

JoPano: Unified Panorama Generation via Joint Modeling

Wancheng Feng^{1,3*} Chen An^{1,2*} Zhenliang He¹✉ Meina Kan^{1,2} Shiguang Shan^{1,2} Lukun Wang³

¹State Key Laboratory of AI Safety, Institute of Computing Technology, CAS, China

²University of Chinese Academy of Sciences (CAS), China

³Shandong University of Science and Technology, China

<https://VIPL-GENUN.github.io/Project-JoPano>

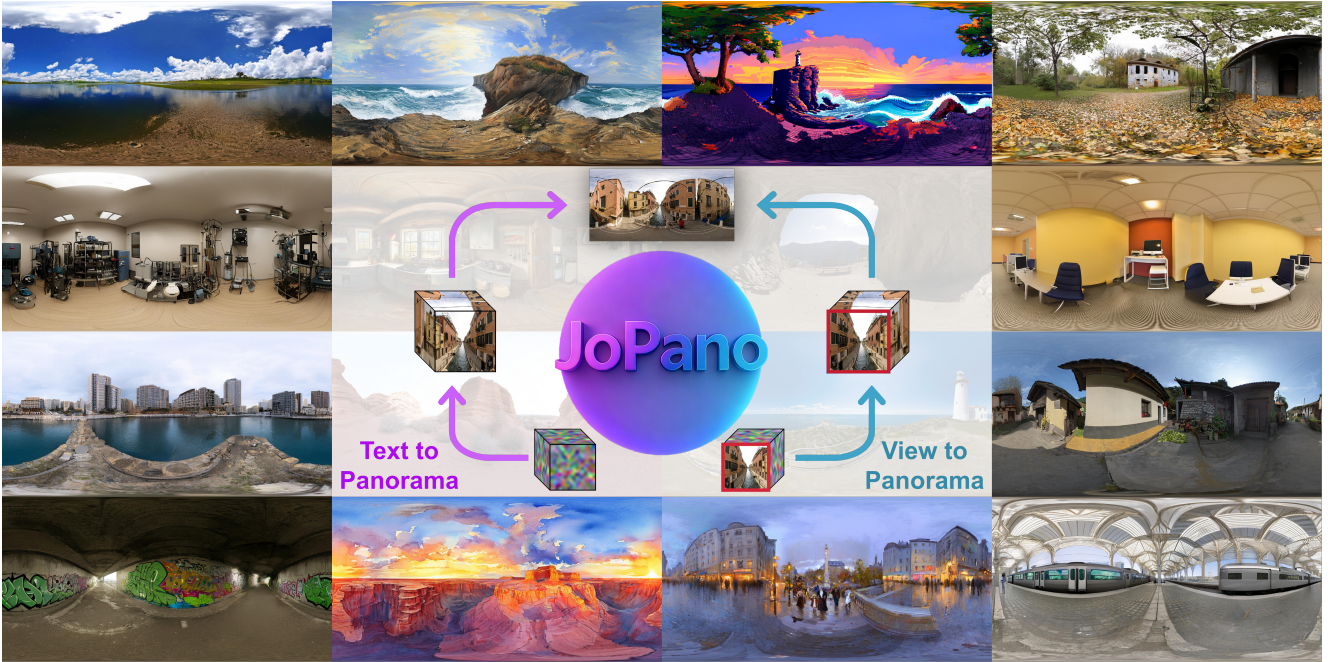


Figure 1. We propose JoPano, a unified panorama generation framework that supports both text-to-panorama (T2P) and view-to-panorama (V2P). The left eight examples show T2P results, while the right eight show V2P results. JoPano generates high-quality panoramas across indoor, outdoor, and stylized scenes.

Abstract

Panorama generation has recently attracted growing interest in the research community, with two core tasks, text-to-panorama and view-to-panorama generation. However, existing methods still face two major challenges: their U-Net-based architectures constrain the visual quality of the generated panoramas, and they usually treat the two core tasks independently, which leads to modeling redundancy and inefficiency. To overcome these challenges, we propose a **joint-face panorama (JoPano)** generation approach that unifies the two core tasks within a DiT-based model. To transfer the rich generative capabilities of existing DiT

backbones learned from natural images to the panorama domain, we propose a Joint-Face Adapter built on the cube-map representation of panoramas, which enables a pre-trained DiT to jointly model and generate different views of a panorama. We further apply Poisson Blending to reduce seam inconsistencies that often appear at the boundaries between cube faces. Correspondingly, we introduce Seam-SSIM and Seam-Sobel metrics to quantitatively evaluate the seam consistency. Moreover, we propose a condition switching mechanism that unifies text-to-panorama and view-to-panorama tasks within a single model. Comprehensive experiments show that JoPano can generate high-quality panoramas for both text-to-panorama and view-to-panorama generation tasks, achieving state-of-the-art performance on FID, CLIP-FID, IS, and CLIP-Score metrics.

* Equal contribution. ✉ Corresponding author.

1. Introduction

A panorama is a 2D representation of a scene that can cover the entire 360° field of view. It has been widely adopted in interactive and immersive applications such as Virtual Reality and Augmented Reality [3, 67], and it has also emerged as a promising representation for World Models [49, 61, 62]. However, the acquisition of real-world panorama data relies on specialized equipment [5, 13, 57], making large-scale collection expensive and challenging. Therefore, the synthesis of panoramas has emerged as a significant research focus [19, 20, 56, 60, 63, 65], particularly facilitated by recent advances in diffusion models [11, 18, 22, 25, 33, 35, 38, 42] for generating high-quality visual content. Nevertheless, current panorama generation methods still encounter challenges related to generation quality and modeling efficiency.

Challenge 1: The visual quality of generated panoramas remains limited in terms of resolution and detail. Most existing methods are based on the U-Net architecture [35, 38], which limits their ability to generate high-quality results. Recently, diffusion transformers (DiT) [11, 22, 33, 58] have demonstrated strong generative capacity in the natural image domain, providing promising foundation models for panorama generation [19, 20, 56, 60, 63, 65]. However, most DiT backbones are designed for and pretrained on natural images, which exhibit a significant domain gap compared to panoramic images [19, 20]. Therefore, a key problem arises: how to adapt a pretrained DiT from the natural image domain to the panorama image domain while maintaining its capability for high-quality generation as well as other practical functionalities such as stylized generation.

Challenge 2: Although the two core tasks of panorama generation are closely related, they have been developed independently, resulting in modeling redundancies and inefficiencies. Panorama generation comprises two core tasks: 1) text-to-panorama (T2P), where the model generates a panorama according to a textual description [23, 31, 48, 56, 59, 60, 63], and 2) view-to-panorama (V2P), where the model completes a panorama given a narrow field of view [1, 19, 20, 30, 53, 65]. Previous methods typically design specific solutions for each task independently. However, since both tasks can be regarded as conditional generation problems within the panorama domain, they may share intrinsic commonalities in their generative mechanisms. Therefore, we believe there is potential to integrate these two tasks under a unified paradigm, thereby reducing the number of models and improving modeling efficiency.

