

SyncDiffusion: Coherent Montage via Synchronized Joint Diffusions

Yuseung Lee Kunho Kim Hyunjin Kim Minhyuk Sung

KAIST

{phillip0701,kaist984,rlaguswls98,mhsung}@kaist.ac.kr

Abstract

The remarkable capabilities of pretrained image diffusion models have been utilized not only for generating fixed-size images but also for creating panoramas. However, naive stitching of multiple images often results in visible seams. Recent techniques have attempted to address this issue by performing joint diffusions in multiple windows and averaging latent features in overlapping regions. However, these approaches, which focus on seamless montage generation, often yield incoherent outputs by blending different scenes within a single image. To overcome this limitation, we propose SYNCDIFFUSION, a plug-and-play module that synchronizes multiple diffusions through gradient descent from a perceptual similarity loss. Specifically, we compute the gradient of the perceptual loss using the predicted denoised images at each denoising step, providing meaningful guidance for achieving coherent montages. Our experimental results demonstrate that our method produces significantly more coherent outputs compared to previous methods (66.35% vs. 33.65% in our user study) while still maintaining fidelity (as assessed by GIQA) and compatibility with the input prompt (as measured by CLIP score). Our project page is at <https://syncdiffusion.github.io>.



Figure 1: Comparison of panoramas generated with prompt “a photo of a rock concert” by Blended Latent Diffusion [1] (top), MultiDiffusion [3] (middle), and our SYNCDIFFUSION (bottom). Blended Latent Diffusion, when applied on image extrapolation, often generates visible seams and repetitive patterns. MultiDiffusion creates seamless panoramas but fails to achieve global coherence across the image. In contrast, our SYNCDIFFUSION synchronizes windows across the panorama by increasing the perceptual similarity of the denoised output predictions. This results in significantly more coherent panorama outputs.

1 Introduction

Diffusion models have recently emerged as the forefront of generative models. Recent breakthroughs in text-to-image generation, such as DALL·E 2 [36], Midjourney [31], and StableDiffusion [39], were diffusion models trained with billions of images. Compared with GANs, diffusion models offer advantages not only in producing high-quality, realistic images but also in being utilized for conditional generation setups such as inpainting [29, 34, 41], editing [30, 16, 7, 1, 2, 34], and deblurring [47, 9], with few-shot [24, 12, 40] or even zero-shot [37, 36, 27, 46, 48] adaptation. The iterative reverse diffusion process can easily incorporate diverse conditions and regularizations at each step while guiding the entire process to produce realistic images. Hence, the diffusion model, once trained on a large-scale dataset, becomes a versatile and multi-purpose pretrained model that can be applied to various tasks and applications.

Recent work has extended the versatility of pretrained diffusion models to generate arbitrary-sized images or panoramas using either sequential [2] or joint [3] diffusion processes. Since typical image diffusion models are trained to generate fixed-sized images, creating panoramic images requires stitching multiple fixed-size images together, which can be impossible to do seamlessly without considering integration in the generation process. Two approaches have been proposed to tackle this issue. The first approach [2] involves generating the final output as a sequence of inpainting starting from an initial image, where each consecutive view image is produced while fixing the overlapped region (which is referred to as image extrapolation in their work [2]). However, this approach often struggles to seamlessly extend the given image and also tends to repeat similar patterns, resulting in unrealistic panoramas as shown in the first row of Fig. 1. The other approach is joint diffusion [3], which operates the reverse generative process simultaneously across multiple views while averaging the intermediate noisy images (or the noisy latent feature images) in the overlapped regions at each reverse process step. The blending of noisy latent feature images among the views at each denoising step can effectively generate a seamless montage of images. However, it is important to note that the content and styles of the images may vary across the views, resulting in a mixture of nighttime and daytime images in one panorama, as shown in the second row of Fig. 1. The lack of consistency occurs because the latent features of the overlapped regions are simply averaged without considering the coherence between them.

To address the limitation of previous work that produces unrealistic or incoherent montages, we present a novel synchronization module for joint diffusion, dubbed **SYNCDIFFUSION**. This module guides the reverse diffusion processes to achieve global coherence across different areas of the panorama image. Similar to previous guided diffusion methods [36, 29, 34, 41], our **SYNCDIFFUSION** guides the reverse diffusion process while adjusting the intermediate noisy images at each step. Our guidance is specifically provided as a gradient descent from a perceptual similarity loss calculated across multiple windows. Various off-the-shelf perceptual similarity losses such as LPIPS [52] or Style loss [22] can be utilized in our framework. However, perceptual similarity losses computed with noisy images cannot effectively guide the denoising process. Thus, we draw inspiration from the non-Markovian formulation of DDIM [45] leveraging the prediction of the *denoised output* from the current noisy image at each denoising step. We compute the perceptual similarity loss using the *foreseen* denoised images at each step and then backpropagate the gradient through the noisy images. By leveraging the synergy with a prior seamless stitching technique [3] based on averaging latent features at each denoising step, our joint diffusion framework demonstrates the capability to generate montages that exhibit both local seamlessness and global coherence, as shown in the last row of Fig. 1. This is achieved in a zero-shot manner, without the need for retraining or fine-tuning of existing diffusion models.

In our experiments generating panoramas using Stable Diffusion 2.0 [39] model, the results demonstrate that our method achieves significantly higher coherence compared to previous methods. Quantitatively, as we increase the weight of the gradient descent, we observe improved coherence, measured by LPIPS [52] and Style loss [22], while maintaining fidelity (measured by GIQA [15]) and compatibility with the input prompt (assessed by the CLIP score [17]). Diversity, measured by FID and KID, exhibits a trade-off with coherence, but our method still achieves much better scores compared to the baseline method. Our user study further confirms a significant preference for our method (66.35%) over the previous method (33.65%).

2 Related Work

2.1 Diffusion Models

Diffusion probabilistic models [44, 11, 19, 33] are a group of generative models that generate data by sampling from an initial Gaussian distribution and iteratively applying a denoising process (referred to as the reverse process). These methods have achieved remarkable strides in image generation [39, 11, 42, 36], surpassing previous image generation models including GANs [23, 6]. DDPM [19] was among the pioneering models that showcased the impressive capability of image generation through Markovian forward and reverse processes, albeit with longer computation times in generation. This limitation was addressed by DDIM [45], which significantly reduced the sampling time in the reverse process using a non-Markovian transition formulation conditioned on the predicted denoised data. Furthermore, LDM [39] proposed incorporating the diffusion process into the latent space, achieving state-of-the-art realism in generated images and gaining attention in the text-to-image generation. Diffusion models have also demonstrated their applicability to diverse data modalities such as audio [49, 21, 28, 14], videos [20, 5], and 3D objects [35, 27, 46, 48].

2.2 Few-Shot or Zero-Shot Adaptation of Diffusion Models

Building upon the remarkable generation capabilities of pretrained public text-to-image diffusion models such as StableDiffusion [39], recent research has introduced various methodologies for leveraging the pretrained models in diverse tasks, including conditional generation, image editing, and manipulation, without the need to retrain the models from scratch. ControlNet [50] is an example of a method that enables the incorporation of additional conditions into existing text-to-image diffusion models through few-shot finetuning, wherein the image encoder is duplicated to handle the additional conditional image, and only a carefully selected subset of parameters is modified during the finetuning process. Custom Diffusion [24] also introduces a similar idea of enabling few-shot tuning while keeping the majority of parameters in the neural network frozen, but with applications of finetuning the model for a particular class or concept of images. Other previous work has also demonstrated that diffusion models can even be applied to novel tasks in a zero-shot manner. SDEdit [30] was the first to show zero-shot conditional image generation using a pretrained diffusion model by dispersing noise over the conditional image and denoising it back to a real image. Repaint [29] introduced an image inpainting idea by combining a generated foreground image and a noised background image at each time step. Similar *guided* diffusion ideas have also been explored for various tasks, such as image super-resolution [26, 13, 9, 43], colorization [41, 10], deblurring [47, 9], and style transfer [25, 24, 12, 40]. We propose a novel guided diffusion framework for image montage generation via joint diffusion.

2.3 Montage Generation via Diffusion Models

Panorama generation is one of the zero-shot applications of diffusion models. Since diffusion models are trained to generate images of a specific size and on a 2D plane, stitching is required to generate panoramas or textures. Most previous methods [38, 8, 2, 1] have employed inpainting-based approaches for seamless stitching. These methods extrapolate the accumulated image and fill only the missing regions to generate the panorama or texture. In contrast, MultiDiffusion [3] and DiffCollage [51], our concurrent works, conduct diffusion in multiple views jointly while combining noisy latent features or scores at each reverse diffusion step. While both approaches have successfully produced continuous images, they have limitations in enforcing global coherence across the panorama or texture. In this work, we propose a simple yet effective synchronization module for joint diffusion to achieve semantic coherence in panorama image generation.

