Linear Fusion MultiDiffusion for Fast Training-Free Spherical Panorama Generation

Akio Hayakawa[1], Yusuke Mukuta[1,2], and Tatsuya Harada[1,2]

[1] The University of Tokyo
[2] RIKEN Center for Advanced Intelligence Project
{hayakawa,mukuta,harada}@mi.t.u-tokyo.ac.jp

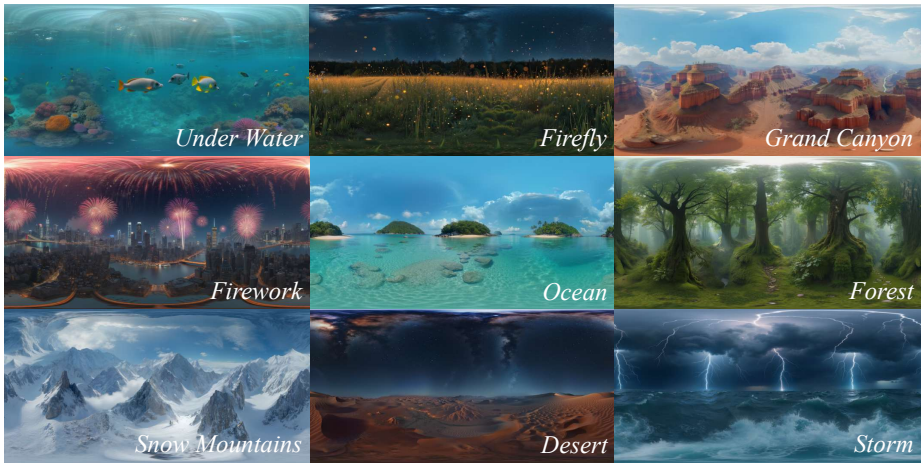


Fig. 1: *LF-MultiDiffusion* enables high-quality panorama generation using a pre-trained text-to-image generator in a training-free manner, achieving SOTA with faster runtime.

Abstract. We propose *LF-MultiDiffusion*, a training-free panorama generation method that extends MultiDiffusion to support linear projections between target and reference image spaces. Our key idea is to reformulate latent aggregation as a regularized least-squares problem and solve it efficiently with a Krylov-based iterative solver inside the denoising loop. This formulation enables denser and more natural mappings than prior training-free methods, yielding more stable generation with far fewer perspective views. As a result, *LF-MultiDiffusion* reduces the number of image generator evaluations during denoising and significantly improves inference efficiency. Experiments show that *LF-MultiDiffusion* achieves better visual quality, text alignment, and panoramic consistency than the strongest training-free baseline, while providing a 15.36× speedup. Our project page is available at: `https://ahykw.github.io/lfmd`.

Keywords: Training-free panorama generation · MultiDiffusion · Latent optimization · Efficient inference

1 Introduction

Panoramic imagery has become a fundamental component of immersive digital experiences, including content for head-mounted displays (HMDs), real estate virtual tours, and street-level mapping. It represents a 360° scene in a single compact image, enabling users to look around seamlessly. As commercial HMDs have become widely available, the demand for panoramic content is growing not only for capturing and sharing real-world scenes but also for creating them.

Recent advances in generative modeling enable high-quality image synthesis [3, 6, 29], but most models are designed for a limited field of view. Several works [16, 25, 31, 37, 40] have extended image generation to panoramic image synthesis using text-ERP paired datasets, but their applicability remains limited to specific domains due to the scarcity of such datasets. Moreover, there is a substantial gap in domain coverage between general image generation models and panorama-specific models. For example, image generation models can be trained on diverse visual domains such as artistic or cartoon-style images, whereas such content is rarely available in panoramic datasets. These limitations make training-free approaches, which fully leverage the capability of pre-trained image generation models, particularly attractive for panoramic content creation.

A few prior works have explored this direction. DynamicScaler [23] proposes Panoramic Projecting Denoise, which generates panoramic images by optimizing panorama-domain noisy latents so that perspective patches obtained via equirectangular projection (ERP) follow the reverse diffusion trajectories of a pre-trained image generator. SphereDiff [27] instead parameterizes latents directly on the sphere surface to reduce distortion near the pole. While these methods achieve seamless and high-quality panorama generation, their optimization relies on MultiDiffusion [2], which restricts the projection between the panorama and perspective spaces to direct pixel sampling (or bijective mapping) for each camera view. As a result, they require a large number of overlapping views (44 in DynamicScaler and 89 in SphereDiff) to cover all pixels in the panorama and maintain seamlessness. Because a pre-trained image generator must be evaluated for every view at every denoising step, this incurs a high computational cost; generating a single panorama takes about 7 minutes with DynamicScaler and 32 minutes with SphereDiff using FLUX [3].

To address this limitation, we propose *Linear Fusion MultiDiffusion* (*LF-MultiDiffusion*), an extension of MultiDiffusion that supports arbitrary linear projections between the target and reference image spaces (Fig. 2). Our formulation casts latent aggregation as a regularized least-squares problem. Its solution is no longer a simple weighted average of reprojected denoised latents, and explicit inversion is computationally impractical. We therefore incorporate a Krylov-based iterative solver into the denoising loop. We empirically show that this iterative update is efficient, adding only 20% of the cost of the denoising step. By supporting arbitrary linear projections, our method enables denser and more natural mappings between panoramic and perspective views, which stabilizes generation even with far fewer perspective views. Experimental results show that *LF-MultiDiffusion* achieves better visual quality, text alignment, and geometric

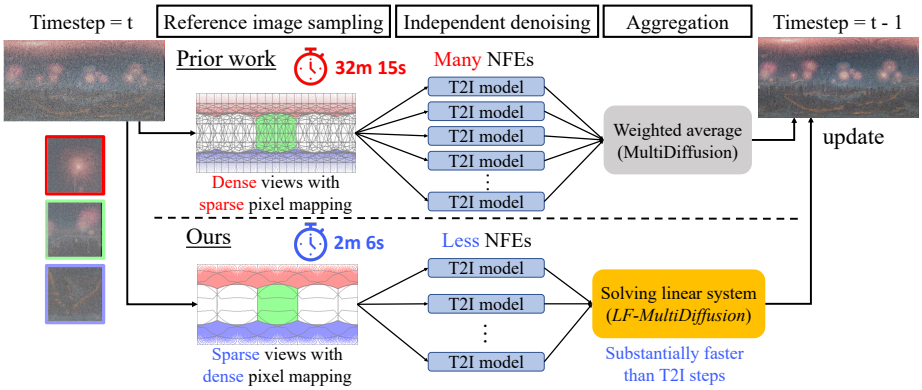


Fig. 2: *LF-MultiDiffusion* for fast training-free panorama generation. Prior methods are constrained to perform direct pixel sampling within the MultiDiffusion framework [2], leading to many neural function evaluations (NFEs) and slow inference. Our method supports denser linear mappings, improving both generation quality and efficiency. Camera frustums and inference times for prior work are taken from SphereDiff [27].

consistency than existing training-free methods while achieving a 15.36× speedup over the best-performing baseline.