In this paper, we introduce **JoPano** to address the aforementioned challenges, achieving high-quality and efficient panorama generation.

For Challenge 1: To improve the generation quality, we choose Sana [58], an efficient DiT architecture, as our backbone model. We represent a panorama as six cube faces

using a cubemap projection [15, 19, 20], where each face corresponds to a perspective image of the scene. Then, our goal is to learn to simultaneously generate all cube faces. The core problem is: how to model the relationship and coherence among the six faces based on the pretrained Sana backbone. To this end, we introduce *Joint-Face Adapter* modules into the backbone, which apply normalization and full attention across all cube faces to jointly model their features and facilitate their interaction. Besides, we optimize only the adapter modules without altering the parameters of the Sana backbone, thereby preserving its original capabilities, such as high-quality generation and stylized generation. Furthermore, to mitigate the seam inconsistencies that often appear at the boundaries between cube faces [19], we apply Poisson Blending [34], which effectively smooths the transitions between adjacent faces. Correspondingly, we introduce Seam-SSIM and Seam-Sobel metrics to quantitatively evaluate the seam consistencies. Overall, we achieve high-quality and seamless panorama generation.

For Challenge 2: To reduce modeling redundancy and enhance efficiency, we propose a condition switching mechanism that unifies T2P and V2P tasks within a single diffusion model. This mechanism allows the model to flexibly switch between the two tasks by changing only the conditioning inputs. Under this unified setting, for the T2P task, the model simultaneously generates all six cubemap faces based on a text condition, where all faces are initialized with noise and denoised through a diffusion process. For the V2P task, one cubemap face is provided as a view condition, and the model generates the remaining five faces from noise under the same diffusion process. Correspondingly, the diffusion loss is computed on all six faces for T2P and on the five generated faces for V2P during training. In this manner, JoPano can flexibly and efficiently switch between the two tasks, eliminating the redundancy of separate modeling.

Our contributions can be summarized as follows:

- We propose a Joint-Face Adapter to transfer the generative capabilities of Sana-DiT from the natural image domain to the panorama domain, achieving high-quality and style-rich panorama generation.
- We unify text-to-panorama and view-to-panorama generation within a single diffusion framework via a condition-switching mechanism, enhancing the modeling efficiency.
- Our model achieves the state-of-the-art performance in panorama generation, surpassing existing methods in both visual quality and quantitative evaluations.

2. Related Work

2.1. Panorama Representations

Panoramas are typically represented using two projection formats: the *equiangular projection* (ERP) and the *cubemap projection* (CMP).

Equirectangular Projection The ERP projects a spherical panorama onto a 2:1 rectangular image, where the horizontal and vertical axes correspond to longitude and latitude, respectively. Due to its simplicity, compact storage, and compatibility with standard image formats, ERP has been widely adopted for representing 360° panoramas [1, 3, 7, 8, 23, 27, 31, 46, 48, 50, 51, 63, 65, 67, 68].

Cubemap Projection The CMP projects the spherical panorama onto the six square faces of a cube, each showing a 90° field of view in a different direction. To mitigate the domain gap introduced by ERP, recent studies have explored panorama generation using cubemap [19, 20, 43, 60].

2.2. Panorama Generation

Current panorama generation methods can be categorized into text-to-panorama and view-to-panorama approaches. Early text-to-panorama methods [7, 29] relied on GANs [14], while diffusion models [12, 52] have recently enabled more sophisticated panorama generation. PanoGen [23] and MVDiffusion [48] generate panoramas via recursive outpainting based on pretrained text-to-image diffusion models. DiffPano [60] instead generates panoramas directly by fine-tuning a diffusion model with LoRA. However, the outputs of these methods still suffer from geometric distortions due to the domain gap between natural images and panorama images. To alleviate this issue, some methods [32, 47, 63] adopt dual-branch architectures within diffusion models, which partially mitigate distortions but introduce substantial computational overhead. SMGD [46] introduces a spherical manifold convolution to guide the diffusion process and reduce distortion. In contrast, PAR [50] employs an autoregressive model [9] to avoid the misalignment between ERP representations and diffusion models. For the view-to-panorama task, early works typically follow an outpainting paradigm [1, 8, 27, 51]. More recent works [19, 20, 43, 64] leverage cubemap representations instead of ERP to mitigate geometric distortions. Among them, CubeDiff [20] generates the six perspective views of a cubemap in parallel, while Dream-Cube [19] proposes a multi-plane synchronization strategy to alleviate discontinuous seams and inconsistent color tones in cubemap panorama generation. Nevertheless, these methods still inevitably produce seam inconsistencies along cubemap face boundaries. Despite these advances in projection design and task formulation, most panorama generation methods still rely on the U-Net-based architecture as the backbone, which limits the overall generation quality. A concurrent work, HunyuanWorld [49], addresses this limitation by building on Flux [22] and training with large-scale data, achieving higher-fidelity panorama generation.

2.3. Diffusion Model

Diffusion models [10, 17, 18, 21, 25, 37, 42, 44] are powerful generative methods that synthesize data by reversing a process that gradually adds noise. To further improve efficiency, Latent Diffusion Models [38] perform the diffusion process in a compact latent space, significantly reducing computational cost while enabling high-quality generation. In terms of model architecture, the DiT-based [11, 22, 28, 33, 58] has recently outperformed traditional U-Net-based designs [35, 38, 39, 41] in both scalability and generative quality.

3. Method

We represent a panorama as six cubemap faces $\{f_i\}_{i=0}^5$ [15, 19, 20]. For T2P task, the model generates all faces from six noise $\{\epsilon_i \sim \mathcal{N}(0, 1)\}_{i=0}^5$ and a text prompt c_{text} :

$$\{f_i\}_{i=0}^5 = G(\epsilon_0, \epsilon_1, \dots, \epsilon_5, c_{text}). \quad (1)$$

For V2P task, one face f_0 is given as a view condition, and the remaining five faces are synthesized from f_0 , five noise $\{\epsilon_i \sim \mathcal{N}(0, 1)\}_{i=1}^5$, and c_{text} :

$$\{f_i\}_{i=0}^5 = G(f_0, \epsilon_1, \dots, \epsilon_5, c_{text}). \quad (2)$$

In the following sections, we introduce Joint-Face Adapter to extend a DiT model to the panorama domain and a condition switching mechanism to equip a single diffusion model with both T2P and V2P capabilities. The overall pipeline is shown in Fig. 2.

3.1. Joint-Face Adapter

To enforce cross-face interaction, we introduce the Joint-Face Adapter, which extends Sana [58] to the panorama domain. Specifically, we insert a Joint-Face Adapter into each DiT block, after the original linear attention and before the cross-attention layer, enabling the model to process all six cubemap faces simultaneously.

