Our proposed Multi-plane Synchronization extends naturally to monocular depth estimation, demonstrating its versatility beyond RGB panorama generation. As shown in Figure 7, applying our synchronization approach to Marigold effectively eliminates the discontinuous seams that occur when processing multi-plane inputs with the original model. Qualitative comparisons with recent panoramic depth estimation methods: Depth Any Camera (DAC) [12] and Depth Anywhere [54] reveal that our approach captures more detailed geometric structures while maintaining the continuity of depth values at left-right seams (which Depth Anywhere fails). These results show that the proposed Multi-plane Synchronization can effectively generalize diffusion-based monocular depth estimation models to

multi-plane omnidirectional representations with painless modification and minimal performance loss.

5.5. Panorama-to-3D Reconstruction

An important application of DreamCube is fast 3D scene generation. Benefiting from the joint RGB-D panorama generation model and the rapid panorama-to-3D projection algorithm, our approach can achieve 3D scene generation from a single view in about ten seconds. We present the visualized results of the generated 3D scenes in both 3D meshes and 3D Gaussian representations, as shown in Figure 8. The visual quality of the reconstructed 3D scene is comparable to that of the original panorama. Additionally, we analyze the impact of different formats of RGB-D panoramas on 3D reconstruction, as illustrated in Figure 9.



Figure 8. **Panorama-to-3D scene reconstruction.** Based on the RGB-D cubemap generated by DreamCube, we can reconstruct the corresponding 3D scenes in seconds and obtain both 3D mesh and 3D Gaussian [23] representations.

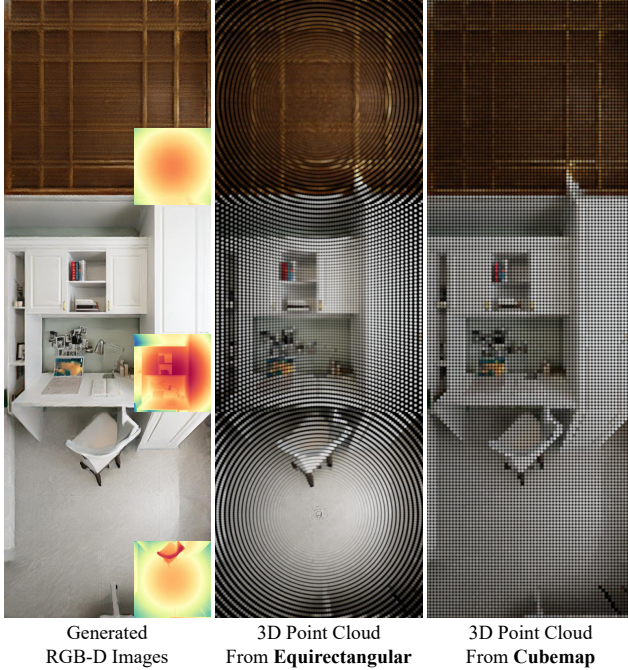


Figure 9. **Qualitative comparison of 3D point clouds** reconstructed from equirectangular-based and cubemap-based RGB-D panoramas. Equirectangular panoramas produce an uneven, ring-shaped 3D point distribution dense near the poles, while cubemap panoramas yield a more uniform and regular distribution.

The 3D point distribution derived from equirectangular-based panoramas is uneven, exhibiting a ring-shaped pattern

with particularly dense points near the poles. In contrast, the 3D point distribution from cubemap-based panoramas tends to be more uniform and regular.

5.6. Ablation Study and Analysis

We perform a series of analyses on the proposed Multi-plane Synchronization and DreamCube to evaluate their effectiveness and robustness under various conditions.

Ablation analysis of Multi-plane Synchronization. To analyze the impact of different synced operators, we perform different synchronization configurations and provide the visualization results in Figure 10. Synced self-attention ensures content consistency across faces, synced group normalization harmonizes color tones, and synced convolutions reduce seam discontinuities. Combining all three operators produces panoramas with seamless transitions, consistent content, and uniform style.

