∞-Brush: Controllable Large Image Synthesis with Diffusion Models in Infinite Dimensions

Minh-Quan Le*, Alexandros Graikos*, Srikar Yellapragada,
Rajarsi Gupta, Joel Saltz, and Dimitris Samaras

Stony Brook University
{mile,agraikos,myellapragad,samaras}@cs.stonybrook.edu

Abstract. Synthesizing high-resolution images from intricate, domain-specific information remains a significant challenge in generative modeling, particularly for applications in large-image domains such as digital histopathology and remote sensing. Existing methods face critical limitations: conditional diffusion models in pixel or latent space cannot exceed the resolution on which they were trained without losing fidelity, and computational demands increase significantly for larger image sizes. Patch-based methods offer computational efficiency but fail to capture long-range spatial relationships due to their overreliance on local information. In this paper, we introduce a novel conditional diffusion model in infinite dimensions, ∞-Brush for controllable large image synthesis. We propose a cross-attention neural operator to enable conditioning in function space. Our model overcomes the constraints of traditional finite-dimensional diffusion models and patch-based methods, offering scalability and superior capability in preserving global image structures while maintaining fine details. To our best knowledge, ∞-Brush is the first conditional diffusion model in function space, that can controllably synthesize images at arbitrary resolutions of up to 4096 × 4096 pixels. The code is available at https://github.com/cvlab-stonybrook/infinity-brush.

Keywords: Diffusion models · Function space models · Image synthesis

1 Introduction

Diffusion models are powerful generative models that have achieved remarkable success in synthesizing diverse and complex data, such as images and audio [23, 24]. Despite their success, it is still difficult to generate high-resolution images, especially when it is necessary to condition them on intricate, domain-specific information. Practical histopathology and satellite imagery applications in medical diagnostics, environmental monitoring, and beyond require precise and controllable very large image synthesis – well beyond 1024 × 1024 pixels, which is impractical with the current state-of-the-art (SoTA) models such as Stable Diffusion-XL (SDXL) [26].

* Equal contribution

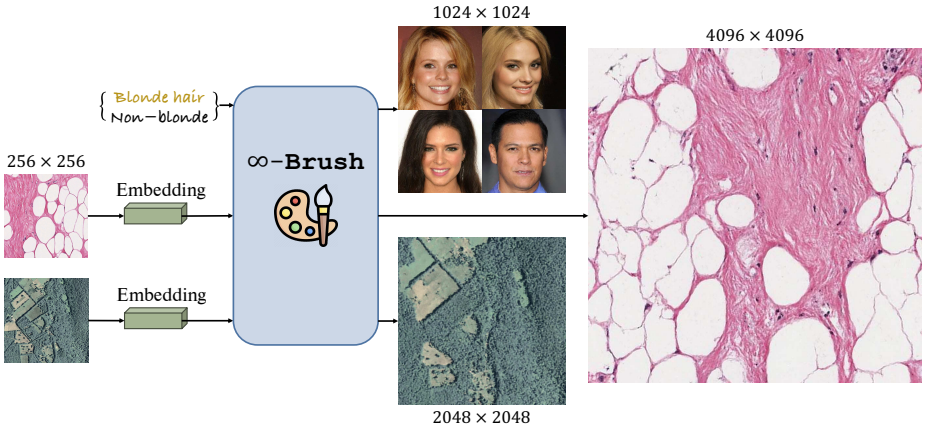

Fig. 1: ∞-Brush is able to controllably generate images at arbitrary resolutions of up to 4096 × 4096, conditioned on any available auxiliary information about the images.

The SoTA methods for controllable large image generation still exhibit significant limitations. We can distinguish two broad categories; the first set of approaches directly employs conditional diffusion models in finite latent or pixel space and is inherently limited by its design to generate images only at the resolution on which it was trained. Examples are SDXL [26] and Matryoshka Diffusion [11] which can produce images at a resolution of up to 1024 × 1024 pixels. While impressive, such methods cannot generate images at resolutions higher than those they were trained in, without a loss in quality or fidelity. Additionally, as the resolution increases, the computational resources required to train and run these models scale quadratically, making the process increasingly inefficient for larger image sizes.

The second strategy, introduced by MultiDiffusion [1] and adapted by Graikos *et al*. [10], involves a patch-based method that splits large image generation into smaller segments. This technique involves training "local" diffusion models on the patches from large images and performing large image synthesis using an outpainting algorithm. While this approach is computationally more efficient and produces sufficiently realistic larger images, it falls short of capturing long-range spatial dependencies (as discussed in the supplementary). This limitation stems from the heavy reliance on local information, as the generation of each patch is predominantly influenced by the local conditioning and not affected by the information of far away patches.

The previously mentioned methods operate in finite image or latent space and cannot significantly exceed the training image sizes during generation. This makes training the models directly on the entire large images a necessity, leading to insurmountable computational costs. Recently, Bond-Taylor *et al*. [2] have demonstrated that by representing images as functions in Hilbert space $\mathcal{H}$ they can synthesize arbitrarily-large images while training on fixed-size inputs. How-

ever, their models can not be conditioned for controllable image generation, which is necessary to efficiently utilize the model in downstream applications, such as data augmentation.

In this work, we propose a novel conditional diffusion model, ∞-Brush, for controllable large image synthesis in function space. Our model learns to synthesize images as continuous functions at arbitrarily sampled coordinates, enabling the generation of images at any desired resolution.

To condition the infinite-dimensional diffusion models we propose a cross-attention neural operator in function space. This operator is necessary since naively trying to condition the diffusion process using existing cross-attention operations on fixed pixel grids is inadequate. Similar to how synthesizing images in finite dimensions cannot capture long-range and intricate details, applying cross-attention on a fixed grid will result in the loss of fine detail. In our experiments, we compare our proposed neural operator to conventional cross-attention and show how we can better capture fine details across all image scales.

The ∞-Diff [2] model samples 25% of the image pixels during training. Directly applying it to large images is infeasible due to memory constraints. Instead, we show that we can train our single, conditional model on much smaller subsets of pixels (0.4%) from each large image without loss in generation quality. This enables us to apply the infinite-dimensional diffusion model on large image datasets, where images can be up to 4096×4096 pixels.

In our experiments, we first demonstrate our infinite-dimensional conditioning mechanism, the cross-attention neural operator, by performing conditional image generation on CelebA-HQ [18]. We then showcase large image generation, where we train models on histopathology and satellite image datasets and demonstrate how our method ∞-Brush outperforms patch-based generation [10] in terms of maintaining global structure without sacrificing local fidelity. In these large image domains, the 4096×4096 resolution that we achieve is not attainable by any existing model [1, 10, 26]. Figure 1 illustrates that ∞-Brush is able to controllably synthesize images at arbitrary resolutions of up to 4096×4096.

In summary, our contributions are as follows:

- We propose a cross-attention neural operator in function space. This operator allows for the incorporation of external information during image generation.
- We use this operator to build a conditional denoiser in function space as part of ∞-Brush, the first conditional diffusion model in function space.
- We ensure tractable training of our model on very large images by only training on 0.4% subsets of pixels while inferring at arbitrary resolutions.
- We show how our method generates images at the hencetofore infeasible size of 4096×4096 pixels while maintaining both global structure and fine details.

2 Related Work

Controllable Generation with Diffusion Models. Diffusion models [14, 21, 32] synthesize data by reversing a diffusion process. Latent Diffusion Models (LDMs) [30] operate in a lower-dimensional latent space rather than pixel space,

significantly reducing the computational load and enabling the generation of high-quality images. Controllable generation is achieved by conditioning on desired attributes, such as class-conditioning [24], gradient-based guidance [8], and classifier-free guidance [15].

Large Image Generation. SDXL [26] makes a step towards large image generation with its ability to generate higher-resolution images. However, controllable generation with SDXL cannot scale to large images because it is constrained to synthesize images only at the resolution on which it was trained (1024×1024), leading to a quadratic increase in computational demands with resolution. Our diffusion model, ∞-`Brush`, learns to controllably synthesize images in function space which enables us to generate large images at any desired resolution of up to 4096×4096 by only training on subsets of 65536 pixels.

The patch-based approach for controllable generation, exemplified by MultiDiffusion [1] and adapted in [10], efficiently generates large images by synthesizing individual patches that are later combined. Despite its computational efficiency and ability to produce realistic images, this method struggles to capture long-range spatial dependencies due to its use of only local information. In contrast, our model operates on the **entirety** of the image, as represented by a function, maintaining large-scale structures and long-range dependencies.

Diffusion Models in Infinite Dimensions. Kerrigan *et al*. [19] introduced the concept of applying diffusion models to functional data, pioneering the idea that generative models can operate beyond the confines of finite-dimensional spaces. Building on the ideas of infinite-dimensional diffusion, Lim *et al*. [22] and ∞-Diff [2] specifically address the generation of images represented in function space. However, the methods cannot be conditioned for controllable image generation. To the best of our knowledge, our ∞-`Brush` with a novel cross-attention neural operator is the first conditional diffusion model in infinite dimensions designed for controllable large image synthesis.

3 Preliminaries

3.1 Notation and Data

Let $(\mathcal{X}, \mathcal{A}, \mu)$ be a measure space where $\mathcal{X} \subseteq \mathbb{R}^{d_x}$, $\mathcal{A}$ is a σ-algebra on the set $\mathcal{X}$ and μ is a measure on $(\mathcal{X}, \mathcal{A})$. Let $\mathcal{H}$ be a separable Hilbert space over the domain $\mathcal{X}$, equipped with its Borel σ-algebra $\mathcal{B}(\mathcal{H})$. For simplicity, we consider the case where $\mathcal{H}$ is the space of L^2 functions $\mathcal{H} = L^2(\mathcal{X}, \mu)$, which is equipped with its inner product $\langle \mathbf{f}, \mathbf{g} \rangle_{L^2(\mathcal{X},\mu)} = \int_{\mathcal{X}} \mathbf{f}\mathbf{g} \mathrm{d}\mu$. It is worth noting that our method is agnostic to the choice of $\mathcal{H}$ and can be applied to other spaces.

Assuming that we have a dataset of the form $\mathcal{D} = \{(\mathbf{u}_k, \mathbf{e}_k)\}_{1 \le k \le D}$, where each $\mathbf{u}_j \in \mathcal{H}$ is an i.i.d. draw from an unknown probability measure $\mathbb{Q}_{\text{data}}$ on $\mathcal{H}$ and $\mathbf{e}_j$ is a control component of the corresponding function $\mathbf{u}_j$. In our experiment settings, $\mathbf{e}_j$ can be a label or an embedding vector (from vision-language or self-supervised models) with finite dimensions.

In practice, it is difficult to represent the function directly and instead, for an input function $\mathbf{u}_j$, we discretize it on the mesh $\mathbf{x}_j = \{\mathbf{x}_j^{(i)}\}_{1 \le i \le N} \subset \mathcal{X}$,

which is a discrete subset on $\mathcal{X}$ with corresponding discretized observations $\{\mathbf{u}_j(\mathbf{x}_j^{(i)})\}_{1\leq i\leq N}$, being the output of function $\mathbf{u}_j$ at the i-th observation point.

