Binary Latent Diffusion

Ze Wang
Purdue University
zewang@purdue.edu

Jiang Wang Zicheng Liu
Microsoft Corporation
{jiangwang, zliu}@microsoft.com

Qiang Qiu
Purdue University
qqiu@purdue.edu

Abstract

In this paper, we show that a binary latent space can be explored for compact yet expressive image representations. We model the bi-directional mappings between an image and the corresponding latent binary representation by training an auto-encoder with a multivariate Bernoulli encoding distribution. On the one hand, the binary latent space provides a compact discrete image representation of which the distribution can be modeled more efficiently than pixels or continuous latent representations. On the other hand, we now represent each image patch as a binary vector instead of an index of a learned cookbook as in discrete image representations with vector quantization. In this way, we obtain binary latent representations that allow for better image quality and high-resolution image representations without any multi-stage hierarchy in the latent space. In this binary latent space, images can now be generated effectively using a binary latent diffusion model tailored specifically for modeling the prior over the binary image representations. We present both conditional and unconditional image generation experiments with multiple datasets, and show that the proposed method performs comparably to state-of-the-art methods while dramatically improving the sampling efficiency to as few as 16 steps without using any test-time acceleration. The proposed framework can also be seamlessly scaled to 1024×1024 *high-resolution image generation without resorting to latent hierarchy or multi-stage refinements.*

1. Introduction

The goal of modeling the image distribution that allows efficient generation of high-quality novel samples drives the research of representation learning and generative models. Directly representing and generating images in the pixel space stimulates various research such as generative adversarial networks [2, 7, 18, 28], flow models [11, 33, 41, 45], energy-based models [12, 13, 64, 69], and diffusion models [24, 39, 52, 53]. As the resolution grows, it becomes increasingly difficult to accurately regress the pixel values. And this challenge usually has to be addressed through hierarchical model architectures [28, 70] or at a notably high cost [24]. Moreover, while demonstrating outstanding generated image quality, GAN models suffer from issues including insufficient mode coverage [36] and training instability [21].

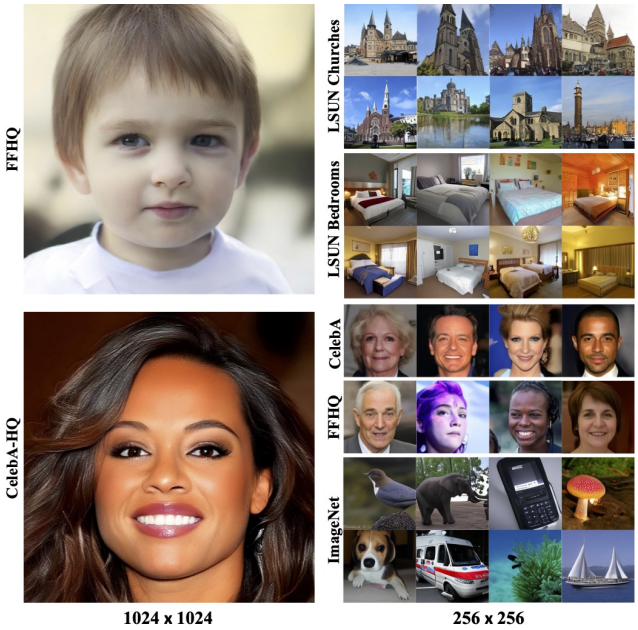


Figure 1. Examples of generated images with different resolutions using the proposed method.

Representing and generating images in a learned latent space [32, 40, 47] provides a promising alternative. Latent diffusion [47] performs denoising in a latent feature space with a lower dimension than the pixel space, therefore reduces the cost of each denoising step. However, regressing the real-value latent representations remains complex and demands hundreds of diffusion steps. Variational auto-encoders (VAEs) [23,32,46] generate images without any iterative steps. However, the static prior of the latent space re-

stricts the expressiveness, and can lead to posterior collapse. To achieve higher flexibility of the latent distribution without significantly increasing the modeling complexity, VQ-VAE [59] introduces a vector-quantized latent space, where each image is represented as a sequence of indexes, each of which points to a vector in a learned codebook. The prior over the vector-quantized representations is then modeled by a trained sampler, which is usually parametrized as an autoregressive model. The success of VQ-VAE stimulates a series of works that model the discrete latent space of codebook indexes with different models such as accelerated parallel autoregressive models [8] and multinomial diffusion models [6, 20]. VQ-based generative models demonstrate surprising image synthesis performance and model coverage that is better than the more sophisticated methods like GANs without suffering from issues like training instability. However, the hard restriction of using one codebook index to represent each image patch introduces a trade-off on the codebook size, as a large enough codebook to cover more image patterns will introduce an over-complex multivariate multinomial latent distribution for the sampler to model.

In this research, we explore a compact yet expressive representation of images in a binary latent space, where each image patch is now represented as a binary vector, and the prior over the discrete binary latent codes is effectively modeled by our improved binary diffusion model tailored for multivariate Bernoulli distribution. Specifically, the bi-directional mappings between images and the binary representations are modeled by a feed-forward autoencoder with a binary latent space. Given an image, the encoder now outputs the normalized parameters of a multivariate Bernoulli distribution, from which a binary representation of this image is sampled, and fed into the decoder to reconstruct the image. The discrete sampling in multivariate Bernoulli distribution does not naturally permit gradient propagation. We find that a simple straight-through gradient copy [4, 17] is sufficient for high-quality image reconstruction while maintaining high training efficiency.

With images compactly represented in the binary latent space, we then introduce how to generate novel samples by modeling the prior over binary latent codes of images. To overcome the shortcomings of many existing generative models such as being uni-directional [44, 59] and the non-regrettable greedy sampling [6, 8], we introduce binary latent diffusion that generates the binary representations of novel samples by a sequence of denoising starting from a random multivariate Bernoulli distribution. Performing diffusion in a binary latent space, modeled as multivariate Bernoulli distribution, reduces the need for precisely regressing the target values as in Gaussian-based diffusion processes [24, 47, 53], and permits sampling at a higher efficiency. We then introduce how to progressively reparametrize the prediction targets at each denoising step as the residual between the inputs and the desired samples, and train the proposed binary latent diffusion models to predict such 'flipping probability' for improved training and sampling stability.

We support our findings with both conditional and unconditional image generation experiments on multiple datasets. We show that our method can deliver remarkable image generation quality and diversity with more compact latent codes, larger image-to-latent resolution ratios, as well as fewer sampling steps and faster sampling speed. We present some examples with different resolutions generated by the proposed method in Figure 1.

We organize this paper as follows: Related works are discussed in Section 2. In Section 3, we introduce binary image representations by training an auto-encoder with a binary latent space. We then introduce in Section 4 binary latent diffusion, a diffusion model for multi-variate Bernoulli distribution, and techniques tailored specifically for improving the training and sampling of binary latent diffusion. We present both quantitative and qualitative experimental results in Section 5 and conclude the paper in Section 6.