2 Related work

2.1 Diffusion and flow-matching models for image generation

Diffusion models [15] and flow-matching models [22,24] have become the dominant backbone for text-to-image generation [3, 7, 26, 29, 30]. However, these models are typically trained for a fixed aspect ratio or a narrow range of resolutions, and generation outside that regime often degrades layout and composition unless additional design choices are introduced.

To support flexible output sizes and aspect ratios, prior work has either adapted pre-trained text-to-image diffusion models at inference time [14] or modified the architecture and training framework to natively support variable resolutions and aspect ratios [34]. MultiDiffusion [2] and its variants [9, 21, 33, 41] extend pre-trained text-to-image diffusion models to larger canvases through region-wise denoising and fusion, enabling arbitrary aspect ratios and higher resolutions in a model-agnostic, zero-shot manner. However, these methods are primarily designed for planar imagery or wide-canvas images rather than geometrically consistent 360° panoramas and do not explicitly enforce loop consistency over 360° scenes.

2.2 Training-based text-conditional 360° panorama generation

Text2Light [4] pioneered text-driven panorama generation using a discrete VQ-VAE latent space and an autoregressive prior. Following the success of diffusion

models in text-to-image (T2I) generation, diffusion-based methods have also become the dominant paradigm for panorama generation [16, 25, 31, 37, 40]. These methods rely on text-panorama paired datasets, which are challenging to collect at scale and, therefore, tend to have limited domain coverage and reduced robustness to diverse text prompts. For example, a model trained primarily on indoor panorama datasets typically generalizes poorly to outdoor scenes. CurvedDiffusion [32] finetunes a pre-trained T2I latent diffusion model using synthetically distorted images and demonstrates that panorama generation can emerge from such adaptation. However, the output aspect ratio is inherited from the base model, and it supports only a fixed resolution.

In contrast, our method directly uses pre-trained T2I models with their parameters fixed, thereby inheriting the base models' prompt coverage. Moreover, it supports arbitrary aspect ratios and resolutions by adopting region-wise generation as in MultiDiffusion-style methods.

2.3 Training-free text-conditional 360° panorama generation

A few prior works have explored training-free panorama generation. DynamicScaler [23] optimizes panorama-domain noisy latents over diffusion timesteps by applying MultiDiffusion with equirectangular projection (ERP). SphereDiff [27] similarly adopts a training-free formulation, but represents latents directly on the sphere to reduce the distortion near the poles. Because the original MultiDiffusion formulation assumes one-to-one pixel correspondence between the large canvas and each local patch, these methods must use direct pixel sampling between panoramas to perspective views, such as nearest-neighbor sampling in DynamicScaler and the dynamic latent sampling specialized for the spherical latent in SphereDiff. To cover the full panorama and maintain seamlessness, they require many overlapping camera views. As a result, they suffer from inefficient inference due to many neural function evaluations.

An orthogonal line of work studies training-free text-to-3D scene generation. WonderJouney [39], LucidDreamer [5], and WonderWorld [38] iteratively expand 3D scenes by optimizing 3D point clouds, 3D Gaussian splatting (3DGS) [18], and 3D Gaussian surfels, respectively, from inpainted views generated by a pre-trained text-to-image generator. Because these methods progressively construct the scene through sequential view expansion and optimization, they often suffer from loop inconsistency and error accumulation. To mitigate these issues, DreamScene360 [43] first generates several 360° panoramas followed by VLM-based initial scene selection, and then lifts them into 3D representations. Our work focuses on 360° panorama generation and is complementary to these training-free text-to-3D scene generation methods, as exemplified in DreamScene360.

3 Method

In this section, we first briefly review MultiDiffusion and discuss the challenges of applying it to 360° panoramic view synthesis. We then introduce

LF-MultiDiffusion, an extension of MultiDiffusion that supports arbitrary linear projections between the target and reference image spaces. Since most diffusion models are latent diffusion models [29], we use "image" and "pixel" to also refer to a latent tensor and its spatial components, respectively.

3.1 Preliminaries

Overview of MultiDiffusion. MultiDiffusion [2] enables the generation of images with arbitrary shapes using a pre-trained diffusion model without additional training or finetuning. Let $\Phi : \mathcal{I} \times \mathcal{Y} \to \mathcal{I}$ denote a pre-trained diffusion model, where $\mathcal{I} \subseteq \mathbb{R}^M$ is the reference image space, and $\mathcal{Y}$ is the corresponding condition space. Starting from $I_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with condition $y \in \mathcal{Y}$, the reverse diffusion process is written as

$$I_T, I_{T-1}, \ldots, I_0 \quad \text{s.t.} \quad I_{t-1} = \Phi(I_t \mid y), \tag{1}$$

which gradually transforms a noisy image I_T into a clean image I_0. MultiDiffusion defines a reverse diffusion process in a potentially different target image space $\mathcal{J} \subseteq \mathbb{R}^{M'}$ with condition space $\mathcal{Z}$. Starting from $J_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with condition $z \in \mathcal{Z}$, the reverse MultiDiffusion process is written as

$$J_T, J_{T-1}, \ldots, J_0 \quad \text{s.t.} \quad J_{t-1} = \Psi(J_t \mid z), \tag{2}$$

where $\Psi : \mathcal{J} \times \mathcal{Z} \to \mathcal{J}$ denotes a denoising operator at diffusion step t, referred to as the MultiDiffuser. In panoramic generation, the target image typically has a larger spatial extent than the reference image, *i.e.*, $M' \geq M$.

