StochSync: Stochastic Diffusion Synchronization for Image Generation in Arbitrary Spaces

Anonymous authors
Paper under double-blind review

Figure 1: Assorted mesh textures and panoramas generated using StochSync, including one in the background (environment map), which is a 360° panorama. StochSync extends the capabilities of image diffusion models trained in square spaces to produce images in arbitrary spaces such as cylinders, spheres, tori, and mesh surfaces.

Abstract

We propose a zero-shot method for generating images in arbitrary spaces (e.g., a sphere for 360° panoramas and a mesh surface for texture) using a pretrained image diffusion model. The zero-shot generation of various visual content using a pretrained image diffusion model has been explored mainly in two directions. First, Diffusion Synchronization–performing reverse diffusion processes jointly across different projected spaces while synchronizing them in the target space–generates high-quality outputs when enough conditioning is provided, but it struggles in its absence. Second, Score Distillation Sampling–gradually updating the target space data through gradient descent–results in better coherence but often lacks detail. In this paper, we reveal for the first time the interconnection between these two methods while highlighting their differences. To this end, we propose StochSync, a novel approach that combines the strengths of both, enabling effective performance with weak conditioning. Our experiments demonstrate that StochSync provides the best performance in 360° panorama generation (where image conditioning is not given), outperforming previous finetuning-based methods, and also delivers comparable results in 3D mesh texturing (where depth conditioning is provided) with previous methods.

1 Introduction

Diffusion models pretrained on billions of images (Rombach et al., 2022; Midjourney) have demonstrated remarkable capabilities in various zero-shot applications. A notable example is the zero-shot generation of diverse visual data, including arbitrary-sized images (Bar-Tal et al., 2023; Lee et al., 2023), 3D mesh textures (Cao et al., 2023), ambiguous images (Geng et al., 2024b), and zoomed-in images (Wang et al., 2024a; Geng et al., 2024a). This extension to other types of data is achieved through mapping from the space in which the diffusion models are trained (referred to as the *instance* *space*) to the space where the new data is generated (the *canonical space*). For instance, while a 2D square is the instance space for typical image diffusion models, a cylinder or a sphere serves as the canonical space for generating $360°$ panoramic images, and a 3D mesh surface becomes the canonical space for mesh texture generation. Examples are shown in Fig. 1. Such zero-shot generation in the canonical space allows for the effective production of various types of data without the need for new data collection or training a separate generative model for each data type.

There have been two main approaches to addressing this problem. The first is Diffusion Synchronization (DS) (Bar-Tal et al., 2023; Kim et al., 2024a), which performs the reverse generative process of diffusion models jointly across multiple instance spaces while synchronizing intermediate outputs by mapping them to the canonical space. This approach has been successfully applied to generating various types of data, though it has a notable limitation: synchronization often fails to converge when strong conditioning, such as depth images, is not provided. As a result, the generated outputs frequently exhibit visible seams and fail to smoothly combine multiple projections from the instance spaces. This becomes a critical drawback for certain applications, such as $360°$ panoramic images, where image conditioning may not be available.

The other line of work is Score Distillation Sampling (SDS) (Poole et al., 2023) and its variants (Lukoianov et al., 2024; Liang et al., 2024). Unlike DS, SDS does not perform the reverse diffusion process but instead uses gradient-descent-based updates from various instance spaces to the canonical space. SDS has been widely applied to the generation of different types of visual data and, compared to DS, has shown greater robustness in scenarios where no image conditioning is provided. However, its quality is less realistic, as the generation process is not based on the reverse diffusion process, which diffusion models are specifically designed for.

In this work, we introduce a novel method named Stochastic Diffusion Synchronization, `StochSync` for short, which combines the best features of the two aforementioned approaches to achieve superior performance in unconditional canonical data generation. `StochSync` is based on our key insights from analysis on the similarities and differences between DS and SDS. Specifically, we observe that each step of SDS can be interpreted as a one-step refinement in DDIM Song et al. (2021a) while maximizing stochasticity in the denoising. We incorporate this maximum stochasticity into DS, resulting in better coherence across instance spaces and improved convergence. To enhance the realism as well, we propose replacing the prediction of the clean sample at each denoising step from Tweedie's formula with a multi-step denoising process, and also using non-overlapping views for the instance space while achieving synchronization over time through the overlap of views across different time steps. Notably, from the SDS perspective, `StochSync` can also be seen as modifying SDS by changing the random time sampling to a decreasing time schedule, resembling the reverse process, and by replacing the gradient descent with fully minimizing the $l2$ loss.

In the experiments, we test `StochSync` on two applications: $360°$ panoramic image generation and mesh texture generation. The former represents the unconditional case (except for a text prompt), while the latter is the conditional case with a depth map as the input. For the panoramic image generation, we demonstrate state-of-the-art performance compared to previous zero-shot (Cai et al., 2024) and finetuning-based methods (Tang et al., 2023b; Zhang et al., 2024a). Notably, our zero-shot method does not suffer from overfitting issues, unlike methods finetuned on small-scale panorama datasets (Chang et al., 2017), and it avoids geometric distortions that occur with inpainting-based methods (Cai et al., 2024). For mesh texture generation, although our method is designed to focus on the unconditional case, it demonstrates comparable results to previous DS methods (Kim et al., 2024a) and outperforms other prior works (Youwang et al., 2023; Zeng et al., 2024; Chen et al., 2023a; Richardson et al., 2023).

2 Related Work

In this section, we first review two approaches that generate samples in canonical space by leveraging pretrained diffusion models trained in instance space: Diffusion Synchronization and Score Distillation Sampling. We then discuss these approaches, along with other related works, in the context of two applications: panorama generation and 3D mesh texturing.

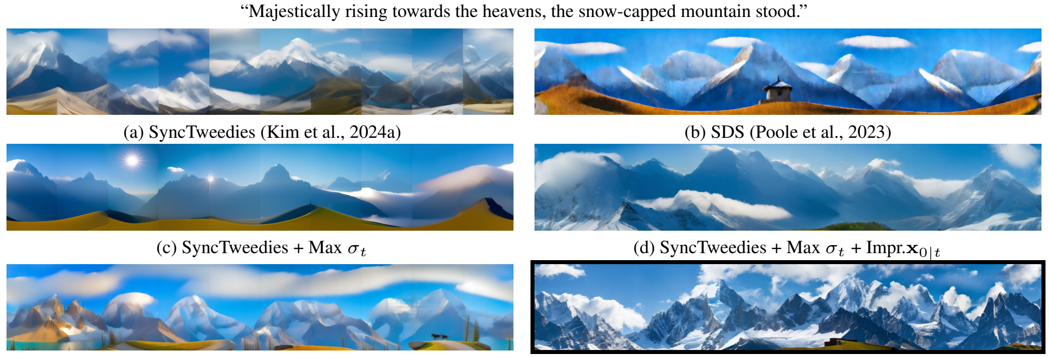
Diffusion Synchronization (DS). Liu et al. (2022) was among the first works to utilize DS, focusing on compositional image generation. Subsequent works, such as (Bar-Tal et al., 2023; Lee et al., 2023), extended DS to support image generation at arbitrary resolutions. Beyond images, DS has been widely applied to generate textures for 3D meshes (Liu et al., 2023; Zhang et al., 2024b; Chen et al., 2024a), long animations (Shafir et al., 2024), and visual spectrograms (Chen et al., 2024b). Recently, Kim et al. (2024a) provided an in-depth analysis of previous DS-based methods and introduced a method demonstrating superior performance across diverse applications, which we will use as the base DS method. While DS performs well under strong input conditions (e.g.,depth images), it struggles to generate plausible data points when the input conditions are weak.

Score Distillation Sampling (SDS). DreamFusion (Poole et al., 2023) first introduced SDS to generate 3D objects from text prompts, and several subsequent works have aimed to improve its quality (Wang et al., 2024b; Katzir et al., 2023; Zhu et al., 2023) and running time (Huang et al., 2023; Tang et al., 2023a). ISM (Liang et al., 2024) and SDI (Lukoianov et al., 2024) utilized DDIM inversion to obtain noisy data points. Beyond 3D generation, SDS has been widely applied in various fields, including image editing (Hertz et al., 2023), 3D scene editing (Koo et al., 2024; Park et al., 2023), and mesh deformation (Yoo et al., 2024). However, SDS-based methods often produce suboptimal samples lacking fine details compared to reverse process outputs. We also discuss the differences between our method and recent variants of SDS in Sec. 6.

Panorama Generation. In text-conditioned panorama generation, Text2Light (Chen et al., 2022) employed VQGAN (Esser et al., 2021) with a multi-stage pipeline. With the release of image diffusion models trained on large-scale datasets (Rombach et al., 2022), approaches leveraging pretrained diffusion models have gained attention. MVDiffusion (Tang et al., 2023b) and PanFusion (Zhang et al., 2024a) finetune these pretrained models using a panoramic images dataset (Chang et al., 2017). However, finetuning diffusion models on a small dataset risks overfitting, reducing their generalizability. In contrast, SyncTweedies (Kim et al., 2024a) employs DS for zero-shot panorama generation but relies on depth map conditions, which are not commonly available in practice. L-MAGIC (Cai et al., 2024), on the other hand, adopts an inpainting diffusion model, sequentially filling in the panoramic images. However, this iterative process cannot refine previous predictions, leading to error accumulation and often resulting in wavy panoramas.

Mesh Texturing. 3D mesh texturing using image diffusion models has gained significant attention. Among these approaches, Paint3D (Zeng et al., 2024) finetunes a pretrained diffusion model on a synthetic 3D mesh dataset (Deitke et al., 2023), but this often results in unrealistic texture images due to overfitting to the synthetic dataset. For zero-shot approaches, previous works have utilized SDS to update the texture of 3D meshes (Metzer et al., 2023; Chen et al., 2023b; Youwang et al., 2023). DS is also widely used for 3D mesh texturing, with previous works (Liu et al., 2023; Zhang et al., 2024b; Kim et al., 2024a) averaging the one-step predicted clean samples across multiple denoising processes. Another line of research explores the outpainting approach (Chen et al., 2023a; Richardson et al., 2023), where the 3D mesh is textured iteratively, often resulting in textures with visible seams.

“Majestically rising towards the heavens, the snow-capped mountain stood.”



(a) SyncTweedies (Kim et al., 2024a) (b) SDS (Poole et al., 2023)
(c) SyncTweedies + Max σ_t (d) SyncTweedies + Max σ_t + Impr.$\mathbf{x}_{0|t}$
(e) SDI (Lukoianov et al., 2024) (f) `StochSync`

Figure 2: A comparison of SyncTweedies (Kim et al., 2024a), a synchronization method, SDS (Poole et al., 2023), and `StochSync` which uses SyncTweedies as a base and incorporates maximum stochasticity (Max σ_t), multi-step $\mathbf{x}_{0|t}$ computation (Impr. $\mathbf{x}_{0|t}$), and non-overlapping view sampling (N.O. Views), alongside others that use only a subset of these components.

3 Problem Definition and Overview

We propose a method for generating data points in one space (referred to as the *canonical space* $\mathcal{Z}$) using a pretrained diffusion model that has been trained on *another space* (referred to as the instance space $\mathcal{X}$), where the mapping from the canonical space to the instance space is known. For example, the canonical space could be a sphere representing $360°$ panoramas, or a 3D mesh surface for creating mesh textures, and the instance space is a 2D square, the space for most pretrained image diffusion models. In general, a region of the canonical space is mapped to the instance space through a specific view. The mapping from a region of the canonical space to the instance space through a view $\mathbf{c}$ is represented by the projection operation $f_{\mathbf{c}}(\mathbf{z}) : \mathcal{Z}_{\mathbf{c}} \to \mathcal{X}$, where $\mathbf{z} \in \mathcal{Z}_{\mathbf{c}} \subseteq \mathcal{Z}$. Our objective is to produce realistic data points in the canonical space without using any generative model trained on samples in that space, but by leveraging pretrained diffusion models in the instance spaces and their multiple denoising processes from different views. This approach can extend the capabilities of pretrained diffusion models to produce diverse types of data, eliminating the need to collect large-scale data and train separate generative models.

In the following sections, we first review the reverse process of a diffusion model (Section 4) and two approaches, Diffusion Synchronization (DS) and Score Distillation Sampling (SDS), which generate data points in the canonical space by leveraging pretrained diffusion models in instance spaces (Section 5). Based on our analysis of the connections and differences between these methods, we propose a novel approach that combines the best features of both and provides an interpretation of the method from the perspectives of DS and SDS (Section 6).

4 Diffusion Reverse Process

The forward process of a diffusion model (Sohl-Dickstein et al. (2015); Ho et al. (2020); Song et al. (2021b)) sequentially corrupts sample data using a predefined variance schedule $\alpha_1, \dots, \alpha_T$, where one can sample $\mathbf{x}_t$ at arbitrary timestep t from a clean sample $\mathbf{x}_0$:

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

Song et al. (2021a) propose DDIM, a diffusion reverse process generalizing DDPM Ho et al. (2020), by defining the posterior distribution $q_{\sigma_t}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ with a parameter σ_t determining the level of stochasticity as follows:

$$q_{\sigma_t}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mu_{\sigma_t}(\mathbf{x}_0, \mathbf{x}_t), \sigma_t^2 \boldsymbol{I}\right), \tag{2}$$

$$\text{where} \quad \mu_{\sigma_t}(\mathbf{x}_0, \mathbf{x}_t) = \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}. \tag{3}$$

In the reverse process, the transitional likelihood distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ becomes the same with the posterior distribution in Eq. 2 while the clean sample $\mathbf{x}_0$ is approximated using the noise predictor $\epsilon_\theta(\mathbf{x}_t, y)$, where y is the input condition (e.g.,a text prompt); note that the time input is omitted for simplicity. When $\boldsymbol{\epsilon}_t = \epsilon_\theta(\mathbf{x}_t, y)$, the prediction of clean sample $\mathbf{x}_0$ at timestep t, denoted as $\mathbf{x}_{0|t}$, is derived as follows based on Tweedie's formula (Robbins (1956)):

$$\mathbf{x}_{0|t} = \psi(\mathbf{x}_t, \boldsymbol{\epsilon}_t) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t}{\sqrt{\alpha_t}}. \tag{4}$$

A clean data sample $\mathbf{x}_0$ is then generated by first sampling standard Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and gradually denoising it over time by iteratively sampling a noisy data point $\mathbf{x}_t$ from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. The mapping from a noisy data point $\mathbf{x}_t$ to $\mathbf{x}_0$ becomes deterministic when $\sigma_t = 0$ for all t and is is equivalent to solving an ODE (Song et al., 2021b; Chen et al., 2018) with a specific discretization.