3 Backgrounds

3.1 Diffusion Models

In this section, we provide a brief overview of the Denoising Diffusion Probabilistic Models (DDPM) [19] and Denoising Diffusion Implicit Models (DDIM) [45], which are the foundations of recent pretrained image diffusion models. The aim of DDPM is to approximate the data distribution $q(\mathbf{x}_0)$ with a tractable model distribution $p_\theta(\mathbf{x}_0)$, which takes the form of a *Markov* chain with learned Gaussian transitions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad \text{where } p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \quad (1)$$

The parameters of the joint distribution (known as the *reverse process*) θ are learned by minimizing the negative evidence lower bound (ELBO):

$$\min_{\theta} \mathbb{E}_{q(\mathbf{x}_0)} [-\log p_{\theta}(\mathbf{x}_0)] \leq \min_{\theta} \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T)} [-\log p_{\theta}(\mathbf{x}_{0:T}) + \log q(\mathbf{x}_{1:T}|\mathbf{x}_0)], \quad (2)$$

where $q_{\theta}(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is the *forward process* adding a sequence of Gaussian noise to the data while increasing the noise scale. Among the variations of the forward processes, DDPM uses the *variance-preserving* diffusion that parameterizes the Gaussian transitions as follows with a decreasing sequence $\alpha_{1:T} \in (0, 1]^T$:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \text{ where } q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}\left(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\mathbf{x}_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)\mathbf{I}\right). \quad (3)$$

The definition of the Gaussian transitions in the forward process derives the following property:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I}), \quad (4)$$

and thus matches the choice of the starting distribution in the reverse process (a unit Gaussian) since $q(\mathbf{x}_T|\mathbf{x}_0)$ converges to a unit Gaussian when α_T is set close to 0. It also allows expressing \mathbf{x}_t with \mathbf{x}_0 and a unit Gaussian noise variable ϵ :

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (5)$$

In DDPM [19], the Gaussian transition $p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ for each \mathbf{x}_t in the reverse process is modeled as follows¹:

$$p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}\left(\sqrt{\frac{\alpha_{t-1}}{\alpha_t}}\left(\mathbf{x}_t - \frac{1}{\sqrt{1 - \alpha_t}}\left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)\epsilon_{\theta}(\mathbf{x}_t, t)\right), \sigma_t^2\mathbf{I}\right), \quad (6)$$

where $\sigma_t^2 = \frac{1 - \alpha_{t-1}}{1 - \alpha_t} \cdot \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)$, and $\epsilon_{\theta}(\mathbf{x}_t, t)$ is a learned function that optimizes the objective in Eq. 2 when it maps each \mathbf{x}_t at time t to a unit Gaussian noise, thus resulting in the following simplified loss:

$$L(\epsilon_{\theta}) := \sum_{t=1}^T \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\epsilon_{\theta}(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon_t, t) - \epsilon_t\|_2^2]. \quad (7)$$

DDIM [45] provides a different perspective of seeing the same forward process as a *non-Markovian* process while taking the input data \mathbf{x}_0 into consideration in *reversed* transitions:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^T q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \quad \text{where } q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_T}\mathbf{x}_0, (1 - \alpha_T)\mathbf{I}) \quad (8)$$

and $\forall s < t, \quad q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\alpha_s}\mathbf{x}_0 + \sqrt{1 - \alpha_s - \sigma_s^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \sigma_s^2\mathbf{I}\right).$

Then, each transition in the reverse process is also redefined as first predicting the *denoised observation* \mathbf{x}_0 given each \mathbf{x}_t and then sampling \mathbf{x}_{t-1} via the conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$:

$$p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t) := \begin{cases} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \phi_{\theta}(\mathbf{x}_t, t)) & \text{if } t \geq 2 \\ \mathcal{N}(\phi_{\theta}(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}) & \text{if } t = 1, \end{cases} \quad (9)$$

where

$$\phi_{\theta}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon_{\theta}(\mathbf{x}_t, t)) \quad (10)$$

is the predicted denoised observation. The key observations of DDIM are twofold. First, the same simplified objective (Eq. 7) can be used to find the best models $\epsilon_{\theta}(\mathbf{x}_t, t)$ in Eq. 10 that minimize the negative ELBO (Eq. 2). This means that the DDIM reverse process can be used with a pretrained

¹Note that β_t in the DDPM [19] is equivalent to $1 - \frac{\alpha_t}{\alpha_{t-1}}$ in our paper and DDIM [45].



Figure 2: Panoramas generated by MultiDiffusion [3] (left) and our SYNCDIFFUSION (right), with a prompt “a photo of a mountain range at twilight”. MultiDiffusion often combines various scenes, such as mountains with trees and snow, and even awkwardly blends them, as evident in the faded background rock. In contrast, SYNCDIFFUSION generates panoramas that are significantly more coherent.

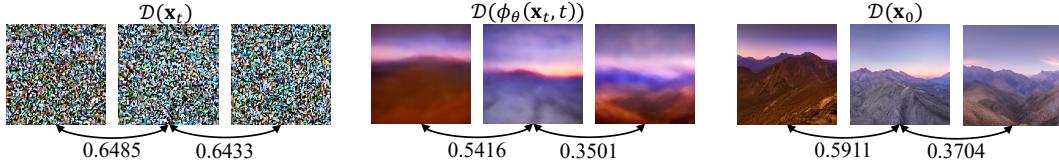


Figure 3: LPIPS [52] scores computed across the noisy images $\mathcal{D}(\mathbf{x}_t)$ at the intermediate step ($t = 45$ out of 50) of the reverse process (left), the *predicted* denoised images $\mathcal{D}(\phi(\mathbf{x}_t, t))$ at the same timestep t (middle), and the final generated images $\mathcal{D}(\mathbf{x}_0)$ at timestep $t = 0$ (right). The indistinguishable noisy images yield similar LPIPS scores among them, whereas the predicted denoised images, which closely resemble the final outputs even at the beginning of the denoising process, exhibit LPIPS scores that align with those of the final generated images. This indicates that the predicted denoised images can provide meaningful guidance for producing coherent panoramas in the diffusion process.

DDPM without retraining. Second, a subset of the time sequence $[1, \dots, T]$ can be used in the reverse process of DDIM since \mathbf{x}_s for any $s < t$ can be sampled from \mathbf{x}_t via the \mathbf{x}_0 prediction, enabling a significant boost in the reverse process computation.

In the rest of the paper, the operation sampling the next denoised data in the reverse process with the learned distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ defined in either Eq. 6 (DDPM) or Eq. 9 (DDIM) is denoted as:

$$\mathcal{S}(\mathbf{x}_t, t, \epsilon), \quad (11)$$

which takes a noisy data \mathbf{x}_t at timestep t and a unit Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as input.

3.2 Joint Diffusion

In an image diffusion model, each sample from the data distribution is either a 2D grid of per-pixel colors, or a 2D grid of latent features (as in Latent Diffusion [39]) that can be encoded from or decoded to a real image through a pretrained encoder \mathcal{E} and decoder \mathcal{D} . In the rest of the paper, the term *image* will thus be used to refer to either a *color* image or a *latent feature* image, unless explicitly stated otherwise. Image diffusion models pretrained on fixed-size images cannot be used directly to produce arbitrary-size images. MultiDiffusion [3] has addressed this limitation by using a multi-window joint diffusion approach. The framework integrates images generated from multiple windows seamlessly by *averaging* colors or features across the windows at *every* reverse diffusion step. For instance, consider the case of generating a panorama image $\mathbf{z} \in \mathbb{R}^{H_z \times W_z \times D}$. The image at each window $\mathbf{x}^{(i)} \in \mathbb{R}^{H_x \times W_x \times D}$ is a subarea of the panorama image whose union across all the windows covers the entire panorama image. Let $\mathbf{m}^{(i)} \in [0, 1]^{H_x \times H_x}$ denote a binary mask for the subregion in the panorama image corresponding to the i -th window. The function $\mathcal{T}_{\mathbf{z} \rightarrow i} : \mathbb{R}^{H_z \times W_z \times D} \rightarrow \mathbb{R}^{H_x \times W_x \times D}$ maps (crops) the panorama image \mathbf{z} to the i -th window image, while $\mathcal{T}_{i \rightarrow \mathbf{z}} : \mathbb{R}^{H_x \times W_x \times D} \rightarrow \mathbb{R}^{H_z \times W_z \times D}$ is its inverse function that fills the region outside of the mask \mathbf{m}_i with zeros. During the joint diffusion process running the reverse process simultaneously for each window, the noisy images from the windows $\mathbf{x}_t^{(i)}$ are first averaged in the panorama space:

$$\mathbf{z}_t = \frac{\sum_i \mathcal{T}_{i \rightarrow \mathbf{z}}(\mathbf{x}_t^{(i)})}{\sum_i \mathbf{m}^{(i)}}, \quad (12)$$

and then, the resulting combined noisy image \mathbf{z}_t is cropped again for each window $\mathbf{x}_t^{(i)} = \mathcal{T}_{i \rightarrow \mathbf{z}}(\mathbf{z}_t)$, modifying the noisy image at each window with a inter-window regularization.