3.2 Gaussian Measures on Hilbert Spaces

Let $\mathbb{Q}$ be a probability measure on $(\mathcal{H}, \mathcal{B}(\mathcal{H}))$. If $\mathbb{Q}$ is Gaussian, then there exists a mean element $\mathbf{m} \in \mathcal{H}$ and a covariance operator $\mathbf{C} : \mathcal{H} \to \mathcal{H}$, such that

$$\int_{\mathcal{H}} \langle \mathbf{u}, \mathbf{x} \rangle \mathbb{Q}(\mathrm{d}\mathbf{x}) = \langle \mathbf{m}, \mathbf{u} \rangle, \quad \forall \mathbf{u} \in \mathcal{H}, \tag{1}$$

$$\int_{\mathcal{H}} \langle \mathbf{u}_1, \mathbf{x} - \mathbf{m} \rangle \langle \mathbf{u}_2, \mathbf{x} - \mathbf{m} \rangle \mathbb{Q}(\mathrm{d}\mathbf{x}) = \langle \mathbf{C}\mathbf{u}_1, \mathbf{u}_2 \rangle, \quad \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathcal{H}. \tag{2}$$

The covariance operator $\mathbf{C}$ is symmetric, positive semi-definite, compact, and has finite trace $\mathrm{Tr}(\mathbf{C}) < +\infty$. Conversely, let C be a positive, symmetric, trace class operator in $\mathcal{H}$ and let $\mathbf{m} \in \mathcal{H}$, then there exists a Gaussian measure in $\mathcal{H}$ with mean $\mathbf{m}$ and covariance $\mathbf{C}$ [6]. From now on, we will denote $\mathbb{Q} = \mathcal{N}(\mathbf{m}, \mathbf{C})$ for such a Gaussian measure.

3.3 Diffusion Models in Function Space

Here we briefly describe diffusion probabilistic models in function space $\mathcal{H}$ [2,19, 22], which is constructed similarly to that of DDPMs [14]. Note that the key difference is that diffusion models in function space operate in infinite dimensions.
Forward process. The forward process of a diffusion model in function space is defined as a discrete-time Markov chain that incrementally perturbs probability measure $\mathbb{Q}_{\text{data}}$ towards a Gaussian measure $\mathcal{N}(\mathbf{m}, \mathbf{C})$ with a zero mean and a specified covariance operator $\mathbf{C}$. It is a time-indexed process where each step $\mathbf{u}_t$ is obtained by applying a transformation to the previous step $\mathbf{u}_{t-1}$, which involves a scaling factor $\sqrt{1-\beta_t}\mathbf{u}_{t-1}$ related to the variance schedule β, and adding scaled Gaussian noise $\sqrt{\beta_t}\boldsymbol{\xi}_t$ with $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$:

$$\mathbf{u}_t = \sqrt{1-\beta_t}\mathbf{u}_{t-1} + \sqrt{\beta_t}\boldsymbol{\xi}_t \qquad t = 1, 2, \ldots, T. \tag{3}$$

Similar to diffusion models in finite dimensions, the forward process in function space also admits sampling $\mathbf{u}_t$ at an arbitrary timestep t in closed form. For $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_t)$, we have:

$$\mathbb{Q}\left(\mathbf{u}_t | \mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_t; \sqrt{\bar{\alpha}_t}\mathbf{u}_0, (1-\bar{\alpha}_t)\mathbf{C}\right). \tag{4}$$

Reverse process. The reverse process in the diffusion model iteratively denoises from the Gaussian measure $\mathcal{N}(\mathbf{m}, \mathbf{C})$ back to the probability measure $\mathbb{Q}_0 = \mathbb{Q}_{\text{data}}$. This is achieved by sampling from the reverse-time transition measures $\mathbb{Q}\left(\mathbf{u}_{t-1} | \mathbf{u}_t\right)$, approximated by a Gaussian measure with parameters θ due to the intractable normalization constant of the Bayes' rule:

$$\mathbb{P}_\theta\left(\mathbf{u}_{t-1} | \mathbf{u}_t\right) = \mathcal{N}\left(\mathbf{u}_{t-1}; \mathbf{m}_\theta(\mathbf{u}_t, t), \mathbf{C}_\theta(\mathbf{u}_t, t)\right) \tag{5}$$

Likewise, we are able to derive a closed-form representation of the forward process posteriors, which are tractable when conditioned on $\mathbf{u}_0$:

$$\mathbb{Q}\left(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_{t-1}; \tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0), \tilde{\beta}_t \mathbf{C}\right), \tag{6}$$

where $\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{u}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{u}_t$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$.
Training objective. Similar to DDPM in finite dimensions, the reparameterization is used to achieve better training, which results in a simplified loss:

$$\mathcal{L}_{\text{simple}} = \left|\left|\mathbf{C}^{-1/2}\left(\boldsymbol{\xi}_t - \boldsymbol{\xi}_\theta(\mathbf{u}_t, t)\right)\right|\right|^2_{\mathcal{H}}. \tag{7}$$

3.4 Neural Operators

Neural operators [4,12,20] are a type of neural network tailored to learn mappings between infinite-dimensional function spaces. In the context of diffusion models in infinite dimensions, a denoiser is parameterized by a neural operator $\mathcal{G}_\theta : \mathcal{U}^* \to \mathcal{U}$, which learns to map from noisy function space $\mathcal{U}^*$ to denoised function space $\mathcal{U}$. With $\mathbf{u} \in \mathcal{U}^*$ and $\mathbf{s} \in \mathcal{U}$, we access their pointwise evaluations by discretizing them on the mesh $\mathbf{x} = \{\mathbf{x}^{(i)}\}_{1\le i\le N} \subset \mathcal{X}$. Neural operators include multiple operator layers akin to those in a finite-dimensional neural network $\mathbf{v}_0 \mapsto \mathbf{v}_1 \mapsto \cdots \mapsto \mathbf{v}_L$, where layer $\mathbf{v}_l \mapsto \mathbf{v}_{l+1}$ is built upon a local linear operator, a non-local integral kernel operator, and a bias function:

$$\mathbf{v}_{l+1}(\mathbf{x}^{(i)}) = \sigma_{l+1}\left(W_l \mathbf{v}_l(\mathbf{x}^{(i)}) + (\mathcal{K}_l(\mathbf{u};\phi)\mathbf{v}_l)(\mathbf{x}^{(i)}) + b_l(\mathbf{x}^{(i)})\right), \tag{8}$$

with $\mathcal{K}_l(\mathbf{u};\phi)$ being an integral kernel operator aggregating information spatially.

4 The Proposed Method

We propose a novel conditional diffusion model in function space $\mathcal{H}$. Based on the background provided, we now formulate the forward and reverse process and the training objective of our conditional diffusion model in infinite dimensions. Furthermore, we present a novel architecture to parameterize the denoising process with a conditional denoiser equipped with cross-attention neural operators.

4.1 Conditional Diffusion Models in Function Space

In the context of image generation, we discretize the function $\mathbf{u}_j$ on the mesh $\mathbf{x}_j = \{\mathbf{x}_j^{(i)}\}_{1\le i\le N} \subset \mathcal{X}$ by sampling N coordinates of each image, which results in non-smooth input space. To achieve a smoother function representation, a smoothing operator [16, 28] $\mathbf{A} : \mathcal{H} \to \mathcal{H}$, *e.g.* a truncated Gaussian kernel, is applied to approximate the rough inputs within the function space $\mathcal{H}$.
Forward process. The forward process of our conditional diffusion model in infinite dimensions is equivalent to that of an unconditional diffusion model in

function space, which gradually perturbs the probability measure $\mathbb{Q}_0 = \mathbb{Q}_{\text{data}}$ towards a Gaussian measure $\mathcal{N}(\mathbf{m}, \mathbf{C})$ and enables sampling at any arbitrary timestep t with $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_t)$:

$$\mathbb{Q}\left(\mathbf{u}_t|\mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_t; \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0, (1-\bar{\alpha}_t)\mathbf{A}\mathbf{C}\mathbf{A}^T\right). \tag{9}$$

Reverse process. We use a variational approach to approximate posterior measures with a variational family of measures on $\mathcal{H}$ and incorporate the conditional embedding $\mathbf{e}$ to control the generation process. We model the underlying posterior measure $\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t)$ with a conditional Gaussian measure:

$$\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{e}) = \mathcal{N}\left(\mathbf{u}_{t-1}; \mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t), \mathbf{A}\mathbf{C}_\theta(\mathbf{u}_t, \mathbf{e}, t)\mathbf{A}^T\right). \tag{10}$$

Proposition 1 (Learning Objective). *The cross-entropy of conditional diffusion models in function space has a variational upper bound of*

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{\mathbb{Q}} \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{e}) \leq \mathbb{E}_{\mathbb{Q}}\Bigg[\underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_T))}_{\mathcal{L}_T} \underbrace{- \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1, \mathbf{e})}_{\mathcal{L}_0} + \sum_{t=2}^{T} \underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{e})}_{\mathcal{L}_{t-1}}\Bigg]. \tag{11}$$

Proof. Please refer to the Appendix A for the full proof. □

To compute the KL divergence between probability measures $\text{KL}(\mathbb{Q} \parallel \mathbb{P})$, we need to utilize a measure-theoretic definition of the KL divergence, which is stated in the following lemmas [6].

Lemma 1 (Measure Equivalence - The Feldman-Hájek Theorem). *Let $\mathbb{Q} = \mathcal{N}(\mathbf{m}_1, \mathbf{C}_1)$ and $\mathbb{P} = \mathcal{N}(\mathbf{m}_2, \mathbf{C}_2)$ be Gaussian measures on $\mathcal{H}$. They are equivalent if and only if $(i) : \mathbf{C}_1^{1/2}(\mathcal{H}) = \mathbf{C}_2^{1/2}(\mathcal{H}) = \mathcal{H}_0$, $(ii) : \mathbf{m}_1 - \mathbf{m}_2 \in \mathcal{H}_0$, and (iii) : The operator $(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})^* - \mathbf{I}$ is a Hilbert-Schmidt operator on the closure $\overline{\mathcal{H}_0}$.*

Lemma 2 (The Radon-Nikodym Derivative). *Let $\mathbb{Q} = \mathcal{N}(\mathbf{m}_1, \mathbf{C}_1)$ and $\mathbb{P} = \mathcal{N}(\mathbf{m}_2, \mathbf{C}_2)$ be Gaussian measures on $\mathcal{H}$. If $\mathbb{P}$ and $\mathbb{Q}$ are equivalent and $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}$, then $\mathbb{P}$-a.s. the Radon-Nikodym derivative $\text{d}\mathbb{Q}/\text{d}\mathbb{P}$ is given by*

$$\frac{\text{d}\mathbb{Q}}{\text{d}\mathbb{P}}(\mathbf{f}) = \exp\left[\left\langle \mathbf{C}^{-1/2}\left(\mathbf{m}_1 - \mathbf{m}_2\right), \mathbf{C}^{-1/2}\left(\mathbf{f} - \mathbf{m}_2\right)\right\rangle - \tfrac{1}{2}\|\mathbf{C}^{-1/2}(\mathbf{m}_1 - \mathbf{m}_2)\|^2\right] \forall \mathbf{f} \in \mathcal{H}. \tag{12}$$

Proof. The proof of both lemmas is in the Appendix A. □

Lemma 1 states the three conditions for the equivalence of two Gaussian measures. Lemma 2, a consequence of the Feldman-Hájek theorem, provides the Radon-Nikodym derivative formula for Gaussian measures on $\mathcal{H}$.