Reverse Process from the Perspective of $\mathbf{x}_{0|t}$. Here, to connect the reverse process of DDIM to the algorithms to be introduced in the next section, we reinterpret the reverse denoising process as an iterative *refinement* process of the prediction of clean sample $\mathbf{x}_{0|t}$. See Alg. 1, where $\mathbf{x}_{0|t}$ and $\boldsymbol{\epsilon}_t$ are computed at each timestep. Note that the mean of the likelihood distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in Eq. 3 can be rewritten in terms of $\mathbf{x}_0$ and $\boldsymbol{\epsilon}_t$:

$$\mu_{\sigma_t}(\mathbf{x}_0, \boldsymbol{\epsilon}_t) = \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \boldsymbol{\epsilon}_t. \tag{5}$$

Apart from setting $\sigma_t = 0$, one can consider a special case when $\sigma_t = \sqrt{1-\alpha_{t-1}}$, which maximizes the level of stochasticity during the sampling process. This cancels out the noise prediction term $\boldsymbol{\epsilon}_t$ in Eq. 5. We denote this case by overriding $\mu_{\sigma_t}(\cdot, \cdot)$ with $\mu^*(\cdot)$, which now takes a single parameter $\mathbf{x}_0$:

$$\mu^*(\mathbf{x}_0) = \sqrt{\alpha_{t-1}}\mathbf{x}_0. \tag{6}$$

Algorithm 1: Diffusion Reverse Process

Inputs: y: Input text prompt

Outputs: $\mathbf{x}_0$: An instance space sample aligned with y

Function `Reverse Process` (y):
- $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- $\boldsymbol{\epsilon}_T \leftarrow \epsilon_\theta(\mathbf{x}_T, y)$
- $\mathbf{x}_{0|T} \leftarrow \psi(\mathbf{x}_T, \boldsymbol{\epsilon}_T)$
- **for** $t = T \dots 2$ **do**
  - $\mathbf{x}_{t-1} \sim \mathcal{N}(\mu_{\sigma_t}(\mathbf{x}_{0|t}, \boldsymbol{\epsilon}_t), \sigma_t^2 \boldsymbol{I})$ // Eq. 5
  - $\boldsymbol{\epsilon}_{t-1} \leftarrow \epsilon_\theta(\mathbf{x}_{t-1}, y)$
  - $\mathbf{x}_{0|t-1} \leftarrow \psi(\mathbf{x}_{t-1}, \boldsymbol{\epsilon}_{t-1})$ // Eq. 4
- **end**

Algorithm 2: Diffusion Synchronization (DS)

Inputs: $\mathbf{z}$: A canonical space sample

y: Input text prompt; $\mathbf{c}^{1:N}$: A set of views.

Outputs: $\mathbf{z}$: Canonical space sample aligned with y

1 **Function** `DS` $(\mathbf{z}, y, \mathbf{c}^{1:N})$:
2 $\quad \mathbf{x}_T^{1:N} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3 $\quad$ **for** $i = 1 \dots N$ **do**
4 $\quad\quad \boldsymbol{\epsilon}_T^{(i)} \leftarrow \epsilon_\theta(\mathbf{x}_T^{(i)}, y)$
5 $\quad\quad \mathbf{x}_{0|T}^{(i)} \leftarrow \psi(\mathbf{x}_T^{(i)}, \boldsymbol{\epsilon}_T^{(i)})$ // Eq. 4
6 $\quad$ **end**
7 $\quad \mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|f_{\mathbf{c}^{(i)}}(\mathbf{z}) - \mathbf{x}_{0|T}^{(i)}\|^2$
8 $\quad$ **for** $t = T \dots 2$ **do**
$\quad\quad$ // $\mathbf{c}^{1:N}$ is fixed for all t.
9 $\quad\quad$ **for** $i = 1 \dots N$ **do**
10 $\quad\quad\quad \mathbf{x}_{0|t}^{(i)} \leftarrow f_{\mathbf{c}^{(i)}}(\mathbf{z})$
11 $\quad\quad\quad \mathbf{x}_{t-1}^{(i)} \sim \mathcal{N}(\mu_{\sigma_t}(\mathbf{x}_{0|t}^{(i)}, \boldsymbol{\epsilon}_t^{(i)}), \sigma_t^2 \boldsymbol{I})$ // Eq. 5
12 $\quad\quad\quad \boldsymbol{\epsilon}_{t-1}^{(i)} \leftarrow \epsilon_\theta(\mathbf{x}_{t-1}^{(i)}, y)$
13 $\quad\quad\quad \mathbf{x}_{0|t-1}^{(i)} \leftarrow \psi(\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\epsilon}_{t-1}^{(i)})$ // Eq. 4
14 $\quad\quad$ **end**
15 $\quad\quad \mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|f_{\mathbf{c}^{(i)}}(\mathbf{z}) - \mathbf{x}_{0|t-1}^{(i)}\|^2$
16 $\quad$ **end**

Algorithm 3: Score Distillation Sampling (SDS)

Inputs: $\mathbf{z}$: A canonical space sample

y: Input text prompt

Outputs: $\mathbf{z}$: Canonical space sample aligned with y

Function `SDS` $(\mathbf{z}, y)$:
- **while** $\mathbf{z}$ not converged **do**
  - $t \sim \mathcal{U}(0, T)$; $\mathbf{c} \leftarrow$ `SampleRandomView()`
  - $\mathbf{x}_{0|t} \leftarrow f_{\mathbf{c}}(\mathbf{z})$
  - // Noise prediction is not used and thus omitted.
  - $\mathbf{x}_{t-1} \sim \mathcal{N}(\mu^*(\mathbf{x}_{0|t}), \sigma_t^2 \boldsymbol{I})$ // Eq. 6
  - $\mathbf{x}_{0|t-1} \leftarrow \psi(\mathbf{x}_{t-1}, \epsilon_\theta(\mathbf{x}_{t-1}, y))$
  - $\mathbf{z} \leftarrow \mathbf{z} - w(t) \left[f_{\mathbf{c}}(\mathbf{z}) - \mathbf{x}_{0|t-1}\right] \frac{\partial f}{\partial \mathbf{z}}$
- **end**

Algorithm 4: **`StochSync`**

Inputs: $\mathbf{z}$: A canonical space sample

y: Input text prompt

Outputs: $\mathbf{z}$: Canonical space sample aligned with y

Function `StochSync` $(\mathbf{z}, y)$:
- $\mathbf{c}^{1:N} \leftarrow$ `SampleNonOverlappingViews` (N)
- $\mathbf{x}_T^{1:N} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- **for** $i = 1 \dots N$ **do**
  - $\mathbf{x}_{0|T}^{(i)} \leftarrow \mathcal{G}(\mathbf{x}_T^{(i)})$
- **end**
- $\mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|f_{\mathbf{c}^{(i)}}(\mathbf{z}) - \mathbf{x}_{0|T}^{(i)}\|^2$
- **for** $t = T \dots T_{stop} + 1$ **do**
  - $\mathbf{c}^{1:N} \leftarrow$ `SampleNonOverlappingViews` (N)
  - **for** $i = 1 \dots N$ **do**
    - $\mathbf{x}_{0|t}^{(i)} \leftarrow f_{\mathbf{c}^{(i)}}(\mathbf{z})$
    - // Noise prediction is not used and thus omitted.
    - $\mathbf{x}_{t-1}^{(i)} \sim \mathcal{N}(\mu^*(\mathbf{x}_{0|t}^{(i)}), \sigma_t^2 \boldsymbol{I})$ // Eq. 6
    - $\mathbf{x}_{0|t-1}^{(i)} \leftarrow \mathcal{G}(\mathbf{x}_{t-1}^{(i)})$
  - **end**
  - $\mathbf{z} \leftarrow \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|f_{\mathbf{c}^{(i)}}(\mathbf{z}) - \mathbf{x}_{0|t-1}^{(i)}\|^2$
- **end**

5 Diffusion Synchronization and Score Distillation Sampling

As methods leveraging pretrained diffusion models to generate data in other spaces, there have been mainly two approaches: Diffusion Synchronization (DS) (Liu et al., 2022; Geng et al., 2024b; Kim et al., 2024a) and Score Distillation Sampling (SDS) (Poole et al., 2023; Wang et al., 2024b; Lukoianov et al., 2024; Liang et al., 2024). In this section, we briefly review these methods, analyze the connections between them as well as their differences, and discuss the limitations of each method.

5.1 Diffusion Synchronization

The idea of Diffusion Synchronization (DS) (Liu et al., 2022; Geng et al., 2024b; Kim et al., 2024a) is to perform the reverse process jointly across multiple instance spaces while synchronizing the processes through mapping to the canonical space. Among the various options for synchronization, Kim et al. (2024a) have demonstrated that averaging the predictions of the clean samples $\mathbf{x}_{0|t}$ in the canonical space and then projecting it back to each instance space provides the best performance across a broad range of applications. Alg. 2 shows the pseudocode, which, at each step, performs one-step denoising of DDIM for each view (lines 10-11), updates the data point in the canonical space $\mathbf{z}$ while averaging $\mathbf{x}_{0|t}$ by solving a $l2$-minimization (line 13), and then projects $\mathbf{z}$ back to each space (line 9). The differences from the reverse process of DDIM (Alg. 1) are highlighted in blue.

For the stochasticity of the denoising process, typically deterministic DDIM reverse process ($\sigma_t = 0$) (Bar-Tal et al., 2023; Zhang et al., 2024b) or DDPM reverse process ($\sigma_t = \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$) (Liu et al., 2023) have been used.

Previous works have shown the effectiveness of the synchronization approach in generating various types of visual data using pretrained image diffusion models, including depth-conditioned panoramic images, textures of 3D meshes and Gaussians (Kim et al., 2024a; Liu et al., 2023). However, we have observed that this approach requires strong conditioning for each instance–such as depth images–to achieve optimal quality. In cases where the input condition is not provided, such as generating depth-free $360°$ panoramas, the outputs tend to show seams as shown in Fig. 2(a), mainly due to the wider data distribution and thus difficulties in achieving convergence during synchronization.

5.2 Score Distillation Sampling

Score Distillation Sampling (SDS) (Poole et al., 2023) and its variants (Wang et al., 2024b; Lukoianov et al., 2024; Liang et al., 2024) are alternatives for generating samples in different spaces. Unlike DS, SDS does not use the reverse diffusion process but instead employs gradient-descent-based updates. The motivation behind SDS is to leverage the loss function from noise predictor training to discriminate real data points while projecting the canonical data point $f_{\mathbf{c}}(\mathbf{z})$, corrupting it through the forward process, and then predicting the added noise from it.

To clarify the similarities and differences between SDS and DS, we provide a different perspective on understanding SDS, as shown in Alg. 3, aligning each computation with those in DS (Alg. 2). There are several key differences, highlighted as green in Alg. 3. First, the timestep t is not decreased from T to 1 but is randomly sampled until convergence (line 3). Second, while synchronization approaches typically make the reverse process deterministic (Bar-Tal et al., 2023; Zhang et al., 2024b) or identical to DDPM (Liu et al., 2023), SDS uses *maximum stochasticity* ($\sigma_t = \sqrt{1-\alpha_{t-1}}$), thus eliminating the need to maintain the noise ϵ_t. Third, the prediction of the clean sample is updated to the canonical space not by solving the $l2$ minimization but by performing a single gradient descent step (line 7). SDS was originally introduced to perform gradient descent for the loss $\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{x}_{t-1}, y)\|^2$ (while omitting the gradient of the U-Net), where $\boldsymbol{\epsilon}$ is the standard normal sample used in $\mathbf{x}_{t-1}$ sampling, i.e.,$\mathbf{x}_{t-1} = \mu^*(\mathbf{x}_{0|t}) + \sigma_t\boldsymbol{\epsilon}$ (line 5), while it is equivalent to the loss used in DS, $\|f_{\mathbf{c}}(\mathbf{z}) - \mathbf{x}_{0|t-1}\|^2$, up to a scale as explained in **Appendix** (Sec. A).

As observed in previous works (Kim et al., 2024a; Huo et al., 2024), when input conditions are provided, the quality of SDS-generated outputs is inferior to that of DS-based methods. However, SDS performs better than DS when no conditions are given (except for the text prompt), effectively integrating images from the instance spaces without producing seams, although it struggles to generate fine details (Fig. 2(b)). In the following section, we introduce our novel method that combines the strengths of both approaches to achieve superior quality in unconditional canonical data point generation while maintaining performance in conditional generation.

6 StochSync: Stochastic Diffusion Synchronization

Based on our analysis comparing Diffusion Synchronization (DS) and Score Distillation Sampling (SDS) in Sec. 5, we propose our novel method, Stochastic Diffusion Synchronization, or `StochSync` for short, which combines the best features of each method to achieve superior performance in unconditional canonical sample generation. From the perspective of DS, we introduce three key changes in the algorithm.