4 SyncDiffusion

While MultiDiffusion [3] can generate seamless panorama images from joint diffusion, it often fails to produce coherent and realistic montages. The left image in Fig. 2 demonstrates that the resulting

Algorithm 1: Pseudocode of one-time denoising in SYNCDIFFUSION.

```

Parameters : $w$ ; // Gradient descent weight.
Inputs:  $\{\mathbf{x}_t^{(i)}\}_{i=0 \dots N-1}$ ; // Noisy images at timestep  $t$ .
Outputs:  $\{\mathbf{x}_{t-1}^{(i)}\}_{i=0 \dots N-1}$ ; // Noisy images at timestep  $t-1$ .

1 Function SyncDiffusion( $\{\mathbf{x}_t^{(i)}\}$ ):
2    $\hat{\mathbf{x}}_t^{(i)} \leftarrow \mathbf{x}_t^{(i)}$ ; // The anchor window at index 0 is not changed.
3   for  $i = 1, \dots, N-1$  do
4      $\hat{\mathbf{x}}_t^{(i)} \leftarrow \mathbf{x}_t^{(i)} - w \nabla_{\mathbf{x}_t^{(i)}} \mathcal{L}(\mathcal{D}(\phi_\theta(\mathbf{x}_t^{(i)}, t)), \mathcal{D}(\phi_\theta(\mathbf{x}_t^{(0)}, t)))$ ; // Gradient descent (Eq. 14)
5   return  $\{\hat{\mathbf{x}}_t^{(i)}\}$ ;

6 Function MultiDiffusion( $\{\tilde{\mathbf{x}}_t^{(i)}\}$ ):
7    $\mathbf{z}_t \leftarrow \frac{\sum_i \mathcal{T}_{i \rightarrow \mathbf{z}}(\tilde{\mathbf{x}}_t^{(i)})}{\sum_i \mathbf{m}^{(i)}}$ ; // Averaging in the global space (Eq. 12).
8   for  $i = 0, \dots, N-1$  do
9      $\mathbf{x}_t^{(i)} \leftarrow \mathcal{T}_{\mathbf{z} \rightarrow i}(\mathbf{z}_t)$ ;
10    return  $\{\mathbf{x}_t^{(i)}\}$ ;

11 Function DenoisingOneStep( $\{\mathbf{x}_t^{(i)}\}$ ):
12    $\{\hat{\mathbf{x}}_t^{(i)}\} \leftarrow \text{SyncDiffusion}(\{\mathbf{x}_t^{(i)}\})$ ;
13   for  $i = 0, \dots, N-1$  do
14      $\tilde{\mathbf{x}}_{t-1}^{(i)} \leftarrow \mathcal{S}(\hat{\mathbf{x}}_t^{(i)}, t, \epsilon)$ ; // Sampling the next denoised data (Eq. 11).
15    $\{\mathbf{x}_{t-1}^{(i)}\} \leftarrow \text{MultiDiffusion}(\{\tilde{\mathbf{x}}_{t-1}^{(i)}\})$ ;
16   return  $\{\mathbf{x}_{t-1}^{(i)}\}$ ;

```

image often oddly combines various scenes, such as mountains with trees and snow. Also, the blending occasionally fails to merge them in a realistic manner, as shown in the figure where distant objects are connected to closer objects. This incoherence issue in MultiDiffusion arises due to two main reasons. Firstly, the averaging operation only aligns the colors or features in the overlapped regions but does not match the *content* or *style* of the images. Secondly, it only enforces *adjacent* views to influence each other, and thus global coherence between distant windows cannot be achieved.

To address this problem, we introduce a module called SYNCDIFFUSION which enables the generation of coherent montages, as shown on the right in Fig. 4. This module can be easily integrated into an existing joint diffusion framework. Similar to MultiDiffusion, our SYNCDIFFUSION module updates the noisy image at every step of the reverse diffusion process. In contrast to averaging the colors or latent features in the overlapped regions, however, SYNCDIFFUSION employs the backpropagation of gradients from a perceptual similarity loss computed across the windows to perform the update. The perceptual similarity loss, denoted as \mathcal{L} , can utilize any off-the-shelf loss function for perceptual similarity, such as LPIPS [52] and style loss [22]. To facilitate efficient computation, we designate an *anchor* window with an index of 0. For each view’s noisy color image $\mathcal{D}(\mathbf{x}_t^{(i)})$ and the anchor window’s noisy color image $\mathcal{D}(\mathbf{x}_t^{(0)})$ (where the decoder \mathcal{D} can be treated as an identity function if the given diffusion model operates in image space rather than latent space), one can measure the coherence using the images and conduct gradient descent through $\mathbf{x}_t^{(i)}$:

$$\hat{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)} - w \nabla_{\mathbf{x}_t^{(i)}} \mathcal{L}(\mathcal{D}(\mathbf{x}_t^{(i)}), \mathcal{D}(\mathbf{x}_t^{(0)})), \quad (13)$$

where w is the weight of the gradient descent. However, the coherence measured with the *noisy* images cannot provide meaningful guidance. Fig. 3 shows examples where the left three images are the intermediate noisy images $\mathcal{D}(\mathbf{x}_t)$ at timestep $t = 45$ out of a total of 50 timesteps in the DDIM reverse process. Note that the LPIPS scores among the noisy images are indistinguishable. Hence, similar to the DDIM reverse process, we utilize the *foreseen denoised* observation of each noisy data $\phi_\theta(\mathbf{x}_t^{(i)}, t)$ in Eq. 10. We measure the coherence not with the current noisy color images $\{\mathcal{D}(\mathbf{x}_t^{(i)})\}$ but with the predicted denoised color images $\{\mathcal{D}(\phi_\theta(\mathbf{x}_t^{(i)}, t))\}$ with the timestep t and perform the backpropagation of the gradient through $\mathbf{x}_t^{(i)}$, resulting in the following updated formulation:

$$\hat{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)} - w \nabla_{\mathbf{x}_t^{(i)}} \mathcal{L}(\mathcal{D}(\phi_\theta(\mathbf{x}_t^{(i)}, t)), \mathcal{D}(\phi_\theta(\mathbf{x}_t^{(0)}, t))). \quad (14)$$

In Fig. 3, the middle three images depict the predicted denoised images $\mathcal{D}(\phi_\theta(\mathbf{x}_t, t))$ at timestep $t = 45$, which closely resemble the final generated images $\mathcal{D}(\mathbf{x}_0)$ at timestep $t = 0$, even during the initial stages of the reverse diffusion process. Therefore, the LPIPS scores among the predicted denoised images also match those of the generated images, providing meaningful guidance for maintaining coherence. During each denoising step in the joint reverse process, we apply this update to the noisy images for all windows $\{\mathbf{x}_t^{(i)}\}$, and sample the one-step denoised images. MultiDiffusion is also applied to average the sampled images at the end. Refer to Alg. 1 for detailed pseudocode.

5 Results

In our experiments, we generate panorama images using our SYNCDIFFUSION method and the pretrained Stable Diffusion 2.0 [39] model. Stable Diffusion model operates in a latent space of $\mathbb{R}^{64 \times 64 \times 4}$ and generates images of $\mathbb{R}^{512 \times 512 \times 3}$. We generate panorama images of resolution 512×3072 (64×384 in the latent space), where the width is six times the width of the output of Stable Diffusion. Each window $\mathbf{x}^{(i)}$ has an image resolution of 512×512 , with a stride of 128 pixels along the width in the image space which is equivalent to stride 16 in the latent space, resulting in a total 21 windows to operate diffusion processes jointly. We use six text prompts from MultiDiffusion [3] (see Fig. 4) and generate 500 panoramas per prompt. For the gradient descent weight w (Eq. 14), we experiment with various initial values while applying a weight decay with a rate of 0.95. We also set the center window as the anchor window with an index of 0.

Baselines We compare our SYNCDIFFUSION with previous methods that generate panoramas using a pretrained diffusion model. Blended Latent Diffusion (BLD) [1] is a diffusion-based image editing method. Following Blended Diffusion [2], we apply BLD on image extrapolation with a single window generated by Stable Diffusion as the source image. MultiDiffusion [3], served as the base of our framework, is a special case of our method when the weight of gradient descent w is 0. The same Stable Diffusion 2.0 model is used for all the methods for a fair comparison.