Joint Modeling We treat each cubemap face as an individual image in the DiT backbone. Given tokens of all faces as $\mathbf{z} \in \mathbb{R}^{(B \times 6) \times N \times C}$, where B is the number of panoramas in a batch, N is the number of tokens per face, and C is the channel dimension, we reshape them into $\hat{\mathbf{z}} \in \mathbb{R}^{B \times (N \times 6) \times C}$ to concatenate the six cube faces of each panorama along the token dimension. We first apply Layer Normalization [2] to $\hat{\mathbf{z}}$, and denote the output as $\mathbf{y} \in \mathbb{R}^{B \times (N \times 6) \times C}$. The normalization is shared across all six faces. To explicitly model cross-face interactions, we then apply full attention over the token sequence \mathbf{y} , allowing each token to attend to all tokens of all faces. We obtain the Query, Key, and Value via linear projections of the normalized features \mathbf{y} :

$$\mathbf{Q} = W_Q \mathbf{y}, \quad \mathbf{K} = W_K \mathbf{y}, \quad \mathbf{V} = W_V \mathbf{y}. \quad (3)$$

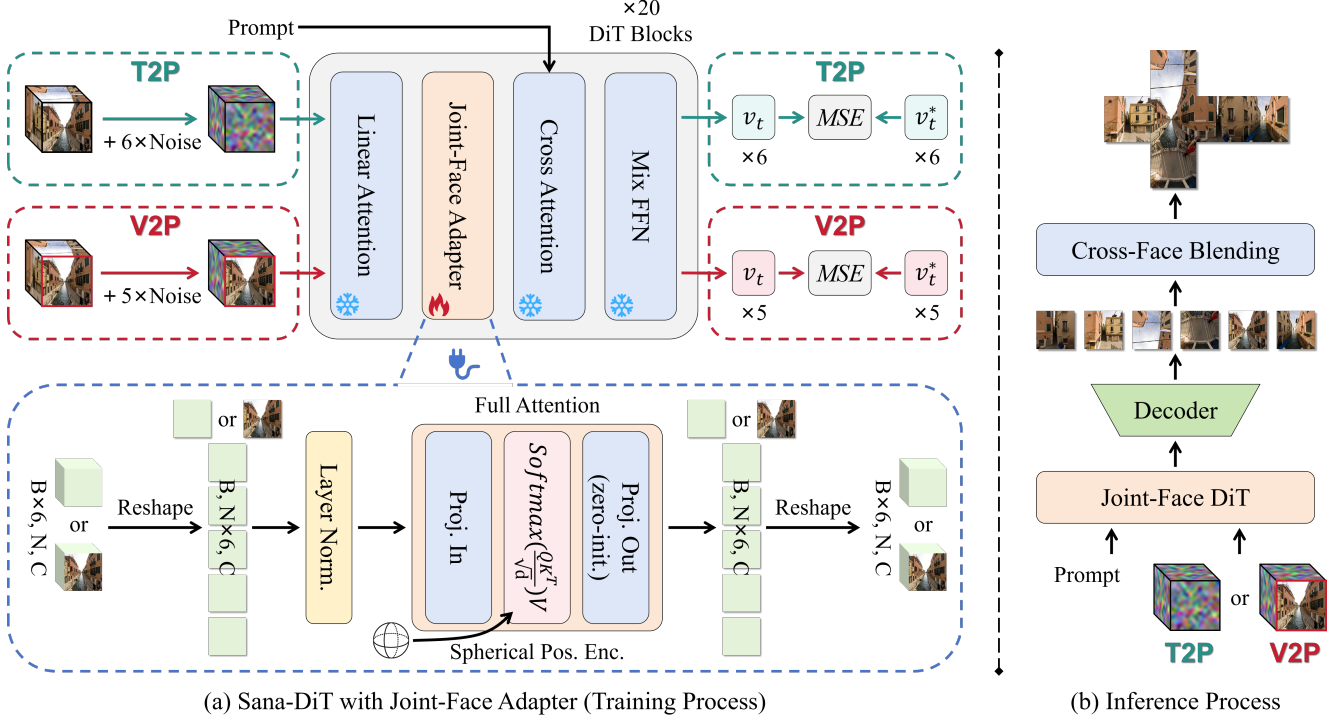


Figure 2. Overview of the JoPano pipeline. (a) Training process. The Joint-Face Adapter is inserted into Sana-DiT to jointly model all six cubemap faces, and a single diffusion process is shared by T2P and V2P. (b) Inference process. The Joint-Face DiT generates the cubemap faces, and the Cross-Face Blender further refines the results across faces.

where W_Q , W_K , and W_V are projection matrices, and then the full attention is computed as

$$\hat{\mathbf{z}}' = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \quad (4)$$

The output $\hat{\mathbf{z}}' \in \mathbb{R}^{B \times (N \times 6) \times C}$ is reshaped back to $\mathbf{z}' \in \mathbb{R}^{(B \times 6) \times N \times C}$, so that the updated features can be fed back into the pretrained diffusion backbone without changing its original shape.

3D Spherical Positional Embedding To better match the geometry of panoramas, we use the unit sphere for position embedding. For each token on the cubemap faces, we place a unit sphere inside the cube and take the intersection of the line from the cube center to the token’s location on the face as its 3D coordinate (x, y, z) . We then feed this unit 3D direction into RoPE [45] to embed positional information into the queries and keys.

Adapter Only Optimization During training, we freeze the pretrained diffusion backbone and train only the Joint-Face Adapter, with its output projection zero-initialized. This design makes the model behave like the pretrained diffusion backbone at the start of training, and then gradually

adapt to panorama generation as the adapter learns cross-face dependencies.

3.2. Unified Generation

Condition Switching During training, following Rectified Flow [26], the model predicts the velocity field from cubemap faces perturbed with Gaussian noise and the given condition. For each face f_i , the noisy sample at timestep t is defined as

$$f_{i,t} = (1-t)f_i + t\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \quad t \in (0, 1). \quad (5)$$

To train a single model for both T2P and V2P, we introduce a condition switching mechanism that selects the input configuration for each training sample. Specifically, we set a binary switch $\gamma \in \{0, 1\}$, applied per training sample. In training, we set $\gamma = 0$ (T2P) or $\gamma = 1$ (V2P) with equal probability (0.5/0.5), ensuring that both tasks are balanced. When $\gamma = 0$, the model is conditioned only on the text c_{text} and predicts the velocity field as

$$v_\theta(f_{0,t}, f_{1,t}, \dots, f_{5,t}, t, c_{text}). \quad (6)$$

When $\gamma = 1$, the model is conditioned on a fixed face f_0 and the text c_{text} and predicts the velocity field as

$$v_\theta(\underline{f}_0, f_{1,t}, \dots, f_{5,t}, t, c_{text}). \quad (7)$$

This fixed-face design avoids sampling over different faces and matches our implementation.