Maximum Stochasticity in Synchronization. One of the key differences between SDS and previous DS methods is that SDS can be interpreted as utilizing maximum stochasticity in the DDIM denoising step (setting $\sigma_t = \sqrt{1-\alpha_{t-1}}$ in Eq. 5 and thus removing the $\boldsymbol{\epsilon}_t$ term), while earlier DS methods have not explored this aspect. We investigated whether maximum stochasticity helps DS achieve better coherence of samples across instance spaces, similar to what is observed in SDS. As the results shown in Fig. 2(c), it indeed helps remove seams, resulting in much smoother transitions across views. However, we also observe a trade-off between coherence and realism: increased stochasticity leads to greater deviation from the data distribution, producing less realistic images.

Multi-Step $\mathbf{x}_{0|t}$ Computation. To resolve the trade-off between coherence and realism, we propose replacing the computation of $\mathbf{x}_{0|t}$ from Tweedie's formula (Eq. 4), the one-step prediction of the clean sample $\mathbf{x}_0$ from $\mathbf{x}_t$, with a multi-step deterministic denoising process of DDIM, denoted as $\mathcal{G}(\mathbf{x}_t)$. We observe that a more accurate prediction of the clean samples $\mathbf{x}_{0|t}$ at each step along with maximum stochasticity level allows us to achieve both high coherence and realism as shown in Fig. 2(d). Notably, when replacing the computation of $\mathbf{x}_{0|t}$ with multi-step denoising, `StochSync` can also be viewed as iterating SDEdit (Meng et al., 2021): performing the forward process from

Table 1: Quantitative results of panorama generation using the prompts provided in PanFusion (Zhang et al. (2024a)). GIQA is scaled by 10^3. The best result in each column is highlighted in **bold**, and the runner-up is underlined.

Method	FID ↓	IS ↑	GIQA ↑	CLIP ↑
SDS	96.44	8.21	17.90	30.87
SDI	143.70	8.08	15.03	29.12
ISM	114.32	8.16	17.08	**31.31**
MVDiffusion	70.49	**10.87**	18.81	30.79
PanFusion	93.85	9.90	17.79	28.21
L-MAGIC	<u>59.83</u>	9.12	<u>19.13</u>	29.73
`StochSync`	**57.88**	<u>10.02</u>	**20.30**	<u>31.01</u>

Table 2: Effectiveness of each components using the prompts provided in PanFusion (Zhang et al. (2024a)). GIQA is scaled by 10^3. The best result in each column is highlighted in **bold**, and the runner-up is underlined.

Id	Max σ_t	Impr. $\mathbf{x}_{0\|t}$	N.O. Views	FID ↓	IS ↑	GIQA ↑	CLIP ↑
1	✗	✗	✗	80.55	<u>8.65</u>	18.22	30.07
2	✔	✗	✗	138.82	6.98	15.68	27.95
3	✗	✔	✗	84.87	7.33	<u>19.06</u>	<u>30.49</u>
4	✔	✔	✗	<u>78.56</u>	8.54	18.44	30.18
5	✔	✗	✔	117.09	7.56	16.32	28.75
6	✔	✔	✔	**57.88**	**10.02**	**20.30**	**31.01**

$\mathbf{x}_{0|t}$ to $\mathbf{x}_{t-1}$ at timestep t (Alg.4, line 10), followed by the reverse process back to $\mathbf{x}_{0|t-1}$ (line 11). As a result, the loop in line 7 can be interpreted not as performing the reverse process but as iterating SDEdit, meaning it does not need to proceed from timestep T to 1. Empirically, we find that stopping the iteration earlier with $T_{\text{stop}} \gg 1$ provides comparable results while saving computation time. More details are provided in **Appendix**.

Non-Overlapping View Sampling. In DS, $\mathbf{x}_{0|t}$ is not directly used in the next timestep; instead, it is first averaged in the canonical space (Alg. 2, line 15) and then projected back to the instance space (line 10). We note that this modification of $\mathbf{x}_{0|t}$ also results in a degradation of realism in the final output. To address this, we propose to sample views at each step *without* overlaps. $\mathbf{x}_{0|t}$ is still synchronized *over time*, as the set of non-overlapping views newly sampled at each step has overlaps with the views sampled in previous steps. In practice, we alternate between two sets of non-overlapping views—one being a shift of the other. The result further improved with the non-overlapping views is also shown in Fig. 2(f).

Pseudocode and Changes from DS. The pseudocode for our `StochSync`, incorporating the aforementioned three major changes from DS, is provided in Alg. 4. Compared to DS (Alg. 2), the ϵ_t computation is omitted due to the use of maximum stochasticity, Tweedie's formula is changed to a multi-step computation $\mathcal{G}(\cdot)$ (line 11), and the set of views is not fixed but is sampled without overlaps within the set at each step (line 8). In Alg. 4, the changes are highlighted in red.

Perspective from SDS. From the SDS perspective, `StochSync` can also be seen as implementing three major changes. First, each iteration is performed not with a random timestep t but with a linearly decreasing timestep (Alg. 4, line 8), following the scheduling of the reverse process. At each timestep, multiple views are selected and updated simultaneously. Second, instead of reflecting $\mathbf{x}_{0|t}$ to the canonical sample $\mathbf{z}$ through gradient descent, we fully minimize the $l2$ loss (line 13). Third, the computation of $\mathbf{x}_{0|t}$ is changed to a multi-step denoising (line 11). In other words, `StochSync` can be seen as a modification of SDS, designed to more closely resemble the reverse process with a decreasing time schedule, while ensuring tighter alignment between the instance space samples and the canonical space sample at each step.

Comparisons to SDS Variants. Recent variants of SDS have proposed changes to certain aspects of SDS, without observing connection to the synchronization framework, which we have explored for the first time to our knowledge. DreamTime (Huang et al., 2023) suggested decreasing the timestep instead of random sampling. We find that additionally replacing gradient descent with solving a minimization leads to significant improvements. SDI (Lukoianov et al., 2024) takes the opposite approach from ours, reducing the stochasticity of SDS to zero while requiring ϵ_t. Since ϵ_t cannot be maintained when views are randomly sampled, it is computed by performing DDIM inversion (Mokady et al., 2023) on $\mathbf{x}_{0|t}$ at every timestep. We empirically observe that this approach is not robust and frequently fails to converge for panorama and mesh texture generation, as shown in Fig. 2(e). ISM (Liang et al., 2024) also discusses the idea of solving an ODE for $\mathbf{x}_{0|t}$ (multi-step computation) at every timestep, but it does not change gradient descent to solving the minimization. In Section 7, we demonstrate the superior performance of `StochSync` compared to these methods in depth-free 360° panorama generation.

7 EXPERIMENT RESULTS

In this section, we present the experimental results of StochSync for two applications: 360° panorama generation and 3D mesh texturing. 360° panorama generation is an example of unconditional canonical data point generation (except for text conditioning), while 3D mesh texturing is an example of using depth maps as conditioning. We provide comparisons with baseline methods, user study results, as well as ablation study results. In the **Appendix**, we include implementation details (Sec. B), details of the user study (Sec. C), and additional qualitative and quantitative results (Sec. D).

7.1 360° PANORAMA GENERATION

In the 360° panorama generation, the projection operation f is equirectangular projection, which maps a 360° panoramic image to perspective view images. We specifically use 'Stable Diffusion 2.1 Base' as the pretrained diffusion model for all methods, except for the baselines that require finetuned models or inpainting models. We evaluate StochSync on sets of prompts provided by the previous works: 121 out-of-distribution prompts from PanFusion (Zhang et al., 2024a) and 20 ChatGPT-generated prompts from L-MAGIC (Cai et al., 2024). The results in the rest of this section are for PanFusion prompts, while the results for L-MAGIC prompts are provided in the **Appendix** (Sec. D). For evaluation, we randomly sample 10 perspective view images from each panorama and generate the same number of images using the pretrained diffusion model, which serves as the reference set for the evaluation metrics.

7.1.1 COMPARISON TO PREVIOUS WORKS

Quantitative and qualitative comparisons with the baseline methods using PanFusion (Zhang et al., 2024a) prompts are presented in Tab. 1 and Fig. 3, respectively. For quantitative evaluations, we report the Fréchet Inception Distance (FID) (Heusel et al., 2018), Inception Score (IS) (Salimans et al., 2016), and GIQA (Gu et al., 2020) to assess fidelity and diversity, as well as the CLIP score (Radford et al., 2021) to evaluate text alignment.

As shown in Tab. 1, StochSync outperforms SDS (Poole et al., 2023) and its variants, SDI (Lukoianov et al., 2024) and ISM (Liang et al., 2024), by significant margins in all metrics, except for the CLIP score, where ours is still close to the best. Notably, SDI and ISM are not robust and often generate poor outputs, as examples are shown on the left in rows 2-3 of Fig. 3 and more at the end of the **Appendix**.

We also compare StochSync with finetuning-based methods such as MVDiffusion (Tang et al., 2023b) and PanFusion (Zhang et al., 2024a), which finetune a pretrained image diffusion model using panoramic images. Due to the lack of large-scale datasets for panoramic images, these finetuning-based methods tend to overfit to the prompts and images used during training, reducing realism for unseen prompts. Hence, our zero-shot method outperforms these methods quantitatively across all metrics, with particularly large margins for FID, except for IS scores where the results are comparable. Qualitatively, our method also demonstrates superior performance compared to theirs, as shown in Fig.3 (rows 4–5, left). More examples can be found in at the end of the **Appendix**.

Lastly, we compare StochSync with the state-of-the-art zero-shot 360° panorama generation method, L-MAGIC (Cai et al., 2024), which uses an inpainting diffusion model to sequentially fill a panoramic images. Quantitatively, StochSync outperforms this method across all metrics. Qualitatively, we observe that L-MAGIC often exhibits a "wavy effect" (Brown & Lowe, 2007) causing the horizon to appear curved, as shown at the bottom left of Fig. 3. While this geometric distortion may not be fully captured in the quantitative metrics, it can significantly detract from the visual quality in terms of human perception. To further evaluate this, we conducted a user study comparing StochSync and L-MAGIC on both the PanFusion prompts and a new set of 20 prompts generated by ChatGPT, specifically including the word "horizon". StochSync was preferred over L-MAGIC by 56.20% for the former, with the preference increasing to 64.75% for the horizon-specific prompts, demonstrating the superior ability of StochSync to avoid producing curved horizons. Details of the user study are provided in the **Appendix** (Sec. C).

7.1.2 ABLATION STUDY RESULTS

Tab. 2 and Fig. 3 (right) demonstrate the effectiveness of each component of StochSync discussed in Sec. 6: maximum stochasticity (Max σ_t), multi-step denoising for $\mathbf{x}_{0|t}$ (Impr. $\mathbf{x}_{0|t}$), and non-overlapping view sampling (N.O. Views). As discussed in Sec. 5, DS, represented by SyncTweedies (Kim et al., 2024a), generates plausible local images but lacks global coherence across views and thus produce visible seams (row 1 of Fig. 3). With maximum stochasticity, global coherence improves but at the cost of realism (row 2 of Fig. 3), which is also reflected in the poor quantitative results (row 2 of Tab. 2). Noticeable improvements occur when the computation of $\mathbf{x}_{0|t}$ is also replaced with multi-step denoising, $\mathcal{G}(\mathbf{x}_t)$ (row 4 of Fig. 3 and Tab. 2). Finally, the full version of `StochSync`, using sets of non-overlapping views, produces the most realistic and coherent panoramic images both qualitatively and quantitatively (row 6 of Fig. 3 and Tab. 2). Refer to the other rows for additional ablation cases. Note that non-overlapping views require maximum stochasticity, as ϵ_t cannot be computed when views are not fixed but sampled differently every time.

Metric	Sync-Tweedies	Paint-it	Paint3D	TEXTure	Text2Tex	Sync-Stoch
FID ↓	**21.76**	28.23	31.66	34.98	26.10	<u>22.29</u>
KID ↓	<u>1.46</u>	2.30	5.69	6.83	2.51	**1.31**
CLIP ↑	**28.89**	28.55	28.04	<u>28.63</u>	27.94	28.57

Table 3: Quantitative results of 3D mesh texturing. KID is scaled by 10^3. The best result in each row is highlighted in **bold**, and the runner-up is <u>underlined</u>.

7.2 3D Mesh Texturing

3D mesh texturing is a task where a depth map from each view can be used as a condition for image generation, allowing the use of conditional diffusion models (e.g., ControlNet (Zhang et al., 2023)). While previous DS-based methods perform well when strong conditions are provided, we demonstrate that `StochSync`, designed to focus on the unconditional case, provides results comparable to previous DS methods and outperforms other state-of-the-art texture generation methods.

In our experiments, we follow the experiment setup of SyncTweedies (Kim et al., 2024a) while using the same 429 mesh and prompt pairs. The quantitative and qualitative results are presented in Tab. 3 and Fig. 4, respectively. Note that the results from other baseline methods are sourced from Kim et al. (2024a). In Tab. 3, `StochSync` outperforms all other baselines across all metrics, with the exception of SyncTweedies, our base synchronization framework, which shows comparable results. This demonstrates the versatility of our method, as it can be adapted to applications regardless of whether strong conditional inputs are present. In Fig. 4, `StochSync` generates texture images with fine details, as seen in the face of the bunny (column 1) and the wood grain patterns of the crate (column 2), whereas Paint-it(Youwang et al., 2023) leveraging SDS produces images that lack such details. Paint3D (Zeng et al., 2024), which finetunes a diffusion model on the textured mesh dataset (Deitke et al., 2023), fails to capture these details, as seen in the globe (column 4) and the pumpkin (column 6). This aligns with the observation made in the 360° panorama generation task, where finetuning on a small-scale dataset may result in the loss of rich priors learned by a pretrained diffusion model. Lastly, outpainting-based methods, TEXTure and Text2Tex (Richardson et al., 2023; Chen et al., 2023a), generate texture images with visible seams due to error accumulation, as shown in the goldfish (column 7) and the screen of the television (column 8).