Ablation analysis of DreamCube. We analyze different components of DreamCube, with results shown in Table 4. We evaluate both RGB and depth panorama generation on the Structured3D test split [64], following evaluation protocol in Sec. 5.3. Both XYZ Positional Encoding (“XYZ Pos.”) and Multi-plane Synchronization (“Sync.”) improve performance, with “Sync.” yielding the most substantial gains, which reduces FID by 8.77 and improves δ -1.25 by 0.103. Specifically, Synced Self-Attention (“SyncSA”) contributes the most performance gain compared to other synced operators. Besides, we further provide a qualitative ablation analysis of XYZ Positional Encoding in Figure 11. Our design effectively alleviates line artifacts and content

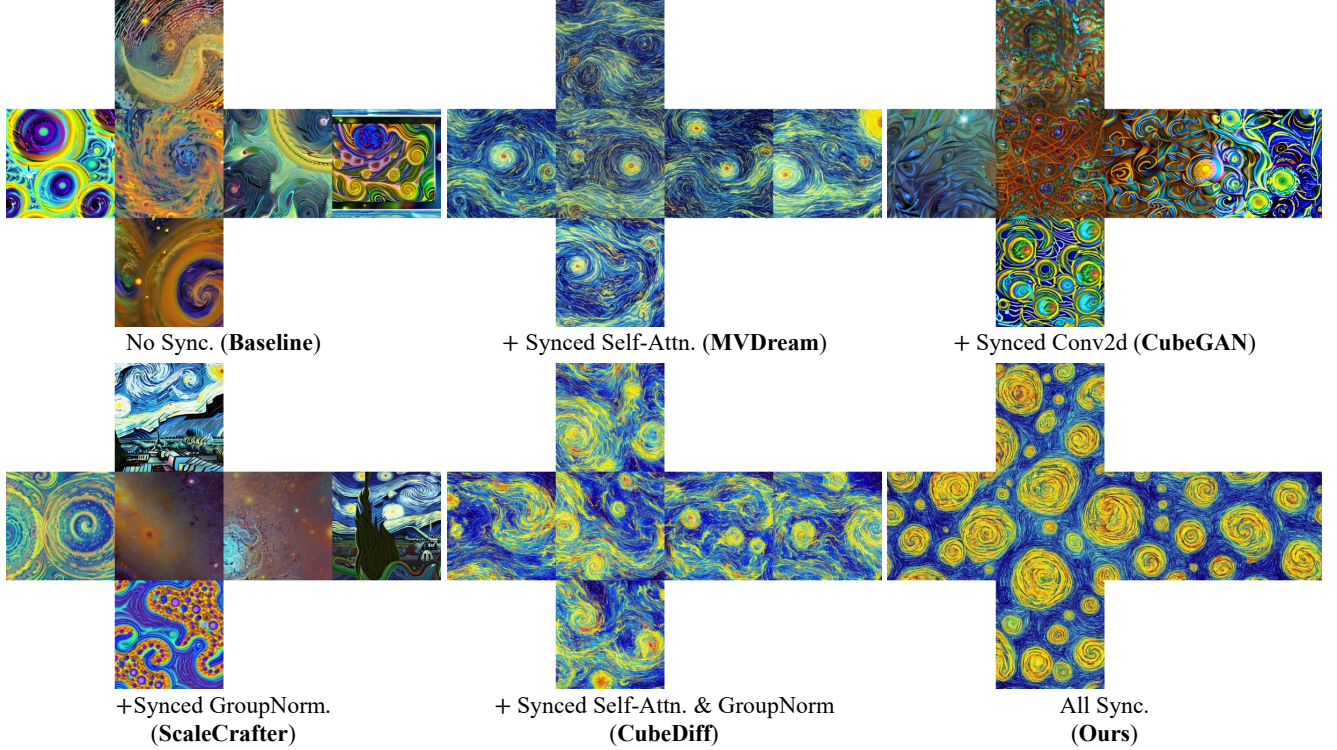


Figure 10. **Ablation analysis of Multi-plane Synchronization.** We adopt Stable-Diffusion v2 as baseline model for multi-plane generation, and the text prompt used is “The vast cosmos in the style of Van Gogh, with swirling patterns and vibrant colors.”