Loss Function We optimize the model with an MSE loss over the supervised faces:

$$\mathcal{L} = \mathbb{E}_{t,\gamma} \left[\frac{1}{6-\gamma} \sum_{i=\gamma}^5 \|v_{\theta}^{(i)} - v^{*(i)}\|_2^2 \right], \quad (8)$$

where $v_{\theta}^{(i)}$ denotes the predicted velocity for face f_i , and $v^{*(i)} = \epsilon - f_i$ is the corresponding ground truth velocity. When $\gamma = 0$, all six faces are supervised; when $\gamma = 1$, f_0 serves as a clean view condition and remaining five faces are supervised.

3.3. Cross-Face Seam Handling

Cross-Face Blending To alleviate seam inconsistencies between cubemap faces, we apply Poisson Blending [34], which obtains the composite image by solving a Poisson equation with Dirichlet boundary conditions:

$$\Delta f = \text{div } \mathbf{v} \text{ in } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \quad (9)$$

In our setting, we apply it independently to each cubemap face g_i . For face g_i , let Ω_i denote its image domain and \mathbf{v}_i the gradient field. For every edge shared by g_i and its neighboring face g_j , we extract from both faces a narrow band of width one pixel along the edge, and set the Dirichlet boundary values on $\partial\Omega_i$ to the pixelwise average of these two bands.

$$\begin{cases} \Delta f_i = \text{div } \mathbf{v}_i & \text{in } \Omega_i, \\ f_i = \frac{1}{2}(g_i + g_j) & \text{on } \partial\Omega_i, \end{cases} \quad (10)$$

where $j = \text{nbr}(i, e)$, $e \in \{N, S, E, W\}$, denotes the face adjacent to g_i in the direction e . The solution f_i serves as the blended version of face g_i , yielding a cubemap with significantly reduced cross-face seams.

Seam Consistency Metrics To quantify seam inconsistencies between cubemap faces, we introduce two metrics: Seam-SSIM, based on SSIM [55], and Seam-Sobel, based on image gradients.

For each cube edge $e \in \{1, \dots, 12\}$, we take narrow boundary bands $B_e^{(L)}$ and $B_e^{(R)}$ (width 1% of the face) from the two faces on either side of the edge and Seam-SSIM is defined as

$$\text{Seam-SSIM} = \frac{1}{12} \sum_{e=1}^{12} \text{SSIM}(B_e^{(L)}, B_e^{(R)}), \quad (11)$$

where higher values indicate better seam consistency.

For Seam-Sobel, we apply an x-direction Sobel operator to the two faces and take the column of Sobel gradients

closest to the shared edge on each side, denoted by $c_e^{(L)}$ and $c_e^{(R)}$. We define

$$\text{Seam-Sobel} = \frac{1}{12} \sum_{e=1}^{12} \frac{\text{mean}|c_e^{(L)}| + \text{mean}|c_e^{(R)}|}{2}, \quad (12)$$

where lower values indicate better seam consistency.

4. Experiment

4.1. Experimental Setup

Implementation Details We use SANA-1.5 (1.6B) as the base model; after adding the Joint-Face Adapter, JoPano contains about 2B parameters. We train with a learning rate of 1×10^{-4} for 1M steps on 8 Nvidia RTX A100-40G GPUs with a batch size of 1 per GPU. We adopt a cubemap resolution of $512 \times 512 \times 6$ and convert the generated cubemaps to ERP panoramas at 2048×1024 for visualization.

Dataset We use the Structure3D [66] and SUN360 [57] datasets for training, containing 41,930 panoramas in total. Following [19, 20, 56], we divide the Structure3D dataset into 16,930 panoramas for training, 2,116 for validation, and 2,117 for testing, and we use Qwen2.5-VL [4] to caption each panorama. For SUN360, we adopt the version provided by PanoDecouple [65], which contains 25,000 training and 4,260 testing panoramas paired with their corresponding text descriptions. We use the test set of 2,117 panoramas from Structure3D (mostly indoor scenes) and 4,260 panoramas from SUN360 (mostly outdoor scenes) to evaluate the performance of T2P and V2P, respectively.

Evaluation Metrics We evaluate our method using six metrics. To assess image quality, we report FID [16], CLIP-FID (CF), and IS [40]. To evaluate text-image alignment, we use CLIP-Score (CS) [36]. In addition, we use Seam-SSIM and Seam-Sobel to evaluate seam consistency, but we only report these two metrics when comparing with DreamCube [19], since they are specifically designed for cubemap panorama generation.

4.2. Compare With Other Works

We perform both quantitative and qualitative comparisons. In the T2P setting, we compare JoPano with PanFusion [63], SMGD [46], and PAR [50]. In the V2P setting, we compare with Diffusion360 [53], PanoDiffusion [56], and DreamCube [19]. PanoDiffusion is an outpainting method, so its input consists only of a view image without any text prompt. For DreamCube, we follow the configuration in the original paper and feed the model with a view image, its corresponding depth map, and six separate text prompts as inputs.



Figure 3. Comparison of JoPano with other T2P methods. The first row shows outdoor scenes, the second row shows indoor scenes, and the third row shows a stylized scene, all generated from text prompts.

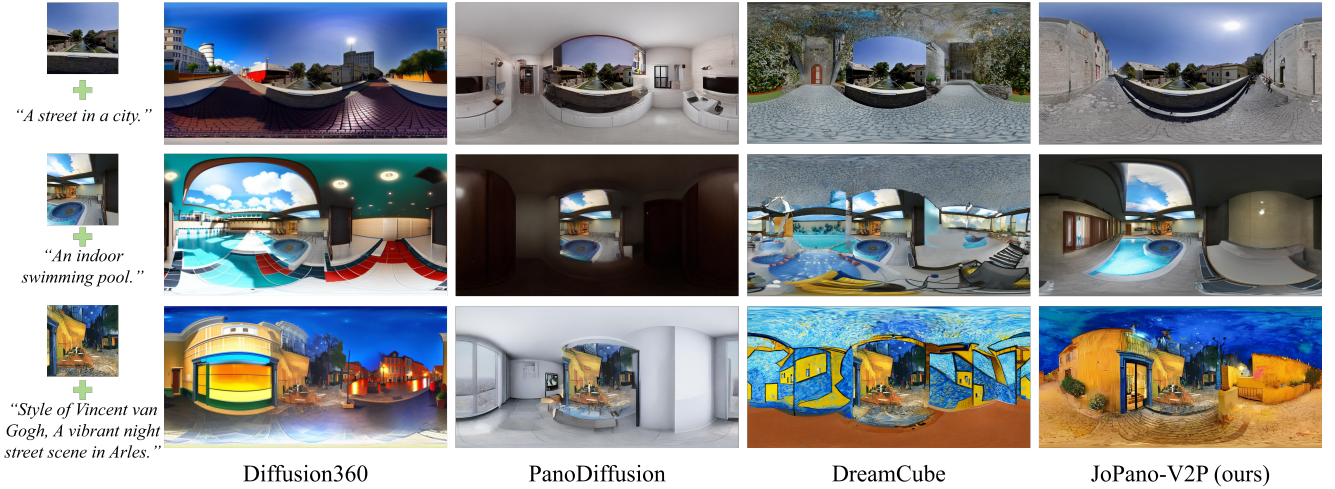


Figure 4. Comparison of JoPano with other V2P methods. The first row shows outdoor scenes, the second row shows indoor scenes, and the third row shows a stylized scene, all generated from view conditions.