To train the diffusion model in functional space we have to minimize the upper bound of Proposition 1, which requires us to compute the KL divergence between the measures $\mathbb{Q}, \mathbb{P}$. In order to satisfy Lemma 1, which will enable us to use Lemma 2 to compute the KL divergence, we make the following assumption:

Assumption 1 *Let* $\mathbb{Q} = \mathcal{N}(\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0), \tilde{\beta}_t \mathbf{C})$ *and* $\mathbb{P}_\theta = \mathcal{N}(\mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t), \tilde{\beta}_t \mathbf{C})$ *be Gaussian measures on* $\mathcal{H}$*. With a conditional component* $\mathbf{e}$*, which can be an element of finite-dimensional space* $\mathbb{R}^d$ *or Hilbert space* $\mathcal{H}$*, there exists a parameter set* θ *such that the difference in mean elements of the two measures falls within the scaled covariance space:*

$$\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) - \mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t) \in (\tilde{\beta}_t \mathbf{C})^{1/2}(\mathcal{H}). \tag{13}$$

By making this assumption we satisfy all three conditions of Lemma 1: (i) : $\mathbf{C}_1^{1/2}(\mathcal{H}) = \mathbf{C}_2^{1/2}(\mathcal{H}) = (\tilde{\beta}_t \mathbf{C})^{1/2}(\mathcal{H}) = \mathcal{H}_0$; (ii) : $\mathbf{m}_1 - \mathbf{m}_2 \in \mathcal{H}_0$ is directly satisfied from Assumption 1; (iii) : $(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})^* - \mathbf{I} = \mathbf{I} - \mathbf{I}$ is the zero operator, which is trivially a Hilbert-Schmidt operator as its Hilbert-Schmidt norm is 0. As a consequence, $\mathbb{Q}$ and $\mathbb{P}$ are equivalent, allowing us to utilize the Radon-Nikodym derivative from Lemma 2.

Theorem 1 (Conditional Diffusion Optimality in Function Space). *Given the specified conditions in Assumption 1, the minimization of the learning objective in Proposition 1 is equivalent to obtaining the parameter set* θ^* *that is the solution to the problem*

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathbf{u}_0 \sim \mathbb{Q}_{\text{data}}, t \sim [1,T]} \lambda_t \left\| \mathbf{C}^{-1/2} \left(\mathbf{A}\boldsymbol{\xi} - \boldsymbol{\xi}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{A}\boldsymbol{\xi}, \mathbf{e}, t) \right) \right\|_{\mathcal{H}}^2, \tag{14}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$*,* $\mathbf{A} : \mathcal{H} \to \mathcal{H}$ *denotes a smoothing operator,* $\mathbf{e} \in (\mathbb{R}^d \cup \mathcal{H})$ *is a conditional component,* $\boldsymbol{\xi}_\theta : \{1, 2, \ldots, T\} \times (\mathbb{R}^d \cup \mathcal{H}) \times \mathcal{H} \to \mathcal{H}$ *is a parameterized mapping,* $\lambda_t = \beta_t^2 / 2\tilde{\beta}_t(1-\beta_t)(1-\bar{\alpha}_t) \in \mathbb{R}$ *is a time-dependent constant.*

Proof. Please refer to the Appendix A for the full proof. □

4.2 Conditional Denoiser with Cross-Attention Neural Operators

Our ∞-`Brush` utilizes a hierarchical denoiser architecture including a *sparse level* for efficiently capturing fine-grained details and a *grid level* for global information (Fig. 2). We discretize the noisy functions $\mathbf{u} \in \mathcal{H}$ and denoised functions $\mathbf{s} \in \mathcal{H}$ by randomly selecting a subset of coordinates $\mathbf{x} = \{\mathbf{x}^{(i)}\}_{1 \le i \le N} \subset \mathcal{X}$. At the sparse level, we successively apply a sparse neural operator, our cross-attention neural operator, and self-attention on pointwise evaluations of the function.

The computational complexity of the vanilla attention is quadratic $\mathcal{O}(N^2 d)$ w.r.t. the sequence length, or number of function samples (in this case N), and linear w.r.t. their dimension d. For learning operators in infinite dimensions, N

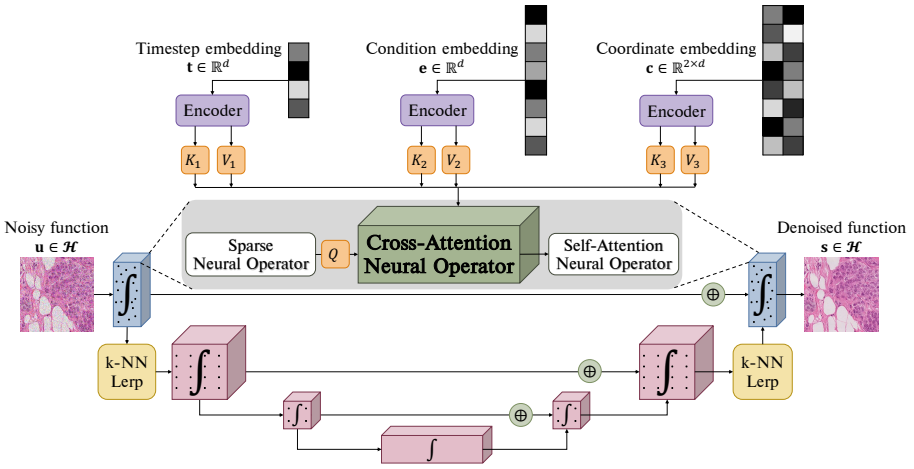

Fig. 2: Given a noisy function $\mathbf{u} \in \mathcal{H}$, we discretize it by randomly selecting a subset of coordinates $\mathbf{x} = \{\mathbf{x}^{(i)}\}_{1\le i\le N} \subset \mathcal{X}$ then feed it into our conditional denoiser returning a denoised function $\mathbf{s} \in \mathcal{H}$. The conditional denoiser architecture of ∞-Brush includes a sparse level and a grid level. The sparse level (in blue) utilizes a sparse neural operator, a cross-attention neural operator, and a self-attention neural operator, focusing on capturing fine-grained details. The grid level (in pink) targets global information. We use k-NN linear interpolation to transform the sparse points to a regularly spaced grid.

could go up to millions of points (*e.g.* when generating 4096×4096 images, $N \approx 16$ million points). We address that problem by proposing a cross-attention neural operator of linear complexity with respect to N.

Specifically in the cross-attention neural operator, suppose we have L conditional embeddings $\{Y_l \in \mathbb{R}^{N_l \times d}\}_{1\le l\le L}$. In our ∞-Brush, $L = 3$ representing the diffusion timestep embedding $\mathbf{t}$, condition embedding $\mathbf{e}$, and coordinate embedding $\mathbf{c}$. First, we compute the queries $Q = (\mathbf{q}_i)$, keys $K_l = (\mathbf{k}_i^l) = Y_l W_k$, and values $V_l = (\mathbf{v}_i^l) = Y_l W_v$, then normalize all $\mathbf{q}_i$ and $\mathbf{k}_i$ to be $\tilde{\mathbf{q}}_i = \texttt{softmax}(\mathbf{q}_i)$ and $\tilde{\mathbf{k}}_i = \texttt{softmax}(\mathbf{k}_i)$. Finally, cross-attention is

$$\mathbf{z}_l = \tilde{\mathbf{q}}_t + \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N_l} \alpha_t^l \left(\tilde{\mathbf{q}}_t \cdot \tilde{\mathbf{k}}_i^l\right) \mathbf{v}_i^l = \tilde{\mathbf{q}}_t + \frac{1}{L}\sum_{l=1}^{L} \alpha_t^l \tilde{\mathbf{q}}_t \cdot \left(\sum_{i=1}^{N_l} \tilde{\mathbf{k}}_i^l \odot \mathbf{v}_i^l\right), \quad (15)$$

where $\alpha_t^l = 1/\sum_{j=1}^{N_l} \tilde{\mathbf{q}}_t \cdot \tilde{\mathbf{k}}_j$ is the normalization coefficient. The key difference with vanilla attention is that we first multiply pointwise vectors $\tilde{\mathbf{k}}_i^l$ and $\mathbf{v}_i^l$, and compute the dot product with $\tilde{\mathbf{q}}_t$ afterward. Hence, the complexity of Eq. 15 is $\mathcal{O}((N + \sum_l N_l)d^2)$, which is linear w.r.t. the number of points N.

The output of the sparse level is linearly interpolated to a regularly spaced grid using k-Nearest Neighbors, which is the input to the grid-level model. The grid data points are passed to a grid-based, finite-dimensional UNO architecture [2, 22] that is utilized to aggregate global information. The UNO architecture is based on the widespread UNet model, which has been widely studied to condition

finite-dimensional diffusion models [30]. Following that literature, we use the vanilla cross-attention at the bottleneck of the UNet denoiser to integrate the conditional information at the grid level. In the experiments, we show that since the coarse interpolation at the grid level is not a complete representation of the function, the conditioning needs to be applied to both finite-dimensional (grid) and infinite-dimensional (sparse) levels to attain high-quality results.

5 Experiments

5.1 Experimental Settings

Datasets. We utilize the CelebA-HQ dataset [18] as a testbed for our cross-attention neural operator. We use $30,000$ images at 1024×1024 resolution along with the facial attribute *blonde/non-blonde hair* to train our conditional diffusion model and compare with the unconditional version [2].

For large image datasets, following the evaluation of [10], we utilize digital histopathology images from The Genome Cancer Atlas (TCGA) [3] and satellite imagery from the National Agriculture Imagery Program (NAIP) [34]. We use the BRCA subset of TCGA which contains breast cancer histopathology images. We select $20\times$ patches of sizes 4096×4096 and 1024×1024, equivalent in scope to patches from $1.25\times$ and $5\times$ magnifications, respectively. To provide conditioning, we resize these images to 256×256 and extract embeddings from Quilt [17].

We utilize the NAIP images from the Chesapeake Land Cover dataset [29], by extracting 1024×1024 non-overlapping patches, resulting in $35,000$ satellite images. We train a Vision Transformer (ViT-B/16) [9] on the resized 224×224 pixel versions of these images using the self-supervised DINO algorithm [5]. We extract the learned DINO embeddings to train our conditional diffusion model on pairs of 1024×1024 images and corresponding SSL embeddings.

Evaluation metrics. Following the standard evaluation metrics of MultiDiffusion [1] and [10] for large image synthesis, we evaluate our method's image quality on both global structure and fine detail via FID scores [13] using the Clean-FID implementation [25]. For global structure, we calculate CLIP FID [27] between the resized version of generated large images and real images. For fine detail, we randomly take 256×256 crops from both synthesized and real large images and measure FID (Crop FID) between the two sets of patches.

5.2 Implementation Details

We train our ∞-`Brush` from scratch for all experiments. At each training iteration, we randomly select a subset of $256^*256 = 65536$ pixels from the image. Regarding the denoiser architecture, we leverage the implementation of the Sparse Neural Operator [20], the unconditional UNO [2], and the general neural operator [12]. For faster runtime and memory efficiency, we implement our cross-attention neural operator using FlashAttention-2 [7]. The model is trained using the Adam optimizer with a learning rate of $5e-5$ and $\beta_1 = 0.9$, $\beta_2 = 0.99$, along

Fig. 3: Large images (1024×1024) generated from our ∞-Brush, conditioned on the facial attribute *blonde/non-blonde* hair.