Table 4. **Ablation analysis of DreamCube**, where the performance evaluation is performed on the Structured3D test split [64]. Both proposed Multi-plane Synchronization and XYZ Positional encoding bring performance improvements.

Methods	RGB		Depth	
	FID ↓	IS ↑	δ -1.25 ↑	AbsRel ↓
w/o XYZ Pos.	13.66	5.57	0.784	0.136
w/o Sync.	21.35	5.62	0.684	0.168
w/o SyncSA	24.38	5.78	0.715	0.158
w/o SyncConv	19.62	5.60	0.779	0.139
w/o SyncGN	18.35	5.51	0.784	0.135
DreamCube (Ours)	12.58	5.50	0.787	0.133

incoherence compared to UV Positional Encoding.

Generalization analysis of DreamCube. To evaluate DreamCube’s generalization capabilities, we present out-domain generation results in Figure 12, where the input RGB images are generated from Flux.1-dev [27]. We obtain the corresponding input depth map using Depth Anything v2 [59]. Despite the significant domain gap between these inputs and our training distribution, DreamCube successfully generates coherent and visually plausible RGB-D

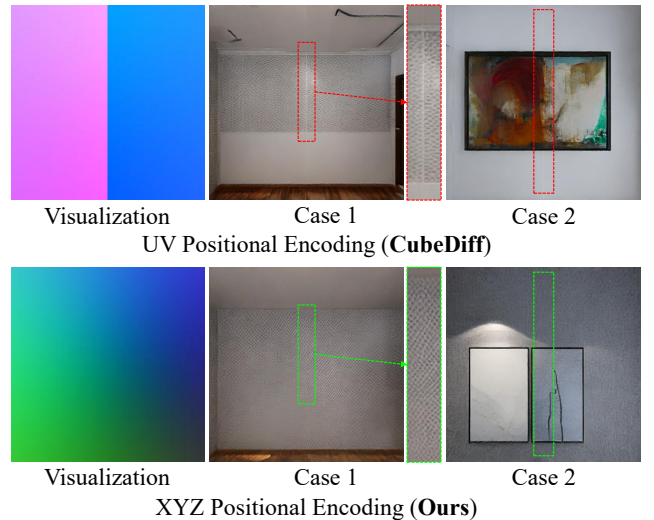


Figure 11. **Ablation analysis of XYZ Positional Encoding.** We present the qualitative results of the back view of cubemap, where the UV positional encoding introduces discontinuous numerical steps. This leads to line artifacts (Case 1) and incoherent visual contents (Case 2), as indicated by the **red** dashed box. In contrast, our proposed XYZ positional encoding alleviates these issues in both cases, as shown within the **green** dashed box.



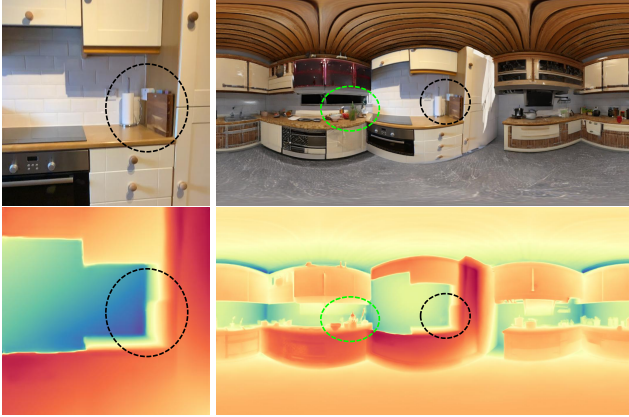
Figure 12. **Out-domain RGB-D panorama generation.** The RGB-D inputs are obtained by Flux.1-dev [27] and Depth Anything v2 [59]. DreamCube demonstrates generalization ability on diverse inputs, maintaining high-quality RGB appearance and geometric consistency.

panoramas, demonstrating its strong generalization ability.