Evaluation Metrics We utilize a range of metrics to assess the coherence, fidelity, diversity, and compatibility of the output panoramas with the input prompt.

- **(Coherence) Intra-LPIPS** [52] and **Intra-Style-L** [22]: To assess the coherence of the generated panoramas, we introduce two metrics. Intra-LPIPS and Intra-Style-L, which are computed as the averages of LPIPS [52] and Style Loss [22], respectively, between a pair of non-overlapping window images from the same panorama. Specifically, we divide the panorama into 6 windows, each with dimensions of 512×512 and then compute the average of LPIPS and Style Loss across the 15 combinations of these cropped views. To provide a reference for the scale of these values, we generate 500 single-window-size images using the same Stable Diffusion model and compute the LPIPS and Style Loss for randomly selected 1,000 pairs of these reference images.
- **(Fidelity) Mean-GIQA** [15]: GIQA quantifies the fidelity of *individual* images by calculating the inverse of the distance between a query image and a reference set in a feature space. Mean-GIQA is computed by taking a *single* random crop of each panorama in 512×512 size and computing the average GIQA score from each cropped image to the reference set of images mentioned above.
- **(Fidelity & Diversity) FID** [18] and **KID** [4]: FID [18] and KID [4] are used to measure both fidelity and diversity. Both of them are measured with the aforementioned randomly cropped images and the set of reference images.
- **(Compatibility with the Input Prompt) Mean-CLIP-S** [17]: The compatibility with the input prompt is assessed using the mean of CLIP scores [17], denoted as Mean-CLIP-S. This metric is calculated using the same set of cropped images and the input prompt.

Qualitative Comparisons Fig. 4 showcases qualitative comparisons between our method and the baseline methods. Here, we show the results of our method generated with a weight parameter of $w = 20$. BLD [1] often exhibits visible seams due to the sequential inpainting scheme and produces repetitive patterns in the extrapolation, as illustrated by the mountains in the second case of the top row and the flowers and trees in both cases of the bottom row. MultiDiffusion [3] achieves seamless outputs, although it often produces incoherent outputs, such as mixing a sunset sky with a blue sky, as shown in the second cases of the first and second rows, and pink and purple backgrounds with blue backgrounds in the two cases of the bottom row. Our SYNCDIFFUSION generates visually and semantically more coherent panoramas with all the prompts. More qualitative comparisons are provided in the **supplementary**.

Quantitative Results Fig. 5 presents a quantitative comparison among the methods. Our method’s results are displayed using two different gradient descent weights, $w = 10$ and 20. MultiDiffusion [3]

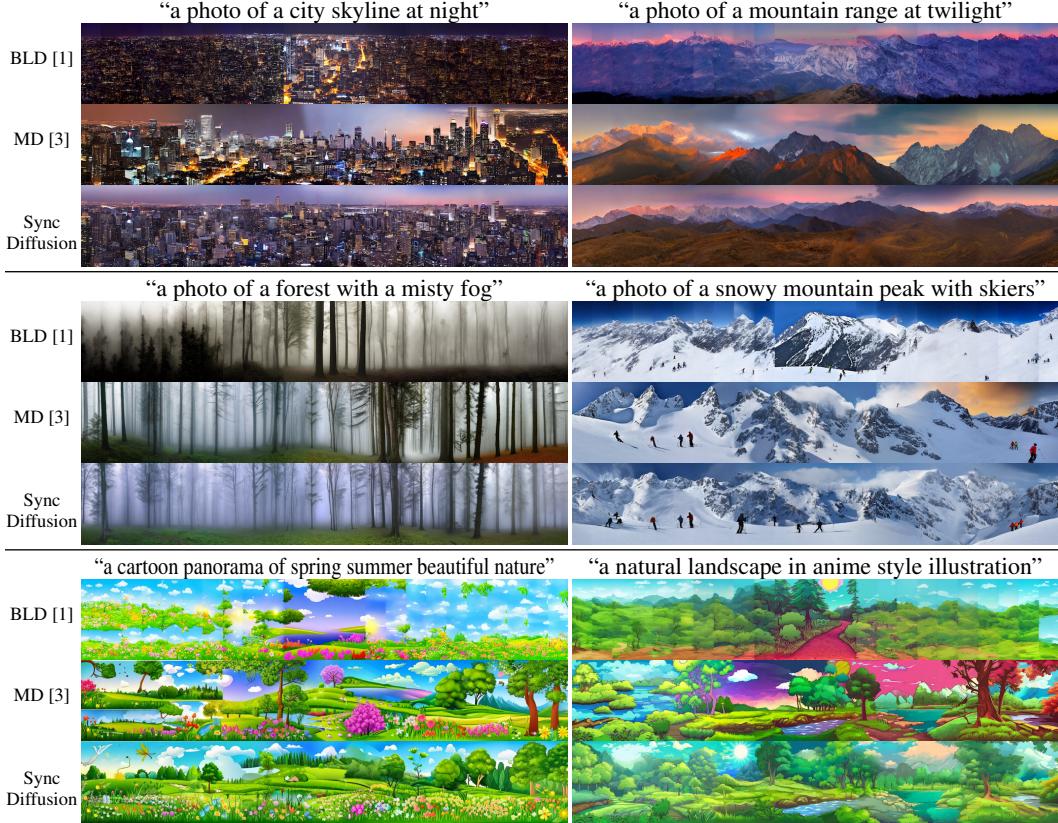


Figure 4: Qualitative comparisons. Blended Latent Diffusion [1] (BLD, the first row of each case) tends to exhibit visible seams and repetitive patterns. MultiDiffusion [3] (MD, the second row of each case) generates seamless results but lacks coherence, such as blending a sunset sky with a blue sky (top right, middle right), and displaying a combination of purple, pink, and blue backgrounds (bottom). In contrast, our SYNC DIFFUSION (the last in each case) produces seamless panoramas with significantly improved coherence. Best view in zoom and color.

is also the case when $w = 0$ in our framework. For results obtained with different weights, please refer to the **supplementary**. The color bars in the plots indicate the average scores across the six prompts, while the black lines depict the standard deviation. Note that as the gradient weight increases, both Intra-LPIPS and Intra-Style-L decrease. When $w = 20$, the Intra-LPIPS and Intra-Style-L of our method are approximately 3/4 and 1/6 of those computed with the reference set images (referred to as SD, Stable Diffusion), respectively, indicating significantly higher coherence. Moreover, the Mean-CLIP-S and Mean-GIQA scores are comparable to those computed with the reference set, meaning that the compatibility with the input prompt and fidelity are not compromised by our diffusion synchronization. The results of FID and KID demonstrate the trade-off between coherence and diversity. As the gradient descent weight w increases, FID and KID also increase slightly, although they are still much lower compared to BLD [1]. This implies that for certain images, it is more difficult to find coherent images. In the **supplementary**, we substantiate this claim with the results of shorter generated panoramas. BLD results in low Intra-LPIPS due to its tendency to repeat similar patterns, but it leads to low Mean-GIQA and very high FID and KID, indicating a significant degradation in fidelity.

User Study We conducted a user study to further evaluate the coherence of the generated panoramas. Following Ritchie [32], participants were presented with panorama images generated by both MultiDiffusion [3] and our SYNC DIFFUSION methods (with $w = 20$). They were then asked to choose one of them by answering the question: Which one appears a more coherent panorama image to you? Choose one of the following images. The order of the images was shuffled. Refer to the **supplementary** for a snapshot of the survey. Given a total of 200 questions with a random pair of panoramas, we collected 20 responses each from 86 participants on Amazon Mechanical Turk who passed our five vigilance tasks. The vigilance tasks were designed to distinguish our outputs

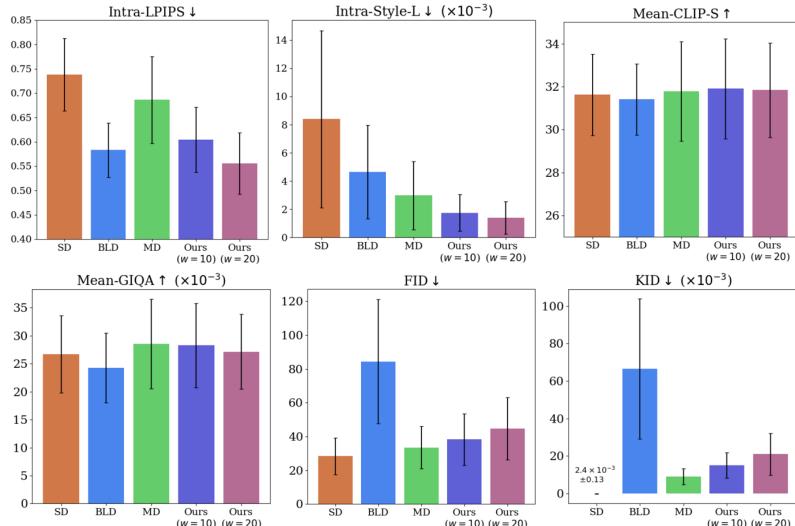


Figure 5: Quantitative results. MultiDiffusion [3] (MD) can be considered as a special case of our method when the gradient descent weight w is set to 0. As w increases, coherence (Intra-LPIPS and Intra-Style-L) improves while maintaining the compatibility with the input prompt (Mean-CLIP-S) and fidelity (Mean-GIQA). There is a trade-off between coherency and diversity, as indicated by the FID and KID results. Note that the FID and KID of our method are still significantly lower than those of Blended Latent Diffusion [1] (BLD). SD (Stable Diffusion) is the score with the reference set images. Refer to the text for the details.