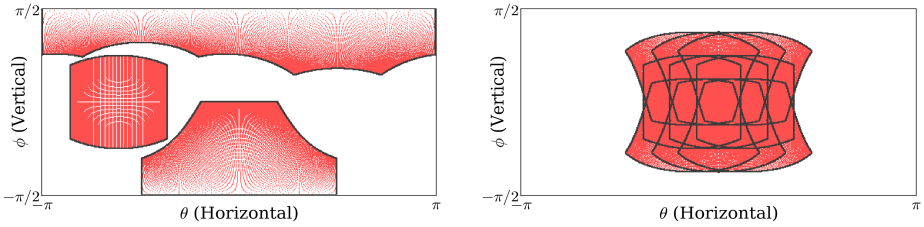
The key idea of MultiDiffusion is to construct Ψ so that it is as consistent as possible with the pre-trained diffusion model Φ. Let $F_{t,i} : \mathcal{J} \to \mathcal{I}$ denote a mapping from the target image space to the reference image space, where $F_{t,i}(J)$ extracts a reference image from J by direct pixel sampling, such as cropping or nearest neighbor reprojection. Let $y_{t,i} = \lambda_{t,i}(z)$ be the corresponding condition for the i-th reference image, where $\lambda_{t,i} : \mathcal{Z} \to \mathcal{Y}$ maps the target condition to the reference condition (*e.g.*, LLM-based conversion from global scene condition to that of each local patch). Then, the MultiDiffuser is defined as

$$\Psi(J_t \mid z) = \underset{J_{t-1} \in \mathcal{J}}{\arg\min} \sum_{i=1}^{N} \left\| W_{t,i} \odot \left[F_{t,i}(J_{t-1}) - \Phi(F_{t,i}(J_t) \mid y_{t,i}) \right] \right\|^2, \tag{3}$$

where $W_{t,i} \in \mathbb{R}^M_{\geq 0}$ is a per-pixel weight and $\odot$ is the Hadamard product. When $F_{t,i}$ is a direct pixel-sampling operator, Eq. (3) admits the closed-form solution

$$\Psi(J_t \mid z) = \sum_{i=1}^{N} \frac{F_{t,i}^{-1}(W_{t,i})}{\sum_{j=1}^{N} F_{t,j}^{-1}(W_{t,j})} \odot F_{t,i}^{-1}(\Phi(F_{t,i}(J_t) \mid y_{t,i})). \tag{4}$$

In practice, $W_{t,i}$ is often set to $\mathbf{1}_M \in \mathbb{R}^M$ so that all reference images contribute equally. We follow this setting and omit the weight term in the following sections for brevity.



(a) ERP + direct pixel sampling from perspective views to panorama.

(b) Covered pixels by nine camera positions near the center used in SphereDiff [27].

Fig. 3: Perspective-to-panorama mapping with direct pixel sampling. Panorama size: 256×512; perspective view: 128×128 with FOV = 90°. (a) Pixels covered by a single view projected onto the ERP panorama at different camera latitudes $\phi = [77.5°, 0°, -45°]$ (top to bottom). ERP-based direct sampling leaves many pixels uncovered (white pixels within the view frustum). (b) Existing methods mitigate this issue by densely sampling overlapping cameras, but they must run the diffusion model for every view, increasing computational cost.

Challenges of applying MultiDiffusion to 360° panoramas. Applying Eq. (4) to non-perspective 360° panoramas is challenging because the direct-sampling assumption makes it difficult to cover all panorama pixels. To use Eq. (4), every target pixel must be assigned to at least one pixel in some perspective view. However, when perspective views are projected onto an ERP panorama and then reprojected by direct pixel sampling through $F_{t,i}^{-1}$, many panorama pixels remain uncovered, as illustrated in Fig. 3a. Existing approaches [23,27] address this issue by densely sampling camera views with substantial overlap (Fig. 3b). Since the pre-trained diffusion model Φ must be evaluated for every perspective view at every diffusion step, this dense sampling leads to high computational cost (*e.g.*, SphereDiff requires more than 30 minutes per panorama, as shown in Tab. 1).

A natural way to reduce the number of uncovered pixels is to use denser mappings, such as bilinear interpolation or pixel splatting from panoramas to perspective views. As shown in Figure 4a, bilinear mapping between a perspective image and the center of a panorama produces no uncovered pixels. One may therefore consider extending Eq. (4) by still averaging reprojected predictions pixel-wise under such dense mappings. However, this naive extension violates consistency with the pre-trained model and severely degrades the intermediate denoising latents. To demonstrate this, we generate only the center region of a panorama using MultiDiffusion with bilinear mapping between the perspective image and the panorama. As shown in Fig. 4b, the naive extension produces noticeably degraded results.

3.2 MultiDiffuser with arbitrary linear projections

To overcome this limitation, we propose *Linear Fusion MultiDiffusion* (*LF-MultiDiffusion*), an extension of MultiDiffusion to the case where the target-to-reference mapping is an arbitrary linear operator. Let $\mathcal{S} \subset \mathbb{R}^{M \times M'}$ denote

(a) Bilinear mapping **(b)** Naive extension of Eq. (4) **(c)** Ours

Fig. 4: Panorama generation with bilinear mapping between the panorama and a perspective view, where the panoramic latent is optimized at each denoising step. We use FLUX [3] as the pre-trained diffusion model with the text prompt *"A room"*. (a) A perspective image is bilinearly mapped to the center of the panorama, leaving no pixels uncovered. (b) Naively extending Eq. (4) by pixel-wise averaging produces a blurry result. (c) Our method generates a high-quality image under the same bilinear mapping.

a family of linear mappings from a target panorama vector $J \in \mathcal{J} \subseteq \mathbb{R}^{M'}$ to a reference image vector in $\mathbb{R}^M$. We replace the direct-sampling operators $\{F_{t,i}\}_{i=1}^N$ with linear projection matrices $\{S_{t,i} \in \mathcal{S}\}_{i=1}^N$. Then, the resulting linear-fusion MultiDiffuser Ψ_{LF} is defined as

$$\Psi_{\mathrm{LF}}(J_t \mid z) = \underset{J_{t-1} \in \mathcal{J}}{\arg\min} \sum_{i=1}^{N} \left\| S_{t,i} J_{t-1} - \Phi(S_{t,i} J_t \mid y_{t,i}) \right\|^2 . \tag{5}$$

Here, $S_{t,i} J \in \mathbb{R}^M$ denotes the i-th reference image obtained from J by the corresponding linear projection.