Table 1: The CLIP FID scores of our ∞-Brush model against ∞-Diff showcases our model's capability in conditionally generating celebrity faces on the CelebA-HQ dataset based on the facial attribute of hair color (blonde vs. non-blonde).

Dataset	# Images	Method	Training Config.	CLIP FID
CelebA-HQ (1024×1024)	30k	∞-Diff [2]	Unconditional	9.44
		∞-Brush	blonde vs. non-blonde hair	**8.38**

with an exponential moving average (EMA) rate of 0.995. During inference, we apply DDIM [33] with 50 steps for all experiments. All ∞-Brush models were trained on 4 NVIDIA A100 GPUs, with a batch size of 20 per GPU.

5.3 Experimental Results

Facial Attribute Conditional Generation. We first validate our cross-attention neural operator as an efficient conditioning mechanism for infinite-dimensional diffusion models by adding control to the generation of CelebA-HQ images.

We synthesize $3,000$ images, maintaining the same ratio of blonde/non-blonde as in the entire dataset, and calculate the CLIP FID to assess quality. We compare between our conditional ∞-Brush and the unconditional ∞-Diff [2]. As shown in Table 1, our method outperforms the unconditional model, while also allowing us to control the attribute used as conditioning. Figure 3 shows examples of large images (1024×1024) generated from ∞-Brush, conditioned on the *blonde/non-blonde* attribute.

Controllable (Very) Large Image Generation. We provide experimental results of controllable generation of large (1024×1024) and very large (4096×4096) images and compare to conditional diffusion models in finite dimensions [26] and a patch-based approach [10]. In addition, we perform an ablation study to evaluate the significance of our cross-attention neural operator. We further compare the computing resources required for the three different model categories.

Our very large image experiments on the TCGA-BRCA dataset, which has 57k image patches at a resolution of 4096×4096 pixels, reveal that ∞-Brush excels at capturing the global structure of images, as indicated by the better

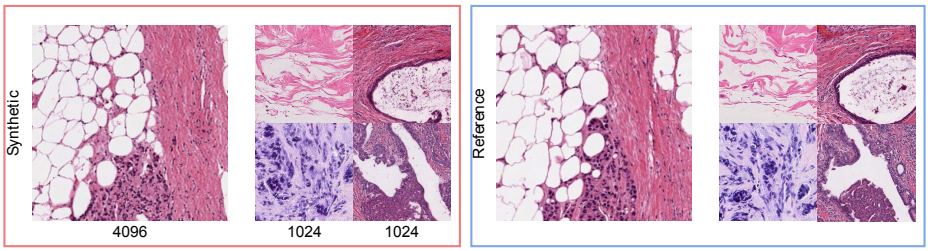

Fig. 4: Very large (4096 × 4096) and large (1024 × 1024) images generated from ∞-Brush, and the corresponding reference real images used to generate them. Given a single embedding vector of a downsampled 256 × 256 real image, ∞-Brush can synthesize images of up to 4096 × 4096 and preserve global structures of the reference image.

Table 2: Performance on controllable very large image synthesis on TCGA-BRCA dataset at 4096 × 4096 resolution. ∞-Brush outperforms the patch-based approach [10] in terms of global structure (CLIP FID) while achieving acceptable local details (Crop FID). SDXL [26] cannot be trained directly on images of 4096 × 4096 images. Additionally, an ablation study on the cross-attention neural operator shows improvement in FID metrics when the proposed mechanism is used. This emphasizes its critical role in the model's ability to synthesize high-resolution images effectively.

Dataset	# Images	Method	Training Config.	CLIP FID	Crop FID
BRCA 1.25× (4096 × 4096)	57k	Graikos *et al.* [10]	976k patches of 1024 × 1024	2.75	**11.30**
		∞-Brush	256*256 pixels of 57k full-size images	**2.63**	14.76
		∞-Brush ✗ Cross-attention neural operator		3.81	16.28

CLIP FID score (Table 2). The model's performance in finer details, reflected in the Crop FID, is slightly worse than the patch-based approach. ∞-Brush is trained on batches of just 65536 ≈ 0.4% of the pixels from the full images, offering a substantial reduction in complexity, while maintaining both global structure and finer details; see Fig. 4 for qualitative results of generated 4096×4096 images.

We can also partially attribute our better CLIP FID and worse Crop FID to the different conditioning provided to our model. The 4096 × 4096 images are first downsampled to 256 × 256 and a single embedding vector is extracted to capture information from the entire image. In comparison, the patch-based approach of [10] employs 16 local conditions that describe each of the 16 patches that form the image, helping the model to focus more on the local appearance.

SDXL [26] would need to be trained on 4096 × 4096 images which is infeasible with most current hardware setups. Thus, we resort to smaller patches from TCGA-BRCA and NAIP, at 1024 × 1024 resolution, to compare with SDXL and the patch-based method of [10]. Table 3 shows that our model attains strong global structure fidelity with superior CLIP FID scores, particularly on the BRCA dataset (3.74 vs. 6.64). Despite this, our approach results in higher Crop

Table 3: Performance on controllable large image synthesis on BRCA 5× and NAIP dataset at 1024×1024 resolution. ∞-Brush outperforms other methods in global structure accuracy, with a marginal trade-off in fine detail as reflected in Crop FID.

Dataset	# Images	Method	Training Config.	CLIP FID	Crop FID
BRCA 5× (1024×1024)	976k	SDXL [26]	976k full-size images	6.64	**6.98**
		Graikos *et al.* [10]	15M patches of 256×256	7.43	15.51
		∞-Brush	$256^* 256$ pixels of 976k full-size images	**3.74**	17.87
NAIP (1024×1024)	35k	SDXL [26]	35k full-size images	10.90	**11.50**
		Graikos *et al.* [10]	667k patches of 256×256	6.86	43.76
		∞-Brush	$256^* 256$ pixels of 35k full-size images	**6.32**	48.65

FID scores, suggesting a trade-off in capturing fine details. While SDXL is able to train at this resolution, it is important to note that ∞-Brush achieves similar performance with only a subset of the pixels, because of its highly efficient training process. Fig. 4 showcases qualitative results of generated 1024×1024 images from our model. Note that SDXL was pre-trained on LAION-5B, a 5 billion image caption pair dataset [31], whereas our model was trained from scratch.

Effectiveness of the Cross-Attention Neural Operator. We evaluate the cross-attention neural operator's advantage in the ∞-Brush model by comparing its performance on the TCGA-BRCA 4096×4096 dataset with and without this operator. When the neural operator is removed, we use vanilla cross-attention between the conditioning vector and the UNet's bottleneck layer. In Table 2, we observe a significant improvement in both CLIP FID and Crop FID scores when the cross-attention neural operator is employed. The improved scores affirm the operator's usefulness in synthesizing high-resolution 4096×4096 images. The vanilla cross-attention only applies conditioning on a regular grid of coordinates, which cannot capture fine details between coarse grid points.

Computing Resource Evaluation. We analyze the computing resources needed for training various image generation models on a single NVIDIA A100 40GB GPU. As detailed in Table 4, the training time and memory requirements for diffusion models in finite dimensions, such as SDXL, increase substantially when scaling from 1024×1024 to 4096×4096, making training infeasible on standard hardware. Conversely, the patch-based approach, while able to train at higher resolutions by dividing images into smaller patches, exhibits a parameter increase and reduced batch size. Our conditional diffusion model in function space maintains a consistent maximum batch size, significantly lower parameter count, and per epoch training time across resolutions (12 hours vs. 300 hours and 140 hours), demonstrating our method's superior scalability to sizes beyond the reach of existing methods.

Table 4: Computing resources requirements for different diffusion models. our ∞-Brush maintains a constant parameter count and batch size across resolutions, highlighting its efficiency and scalability for controllable large image generation.

Method	# Params.	Training at 1024×1024		Training at 4096×4096	
		Max batch size	Epoch time	Max batch size	Epoch time
SDXL [26]	3.5B	4	140 hr	O.O.M	1000 hr (estimated) currently infeasible
Graikos *et al.* [10]	860M	100	45 hr	4	300 hr
∞-Brush	450M	20	12 hr	20	12 hr

6 Limitations

Although ∞-Brush images exhibit better global structure consistency and maintain a degree of fine detail, they are not better than other methods in terms of local details. We highlight a few key reasons which we hypothesize hinder our model's performance. Our model has the smallest parameter count, with just half the parameters of the model of [10]. We expect model sizes to scale as more works focus on infinite-dimensional diffusion models and performance to increase, as was observed in regular, finite diffusion models. Additionally, both SDXL and [10] utilize pre-trained models as initialization, whereas, ours is trained from scratch as no infinite-dimensional pre-trained models are available, leading to worse performance in smaller datasets.

7 Conclusion

In conclusion, ∞-Brush presents a necessary leap forward in the domain of conditional large image generation, particularly for applications demanding high-resolution and domain-specific conditional generation. This paper has demonstrated that our approach effectively addresses the scalability limitations inherent in previous diffusion models while retaining a high degree of control over the generated output. By proposing a novel conditional diffusion model in function space, complemented by a cross-attention neural operator, we achieve not only state-of-the-art fidelity in the global structure of the images but also maintain acceptable detail in higher-resolution images without the excessive computational costs typically associated with such tasks. In future work, we plan to design local neural operators to capture fine details and transfer knowledge from finite-dimensional diffusion models for powerful initialization.

Acknowledgments

This research was partially supported by NCI awards 1R21CA258493-01A1, 5U24CA215109, UH3CA225021, U24CA180924, NSF grants IIS-2123920, IIS-2212046, Stony Brook Profund 2022 seed funding, and generous support from Bob Beals and Betsy Barton.