Fig. 5 also showcases 3D mesh textures on spheres and tori generated by `StochSync` *without* depth conditioning, showing the potential for various visual content generation (e.g.,game maps).

8 Conclusion and Future Work

We have introduced `StochSync`, a novel zero-shot method for data generation in arbitrary spaces that fuses Diffusion Synchronization (DS) and Score Distillation Sampling (SDS) into the best form for achieving superior performance in cases where strong conditioning is not provided. Our key insights, based on analyses of the differences between DS and SDS, were to maximize stochasticity in the denoising process to achieve coherence across views, while enhancing realism through multi-step denoising for clean sample predictions at each step and sampling non-overlapping views. We demonstrated state-of-the-art performance in depth-free 360° panorama generation and depth-based mesh texture generation.

Limitation and Future Work. Synchronization methods, including ours, face challenges in 3D NeRF (Mildenhall et al., 2021) or Gaussian splat Kerbl et al. (2023) generation, as solving the $l2$-minimization at each step typically leads to overfitting to individual views when the intermediate images are inconsistent. This issue could be resolved by initializing the 3D geometry with 3D generative models (Hong et al., 2023; Tang et al., 2024), which we plan to explore in future work.

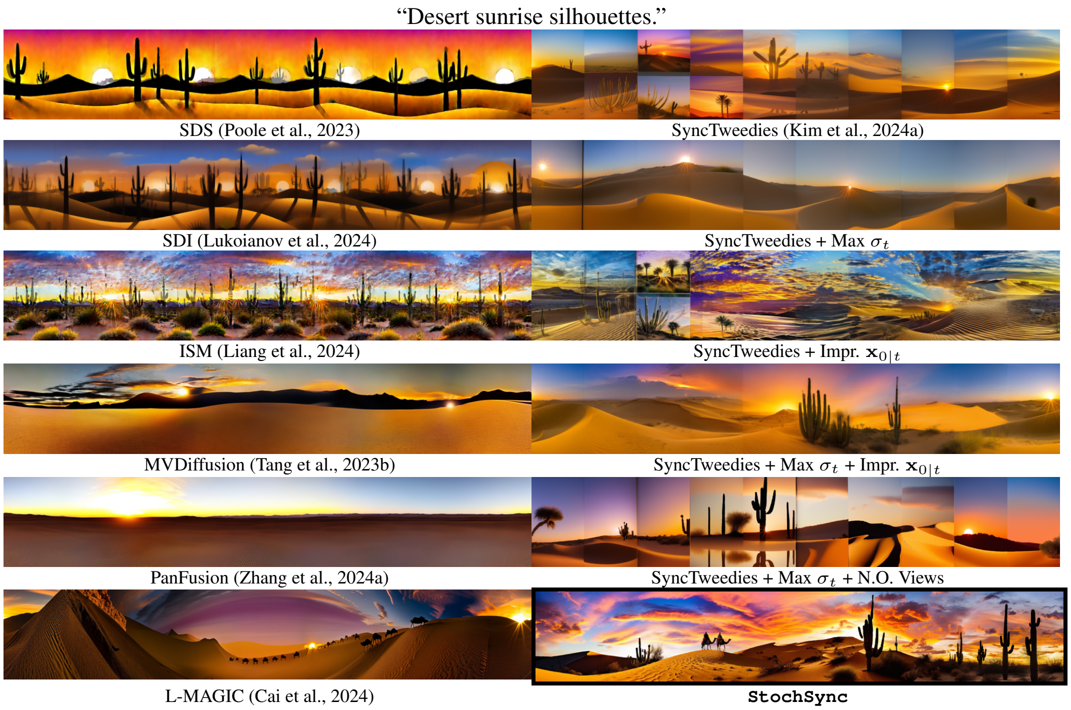


Figure 3: Qualitative results of panorama generation using PanFusion (Zhang et al., 2024a) prompts. Comparisons to previous works are presented in the left column, while the ablation cases are shown in the right column along with `StochSync`.

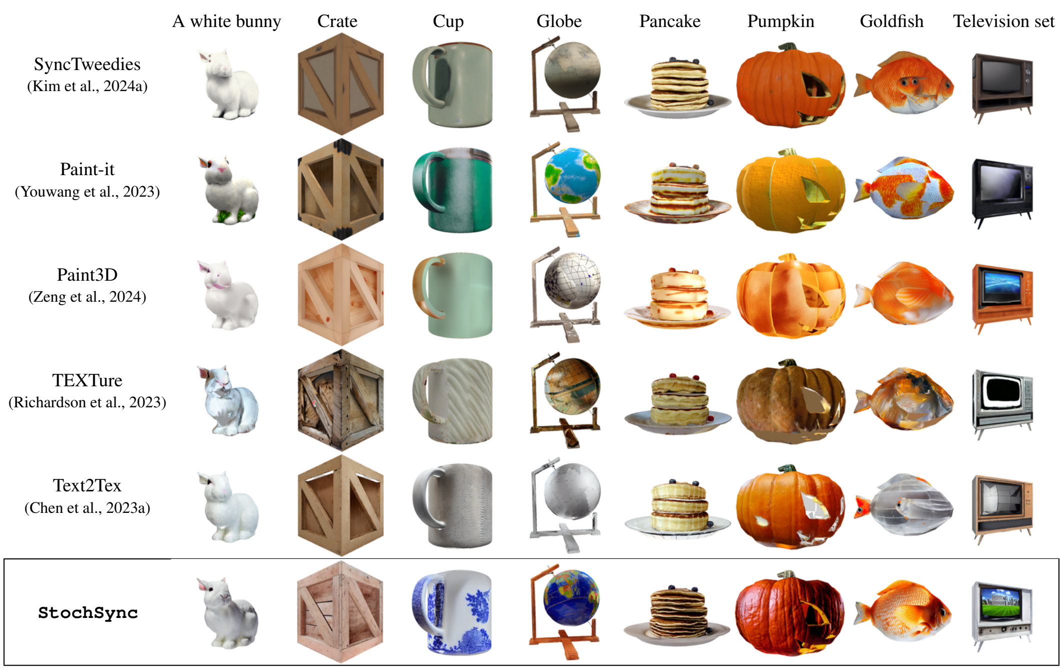


Figure 4: Qualitative result of 3D mesh texturing. `StochSync` generates realistic texture images, demonstrating its applicability even in the conditional generation case.

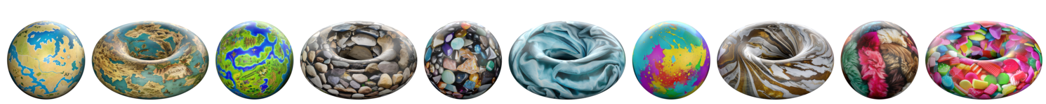


Figure 5: 3D mesh textures on spheres and tori generated by `StochSync`.

Ethics Statement

`StochSync` leverages a diffusion model (Rombach et al., 2022) trained on the LAION-5B dataset (Schuhmann et al., 2022), which has been preprocessed to remove unethical content. However, despite these efforts, the pretrained diffusion model may still generate undesirable content when presented with misleading or harmful prompts, a limitation that our method also inherits. It is important to acknowledge this risk, as models like `StochSync` could inadvertently produce biased or inappropriate outputs and should be used with caution. Additionally, `StochSync` may impact the creative industry by automating parts of the generative process. However, it also offers opportunities to enhance productivity and accessibility to generative tools.

Reproducibility Statement

`StochSync` uses the 'Stable Diffusion 2.1 Base' (Rombach et al., 2022) and the depth-conditioned ControlNet (Zhang et al., 2023), both of which are publicly available. We also provide the pseudocode of `StochSync` in Alg. 4 and the implementation details including hyperparameters in Sec. B. We will also release our code publicly.

References

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023.

Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 2007.

Zhipeng Cai, Matthias Mueller, Reiner Birkl, Diana Wofk, Shao-Yen Tseng, JunDa Cheng, Gabriela Ben-Melech Stan, Vasudev Lai, and Michael Paulitsch. L-magic: Language model assisted generation of images with coherence. In *CVPR*, 2024.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *CVPR*, 2023.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *ICCV*, 2023a.

Hansheng Chen, Ruoxi Shi, Yulin Liu, Bokui Shen, Jiayuan Gu, Gordon Wetzstein, Hao Su, and Leonidas Guibas. Generic 3d diffusion adapter using controlled multi-view editing. In *SIGGRAPH*, 2024a.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 2018.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22246–22256, 2023b.

Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2Light: Zero-shot text-driven hdr panorama generation. *ACM TOG*, 2022.

Ziyang Chen, Daniel Geng, and Andrew Owens. Images that sound: Composing images and sounds on a single canvas. *arXiv preprint arXiv:2405.12221*, 2024b.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Daniel Geng, Inbum Park, and Andrew Owens. Factorized diffusion: Perceptual illusions by noise decomposition. In *ECCV*, 2024a.

Daniel Geng, Inbum Park, and Andrew Owens. Visual anagrams: Generating multi-view optical illusions with diffusion models. In *CVPR*, 2024b.

Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Giqa: Generated image quality assessment. In *ECCV*, 2020.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *ICCV*, 2023.

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 *NeurIPS*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. *ICLR*, 2023.

Yukun Huang, Jianan Wang, Yukai Shi, Boshi Tang, Xianbiao Qi, and Lei Zhang. Dreamtime: An improved optimization strategy for diffusion-guided 3d generation. In *ICLR*, 2023.

Dong Huo, Zixin Guo, Xinxin Zuo, Zhihao Shi, Juwei Lu, Peng Dai, Songcen Xu, Li Cheng, and Yee-Hong Yang. Texgen: Text-guided 3d texture generation with multi-view sampling and resampling. In *ECCV*, 2024.

Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv preprint arXiv:2310.17590*, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.

Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. Synctweedies: A general generative framework based on synchronized diffusions. *arXiv preprint arXiv:2403.14370*, 2024a.

Jeongsol Kim, Geon Yeong Park, and Jong Chul Ye. Dreamsampler: Unifying diffusion sampling and score distillation for image manipulation. In *ECCV*, 2024b.

Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, 2024.

Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *NeurIPS*, 2023.

Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *CVPR*, 2024.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022.

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pp. 11461–11471, 2022.

Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Campagnolo Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin Solomon. Score distillation via reparametrized ddim. *arXiv preprint arXiv:2405.15891*, 2024.

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*, 2021.

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-NeRF for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12663–12673, 2023.

Midjourney. Midjourney. `https://www.midjourney.com/`.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2021.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023.

Jangho Park, Gihyun Kwon, and Jong Chul Ye. ED-NeRF: Efficient text-guided editing of 3D scene using latent space NeRF. In *ICLR*, 2023.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *ACM TOG*, 2023.

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*. Springer, 1956.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 2016.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2023a.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In *NeurIPS*, 2023b.

Xiaojuan Wang, Janne Kontkanen, Brian Curless, Steven M Seitz, Ira Kemelmacher-Shlizerman, Ben Mildenhall, Pratul Srinivasan, Dor Verbin, and Aleksander Holynski. Generative powers of ten. In *CVPR*, 2024a.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2024b.

Seungwoo Yoo, Kunho Kim, Vladimir G Kim, and Minhyuk Sung. As-Plausible-As-Possible: Plausibility-Aware Mesh Deformation Using 2D Diffusion Priors. In *CVPR*, 2024.

Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. *arXiv preprint arXiv:2312.11360*, 2023.

Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, BIN FU, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *CVPR*, 2024.

Cheng Zhang, Qianyi Wu, Camilo Cruz Gambardella, Xiaoshui Huang, Dinh Phung, Wanli Ouyang, and Jianfei Cai. Taming stable diffusion for text to 360○ panorama image generation. In *CVPR*, 2024a.

Hongkun Zhang, Zherong Pan, Congyi Zhang, Lifeng Zhu, and Xifeng Gao. Texpainter: Generative mesh texturing with multi-view consistency. In *SIGGRAPH*, 2024b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023.

Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. In *ICLR*, 2023.

Appendix

A Reformulation of SDS Loss

Here, we show that the SDS loss introduced in Sec. 5.2 of the main paper is equivalent to the original loss presented in DreamFusion (Poole et al., 2023) up to a scale. In Sec. 5.2, the SDS loss is presented from the perspective of clean samples:

$$\left\| f_{\mathbf{c}}(\mathbf{z}) - \mathbf{x}_{0|t-1} \right\|^2 = \left\| \frac{\mathbf{x}_{t-1} - \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}}{\sqrt{\alpha_{t-1}}} - \frac{\mathbf{x}_{t-1} - \sqrt{1-\alpha_{t-1}}\epsilon_\theta(\mathbf{x}_{t-1}, y)}{\sqrt{\alpha_{t-1}}} \right\|^2 \tag{7}$$

$$= \frac{1-\alpha_{t-1}}{\alpha_{t-1}} \left\| \boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{x}_{t-1}, y) \right\|^2, \tag{8}$$

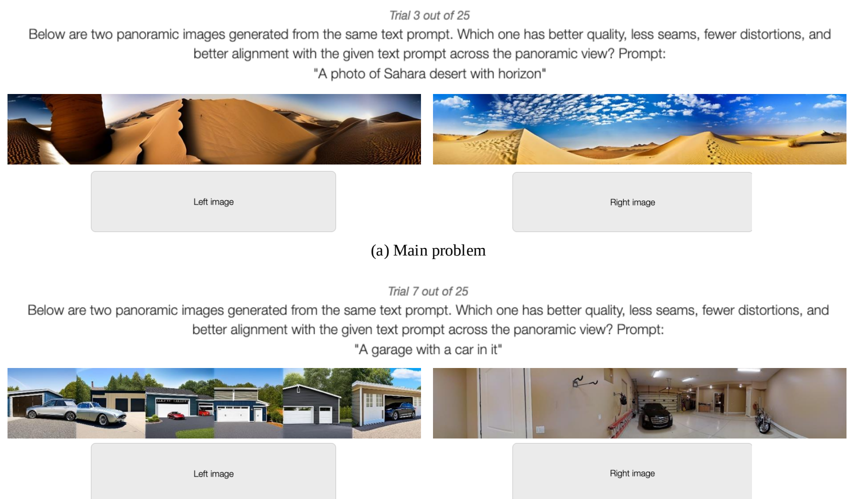
where the equality in the first line holds from Eq. 4 and $\boldsymbol{\epsilon}$ is sampled from a standard Gaussian, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Previous works (Kim et al., 2024b; Lukoianov et al., 2024) have also made a similar observation.