2. Related Work

Diffusion models and image generation. Diffusion models [52] are proposed as a flexible way of modeling complex data distribution using tractable families of probability distributions. Denoising diffusion probabilistic models (DDPM) [24] build a reparameterization of the learning objective that connects diffusion probabilistic models and denoising score matching [54]. The resulting diffusion models based on multivariate Gaussian distributions scale up the generation resolution and achieve high-quality image generation with quality comparable with strong generative families such as GANs [7, 30, 37], score-based models [54–56], and energy-based models [14, 19], without suffering from issues like mode collapse and training instability. Further improvements have been proposed to encourage higher log-likelihoods [39], reduce computational demands [47], and extend diffusion models to broader applications including text-to-image generation [42, 48], planning [27], super-resolution [49], temporal data modeling [1, 34, 58], and adversarial robustness [5, 61]. Particularly, diffusion models with discrete states are advocated in [3] for text generation. A series of structured transition matrices are introduced in [3], among which diffusion with an absorbing state is further extended to image generation with VQ codes in [6].

Discrete representations. Modern deep neural networks trained with back-propagation and gradient descent prevalently advocate continuous features across all layers. However, discrete representations still demonstrate their unique advantages and applications. [4] discusses several approaches to estimating the gradient through stochastic neu-

rons. Among the four approaches introduced in [4], the practically simplest one with heuristic gradient copies is further adopted in learning fully discrete vector-quantized auto-encoders [59], which builds the foundations for many follow-up methods [6, 8, 16, 44]. [9] proposes rehearing training samples for continual learning in a binary latent space. [17] introduces a binary latent space by a hard threshold instead of sampling in our methods. And the sampling with random hyperplane rounding in [47] can hardly be scaled to a higher resolution.

3. Binary Image Representations

Given an image dataset, we begin with learning the bi-directional mappings between images and their binary representations. This is achieved by training an auto-encoder with a binary latent space, where the binary code of each image is obtained as a sample of a multivariate Bernoulli distribution inferred from the image. Specifically, denoting an image as $\mathbf{x} \in \mathbb{R}^{h \times w \times 3}$, we train an image encoder $\mathbf{\Psi}$ that outputs the unnormalized parameters for the corresponding multivariate Bernoulli distribution $\mathbf{\Psi}(\mathbf{x})$. A Sigmoid non-linearity σ is then adopted to normalize the parameters $\mathbf{y} \in \mathbb{R}^{\frac{h}{k} \times \frac{w}{k} \times c} = \sigma(\mathbf{\Psi}(\mathbf{x}))$, where h and w denote the spatial resolution of the image, k is the downsampling factor of the encoder $\mathbf{\Psi}$, and c is the number of encoded feature channels.

To obtain the binary representations of images, we perform Bernoulli sampling given the normalized parameters $\mathbf{z} = \text{Bernoulli}(\mathbf{y})$. Note that the Bernoulli sampling operation here does not naturally permit gradient propagation through it, and prevents the end-to-end training of the encoder-decoder architecture. In practice, we consistently observe that a straight-through gradient estimation by direct copying the gradients and skipping the non-differentiable sampling in backpropagation can maintain both stable training and superior performance. The straight-through gradient estimation can be easily implemented by a surrogate function as:

$$\tilde{\mathbf{z}} = \oslash(\mathbf{z}) + \mathbf{y} - \oslash(\mathbf{y}), \tag{1}$$

where $\oslash(\cdot)$ denotes the stop gradient operation. $\tilde{\mathbf{z}}$, which is binary and identical to $\mathbf{z}$, will be sent to the following decoder network $\mathbf{\Phi}$ for image reconstruction. The gradient propagated back from the decoder $\mathbf{\Phi}$ to $\tilde{\mathbf{z}}$ is directly sent to $\mathbf{y}$, which is differentiable w.r.t. the encoder $\mathbf{\Psi}$ and permits the end-to-end training of the entire auto-encoder with a discrete binary latent space.

The reconstruction of the image is obtained using a decoder network $\mathbf{\Phi}$ as $\hat{\mathbf{x}} = \mathbf{\Phi}(\tilde{\mathbf{z}})$. The overall framework of the binary autoencoder is visualized in Figure 2. With the gradient surrogate function ensuring the end-to-end gradient propagation, the network is trained by minimizing the final objective

$$\mathcal{L} = \sum_{i}^{||\mathcal{C}||} \omega_i \mathcal{C}[i](\hat{\mathbf{x}}, \mathbf{x}), \tag{2}$$

where $\mathcal{C}$ denotes a collection of loss functions such as mean squared error, perception loss, and adversarial loss. And ω_i denotes the weights that balance each loss term.

4. Multivariate Bernoulli Diffusion

Given an image dataset and the corresponding binary latent representations of each image, we then discuss how to effectively model the prior over the binary latent codes with a parametrized model $p_\theta(\mathbf{z})$, from which novel samples of the binary latent codes can be efficiently sampled. To do so, we introduce binary latent diffusion, a diffusion model tailored specifically for multivariate Bernoulli distribution accompanied with improvement techniques that promote stable and effective training and sampling.

A diffusion model is usually established by first defining a T-step diffusion process consisting of a sequence of variation distributions $q(\mathbf{z}^t|\mathbf{z}^{t-1})$, with $t \in \{1, \dots T\}$. Each variation distribution in $q(\mathbf{z}^t|\mathbf{z}^{t-1})$ is defined to progressively add noise to $\mathbf{z}^{t-1}$, so that with sufficient steps T and a valid noise scheduler defining $q(\mathbf{z}^t|\mathbf{z}^{t-1})$, the final state $q(\mathbf{z}^T|\mathbf{z}^0)$ converges to a known random distribution that is easy to evaluate and sample from, and conveys almost no valid information of $\mathbf{z}^0$. Specifically, in our case, we are interested in modeling the prior distribution of the binary latent code of images. Therefore, we define the starting point of the diffusion process with our latent code distribution $\mathbf{z} \sim q(\mathbf{z}^0)$, where $q(\mathbf{z}^0)$ is a multivariate Bernoulli distribution as the prior of latent codes. And the full diffusion process q can be defined as

$$q(\mathbf{z}^{1:T}) := \prod_{t=1}^{T} q(\mathbf{z}^t|\mathbf{z}^{t-1}), \quad \text{with} \tag{3}$$

$$q(\mathbf{z}^t|\mathbf{z}^{t-1}) = \mathcal{B}(\mathbf{z}^t; \mathbf{z}^{t-1}(1-\beta^t) + 0.5\beta^t), \tag{4}$$

where $\mathcal{B}$ denotes a multivariate Bernoulli distribution and β^t defines the noise scale at each step t. With sufficient steps T, and β^t at adequate scales, the forward diffusion process will converge to $\mathcal{B}(\mathbf{z}^T; 0.5)$, which is a random Bernoulli distribution that is easy to sample from. Figure 3 illustrates an example of the binary latent diffusion process.