A Conditional Diffusion Models in Function Space

Forward process. The forward process of a conditional diffusion model in function space is defined as a discrete-time Markov chain that incrementally perturbs probability measure $\mathbb{Q}_{\text{data}}$ towards a Gaussian measure $\mathcal{N}(\mathbf{m}, \mathbf{C})$ with a zero mean and a specified covariance operator $\mathbf{C}$. It is a time-indexed process where each step $\mathbf{u}_t$ is obtained by applying a transformation to the previous step $\mathbf{u}_{t-1}$, which involves a scaling factor $\sqrt{1-\beta_t}\mathbf{u}_{t-1}$ related to the variance schedule β, and adding scaled Gaussian noise $\sqrt{\beta_t}\boldsymbol{\xi}_t$ with $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$:

$$\mathbf{u}_t = \sqrt{1-\beta_t}\mathbf{u}_{t-1} + \sqrt{\beta_t}\boldsymbol{\xi}_t \qquad t = 1, 2, \ldots, T. \tag{16}$$

Similar to diffusion models in finite dimensions, the forward process in function space also admits sampling $\mathbf{u}_t$ at an arbitrary timestep t in closed form. For $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_t)$, we have:

$$\begin{aligned}
\mathbf{u}_t &= \sqrt{1-\beta_t}\mathbf{u}_{t-1} + \sqrt{\beta_t}\boldsymbol{\xi}_t \quad ;\text{where } \boldsymbol{\xi}_t, \boldsymbol{\xi}_{t-1}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \\
&= \sqrt{(1-\beta_t)(1-\beta_{t-1})}\mathbf{u}_{t-2} + \sqrt{(1-\beta_t)\beta_{t-1}}\boldsymbol{\xi}_{t-1} + \sqrt{\beta_t}\boldsymbol{\xi}_t \\
&= \sqrt{(1-\beta_t)(1-\beta_{t-1})}\mathbf{u}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\xi}}_{t-1} \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{u}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\xi}
\end{aligned} \tag{17}$$

Based on the above analysis, we obtain:

$$\mathbb{Q}\left(\mathbf{u}_t|\mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_t; \sqrt{\bar{\alpha}_t}\mathbf{u}_0, (1-\bar{\alpha}_t)\mathbf{C}\right). \tag{18}$$

In the context of image generation, we discretize the function $\mathbf{u}_j$ on the mesh $\mathbf{x}_j = \{\mathbf{x}_j^{(i)}\}_{1\le i\le N} \subset \mathcal{X}$ by sampling N coordinates of each image, which results in a non-smooth input space. To achieve a smoother function representation, a smoothing operator [16, 28] $\mathbf{A} : \mathcal{H} \to \mathcal{H}$, *e.g.* a truncated Gaussian kernel, is applied to approximate the rough inputs within the function space $\mathcal{H}$:

$$\mathbb{Q}\left(\mathbf{u}_t|\mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_t; \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0, (1-\bar{\alpha}_t)\mathbf{A}\mathbf{C}\mathbf{A}^T\right). \tag{19}$$

Reverse process. The reverse process in the diffusion model framework is achieved by iteratively denoising from the Gaussian measure $\mathcal{N}(\mathbf{m}, \mathbf{C})$ back towards the probability measure $\mathbb{Q}_0 = \mathbb{Q}_{\text{data}}$. We use a variational approach to approximate posterior measures with a variational family of measures on $\mathcal{H}$ and incorporate the conditional embedding $\mathbf{e}$ to control the generation process. We model the underlying posterior measure $\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t)$ with a conditional Gaussian measure:

$$\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{e}) = \mathcal{N}\left(\mathbf{u}_{t-1}; \mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t), \mathbf{A}\mathbf{C}_\theta(\mathbf{u}_t, \mathbf{e}, t)\mathbf{A}^T\right). \tag{20}$$

Likewise, we are able to derive a closed-form representation of the forward process posteriors, which are tractable when conditioned on $\mathbf{u}_0$:

$$\mathbb{Q}\left(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{u}_0\right) = \mathcal{N}\left(\mathbf{u}_{t-1}; \tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0), \tilde{\beta}_t\mathbf{C}\right). \tag{21}$$

Using Bayes' rule, we obtain:

$$
\begin{aligned}
&\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)=\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_{t-1},\mathbf{u}_0)\frac{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_0)}{\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_0)}\\
&\propto\exp\Big(-\frac{1}{2}\big(\frac{\langle\mathbf{C}^{-1}(\mathbf{u}_t-\sqrt{\alpha_t}\mathbf{u}_{t-1}),\mathbf{u}_t-\sqrt{\alpha_t}\mathbf{u}_{t-1}\rangle}{\beta_t}+\frac{\langle\mathbf{C}^{-1}(\mathbf{u}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{A}\mathbf{u}_0),\mathbf{u}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{A}\mathbf{u}_0\rangle}{1-\bar{\alpha}_{t-1}}\\
&-\frac{\langle\mathbf{C}^{-1}(\mathbf{u}_t-\sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0),\mathbf{u}_t-\sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0\rangle}{1-\bar{\alpha}_t}\big)\Big)\\
&=\exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\langle\mathbf{C}^{-1}\mathbf{u}_{t-1},\mathbf{u}_{t-1}\rangle-2\langle\mathbf{C}^{-1}(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{u}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{A}\mathbf{u}_0),\mathbf{u}_{t-1}\rangle+C(\mathbf{u}_t,\mathbf{u}_0)\big)\Big),
\end{aligned}
\tag{22}
$$

where $C(\mathbf{u}_t,\mathbf{u}_0)$ is some function not involving $\mathbf{u}_{t-1}$ and details are omitted. Following the standard Gaussian density function, the mean and covariance of $\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)$ can be parameterized as follows (recall that $\alpha_t=1-\beta_t$ and $\bar{\alpha}_t=\prod_{i=1}^{T}\alpha_i$):

$$
\begin{aligned}
\tilde{\mathbf{m}}_t(\mathbf{u}_t,\mathbf{u}_0)&=(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{u}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{A}\mathbf{u}_0)/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\\
&=(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{u}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{A}\mathbf{u}_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t\\
&=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0+\frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{u}_t.
\end{aligned}
\tag{23}
$$

$$
\tilde{\beta}_t=1/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})=1/(\frac{\alpha_t-\bar{\alpha}_t+\beta_t}{\beta_t(1-\bar{\alpha}_{t-1})})=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t. \tag{24}
$$

Proposition 2 (Learning Objective). *The cross-entropy of conditional diffusion models in function space has a variational upper bound of*

$$
\begin{aligned}
\mathcal{L}_{\text{CE}}=-\mathbb{E}_{\mathbb{Q}}\log\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{e})\leq\mathbb{E}_{\mathbb{Q}}\Bigg[&\underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0)\parallel\mathbb{P}_\theta(\mathbf{u}_T))}_{\mathcal{L}_T}\underbrace{-\log\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})}_{\mathcal{L}_0}\\
&+\sum_{t=2}^{T}\underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)\parallel\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})}_{\mathcal{L}_{t-1}}\Bigg].
\end{aligned}
\tag{25}
$$

Proof. The conditional diffusion model in function space is trained to minimize the cross entropy as the learning objective, which is equivalent to minimizing

variational upper bound (VUB):

$$
\begin{aligned}
L_{\mathrm{CE}} &= -\mathbb{E}_{\mathbb{Q}(\mathbf{u}_0|\mathbf{e})} \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{e}) \\
&= -\mathbb{E}_{\mathbb{Q}(\mathbf{u}_0|\mathbf{e})} \log \Big(\int \mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e}) d\mathbf{u}_{1:T} \Big) \\
&= -\mathbb{E}_{\mathbb{Q}(\mathbf{u}_0|\mathbf{e})} \log \Big(\int \mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e}) \frac{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})}{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e})} d\mathbf{u}_{1:T} \Big) \\
&= -\mathbb{E}_{\mathbb{Q}(\mathbf{u}_0|\mathbf{e})} \log \Big(\mathbb{E}_{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e})} \frac{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})}{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e})} \Big) \\
&\leq -\mathbb{E}_{\mathbb{Q}(\mathbf{u}_{0:T}|\mathbf{e})} \log \frac{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})}{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e})} \\
&= \mathbb{E}_{\mathbb{Q}(\mathbf{u}_{0:T}|\mathbf{e})} \Big[\log \frac{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0,\mathbf{e})}{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}(\mathbf{u}_{0:T}|\mathbf{e})} \Big[\log \frac{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})} \Big] = L_{\mathrm{VUB}}.
\end{aligned}
\tag{26}
$$

To convert each term in the equation to be analytically computable, the objective can be further rewritten to be a combination of several KL-divergence and entropy terms:

$$
\begin{aligned}
L_{\mathrm{VUB}} &= \mathbb{E}_{\mathbb{Q}(\mathbf{u}_{0:T}|\mathbf{e})} \Big[\log \frac{\mathbb{Q}(\mathbf{u}_{1:T}|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{0:T}|\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[\log \frac{\prod_{t=1}^{T} \mathbb{Q}(\mathbf{u}_t|\mathbf{u}_{t-1})}{\mathbb{P}_\theta(\mathbf{u}_T) \prod_{t=1}^{T} \mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[-\log \mathbb{P}_\theta(\mathbf{u}_T) + \sum_{t=1}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_{t-1})}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[-\log \mathbb{P}_\theta(\mathbf{u}_T) + \sum_{t=2}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_{t-1})}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} + \log \frac{\mathbb{Q}(\mathbf{u}_1|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[-\log \mathbb{P}_\theta(\mathbf{u}_T) + \sum_{t=2}^{T} \log \Big(\frac{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} \cdot \frac{\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_0)}{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_0)} \Big) + \log \frac{\mathbb{Q}(\mathbf{u}_1|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[-\log \mathbb{P}_\theta(\mathbf{u}_T) + \sum_{t=2}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} + \sum_{t=2}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_t|\mathbf{u}_0)}{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_0)} + \log \frac{\mathbb{Q}(\mathbf{u}_1|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[-\log \mathbb{P}_\theta(\mathbf{u}_T) + \sum_{t=2}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} + \log \frac{\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0)}{\mathbb{Q}(\mathbf{u}_1|\mathbf{u}_0)} + \log \frac{\mathbb{Q}(\mathbf{u}_1|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})} \Big] \\
&= \mathbb{E}_{\mathbb{Q}} \Big[\log \frac{\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_T)} + \sum_{t=2}^{T} \log \frac{\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0)}{\mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e})} - \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e}) \Big] \\
&= \mathbb{E}_{\mathbb{Q}} [\underbrace{\mathrm{KL}(\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{\mathrm{KL}(\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t,\mathbf{e}))}_{L_{t-1}} \underbrace{- \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1,\mathbf{e})}_{L_0}]
\end{aligned}
\tag{27}
$$

Combine Eq. 26 and Eq. 27, we obtain:

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{\mathbb{Q}} \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{e}) \leq \mathbb{E}_{\mathbb{Q}} \Bigg[\underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_T|\mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_T))}_{\mathcal{L}_T} \underbrace{- \log \mathbb{P}_\theta(\mathbf{u}_0|\mathbf{u}_1, \mathbf{e})}_{\mathcal{L}_0} + \sum_{t=2}^{T} \underbrace{\text{KL}(\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{e})}_{\mathcal{L}_{t-1}} \Bigg]. \tag{28}$$

□

To compute the KL divergence between probability measures $\text{KL}(\mathbb{Q} \parallel \mathbb{P})$, we need to utilize a measure-theoretic definition of the KL divergence, which is stated in the following lemmas [6].