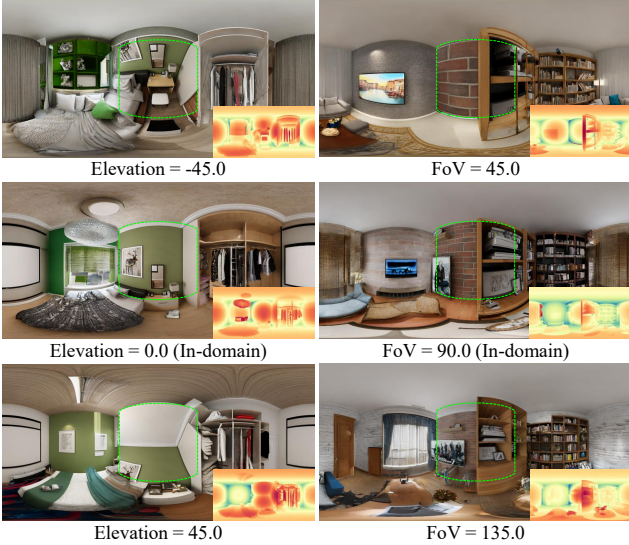
Robustness analysis of DreamCube. DreamCube takes single-view RGB-D images as input for cubemap generation. To evaluate the robustness of DreamCube, we test various types of RGB-D inputs and provide the generated results in Figure 13. Specifically, we test real-world inputs captured by sensors [3]. Unlike synthetic training data, real-world inputs have low-resolution depth maps and non-standard camera views. Even so, our method is still able to generate reasonable panoramas with high-resolution depth

maps, as shown in Figure 13a. In addition, we also test inputs with extreme camera views (*e.g.*, elevation and FoV). DreamCube struggles to generate correct panoramas under inputs with extreme elevation angles, but shows robustness to perturbations of the FoV, as shown in Figure 13b.

Efficiency analysis. We provide an efficiency analysis of our approach in Table 5 compared to the baseline model, Stable Diffusion v2 (SD2) [41]. Among all synchronized operators, synchronized Self-Attention (“+SyncSA”) incurs the most computational cost, increasing TFLOPs by 76.1%



(a) Generated results from real-world inputs captured by sensors [3]. Even though the input depth is low-resolution (as indicated by the **black** dashed circles), our method is still able to generate high-definition depth maps (as indicated by the **green** dashed circles).



(b) Generated results from inputs with extreme viewing angles, where the **green** dashed boxes highlight the input views.

Figure 13. **Robustness analysis** of DreamCube to out-domain RGB-D inputs from real world and extreme viewing angles.

and latency (ms) by 113.1% than no synchronization (“No Sync.”). This accounts for 86.0% of the latency cost and almost 100% of the TFLOPs cost incurred by our approach.

6. Limitation

The limitations of our method include high computational cost and the restricted input conditions. First, DreamCube samples six image latents at a time, which incurs additional computational cost and hinders the scalability of training batches. Nevertheless, compared with existing panorama generation methods, our method has the superior computational utilization (effective pixel ratio obtained at the same

Table 5. **Efficiency analysis.** We evaluate the computational efficiency of SD2’s and DreamCube’s U-Nets in a single forward pass and report the metrics: TFLOPs and Latency (ms).

Methods		FLOPs (T)	Latency (ms)
SD2’s U-Net	batch-size=1	0.804	35.4
	batch-size=6	4.826	138.8
DreamCube’s U-Net	No Sync.	4.827	139.4
	+SyncSA	8.502	297.1
	+SyncConv	4.827	144.3
	+SyncGN	4.827	152.0
	All Sync.	8.502	322.7

computational cost), because it uses a less distorted cube-map instead of equirectangular, and does not require redundant FoV overlapping for seam continuity. Second, DreamCube is trained with cubemap’s front face as input conditions. When the input distribution deviates from the training domain, for example, non-frontal view or extreme FoV, the generated results may fail, as shown in Figure 13.

7. Conclusion

In this work, we thoroughly analyze the limitations of existing 2D diffusion models when applied to multi-plane panoramic representations, and propose a multi-plane synchronization strategy to painlessly generalize pre-trained 2D diffusion models to the omnidirectional image domain. This strategy ensures translation equivariance in the omnidirectional image domain by adapting the spatial operators in the model without the need for additional fine-tuning or constructing overlapping views. Benefiting from multi-plane synchronization, we further introduce DreamCube, which jointly models RGB and depth cube maps by leveraging image priors from pre-trained 2D diffusion. Extensive experiments show that the proposed approach outperforms previous equirectangular-based RGB-D panoramic generation methods and holds potential in panoramic depth estimation and 3D scene generation.