Quantitative Results We quantitatively compare our method with several panorama generation approaches. We evaluate generation quality using FID, CLIP-FID, IS, and CLIP-Score. As shown in Tab. 1, JoPano consistently outperforms existing methods on both SUN360 and Structure3D, achieving the best FID, CLIP-FID, and CLIP-Score for both T2P and V2P, while maintaining competitive IS. Note that we do not report CLIP-Score for PanoDiffusion or DreamCube. PanoDiffusion has no text input, and DreamCube requires six separate per-face text prompts; computing CLIP-Score under our single-prompt setting would be inconsistent and unfair. In summary, these results demonstrate that JoPano delivers better panorama generation quality, validating the effectiveness of our unified joint modeling framework.

Qualitative Results We further provide qualitative comparisons to evaluate the effectiveness of our method. We evaluate both indoor and outdoor scenes and include a stylized example to demonstrate style controllability. As shown in Fig. 3 and Fig. 4, JoPano produces panoramas with sharp details, few distortions, improved seam consistency, and good text-image alignment in both T2P and V2P settings. In the T2P comparison, only PAR and JoPano produce stylized results, and JoPano shows clearer brush strokes and better matches the target style. In the V2P comparison, Diffusion360 and DreamCube can also generate stylized panoramas, but JoPano better matches both the input view and the requested style. These results indicate that JoPano preserves the base model’s stylized image generation ability in the panorama setting.

Table 1. Quantitative comparison on SUN360 and Structure3D in terms of FID, CLIP-FID (CF), IS, and CLIP-Score (CS). For the T2P task, our method achieves state-of-the-art results on all metrics except the IS score on Structure3D, while for the V2P task it achieves state-of-the-art results on all metrics.

Task	Methods	SUN360				Structure3D			
		FID ↓	CF ↓	IS ↑	CS ↑	FID ↓	CF ↓	IS ↑	CS ↑
T2P	PanFusion	30.92	23.76	7.12	29.62	48.53	23.34	3.97	22.31
	SMGD	48.87	38.85	4.77	17.09	53.34	27.97	3.37	23.73
	PAR	33.60	20.90	6.57	28.77	52.27	27.96	3.76	27.39
	JoPano (ours)	29.83	10.95	7.80	30.12	34.44	16.17	3.51	27.96
V2P	Diffusion360	33.12	24.88	6.34	25.85	36.68	13.83	2.67	25.28
	PanoDiffusion	125.33	37.42	3.48	-	22.47	7.78	3.00	-
	DreamCube	43.83	17.02	4.80	-	25.10	6.89	2.84	-
	JoPano (ours)	13.07	4.06	7.05	27.93	16.75	3.97	3.04	27.33

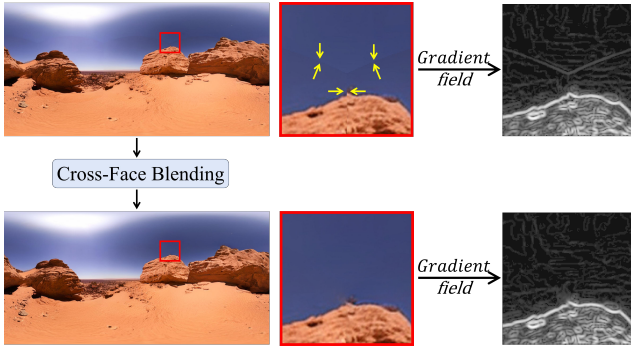


Figure 5. Comparison of ERP panoramas with and without Cross-Face Blending (CFB). The first row shows the panorama before CFB, with visible seam artifacts (red box). The second row shows the panorama after applying CFB, where the seams are smoothed. The improvement is more apparent in the gradient fields.

4.3. Ablation Study

Position Embedding We conduct an ablation study on different types of position embeddings. Under identical training settings, We compare two types of positional embeddings under identical training settings: (1) UV coordinates on each cubemap face, and (2) our 3D spherical embedding, both encoded with RoPE. In terms of quantitative image quality. As shown in Tab. 2, the spherical positional embedding outperforms the UV positional embedding, as the spherical representation better captures the geometry of panoramas.

Seam Consistency Comparison Cubemap panoramas are prone to seams at face boundaries. We evaluate these artifacts using our proposed metrics and assess the effect of Cross-Face Blending (CFB). To validate the metrics, we compute them on pure noise cubemaps and on the ground truth cubemaps converted from ERP panora-

Table 2. Image quality comparison of different positional encoding (PE) types on SUN360 dataset.

Task	PE Type	FID ↓	CF ↓	IS ↑
T2P	UV	52.26	14.15	9.86
	Sphere	48.72	13.13	10.21
V2P	UV	21.32	5.95	8.96
	Sphere	19.97	5.87	9.09

Table 3. Comparison of seam consistency on SUN360 dataset.

Method	Seam-SSIM ↑	Seam-Sobel ↓
Noise	0.004	129.62
Ground Truth	0.847	11.16
DreamCube	0.725	35.85
JoPano-T2P w/o CFB	0.762	39.69
JoPano-T2P w/ CFB	0.831	12.66
JoPano-V2P w/o CFB	0.786	41.16
JoPano-V2P w/ CFB	0.861	12.18

mas in the SUN360 test set. As shown in Tab. 3, noise cubemaps obtain very poor scores on both metrics, while ground truth achieves high Seam-SSIM (0.847) and low Seam-Sobel (11.16), confirming that the metrics correlate with seam smoothness. We then evaluate JoPano. Models with CFB outperform both their versions without CFB and DreamCube on the two metrics. In particular, JoPano-V2P with CFB nearly matches the ground truth seam quality. These results, together with the visual examples in Fig. 5, indicate that CFB effectively improves seam consistency.



Figure 6. Style panorama generation. T2P generates panoramas from text descriptions that include style, while V2P generates panoramas from stylized view conditions.



Figure 7. Multi-text generation. In T2P, JoPano generates a panorama from six different text descriptions, while in V2P it generates from one view condition and five text descriptions.

4.4. Additional Experiment

To further assess the generative generalization capability of JoPano, we conduct two zero-shot experiments: stylized panorama generation and multi-text generation.