Lemma 3 (Measure Equivalence - The Feldman-Hájek Theorem). *Let $\mathbb{Q} = \mathcal{N}(\mathbf{m}_1, \mathbf{C}_1)$ and $\mathbb{P} = \mathcal{N}(\mathbf{m}_2, \mathbf{C}_2)$ be Gaussian measures on $\mathcal{H}$. They are equivalent if and only if $(i) : \mathbf{C}_1^{1/2}(\mathcal{H}) = \mathbf{C}_2^{1/2}(\mathcal{H}) = \mathcal{H}_0$, $(ii) : \mathbf{m}_1 - \mathbf{m}_2 \in \mathcal{H}_0$, and (iii) : The operator $(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})^* - \mathbf{I}$ is a Hilbert-Schmidt operator on the closure $\overline{\mathcal{H}_0}$.*

Proof. Refer to the proof of Theorem 2.25 of Da Prato and Zabczyk [6]. □

Lemma 4 (The Radon-Nikodym Derivative). *Let $\mathbb{Q} = \mathcal{N}(\mathbf{m}_1, \mathbf{C}_1)$ and $\mathbb{P} = \mathcal{N}(\mathbf{m}_2, \mathbf{C}_2)$ be Gaussian measures on $\mathcal{H}$. If $\mathbb{P}$ and $\mathbb{Q}$ are equivalent and $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}$, then $\mathbb{P}$-a.s. the Radon-Nikodym derivative $\text{d}\mathbb{Q}/\text{d}\mathbb{P}$ is given by*

$$\frac{\text{d}\mathbb{Q}}{\text{d}\mathbb{P}}(\mathbf{f}) = \exp\left[\left\langle \mathbf{C}^{-1/2}(\mathbf{m}_1 - \mathbf{m}_2), \mathbf{C}^{-1/2}(\mathbf{f} - \mathbf{m}_2)\right\rangle - \tfrac{1}{2}\|\mathbf{C}^{-1/2}(\mathbf{m}_1 - \mathbf{m}_2)\|^2\right] \forall \mathbf{f} \in \mathcal{H}. \tag{29}$$

Proof. Refer to the proof of Theorem 2.23 of Da Prato and Zabczyk [6]. □

Lemma 3 states the three conditions under which two Gaussian measures are equivalent. Lemma 4 is the consequence of the Feldman-Hájek theorem, providing the Radon-Nikodym derivative formula when dealing with Gaussian measures on $\mathcal{H}$.

To train the diffusion model in functional space we have to minimize the upper bound of Proposition 2, which requires us to compute the KL divergence between the measures $\mathbb{Q}, \mathbb{P}$. In order to satisfy Lemma 3, which will enable us to use Lemma 4 to compute the KL divergence, we make the following assumption:

Assumption 2 *Let $\mathbb{Q} = \mathcal{N}(\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0), \tilde{\beta}_t\mathbf{C})$ and $\mathbb{P}_\theta = \mathcal{N}(\mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t), \tilde{\beta}_t\mathbf{C})$ be Gaussian measures on $\mathcal{H}$. With a conditional component $\mathbf{e}$, which can be an element of finite-dimensional space $\mathbb{R}^d$ or Hilbert space $\mathcal{H}$, there exists a parameter set θ such that the difference in mean elements of the two measures falls within the scaled covariance space:*

$$\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) - \mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t) \in (\tilde{\beta}_t\mathbf{C})^{1/2}(\mathcal{H}). \tag{30}$$

By making this assumption we satisfy all three conditions of Lemma 3: (i) : $\mathbf{C}_1^{1/2}(\mathcal{H}) = \mathbf{C}_2^{1/2}(\mathcal{H}) = (\tilde{\beta}_t\mathbf{C})^{1/2}(\mathcal{H}) = \mathcal{H}_0$; (ii) : $\mathbf{m}_1 - \mathbf{m}_2 \in \mathcal{H}_0$ is directly satisfied from Assumption 2; (iii) : $(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})(\mathbf{C}_1^{-1/2}\mathbf{C}_2^{1/2})^* - \mathbf{I} = \mathbf{I} - \mathbf{I}$ is the zero operator, which is trivially a Hilbert-Schmidt operator as its Hilbert-Schmidt norm is 0. As a consequence, $\mathbb{Q}$ and $\mathbb{P}$ are equivalent, allowing us to utilize the Radon-Nikodym derivative from Lemma 4.

Theorem 2 (Conditional Diffusion Optimality in Function Space). *Given the specified conditions in Assumption 2, the minimization of the learning objective in Proposition 2 is equivalent to obtaining the parameter set θ^* that is the solution to the problem*

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathbf{u}_0\sim\mathbb{Q}_{\text{data}},t\sim[1,T]}\lambda_t \left|\left|\mathbf{C}^{-1/2}\left(\mathbf{A}\boldsymbol{\xi} - \boldsymbol{\xi}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{A}\boldsymbol{\xi}, \mathbf{e}, t)\right)\right|\right|_{\mathcal{H}}^2, \tag{31}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, $\mathbf{A} : \mathcal{H} \to \mathcal{H}$ denotes a smoothing operator, $\mathbf{e} \in (\mathbb{R}^d \cup \mathcal{H})$ is a conditional component, $\boldsymbol{\xi}_\theta : \{1, 2, \ldots, T\} \times (\mathbb{R}^d \cup \mathcal{H}) \times \mathcal{H} \to \mathcal{H}$ is a parameterized mapping, $\lambda_t = \beta_t^2/2\tilde{\beta}_t(1-\beta_t)(1-\bar{\alpha}_t) \in \mathbb{R}$ is a time-dependent constant.

Proof. Under Assumption 2, we are now able to use the Radon-Nikodym derivative to compute the KL divergence:

$$\begin{aligned}\mathrm{KL}\left[\mathbb{Q} \parallel \mathbb{P}\right] &= \int_{\mathcal{H}} \log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(\mathbf{f})\ \mathrm{d}\mathbb{Q}(\mathbf{f}) \\ &= -\frac{1}{2}\|\mathbf{C}^{-1/2}(\mathbf{m}_1 - \mathbf{m}_2)\|_{\mathcal{H}}^2 + \int_{\mathcal{H}} \left\langle \mathbf{C}^{-1/2}\left(\mathbf{m}_1 - \mathbf{m}_2\right), \mathbf{C}^{-1/2}\left(\mathbf{f} - \mathbf{m}_2\right)\right\rangle\ \mathrm{d}\mathbb{Q}(\mathbf{f}).\end{aligned} \tag{32}$$

We now use spectral decomposition to compute the integral term. Let $\{\lambda_j, \mathbf{e}_j\}_{j=1}^\infty$ be the eigenvalues and eigenvectors of $\mathbf{C}$. The eigenvector of $\mathbf{C}$ form an orthonormal basis for $\mathcal{H}$ by the spectral theorem, as $\mathbf{C}$ is a self-adjoint compact operator. Hence, the second integral is:

$$\begin{aligned}&\int_{\mathcal{H}} \left\langle \mathbf{C}^{-1/2}\left(\mathbf{m}_1 - \mathbf{m}_2\right), \mathbf{C}^{-1/2}\left(\mathbf{f} - \mathbf{m}_2\right)\right\rangle\ \mathrm{d}\mathbb{Q}(f) \\ &= \int_{\mathcal{H}} \sum_{j=1}^\infty \left\langle \mathbf{m}_1 - \mathbf{m}_2, \mathbf{e}_j\right\rangle \left\langle \mathbf{f} - \mathbf{m}_2, \mathbf{e}_j\right\rangle \lambda_j^{-1}\ \mathrm{d}\mathbb{Q}(f) \\ &= \sum_{j=1}^\infty \lambda_j^{-1} \left\langle \mathbf{m}_1 - \mathbf{m}_2, \mathbf{e}_j\right\rangle \int_{\mathcal{H}} \left\langle \mathbf{f} - \mathbf{m}_2, \mathbf{e}_j\right\rangle\ \mathrm{d}\mathbb{Q}(f) \\ &= \sum_{j=1}^\infty \lambda_j^{-1} \left\langle \mathbf{m}_1 - \mathbf{m}_2, \mathbf{e}_j\right\rangle^2 \\ &= \left\langle \mathbf{C}^{-1/2}\left(\mathbf{m}_1 - \mathbf{m}_2\right), \mathbf{C}^{-1/2}\left(\mathbf{m}_1 - \mathbf{m}_2\right)\right\rangle.\end{aligned} \tag{33}$$

Combine Eq. 32 and Eq. 33, we obtain:

$$\mathrm{KL}\left[\mathbb{Q} \parallel \mathbb{P}\right] = \frac{1}{2}\|\mathbf{C}^{-1/2}(\mathbf{m}_1 - \mathbf{m}_2)\|_{\mathcal{H}}^2 \tag{34}$$

From Proposition 2, the KL divergence between Gaussian measures $\mathbb{Q}$ and $\mathbb{P}$ now becomes:

$$\begin{aligned} L_{t-1} &= \mathrm{KL}\left[\mathbb{Q}(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{u}_0) \parallel \mathbb{P}_\theta(\mathbf{u}_{t-1}|\mathbf{u}_t, \mathbf{e})\right] \\ &= \frac{1}{2}\|(\tilde{\beta}_t \mathbf{C})^{-1/2}(\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) - \mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t))\|_{\mathcal{H}}^2 \end{aligned} \tag{35}$$

Our model must predict the mean function $\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0)$. Recall that we got the expression of $\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0)$ and $\mathbf{u}_0$ depending on $\mathbf{u}_t$:

$$\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{u}_t. \tag{36}$$

$$\mathbf{A}\mathbf{u}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{u}_t - \sqrt{1-\bar{\alpha}_t}\mathbf{A}\boldsymbol{\xi}\right) \quad ; \text{where } \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) \tag{37}$$

Combine these two expressions, we have:

$$\tilde{\mathbf{m}}_t(\mathbf{u}_t, \mathbf{u}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{u}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{A}\boldsymbol{\xi}\right) \tag{38}$$

Thus, we parameterize the variational mean via:

$$\mathbf{m}_\theta(\mathbf{u}_t, \mathbf{e}, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{u}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\xi}_\theta(\mathbf{u}_t, \mathbf{e}, t)\right) \tag{39}$$

Finally, plugging Eq. 38 and Eq. 39 into L_{t-1}, we obtain:

$$\begin{aligned} L_{t-1} &= \frac{1}{2}\left\|(\tilde{\beta}_t \mathbf{C})^{-1/2}\left(\frac{1}{\sqrt{1-\beta_t}}\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{A}\boldsymbol{\xi} - \frac{1}{\sqrt{1-\beta_t}}\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\xi}_\theta(\mathbf{u}_t, \mathbf{e}, t)\right)\right\|_{\mathcal{H}}^2 \\ &= \frac{\beta_t^2}{2\tilde{\beta}_t(1-\beta_t)(1-\bar{\alpha}_t)}\left\|\mathbf{C}^{-1/2}\left(\mathbf{A}\boldsymbol{\xi} - \boldsymbol{\xi}_\theta(\mathbf{u}_t, \mathbf{e}, t)\right)\right\|_{\mathcal{H}}^2 \\ &= \frac{\beta_t^2}{2\tilde{\beta}_t(1-\beta_t)(1-\bar{\alpha}_t)}\left\|\mathbf{C}^{-1/2}\left(\mathbf{A}\boldsymbol{\xi} - \boldsymbol{\xi}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{u}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{A}\boldsymbol{\xi}, \mathbf{e}, t)\right)\right\|_{\mathcal{H}}^2 \end{aligned} \tag{40}$$

□

B Experiments

B.1 Long-range dependencies

We obtained the patch-based large-image model of Graikos et al. [10] directly from the authors and tried to apply it to synthesize images larger than 1024 × 1024 pixels. The overreliance of the patch-based model on the local descriptors (patch SSL embeddings) leads to the loss of large-scale structures and fails to capture long-range dependencies across the image. As a qualitative example (Figure 5), we get a reference image of size 2048 × 2048 pixels from TCGA-BRCA and extract embeddings in an attempt to generate a variation of it using our model and the patch-based model of [10]. As illustrated, ∞-`Brush` retains large-scale structures (such as clearly-separated clusters of cells) that can span multiple patches, in comparison to the image generated from [10].