Given an arbitrary time step t and a sample $\mathbf{z}^0$, the posterior distribution can be easily obtained as

$$\begin{aligned} q(\mathbf{z}^t|\mathbf{z}^0, \mathbf{z}^T) =& \mathcal{B}(\mathbf{z}^t; k^t\mathbf{z}^0 + b^t), \quad \text{with} \\ & k^t = \prod_{i=1}^{t}(1-\beta^i), \\ & b^t = (1-\beta^t)b^{t-1} + 0.5\beta^t, \quad \text{and} \\ & b^1 = 0.5\beta^1, \end{aligned} \tag{5}$$

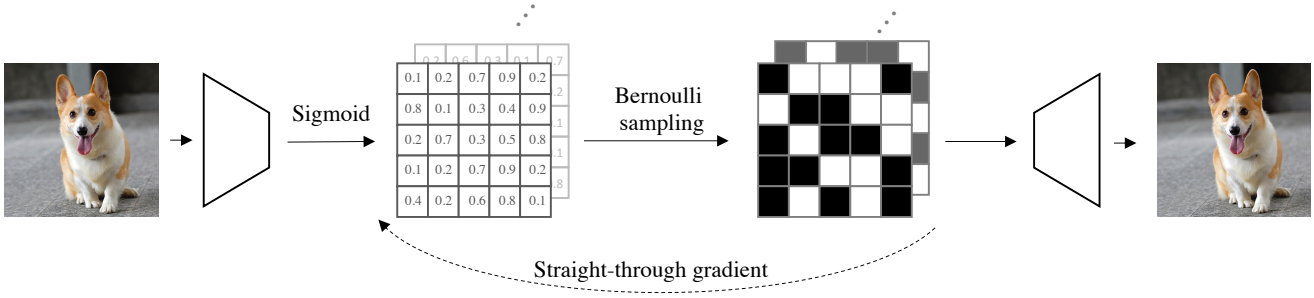


Figure 2. Illustration of the proposed binary auto-encoder. The gradient flow estimated with a straight-through surrogate function is denoted as the dashed line.

where k^t and b^t here jointly define the accumulated noise scale till time step t. The Markov chain and the analytic form of the posterior in (5) permit stochastic sampling in each training batch.

The noise schedulers of the proposed binary latent diffusion process can be constructed by either simply defining the noise scale β^t at each step, or directly defining the accumulated noise factors k^t and b^t. Note that even with k^t and b^t directly defined, the corresponding β^t can still be obtained as $\beta^t = 1 - \frac{k^t}{k^{t-1}}$. We present in Appendix Section A discussions on different choices of noise schedulers.

With the forward diffusion process properly defined. the goal now is to train a function f_θ with $\hat{\mathbf{z}}^{t-1} = f_\theta(\mathbf{z}^t, t)$ to model the reverse diffusion (denoising) process

$$p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t) = \mathcal{B}(\mathbf{z}^{t-1}; f_\theta(\mathbf{z}^t, t)), \tag{6}$$

which allows sampling to be performed by reversing a sample from $q(\mathbf{z}^T)$ to a sample in $q_\theta(\mathbf{z}^0)$. The diffusion process q and the denoising process p jointly define a variational

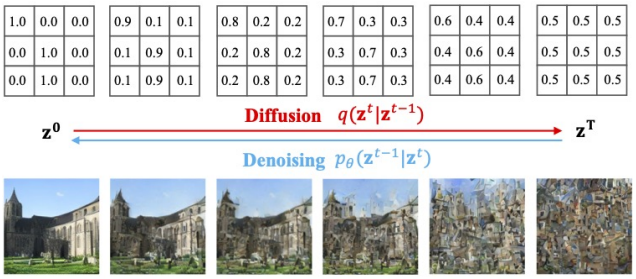


Figure 3. Illustration of the binary latent diffusion and denoising processes. A binary representation is progressively diffused towards a random multivariate Bernoulli distribution with 0.5 everywhere.

auto-encoder [32], with the variational lower bound:

$$\begin{aligned}\mathcal{L}_{\text{vlb}} &:= \mathcal{L}_0 + \sum_{t=1}^{T-1} \mathcal{L}_t + \mathcal{L}_T \\ &:= -\log p_\theta(\mathbf{z}^0|\mathbf{z}^1) \\ &+ \sum_{t=1}^{T-1} \text{KL}(q(\mathbf{z}^{t-1}|\mathbf{z}^t, \mathbf{z}^0)||p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t)) \\ &+ \text{KL}(q(\mathbf{z}^T|\mathbf{z}^0)||p(\mathbf{z}^T)).\end{aligned} \tag{7}$$

Note that in this paper we focus on the diffusion process with fixed predefined noise scheduler β^t, therefore the third term on the RHS of (7) does not depend on θ and is always nearly 0. All the KL divergence and likelihood terms in (7) can be analytically calculated in a closed form as all distributions involved are multivariate Bernoulli.

4.1. Binary Latent Diffusion Reparameterization

To learn the reverse diffusion process, a straightforward way is to train a neural network f_θ using (7) to model $p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t)$. According to (5), each $\mathbf{z}^t$ can be considered as a *linear interpolation* between $\mathbf{z}^0$ and $\mathbf{z}^T$, whose parameters can take arbitrary numbers between 0 and 1 depending on the noise scheduler. Therefore, directly training f_θ to model $p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t)$ can be challenging as the model needs to accurately regress such sophisticated interpolations.

Predicting $\mathbf{z}^0$. Inspired by [20], we first introduce a reparameterization to the prediction target as $p_\theta(\mathbf{z}^0|\mathbf{z}^t)$ and train the model f_θ to directly predict $\mathbf{z}^0$ as $\hat{\mathbf{z}}^0 = f_\theta(\mathbf{z}^t, t)$ at each step. During sampling, at each step, with $p_\theta(\mathbf{z}^0|\mathbf{z}^t)$ predicted, we can recover the corresponding $p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t)$ by:

$$\begin{aligned}p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t) =& q(\mathbf{z}^{t-1}|\mathbf{z}^t, \mathbf{z}^0 = \mathbf{0}) p_\theta(\mathbf{z}^0 = \mathbf{0}|\mathbf{z}^t) \\ &+ q(\mathbf{z}^{t-1}|\mathbf{z}^t, \mathbf{z}^0 = \mathbf{1}) p_\theta(\mathbf{z}^0 = \mathbf{1}|\mathbf{z}^t),\end{aligned} \tag{8}$$

with

$$q(\mathbf{z}^{t-1}|\mathbf{z}^t, \mathbf{z}^0) = \frac{q(\mathbf{z}^t|\mathbf{z}^{t-1}, \mathbf{z}^0) q(\mathbf{z}^{t-1}|\mathbf{z}^0)}{q(\mathbf{z}^t|\mathbf{z}^0)}. \tag{9}$$

Specifically, with a binary code $\mathbf{z}^t$ and the noise scheduler

defined in (5), we have

$$p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t) = \mathcal{B}\big(\mathbf{z}^{t-1}|\frac{[(1-\beta^t)\mathbf{z}^t + 0.5\beta^t] \odot [k^t f_\theta(\mathbf{z}^t, t) + 0.5b^t]}{\mathbf{Z}}\big), \tag{10}$$

where

$$\mathbf{Z} = [(1-\beta^t)\mathbf{z}^t + 0.5\beta^t] \odot [k^t f_\theta(\mathbf{z}^t, t) + 0.5b^t] + [(1-\beta^t)(1-\mathbf{z}^t) + 0.5\beta^t] \odot [k^t(1 - f_\theta(\mathbf{z}^t, t)) + 0.5b^t] \tag{11}$$

is a normalization term that guarantees valid probabilities in (10), and $\odot$ denotes element-wise product. With the introduced reparameterization, the prediction targets at all time steps become $\mathbf{z}^0$, which is strictly binary, and eases the training according to our observations.

