Variational Test-time Optimization for Diffusion Synchronization

Hyunsoo Lee[1,2*†] **Farrin Marouf Sofian**[2*] **Kushagra Pandey**[2] **Stephan Mandt**[2]
[1]Seoul National University [2]University of California, Irvine
philip21@snu.ac.kr, {fmaroufs, pandeyk1, mandt}@uci.edu

Abstract

Collaborative generation, which coordinates multiple diffusion trajectories to extend the capabilities of pretrained priors, has emerged as a powerful paradigm for extending the applicability of diffusion models. Among existing approaches, diffusion synchronization provides a scenario-agnostic solution by introducing general guidance mechanisms. However, current synchronization approaches rely heavily on heuristics and still require task-specific tailoring, which limits their generalizability and performance. In this work, we mathematically derive a synchronization framework based on optimal control, providing a principled explanation of diffusion synchronization. During sampling, we optimize control variables to guide multiple trajectories toward coherent solutions while remaining close to the underlying diffusion prior. Our method operates entirely at test-time without additional training, thereby enabling broad applicability across diverse generation scenarios when combined with strong pretrained priors. We demonstrate consistent improvements over baselines on three representative collaborative generation tasks, covering a wide range of modalities and applications. Beyond performance gains, our work establishes a novel foundation for collaborative generation, opening a principled path toward extending pretrained generative models to new collaborative generation settings. Project Website: https://hleephilip.github.io/SyncVC/.

1 Introduction

Diffusion models [55, 19, 41] and flow-matching frameworks [36, 39] have demonstrated strong generative priors, achieving impressive results within their training domains [51, 53, 14, 45, 28, 60, 21, 62, 63, 33, 37, 38, 59, 24]. However, extending these models beyond their native regimes, such as generating long-horizon structures from short-horizon training, still requires heavy retraining or engineering, limiting practical usability. This highlights the importance of *collaborative generation* [31], where multiple diffusion trajectories are coupled so that they are mutually consistent while each remains plausible under the diffusion prior. While several methods address specific tasks of collaborative generation [34, 58, 67, 7, 16, 61, 10, 68, 49, 40, 65], many rely on task-specific heuristics, limiting generalizability and requiring substantial engineering effort to extend to new settings. A more desirable paradigm is *diffusion synchronization* [4, 26, 31, 64], which provides task-agnostic and unified guidance for collaborative generation. Since training a diffusion model to generate multiple coordinated trajectories is computationally expensive, synchronization is performed via test-time guidance. Rather than requiring task-specific strategies, diffusion synchronization offers a general approach that can be integrated into arbitrary priors, enabling scalable content creation across diverse scenarios.

*Equal contribution.
†Work done while visiting UC Irvine.

Preprint.

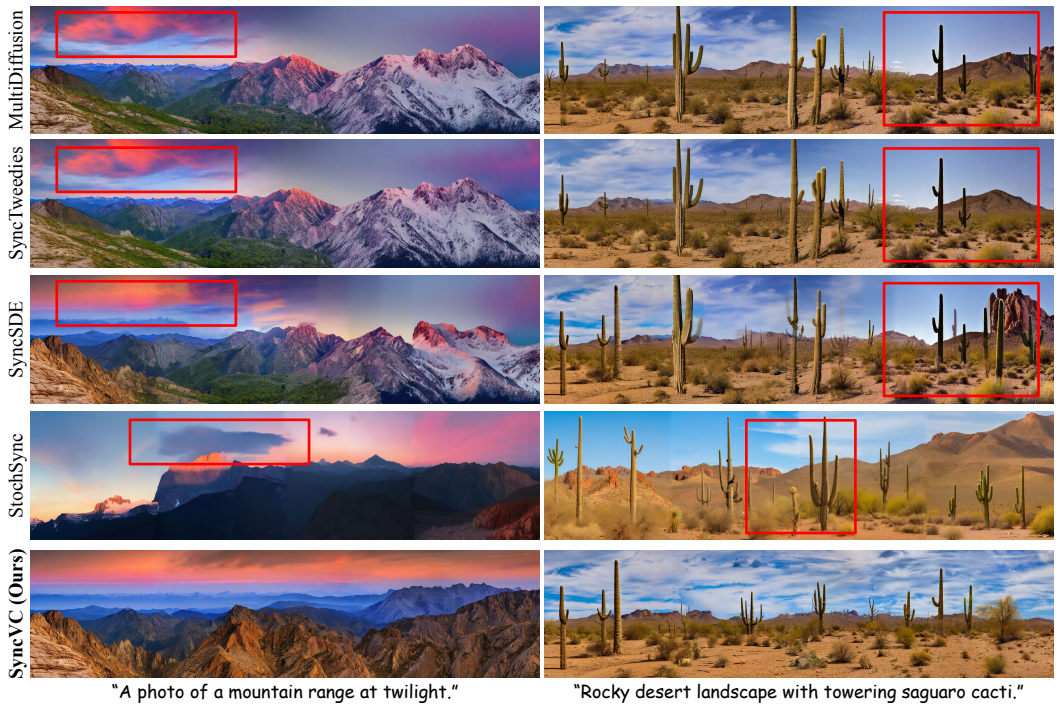


Figure 1: **Our method produces style-consistent, high-quality wide images, outperforming all baselines.** (Left) Only ours maintains a unified sky and mountain style, while baselines suffer from color inconsistency and structural discontinuities. (Right) SyncVC shows consistent sky, cacti, and ground appearance, whereas the others show varying colors in the sky and cacti, along with boundary artifacts.

Despite recent research, existing diffusion synchronization methods are largely driven by heuristics, such as relying on extensive empirical tests over numerous strategies [26] or introducing impractical Gaussian approximations of conditional scores [31]. Consequently, these approaches fail to provide a principled understanding, leading to suboptimal performance and limited applicability across diverse tasks. In this work, we address this limitation by deriving a control-based framework for diffusion synchronization. We introduce control variables into the diffusion process and formulate synchronization as a variational inference problem over trajectories.

At each diffusion timestep, we optimize the control variables using a novel loss function derived from our theoretical formulation. It balances two competing goals of collaborative generation: enforcing consistency across trajectories while remaining close to the pretrained diffusion prior. This formulation provides a principled explanation for collaborative generation, moving beyond heuristic approaches and interpreting it as controlled sampling. This principle yields a well-founded synchronization mechanism while still allowing task-specific parameterizations. To the best of our knowledge, this work is the first to propose a unified framework for collaborative generation based on optimal control. We refer to our method as **Synchronized Diffusion with Variational Controls (SyncVC)**.

The proposed framework is not only mathematically grounded but also widely applicable. Since it operates through test-time optimization, it can be applied to diverse pretrained generative models [1, 51, 69, 60, 30] without additional training. Moreover, it naturally extends across diverse collaborative generation tasks, regardless of modality. We validate our approach using three representative tasks: wide image generation, optical illusion generation, and 3D mesh texturing, where our method consistently outperforms baselines, as shown in Figure 1. Furthermore, unlike prior approaches, SyncVC accommodates external constraints such as style guidance without requiring redesign of an overall framework. This flexibility highlights the advantage of our principled formulation, enabling both strong performance and practical applicability. We summarize our contributions as follows:

- We propose SyncVC, a mathematically grounded test-time optimization framework for collaborative generation, providing a fundamental explanation of diffusion synchronization.
- SyncVC introduces control-based guidance for generative modeling, where control variables are optimized via a variational objective, yielding a general and extensible sampling mechanism across tasks and modalities.
- Our method incurs no training cost and is broadly applicable, enabling direct integration with pretrained diffusion priors while naturally benefiting from advances in stronger models.
- We demonstrate strong empirical performance across diverse collaborative generation tasks, spanning both 2D and 3D generation scenarios.

2 Related work

Task-specific methods for collaborative generation. A representative example of collaborative generation is wide image generation, where multiple trajectories for fixed-size, partially-overlapping patches are fused into a single wide image. SyncDiffusion [34] ensures consistent style along the wide image by minimizing LPIPS distance [70] between patches. Another task is optical illusion (ambiguous image) generation, which synthesizes a single image that conveys different semantics under different transformations. Anagram-MTL [61] formulates this task as a multi-task learning problem [71, 72] with attention-based regularization and CLIP-based [47] adaptive noise reweighting. In 3D graphics, text-guided mesh texturing requires consistency across multiple views and has been addressed using diffusion-based approaches [68, 65, 49, 40, 66, 8, 13]. For example, TexPainter [68] enforces multi-view consistency via color-space fusion at each diffusion step, guiding the denoising process for coherent texture generation. Meanwhile, TEXTure [49] employs a texturing-tailored diffusion process with a trimap representation, and iteratively updates texture maps from different viewpoints. However, these methods are tailored to specific tasks and lack generalizability across different collaboration scenarios. In contrast, our method does not rely on task-specific designs, but provides a general framework applicable to arbitrary tasks and modalities with strong performance.

Diffusion synchronization. Diffusion synchronization methods [4, 26, 31, 64] aim to provide general mechanisms across diverse collaborative generation scenarios. MultiDiffusion [4] aligns trajectories by optimizing diffusion latents with respect to a heuristically designed objective, resulting in a closed-form solution via latent averaging. SyncTweedies [26] empirically evaluates 60 synchronization strategies and selects the best-performing configuration, identifying averaged Tweedie estimates [50] as the optimal strategy. However, its reliance on heuristics limits its scalability and generalization beyond the evaluated settings. On the other hand, SyncSDE [31] proposes auto-regressive trajectory sampling, where the conditional score of the current trajectory given the previously generated trajectory is approximated with a Gaussian. This strong assumption limits its general applicability. StochSync [64] builds upon the SyncTweedies and interprets synchronization via score distillation sampling (SDS) [46], but still depends on heuristic engineering techniques such as non-overlapping view sampling. In contrast, our method introduces control variables into the sampling process and optimizes them using a loss function derived from variational inference. This minimizes the need for heuristic modeling, providing a principled framework that leads to improved performance across a wide range of collaborative generation tasks.

Diffusion with optimal controls. Recently, optimal control [23] has been used to design guidance in diffusion models [22, 43, 17, 35, 5, 3, 9, 52]. Stochastic Control Guidance [22] formulates guidance as a stochastic optimal control problem, leveraging path-integral control for plug-and-play guidance with non-differentiable rewards. For general inverse problems, Diffusion Trajectory Matching [43] formulates guidance as a variational control problem, where control variables are optimized to follow terminal constraints while regularizing deviations from the pretrained diffusion prior; related fast samplers have also been studied for iterative refinement models [42]. *Azangulov et al.* [3] formulates adaptive guidance scheduling as a stochastic optimal control problem, dynamically selecting the guidance scale via controls. While these approaches primarily focus on guiding a single diffusion trajectory with external constraints, our work addresses *collaborative generation*, where multiple trajectories must be coordinated simultaneously. To the best of our knowledge, we are the first to introduce a control-based framework for diffusion synchronization.