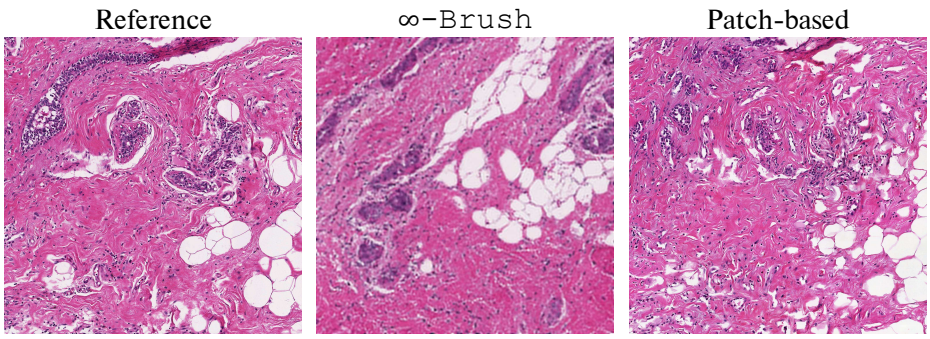

Fig. 5: Long-range dependencies comparison between our ∞-Brush and patched-based method [10]. ∞-Brush retains large-scale structures (such as clearly-separated clusters of cells) that can span multiple patches in comparison to the image generated from [10].

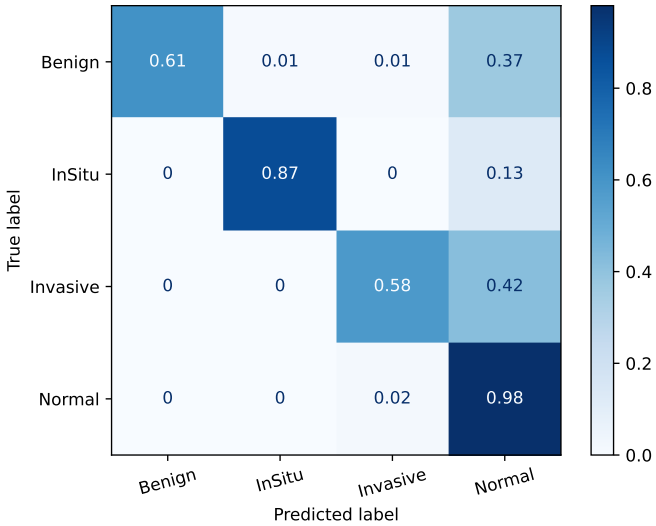

Fig. 6: Confusion matrix of zero-shot classification of generated images.

B.2 Zero-shot classification

Following the experiment of [10], we generate images from a pre-defined set of four classes: benign tissue, in-situ, invasive carcinoma, and normal tissue. We use a VLM (Quilt) as a zero-shot classifier and compute the confusion matrix (CM). Figure 6 shows that ∞-Brush generates images semantically aligned with the text prompts.

Table 5: Synthetic data improves the accuracy significantly on the BACH test set.

Training Data	Test Acc
Real	79 %
Real + synthetic	**83 %**

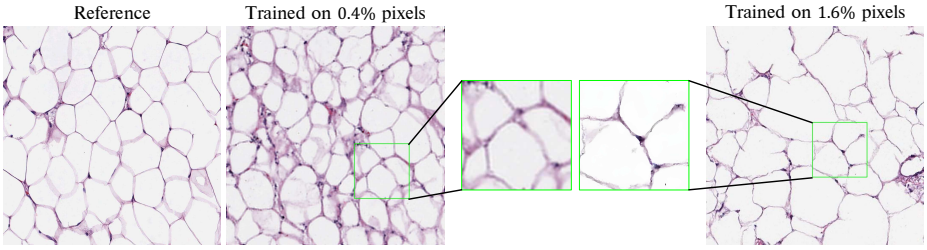

Fig. 7: Ablation on % pixels for training and zoomed-in views.

B.3 Application of synthetic data on downstream task

As a practical application, we double the number of training images of the BACH dataset by synthesizing images using real data embedding and evaluating the test set. Table 5 shows a significant accuracy boost from these synthetic images.

B.4 Ablation study on % of pixels for training

We compare our model when training on 256*256 (0.4%) *vs.* 512*512 pixels (1.6%). Figure 7 shows that training with more pixels improves performance. Our model efficiently uses 0.4% of pixels compared to 25% of ∞-Diff's due to the incorporation of coordinate embedding in CANO, functioning as positional embedding.

B.5 Qualitative results

In Figure 8 and Figure 9, we illustrate the generated very large (4096×4096) and large (1024×1024) images of TCGA-BRCA [3] dataset. We also show synthesized satellite images at 2048×2048 and 1024×1024 resolutions in Figure 10. Qualitative results show that given a single embedding vector of a downsampled 256×256 real image, ∞-`Brush` can synthesize images of arbitrary resolutions up to 4096×4096 and preserve global structures of the reference image.

Figure 11 shows examples where the model did not successfully capture spatial structures and details from the reference images. This can be attributed to both the model and the conditioning used to represent the images.

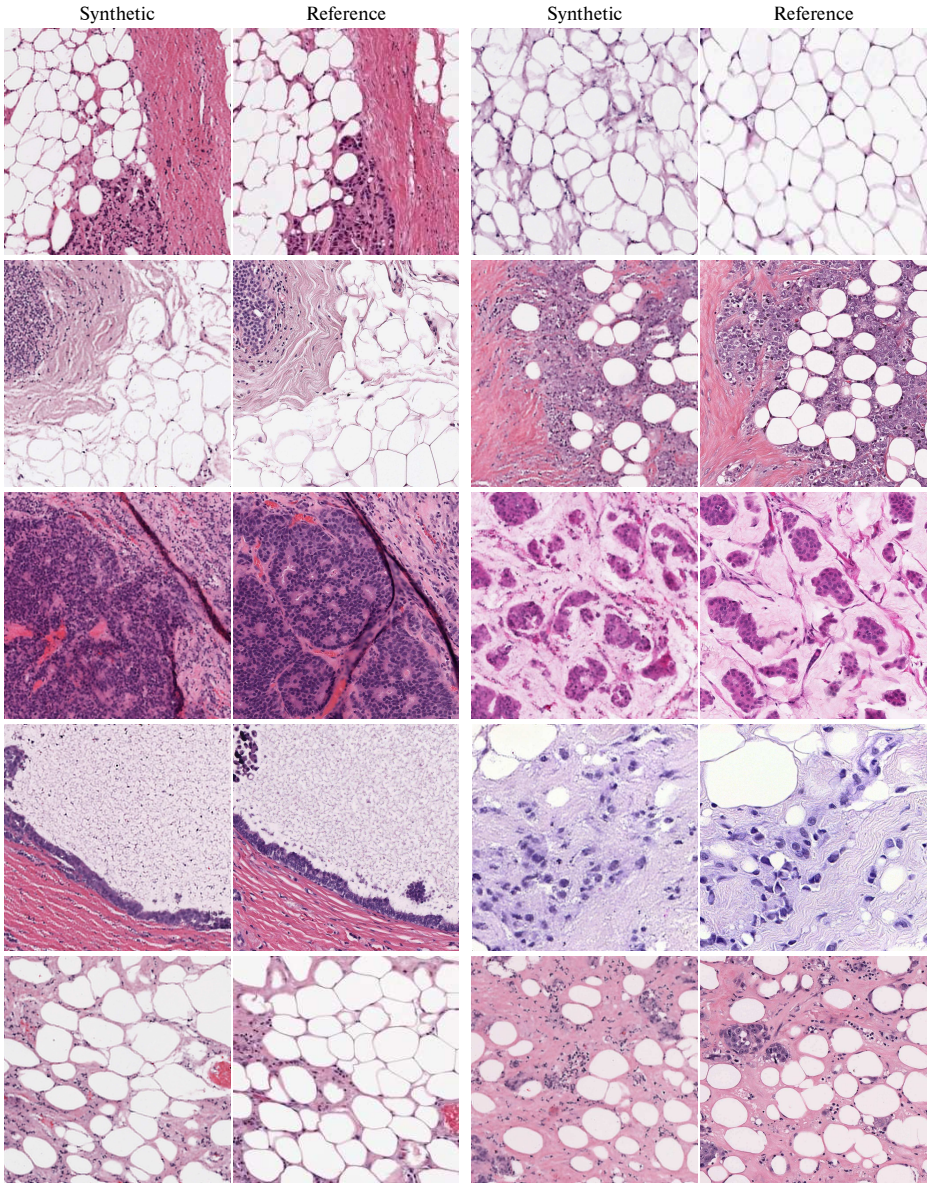

Fig. 8: Very large (4096×4096) images generated from ∞-Brush, and the corresponding reference real images used to generate them. Given a single embedding vector of a downsampled 256×256 real image, ∞-Brush can synthesize images of up to 4096×4096 and preserve global structures of the reference image.

References

1. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation. In: Krause, A., Brunskill, E., Cho, K., Engelhardt,

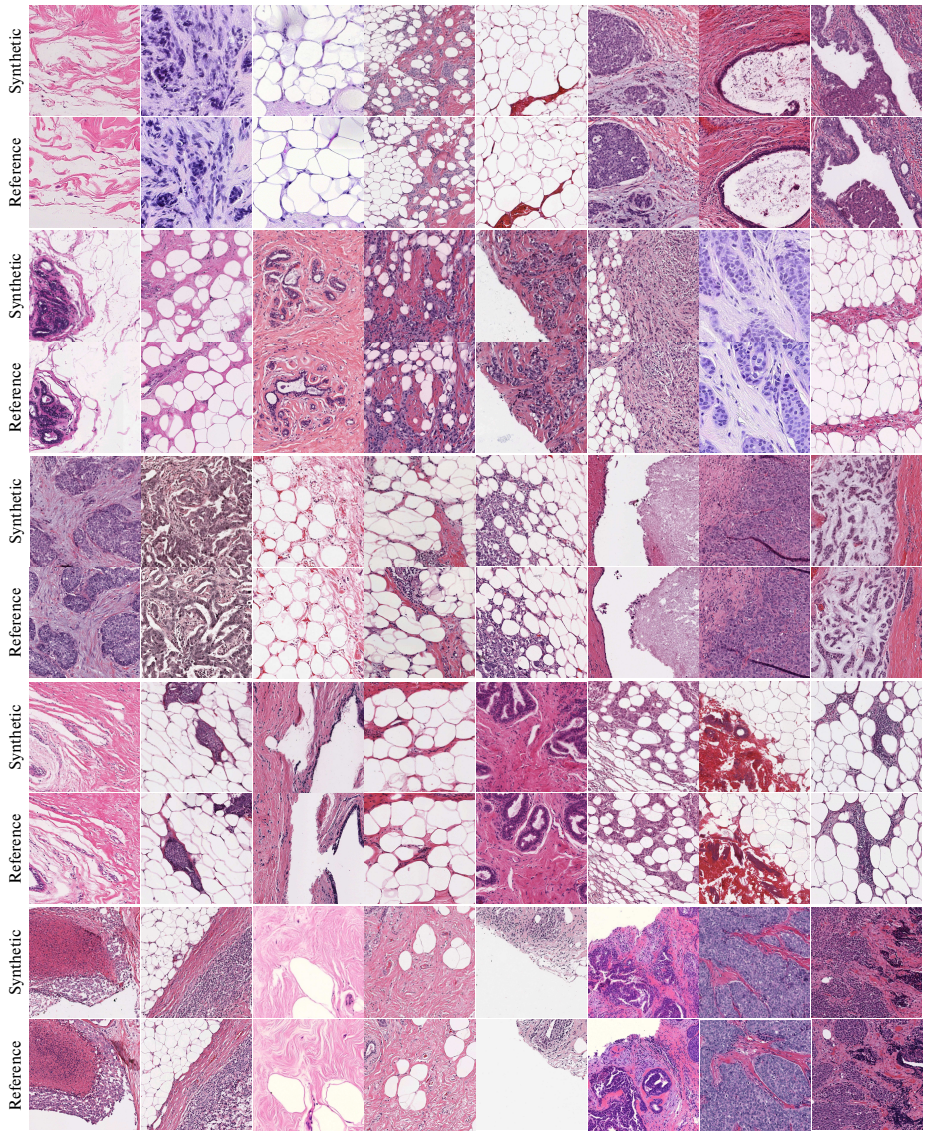

Fig. 9: Large (1024×1024) images generated from ∞-**Brush**, and the corresponding reference real images used to generate them. Given a single embedding vector of a downsampled 256×256 real image, ∞-**Brush** can synthesize images at arbitrary resolutions and preserve global structures of the reference image.