Predicting the residual. A reparameterization to the prediction target as the residual is introduced in [24], which bridges the connections between diffusion and denoising score matching [54] and improves the sampling results. In the proposed binary latent diffusion, the decreasing noise scales result in $\mathbf{z}^t$ getting closer to $\mathbf{z}^0$ as t decreases. Thus, the sampling in the binary latent space naturally favors fewer flippings to the binary codes as t gets closer to $t = 0$. To better capture this *sparsity* in predictions and stabilize the sampling to prevent divergence, we show that, while predicting the residual is considered non-trivial for discrete-state diffusion models [26], the prediction targets can be further reparametrized as the residual between $\mathbf{z}^0$ and $\mathbf{z}^t$ in our binary latent diffusion. Specifically, we train $f_\theta(\mathbf{z}^t, t)$ to fit $\mathbf{z}^t \oplus \mathbf{z}^0$, where $\oplus$ denotes the element-wise logic XOR operation. The model f_θ is now trained to predict the 'flipping probability' of the binary code. And the prediction targets remain strictly binary.

Final training objective. The final simplified training objective can be formulated as:

$$\mathcal{L}_{\text{residual}} = \mathbb{E}_{t,\mathbf{z}^0}\text{BCE}(f_\theta(\mathbf{z}^t, t), \mathbf{z}^t \oplus \mathbf{z}^0), \tag{12}$$

where $\text{BCE}(\cdot,\cdot)$ denotes the binary cross-entropy loss. In practice, following [39], we find it beneficial to set the final learning objective as a combination of $\mathcal{L}_{\text{residual}}$ and $\mathcal{L}_{\text{vlb}}$ as

$$\mathcal{L} = \mathcal{L}_{\text{residual}} + \lambda\mathcal{L}_{\text{vlb}}, \tag{13}$$

with λ is a small number.

Sampling temperatures. In practice, the prediction of the residual $f_\theta(\mathbf{z}^t, t)$ is implemented as $f_\theta(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t))$, where $\mathcal{T}_\theta$ is a plain transformer that outputs the unnormalized flipping probability, which is then normalized by a Sigmoid function σ. During sampling, we can manually adjust the sampling diversity by inserting a temperature τ which turns the prediction to $f_\theta(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t)/\tau)$. Note that τ is only used to adjust the diversity during post-training sampling, and is not included as a hyperparameter in training. Examples of how τ affects the sample diversity are shown in Figure 4.

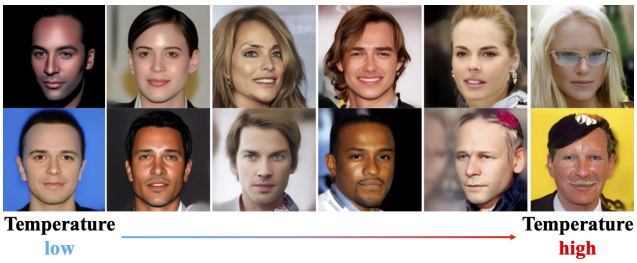


Figure 4. Temperature values directly decide the diversity of the sampled images. High temperatures improve coverage and reveal rare patterns such as reflections. But overly high temperatures can result in sampling procedures failing to converge and produce artifacts. Demonstrative results are obtained on the 1024×1024 CelebA-HQ experiment with temperature values from 0.2 to 1.2.

Classifier-free guidance. Classifier-free guidance [25] is introduced to improve the generation fidelity and promote high correspondence to the conditions for conditional diffusion models. While it is previously believed that classifier-free guidance can hardly be extended to diffusion models, we show that the reparameterization to the prediction target as the binary residuals allows classifier-free guidance to apply seamlessly to the propose binary latent diffusion for improved image fidelity in conditional image generation. In conditional image generation, we introduce an extra condition token c that carries the conditions such as class embedding in ImageNet class-conditional image generation or text embedding in the text-to-image generation. The model performs class-conditional prediction as $f^{\text{c}}_\theta(\mathbf{z}^t, t, c) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t, c))$. An unconditional prediction can be drawn by simply dropping the condition token c as $f^{\text{u}}_\theta(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t))$. The final prediction at each step with classifier-free guidance can then be implemented as $f_\theta(\mathbf{z}^t, t, c) = \sigma((1+\omega)\mathcal{T}_\theta(\mathbf{z}^t, t, c) - \omega\mathcal{T}_\theta(\mathbf{z}^t, t))$, where ω is a non-negative scalar controlling the strength of the guidance. Note that same the temperature scale τ, the guidance strength ω is only effective in the sampling stage and is not involved in training, which allows as to arbitrarily adjust the sampling quality without retraining the models. While it is inevitable that the classifier-free guidance doubles that computation cost at each sampling step due to the extra unconditional predictions, we consistently observe performing classifier-free guidance allows us to skip sampling steps. For example, performing only a quarter of the sampling steps enhanced with classifier-free guidance reduce the overall computation cost by half, and does not noticeably reduce the sample quality. We present in Appendix Figure M results generated with different scales of classifier-free guidance.

4.2. Comparisons with Existing Methods

In this section, we briefly discuss the advantages of adopting the proposed binary representations over other alternatives of latent space image representations. As visualized in Figure 5, vector-quantized latent space [16, 59] represents each image patch as a discrete index, or equivalently, a one-hot vector. The one-hot vector then multiplies with the learned codebook to obtain the feature representation of an image patch. Image representations in continuous latent space [47] can be interpreted in a similar way by simply treating the first weight matrix in the decoder network as the learned codebook. The real-value latent code of an image patch performs arbitrary linear combinations of vectors in the codebook and is able to cover diverse feature space even with a low-dimensional code for higher efficiency. Our method with binary representation strives for a balance between these two methods by restricting the vectors composing the codebook to be binary. On the one hand, the binary composition of a codebook offers a much more diverse and flexible composition of features compared to the vector-quantized representations. For example, a compact 32-bit binary vector can represent over 4,000 million patterns, which is much larger than the 1,024 patterns commonly used in vector-quantized representations [6, 8, 32]. As we will further show empirically in Section 5, this improved coverage of patterns allows for higher expressiveness and enables high-resolution image generation that can hardly be accomplished by VQ representations at high quality. On the other hand, the binary restriction guarantees that the representations remain compact. As we will show in Section 5.3, our 8k-bit binary representations perform comparably with a latent diffusion model [47] with 131k-bit representations.