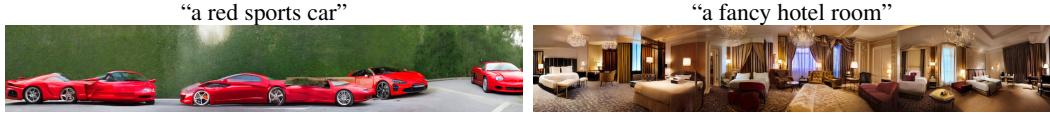


Figure 6: Our failure cases. A suitable input prompt is required to generate realistic panoramas.

from concatenations of Stable Diffusion images generated without joint diffusion. Each question received answers from at most 10 participants. Out of the 1,720 responses from 86 participants who successfully completed all the vigilance tasks, our method was preferred over MultiDiffusion on average **66.35%** of the responses, with a standard deviation of **5.19%**.

Limitations While our SYNCDIFFUSION module can significantly enhance the coherence of generated panoramas, it relies on appropriate input prompts to achieve realistic results, as illustrated in Fig. 6. Also, the SYNCDIFFUSION module that includes a forward pass through the neural network and gradient descent computation introduces additional computational overhead.

Supplementary Due to space constraints, we present the following additional results in the supplementary: 1) results with additional prompts, 2) various panorama sizes, 3) an ablation study employing Style Loss [22] as the perceptual loss, 4) using Eq. 13 instead of Eq. 14 for gradient descent, and 6) an analysis on the computation time.

6 Conclusion

We presented SYNCDIFFUSION, a diffusion synchronization module designed to generate coherent montages through joint diffusions. Using a pre-trained diffusion model, we propose guiding the reverse process by updating the noisy images at each intermediate step using gradient descent. This update is based on a perceptual similarity loss calculated with the predictions of the denoised images. While we experimented with our method to generate panorama images, the idea can be applied to create various montages, including 360 images and textures of 3D models. We plan to further investigate these possibilities in future work.

Potential Negative Societal Impacts Image generative models can potentially generate deepfakes, images resembling copyrighted material, biased or discriminatory images, and harmful outputs. Future research is needed to advance the detection of manipulated content and establish societal barriers to protect intellectual property.

References

- [1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. In *SIGGRAPH*, 2023.
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022.
- [3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023.
- [4] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.
- [5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your Latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.
- [6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.
- [7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.
- [8] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2Tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023.
- [9] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023.
- [10] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *NeurIPS*, 2022.
- [11] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.
- [12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.
- [13] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *CVPR*, 2023.
- [14] Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction-tuned lilm and latent diffusion model. *arXiv preprint arXiv:2304.13731*, 2023.
- [15] Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. GIQA: Generated image quality assessment. In *ECCV*, 2020.
- [16] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023.
- [17] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021.
- [18] 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 *NIPS*, 2018.
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *ICLR*, 2022.
- [21] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-An-Audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023.
- [22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.
- [23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.
- [24] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023.
- [25] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation. In *ICLR*, 2023.
- [26] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhihai Xu, Qi Li, and Yuetong Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 2022.

- [27] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *CVPR*, 2023.
- [28] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.
- [29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.
- [30] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.
- [31] Midjourney. <https://www.midjourney.com>.
- [32] Rudimentary framework for running two-alternative forced choice (2afc) perceptual studies on mechanical turk. <https://github.com/brownvc/mturk-2afc>.
- [33] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.
- [34] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.
- [35] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023.
- [36] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [37] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.
- [38] Elad Richardson, Gal Metzger, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. TEXTure: Text-guided texturing of 3D shapes. *arXiv preprint arXiv:2302.01721*, 2023.
- [39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
- [40] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dream-Booth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.
- [41] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022.
- [42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.
- [43] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [44] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.
- [45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.
- [46] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-It-3D: High-fidelity 3D creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023.
- [47] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, 2022.
- [48] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3D: Zero-shot text-to-3D synthesis using 3D shape prior and text-to-image diffusion models. In *CVPR*, 2023.
- [49] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE Transactions on Audio, Speech and Language Processing*, 2022.
- [50] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.
- [51] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. DiffCollage: Parallel generation of large content with diffusion models. In *CVPR*, 2023.
- [52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Appendix

In this supplementary document, we first show more qualitative results with various prompts in Sec. 6.1. Sec. 6.2 includes a detailed quantitative evaluation of our method with gradient descent weights $w = 0, 5, 10, 15$, and 20 . Sec. 6.3 shows quantitative evaluation of our method on generating panoramas of different resolutions. In Sec. 6.4, we show the comparisons of our method with different perceptual similarity loss functions. Sec. 6.5 shows an ablation study result substituting Eq. 14 in the main paper with Eq. 13. Sec. 6.6 analyzes the computation time of SYNCDIFFUSION. Sec. 6.7 explains the details of our user study. Lastly, Sec. 6.8 provides additional qualitative comparisons.

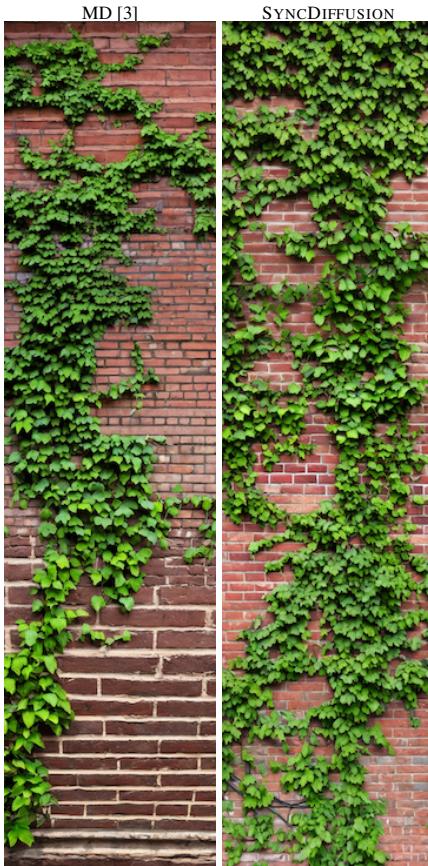
6.1 More Qualitative Results with Various Prompts

More qualitative results with various prompts are shown in the figures below. The resolutions of images are 512×3072 for horizontal panoramas and 2048×512 for vertical panoramas.





“a photo of vines on a brick wall”



“a bird’s eye view of an alley with shops”



“a beach with palm trees”



“a photo of a grassland with animals”



“a cinematic view of a castle in the sunset”

MD [3]



SYNC-
DIFFUSION



“a film photo of a beachside street under the sunset”

MD [3]



SYNC-
DIFFUSION



“a photo of a beautiful ocean with coral reef”

MD [3]



SYNC-
DIFFUSION

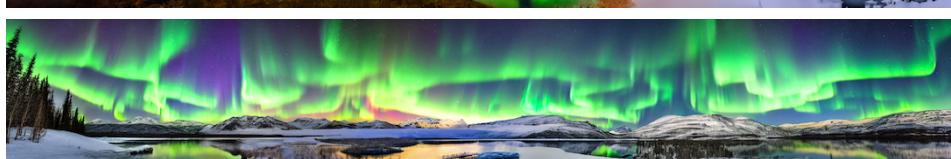


“a photo of a lake under the northern lights”

MD [3]



SYNC-
DIFFUSION



6.2 Details About Quantitative Evaluation

Tab. 2 shows the detailed quantitative results of SYNCDIFFUSION on panorama generation, reported in Fig. 5 of the main paper. Here we additionally show the results with the gradient descent weight $w = 5$ and $w = 15$, along with the weights $w = 10$ and $w = 20$ reported in Sec. 5 of the main paper. Note that we used KNN-GIQA [15] with $K = 8$ to measure Mean-GIQA in all our experiments. As shown in Tab. 2 (rows 3-7), as the gradient descent weight w increases from 0 to 20, the results of our method display a significant improvement in global coherence, as shown in Intra-LPIPS [52] which decreases from 0.69 ($w = 0$) to 0.56 ($w = 20$), and Intra-Style-L [22] which decreases from 2.98 ($w = 0$) to 1.39 ($w = 20$). These results are more apparent in the line plot of Intra-LPIPS and Intra-Style-L displayed in Fig. 8. Fig. 7 shows the qualitative comparison of the panorama images generated with different weights.