References

- [1] Naofumi Akimoto, Yuhi Matsuo, and Yoshimitsu Aoki. Diverse Plausible 360-Degree Image Outpainting for Efficient 3DCG Background Creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 7
- [2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. In *International Conference on Machine Learning*, pages 1737–1752. PMLR, 2023. 3
- [3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARK-

- itScenes - A Diverse Real-World Dataset for 3D Indoor Scene Understanding Using Mobile RGB-D Data. In *Neural Information Processing Systems*, 2021. 11, 12
- [4] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 2, 3
- [5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–119, 2018. 7
- [6] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *European Conference on Computer Vision*, pages 432–449. Springer, 2024. 3
- [7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 3
- [8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1
- [9] Mengyang Feng, Jinlin Liu, Miaomiao Cui, and Xuansong Xie. Diffusion360: Seamless 360 Degree Panoramic Image Generation based on Diffusion Models. *arXiv preprint arXiv:2311.13141*, 2023. 1, 2, 7
- [10] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision*, pages 241–258. Springer, 2024. 2, 3
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3
- [12] Yuliang Guo, Sparsh Garg, S Mahdi H Miangoleh, Xinyu Huang, and Liu Ren. Depth any camera: Zero-shot metric depth estimation from any camera. *arXiv preprint arXiv:2501.02464*, 2025. 3, 7, 8
- [13] hdri skies. HDRIs. <https://hdri-skies.com/>, accessed 02/2025. 6
- [14] hdri skies. HDRIs. <https://www.ihdri.com/hdri-skies-outdoor/>, accessed 02/2025. 6
- [15] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 2
- [16] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. ScaleCrafter: Tuning-free Higher-Resolution Visual Generation with Diffusion Models. In *International Conference on Learning Representations*, 2024. 4
- [17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 6
- [19] Xinya Ji, Gaspard Zoss, Prashanth Chandran, Lingchen Yang, Xun Cao, Barbara Solenthaler, and Derek Bradley. Joint learning of depth and appearance for portrait image animation. *arXiv preprint arXiv:2501.08649*, 2025. 3
- [20] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21741–21752, 2023. 3
- [21] Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. Cubediff: Repurposing diffusion-based image models for panorama generation, 2025. 1, 2, 3, 4, 6
- [22] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4
- [23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 2023. 6, 9
- [24] Rawal Khiradkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. *arXiv preprint arXiv:2408.12569*, 2024. 3
- [25] Muhammed Kocabas, Chun-Hao P Huang, Joachim Tesch, Lea Müller, Otmar Hilliges, and Michael J Black. SPEC: Seeing People in the Wild With an Estimated Camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11035–11045, 2021. 6
- [26] Akshay Krishnan, Xinchun Yan, Vincent Casser, and Abhijit Kundu. Orchid: Image latent diffusion for joint appearance and geometry generation. *arXiv preprint arXiv:2501.13087*, 2025. 2, 3
- [27] Black Forest Labs. Flux.1-dev. <https://huggingface.co/black-forest-labs/FLUX.1-dev>, 2025. Accessed: 2025-01-19. 10, 11
- [28] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2025. Accessed: 2025-01-19. 1
- [29] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1119–1127, 2015. 3