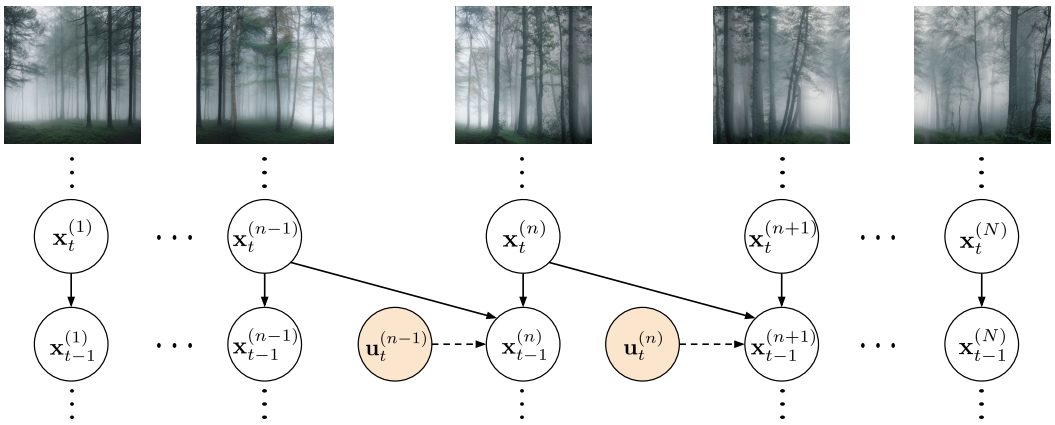


Figure 2: **Overall mechanism of SyncVC.** Control variables are introduced into the diffusion process for collaborative generation through synchronized diffusion. We visualize the case of wide image generation, where each diffusion trajectory models a partially overlapping image patch.

3 Collaborative Generation with Synchronized Variational Controls

Problem formulation. Given an observation $\boldsymbol{y} \in \mathbb{R}^m$, our goal is to generate a set of N consistent elements $\boldsymbol{X} = \{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}\}$ ($\mathbf{x}_0^{(i)} \in \mathbb{R}^d$) that maximizes the likelihood of the observation $\boldsymbol{y}$. For instance, for wide image generation, the sequence elements can be overlapping image patches. We begin by defining the likelihood $p(\boldsymbol{y} \mid \boldsymbol{X})$ as an energy-based reward function, $r : \mathbb{R}^m \times \mathbb{R}^{Nd} \to \mathbb{R}$,

$$p(\boldsymbol{y} \mid \boldsymbol{X}) \propto \exp\left(r(\boldsymbol{y}, \boldsymbol{X})\right). \tag{1}$$

The form of the reward is task-specific. For instance, for stylized wide image generation, the reward can be defined as maximizing the overlap between consecutive sequence elements [31], and $\boldsymbol{y}$ may additionally encode conditioning information from a style transfer task [15]. We will discuss several parameterizations of the rewards considered in this work in the following paragraphs. For the remainder of this work, we restrict our focus to rewards which are **known**, **differentiable**, and can be evaluated in **closed form**. We model the prior over each set element $p(\mathbf{x}_0^{(n)})$ with a pretrained diffusion model using T denoising steps,

$$p(\mathbf{x}_0^{(n)}) = \int p(\mathbf{x}_T^{(n)}) \prod_t p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)})\, d\mathbf{x}_{1:T}^{(n)}, \tag{2}$$

where $p_{\boldsymbol{\phi}}$ denotes the denoising kernel with mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}_t^{(n)}, t)$, and t denotes the diffusion timestep. The distribution $p(\mathbf{x}_T^{(n)})$ is typically modeled as a standard Gaussian. We make two further assumptions. First, we operate under the regime of training-free test-time guidance and thus keep the pretrained diffusion model fixed. Secondly, while the pretrained diffusion model can also be conditioned on additional information (*e.g.* a text prompt [51, 60]), we drop this in the notation for convenience. As noted earlier, we aim to generate a sequence $\boldsymbol{X}$ which maximizes the likelihood of $\boldsymbol{y}$. Formally, we want to sample from the following tilted distribution,

$$q(\boldsymbol{X} \mid \boldsymbol{y}) \propto p(\boldsymbol{X}) \exp\left(r(\boldsymbol{y}, \boldsymbol{X})/\beta\right). \tag{3}$$

We define a prior over the sequence as $p(\boldsymbol{X}) = \prod_n^N p(\mathbf{x}_0^{(n)})$. We use this factorized prior for simplicity and because it works well empirically (see Sec. 4). However, we note that this is not a limitation of our framework, as more expressive parameterizations of the prior are possible and can be an interesting direction for future work. In most cases, this tilted distribution $q(\boldsymbol{X} \mid \boldsymbol{y})$ is intractable to compute in closed form. Therefore, we rely on variational inference to approximate the latter, which we discuss next.

Diffusion synchronization via variational inference. We define the variational distribution over the sequence $\boldsymbol{X}$ with a diffusion process:

$$q(\boldsymbol{X} \mid \boldsymbol{y}) = \int q\Big(\mathbf{x}_{0:T}^{(1)}\Big) \prod_{n=2}^{N} \prod_{t=1}^{T} q\Big(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \boldsymbol{y}\Big)\, d\mathbf{x}_{1:T}^{(1:n)}. \tag{4}$$

This factorization is natural for an ordered sequence $\{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}\}$ and may be sub-optimal in settings without such ordering, but we find that it works well empirically (see Sec. 4) and therefore do not explore alternative schemes. By conditioning the reverse transition $q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \boldsymbol{y})$ on the noisy latent $\mathbf{x}_t^{(n-1)}$ of the previous trajectory, the generation of $\mathbf{x}_0^{(n)}$ is synchronized with $\mathbf{x}_0^{(n-1)}$ at every diffusion step.

A natural choice to model the distribution $q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \boldsymbol{y})$ is a conditional Gaussian approximation [31]. To steer generation toward the reward r, we augment the variational distribution with additional variational parameters $\mathbf{u}_t^{(n-1)}$ at each step. These auxiliary variables couple adjacent denoising trajectories, thereby enabling collaborative generation (see Figure 6). Combining the generative model in Eq. 2 with the augmented variational distribution, the evidence lower bound (ELBO) takes the form (see Appendix A),

$$\mathcal{L}(\boldsymbol{y}) := \mathbb{E}_q[r(\boldsymbol{y}, \boldsymbol{X})] - \lambda \sum_{n=2}^{N} \sum_{t=1}^{T} D_{\mathrm{KL}}\Big(q\Big(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \mathbf{u}_t^{(n-1)}, \boldsymbol{y}\Big) \,\|\, p\Big(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}\Big)\Big). \tag{5}$$

Eq. 5 captures the two competing goals of collaborative generation. The first term encourages sequences that maximize the expected reward, while the second pulls samples from the variational distribution toward the noisy submanifold defined by the prior diffusion model at each denoising step. The scalar hyperparameter λ controls the strength of this regularization. For any observation $\boldsymbol{y}$, we optimize Eq. 5 to infer the variational parameters $\mathbf{u}_t^{(n-1)}$ directly at test time. We parameterize the augmented variational distribution as a unimodal Gaussian distribution with mean

$$\bar{\boldsymbol{\mu}}_t^{(n)} = \underbrace{\boldsymbol{\mu}_{\boldsymbol{\phi}}\Big(\bar{\mathbf{x}}_t^{(n)}, t\Big)}_{\text{Terminal Step}} - \frac{\gamma}{2}\sigma_t^2 \underbrace{\nabla_{\mathbf{x}_t^{(n)}} \Big\| f\left(\mathbf{x}_t^{(n-1)}, \boldsymbol{y}\right) - \bar{\mathbf{x}}_t^{(n)} \Big\|_2^2}_{\text{Regularizer}}, \tag{6}$$

where $\bar{\mathbf{x}}_t^{(n)} = \mathbf{x}_t^{(n)} + \beta \mathbf{u}_t^{(n-1)}$ and $\beta > 0$ is a hyperparameter that defines the strength of the controls. $f(\cdot, \cdot)$ is an operator defined by the reward function, and practical choices are described in the next paragraph. The first term perturbs the denoising trajectory at time t along the direction of $\mathbf{u}_t^{(n-1)}$ to maximize the reward; following [43], we refer to these auxiliary variables as *variational controls*. The second term steers the trajectory to reduce the gap between adjacent denoising chains, acting as a regularizer. Together, the two terms guide each denoising trajectory toward higher reward while maintaining consistency across trajectories. We refer to this overall framework as **Synchronized Diffusion with Variational Controls (SyncVC)** and summarize it in Fig. 2.

Choice of the reward. Because the variational distribution in Eq. 4 generates the sequence autoregressively, the reward function must respect the same causal ordering. We therefore decompose the overall reward into a sum of sub-rewards, where the n-th term depends only on the current element and those preceding it:

$$r(\boldsymbol{y}, \boldsymbol{X}) := \sum_{n=2}^{N} \tilde{r}\Big(\boldsymbol{y}, \mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(n)}\Big). \tag{7}$$

This decomposition exposes a natural design principle: each $\tilde{r}$ should measure how well the new element $\mathbf{x}_0^{(n)}$ agrees with the elements already generated, under a task-specific notion of consistency. Below, we instantiate $\tilde{r}$ for the three collaborative generation tasks studied in this work. We deliberately keep these designs simple and intuitive rather than relying on heavily engineered components; as shown in Sec. 4, even these straightforward choices already outperform prior baselines, and we leave more sophisticated parameterizations to future work.

Wide image generation. The goal is to synthesize a horizontally elongated image from a text prompt $\boldsymbol{y}$, where each sequence element $\mathbf{x}_0^{(n)}$ corresponds to a patch and adjacent patches partially overlap. Consistency then amounts to agreement on the shared region between neighbors:

$$\tilde{r}\Big(\boldsymbol{y}, \mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(n)}\Big) = -\frac{\gamma}{2} \Big\| \mathbf{M} \odot \Big(f\Big(\mathbf{x}_0^{(n-1)}\Big) - \mathbf{x}_0^{(n)}\Big) \Big\|_2^2, \tag{8}$$

where $f(\cdot)$ shifts its input along the x-axis by the patch stride, $\mathbf{M}$ is a binary mask selecting the overlap region, and γ is a tunable weight. Intuitively, this reward pulls the left side of the n-th

patch toward the right side of the $(n-1)$-th patch. Although our framework can incorporate more general reward functions, we use Eq. 8 as the main reward because it yielded the best results in our experiments. We also evaluate a CLIP-augmented variant [47] that adds a semantic guidance term based on the similarity between $\mathbf{x}_0^{(n)}$ and the text prompt $\boldsymbol{y}$, and provide results in Appendix D.