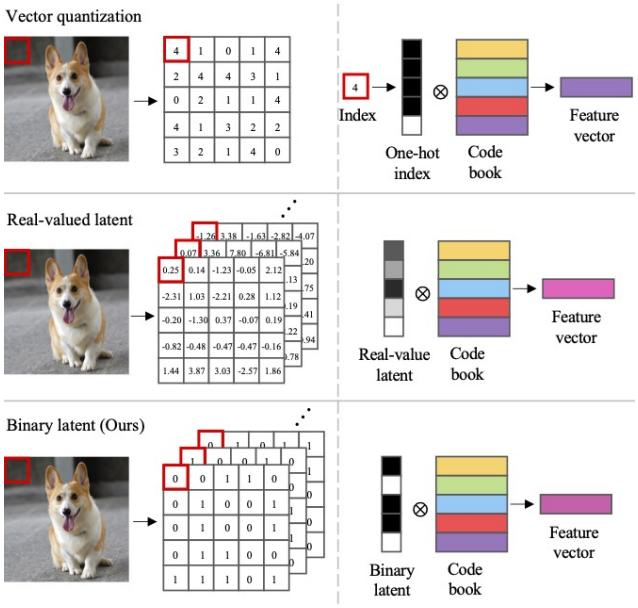


Figure 5. Interpreting different latent representations from a codebook view. Vector-quantized (top) methods perform one-hot selections of codes. Latent diffusion composes codes with arbitrary linear combinations. Our method also permits composition of codes but restricts the composition to be binary for improved efficiency.

5. Experiments

In this section, we present both unconditional and conditional image generation experiments with multiple datasets. Following common practice [6, 8], we train a plain transformer network [60] to parametrize the sampler $\mathcal{T}_\theta$. We use a temperature of $\tau = 0.9$ as the default setting for sampling, and $\lambda = 0.1$ in all the experiments, if not otherwise specified. For the number of channels in the binary latent space, we use $c = 32$ for 256×256 unconditional image generation, $c = 64$ for class-conditional image generation, and 1024×1024 high-resolution unconditional image generation. Additional details regarding the implementation and evaluation metrics can be found in Appendix Section B.

5.1. Unconditional Image Generation

In this section, we present results and comparisons on unconditional image generation experiments with multiple datasets including LSUN Bedrooms and Churches [66], FFHQ [29], and CelebA-HQ [28]. To allow for fair comparisons, we first present 256×256 image generation. We apply our method on a $16 \times 16 \times 32$ latent space, which corresponds to the same $16\times$ downsampling of the spatial resolution as in methods like [6, 8]. Note that as we further show in Section 5.3, the improved expressiveness of our binary representation compared to the vector-quantized representation allows our method to perform well with a larger downsampling ratio such as $32\times$. Each image patch is now represented with a 32-bit binary vector, and an image is represented as an 8k-bit tensor, which is more compact compared to the 131k-bit representations in Latent Diffusion models [47].

For comprehensive evaluations and comparisons, we adopt Fréchet Inception Distance (FID) and precision-recall [35] as the evaluation metrics. We present comparisons against various state-of-the-art methods in Table 1. We conduct comparisons by generating 50,000 samples and comparing them with the corresponding training dataset in every experiment. Quantitative comparisons against state-of-the-art methods are presented in Table 1. Despite using much fewer denoising steps compared to latent diffusion models (LDM) [47] and absorbing diffusion models [6], our binary latent diffusion models achieve comparable image generation quality with desirable diversity. Note that our results are obtained without any test-time accelerations or skipping denoising steps. While LDM [47] results are reported with

Table 1. Quantitative comparisons on unconditional image generation with LSUN Bedrooms, LSUN Churches, FFHQ, and CelebA-HQ. We also provide comparisons on the number of denoising steps in *training*. All results are obtained with a resolution of 256×256.

Methods	Steps	Bedrooms			Churches		
		FID ↓	P ↑	R ↑	FID ↓	P ↑	R ↑
DCT [38]	-	6.40	0.44.	0.56	7.56	0.60	0.48
ImageBART [15]		5.51	-	-	7.32	-	-
VQGAN [16]	256	6.35	0.61	0.33	7.81	0.67	0.29
DDPM [24]	1,000	6.36	-	-	7.89	-	-
PGGAN [28]	-	8.34	0.43	0.40	6.42	0.61	0.38
StyleGAN [29, 30]	-	2.35	0.55	0.48	3.86	0.60	0.43
Absorbing [6]	256	3.64	0.67	0.38	4.07	0.71	0.45
LDM [47]	1,000	2.95	0.66	0.48	4.02	0.64	0.52
Ours	16	4.11	0.62	0.44	4.66	0.66	0.48
Ours	64	3.85	0.65	0.44	4.36	0.68	0.50

Methods	Steps	FFHQ			CelebA-HQ		
		FID ↓	P ↑	R ↑	FID ↓	P ↑	R ↑
VQGAN	400	-	-	-	10.2	-	-
StyleGAN [29]	1	4.16	0.71	0.46	-	-	-
UDM [31]	1,000	5.52	-	-	7.16	-	-
Absorbing [6]	256	6.11	0.73	0.48	-	-	-
LDM [47]	1,000	4.98	0.73	0.50	5.11	0.72	0.49
Ours	16	6.48	0.71	0.45	7.80	0.67	0.44
Ours	64	5.85	0.73	0.50	6.24	0.71	0.48

accelerated 200 DDIM [53] steps at test time, our method still demonstrates significantly higher sampling efficiency as we will show in Section 5.3. We present further qualitative results in Appendix Section C.1.

High-resolution image generation in one shot. Compared to vector-quantization based methods, the proposed binary latent space allows each image patch to be represented as a composition of features, and therefore potentially permits latent space image generation with a large downsampling (image-to-latent resolution) ratio. And the larger resolution ratio permits image generation with larger image resolution without increasing the difficulty of modeling the latent prior. To show this, we directly present high-resolution image generation experiments with FFHQ and CelebA-HQ by increasing the target image resolution to 1024×1024. Generating such high-resolution images in a discrete latent space used to be handled by a multi-stage hierarchy of the latent representations [44, 65, 67]. We show that in our method, high-resolution image generation can be directly achieved by a single latent space without significantly scaling up the size of the latent space. Modeling the discrete latent space of such high-resolution images was previously handled by sophisticated designs, such as a three-layer hierarchy with 32×32, 64×64, and 128×128 latent space as in [44], which leads to over 5,000 sampling steps. In our method, we adopt a single latent space with a resolution of 32×32. Note that this setting gives a higher $32\times$ downsampling ratio since the commonly used $16\times$ downsampling ratio will lead to a 64×64 latent resolution that is unaffordable for the plain transformer architecture we adopt. As we will further discuss and compare in Section 5.3, we consistently notice little compromise in the image quality with such a higher downsampling ratio. We present quantitative comparisons in Table 3 and qualitative results in Appendix Section C.2. To the best of our knowledge, our binary latent diffusion is the first diffusion model that generates such high-resolution image *in one-shot*, i.e., without any latent hierarchy [44] or multi-stage upsampling [42].

5.2. Conditional Image Generation

We adopt the ImageNet-1K dataset for image generation with class labels as conditions. In addition to FID and precision-recall, we follow the common practice and include the inception score (IS) [50] and the classification accuracy score (CAS) [43] with ResNet-50 [22] as the evaluation metrics. Following the same settings as in the unconditional image generation experiments, we generate images at a resolution of 256×256, with a latent space of $16\times16\times64$. We present quantitative comparisons against state-of-the-art methods in Table 2. Note that the results of our method are achieved without any auxiliary sampling strategies or accelerations such as classifier [10] or classifier-free guidance [25]. Our results demonstrate comparable image quality to state-of-the-art image generation methods and appealing sample diversity. The effectiveness is validated by the improved CAS scores. We present qualitative results in Figure 6 and Appendix Section C.3.