- [30] Jialu Li and Mohit Bansal. PanoGen: Text-Conditioned Panoramic Environment Generation for Vision-and-Language Navigation. In *International Conference on Neural Information Processing Systems*, 2023. 1, 3
- [31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 7
- [32] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. *arXiv preprint arXiv:2412.14015*, 2024. 7
- [33] Aoming Liu, Zhong Li, Zhang Chen, Nannan Li, Yi Xu, and Bryan A Plummer. Panofree: Tuning-free holistic multi-view image generation with cross-view self-guidance. In *European Conference on Computer Vision*, pages 146–164. Springer, 2024. 3
- [34] Xian Liu, Jian Ren, Aliaksandr Siarohin, Ivan Skorokhodov, Yanyu Li, Dahua Lin, Xihui Liu, Ziwei Liu, and Sergey Tulyakov. Hyperhuman: Hyper-realistic human generation with latent structural diffusion. *arXiv preprint arXiv:2310.08579*, 2023. 3
- [35] Zhiheng Liu, Ka Leong Cheng, Qiuyu Wang, Shuzhe Wang, Hao Ouyang, Bin Tan, Kai Zhu, Yujun Shen, Qifeng Chen, and Ping Luo. Depthlab: From partial to complete. *arXiv preprint arXiv:2412.18153*, 2024. 5
- [36] Christopher May and Daniel Aliaga. CubeGAN: Omnidirectional Image Synthesis Using Generative Adversarial Networks. In *Computer Graphics Forum*, pages 213–224. Wiley Online Library, 2023. 3, 4
- [37] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4
- [38] Emil Persson. Texture from Humus. <https://www.humus.name/index.php?page=Textures>, accessed 02/2025. 6
- [39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *International Conference on Learning Representations*, 2024. 4
- [40] polyhaven.com. HDRIs. <https://polyhaven.com/hdri/>, accessed 02/2025. 6
- [41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3, 4, 5, 6, 11
- [42] Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*, 2022. 5
- [43] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 7
- [44] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3
- [45] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 3
- [46] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view Diffusion for 3D Generation. In *International Conference on Learning Representations*, 2024. 4
- [47] Gowri Somanath and Daniel Kurz. Hdr environment map estimation for real-time augmented reality. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11298–11306, 2021. 1
- [48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021. 6
- [49] Gabriela Ben Melech Stan, Diana Wofk, Estelle Aflalo, Shao-Yen Tseng, Zhipeng Cai, Michael Paulitsch, and Vasudev Lal. LDM3D-VR: Latent Diffusion Model for 3D VR. *arXiv preprint arXiv:2311.03226*, 2023. 2, 3, 5, 7
- [50] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, et al. LDM3D: Latent Diffusion Model for 3D. *arXiv preprint arXiv:2305.10853*, 2023. 2, 7, 8
- [51] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. MVDiffusion: Enabling Holistic Multi-view Image Generation with Correspondence-Aware Diffusion. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 7
- [52] Hai Wang, Xiaoyu Xiang, Yuchen Fan, and Jing-Hao Xue. Customizing 360-degree panoramas through text-to-image diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4933–4943, 2024. 1, 3
- [53] Jionghao Wang, Ziyu Chen, Jun Ling, Rong Xie, and Li Song. 360-degree panorama generation from few unregistered nfov images. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6811–6821, 2023. 1, 3
- [54] Ning-Hsu Albert Wang and Yu-Lun Liu. Depth Anywhere: Enhancing 360 Monocular Depth Estimation via Perspective Distillation and Unlabeled Data Augmentation. *Advances in Neural Information Processing Systems*, 37: 127739–127764, 2024. 7, 8
- [55] Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. PanoDiffusion: 360-degree Panorama Outpainting via Diffusion. In *International Conference on Learning Representations*, 2024. 2, 3, 5, 6, 7, 8
- [56] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2695–2702. IEEE, 2012. 6, 7

- [57] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 675–684, 2018. [3](#)
- [58] Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. Dreamspace: Dreaming your room space with text-driven panoramic texture propagation. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 650–660. IEEE, 2024. [1](#)
- [59] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. [7](#), [10](#), [11](#)
- [60] Weicai Ye, Chenhao Ji, Zheng Chen, Junyao Gao, Xiaoshui Huang, Song-Hai Zhang, Wanli Ouyang, Tong He, Cairong Zhao, and Guofeng Zhang. Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. *arXiv preprint arXiv:2410.24203*, 2024. [1](#), [3](#)
- [61] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024. [6](#)
- [62] Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming Stable Diffusion for Text to 360 Panorama Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6347–6357, 2024. [1](#), [2](#), [3](#), [7](#)
- [63] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5729–5739, 2023. [3](#)
- [64] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. [6](#), [7](#), [9](#), [10](#)
- [65] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *Advances in Neural Information Processing Systems*, 37:77799–77830, 2024. [1](#)