Optical illusion generation. The task is to synthesize a single image whose semantic content changes under a fixed transformation, such as rotation or flipping. The observation $\boldsymbol{y}$ comprises two text prompts, and we sample two trajectories—one per prompt—under the corresponding views. Consistency here means that the two trajectories should agree once the transformation is applied:

$$\tilde{r}\Big(\boldsymbol{y}, \mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}\Big) = -\frac{\gamma}{2}\left\| f\Big(\mathbf{x}_0^{(1)}\Big) - \mathbf{x}_0^{(2)} \right\|_2^2, \tag{9}$$

where $f(\cdot)$ is the illusion transformation operator. The reward therefore encourages the second trajectory to match the transformed version of the first, so that both prompts are simultaneously satisfied in a single image.

Text-guided 3D mesh texturing. The main challenge in this task is multi-view consistency: each trajectory $\mathbf{x}_0^{(n)}$ generates a 2D image from a different viewpoint, and these images must collectively define a coherent texture on the input mesh. Here $\boldsymbol{y}$ consists of the text prompt together with the source mesh. To define the sub-reward at step n, we first bake an auxiliary texture from the previously generated views $\{\mathbf{x}_0^{(j)}\}_{j=1}^{n-1}$, then render this texture from the n-th viewpoint and compare with $\mathbf{x}_0^{(n)}$:

$$\tilde{r}\Big(\boldsymbol{y}, \mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(n)}\Big) = -\frac{\gamma}{2}\left\| \mathbf{M}^{(n)} \odot \Big(f\Big(\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(n-1)}, \boldsymbol{y}, n\Big) - \mathbf{x}_0^{(n)} \Big) \right\|_2^2, \tag{10}$$

Here, $f(\cdots, \boldsymbol{y}, n)$ composes texture baking and rendering: texture baking fuses the previous multi-view images into a texture map, and rendering projects that texture from the n-th viewpoint. The mask $\mathbf{M}^{(n)}$ selects the foreground region in the rendered image. In effect, this reward asks each new view to remain faithful to what the texture already implies from earlier views.

Practical considerations. The reward functions in Eqs. 8 and 9 are defined on the clean sequence elements $\mathbf{x}_0^{(n)}$. Direct optimization of the variational objective in Eq. 5 would therefore require rolling out the full reverse diffusion chain at every timestep to evaluate $r(\boldsymbol{y}, \boldsymbol{X})$, which is computationally prohibitive. Instead, we approximate the clean sample at the current timestep using Tweedie's estimate. Therefore, the loss function simplifies to the following objective:

$$\mathbf{u}_t^\star = \arg\min_{\mathbf{u}_t} \sum_{n=2}^{N} \left[-\tilde{r}\Big(\boldsymbol{y}, \hat{\mathbf{x}}_{0|t}^{(1)}, \ldots, \hat{\mathbf{x}}_{0|t}^{(n)}\Big) + \lambda \left\| \bar{\boldsymbol{\mu}}_t^{(n)} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\Big(\mathbf{x}_t^{(n)}, t\Big) \right\|_2^2 \right], \tag{11}$$

where Tweedie's estimate is given by $\hat{\mathbf{x}}_{0|t}^{(n)} = \mathbb{E}\left[\mathbf{x}_0 \mid \bar{\mathbf{x}}_t^{(n)}\right]$.

Reformulated objective for DDIM . Under the DDIM [56] parameterization, we derive a simplified objective as follows (see Appendix B):

$$\begin{aligned} \mathbf{u}_t^\star = \arg\min_{\mathbf{u}_t} \sum_{n=2}^{N} \Bigg[& -\tilde{r}\Big(\boldsymbol{y}, \hat{\mathbf{x}}_{0|t}^{(1)}, \ldots, \hat{\mathbf{x}}_{0|t}^{(n)}\Big) + \lambda a_t^2 \left\| \mathbf{u}_t^{(n-1)} \right\|_2^2 \\ & + \lambda b_t^2 \left\| \epsilon_\theta(\bar{\mathbf{x}}_t^{(n)}, t) + \frac{\gamma}{2}\sqrt{1-\alpha_t}\nabla_{\mathbf{x}_t^{(n)}} \left\| f\left(\mathbf{x}_t^{(n-1)}, \boldsymbol{y}\right) - \bar{\mathbf{x}}_t^{(n)} \right\|_2^2 - \epsilon_\theta(\mathbf{x}_t^{(n)}, t) \right\|_2^2 \Bigg], \end{aligned} \tag{12}$$

where $\epsilon_\theta(\cdot, \cdot)$ denotes the noise prediction network,

$$a_t = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \quad \text{and} \quad b_t = \sqrt{1-\alpha_{t-1}} - \sqrt{\frac{(1-\alpha_t)\alpha_{t-1}}{\alpha_t}}. \tag{13}$$

4 Experiments

In this section, we evaluate the practical effectiveness of SyncVC across key tasks introduced in Sec. 3. Our method is implemented with PyTorch [44], and control variables are optimized using Adam optimizer [27]. DDIM [56] and classifier-free guidance [20] are used for diffusion sampling across all tasks. For all tables, we **bold the best** and underline the second-best results. Task-specific experimental details and results are provided in the corresponding subsections and Appendix D.

Table 1: Quantitative evaluation on wide image generation. The proposed method outperforms all baselines, with a particularly large margin in Intra-Style-Loss [15] and χ^2-Histogram distance, which measure style and color consistency across the wide image, respectively. KID [6] is scaled by 10^3.

Method	MultiDiffusion [4]	SyncTweedies [26]	SyncSDE [31]	StochSync [64]	**SyncVC (Ours)**
Intra-LPIPS [70] ↓	0.637	0.620	0.653	<u>0.617</u>	**0.592**
Intra-Style-Loss [15] ↓	<u>58.46</u>	78.05	63.98	67.56	**44.34**
χ^2-Histogram dist. ↓	<u>1.211</u>	1.345	1.307	1.352	**0.751**
Histogram intersect. ↑	<u>0.549</u>	0.519	0.526	0.518	**0.665**
KID [6] ↓	58.26	60.81	<u>57.08</u>	100.47	**52.07**

4.1 Wide image generation

Evaluation protocol. We generate 2048×512 wide images using the pretrained Stable Diffusion [51], with each patch of size 512^2. For SyncVC, five patches are sampled with an overlap of 128 pixels and are sequentially composited to form the final wide image, whereas baselines use their default overlapping configurations. We compare the proposed approach with diffusion synchronization methods [4, 31, 26, 64]. For evaluation, we adopt 15 text prompts from prior works [4, 31, 26, 64] and generate 50 wide images per prompt.

For wide image generation, maintaining consistency across patches is the most important criterion. To evaluate coherence, we crop each generated wide image into four non-overlapping views and measure all possible pairwise relationships among them. Specifically, for perceptual and stylistic alignment, we leverage Intra-LPIPS and Intra-Style-Loss from prior work [34]. In addition, to assess color alignment, we compute color histograms in the HSV space for each non-overlapping view and measure their χ^2 distance and histogram intersection. We also measure KID [6] using randomly cropped views, to assess image quality and diversity.

Results. Table 1 shows that our method consistently outperforms all baselines, with particularly large gains in Intra-Style-Loss and χ^2-Histogram distance, which measure style and color consistency, respectively. It also demonstrates strong distributional alignment as reflected by KID. Figure 1 visualizes qualitative results. While baselines exhibit inconsistent styles and discontinuities along the horizontal axis, SyncVC produces smooth transitions with a unified style. We further demonstrate that our method can be applied beyond Stable Diffusion by synthesizing high-resolution wide images using the pretrained SANA model [60], which provides stronger generative priors. These results are visualized in Appendix D.

Incorporating additional conditions. Unlike prior works that rely on closed-form guidance, SyncVC naturally accommodates additional constraints through a reasonable reward design. As a representative example, we consider style guidance for wide image generation by adding a style transfer loss [15] between a style reference and $\mathbf{x}_0^{(n)}$ within the reward function of Eq. 8. Figure 3 (b) shows that our method can flexibly incorporate style guidance, highlighting its strength as a fundamental framework that can accommodate a broad class of reward parameterizations.

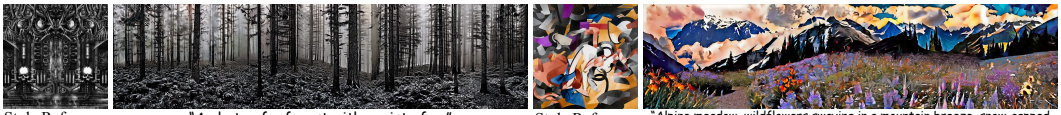


Figure 3: **SyncVC enables flexible generation under external constraints such as style guidance**, transferring texture and overall color from the style reference while preserving the semantics of the given prompt without artifacts.

4.2 Optical illusion generation

Evaluation protocol. We generate images using the pretrained DeepFloyd-IF [1]. For evaluation, we adopt 5 pairs of (transformation, prompt) from prior work [16, 31] and generate 100 images for each pair. The final output consists of two views: the image from the second trajectory (view 2) and its inverse-transformed counterpart (view 1), both of which are used for evaluation. We compare against synchronization methods [26, 31], and a task-specific method [61]. For metrics, we measure FID [18], KID [6] to quantify distributional alignment and MUSIQ [25] to assess image quality.

Results. We show quantitative results in Table 2, with qualitative comparisons in Figure 4. Our method consistently achieves outstanding performance across all metrics. In particular, SyncTweedies [26] tends to produce blurry images with lower aesthetic scores, whereas the proposed method maintains high visual quality while clearly encoding both semantic interpretations within a single image. This highlights both the limitations of heuristic-based modeling and the advantages of our principled formulation that coherently models the interaction between trajectories.

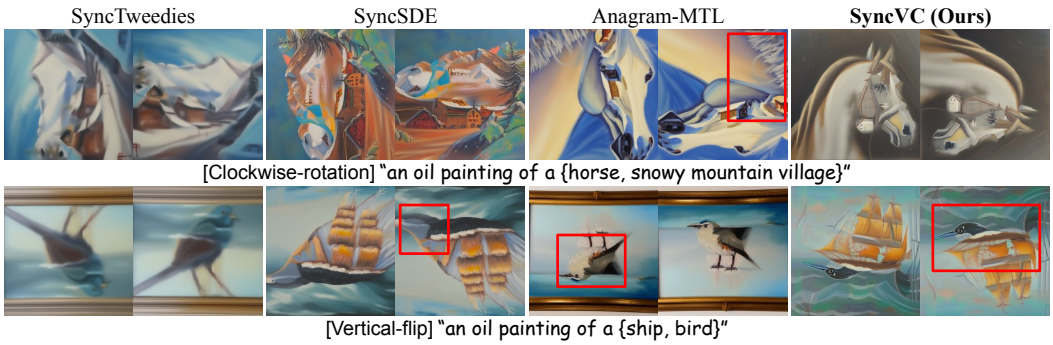


Figure 4: **Our method outperforms all baselines by clearly encoding both semantics under illusion while maintaining high quality.** For each method, we visualize both views (view 1 & 2) of the final result. (Row 1) SyncTweedies [26] produces a blurry image with low quality, Anagram-MTL [61], although tailored for this task, also generates some artifacts (denoted as bounding box). (Row 2) SyncTweedies still results in a blurry image, while both SyncSDE [31] and Anagram-MTL struggle to simultaneously encode both semantics (bird and ship, respectively).