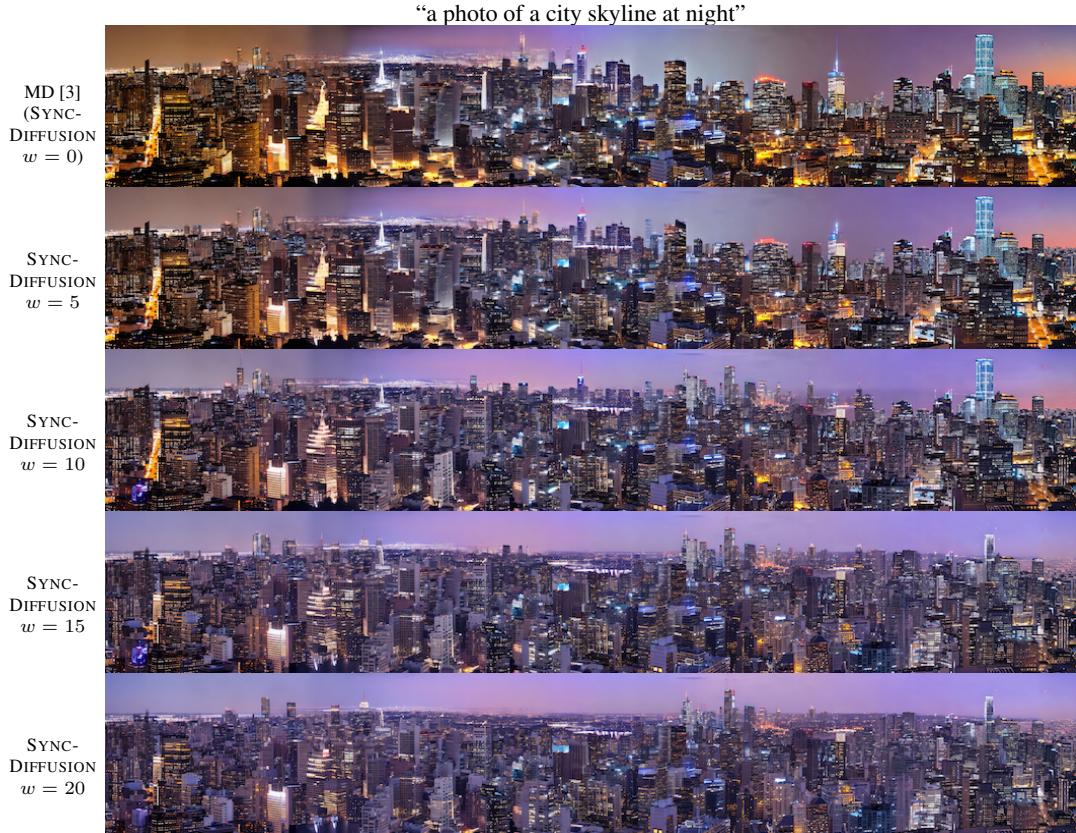


Figure 7: Qualitative comparison of different weights w . As w increases, the generated panorama image gradually becomes globally coherent. Compared to MD, as w increases, the left and right sides of the panorama image become more coherent.

6.3 Quantitative Evaluation on Different Resolutions

We show the quantitative results on different resolutions in Tab. 2 (row 10-13). In addition to the original 512×3072 resolution, Tab. 2 shows the quantitative comparison of SYNCDIFFUSION and MultiDiffusion [3] for smaller resolution panoramas (512×2048 and 512×1024). In Fig. 2, when comparing the rows 10 and 11, 8 and 9, 3 and 7 respectively, the gap of Intra-LPIPS between our method and MultiDiffusion is preserved (0.13, 0.14, and 0.13, respectively), meaning that our method constantly produces more coherent panoramas than MultiDiffusion regardless of the resolution. The gap of Intra-Style-L between our method and MultiDiffusion even increases as the resolution increases (1.48, 1.57, and 1.59, respectively). On the other hand, the gap of FID and KID between the two methods also increases as the resolution increases: 9.69, 10.26, 11.08 for FID and 7.96, 10.19, 11.96 for KID. We hypothesize that the increase in FID and KID of our method with longer panoramas is due to the tendency that for certain images it is more difficult to find other images that

		Intra-LPIPS ↓	Intra-Style-L ↓ ($\times 10^{-3}$)	Mean-GIQA ↑ ($\times 10^{-3}$)	FID ↓	KID ↓ ($\times 10^{-3}$)	Mean-CLIP-S ↑
1	SD [39]	0.74 ± 0.07	8.40 ± 6.27	26.70 ± 6.90	28.31 ± 10.89	$< 0.01 \pm 0.13$	31.63 ± 1.89
2	BLD [1]	0.58 ± 0.06	4.64 ± 3.32	24.27 ± 6.19	84.29 ± 36.74	66.54 ± 37.30	31.41 ± 1.66
SYNCDIFFUSION with Various Gradient Descent Weight w (Eq. 14)							
3	$w = 0$ (MD [3])	0.69 ± 0.09	2.98 ± 2.41	28.54 ± 7.99	33.52 ± 12.43	9.04 ± 4.23	31.77 ± 2.32
4	$w = 5$	0.64 ± 0.07	2.15 ± 1.61	28.58 ± 7.84	35.57 ± 12.43	12.09 ± 4.98	31.85 ± 2.33
5	$w = 10$	0.60 ± 0.07	1.75 ± 1.31	28.28 ± 7.54	38.24 ± 15.24	15.08 ± 6.77	31.90 ± 2.33
6	$w = 15$	0.58 ± 0.06	1.54 ± 1.21	27.74 ± 7.19	41.04 ± 16.74	17.47 ± 8.29	31.86 ± 2.25
7	$w = 20$	0.56 ± 0.06	1.39 ± 1.15	27.17 ± 6.66	44.60 ± 18.45	21.00 ± 11.06	31.84 ± 2.19
Panorama Size: 512×2048							
8	MD [3]	0.69 ± 0.09	2.96 ± 2.41	28.33 ± 7.79	33.07 ± 12.38	8.58 ± 3.99	31.77 ± 2.14
9	SYNCDIFFUSION	0.55 ± 0.06	1.39 ± 1.19	27.08 ± 6.65	43.33 ± 17.98	18.77 ± 10.19	31.77 ± 2.14
Panorama Size: 512×1024							
10	MD [3]	0.66 ± 0.09	2.57 ± 1.97	28.17 ± 7.54	30.66 ± 11.79	5.24 ± 3.04	31.73 ± 2.22
11	SYNCDIFFUSION	0.53 ± 0.06	1.09 ± 0.77	26.41 ± 6.38	40.35 ± 16.43	13.20 ± 7.61	31.71 ± 2.01
SYNCDIFFUSION Ablation Study							
12	Eq. 14 → Eq. 13	0.68 ± 0.09	2.95 ± 2.39	28.53 ± 7.99	33.58 ± 0.09	9.15 ± 4.25	31.78 ± 2.32
13	Style Loss [22]	0.64 ± 0.10	1.08 ± 1.10	25.74 ± 6.31	73.05 ± 37.56	56.64 ± 39.58	31.15 ± 2.32

Table 2: Quantitative results on panorama generation.