Regularized formulation. Solving Eq. (5) in closed form would require explicitly forming and inverting the associated normal-equation matrix, which is computationally prohibitive for high-resolution panoramas. Instead, we introduce a quadratic regularizer and estimate J_{t-1} by solving the following regularized least-squares problem:

$$J_{t-1} = \underset{J \in \mathcal{J}}{\arg\min} \sum_{i=1}^{N} \left\| S_{t,i} J - \Phi(S_{t,i} J_t \mid y_{t,i}) \right\|^2 + \lambda \|LJ\|^2, \tag{6}$$

where $\lambda \geq 0$ is a regularization coefficient, and $L = W^{1/2} D \in \mathbb{R}^{P \times M'}$ is a weighted first-order regularizer with a discrete difference operator $D \in \mathbb{R}^{P \times M'}$ and a diagonal weight matrix $W = \mathrm{diag}(\mathbf{w}) \in \mathbb{R}^{P \times P}$ for $\mathbf{w} \in \mathbb{R}^P_{\geq 0}$. This formulation includes, for example, ridge- or Laplacian-type regularization depending on the choice of D and W.

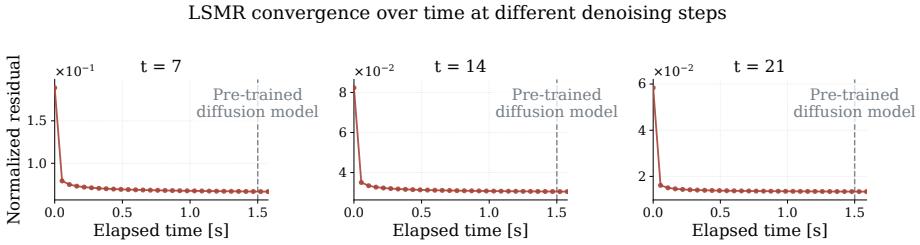


Fig. 5: Runtime comparison between the LSMR [8] update and the total denoising cost of a full MultiDiffusion step. We plot the normalized residual in Eq. (5) every 10 solver updates at denoising timestep $t \in \{7, 14, 21\}$. The total denoising cost for $N = 14$ using Flux [3] is shown as a gray dashed line. The Krylov solver update converges far more quickly than the denoising cost of the corresponding MultiDiffusion step.

Augmented system. We rewrite Eq. (6) in an equivalent augmented form. Define the stacked operator and target vector as

$$\tilde{S} := \begin{bmatrix} S_{t,1} \\ \vdots \\ S_{t,N} \\ \sqrt{\lambda}\, L \end{bmatrix} \in \mathbb{R}^{(NM+P)\times M'}, \qquad \tilde{d} := \begin{bmatrix} \Phi(S_{t,1} J_t \mid y_{t,1}) \\ \vdots \\ \Phi(S_{t,N} J_t \mid y_{t,N}) \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{NM+P}. \tag{7}$$

Then, Eq. (6) is equivalently written as

$$J_{t-1} = \underset{J \in \mathcal{J}}{\arg\min} \ \|\tilde{S} J - \tilde{d}\|^2. \tag{8}$$

We solve Eq. (8) using a matrix-free Krylov subspace method, either by applying preconditioned conjugate gradients (PCG) [11] to the corresponding normal equations or by directly applying LSMR [8]. In all cases, the solver requires only the operations $J \mapsto S_{t,i} J$, $I \mapsto S_{t,i}^\top I$, and $J \mapsto L^\top (LJ)$, together with their stacked counterparts for $\tilde{S}$. Therefore, the projection matrices need not be formed explicitly in memory during the iterative update.

Figure 5 compares the runtime of the LSMR update with the total denoising cost of one MultiDiffusion step at different denoising steps. For this experiment, we used $N = 14$, which is also the default setting in our main experiments. We plot the normalized residual $\sum_i \|S_{t,i} J - \Phi(S_{t,i} J_t \mid y_{t,i})\|^2 \big/ \sum_i \|\Phi(S_{t,i} J_t \mid y_{t,i})\|^2$ every 10 solver iterations. The residual decreases rapidly within a few tens of iterations, and the solver update requires less than 20% of the computational cost of a full MultiDiffusion step. This overhead is substantially smaller than the additional cost incurred by increasing the number of views N, which scales linearly with the number of diffusion model evaluations. For reference, DynamicScaler and SphereDiff use $N = 44$ and $N = 89$, respectively, corresponding to roughly 3-7$\times$ longer denoising runtime. These results show that the proposed iterative update adapts the MultiDiffusion to panoramic view synthesis in a computationally efficient manner.

3.3 Implementation of the target space and projection

Definition of the reference image space. We define the reference image space $\mathcal{I}$ as a rectangle of size $H \times W$. The mapping operations $J \mapsto S_{t,i}J$ and $I \mapsto S_{t,i}^{\top}I$ are implemented using pixel correspondences in 3D space. To establish these correspondences, we assume that each pixel in the reference image corresponds to a point on the unit sphere. Specifically, for a pixel $\mathbf{u} \in [0, W) \times [0, H)$ in $I_{t,i} \in \mathcal{I}$, we define the corresponding 3D point $\mathbf{x} \in \mathbb{S}^2$ by back-projecting its viewing ray onto the unit sphere, where $\mathbf{u}$ and $\mathbf{x}$ satisfy

$$\mathbf{u} \sim M_{t,i}[\mathbf{x}^{\top}, 1]^{\top}, \quad M_{t,i} = K[R_{t,i} \mid \mathbf{c}], \tag{9}$$

where $M_{t,i}$ denotes the camera projection matrix, $K \in \mathbb{R}^{3\times3}$ is the intrinsic matrix, $R_{t,i} \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $\mathbf{c} \in \mathbb{R}^3$ is the translation vector.