Table 2: Quantitative evaluation on optical illusion generation. Our approach outperforms all baselines, in terms of both distributional alignment (FID [18], KID [6]) and image quality (MUSIQ [25]). KID score is scaled by 10^3.

Method	SyncTweedies [26]	SyncSDE [31]	Anagram-MTL [61]	**SyncVC (Ours)**
FID [18] ↓	<u>255.37</u>	264.10	256.96	**252.87**
KID [6] ↓	195.52	<u>176.14</u>	190.64	**175.47**
MUSIQ [25] ↑	32.28	<u>52.74</u>	41.92	**54.40**

4.3 Text-guided 3D mesh texturing

Evaluation protocol. We generate diffusion trajectories using the pretrained depth-conditioned ControlNet [69]. For SyncVC, we define 8 trajectories by uniformly sampling azimuth angles at a fixed elevation, producing partially overlapping views, while following default setups for baselines. Images generated from each trajectory are used to synthesize the final texture. We compare our method against synchronization approaches [26, 31, 64] and task-specific methods [68, 49]. For evaluation, we use 350 (mesh, prompt) pairs sampled from the Objaverse dataset [11]. Each textured mesh is rendered from 10 viewpoints, and the resulting images are used to compute FID [18], KID [6], and CLIP image–text similarity (CLIP-S) [47].

Results. Table 3 shows that the proposed method outperforms all baselines. As illustrated in Figure 5, baselines often exhibit artifacts and inconsistent textures, whereas our method produces high-quality and detailed textures while alleviating such artifacts. These results demonstrate that SyncVC generalizes well across different modalities and dimensional settings, highlighting its effectiveness as a fundamental framework for collaborative generation.

Table 3: Quantitative evaluation on text-guided 3D mesh texturing. SyncVC shows superior performance across all baselines in terms of distributional alignment (FID [18], KID [6]), and comparable results on image-text alignment (CLIP-S [47]). KID score is scaled by 10^3.

Method	TexPainter [68]	TEXTure [49]	SyncTweedies [26]	SyncSDE [31]	StochSync [64]	**SyncVC (Ours)**
FID [18] ↓	192.08	188.26	163.08	172.74	<u>162.71</u>	**161.63**
KID [6] ↓	110.84	96.28	82.05	85.12	<u>80.11</u>	**75.17**
CLIP-S [47] ↑	0.283	0.288	**0.292**	0.288	0.289	<u>0.290</u>

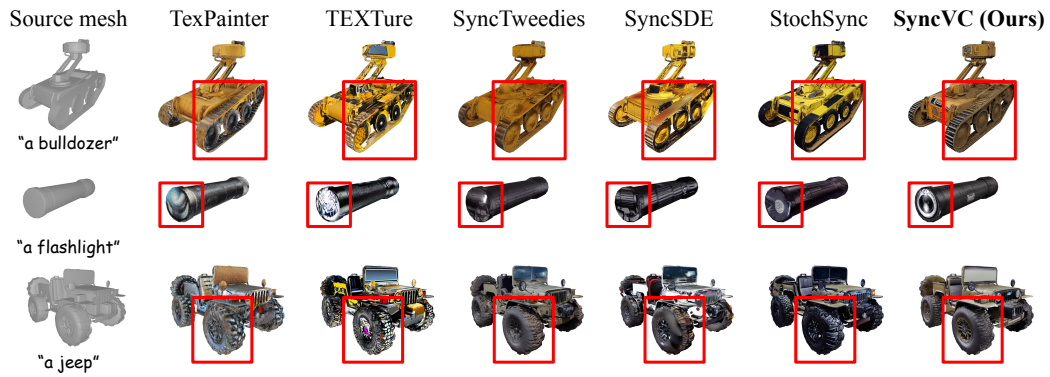


Figure 5: [Best viewed when magnified.] **Our method outperforms baselines on 3D mesh texturing by producing artifact-free and realistic textures.** We emphasize that SyncVC well preserves fine details such as the chain structure on the bulldozer tracks (Row 1), detailed view of the flashlight's front lens (Row 2), and the overall natural appearance of the vehicle, including fine-grained textures on the tires (Row 3), while baselines generate over-smoothed and unrealistic textures.

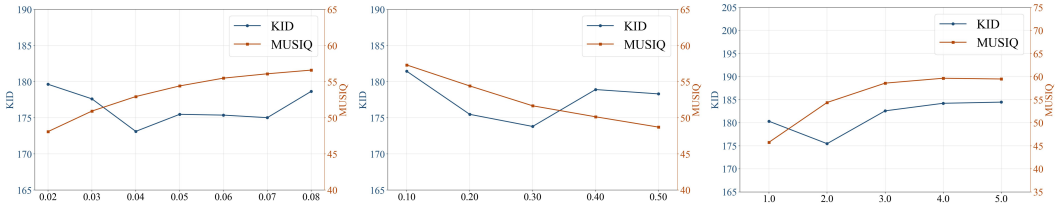
4.4 Additional analysis

Effectiveness of SyncVC on extreme scenarios. We further consider a more challenging wide image generation setting that uses a very small overlap of 16 pixels, to show the effectiveness of SyncVC under extreme conditions. This setting is also practically important, as smaller overlaps require fewer trajectories for the same resolution, thereby reducing computational cost. Under this setting, we compare our method with MultiDiffusion [4], which achieves the best performance among the baselines (see Table 1). As shown in Figure 6, baselines exhibit significant performance degradation, whereas our method maintains strong style consistency.- We attribute this robustness to the coupling kernel introduced in Eq. 5, since marginalizing the control variables yields a multimodal distribution $q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \boldsymbol{y})$ that is capable of modeling complex relationships between trajectories.

Figure 6: **Our method shows superior performance in wide image generation under an extreme small-overlap setting** (16 pixels, 3.125% of patch width). SyncVC maintains coherent style and consistent colors across patches, whereas MultiDiffusion [4] fails to produce visually consistent results. This result stems from introducing variational controls, which more effectively models complex correlations between trajectories than heuristic approximations used in baselines.

Effects of hyperparameters. Our parameterization contains three tunable coefficients: the weight of the reward function (γ), the weight of the KL-divergence term in ELBO (λ), and the control strength (β). Figure 7 (a) shows the effects of these coefficients in optical illusion generation. Firstly, as γ increases from a small value, it better captures both semantics simultaneously, leading to improved KID scores. However, excessively large γ over-constrains the 2nd trajectory to resemble the 1st one, causing only one semantic to dominate. As a result, although the visual quality (MUSIQ) may improve, both semantics are no longer jointly captured and KID degrades. Secondly, increasing λ regularizes the 2nd trajectory toward the original diffusion prior, thereby reducing the influence of the 1st trajectory. This produces an effect analogous to decreasing γ. Lastly, increasing β initially

improves the overall quality as the controls strongly guide the trajectory to satisfy the objective in Eq. 11. However, too large β may weaken the ability to jointly capture both semantics.



(a) Effect of γ (b) Effect of λ (c) Effect of β

Figure 7: **Effects of hyperparameters (γ, λ, β) on the optical illusion generation task.** These values offer a trade-off between jointly capturing both semantics (KID) and visual quality (MUSIQ).

5 Conclusion

In this work, we propose a principled framework for collaborative generation based on optimal control. Unlike prior approaches that rely heavily on heuristic designs, our method is derived from a mathematically grounded formulation, providing a novel perspective on diffusion synchronization. The proposed method demonstrates strong performance across diverse collaborative generation tasks, establishing a promising direction for extending pretrained generative priors to more versatile settings. Despite these advantages, our approach has several limitations. Because it relies on test-time optimization, it incurs additional computational cost compared to optimization-free approaches, motivating the development of more efficient guidance strategies. Furthermore, the formulation is currently restricted to differentiable rewards; extending it to incorporate non-differentiable objectives would enable broader applicability across diverse scenarios.

Acknowledgments and Disclosure of Funding

We thank Justus Will and Jan Groeneveld for additional discussions and feedback. Stephan Mandt acknowledges funding from the National Science Foundation (NSF) through an NSF CAREER Award IIS-2047418, IIS2007719, the NSF LEAP Center.

References

[1] DeepFloyd Lab at StabilityAI. Deepfloyd if. `https://github.com/deep-floyd/IF`, 2023.

[2] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM computing surveys (CSUR)*, 1991.

[3] Iskander Azangulov, Peter Potaptchik, Qinyu Li, Eddie Aamari, George Deligiannidis, and Judith Rousseau. Adaptive diffusion guidance via stochastic optimal control. *AISTATS*, 2026.

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

[5] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *TMLR*, 2024.

[6] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *ICLR*, 2018.

[7] 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.

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

[9] Tianrong Chen, Jiatao Gu, Laurent Dinh, Evangelos Theodorou, Joshua M. Susskind, and Shuangfei Zhai. Generative modeling with phase stochastic bridge. In *ICLR*, 2024.

[10] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *SIGGRAPH*, 2024.

[11] 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. *arXiv:2212.08051*, 2022.

[12] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021.

[13] Yan Dongyu, Leyi Wu, Jiantao Lin, Luozhou Wang, Tianshuo Xu, Zhifei Chen, Zhen Yang, Lie Xu, Shunsi Zhang, and Yingcong Chen. Flexpainter: Flexible and multi-view consistent texture generation. *arXiv:2506.02620*, 2025.

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

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

[17] D. Geyfman, F. Draxler, J. N. Groeneveld, H. Lee, T. Karaletsos, and S. Mandt. Calibrated test-time guidance for bayesian inference. In *ICML*, 2026.

[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. *NIPS*, 2017.

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

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022.

[22] Yujia Huang, Adishree Ghatare, Yuanzhe Liu, Ziniu Hu, Qinsheng Zhang, Chandramouli S Sastry, Siddharth Gururani, Sageev Oore, and Yisong Yue. Symbolic music generation with non-differentiable rule guided diffusion. *ICML*, 2024.

[23] HJ Kappen. Stochastic optimal control theory. *ICML*, 2008.

[24] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *ICCV*, 2023.

[25] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, 2021.

[26] Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. Synctweedies: A general generative framework based on synchronized diffusions. *NeurIPS*, 2024.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[28] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[29] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM TOG*, 2020.

[30] D. Le, T. Pham, S. Lee, C. Clark, A. Kembhavi, S. Mandt, R. Krishna, and J. Lu. One diffusion to generate them all. In *CVPR*, 2025.

[31] Hyunjun Lee, Hyunsoo Lee, and Sookwan Han. Syncsde: A probabilistic framework for diffusion synchronization. In *CVPR*, 2025.