Stylized Panorama Generation We evaluate stylized panorama generation by using artistic text prompts that do not appear in the training data. Thanks to the pretrained Sana-DiT backbone, JoPano can follow these prompts with style while still producing panoramas with consistent global structure. As shown in Fig. 6, our method preserves Sana’s stylized image generation ability and extends it to the panorama domain without breaking the underlying scene layout.

Multi-Text Generation We train the model using only a single text prompt per panorama; nevertheless, it generalizes to the multi-text setting. We show T2P and V2P results with multi-text prompting in Fig. 7. In the T2P setting, each of the six cubemap faces is generated from its own text prompt, and in the V2P setting, one face is given as a view

condition and the remaining five faces are generated from five text prompts. In both cases, JoPano produces a single coherent panorama.

5. Conclusion

In this paper, we present JoPano, a joint-face panorama generation framework built on a DiT-based model. We extend the pretrained Sana backbone with a Joint-Face Adapter that jointly models all six cubemap faces and transfers Sana’s image generation capability to the panorama domain. In addition, we introduce a condition-switching mechanism that unifies the two key tasks in panorama generation, text-to-panorama and view-to-panorama, within a single diffusion process. These components enable high-quality panorama generation while handling both tasks in a single, efficient model. To validate the effectiveness of our approach, we compare JoPano with existing methods on both text-to-panorama and view-to-panorama tasks. The experimental results show that our model not only unifies these two tasks within a single framework, but also achieves superior quantitative performance and better visual quality than prior methods.

References

- [1] Naofumi Akimoto, Yuhi Matsuo, and Yoshimitsu Aoki. Diverse plausible 360-degree image outpainting for efficient 3dcg background creation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 3
- [3] Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li, and Yanwen Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. In *International Conference on 3D Vision*, 2025. 2, 3
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv:2502.13923*, 2025. 5
- [5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision*, 2017. 2
- [6] Canyu Chen and Kai Shu. Promptda: Label-guided data augmentation for prompt-based few shot learners. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 562–574, 2023. 1
- [7] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics*, 41(6):1–16, 2022. 3
- [8] Mohammad Reza Karimi Dastjerdi, Yannick Hold-Geoffroy, Jonathan Eisenmann, Siavash Khodadadeh, and Jean-François Lalonde. Guided co-modulated gan for 360 field of view extrapolation. In *International Conference on 3D Vision*, 2022. 3
- [9] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *International Conference on Learning Representations*, 2025. 3
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 3
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024. 2, 3
- [12] Mengyang Feng, Jinlin Liu, Miaomiao Cui, and Xuansong Xie. Diffusion360: Seamless 360 degree panoramic image generation based on diffusion models. *arXiv:2311.13141*, 2023. 3
- [13] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics*, 36(6):1–14, 2017. 2
- [14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3
- [15] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, 1986. 2, 3
- [16] 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. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 5
- [17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2022. 3
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 2, 3
- [19] Yukun Huang, Yanning Zhou, Jianan Wang, Kaiyi Huang, and Xihui Liu. Dreamcube: Rgb-d panorama generation via multi-plane synchronization. In *International Conference on Computer Vision*, 2025. 2, 3, 5, 1
- [20] Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. Cubediff: Repurposing diffusion-based image models for panorama generation. In *International Conference on Learning Representations*, 2025. 2, 3, 5
- [21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 3
- [22] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 2, 3
- [23] Jialu Li and Mohit Bansal. Panogen: Text-conditioned panoramic environment generation for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, 2023. 2, 3
- [24] 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, 2023. 1
- [25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 2, 3
- [26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 4
- [27] Zhuqiang Lu, Kun Hu, Chaoyue Wang, Lei Bai, and Zhiyong Wang. Autoregressive omni-aware outpainting for open-vocabulary 360-degree image generation. In *AAAI Conference on Artificial Intelligence*, 2024. 3
- [28] Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative

- models with scalable interpolant transformers. In *European Conference on Computer Vision*, 2024. 3
- [29] Christopher May and Daniel Aliaga. Cubegan: omnidirectional image synthesis using generative adversarial networks. In *Computer Graphics Forum*, 2023. 3
- [30] Atsuya Nakata and Takao Yamanaka. 2s-odis: Two-stage omni-directional image synthesis by geometric distortion correction. In *European Conference on Computer Vision*, 2024. 2
- [31] Jinhong Ni, Chang-Bin Zhang, Qiang Zhang, and Jing Zhang. What makes for text to 360-degree panorama generation with stable diffusion? *arXiv:2505.22129*, 2025. 2, 3
- [32] Minh Park, Taewoong Kang, Jooyeol Yun, Sungwon Hwang, and Jaegul Choo. Spherediff: Tuning-free omnidirectional panoramic image and video generation via spherical latent representation. *arXiv:2504.14396*, 2025. 3
- [33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 2023. 2, 3
- [34] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Special Interest Group on Computer Graphics*, 2003. 2, 5
- [35] 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. *arXiv:2307.01952*, 2023. 2, 3
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 5
- [37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv:2204.06125*, 1(2):3, 2022. 3
- [38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3
- [39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, pages 36479–36494, 2022. 3
- [40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016. 5
- [41] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, 2024. 3
- [42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3
- [43] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. *arXiv:2305.11337*, 2023. 3
- [44] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 3
- [45] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4
- [46] Xiancheng Sun, Mai Xu, Shengxi Li, Senmao Ma, Xin Deng, Lai Jiang, and Gang Shen. Spherical manifold guided diffusion model for panoramic image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 3, 5
- [47] Jing Tan, Shuai Yang, Tong Wu, Jingwen He, Yuwei Guo, Ziwei Liu, and Dahua Lin. Imagine360: Immersive 360 video generation from perspective anchor. *arXiv:2412.03552*, 2024. 3
- [48] Shitao Tang, Fuayng Zhang, Jiacheng Chen, Peng Wang, and Furukawa Yasutaka. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In *Advances in Neural Information Processing Systems*, 2023. 2, 3
- [49] HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv:2507.21809*, 2025. 2, 3
- [50] Chaoyang Wang, Xiangtai Li, Lu Qi, Xiaofan Lin, Jinbin Bai, Qianyu Zhou, and Yunhai Tong. Conditional panoramic image generation via masked autoregressive modeling. *arXiv:2505.16862*, 2025. 3, 5
- [51] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. Stylelight: Hdr panorama generation for lighting estimation and editing. In *European Conference on Computer Vision*, 2022. 3
- [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*, 2024. 3
- [53] Jionghao Wang, Ziyu Chen, Jun Ling, Rong Xie, and Li Song. 360-degree panorama generation from few unregistered nfov images. In *ACM International Conference on Multimedia*, 2023. 2, 5
- [54] Ning-Hsu Albert Wang and Yu-Lun Liu. Depth anywhere: Enhancing 360 monocular depth estimation via perspective distillation and unlabeled data augmentation. In *Advances in Neural Information Processing Systems*, pages 127739–127764, 2024. 1
- [55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5
- [56] Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. Panodiffusion: 360-degree panorama outpainting via diffusion.