The intrinsic matrix K and the translation vector $\mathbf{c}$ are shared across all views and timesteps. Unless otherwise noted, we set $\mathrm{FOV}_x = \mathrm{FOV}_y = 90^\circ$ when computing K, and use $\mathbf{c} = [0, 0, 0]^{\top}$ since all camera centers coincide with the origin. In contrast, the rotation matrix $R_{t,i}$ depends on both the timestep and the view index. Following the Panoramic Projecting Denoising proposed in DynamicScaler [23], we sweep the viewing direction horizontally as the reverse diffusion process proceeds, allowing the model to efficiently extract different subregions of the target image space over time. Specifically, let $\{(\theta_i, \phi_i)\}_{i=1}^{N}$ denote the initial longitude-latitude pairs with $\theta_i \in (-\pi, \pi]$ and $\phi_i \in (-\pi/2, \pi/2]$. At timestep t, we define the view angle as $(\theta_{t,i}, \phi_{t,i}) = (\theta_i + st, \phi_i)$, where $s \in (-\pi, \pi)$ is a constant angular shift per timestep. We then construct the rotation matrix $R_{t,i}(\theta_{t,i}, \phi_{t,i})$ such that its forward axis is aligned with the viewing direction $d(\theta_{t,i}, \phi_{t,i})$ defined by $d(\theta, \phi) = (\cos\phi \sin\theta, \sin\phi, \cos\phi\cos\theta)$.

ERP latent space with bilinear projection. Inspired by DynamicScaler [23], we represent the target space $\mathcal{J}$ as an ERP latent. Specifically, we define $\mathcal{J}$ as a rectangle of size $H_p \times W_p$, where each pixel $(i, j) \in [0, W_p) \times [0, H_p)$ corresponds to a point on the sphere with longitude θ and latitude ϕ as

$$\theta = 2\pi \cdot \frac{i}{W_p} - \pi, \quad \phi = \pi \cdot \frac{j}{H_p} - \frac{\pi}{2}. \tag{10}$$

We define the mapping $J \mapsto S_{t,i}J$ from the panorama to a perspective view by bilinear interpolation. Specifically, to compute the pixel value $I[u, v]$ at pixel position $\mathbf{u} = (u, v)$, we first back-project its viewing ray to a 3D point $(x_{\mathbf{u}}, y_{\mathbf{u}}, z_{\mathbf{u}})$ using Eq. (9), and then compute the corresponding longitude and latitude $(\theta_{\mathbf{u}}, \phi_{\mathbf{u}}) \in \mathbb{R}^2$ as $\theta_{\mathbf{u}} = \mathrm{atan2}(x_{\mathbf{u}}, z_{\mathbf{u}})$ and $\phi_{\mathbf{u}} = \arcsin(y_{\mathbf{u}})$. The corresponding continuous panorama coordinate $(i_{\mathbf{u}}, j_{\mathbf{u}})$ is then obtained by inverting Eq. (10). Using the 1D linear interpolation kernel $\kappa(r) = \max(0, 1 - |r|)$, we compute $I[u, v]$ as

$$I[u, v] = \sum_{(i,j)\in\mathcal{N}(i_{\mathbf{u}}, j_{\mathbf{u}})} \kappa(i - i_{\mathbf{u}})\kappa(j - j_{\mathbf{u}})J[i, j], \tag{11}$$

Algorithm 1 Panorama generation with *LF-MultiDiffusion*

Require: Pre-trained diffusion model Φ, view conditions $\{y_{t,i}\}_{i=1}^{N}$, linear mappings $\{S_{t,i}\}_{i=1}^{N}$, number of views N, total number of diffusion steps T, and *LF-MultiDiffusion* stopping timestep T_M

$J_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Phase 1: Panorama latent optimization (*LF-MultiDiffusion*)

for $t = T, T-1, \ldots, T_M+1$ **do**

  $I_{t,i} \leftarrow S_{t,i} J_t \quad \forall i \in [N]$ ▷ Render

  $\tilde{I}_{t-1,i} \leftarrow \Phi(I_{t,i}, t, y_{t,i}) \quad \forall i \in [N]$ ▷ Denoise

  $J_{t-1} \leftarrow \mathrm{KrylovSolve}\Big(\{S_{t,i}\}_{i=1}^{N}, \{\tilde{I}_{t-1,i}\}_{i=1}^{N}\Big)$ ▷ Solve Eq. (8)

end for

Phase 2: View-wise post refinement

$I_{T_M,i} \leftarrow S_{T_M,i} J_{T_M} \quad \forall i \in [N]$ ▷ Render

for $t = T_M, T_M-1, \ldots, 1$ **do**

  $I_{t-1,i} \leftarrow \Phi(I_{t,i}, t, y_{t,i}) \quad \forall i \in [N]$ ▷ Denoise

end for

$J_0 \leftarrow \mathrm{DistortionAwareWeightedAveraging}\big(\{S_{T_M,i}\}_{i=1}^{N}, \{I_{0,i}\}_{i=1}^{N}\big)$ ▷ Aggregate

return J_0

where $\mathcal{N}(i_{\mathbf{u}}, j_{\mathbf{u}}) = \{(\lfloor i_{\mathbf{u}} \rfloor + \delta_i,\ \lfloor j_{\mathbf{u}} \rfloor + \delta_j) \mid \delta_i, \delta_j \in \{0,1\}\}$ denotes the four neighboring panorama pixels of $(i_{\mathbf{u}}, j_{\mathbf{u}})$. The horizontal index is treated as periodic, whereas the vertical index is clipped to the valid range. For the reverse mapping $I \mapsto S_{t,i}^{\top} I$, we use the same correspondences and interpolation weights as in Eq. (11), and scatter each pixel value $I[u, v]$ onto the target ERP latent J.

3.4 View-wise post refinement

Although *LF-MultiDiffusion* produces globally consistent panorama latents, the final decoded ERP images can still appear slightly blurry compared with images generated directly by the base model in its native perspective domain. We conjecture that this gap arises because the panorama is optimized in the VAE [19] latent space and decoded into the ERP format, where both latent-space compression and the geometric distortion inherent to ERP representations can weaken local details.