B Implementation Details

Panorama Generation. We set the resolution of the perspective view images to 512×512, and the panorama to $2,048 \times 4,096$. A linearly decreasing timestep schedule is employed, starting from $T = 900$ and decreasing to $T_{\text{stop}} = 270$, with a total of 25 denoising steps. For multi-step $\mathbf{x}_{0|t}$ computation, the total number of steps is initially set to 50, decreasing linearly as the denoising process progresses. For view sampling, we alternate between two sets containing five views each, with azimuth angles of $[0°, 72°, 144°, 216°, 288°]$ and $[36°, 108°, 180°, 252°, 324°]$. The elevation angle is set to $0°$, and the field of view (FoV) is set to $72°$.

For methods utilizing multi-step $\mathbf{x}_{0|t}$ predictions, computing $\mathbf{x}_{0|t-1} = \mathcal{G}(\mathbf{x}_{t-1})$ as in line 11 of Alg. 4, only for the last two steps in the loop of line 7, we leverage the previous $\mathbf{x}_{0|t}$ to better preserve the boundary regions. We perform the denoising process while blending the noisy data point as foreground and the previous $\mathbf{x}_{0|t}$ as background, as done in RePaint (Lugmayr et al., 2022). For the background mask, we start from the entire region and gradually decrease the regions over time to be close to the boundaries.

3D Mesh Texturing. For 3D mesh texturing, we follow the approach in SyncTweedies (Kim et al., 2024a) and use the same image and texture resolutions. We use the same number of steps as in the 360° panorama generation task with a linearly decreasing time schedule from $T = 1,000$ to $T_{\text{stop}} = 270$. We use 4 views to minimize overlaps between the views. For multi-step $\mathbf{x}_{0|t}$ predictions, we use the same refinement mentioned above.



(a) Main problem

(b) Vigilance test

Figure 6: Screenshots of the user study. The main test is shown in (a), and the vigilance test in (b).

C User Study Details

In this section, we provide details of the user study described in Sec. 7.1.1 of the main paper. We evaluated user preferences across two prompt sets: PanFusion (Zhang et al., 2024a) prompts and horizon-specific prompts. The study was conducted via Amazon Mechanical Turk (AMT).

Screenshots of the user study are shown in Fig. 6. Participants were presented with two panoramic images (in random order) generated using the same text prompt: one by L-MAGIC(Cai et al., 2024) and the other by `StochSync`. They were asked to answer the following question: "Which image has better quality, fewer seams, fewer distortions, and better alignment with the given text prompt across the panoramic view?" In each user study, 25 panoramic images were shown in a shuffled order, including five vigilance tests. For the vigilance tests, participants were shown a wide image composed of concatenated 2D square images alongside a ground truth 360° panorama, with the same resolution and question format. For the final results, we collected responses from 50 out of 96 participants from the PanFusion set and 59 out of 100 participants from the horizon set, passing at least three vigilance tests. We required participants to be AMT Masters and have an approval rate of over 95%.

Figure 7: Qualitative comparisons between L-MAGIC (Cai et al., 2024) and `StochSync` on the horizon-specific prompts.

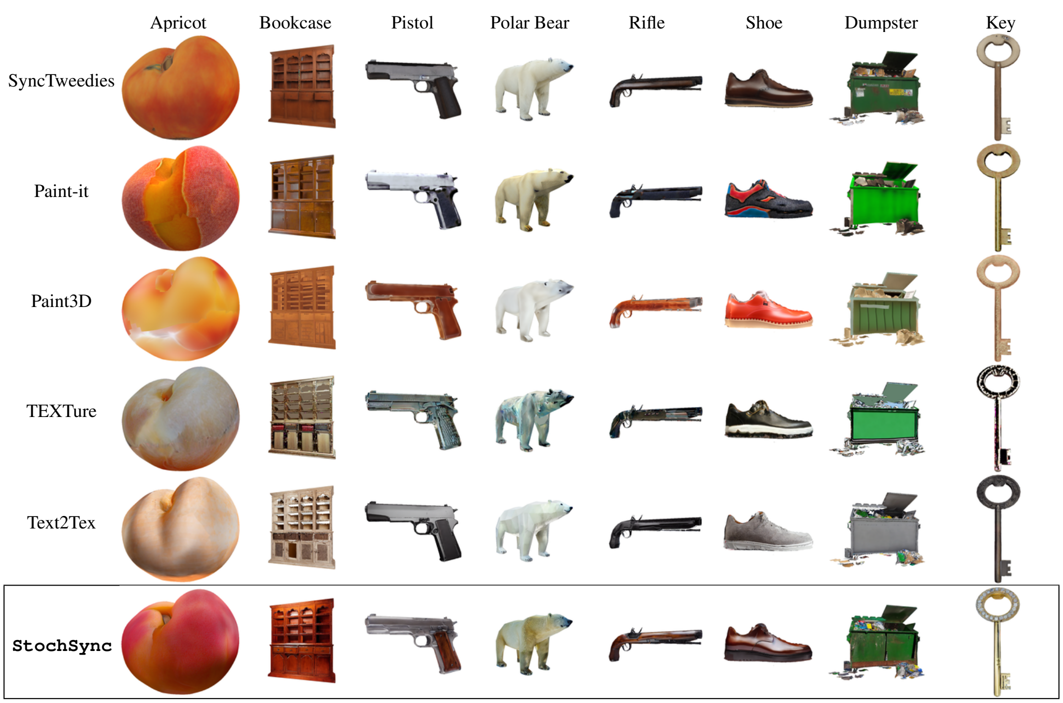


Figure 8: Additional qualitative results of 3D mesh texturing.

Table 4: Quantitative results of panorama generation using the prompts provided in L-MAGIC (Cai et al. (2024)). GIQA is scaled by 10^3. The best result in each column is highlighted in **bold**, and the runner-up is underlined.

Method	FID ↓	IS ↑	GIQA ↑	CLIP ↑
SDS	163.23	5.60	17.41	30.37
SDI	171.69	5.93	16.42	29.33
ISM	197.10	4.92	16.52	29.44
MVDiffusion	<u>111.12</u>	<u>6.17</u>	**20.71**	<u>31.07</u>
PanFusion	151.60	5.48	18.19	28.46
L-MAGIC	112.72	5.94	19.73	30.39
`StochSync`	**109.41**	**6.20**	<u>20.31</u>	**31.22**

Table 5: Effectiveness of each components using the prompts provided in L-MAGIC (Cai et al. (2024)). GIQA is scaled by 10^3. The best result in each column is highlighted in **bold**, and the runner-up is underlined.

Id	Max σ_t	Impr. $\mathbf{x}_{0\|t}$	N.O. Views	FID ↓	IS ↑	GIQA ↑	CLIP ↑
1	✗	✗	✗	<u>120.19</u>	<u>5.58</u>	<u>19.68</u>	29.34
2	✔	✗	✗	178.03	4.76	17.43	28.02
3	✗	✔	✗	139.34	4.83	18.94	<u>30.08</u>
4	✔	✔	✗	126.58	5.41	19.34	30.04
5	✔	✗	✔	169.32	4.74	16.67	28.53
6	✔	✔	✔	**109.41**	**6.20**	**20.31**	**31.22**

D ADDITIONAL RESULTS

Quantitative Results of $360°$ Panorama Generation Using L-MAGIC Prompts. The quantitative results of panorama generation using the prompts from L-MAGIC (Cai et al., 2024), as well as the ablation study results, are presented in Tab. 4 and Tab. 5, respectively. We observe the same trend as discussed in Sec. 7.1, where the results with PanFusion (Zhang et al., 2024a) prompts are discussed. `StochSync` generates high-fidelity panoramic images, while L-MAGIC tends to produce panoramas with curved horizons. Refer to Sec. D.2 for qualitative results.

Additional Results of $360°$ Panorama Generation Using Horizon Prompts. Qualitative comparisons of `StochSync` and L-MAGIC (Cai et al., 2024) on the horizon-specific prompt set discussed in Sec. 7.1.1 are shown in Fig. 7. As discussed in Sec. 7.1.1, L-MAGIC tends to generate wavy panoramas with global distortions, while `StochSync` produces more realistic panoramic images. This aligns with the results of the user preference test presented in Sec. 7.1.1, where `StochSync` outperforms L-MAGIC on both the PanFusion and horizon-specific prompts.

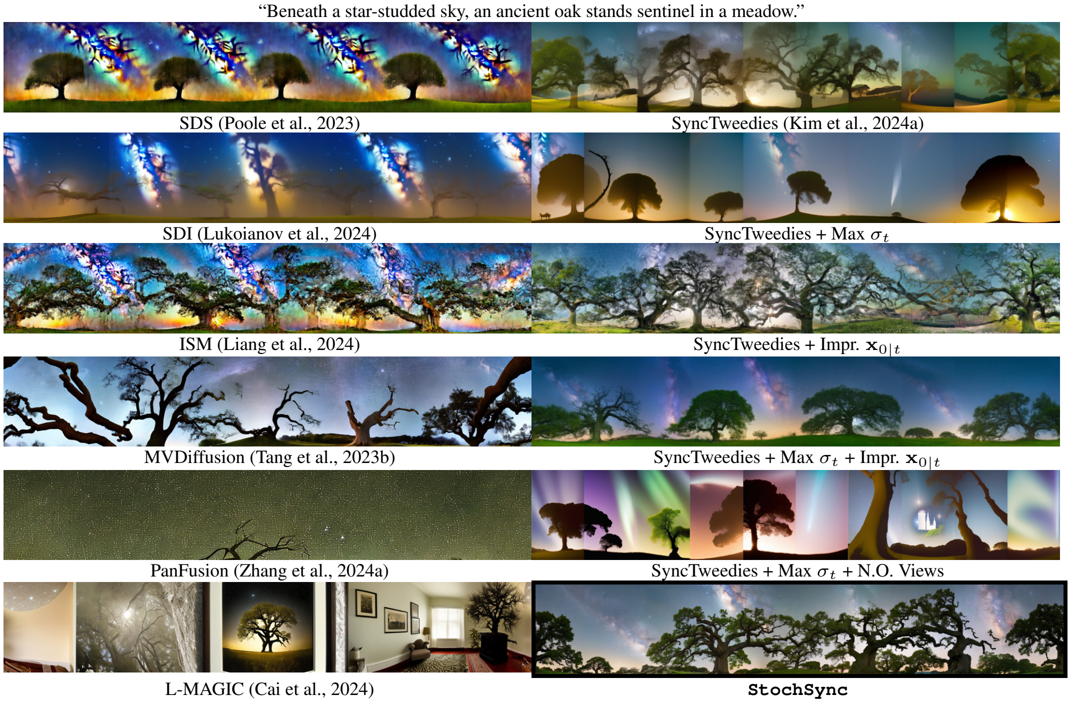
Additional Results of 3D Mesh Texturing. Extending the qualitative results presented in Fig. 4, we provide more qualitative results of 3D mesh texturing in Fig. 8.

More qualitative results of $360°$ panorama generation are presented in the following pages.

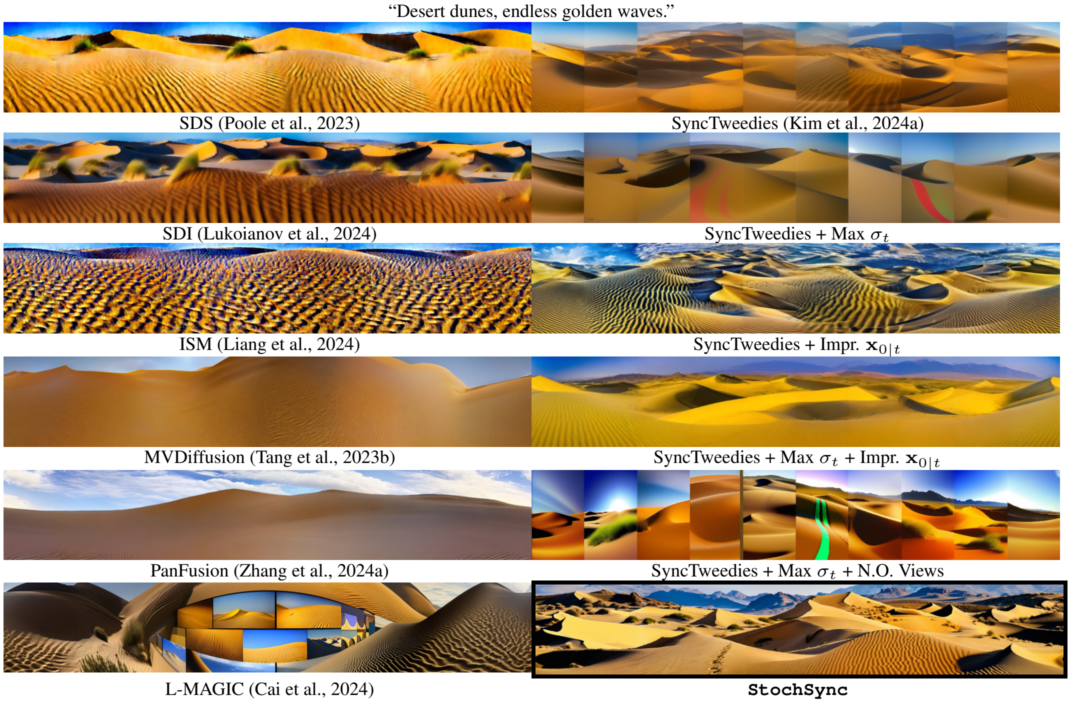
D.1 Additional 360° Panorama Generation Results Using PanFusion Prompts

"Desert canyon, sculpted sandstone."