[32] Hyunsoo Lee, Minsoo Kang, and Bohyung Han. Conditional score guidance for text-driven image-to-image translation. *NeurIPS*, 2023.

[33] Taegyeong Lee, Soyeong Kwon, and Taehwan Kim. Grid diffusion models for text-to-video generation. In *CVPR*, 2024.

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

[35] Henry Li and Marcus Pereira. Solving inverse problems via diffusion optimal control. *NeurIPS*, 2024.

[36] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023.

[37] 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. *ICML*, 2023.

[38] Huadai Liu, Jialei Wang, Rongjie Huang, Yang Liu, Heng Lu, Zhou Zhao, and Wei Xue. Flashaudio: Rectified flow for fast and high-fidelity text-to-audio generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

[39] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ICLR*, 2023.

[40] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *SIGGRAPH Asia*, 2024.

[41] K. Pandey and S. Mandt. A complete recipe for diffusion generative models. In *ICCV*, 2023.

[42] K. Pandey, R. Yang, and S. Mandt. Fast samplers for inverse problems in iterative refinement models. In *NeurIPS*, 2024.

[43] Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. *ICML*, 2025.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

[46] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ICLR*, 2023.

[47] 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.

[48] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[49] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *SIGGRAPH*, 2023.

[50] Herbert E Robbins. An empirical bayes approach to statistics. *Breakthroughs in Statistics: Foundations and basic theory*, 1992.

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

[52] Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. RB-modulation: Training-free stylization using reference-based modulation. In *ICLR*, 2025.

[53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

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

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

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

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

[59] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *ICLR*, 2023.

[60] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *ICLR*, 2025.

[61] Zhiyuan Xu, Yinhe Chen, Huan-ang Gao, Weiyan Zhao, Guiyu Zhang, and Hao Zhao. Diffusion-based visual anagram as multi-task learning. In *WACV*, 2025.

[62] R. Yang, P. Srivastava, and S. Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 2023.

[63] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *ICLR*, 2025.

[64] Kyeongmin Yeo, Jaihoon Kim, and Minhyuk Sung. Stochsync: Stochastic diffusion synchronization for image generation in arbitrary spaces. *ICLR*, 2025.

[65] 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. In *CVPR*, 2024.

[66] 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.

[67] 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*, 2024.

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

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

[70] 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.

[71] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 2018.

[72] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 2021.

A Derivation of the ELBO (Eq. 5)

Let $\boldsymbol{U} := \{\mathbf{u}_t^{(n-1)}\}_{n=2,\,t=1}^{N,\,T}$. The joint generative model factorizes as

$$p(\mathbf{x}_{0:T}^{(1:N)}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \mathbf{x}_0^{(1:N)}) \prod_{n=1}^{N} p(\mathbf{x}_T^{(n)}) \prod_{n=1}^{N} \prod_{t=1}^{T} p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}), \tag{14}$$

with $\log p(\boldsymbol{y} \mid \mathbf{x}_0^{(1:N)}) = r(\boldsymbol{y}, \boldsymbol{X}) - \log Z(\boldsymbol{y})$ from Eq. 1. Augmenting the variational distribution of Eq. 4 with the controls gives

$$q(\mathbf{x}_{0:T}^{(1:N)} \mid \boldsymbol{y}; \boldsymbol{U}) = q(\mathbf{x}_{0:T}^{(1)}) \prod_{n=2}^{N} q(\mathbf{x}_T^{(n)}) \prod_{n=2}^{N} \prod_{t=1}^{T} q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \mathbf{u}_t^{(n-1)}, \boldsymbol{y}). \tag{15}$$

The first trajectory is sampled from the pretrained prior and reverse sampling is initialized at the prior's terminal Gaussian; we therefore set $q(\mathbf{x}_{0:T}^{(1)}) = p(\mathbf{x}_{0:T}^{(1)})$ and $q(\mathbf{x}_T^{(n)}) = p(\mathbf{x}_T^{(n)})$ for $n \geq 2$. Jensen's inequality yields

$$\log p(\boldsymbol{y}) \geq \mathbb{E}_q \Big[\log p(\mathbf{x}_{0:T}^{(1:N)}, \boldsymbol{y}) - \log q(\mathbf{x}_{0:T}^{(1:N)} \mid \boldsymbol{y}; \boldsymbol{U}) \Big]. \tag{16}$$

Substituting Eqs. 14 and 15 and cancelling the factors that coincide under the assumed q,

$$\begin{aligned} \log p(\mathbf{x}_{0:T}^{(1:N)}, \boldsymbol{y}) - \log q(\mathbf{x}_{0:T}^{(1:N)} \mid \boldsymbol{y}; \boldsymbol{U}) &= \log p(\boldsymbol{y} \mid \mathbf{x}_0^{(1:N)}) \\ &\quad + \sum_{n=2}^{N} \sum_{t=1}^{T} \log \frac{p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)})}{q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \mathbf{u}_t^{(n-1)}, \boldsymbol{y})}. \end{aligned} \tag{17}$$

The likelihood term contributes $\mathbb{E}_q[r(\boldsymbol{y}, \boldsymbol{X})] - \log Z(\boldsymbol{y})$, while the tower property turns each transition log-ratio into a negative KL divergence under the marginal $q(\mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)})$. Combining with Eq. 16,

$$\begin{aligned} \log p(\boldsymbol{y}) \geq\ & \mathbb{E}_q[r(\boldsymbol{y}, \boldsymbol{X})] - \log Z(\boldsymbol{y}) \\ & - \sum_{n=2}^{N} \sum_{t=1}^{T} D_{\mathrm{KL}} \Big(q(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \mathbf{u}_t^{(n-1)}, \boldsymbol{y}) \,\|\, p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}) \Big). \end{aligned} \tag{18}$$

Since $\log Z(\boldsymbol{y})$ is independent of $\boldsymbol{U}$, optimizing this bound is equivalent to optimizing the objective in Eq. 5, with λ generalizing the unit weighting on the KL terms. □

B Derivation of the objective for DDIM (Eq. 12)

Under the DDIM [56] parameterization, $\boldsymbol{\mu}_{\boldsymbol{\phi}}\left(\mathbf{x}_t^{(n)}, t\right)$ is defined as follows:

$$\boldsymbol{\mu}_{\boldsymbol{\phi}}\left(\mathbf{x}_t^{(n)}, t\right) = \underbrace{\sqrt{\frac{\alpha_{t-1}}{\alpha_t}}}_{a_t} \mathbf{x}_t^{(n)} + \underbrace{\left(\sqrt{1-\alpha_{t-1}} - \sqrt{\frac{(1-\alpha_t)\alpha_{t-1}}{\alpha_t}} \right)}_{b_t} \epsilon_\theta(\mathbf{x}_t^{(n)}, t). \tag{19}$$

For $\bar{\boldsymbol{\mu}}_t^{(n)}$ in Eq. 6, we use an analogous parameterization:

$$\bar{\boldsymbol{\mu}}_t^{(n)} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \bar{\mathbf{x}}_t^{(n)} + \left(\sqrt{1-\alpha_{t-1}} - \sqrt{\frac{(1-\alpha_t)\alpha_{t-1}}{\alpha_t}} \right) \bar{\epsilon}_\theta(\bar{\mathbf{x}}_t^{(n)}, t), \tag{20}$$

where $\bar{\epsilon}_\theta(\bar{\mathbf{x}}_t^{(n)}, t)$ is calculated using conditional score-based sampling [12, 57, 32] as:

$$\bar{\epsilon}_\theta(\bar{\mathbf{x}}_t^{(n)}, t) = \epsilon_\theta(\bar{\mathbf{x}}_t^{(n)}, t) + \frac{\gamma}{2} \sqrt{1-\alpha_t} \nabla_{\mathbf{x}_t^{(n)}} \left\| f\left(\mathbf{x}_t^{(n-1)}, \boldsymbol{y}\right) - \bar{\mathbf{x}}_t^{(n)} \right\|_2^2. \tag{21}$$

Using these parameterizations, we obtain the practical objective optimized with the DDIM sampler:

$$\begin{aligned}
\mathcal{J} &= -\tilde{r}\left(\boldsymbol{y}, \hat{\mathbf{x}}_{0|t}^{(1)}, \ldots, \hat{\mathbf{x}}_{0|t}^{(n)}\right) + \lambda \left\| \bar{\boldsymbol{\mu}}_t^{(n)} - \boldsymbol{\mu}_{\boldsymbol{\phi}}\left(\mathbf{x}_t^{(n)}, t\right) \right\|_2^2 \\
&\leq -\tilde{r}\left(\boldsymbol{y}, \hat{\mathbf{x}}_{0|t}^{(1)}, \ldots, \hat{\mathbf{x}}_{0|t}^{(n)}\right) + \lambda \left\| a_t \beta \mathbf{u}_t^{(n-1)} + b_t(\bar{\epsilon}_\theta(\bar{\mathbf{x}}_t^{(n)}, t) - \epsilon_\theta(\bar{\mathbf{x}}_t^{(n)}, t)) \right\|_2^2 \\
&\leq -\tilde{r}\left(\boldsymbol{y}, \hat{\mathbf{x}}_{0|t}^{(1)}, \ldots, \hat{\mathbf{x}}_{0|t}^{(n)}\right) + \lambda a_t^2 \beta^2 \left\| \mathbf{u}_t^{(n-1)} \right\|_2^2 \\
&\quad + \lambda b_t^2 \left\| \epsilon_\theta(\bar{\mathbf{x}}_t^{(n)}, t) + \frac{\gamma}{2}\sqrt{1-\alpha_t} \nabla_{\mathbf{x}_t^{(n)}} \left\| f\left(\mathbf{x}_t^{(n-1)}, \boldsymbol{y}\right) - \bar{\mathbf{x}}_t^{(n)} \right\|_2^2 - \epsilon_\theta(\mathbf{x}_t^{(n)}, t) \right\|_2^2 . \qquad (22)
\end{aligned}$$

Here, we empirically omit the coefficient β^2 in the second term of Eq. 22, which improves performance and yields the final objective presented in Eq. 12.

C Pseudocode of SyncVC

Algorithm 1 provides a high-level overview of the practical implementation. Here, we note that controls are optimized sequentially in a greedy manner. Specifically, for each diffusion timestep t, instead of jointly optimizing $N-1$ control variables $\{\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(N-1)}\}$, we iterate over n and optimize each control variable $\mathbf{u}_t^{(n)}$ using the previously optimized variables $\{\mathbf{u}_t^{(1)}, \cdots, \mathbf{u}_t^{(n-1)}\}$. The optimization objective in Eq. 11 is factorized to $N-1$ terms (*i.e.*, each term in the summation) and $(n-1)$-th term is used as the optimization objective for $\mathbf{u}_t^{(n)}$. We adopt this setting since we empirically observe that greedy optimization yields better results.