- In International Conference on Learning Representations, 2024. [2](#), [5](#)
- [57] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In IEEE Conference on Computer Vision and Pattern Recognition, 2012. [2](#), [5](#)
 - [58] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: efficient high-resolution text-to-image synthesis with linear diffusion transformers. In International Conference on Learning Representations, 2025. [2](#), [3](#)
 - [59] 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 IEEE Conference on Virtual Reality and 3D User Interfaces, 2024. [2](#)
 - [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. In Advances in Neural Information Processing Systems, 2024. [2](#), [3](#)
 - [61] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In IEEE Conference on Computer Vision and Pattern Recognition, 2024. [2](#)
 - [62] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In IEEE Conference on Computer Vision and Pattern Recognition, 2025. [2](#)
 - [63] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2024. [2](#), [3](#), [5](#)
 - [64] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In IEEE Conference on Computer Vision and Pattern Recognition, 2023. [3](#)
 - [65] Dian Zheng, Cheng Zhang, Xiao-Ming Wu, Cao Li, Chengfei Lv, Jian-Fang Hu, and Wei-Shi Zheng. Panorama generation from nfov image done right. In IEEE Conference on Computer Vision and Pattern Recognition, 2025. [2](#), [3](#), [5](#)
 - [66] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In European Conference on Computer Vision, 2020. [5](#)
 - [67] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suyu You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In European Conference on Computer Vision, 2024. [2](#), [3](#)
 - [68] Yikang Zhou, Tao Zhang, Dizhe Zhang, Shunping Ji, Xiangtai Li, and Lu Qi. Dense360: Dense understanding from omnidirectional panoramas. arXiv:2506.14471, 2025. [3](#)

JoPano: Unified Panorama Generation via Joint Modeling

Supplementary Material

A. Details of Cross-Face Blending

A.1. Implementation Details

In this section, we provide implementation details of Cross-Face Blending (CFB), whose Poisson formulation is given in Eq. (10). For each cubemap face g_i , we discretize the domain Ω_i on the image grid and solve for a discrete unknown $f_i(u, v)$ defined on pixels (u, v) . We first compute the discrete Laplacian of g_i using the standard 5-point stencil:

$$\Delta g_i(u, v) = g_i(u + 1, v) + g_i(u - 1, v) + g_i(u, v + 1) + g_i(u, v - 1) - 4g_i(u, v). \quad (13)$$

For interior pixels, the Poisson equation is discretized as

$$f_i(u + 1, v) + f_i(u - 1, v) + f_i(u, v + 1) + f_i(u, v - 1) - 4f_i(u, v) = \Delta g_i(u, v). \quad (14)$$

The Dirichlet boundary values on $\partial\Omega_i$ are fixed to the pixelwise averages of g_i and its neighboring faces along each shared edge. To solve this linear system, we apply the iterative Gauss–Seidel method. Starting from the initial guess $f_i^{(0)} = g_i$, we update interior pixels sequentially. At iteration $k + 1$, the value at pixel (u, v) is updated using the most recent values of its neighbors:

$$f_i^{(k+1)}(u, v) = \frac{1}{4} \left(f_i^{(k)}(u + 1, v) + f_i^{(k+1)}(u - 1, v) + f_i^{(k)}(u, v + 1) + f_i^{(k+1)}(u, v - 1) - \Delta g_i(u, v) \right), \quad (15)$$

where the left and top neighbors have already been updated at iteration $k + 1$, while the right and bottom neighbors use values from iteration k . In practice, we run 200 iterations in all our experiments to obtain the blended solution f_i for each face.

A.2. Visual Effect of Cross-Face Blending

We further illustrate the visual effect of Cross-Face Blending on ERP panoramas. Fig. 8 compares results with and without CFB, showing that it produces smoother cross-face transitions and reduces seam artifacts.

B. Experimental Details of Compared Methods

In this section, we describe how we reproduce DreamCube [19] for comparison. We follow the data processing pipeline described in the original paper on our datasets. For datasets that are not provided in cubemap format, we

first convert ERP panoramas into cubemaps using standard perspective projection. We then use BLIP-2 [24] to generate an image caption for each cube face. Next, to annotate the depth of these panoramas, we build a high resolution panorama depth estimation pipeline by connecting the panorama depth estimation work Depth Anywhere [54] and the image-guided depth upsampling work PromptDA [6], which supports panorama depth estimation. We use this pipeline to perform depth estimation on ERP panoramas and then project the obtained depth panoramas into cubemaps. After these preprocessing steps, we run DreamCube with a single view image, its corresponding depth map, and six separate text prompts.

C. Additional High Resolution Experiment

We further validate the high resolution generation capability of JoPano by retraining the model at a resolution of $1024 \times 1024 \times 6$, and visualizing the outputs in ERP format at 4096×2048 . From the comparisons in Tab. 4, we evaluate FID, CLIP-FID, IS, and CLIP-Score. The results prove that our method maintains strong performance even in this high resolution setting.

D. More Panorama Results

We present additional results in Figs. 9 to 12 to further illustrate the performance of JoPano on panorama generation. These examples show that JoPano consistently produces sharp, low-distortion panoramas with improved seam consistency, faithful text alignment, and high-quality stylized results in both T2P and V2P settings.

E. Limitation and Future Work

Although JoPano achieves promising results, it still exhibits several limitations, mainly for the following two reasons. First, since the original training images are 1024×512 , which are simply resized to 2048×1024 or 4096×2048 during training, the generated panoramas exhibit noticeable blurriness in fine details. Second, although Sana is an efficient DiT model with reduced memory consumption and fast inference, it still lags behind Flux in terms of visual quality. In future work, we plan to enhance panorama generation quality from both the data and model perspectives: on the one hand, by constructing a large-scale, high-resolution dataset, and on the other hand, by using Flux as the base model to train the Joint-Face Adapter, thereby achieving higher-quality panorama generation.

Table 4. Quantitative evaluation at 1024×6 and 512×6 resolutions measured by FID, CLIP-FID (CF), IS, and CLIP-Score (CS).