5.3. Discussions

Image Reconstruction. In this section, we present comparisons of the image reconstruction quality with different ways of formulating the latent space. To assess the reconstruction quality, we use the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) [62] as the metrics to measure the reconstruction quality. We measure the performance on the validation set of the LSUN bedrooms dataset at a resolution of 256×256, and present the results in Table 4. Compared to VQ-based representation, our method permits better image reconstruction quality with a compact 32-bit binary representation for each image patch. The advantage is further amplified when a more compact spatial size (8×8), which corresponds to a $32\times$ downsampling ratio is adopted. The higher downsampling ratio allows our method to generate high-resolution images in a one-shot fashion as reported in Section 5.1.

Alternative samplers. To show the advantages of using the binary latent diffusion for the modeling of the prior over the multivariate Bernoulli latent distribution, we present in Table 5 quantitative results with the samplers parametrized by alternative architectures including autoregressive models as in [59] and absorbing diffusion [6]. We report FID results on 256×256 unconditional image generation with

Table 2. Quantitative comparisons on 256×256 class-conditional image generation. We report both the top-1 and top-5 accuracy of CAS. Using the official training set obtains Top-1 and Top-5 accuracy of 76.6% and 93.1%, respectively.

Methods	Params	Step	FID ↓	IS ↑	P ↑	R ↑	Top-1 ↑	Top-5 ↑
DCT [38]	738M	>1024	36.51	-	0.36	0.67	-	-
BigGAN-deep [7]	160M	1	6.95	198.2	0.87	0.28	43.99	67.89
Improved DDPM [39]	280M	250	12.26	-	0.70	0.62	-	-
ADM [10]	554M	250	10.94	101.0	0.69	0.63	-	-
VQVAE2 [44]	13.5B	5120	31.11	∼ 45	0.36	0.57	54.83	77.59
VQGAN [16]	1.4B	256	15.78	78.3	-	-	-	-
MaskGIT [8]	227M	8	6.18	182.1	0.80	0.51	63.14	84.45
LDM [47]	400M	250	10.56	103.49	0.71	0.62	-	-
Ours	172M	64	8.52	158.12	0.72	0.64	65.07	85.60

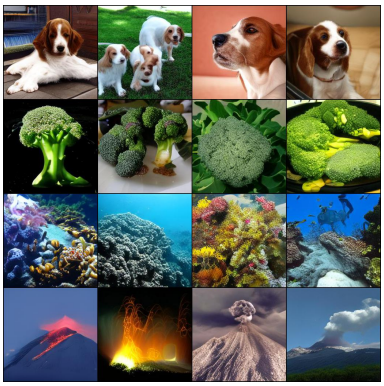


Figure 6. Class-conditional image generation with classes: 218, 937, 973, 980.

Table 3. FID comparisons on 1024×1024 high-resolution image generation with FFHQ and CelebA-HQ. † indicates results we obtain based on the official implementations.

Methods	Steps	FFHQ ↓	CelebA-HQ ↓
StyleSwin [68]	1	5.07	4.43
StyleGAN-XL [51]	1	2.02	-
Diffusion StyleGAN2 [63]	<1,000	2.83	-
LDM† [47]	1,000	10.09	8.68
Absorbing† [6]	1,024	14.12	13.82
Ours	64	6.53	5.73
Ours	256	6.07	5.22

Table 4. Image reconstruction quality with different formats of latent code. The sizes are formulated as height × width × code (or codebook) size.

Methods	Size	PSNR ↑	SSIM ↑
VQ [59]	16x16x1024	20.08 ± 1.84	0.62 ± 0.09
Real value [47]	16x16x16	24.08 ± 4.22	0.70 ± 0.12
Binary (Ours)	16x16x32	24.02 ± 2.11	0.75 ± 0.06
VQ [59]	8x8x2048	17.20 ± 1.45	0.53 ± 0.09
Real value [47]	8x8x64	22.74 ± 1.87	0.62 ± 0.15
Binary (Ours)	8x8x128	22.82 ± 1.84	0.64 ± 0.07

LSUN Churches and FFHQ. The proposed binary latent diffusion model tailored for the multivariate Bernoulli distribution achieves the best results on modeling the prior over the binary latent image representations. Compared to autoregressive models and absorbing diffusion models which perform greedy sampling, our method allows the prediction at previous denoising steps to be modified and improved in later steps, and prevents error accumulation across steps.

Efficiency. One of the major advantages of the proposed efficient image generation with binary latent diffusion is that good results can be achieved with fewer steps of denoising, which results in a faster sampling speed in practice.

Table 5. FID results with alternative samplers for the modeling of the prior over the multivariate Bernoulli latent distribution.

Samplers	Steps	Churches ↓	FFHQ ↓
Autoregressive	256	7.25	8.23
Absorbing diffusion	256	5.44	7.64
Binary Diffusion (Ours)	64	4.36	5.85

Table 6. Comparisons on image generation speed with seconds per sample (s/sample). All results are obtained by averaging 1,000 times of sampling with a batch size of 1. 64s and 16s denote 64 and 16 denoising steps, respectively.

Methods	StyleGAN-2	Absorbing	LDM	DDPM	Ours 64s	Ours 16s
s/sample	0.04	3.40	15.68	63.85	0.82	0.20

We compare sampling speed against SyleGAN-2 [30], absorbing diffusion [6], latent diffusion [47], and DDPM [24] in Table 6. Our method demonstrates clear advantages in image sampling speed compared to other diffusion-based methods, and further closes the gap between the speed of GANs and diffusion models. We further present comparisons on the generation results with different denoising steps against absorbing diffusion models [6] in Figure 7. Our method demonstrates noticeably higher robustness to the number of steps, and fair results can be obtained with as few as 8 denoising steps. Meanwhile, we consistently observe that it is extremely hard to train Gaussian-based diffusion models such as latent diffusion models [47] and DDPM [24] to perform reasonably with fewer than 100 denoising steps in *training*.

Ablation studies. We present in Table 7 quantitative results that validate the values of λ in (13) and the prediction targets of our binary latent diffusion. We consistently observe that the value of λ correlates noticeably with the sample diversity, while a very large value of λ, such as $\lambda = 1.0$, fre-

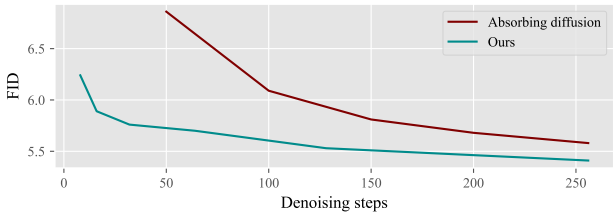


Figure 7. FID with different denoising steps on the 256×256 LSUN churches experiment. For fair comparisons, we use $\tau = 1.0$ in all experiments.