Algorithm 1 SyncVC

Inputs: Observation $\boldsymbol{y}$, Noisy latent variables $\mathbf{x}_T^{(1)}, \ldots, \mathbf{x}_T^{(N)}$, Denoising kernel $p_{\boldsymbol{\phi}}$, Optimization step K, Hyperparameters γ, λ, β
for $t \leftarrow T, \cdots, 1$ **do**
  Initialize $\{\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(N-1)}\}$ as zero
  Calculate $\mathbf{x}_{t-1}^{(1)}$ using the denoising kernel $p_{\boldsymbol{\phi}}$
  for $n \leftarrow 2, \cdots, N$ **do**
    for $i \leftarrow 1, \cdots, K$ **do**
      Calculate the $(n-1)$-th term of the objective in Eq. 11
      Optimize $\mathbf{u}_t^{(n-1)}$ using the calculated optimization objective
    end for
    Calculate $\mathbf{x}_{t-1}^{(n)}$ using Eq. 6 with optimized $\mathbf{u}_t^{(n-1)}$ and $\mathbf{x}_t^{(n)}$
  end for
end for
Outputs: Sequence $\boldsymbol{X} = \{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}\}$

D Additional results

D.1 Wide image generation

Experimental details. We generate 2048×512 sized wide images for evaluation, with each patch of size 512^2. For our method, five distinct trajectories are sampled with an overlap of 128 pixels. The generated patches are then sequentially concatenated so that the patch from the n-th trajectory is placed on top of that from the $(n+1)$-th trajectory, yielding a single wide image. Each control variable $\mathbf{u}_t^{(n)}$ is optimized for five steps with a learning rate of 10^{-2} using the Adam optimizer [27]. Hyperparameters are set to $\beta = 1.0$, $\gamma = 2.5$, $\lambda = 2.0$. Across all methods, we use the pretrained Stable Diffusion v2.1-base [51][3] with 50 steps of DDIM [56] sampling and classifier-

[3]Accessed via https://huggingface.co/Manojb/stable-diffusion-2-1-base. CreativeML Open RAIL++-M License

free guidance [20] to ensure a fair comparison. For baselines, we run their official codes[4] [5] [6] [7]. In addition, we follow the baseline setup for noise initialization. Instead of independently sampling noise for each patch, a single wide latent noise map is first sampled from a Gaussian distribution, then cropped into the corresponding patch regions for each trajectory. We adopt the same process to ensure fairness in comparison.

To measure Intra-LPIPS [70], Intra-Style-Loss [15], χ^2-Histogram distance, and Histogram intersection, each wide image is cropped into four non-overlapping views of size 515^2, and the distances over all pairwise combinations (which is 6) are calculated. Note that the color histograms are computed in the HSV space. KID [6] score is calculated using randomly cropped 512^2 views from each wide image. These five metrics are measured across all prompts and then averaged. Reference images for KID measurement are constructed by generating 1,000 images per prompt using the pretrained Stable Diffusion v1.5 [51][8].

Alternative reward functions. We further demonstrate that the proposed method can accommodate different reward designs by considering a variant with an additional semantic guidance term. Specifically, we augment the reward in Eq. 8 with the CLIP similarity [47] between the generated patch $\mathbf{x}_0^{(n)}$ and the text prompt $\boldsymbol{y}$. The resulting reward is defined as

$$\tilde{r}\Big(\boldsymbol{y}, \mathbf{x}_0^{(1)}, \dots, \mathbf{x}_0^{(n)}\Big) = -\frac{\gamma}{2}\left\|\mathbf{M} \odot \left(f\Big(\mathbf{x}_0^{(n-1)}\Big) - \mathbf{x}_0^{(n)}\right)\right\|_2^2 + \lambda^{\text{clip}} \cdot \text{Sim}\left(f^{\text{img}}(\mathbf{x}_0^{(n)}), f^{\text{txt}}(\boldsymbol{y})\right), \tag{23}$$

where f^{img} and f^{txt} denote the image and text encoders of the pretrained CLIP model, respectively, $\text{Sim}(\cdot,\cdot)$ denotes cosine similarity, and λ^{clip} is a scalar hyperparameter. We select $\lambda^{\text{clip}} = 0.05$.

We apply the CLIP-based guidance term to all patches, *i.e.*, for $1 \le n \le N$. Since the first patch $\mathbf{x}_t^{(1)}$ is also guided by the CLIP-based term, we introduce an additional control variable $\mathbf{u}_t^{(0)}$ for this patch. Consequently, the ELBO includes an additional KL term associated with $\mathbf{u}_t^{(0)}$, yielding

$$\begin{aligned}\mathcal{L}(\boldsymbol{y}) := \ \mathbb{E}_q[r(\boldsymbol{y}, \boldsymbol{X})] &- \lambda \sum_{n=2}^{N}\sum_{t=1}^{T} D_{\text{KL}}\Big(q\Big(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}, \mathbf{x}_t^{(n-1)}, \mathbf{u}_t^{(n-1)}, \boldsymbol{y}\Big) \parallel p\Big(\mathbf{x}_{t-1}^{(n)} \mid \mathbf{x}_t^{(n)}\Big)\Big) \\ &- \lambda \sum_{t=1}^{T} D_{\text{KL}}\Big(q\Big(\mathbf{x}_{t-1}^{(1)} \mid \mathbf{x}_t^{(1)}, \mathbf{u}_t^{(0)}, \boldsymbol{y}\Big) \parallel p\Big(\mathbf{x}_{t-1}^{(1)} \mid \mathbf{x}_t^{(1)}\Big)\Big).\end{aligned} \tag{24}$$

We report the quantitative results using an alternative reward function from Eq. 23 in Table 4 (See "SyncVC*" column). As shown, this variant still outperforms the best baseline, MultiDiffusion [4], demonstrating that our framework can accommodate different reward choices while maintaining strong performance.

Table 4: Quantitative evaluation on wide image generation with an alternative reward choice. SyncVC* denotes our method using the reward function in Eq. 23. Our framework consistently outperforms the best baseline, MultiDiffusion [4], across all metrics, demonstrating its flexibility in incorporating different reward designs. KID [6] score is scaled by 10^3.

Method	MultiDiffusion [4]	**SyncVC (Ours, Eq. 8)**	**SyncVC* (Ours, Eq. 23)**
Intra-LPIPS [70] ↓	0.637	**0.592**	<u>0.594</u>
Intra-Style-Loss [15] ↓	58.46	**44.34**	<u>47.13</u>
χ^2-Histogram dist. ↓	1.211	**0.751**	<u>0.784</u>
Histogram intersect. ↑	0.549	**0.665**	<u>0.657</u>
KID [6] ↓	58.26	**52.07**	<u>53.43</u>

[4] MultiDiffusion: https://github.com/omerbt/MultiDiffusion

[5] SyncTweedies: https://github.com/KAIST-Visual-AI-Group/SyncTweedies, MIT License

[6] SyncSDE: https://github.com/hjl1013/SyncSDE

[7] StochSync: https://github.com/KAIST-Visual-AI-Group/StochSync, MIT License

[8] Accessed via https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5. CreativeML OpenRAIL M License

Details on style guidance. We incorporate style guidance by adding a style transfer loss between a style reference image and $\mathbf{x}_0^{(n)}$ $(1 \leq n \leq N)$ within the reward function of Eq. 8. Since the first patch $\mathbf{x}_t^{(1)}$ is also subject to the style guidance, the ELBO for stylized wide-image generation is identically derived as Eq. 24.

Style references used for experiments are borrowed from the internet[9]. Specifically, following Gatys et al. [15], the style transfer loss is defined as a weighted sum of a content loss and a style loss. Let $\mathbf{F}(\mathbf{I}) \in \mathbb{R}^{C \times M}$ denote the features of an image $\mathbf{I}$ extracted using the pretrained VGG network [54], where C is the total number of feature maps and M is the spatial resolution of each feature map. The content loss is defined as the squared error between features of two images:

$$\mathcal{L}_{\text{content}}(\mathbf{I}_1, \mathbf{I}_2) = \frac{1}{2} \|\mathbf{F}(\mathbf{I}_1) - \mathbf{F}(\mathbf{I}_2)\|_F^2 \tag{25}$$

where $\mathbf{I}_1$ denotes the content image and $\mathbf{I}_2$ is the stylized image. In the stylized wide image generation scenario, the content image is defined as $\mathbf{x}_0^{(n)}$ obtained without optimizing controls. For the style representation of image $\mathbf{I}$, we use the Gram matrix $\mathbf{G}(\mathbf{I}) \in \mathbb{R}^{C \times C}$:

$$\mathbf{G}(\mathbf{I})[i, j] = \sum_k \text{Vec}(\mathbf{F}_{ik}(\mathbf{I}))\text{Vec}(\mathbf{F}_{jk}(\mathbf{I})) \quad \text{for} \quad 1 \leq i, j \leq C, \tag{26}$$

where $\text{Vec}(\cdot)$ stands for the vectorization operation of a given matrix. The style loss is then defined as

$$\mathcal{L}_{\text{style}}(\mathbf{I}_2, \mathbf{I}_3) = \frac{1}{4C^2M^2} \sum_{i,j} \left(G_{ij}(\mathbf{I}_2) - G_{ij}(\mathbf{I}_3)\right)^2, \tag{27}$$

where $\mathbf{I}_3$ denotes the style reference image. In practice, we use features from multiple layers of VGG network to calculate style loss and use the averaged value. The style transfer loss is finally given by

$$\mathcal{L}_{\text{style-transfer}} = w_{\text{content}} \mathcal{L}_{\text{content}} + w_{\text{style}} \mathcal{L}_{\text{style}}. \tag{28}$$

We choose $w_{\text{content}} = 1.0$ and $w_{\text{style}} = 10^{-4}$.

Results on extreme scenarios. For the sample shown in Figure 6 of the main paper, we additionally visualize the full figure including the results of SyncTweedies [26] and SyncSDE [31] in Figure 8. This corresponds to the wide image generation setting with a small overlap of only 16 pixels (over the patch width of 512 pixels). As shown, baselines exhibit noticeable color inconsistencies between patches, while the proposed method produces consistent outputs.

Additional qualitative results. We show additional qualitative comparisons with baselines in Figure 9, and additional results of our method in Figure 10. Furthermore, we show that the proposed method can be also applied to recent generative model with stronger priors that synthesize high-resolution images. Specifically, we use the pretrained SANA model [60][10] [11] to generate wide images with the resolution of 4096×1024 and 8192×2048, and visualize it in Figure 11 and 12.

[9] https://github.com/gordicaleksa/pytorch-neural-style-transfer, MIT license

[10] Accessed via https://huggingface.co/Efficient-Large-Model/Sana_1600M_1024px_diffusers, NVIDIA License

[11] Accessed via https://huggingface.co/Efficient-Large-Model/Sana_1600M_2Kpx_BF16_diffusers, NVIDIA License

Figure 8: **Our method demonstrates superior performance in wide image generation under a small-overlap setting**, maintaining strong style and color consistency across the horizontal axis. All baseline methods exhibit significant color changes.