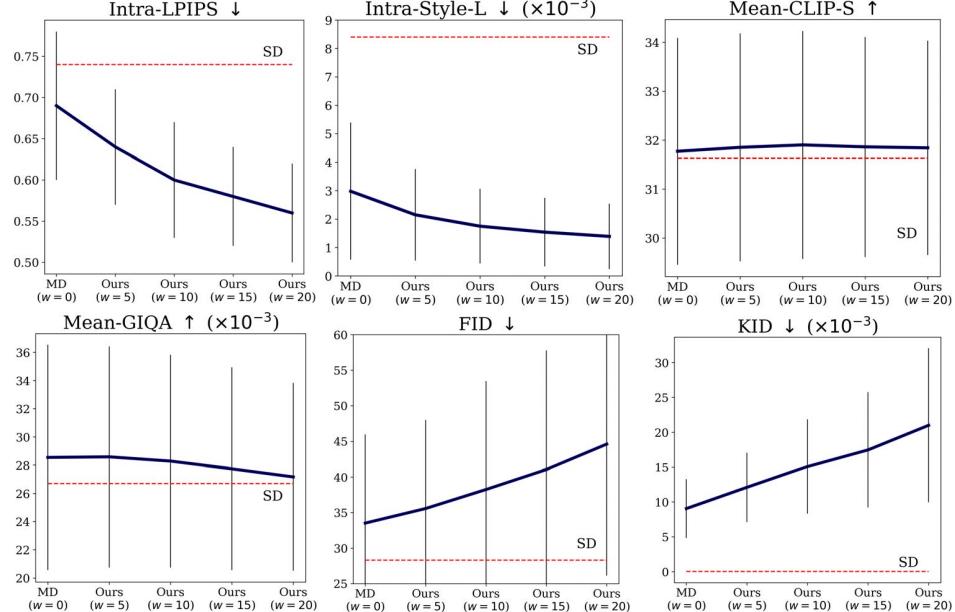


Figure 8: Line plots of the quantitative results shown in Tab. 2 with varying gradient descent weight w . The dashed lines (SD) represent the evaluation results of the Stable Diffusion [39] reference set images. The vertical lines represent the standard deviation.

can be merged into a single coherent panorama. The above results indicate that while our method can guide the joint diffusion process to generate highly coherent images regardless of the resolution, generating longer panoramas that are globally coherent can lead to a decrease in the diversity of generations, thus resulting in a negative effect on FID and KID.

6.4 Results of SYNCDIFFUSION with Style Loss

As described in Sec. 4 in the main paper, any off-the-shelf perceptual similarity loss can be utilized in our method. Here we show the results of our method with Style Loss [22] as the loss function \mathcal{L} in Eq. 14 in the main paper. Fig. 9 shows panorama images generated by MultiDiffusion [3], and our method with LPIPS [52] and Style Loss [22] as the perceptual similarity loss function, respectively. To observe visible changes in the panorama outputs, we multiplied 10^6 to the Style Loss and set the gradient descent weight w to 0.1. Tab. 2 (row 13) demonstrates that SYNCDIFFUSION with Style Loss achieves better coherence compared to MultiDiffusion as measured by Intra-LPIPS and Intra-Style-L, while showing a negative effect on the metrics regarding fidelity: Mean-GIQA, FID, and KID. Note that Intra-Style-L is significantly decreased as the guidance was provided with Style Loss. The second row in Fig. 9 shows that Style Loss can guide the joint diffusion processes to generate a globally coherent panorama image, as compared to the MultiDiffusion output in the first row.

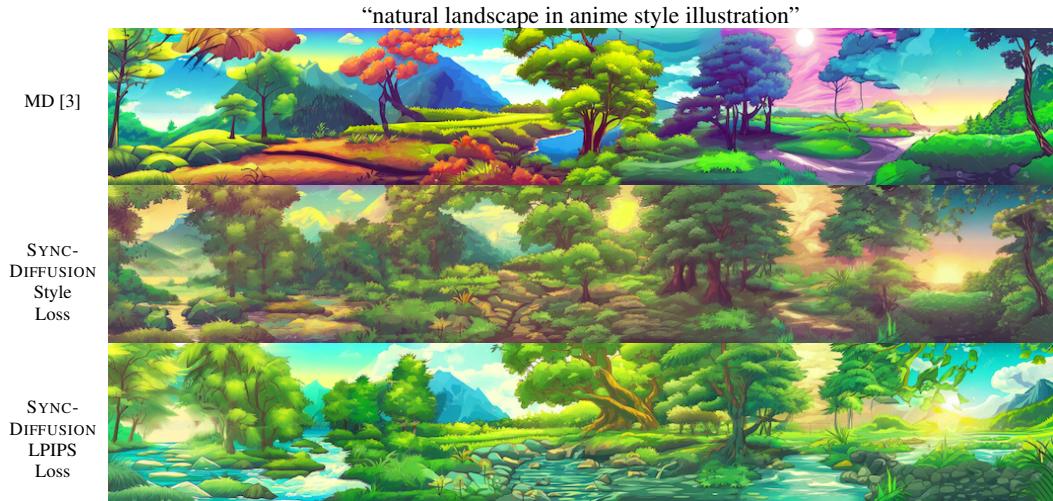


Figure 9: Qualitative comparisons of MultiDiffusion and SYNCDIFFUSION with Style Loss.

6.5 Ablation on Predicting the Foreseen Denoised Observation

Tab. 2 (row 12) shows the quantitative comparison of the panorama generations using our method and after substituting the original Eq. 14 in the main paper with Eq. 13 in which the noisy image $\mathbf{x}_t^{(i)}$ is decoded instead of utilizing the foreseen denoised observation $\phi_\theta(\mathbf{x}_t^{(i)}, t)$. Although Intra-LPIPS is still slightly reduced compared to MultiDiffusion when using Eq. 13, the change is negligible compared to that of the original formulation Eq. 14. This result is straightforward as measuring the perceptual loss between noisy images would not provide meaningful guidance to the diffusion process, whereas comparing the perceptual similarity of *foreseen denoised* observations can give a meaningful guidance for global coherence.

6.6 Analysis on the Computation Time

As our SYNCDIFFUSION module requires the gradient descent computation, it introduces additional computational overhead during the sampling process. Since our method is based on the DDIM reverse process with 50 timesteps, the gradient descent is applied 50 times. We define f as the frequency of the gradient descent during the DDIM reverse process of SYNCDIFFUSION, with the default value of $f = 50$. Tab. 3 shows the quantitative results and the computation time when the gradient descent is performed 10 times ($f = 10$) and 5 times ($f = 5$) in total with uniform intervals, with the gradient descent weight fixed to $w = 20$. Although applying the gradient descent for every step leads to the highest global coherence with Intra-LPIPS of 0.56 and Intra-Style-L of 1.39, in practice applying the gradient descent for 5 or 10 times can still achieve meaningful improvement in the coherence compared to MultiDiffusion as shown in rows 2-4 of Tab. 3, while reducing the computation time compared to the $f = 50$ case. Note that Intra-LPIPS decreases from 0.69 to 0.62 and Intra-Style-L

decreases from 2.98 to 2.14 for $f = 10$. The resolution is fixed to 512×3072 , and each generation process is done on a single NVIDIA A100 GPU.

	Intra-LPIPS ↓	Intra-Style-L ↓ ($\times 10^{-3}$)	Mean-GIQA ↑ ($\times 10^{-3}$)	FID ↓	KID ↓ ($\times 10^{-3}$)	Mean-CLIP-S ↑	Time(s)
SD [39]	0.74 ± 0.07	8.40 ± 6.27	26.70 ± 6.90	28.31 ± 10.89	$<0.01 \pm 0.13$	31.63 ± 1.89	-
MD [3]	0.69 ± 0.09	2.98 ± 2.41	28.54 ± 7.99	33.52 ± 12.43	9.04 ± 4.23	31.77 ± 2.32	46.10 ± 1.07
SYNCDIFFUSION							
$f = 5$	0.64 ± 0.07	2.33 ± 1.83	28.44 ± 7.85	35.18 ± 13.31	11.43 ± 4.68	31.81 ± 2.24	81.17 ± 0.53
$f = 10$	0.62 ± 0.07	2.14 ± 1.72	28.43 ± 7.75	36.22 ± 14.03	12.84 ± 5.59	31.85 ± 2.27	104.83 ± 3.38
$f = 50$	0.56 ± 0.06	1.39 ± 1.15	27.17 ± 6.66	44.60 ± 18.45	21.00 ± 11.06	31.84 ± 2.19	339.53 ± 2.85

Table 3: Analysis on the computation time of our SYNCDIFFUSION and MultiDiffusion [3].

6.7 Details of User Study

Fig. 10 shows screenshots of our user study. We set all participants to be Amazon Mechanical Turk Masters who are located in the US. The average time that participants spent on solving a set of 25 problems (including the vigilance tasks) was 248.21 seconds, and we compensated them with a payment of 0.76\$ per person. This is equal to 11.02\$ per hour, which exceeds the US federal minimum wage.