B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 1737–1752. PMLR (2023), https://proceedings.mlr.press/v202/bar-tal23a.html 2, 3, 4, 10

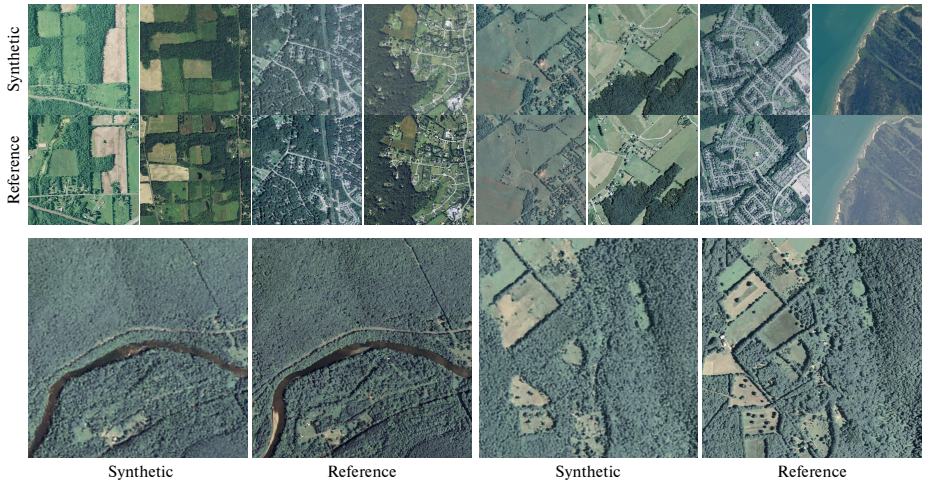

Fig. 10: Satellite large (1024×1024 and 2048×2048) images generated from ∞-Brush, and the corresponding reference real images used to generate them. Given a single embedding vector of a downsampled 256×256 real image, ∞-Brush can synthesize images at arbitrary resolutions and preserve global structures of the reference image.

2. Bond-Taylor, S., Willcocks, C.G.: ∞-diff: Infinite resolution diffusion with subsampled mollified states. arXiv preprint arXiv:2303.18242 (2023) 2, 3, 4, 5, 9, 10, 11
3. Cancer Genome Atlas Research Network, J., et al.: The cancer genome atlas pan-cancer analysis project. Nat. Genet **45**(10), 1113–1120 (2013) 10, 22
4. Cao, S.: Choose a transformer: Fourier or galerkin. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=ssohLcmn4-r 6
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021) 10
6. Da Prato, G., Zabczyk, J.: Stochastic Equations in Infinite Dimensions. Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2 edn. (2014) 5, 7, 18
7. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=mZn2Xyh9Ec 10
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021) 4
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 10
10. Graikos, A., Yellapragada, S., Le, M.Q., Kapse, S., Prasanna, P., Saltz, J., Samaras, D.: Learned representation-guided diffusion models for large-image generation. arXiv preprint arXiv:2312.07330 (2023) 2, 3, 4, 10, 11, 12, 13, 14, 20, 21

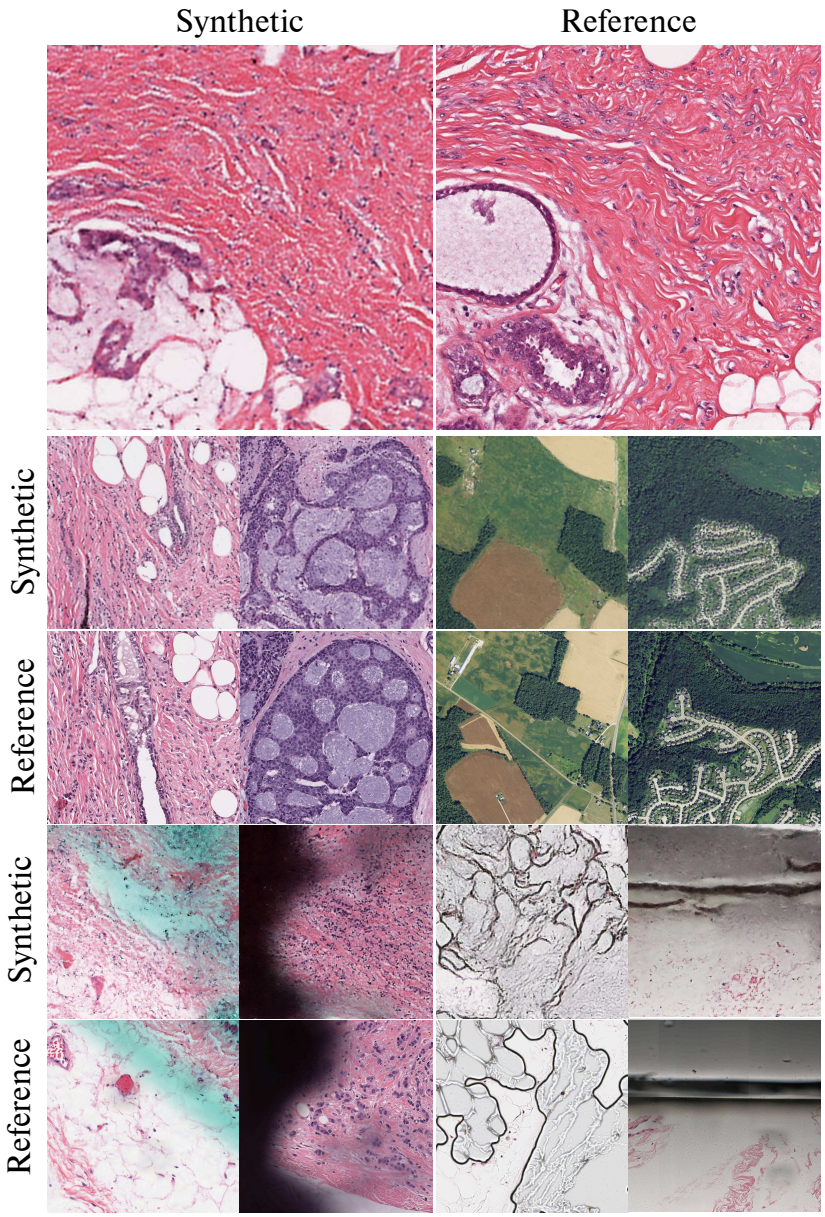

Fig. 11: Uncurated (4096×4096 and 2048×2048) images generated from ∞-`Brush`, and the corresponding reference real images used to generate them. Our model fails to capture spatial structure and details in specific regions of reference images (top 3 rows). In the last 2 rows, it shows that our model fails to controllably synthesize images due to bad conditioning information.

11. Gu, J., Zhai, S., Zhang, Y., Susskind, J.M., Jaitly, N.: Matryoshka diffusion models. In: The Twelfth International Conference on Learning Representations (2023) 2
12. Hao, Z., Wang, Z., Su, H., Ying, C., Dong, Y., Liu, S., Cheng, Z., Song, J., Zhu, J.: Gnot: a general neural operator transformer for operator learning. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023) 6, 10
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 10
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 3, 5
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 4
16. Hoogeboom, E., Salimans, T.: Blurring diffusion models. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=0jDkC57x5sz 6, 15
17. Ikezogwo, W.O., Seyfioglu, M.S., Ghezloo, F., Geva, D.S.C., Mohammed, F.S., Anand, P.K., Krishna, R., Shapiro, L.: Quilt-1m: One million image-text pairs for histopathology. arXiv preprint arXiv:2306.11207 (2023) 10
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Hk99zCeAb 3, 10
19. Kerrigan, G., Ley, J., Smyth, P.: Diffusion generative models in infinite dimensions. In: Ruiz, F., Dy, J., van de Meent, J.W. (eds.) Proceedings of The 26th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 206, pp. 9538–9563. PMLR (25–27 Apr 2023), https://proceedings.mlr.press/v206/kerrigan23a.html 4, 5
20. Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A.: Neural operator: Learning maps between function spaces with applications to pdes. Journal of Machine Learning Research **24**(89), 1–97 (2023), http://jmlr.org/papers/v24/21-1524.html 6, 10
21. Le, M.Q., Nguyen, T.V., Le, T.N., Do, T.T., Do, M.N., Tran, M.T.: Maskdiff: Modeling mask distribution with diffusion probabilistic model for few-shot instance segmentation. Proceedings of the AAAI Conference on Artificial Intelligence **38**(3), 2874–2881 (Mar 2024). https://doi.org/10.1609/aaai.v38i3.28068, https://ojs.aaai.org/index.php/AAAI/article/view/28068 3
22. Lim, J.H., Kovachki, N.B., Baptista, R., Beckham, C., Azizzadenesheli, K., Kossaifi, J., Voleti, V., Song, J., Kreis, K., Kautz, J., Pal, C., Vahdat, A., Anandkumar, A.: Score-based diffusion models in function space (2023) 4, 5, 9
23. Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., Plumbley, M.D.: Audioldm: Text-to-audio generation with latent diffusion models. arXiv preprint arXiv:2301.12503 (2023) 1
24. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021) 1, 4
25. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: CVPR (2022) 10
26. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023) 1, 2, 3, 4, 11, 12, 13, 14

27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 10
28. Rissanen, S., Heinonen, M., Solin, A.: Generative modelling with inverse heat dissipation. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=4PJUBT9f2Ol 6, 15
29. Robinson, C., Hou, L., Malkin, K., Soobitsky, R., Czawlytko, J., Dilkina, B., Jojic, N.: Large scale high-resolution land cover mapping with multi-resolution data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12726–12735 (2019) 10
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 3, 10
31. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022) 13
32. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 2256–2265. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/sohl-dickstein15.html 3
33. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020) 11
34. USGS: National agriculture imagery program (NAIP) (2023), https://www.usgs.gov/centers/eros/science/usgs-eros-archive-aerial-photography-national-agriculture-imagery-program-naip 10