In practice, we stop the panorama optimization at an intermediate timestep $T_M > 0$ and use the resulting latent J_{T_M} for a view-wise post-refinement stage. Specifically, we first render a set of perspective-view latents $\{I_i\}_{i=1}^{N_{\mathrm{ref}}}$ by applying the corresponding projections $S'_{T_M,i}$ to J_{T_M}. We then continue denoising each rendered view independently using the same pre-trained image generator Φ until the final step. Intuitively, this step brings each local view back to the native generation domain, allowing high-frequency details to be recovered while preserving the global structure established by *LF-MultiDiffusion*.

After denoising, each refined latent is decoded independently into an RGB image. We then project the refined perspective images back to the ERP image and aggregate them using Distortion-Aware Weighted Averaging from SphereDiff [27],

which accounts for the non-uniform distortion of ERP coordinates. Algorithm 1 summarizes an overview of the proposed generation process.

4 Experiments

4.1 Experimental setup

Implementation and inference details. We used FLUX [3] as the pre-trained image generator with the classifier-free guidance [12] scale set to 3.5. For text prompts, we followed SphereDiff [27] and used their 20 prompt sets. For each prompt set, we generated 10 panorama images with different random seeds. The total number of denoising steps T is set to 28, and the stopping step for *LF-MultiDiffusion* T_M is set to 23.

To construct mappings between the panorama and perspective views, we used $N = 14$ initial view directions: two polar views $(\theta, \phi) = (0, \pm\frac{\pi}{2})$, eight tilted views $(\theta, \phi) = (\frac{\pi}{2}i, \pm\frac{\pi}{3})$ for $i \in \{0, 1, 2, 3\}$, and four equatorial views $(\theta, \phi) = (\frac{\pi}{2}i, 0)$ for $i \in \{0, 1, 2, 3\}$. We set the per-step angular shift $s = \frac{\pi}{18}$. We generated 2048×4096 ERP panoramas with 512×512 perspective views. All experiments were conducted on a single NVIDIA A100 GPU (40GB).

Baselines. We compared against DynamicScaler [23] and SphereDiff [27] as training-free baselines. We set $N = 44$ for DynamicScaler and $N = 89$ for SphereDiff, which are the default values. For a fair comparison, we used FLUX [3] as the base text-to-image generator. We additionally reported results for representative training-based models, Text2Light [4] and PanFusion [40].

Evaluation process and metrics. We followed the evaluation protocol of SphereDiff [27]. Specifically, we rendered 14 perspective views from each generated panorama using FOV = 90° for evaluation. Inspired by VBench [13], we used MUSIQ [17] to assess imaging quality (e.g., exposure, noise, and blur), the LAION aesthetic predictor [20] to measure aesthetic preference, and CLIP [28] feature similarity between each view and the corresponding text prompt to evaluate text alignment. We also reported Q-Align [35], an LLM-based visual evaluator, as an additional measure of perceptual quality. Furthermore, we conducted the VLM-based evaluation [42] similar to SphereDiff [27] to assess the panoramic distortion and continuity using Qwen-VL [1]. We use the text prompt proposed in SphereDiff [27] with slight modifications tailored for Qwen-VL (see Section C in the supplementary material for the full evaluation prompt). For runtime, we measured the end-to-end generation time excluding I/O overhead.

Additional analyses in the supplementary material. Beyond the main experimental results presented in this section, we provide additional analyses in the supplementary material. They include experiments with SANA [36], an extension to text-to-panoramic video generation with LTX-Video [10], latent and projection variants, a sensitivity study on the post-refinement timestep, and additional generated samples.

Table 1: Comparison with baselines. 2048 × 4096 panoramas are generated except for PanFusion (which only supports 512 × 1024). For the training-free methods, Flux [3] is used as a base T2I model. Runtime denotes the mean wall-clock time measured on a single Nvidia A100 40GB GPU (lower is better). Higher is better for all other metrics. *LF-MultiDiffusion* supports a dense mapping between panoramic and perspective views, thereby achieving higher generation quality with faster inference. *: The runtime for PanFusion is shown for reference due to its small resolution.

Method	Runtime	Aesthetic	Imaging	QAlign	CLIP	Distortion	Continuity
Training-based panorama generation							
Text2Light	56s	0.424	0.434	2.22	19.60	3.26	3.56
PanFusion	27s*	0.473	0.527	2.62	25.44	1.93	2.31
Training-free panorama generation							
DynamicScaler	7m 31s	0.463	0.516	3.11	26.32	3.15	3.83
SphereDiff	32m 15s	0.597	0.533	3.23	27.65	**4.64**	4.82
Ours	**2m 6s**	**0.616**	**0.568**	**3.34**	**29.19**	**4.64**	**4.86**

4.2 Comparison with baselines

Quantitative comparison. Table 1 compares our method with baselines. Training-based methods often struggle with out-of-domain prompts due to the limited coverage of the training data. For instance, for the prompt *"Aurora"*, we observed that Text2Light tended to produce a collapsed black image, while PanFusion often generated an unrelated indoor scene. Among training-free approaches, *LF-MultiDiffusion* outperforms DynamicScaler and SphereDiff across all metrics. DynamicScaler applies an additional offset-shifting denoising stage that treats the panorama as a wide-canvas perspective image, rather than enforcing a strictly spherical (ERP-consistent) representation throughout. This can introduce geometric inconsistencies and degrade quality when evaluated via perspective-view rendering. In contrast, *LF-MultiDiffusion* and SphereDiff explicitly model curved panoramic geometry and mitigate these distortions, resulting in consistently higher scores across evaluation metrics. We attribute the improvement of *LF-MultiDiffusion* over SphereDiff to the use of denser linear projections, which stabilize optimization on curved representations and enable coherent panorama generation with substantially fewer views. Figure 6 subjectively compares the generated panorama and perspective views among training-free approaches.

Notably, *LF-MultiDiffusion* achieves 3.58× and 15.36× speedup over DynamicScaler and SphereDiff, respectively. The dominant cost in training-free panorama generation is the number of evaluations of the pre-trained image generator, which scales linearly with the number of perspective views. By removing the direct-sampling constraint and supporting dense projections between panoramas and perspective views, *LF-MultiDiffusion* achieves better quality with significantly lower computational cost.