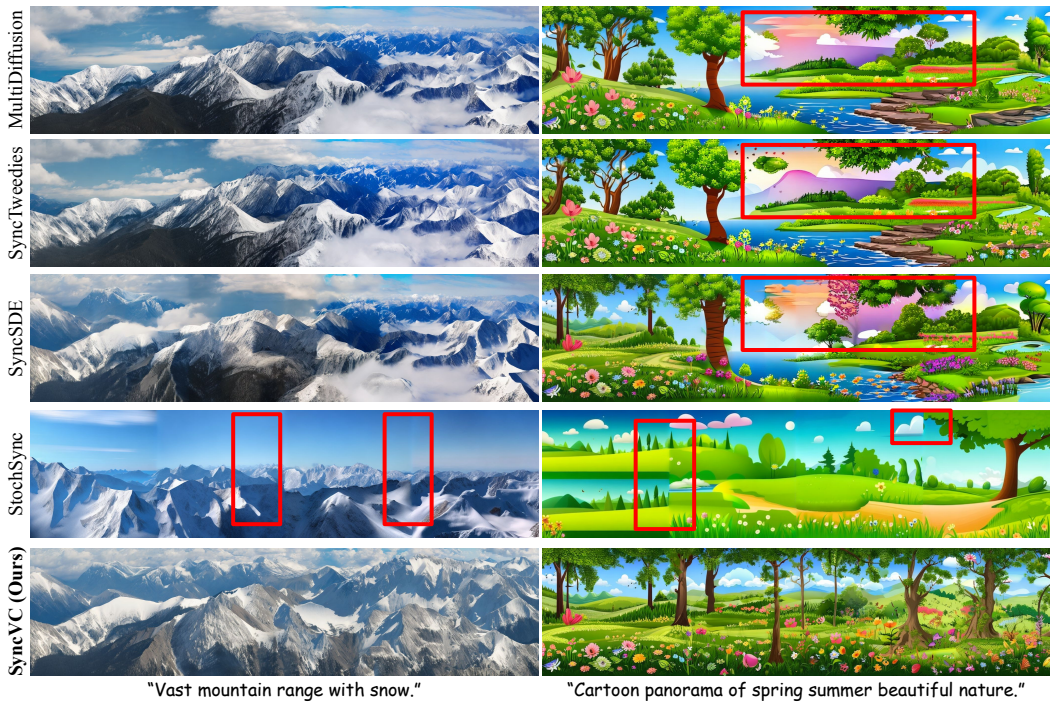


Figure 9: **Our method shows superior performance in wide image generation.** (Left) SyncVC maintains a unified color and style, while baselines suffer from varying mountain and sky colors, or discontinuities (see bounding box). (Right) Our method generates cartoon-like panorama with consistent styles of tree and flowers, while baselines result in artifacts with inconsistent colors or discontinuities (see bounding box).

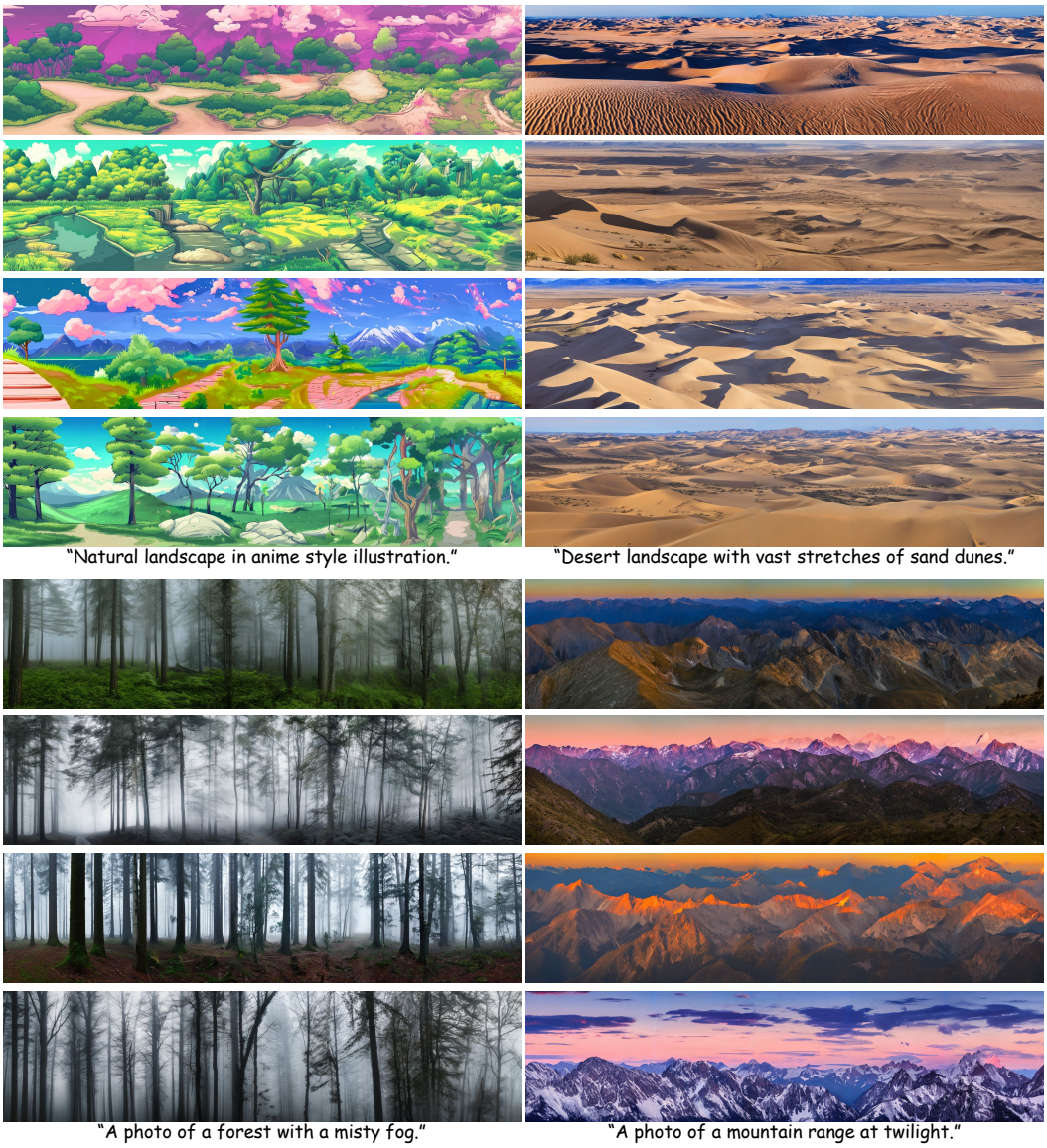


Figure 10: **Our method generates high-quality wide images conditioned on diverse text prompts.** We present multiple wide image samples generated using the pretrained Stable Diffusion [51] for various text prompts, all exhibiting strong style consistency.

Figure 11: **Our method can also synthesize high-resolution wide image when combined with SANA model [60].** We visualize various wide images at the resolution of 4096×1024 using the pretrained SANA model, where each generated patch has a resolution of 1024^2.

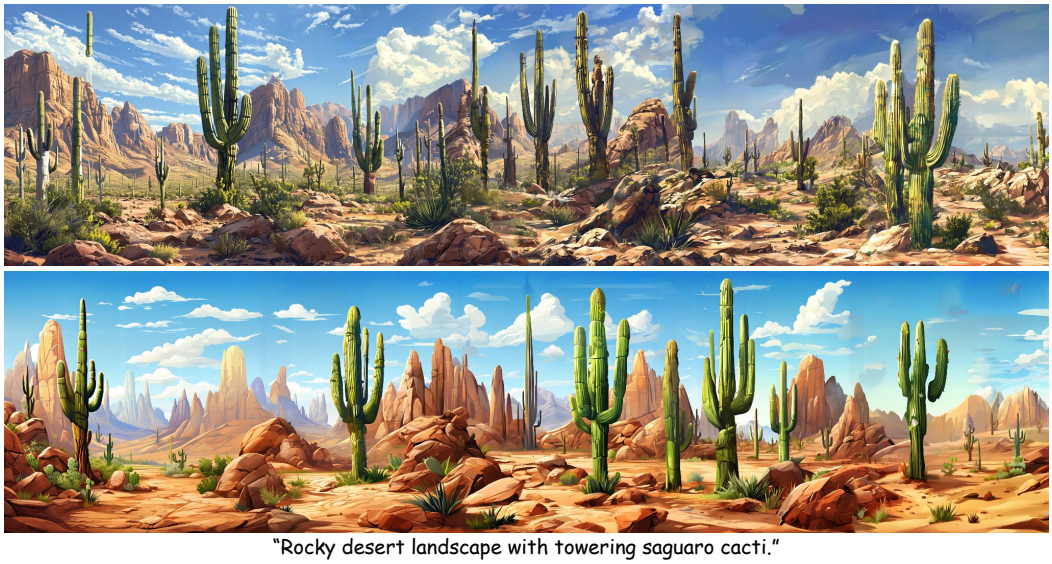


Figure 12: **Our method is even capable of generating 8192×2048-sized wide image.** We use the pretrained SANA model [60] that generates patches at a resolution of 2048^2, and extend it along the horizontal axis using our method to generate ultra-high-resolution images.

D.2 Optical illusion generation

Experimental details. We generate images of 256^2 resolution for all methods. Each image is associated with two trajectories, corresponding to two different (transformation, prompt) pairs. Specifically, we use the pretrained two-stage DeepFloyd-IF model [1], where the first and second stage models are IF-I-M-v1.0[12] and IF-II-M-v1.0[13], respectively. For baselines, we run their official implementation[14] for experiments. For our method, guidance is applied only during the first-stage sampling. We use 30 steps of DDIM [56] reverse process with classifier-free guidance [20], and the noise scale parameter for second stage of DeepFloyd-IF model is fixed to 50 across all methods. Each control variable $\mathbf{u}_t^{(n)}$ is optimized for five iterations with a learning rate of 10^{-2} using the Adam optimizer [27]. We use $\beta = 2.0$, $\gamma = 0.05$, and $\lambda = 0.2$ as default hyperparameters.

For evaluation, two views are sampled from each generated image. The 2nd view is obtained directly from the second trajectory, while the 1st view is constructed by applying the inverse illusion transformation to the second view (*e.g.*, a counterclockwise rotation for a clockwise illusion transformation). Note that each view is associated with its corresponding prompt. FID [18] and KID [6] values are computed between the images of each view and the reference images for each (transformation, prompt) pair, then averaged. We generate the reference images by synthesizing 1,000 images per prompt using the pretrained Stable Diffusion v1.5 [51]. Furthermore, MUSIQ [25] is computed for both views, and the scores are averaged over all generated images.