Table 7. Performance (FID / Recall) with different values of λ and prediction targets. All results are obtained on the LSUN Churches experiments.

λ values	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$
FID / R	5.07 / 0.42	4.68 / 0.46	**4.36 / 0.50**	5.12 / 0.48
Targets	$\mathbf{z}^{t-1}$	$\mathbf{z}^0$	$\mathbf{z}^t \oplus \mathbf{z}^0$	
FID / R	5.09 / 0.48	4.72 / 0.50	**4.36 / 0.50**	-

quently causes sampling divergence, which results in samples that appear as random compositions of multiple images, and thus degrades the quality. Setting the prediction targets as $\mathbf{z}^{t-1}$ corresponds to using only $\mathcal{L}_{\text{vlb}}$ as the supervision. While different parameterizations to the prediction targets achieve similar sample diversity, the proposed reparameterization of the targets with the flipping probability $\mathbf{z}^t \oplus \mathbf{z}^0$ achieves the best overall performance as it noticeably reduces the samples with strong artifacts caused by potentially diverged sampling according to our observations.

Image inpainting. Compared to the autoregressive-based samplers, one of the clear advantages of diffusion models is that the iterative sampling process in diffusion models does not assume any specific generation order of the spatial positions. This means that after training the generative models, we can generate images with partial observations as conditions. We present in Figure 8 examples of inpainting results generated by our models. Additional inpainting results can be found in Appendix Section C.4.

6. Conclusion

In this paper, we presented representing and generating images in a binary latent space. We learned binary image representations by training an auto-encoder with a multi-variate Bernoulli latent distribution and simple gradient copying to bypass the non-differential Bernoulli sampling operation. Compared to the real-value image representations in either the pixel or latent space, we showed that binary representation allows for a compact representation with the corresponding distribution that can be effectively modeled by a binary latent diffusion model. Compared to vector-quantized discrete representation, the binary representation in our work enables a higher expressiveness that permits high-resolution images to be generated without any multi-stage latent hierarchy. We examined our idea on multiple datasets with both conditional and unconditional image generation experiments. The comparable results to the recent state-of-the-art methods validated the effectiveness of the proposed method of representing and generating images in a binary latent space.

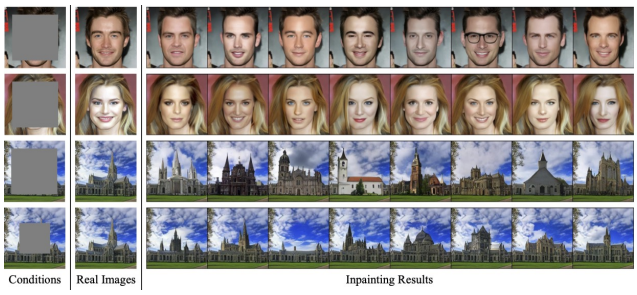


Figure 8. Inpainting results. Given partial observations as conditions, our learned model can generate diverse samples with strong correspondence to the conditions. Diverse results can be obtained even with strong conditions given (last row). Zoom in for details.

Social impacts. As a framework for representing and generating images, the proposed binary latent diffusion model shares with other generative models the risk of malicious uses such as deepfake. And the generated sample may exhibit bias residing in the training datasets.

References

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint*, 2022. 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 1

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021. 2

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint*, 2013. 2, 3

[5] Tsachi Blau, Roy Ganz, Bahjat Kawar, Alex Bronstein, and Michael Elad. Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint*, 2022. 2

[6] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. *arXiv preprint*, 2021. 2, 3, 6, 7, 8

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ICLR*, 2019. 1, 2, 8

[8] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 2, 3, 6, 8

[9] Kamil Deja, Paweł Wawrzyński, Daniel Marczak, Wojciech Masarczyk, and Tomasz Trzciński. Binplay: A binary latent autoencoder for generative replay continual learning. In *IJCNN*, 2021. 3

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 7, 8

[11] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *ICLR*, 2015. 1

[12] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *ICML*, 2020. 1

[13] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *NeurIPS*, 2019. 1

[14] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *NeurIPS*, 2019. 2

[15] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *NeurIPS*, 2021. 7

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 3, 6, 7, 8, 1, 2

[17] Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Latent bernoulli autoencoder. In *ICML*, 2020. 2, 3

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1

[19] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *ICLR*, 2020. 2

[20] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022. 2, 4

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *NeurIPS*, 2017. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7, 2

[23] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 1

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1, 2, 5, 7, 8

[25] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*, 2022. 5, 7

[26] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *NeurIPS*, 2021. 5

[27] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint*, 2022. 2

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018. 1, 6, 7

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 6, 7

[30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2, 7, 8

[31] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *arXiv preprint*, 2021. 7

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, 2013. 1, 4, 6

[33] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE T-PAMI*, 2020. 1

[34] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *ICLR*, 2021. 2

[35] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019. 6, 1

[36] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017. 1

[37] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 2

[38] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint*, 2021. 7, 8

[39] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 1, 2, 5, 8

[40] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *NeurIPS*, 2020. 1

[41] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *JMLR*, 2021. 1

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint*, 2022. 2, 7

[43] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. *NeurIPS*, 2019. 7, 2

[44] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 2019. 2, 3, 7, 8

[45] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015. 1

[46] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014. 1

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3, 6, 7, 8

[48] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint*, 2022. 2

[49] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE T-PAMI*, 2022. 2

[50] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 2016. 7, 2

[51] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH*, 2022. 8

[52] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 1, 2

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2, 7

[54] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019. 2, 5

[55] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 2020. 2

[56] 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. 2

[57] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1

[58] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *NeurIPS*, 2021. 2

[59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017. 2, 3, 6, 7, 8

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 6

[61] Jinyi Wang, Zhaoyang Lyu, Dahua Lin, Bo Dai, and Hongfei Fu. Guided diffusion model for adversarial purification. *arXiv preprint*, 2022. 2

[62] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 7

[63] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint*, 2022. 8

[64] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *ICML*, 2016. 1

[65] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint*, 2021. 7

[66] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint*, 2015. 6

[67] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *ICLR*, 2022. 7

[68] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *CVPR*, 2022. 8

[69] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *ICLR*, 2017. 1

[70] Yang Zhao, Jianwen Xie, and Ping Li. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *ICLR*, 2020. 1

A. Noise Scheduler

We show in Figure A that with different noise schedulers, how 1 (blue curves) and 0 (red curves) in the binary latent representations progressively reach 0.5 in the diffusion processes. We use the *Linear* noise scheduler in all the experiments.