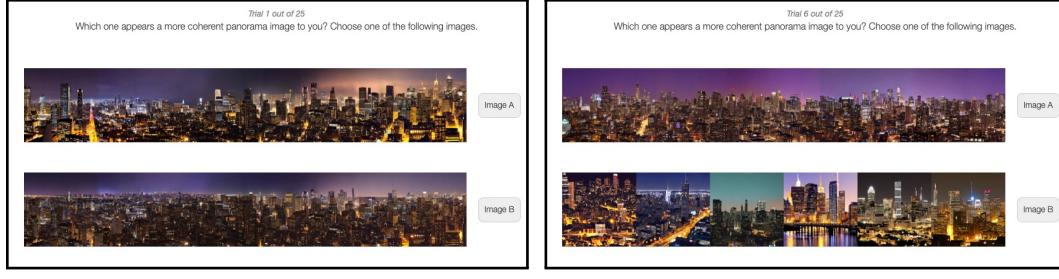


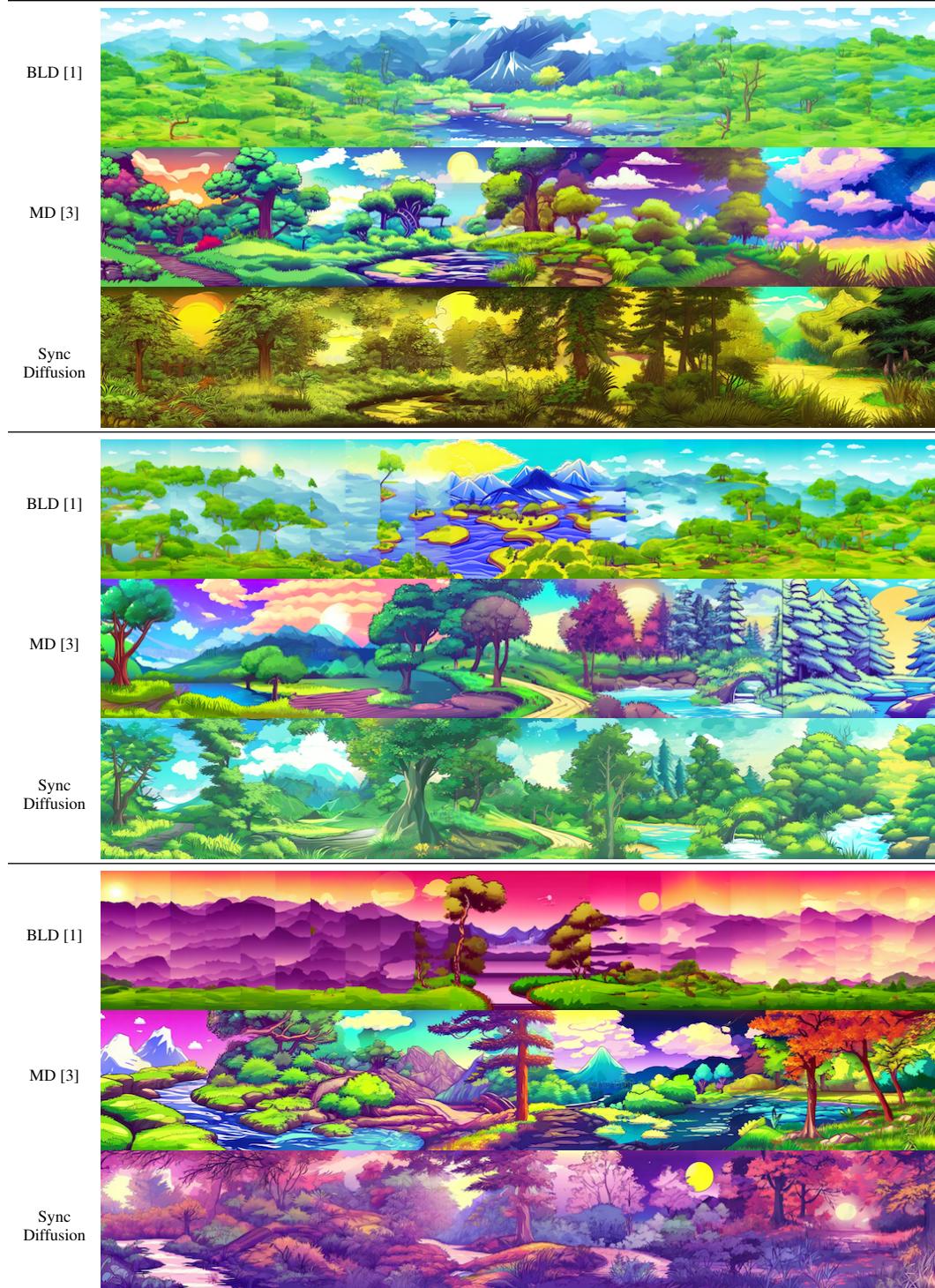
Figure 10: User study screenshots.

6.8 More Qualitative Results

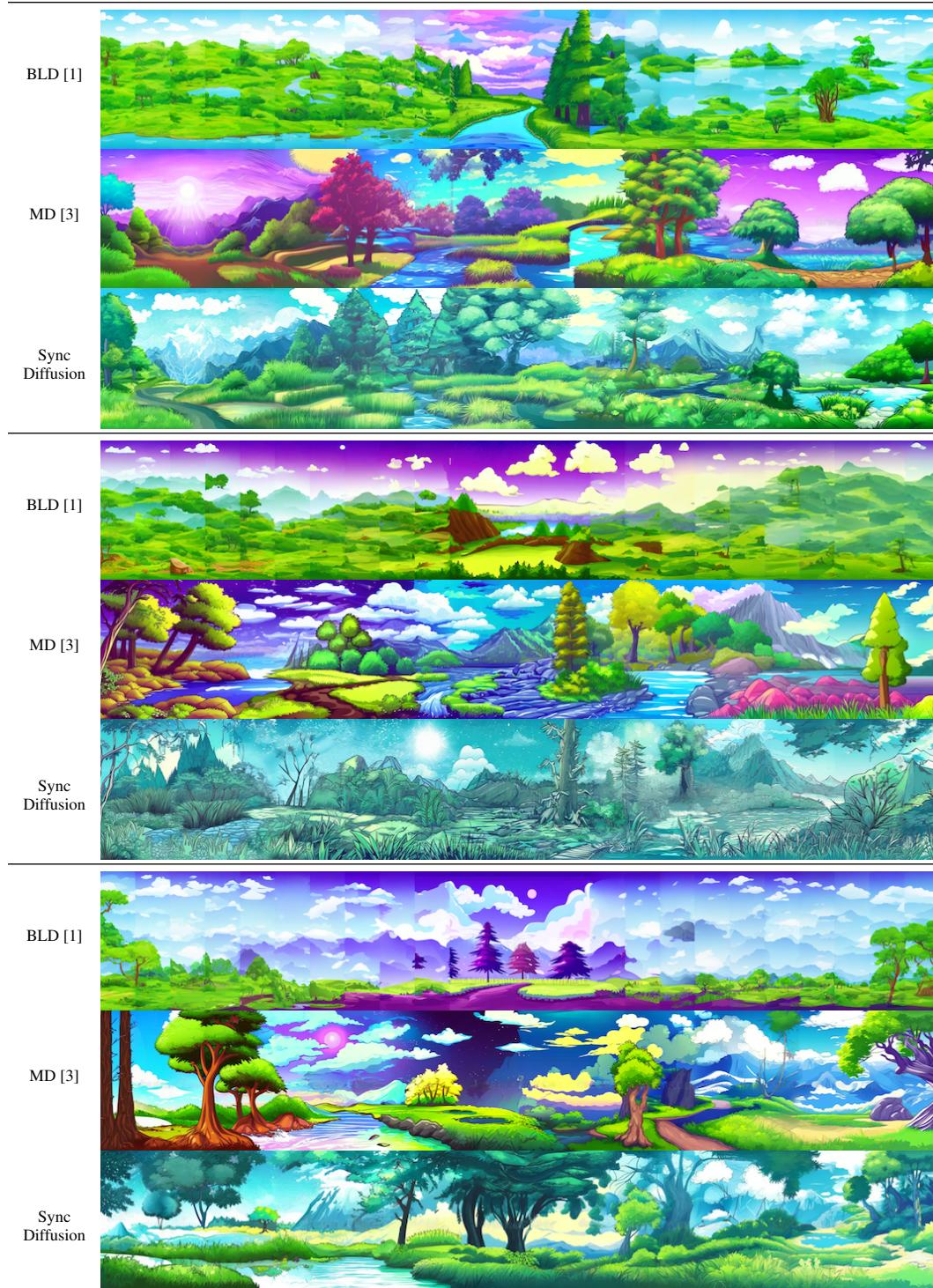
More qualitative results are shown in the figures below.



“natural landscape in anime style illustration”



“natural landscape in anime style illustration”



“a photo of a forest with a misty fog”

BLD [1]



MD [3]



Sync
Diffusion



“a photo of a mountain range at twilight”

BLD [1]



MD [3]



Sync
Diffusion



BLD [1]



MD [3]



Sync
Diffusion



“a photo of a mountain range at twilight”

BLD [1]



MD [3]



Sync
Diffusion



BLD [1]



MD [3]



Sync
Diffusion



BLD [1]



MD [3]



Sync
Diffusion



“cartoon panorama of spring summer beautiful nature”