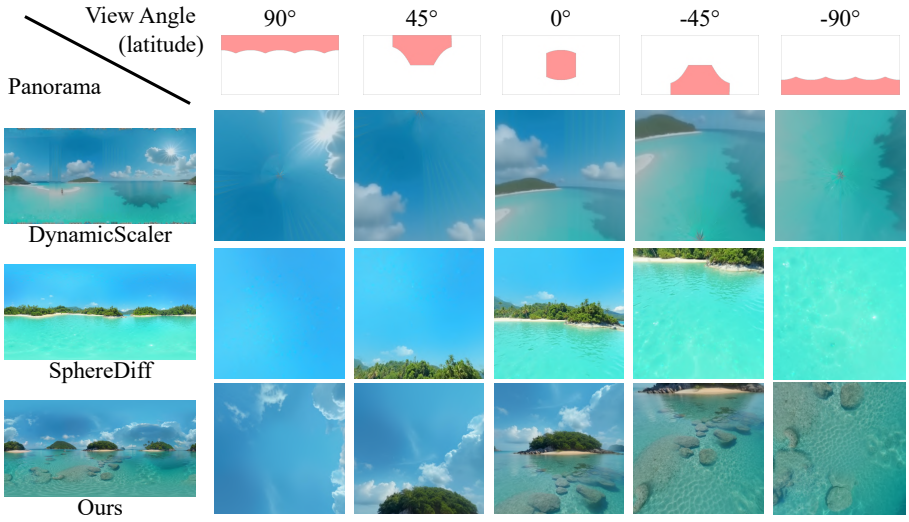


Fig. 6: Subjective comparison. DynamicScaler often exhibits pole distortions (±90°). SphereDiff and *LF-MultiDiffusion* generate natural views across directions, and *LF-MultiDiffusion* achieves higher overall fidelity.

Table 2: User study results on subjective panorama quality. Participants rated each method on a 5-point Likert scale. Higher scores indicate better image quality, fewer perceived distortions, and better continuity. We report the mean rating with 95% confidence intervals.

Method	Image Quality ↑	Distortion ↑	Continuity ↑
DynamicScaler	2.39±0.17	2.36±0.16	2.25±0.20
SphereDiff	3.64±0.14	3.73±0.12	3.96±0.12
LF-MultiDiffusion (Ours)	**4.07**±0.10	**3.75**±0.16	**4.07**±0.11

User study. We further conducted a user study to evaluate the subjective quality of the generated panoramas. Participants rated samples from DynamicScaler, SphereDiff, and *LF-MultiDiffusion* in terms of image quality, geometric naturalness, and continuity on a 5-point Likert scale. As shown in Table 2, *LF-MultiDiffusion* achieved the highest scores across all three aspects, indicating that our formulation improves perceived visual quality while preserving geometric naturalness and seamlessness. See the supplementary material for the setup details.

4.3 Ablation studies

Solver types and number of updates. Table 3 reports the effect of the solver choice and the number of solver iterations. We evaluated PCG [11] and LSMR [8] with iteration in $\{10, 30, 100\}$. For PCG, we used a diagonal preconditioner derived

Table 3: Results when using different solvers and the number of solver iterations.

Solver	Iters	QAlign↑	CLIP↑
PCG	10	3.00	28.94
	30	3.04	28.85
	100	2.97	28.95
LSMR	10	3.27	29.07
	30	**3.34**	**29.19**
	100	3.30	28.93

Table 4: Results when using different regularizers and coefficients.

Regularizer	λ	QAlign↑	CLIP↑
-	0	3.282	29.02
Ridge	10^{-3}	3.330	29.10
	10^{-4}	3.340	29.03
	10^{-5}	3.336	28.99
Laplacian	10^{-3}	3.297	28.89
	10^{-4}	**3.341**	**29.19**
	10^{-5}	3.321	29.00

Table 5: Results when using a different number of perspective views N.

N	Runtime↓	Aesthetic↑	Imaging↑	QAlign↑	CLIP↑	Distortion↑	Continuity↑
6	1m 15s	0.518	**0.634**	**3.549**	27.55	3.98	4.53
14	2m 6s	0.616	0.568	3.341	29.19	**4.64**	**4.86**
20	2m 42s	**0.621**	0.552	3.223	**29.21**	4.45	4.77
26	3m 18s	0.603	0.538	3.165	29.12	4.33	4.69

from the squared interpolation weights computed from $S_{i,t}^{\top} S_{i,t}$. Overall, LSMR achieves better scores across all metrics. Increasing the number of iterations from 10 to 30 improved performance, whereas further increasing it to 100 yielded diminishing returns.

Regularizer and its coefficient. Table 4 reports an ablation on regularizer types and the coefficient λ. Using LSMR, we compared two regularizers: ridge and discrete Laplacian. For each regularizer, we tested $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. Across these settings, both regularizers performed comparably. We therefore used the discrete Laplacian with $\lambda = 10^{-4}$ as the default, as it provided consistently higher *QAlign* and *CLIP* scores.

Number of reference perspective views. Table 5 reports the effect of the number of reference perspective views, $N \in \{6, 14, 20, 26\}$. For $N = 6$, we used two polar views $(\theta, \phi) = (0, \pm\frac{\pi}{2})$ and four equatorial views $(\theta, \phi) = (\frac{\pi}{2}i, 0)$ for $i \in \{0, 1, 2, 3\}$. For larger N, we used two polar views, $2L$ tilted views $(\theta, \phi) = (\frac{2\pi}{L}i, \pm\frac{\pi}{3})$, and L equatorial views $(\theta, \phi) = (\frac{2\pi}{L}i, 0)$ with $L \in \{4, 6, 8\}$ (corresponding to $N = 14, 20, 26$).

Using $N = 6$ yields the highest scores for *Imaging* and *QAlign*, but substantially degrades geometric metrics such as *Distortion* and *Continuity*. In this setting, reference views have little to no overlap at each diffusion step, so the optimization can produce high-fidelity local views while failing to enforce global seamlessness over the full panorama. Increasing the number of views to

$N = 14$ improves *Distortion* and *Continuity* while maintaining strong perceptual and text alignment scores, providing the best overall trade-off between quality, consistency, and runtime in our experiments. Further increasing N beyond 14 slightly improves some metrics but tends to degrade others, while incurring higher runtime. We conjecture that with heavier overlap, the least-squares fusion becomes more redundant and can make the iterative solve less well-conditioned, leading to diminishing returns under a fixed iteration budget.