Methods	SUN360				Structure3D			
	FID ↓	CF ↓	IS ↑	CS ↑	FID ↓	CF ↓	IS ↑	CS ↑
JoPano-T2P (1024×6)	28.58	12.37	7.97	30.04	32.70	15.71	3.25	28.52
JoPano-T2P (512×6)	29.83	10.95	7.80	30.12	34.44	16.17	3.51	27.96
JoPano-V2P (1024×6)	12.93	4.14	7.27	27.84	16.28	2.93	3.08	27.55
JoPano-V2P (512×6)	13.07	4.06	7.05	27.93	16.75	3.97	3.04	27.33

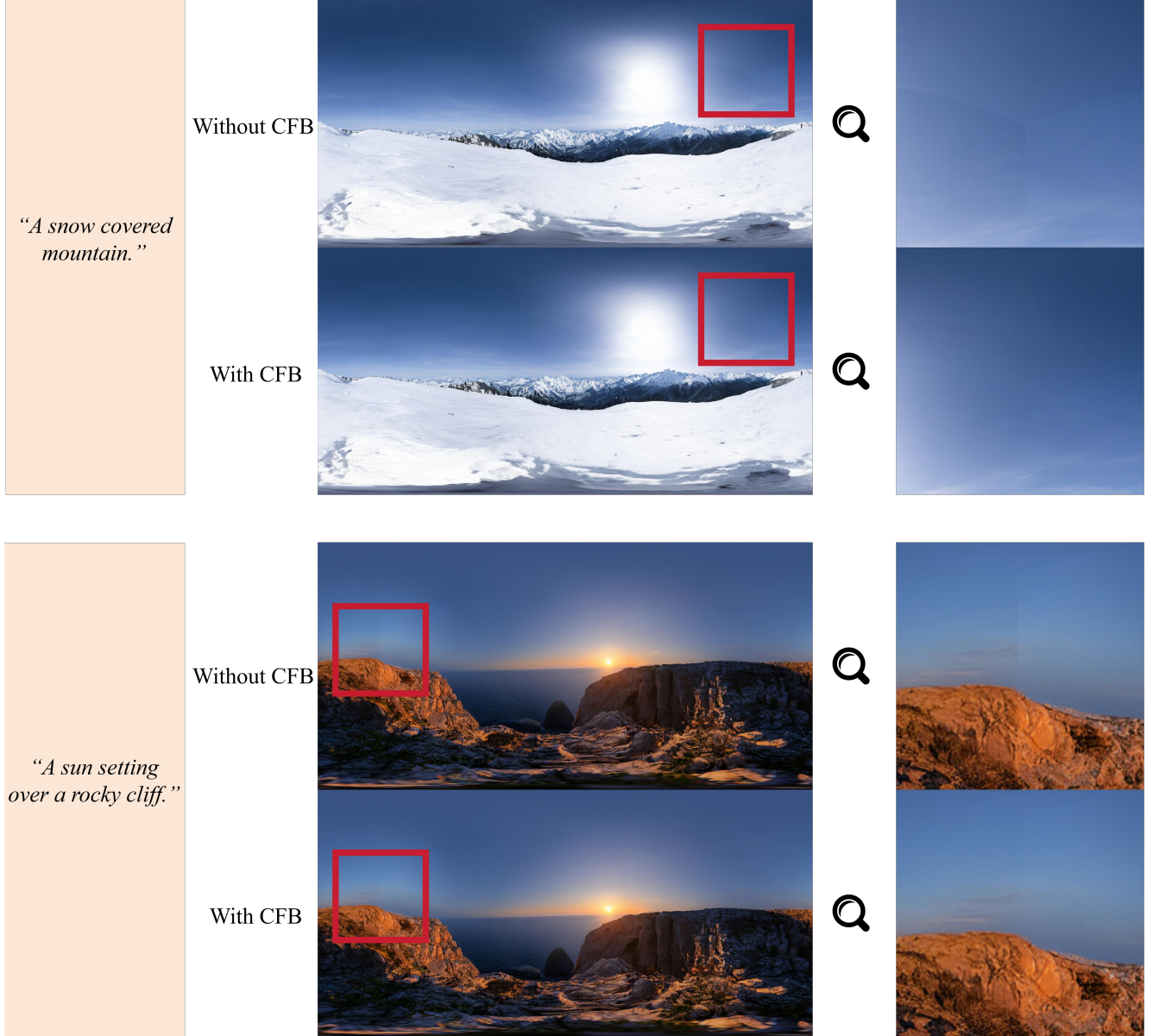


Figure 8. More results of ERP panoramas with and without Cross-Face Blending (CFB).



“An abandoned building with graffiti.”



“A room with a bed and a mirror.”



“A garden with flowers and trees.”



“A church with trees in the background.”



“A park with trees and leaves.”



“A beach with grass and clouds.”



“A snowy city square at night.”



“A canal in Venice.”

Figure 9. More results of T2P generation.



*"A mountain temple in clouds, in **sumi-e ink style**."*



*"A rainforested hot spring, in **palette-knife impasto painting style**."*



*"A snowy railway platform with steam, in **sepia film photograph style**."*



*"A thunderstorm over mountains, in **dramatic oil painting**."*



*"An interior scene painted in the **style of Vincent van Gogh**, **thick oil paint texture**."*



*"A streetcar interior on a rainy evening, in **Edward Hopper realism style**."*



*"A rainy neon alley, in **cyberpunk photography style**."*



*"A small Italian town square, in **watercolor travel sketch style**."*

Figure 10. More results of stylized T2P generation.



Figure 11. More results of V2P generation.

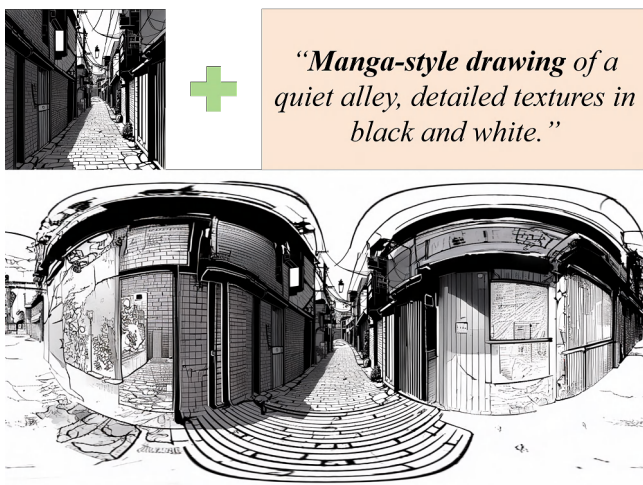
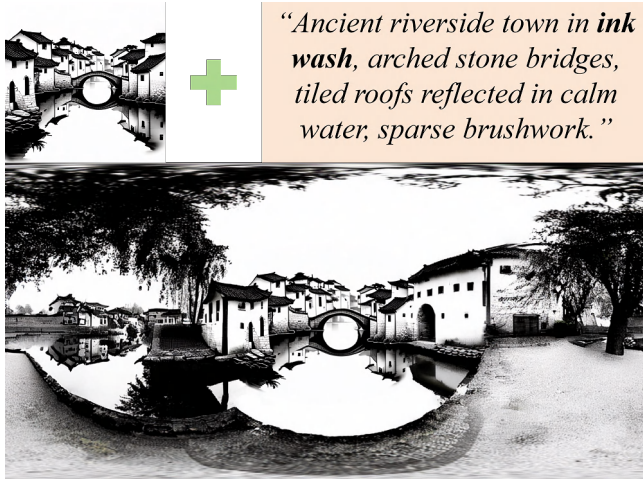


Figure 12. More results of stylized V2P generation.