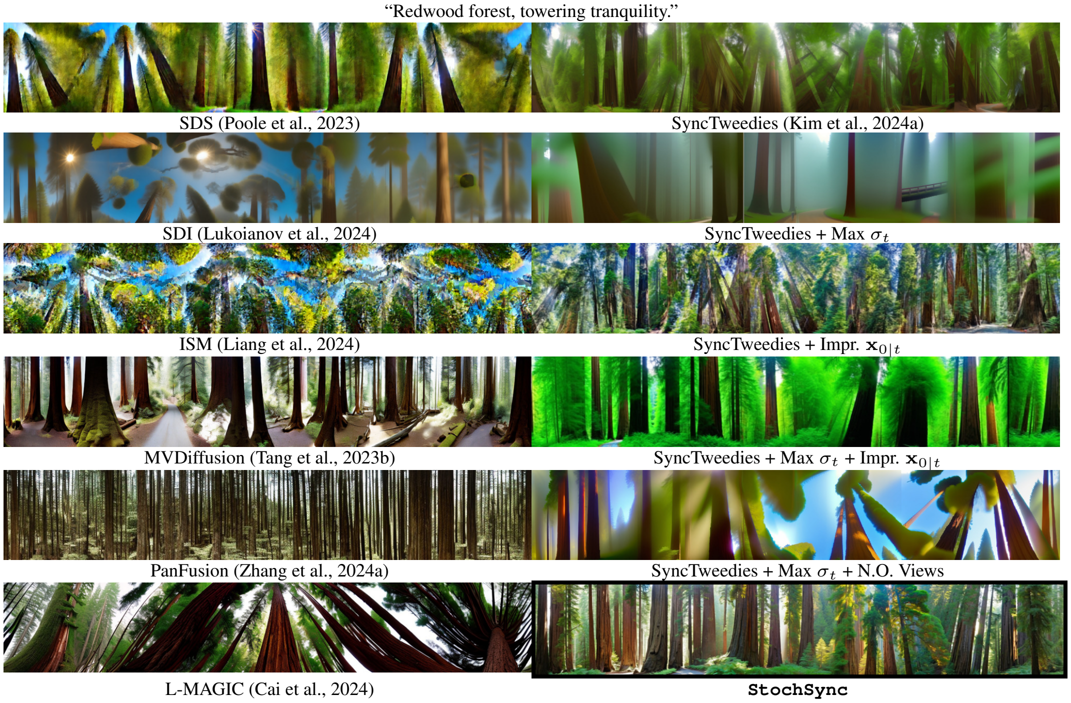
"Beneath a star-studded sky, an ancient oak stands sentinel in a meadow."



"Desert dunes, endless golden waves."



"Redwood forest, towering tranquility."

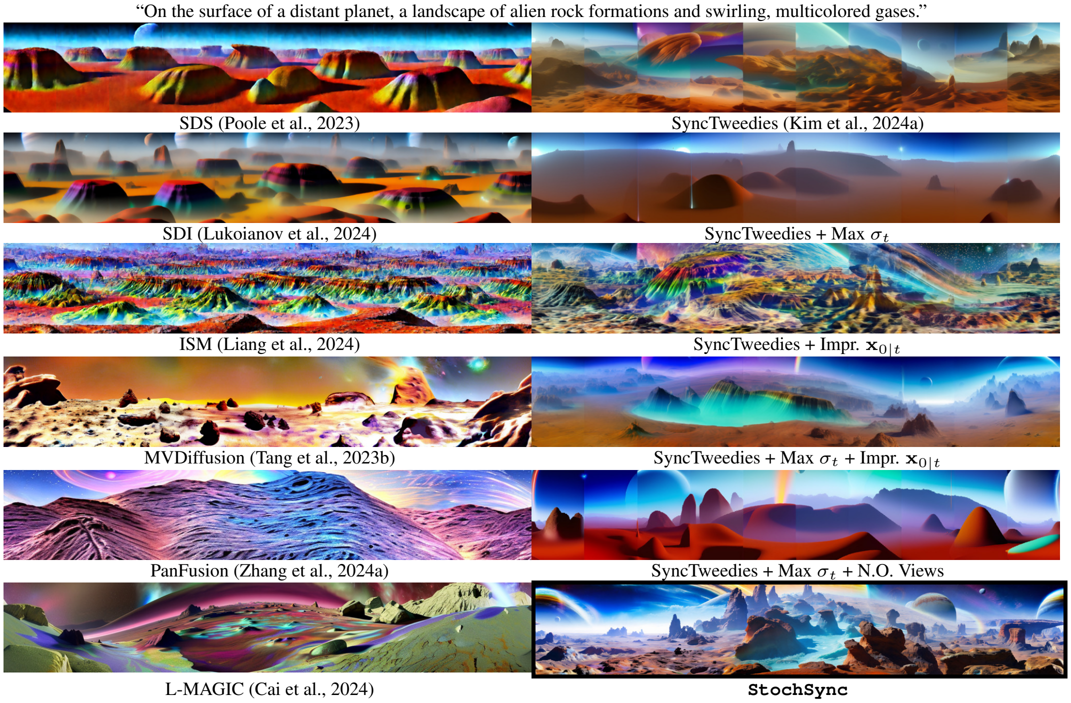


"Moonlit beach, waves whispering secrets."

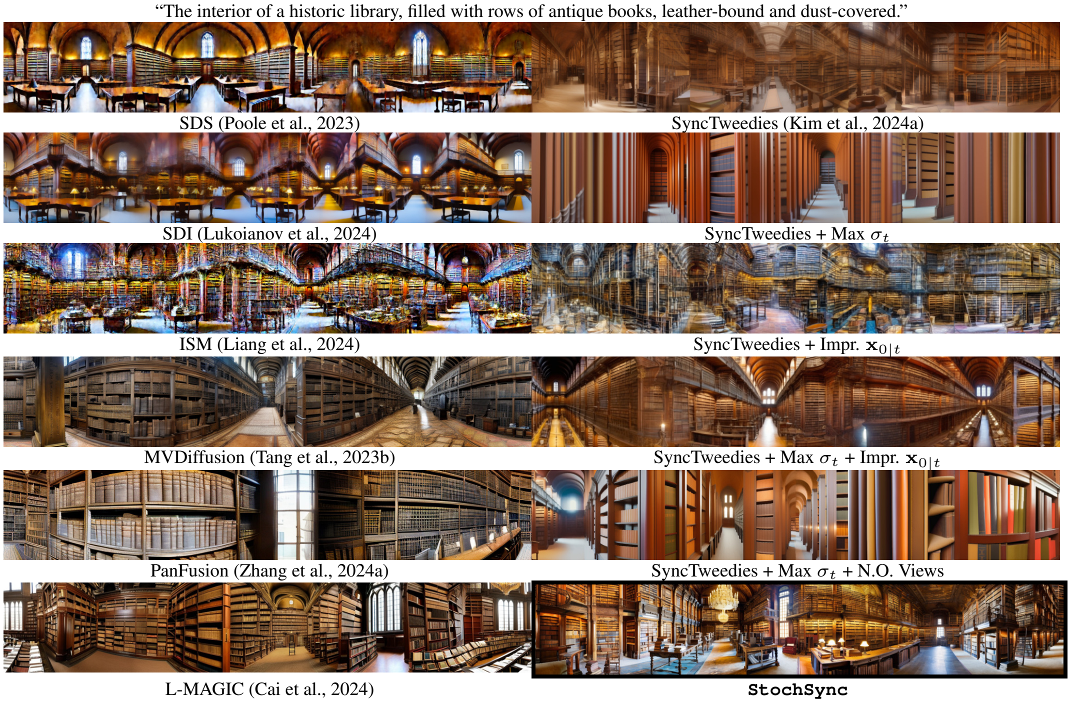
"On a distant planet surface, towering crystalline structures rise against an alien sky."

"Nestled in a canyon, a pueblo village stands against the red earth."

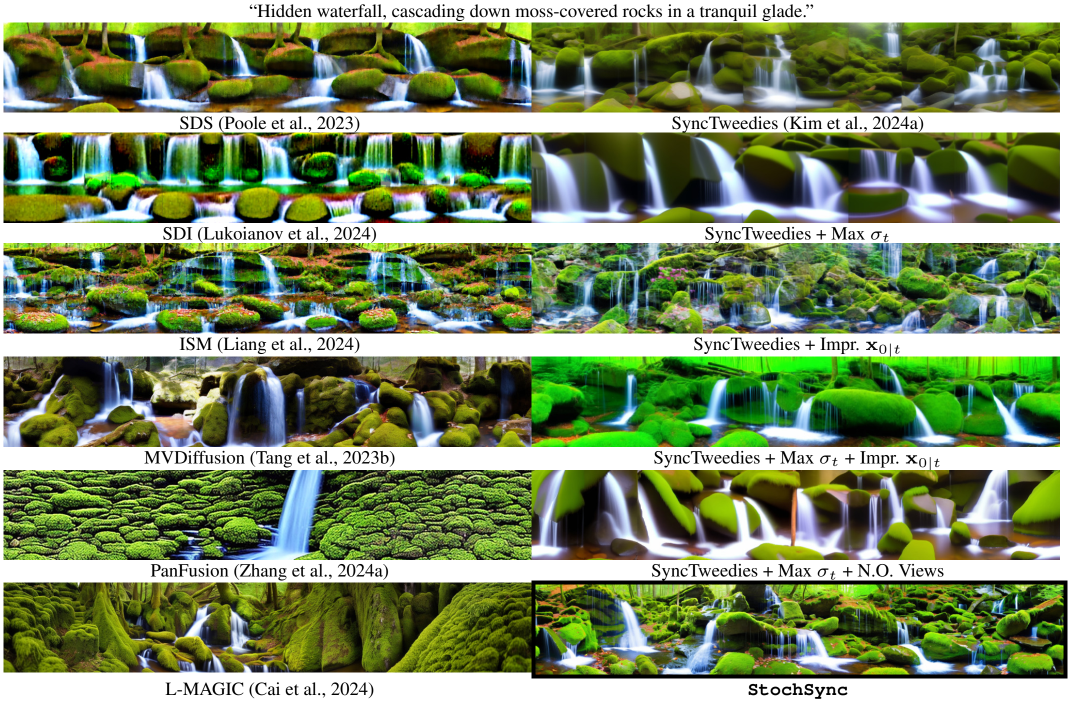
"On the surface of a distant planet, a landscape of alien rock formations and swirling, multicolored gases."



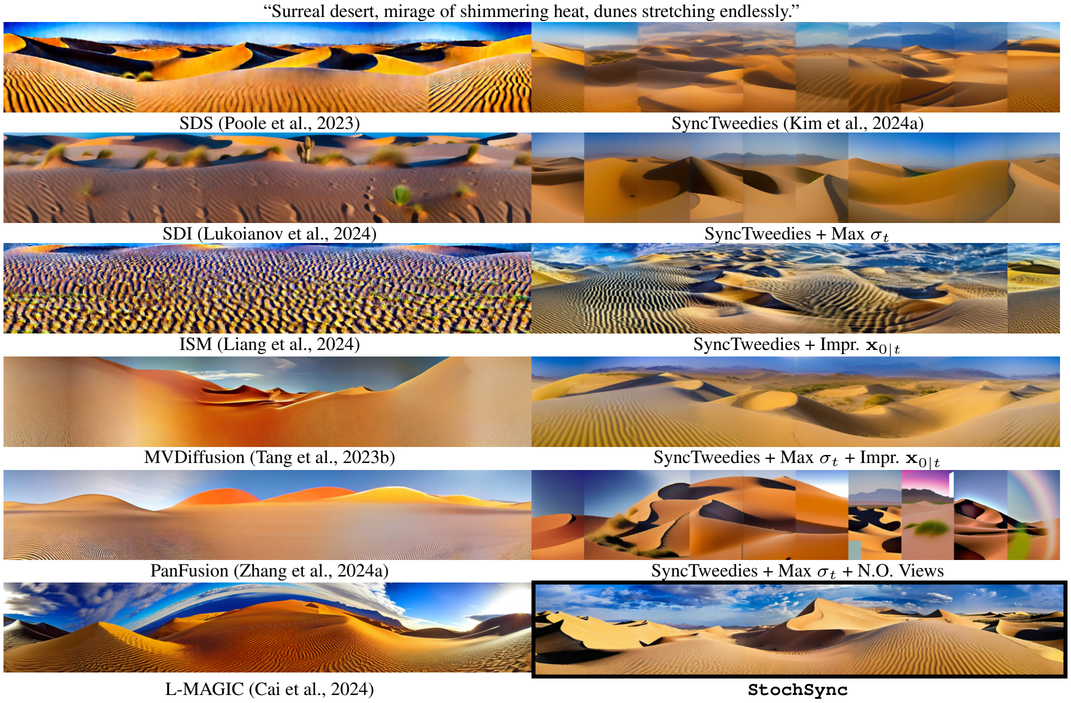
"The interior of a historic library, filled with rows of antique books, leather-bound and dust-covered."