5 Conclusion

We presented *LF-MultiDiffusion*, a fast training-free framework for panoramic image synthesis that extends MultiDiffusion to arbitrary linear projections between perspective views and ERP panoramas. By casting the update under linear mappings as a regularized least-squares problem and solving it with a matrix-free Krylov method, our approach removes the direct-sampling constraint of prior training-free baselines, enabling denser and more natural mappings between panorama and perspective views. This reduces the number of view evaluations during denoising. As a result, *LF-MultiDiffusion* improves generation quality, text alignment, and geometric consistency while achieving about $15.36\times$ faster inference than the best-performing training-free baseline.

Beyond panoramas, this formulation suggests a general recipe for optimization-based synthesis on non-planar domains; future work includes extending it to other surface topologies (e.g., torus-like representations for walk-through content) and to texture optimization directly on mesh surfaces.

Acknowledgment

This work was partially supported by JST Moonshot R&D Grant Number JPMJPS2011.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025)
2. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: fusing diffusion paths for controlled image generation. In: ICML (2023)
3. Black Forest Labs: Flux. `https://github.com/black-forest-labs/flux` (2024)
4. Chen, Z., Wang, G., Liu, Z.: Text2light: Zero-shot text-driven hdr panorama generation. ACM Trans. Graph. (2022)
5. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. IEEE TVCG (2025)

6. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
8. Fong, D.C.L., Saunders, M.: Lsmr: An iterative algorithm for sparse least-squares problems. SIAM J. Sci. Comput. (2011)
9. Frolov, S., Moser, B.B., Dengel, A.: Spotdiffusion: A fast approach for seamless panorama generation over time. In: WACV (2025)
10. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., Panet, P., Weissbuch, S., Kulikov, V., Bitterman, Y., Melumian, Z., Bibi, O.: Ltx-video: Realtime video latent diffusion
11. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. (1952)
12. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPSW on Deep Generative Models and Downstream Applications (2021)
13. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
14. Jin, Z., Shen, X., Li, B., Xue, X.: Training-free diffusion model adaptation for variable-sized text-to-image synthesis. In: NeurIPS (2023)
15. Jonathan Ho, Ajay N. Jain, P.A.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
16. Kalischek, N., Oechsle, M., Manhardt, F., Henzler, P., Schindler, K., Tombari, F.: Cubediff: Repurposing diffusion-based image models for panorama generation. In: ICLR (2025)
17. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: ICCV (2021)
18. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG (2023)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
20. LAION-AI: aesthetic-predictor. `https://github.com/LAION-AI/aesthetic-predictor` (2022)
21. Lee, Y., Kim, K., Kim, H., Sung, M.: Syncdiffusion: Coherent montage via synchronized joint diffusions. In: NeurIPS (2023)
22. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
23. Liu, J., Lin, S., Li, Y., Yang, M.H.: Dynamicscaler: Seamless and scalable video generation for panoramic scenes. In: CVPR (2025)
24. Liu, X., Gong, C., qiang liu: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023)
25. Ni, J., Zhang, C.B., Zhang, Q., Zhang, J.: What makes for text to 360-degree panorama generation with stable diffusion? In: ICCV (2025)
26. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022)

27. Park, M., Kang, T., Yun, J., Hwang, S., Choo, J.: Spherediff: Tuning-free omnidirectional panoramic image and video generation via spherical latent representation. In: AAAI (2026)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
30. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Gontijo-Lopes, R., Ayan, B.K., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022)
31. Tang, S., Zhang, F., Chen, J., Wang, P., Furukawa, Y.: MVDiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In: NeurIPS (2023)
32. Voynov, A., Hertz, A., Arar, M., Fruchter, S., Cohen-Or, D.: Curved diffusion: A generative model with optical geometry control. In: ECCV (2024)
33. Wang, H., Xiang, X., Fan, Y., Xue, J.H.: Customizing 360-degree panoramas through text-to-image diffusion models. In: WACV (2024)
34. Wang, Z., BAI, L., Yue, X., Ouyang, W., Zhang, Y.: Native-resolution image synthesis. In: NeurIPS (2025)
35. Wu, H., Zhang, Z., Zhang, W., Chen, C., Li, C., Liao, L., Wang, A., Zhang, E., Sun, W., Yan, Q., Min, X., Zhai, G., Lin, W.: Q-align: Teaching lmms for visual scoring via discrete text-defined levels. In: ICML (2024)
36. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., Han, S.: SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In: ICLR (2025)
37. Ye, W., Ji, C., Chen, Z., Gao, J., Huang, X., Zhang, S.H., Ouyang, W., He, T., Zhao, C., Zhang, G.: Diffpano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. In: NeurIPS (2024)
38. Yu, H.X., Duan, H., Herrmann, C., Freeman, W.T., Wu, J.: Wonderworld: Interactive 3d scene generation from a single image. In: CVPR (2025)
39. Yu, H.X., Duan, H., Hur, J., Sargent, K., Rubinstein, M., Freeman, W.T., Cole, F., Sun, D., Snavely, N., Wu, J., Herrmann, C.: Wonderjourney: Going from anywhere to everywhere. In: CVPR (2024)
40. Zhang, C., Wu, Q., Cruz Gambardella, C., Huang, X., Phung, D., Ouyang, W., Cai, J.: Taming stable diffusion for text to 360∘ panorama image generation. In: CVPR (2024)
41. Zhang, X., Zhou, T., Zhang, X., Wei, J., Tang, Y.: Multi-scale diffusion: Enhancing spatial layout in high-resolution panoramic image generation (2025)
42. Zhou, F., Gu, T., Huang, Z., Qiu, G.: Vision language modeling of content, distortion and appearance for image quality assessment. JSTSP (2024)
43. Zhou, S., Fan, Z., Xu, D., Chang, H., Chari, P., Bharadwaj, T., You, S., Wang, Z., Kadambi, A.: Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In: ECCV (2024)