Additional qualitative results. In Figure 13, we show the results of our method on two additional illusion transformations. SyncVC generates high-resolution images that successfully encode both semantics under the illusion transformation.

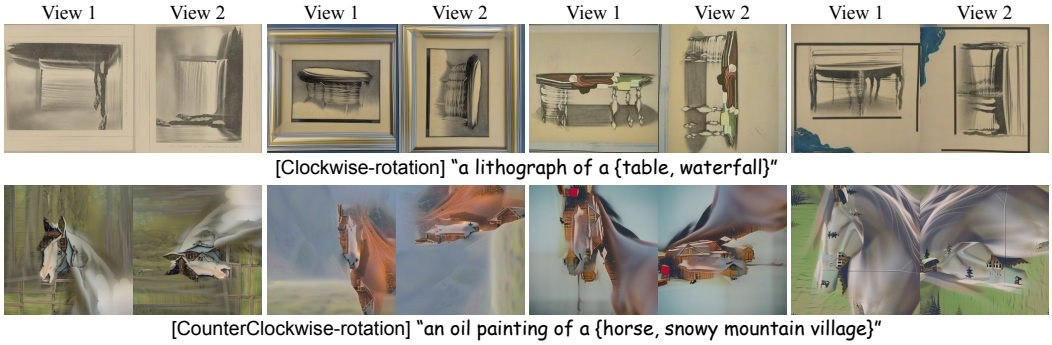


Figure 13: **Our method shows superior performance on the optical illusion generation task** by clearly incorporating two semantics specified by different text prompts. (Row 1) The generated image can be viewed as both a table and a waterfall under clockwise rotation. (Row 2) Each view encodes both a horse and a snowy mountain village under counterclockwise rotation.

Visualization of controls. To provide intuition on the role of controls, we visualize the optimized control variables in Figure 14. At early timesteps (large t), the controls focus on shaping the overall semantic of an image to satisfy the optimization objective, whereas at later timesteps (small t), they progressively manipulate fine-grained details.

D.3 Text-guided 3D mesh texturing

Experimental details. We use the pretrained Stable Diffusion v1.5 [51] with the pretrained depth-conditioned ControlNet [69][15] for synchronization-based methods [26, 31, 64] (including ours), and Stable Diffusion v2-depth model[16] for task-specific methods [68, 49] following their original configuration. Regarding the viewpoint setting, we fix the elevation to 15° and uniformly sample eight azimuth angles from $[0^\circ, 360^\circ)$, resulting in eight diffusion trajectories. We use 8 DDIM [56] steps,

[12] Accessed via https://huggingface.co/DeepFloyd/IF-I-M-v1.0, DeepFloyd IF License Agreement
[13] Accessed via https://huggingface.co/DeepFloyd/IF-II-M-v1.0, DeepFloyd IF License Agreement
[14] Anagram-MTL: https://github.com/Pixtella/Anagram-MTL, Apache-2.0 License
[15] Accessed via https://huggingface.co/lllyasviel/control_v11f1p_sd15_depth, The CreativeML OpenRAIL M License
[16] Accessed via https://huggingface.co/sd2-community/stable-diffusion-2-depth, CreativeML Open RAIL++-M License

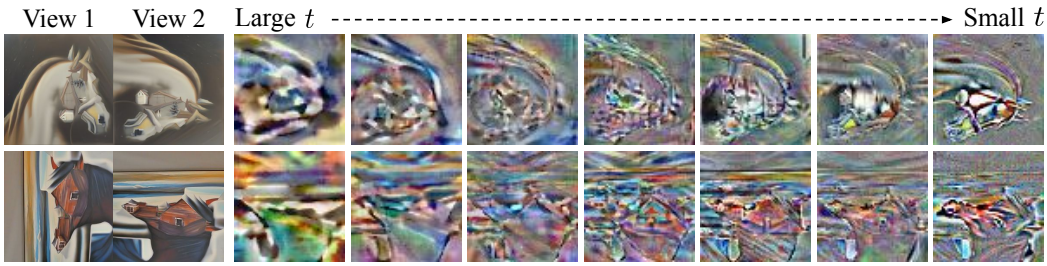


Figure 14: **Visualization of optimized controls.** The controls first capture coarse and low-level structures, then refine high-level features. We use a text prompt of "an oil painting of a horse" and "an oil painting of a snowy mountain village", with clockwise rotation.

with the resolution of 768^2 for each patch. Meanwhile, we follow the default viewpoint sampling and diffusion sampling configurations for baselines and run their official codes[17] [18]. For all methods, we prepend the prompt with the phrase "Best quality, extremely detailed" and use classifier-free guidance [20] with the negative prompt "oversmoothed, blurry, depth of field, out of focus, low quality, bloom, glowing effect". To synthesize the texture map, we optimize it by minimizing the distance between the rendered view of the texture-projected mesh and the generated patch at each viewpoint using the SGD optimizer. Source meshes used for experiments are borrowed from the Objaverse dataset [11][19]. Following prior works [40, 26, 31], we apply Voronoi diagram-augmented filling [2] and a modified self-attention mechanism in the noise prediction network. The latent texture map resolution is set to 1536^2, and the RGB texture map resolution is 1024^2. Each control variable $\mathbf{u}_t^{(n)}$ is optimized for three iterations with a learning rate of 10^{-2} using the Adam optimizer [27]. We use $\beta = 1.0$, $\gamma = 0.1$, and $\lambda = 2.0$ as default hyperparameter.

For evaluation, we render the textured meshes from 10 different viewpoints using PyTorch3D renderer [48][20]. Eight views have an elevation of $0°$ and azimuths uniformly sampled from $[0°, 360°)$, while two additional views have an elevation of $30°$ with azimuths of $0°$ and $180°$, which corresponds to front and back view, respectively. FID [18] and KID [6] scores are calculated between the rendered images and the reference sets for each (mesh, prompt) pair, then averaged. For each (mesh, prompt) pair, we render depth maps from same 10 viewpoints that are used for evaluation and generate 50 images per each depth map using the pretrained depth-conditioned ControlNet, resulting in 500 reference images. CLIP-S [47] is measured by averaging the cosine similarity between each of the 10 rendered views and the corresponding prompt for every (mesh, prompt) pair. For qualitative visualization in Figure 5, we use Nvdiffrast [29][21] rasterizer for more sophisticated rasterization.

Additional qualitative results. We show additional qualitative results of SyncVC on text-guided 3D mesh texturing task in Figure 15. As shown, our method synthesizes textures that are not only realistic but also rich in fine-grained details.

[17] TexPainter: https://github.com/Quantuman134/TexPainter, MIT License
[18] TEXTure: https://github.com/TEXTurePaper/TEXTurePaper, MIT License
[19] Objaverse: https://huggingface.co/datasets/allenai/objaverse, ODC-By v1.0 License
[20] PyTorch3D: https://github.com/facebookresearch/pytorch3d, BSD License
[21] Nvdiffrast: https://github.com/NVlabs/nvdiffrast, Nvidia Source Code License (1-Way Commercial)

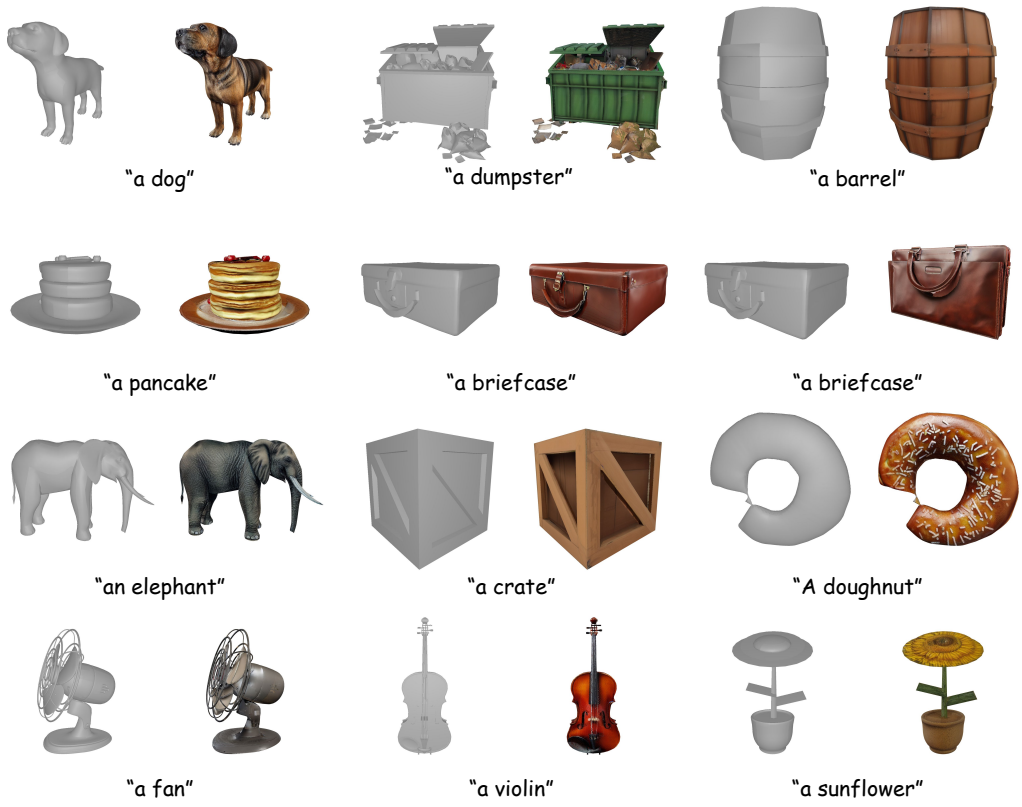


Figure 15: [Best viewed when magnified.] **Our method generates artifact-free and realistic textures for diverse 3D meshes.**

D.4 Discussion on computational cost

We measure the required runtime of each method to generate a single image for wide image generation and optical illusion generation task. Runtimes are measured using a single NVIDIA A6000 GPU with the official implementation of each method. Table 5 and 6 summarize the results. Since our method involves an optimization while the baselines do not, it incurs a longer runtime. Nevertheless, as demonstrated in Table 1 and 2, SyncVC achieves stronger performance while maintaining practical usability.

Table 5: Quantitative runtime measurement for wide image generation task.

Method	MultiDiffusion [4]	SyncTweedies [26]	SyncSDE [31]	StochSync [64]	**SyncVC (Ours)**
Runtime (s/image)	68.70	73.98	17.51	42.37	232.41

Table 6: Quantitative runtime measurement for optical illusion generation task.

Method	SyncTweedies [26]	SyncSDE [31]	Anagram-MTL [61]	**SyncVC (Ours)**
Runtime (s/image)	6.04	6.25	12.38	33.01

E Societal impacts

Our method enables collaborative generation in diverse and challenging scenarios, making it applicable to various visual generation tasks that require globally consistent outputs. It may improve the practicality of generative models in real-world applications such as content creation and design. However, it may also inherit the pretrained generative model's potential limitations, including the generation of harmful or unethical content.