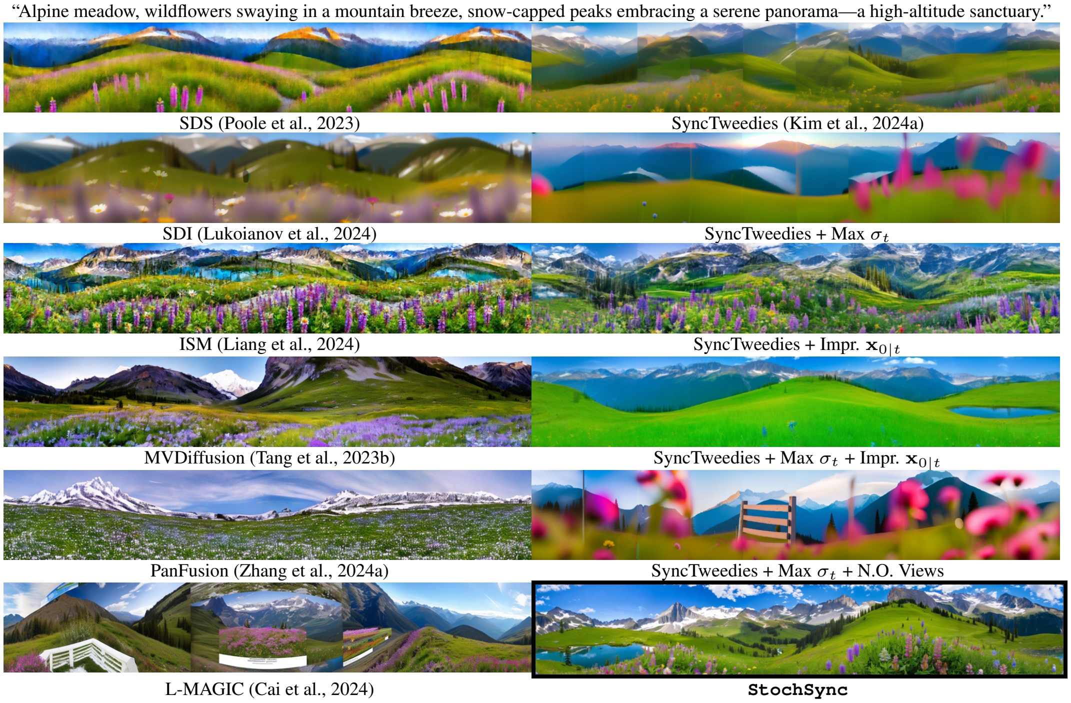
"Hidden waterfall, cascading down moss-covered rocks in a tranquil glade."



“Surreal desert, mirage of shimmering heat, dunes stretching endlessly.”

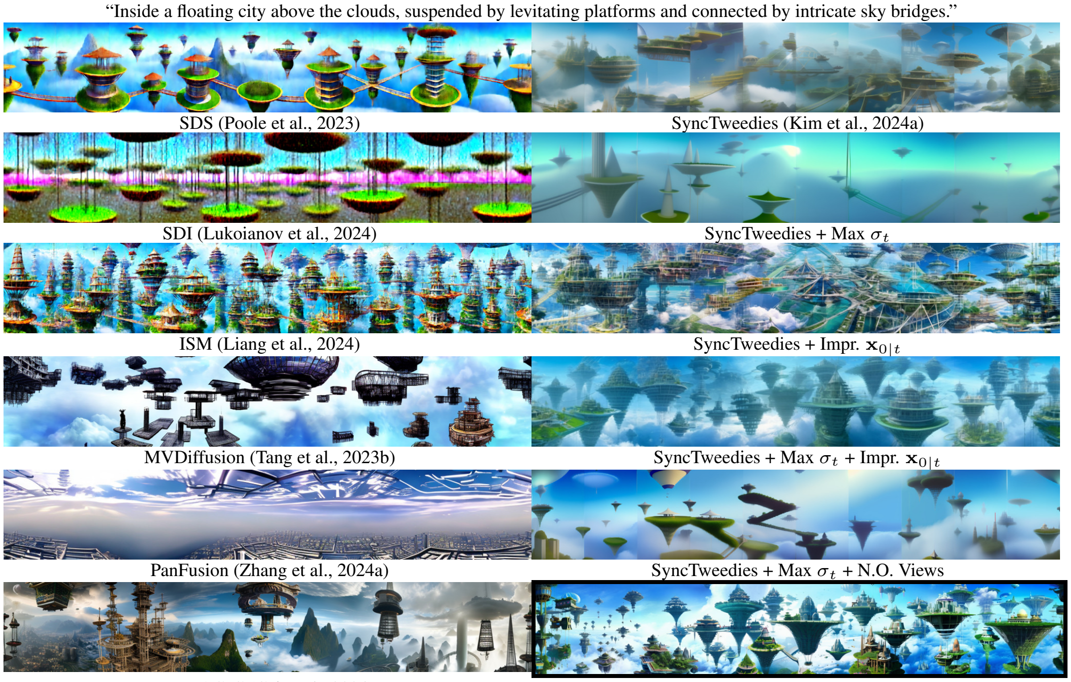


“Alpine meadow, wildflowers swaying in a mountain breeze, snow-capped peaks embracing a serene panorama—a high-altitude sanctuary.”



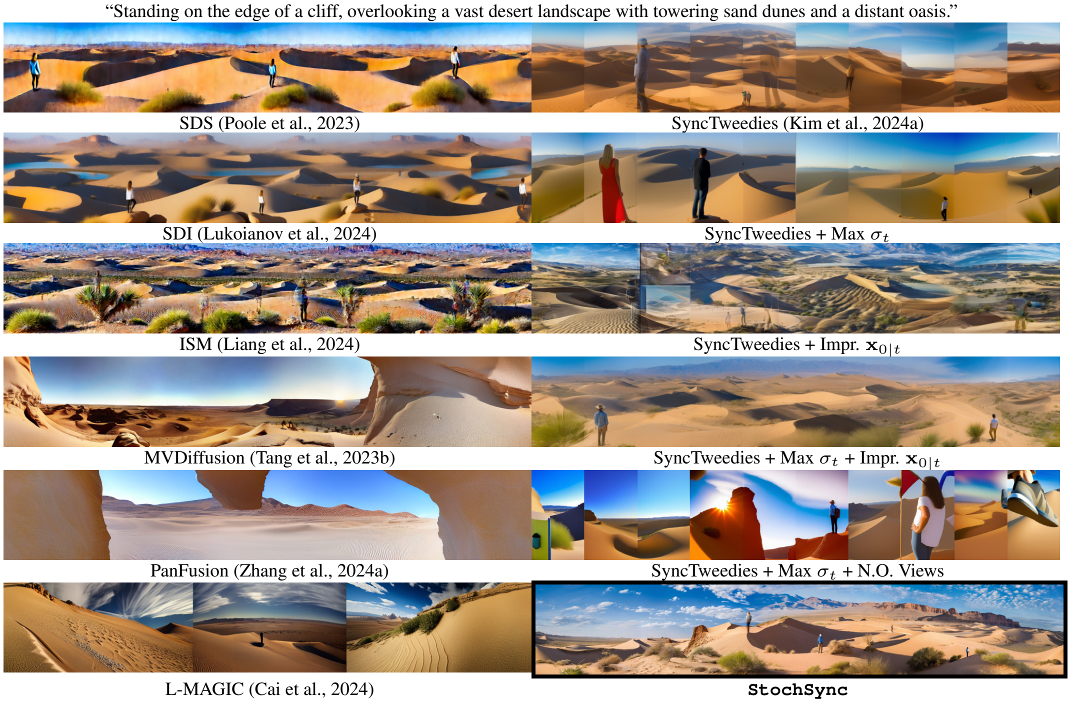
"Alpine village, snow-covered rooftops, nestled between majestic peaks—a picture-perfect scene of winter tranquility."

"Inside a floating city above the clouds, suspended by levitating platforms and connected by intricate sky bridges."



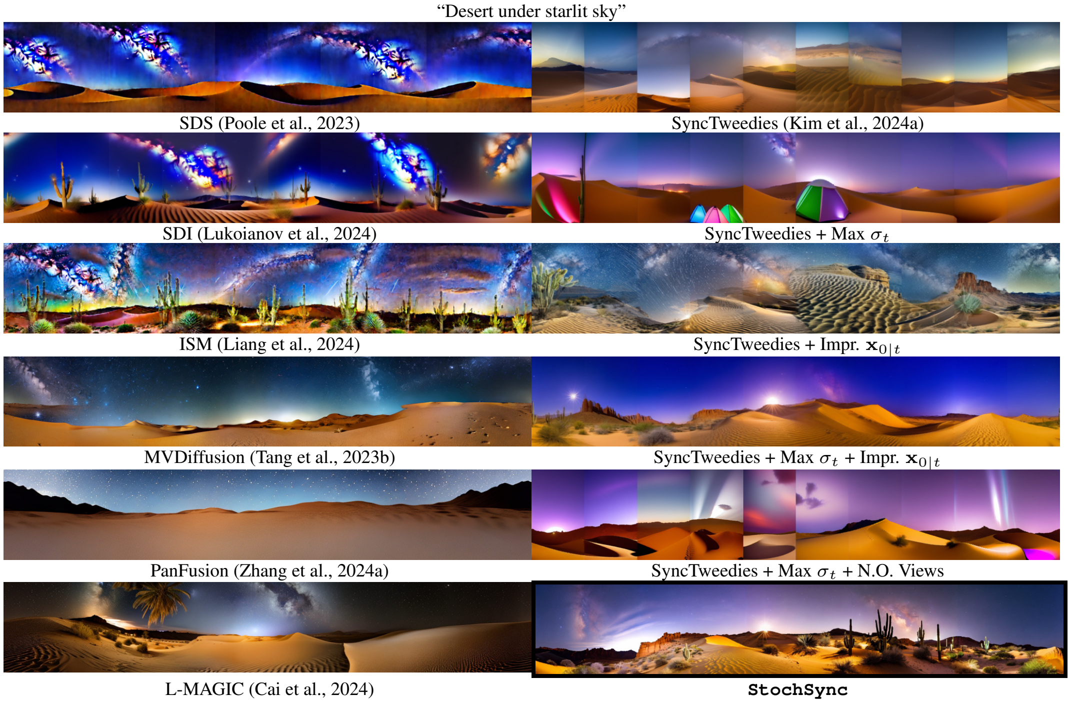
"Desert canyon, ancient rock formations sculpted by time, a vast expanse of terracotta hues—an arid symphony of textures."

"Standing on the edge of a cliff, overlooking a vast desert landscape with towering sand dunes and a distant oasis."



D.2 More 360° Panorama Generation Results Using L-MAGIC Prompts

"Desert under starlit sky"



"Snowy mountain peak view"

"Japanese Zen meditation room"

SDS (Poole et al., 2023) | SyncTweedies (Kim et al., 2024a)
SDI (Lukoianov et al., 2024) | SyncTweedies + Max σ_t
ISM (Liang et al., 2024) | SyncTweedies + Impr. $\mathbf{x}_{0|t}$
MVDiffusion (Tang et al., 2023b) | SyncTweedies + Max σ_t + Impr. $\mathbf{x}_{0|t}$
PanFusion (Zhang et al., 2024a) | SyncTweedies + Max σ_t + N.O. Views
L-MAGIC (Cai et al., 2024) | **StochSync**

"Sakura blossom park Kyoto"

SDS (Poole et al., 2023) | SyncTweedies (Kim et al., 2024a)
SDI (Lukoianov et al., 2024) | SyncTweedies + Max σ_t
ISM (Liang et al., 2024) | SyncTweedies + Impr. $\mathbf{x}_{0|t}$
MVDiffusion (Tang et al., 2023b) | SyncTweedies + Max σ_t + Impr. $\mathbf{x}_{0|t}$
PanFusion (Zhang et al., 2024a) | SyncTweedies + Max σ_t + N.O. Views
L-MAGIC (Cai et al., 2024) | **StochSync**

E ADDITIONAL QUALITATIVE RESULTS

In this section, we provide qualitative results of additional applications of `StochSync` including image inpainting (Fig. 9-10 and Fig. 11), high resolution panorama generation (Fig. 13), 3D mesh texturing with PBR materials (Fig. 14), panorama generation using a pose-conditioned video diffusion model (He et al., 2024) (Fig. 15 and Fig. 16), and texturing 3D Gaussians (Kerbl et al., 2023) (Fig. 17). In Fig. 12, we present qualitative results of image generation using Max. σ_t over multiple iterations.

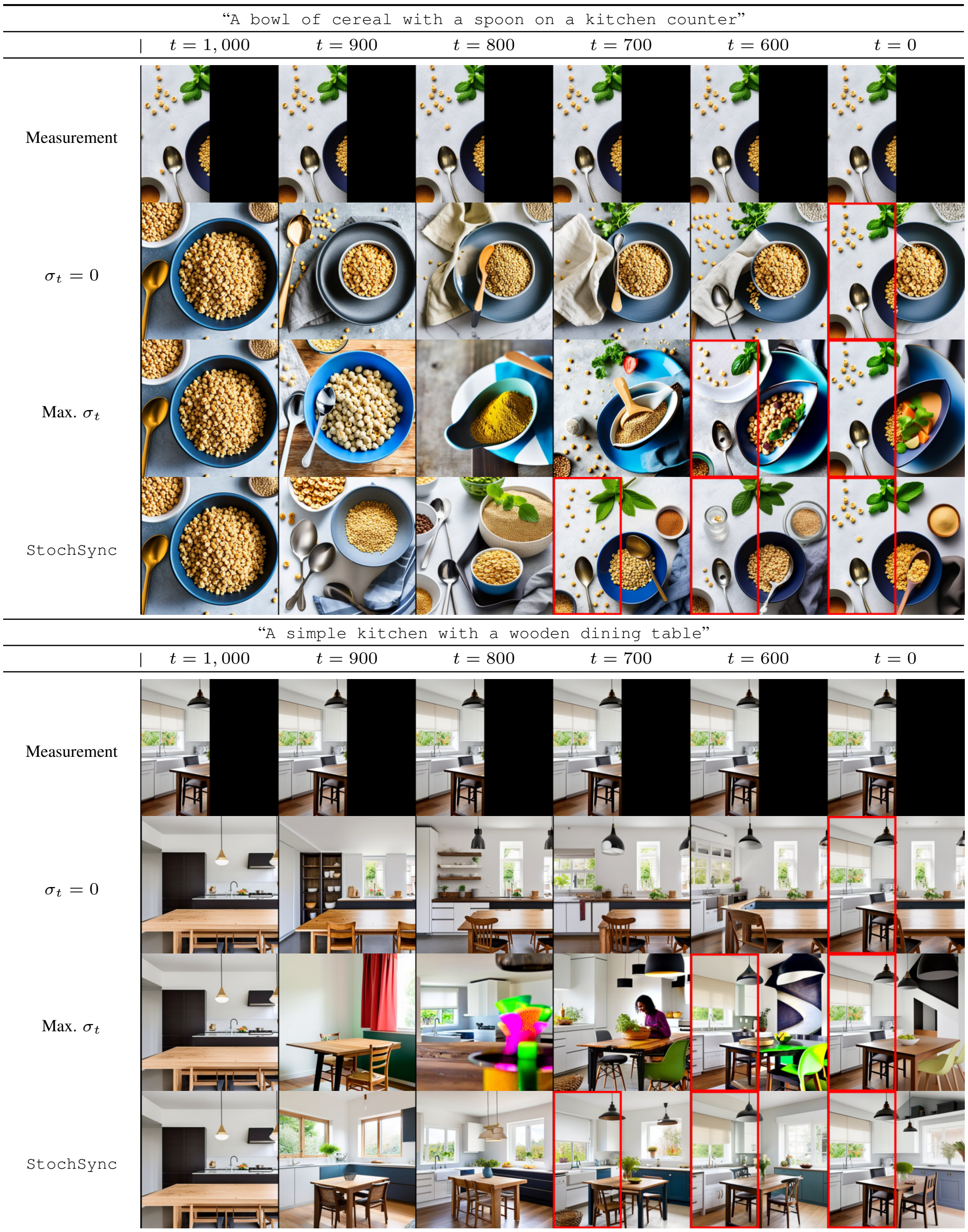


Figure 9: Qualitative result of image inpainting.

Figure 10: Qualitative results of image inpainting.

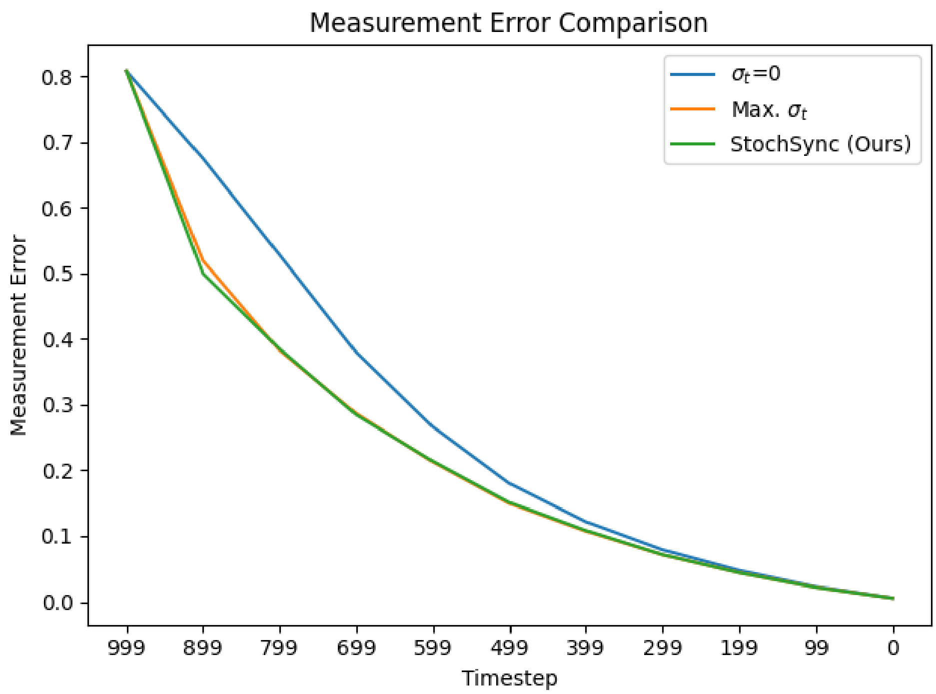


Figure 11: Measurement error plotted against denosing process timesteps. The measurement error for the case of $\sigma_t = 0$ remains larger than the cases utilizing the maximum level of stochasticity (Max. σ_t and `StochSync`).

"`A DSLR photo of a dog`"			
Number of Steps $= 10$	Number of Steps $= 100$	Number of Steps $= 1,000$	Number of Steps $= 10,000$

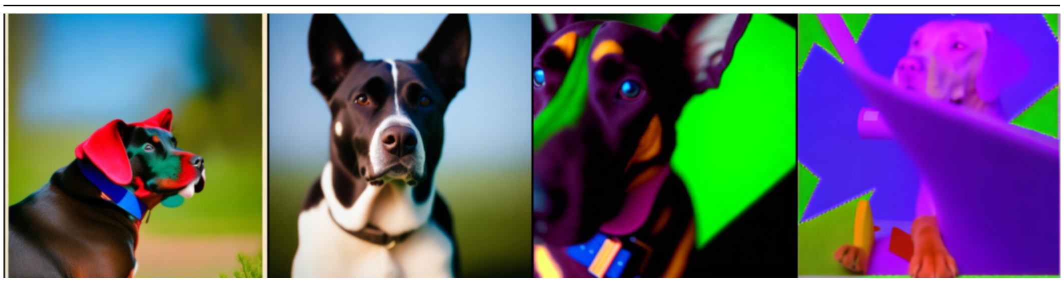


Figure 12: Qualitative results of image generation with Max. σ_t. Each image is obtained by running different number of steps. Sampling images with Max. σ_t for a large number of steps fails to generate plausible images.

Figure 13: Qualitative results of high resolution panorama generation using `StochSync`.

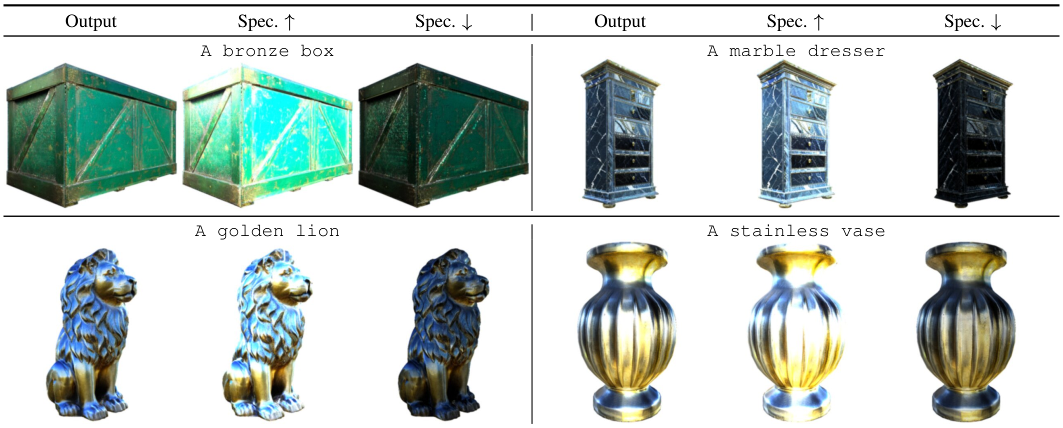


Figure 14: Qualitative results of 3D mesh texturing with PBR materials using `StochSync`.

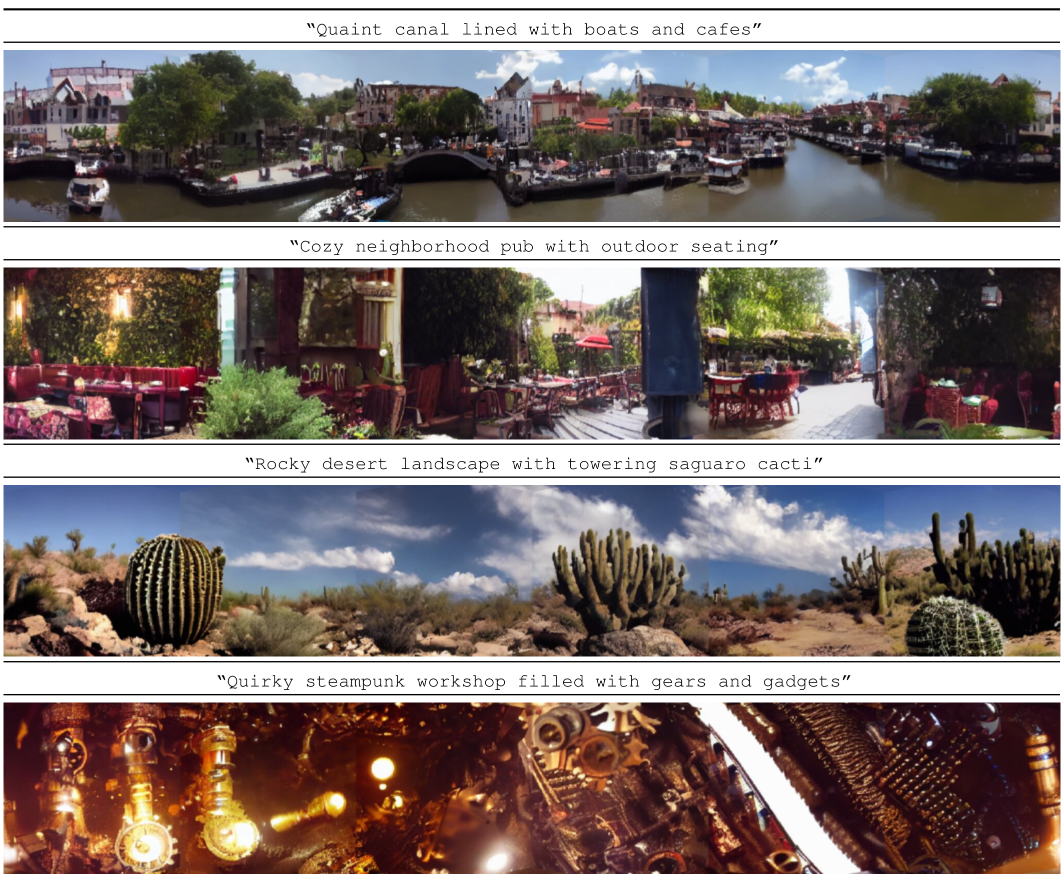


Figure 15: Qualitative results of 360° panorama generation using a video diffusion model, CameraCtrl (He et al., 2024) with `StochSync`.

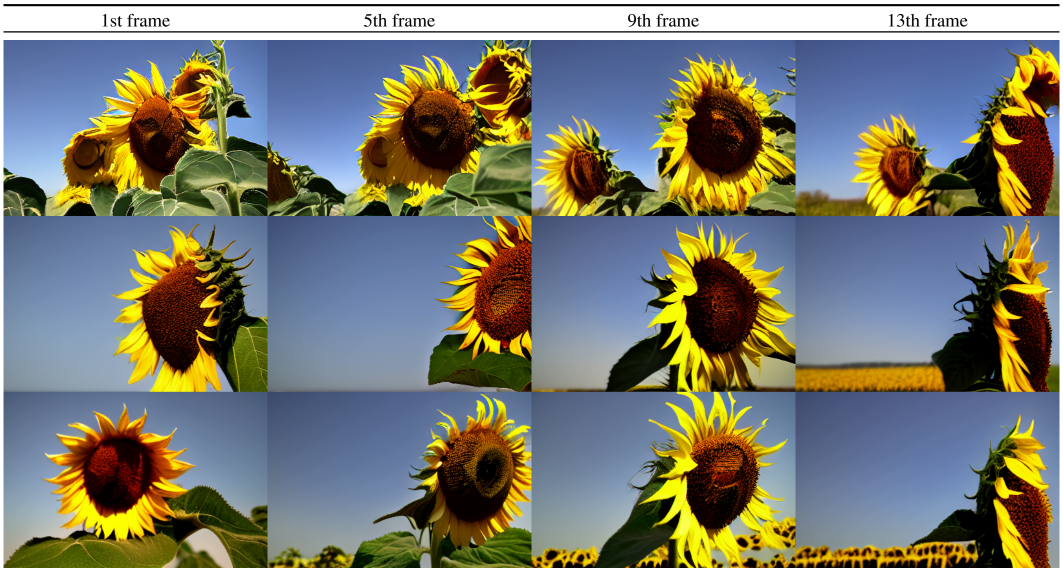


Figure 16: Videos generated using a pose-conditioned video diffusion model, CameraCtrl (He et al., 2024). Each row shows sampled frames from videos conditioned on camera trajectories with rotation angles of $90°$, $180°$, and $360°$ (from top to bottom).

Figure 17: Qualitative results of texturing 3D Gaussians (Kerbl et al., 2023) using `StochSync`.