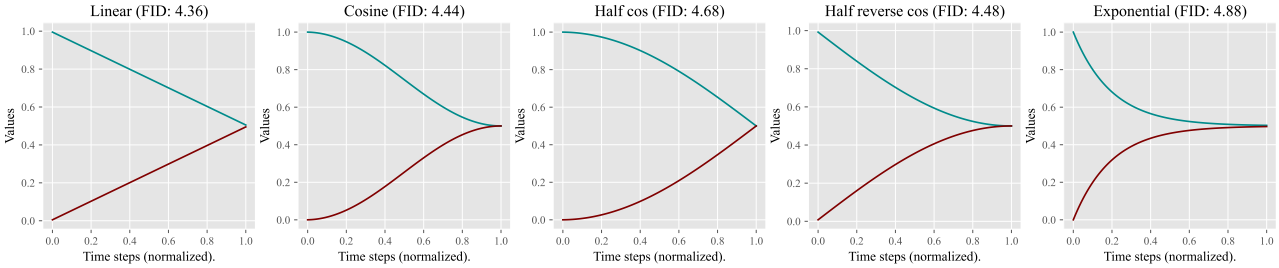


Figure A. Noise schedulers. FID numbers are obtained with the 256×256 LSUN Churches experiments.

B. Implementation Details

We use Adam as the default optimizer for all experiments. For the training of the binary auto-encoder in Section 3, we use a consistent learning rate of 5×10^{-4} with a linear learning rate warm-up for 10k iterations, and a batch size of 8. For $\mathcal{C}$ in (2), we use mean squared error, perception loss, and adversarial loss, with an equal $\omega_i = 1.0$ for each term.

For the training of the binary latent diffusion models in Section 4, we use a consistent learning rate of 1×10^{-4} with a linear learning rate warm-up for 10k iterations. All models are trained for 20K iterations. We do not use any additional regularizations such as drop-out or weight decay. We summarize the general training and sampling of the proposed binary diffusion models in Algorithm 1 and Algorithm 2, respectively.

All 256×256 unconditional binary latent diffusion models are trained with a batch size of 32. All 1024×1024 unconditional binary latent diffusion models are trained with a batch size of 16. All 256×256 conditional binary latent diffusion models are trained with a batch size of 256.

All training is conducted on cloud servers with V100 GPUs. All the speeding testing is conducted on a single RTX3090 GPU.

B.1. Network Architectures

For the binary auto-encoder in Section 3, we adopt a standard architecture using convolutional layers enhanced with self-attention layers, which is nearly identical to the one used in [16]. The only major difference is that the vector quantization layer is replaced by our binary latent layer implementing (1). This architecture gives a downsampling ratio of $16\times$. For the 1024×1024 high-resolution image generation experiments, of which the downsampling ratio raises to $32\times$, we simply insert one more downsampling block and one more upsampling block into the architecture.

For the conditional image generation experiments with ImageNet-1K, we train transformer networks with 24 layers and 768 feature channels with 12 heads in the self-attention layers. For the unconditional image generation experiments, we train transformer networks with 24 layers and 512 feature channels with 8 heads in the self-attention layers. In both conditional and unconditional image generation experiments, we use an extra time embedding token to specify the time step t. And the class labels in conditional image generation with ImageNet are also specified by an additional token. Both class tokens and time-step tokens are trainable. We initialize the trainable position embedding in the transformer networks with 2D sine embedding.

B.2. Evaluation Metrics

FID. Fréchet Inception Distance (FID) is a commonly used metric for evaluating the quality of the generated images. It estimates the distance between two image distributions by comparing the mean and standard deviation of the deep image features extracted by a trained Inception network [57]. We conduct comparisons by generating 50,000 samples and compare them with the corresponding training dataset in every experiment.

PR. To comprehensively evaluate both the quality and mode-coverage of the generated samples compared to the training datasets, we further include precision-recall [35] as additional performance measurements.

IS. Inception score [50] is a commonly used metric for evaluating the performance of class-conditional image generation. It favors generated images with low entropy of label predictions and diverse labels given a pretrained Inception-V3 network.
CAS. Classification accuracy score [43] works by first training a ResNet-50 [22] using the generated image across classes, and measuring the results of applying the trained classifier to the ImageNet validation set. CAS offers a comprehensive measurement of the generative quality as a robust classifier demands the generated images used for network training to be both diverse and of high quality.

Algorithm 1 Training procedure. We assume unconditional image generation with a batch size of *one* for the sake of discussion. The described training process can be easily extended to practical cases with arbitrary batch sizes by batching multiple samples.

Given: Trained encoder $\mathbf{\Psi}$; Binary diffusion model f_θ parametrized by $\mathcal{T}_\theta$; An image dataset $\mathbf{X}$.
Given: Diffusion steps T; Noise scheduler defined by $\{k^t\}_{t=1}^T$ and $\{b_t\}_{t=1}^T$; Training steps I; and λ in (13).
Initializing $\mathcal{T}_\theta$.
for Step $i = 1 : I$ **do**
  Sampling image $\mathbf{x} \sim \mathbf{X}$, and time step $t \sim \{1, \dots, T\}$.
  Obtaining binary code $\mathbf{z}^0 = \text{Bernoulli}(\sigma(\mathbf{\Psi}(\mathbf{x})))$.
  Obtaining $\mathbf{z}^t$ using $\mathbf{z}^0$, t, and noise scheduler with (5).
  Predicting flipping probability $f_\theta(\mathbf{z}^t, t)$.
  Obtaining predicted $\mathbf{z}^0$ as $p_\theta(\mathbf{z}^0) = (1 - \mathbf{z}^t) \odot f_\theta(\mathbf{z}^t, t) + \mathbf{z}^t \odot (1 - f_\theta(\mathbf{z}^t, t))$.
  Obtaining predicted $p_\theta(\mathbf{z}^{t-1})$ using $p_\theta(\mathbf{z}^0)$ and $\mathbf{z}^t$ with (8).
  Calculating loss $\mathcal{L}$ using (13).
  Backpropagating $\mathcal{L}$ and updating θ.
end for
Return Binary diffusion model f_θ.

Algorithm 2 Sampling procedure. We assume unconditional image generation with a batch size of *one* for the sake of discussion.

Given: Trained decoder $\mathbf{\Phi}$; Trained binary diffusion model f_θ.
Given: Diffusion steps T; Noise scheduler defined by $\{k^t\}_{t=1}^T$ and $\{b_t\}_{t=1}^T$; Temperature τ; Latent dimension specified by h', w', c, e.g., $h' = w' = 16, c = 32$ for the 256×256 image generation experiments.
Sampling $\mathbf{z}^T = \text{Bernoulli}(\mathbf{z}^{\text{init}})$, where $\mathbf{z}^{\text{init}} \in \mathbb{R}^{h' \times w' \times c}$ and contains 0.5 only.
for Step $t = T : 2$ **do**
  Predicting $p_\theta(\mathbf{z}^{t-1})$ with $f_\theta(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t)/\tau)$ and (8).
  Sampling $\mathbf{z}^{t-1} = \text{Bernoulli}(p_\theta(\mathbf{z}^{t-1}))$
end for
Return the sampled image as $\mathbf{\Phi}(\mathbf{z}^{t-1})$.

C. Additional Qualitative Results

C.1. Unconditional Image Generation

We present additional qualitative results of 256×256 LSUN Bedrooms and LSUN Churches unconditional image generation experiments in Figure B and Figure C, and qualitative comparisons of FFHQ unconditional image generation against Absorbing Diffusion models [6] and Latent Diffusion Models [47] in Figure D.

C.2. High-resolution Image Generation

We present additional qualitative results of 1024×1024 unconditional image generation experiments in Figure E and Figure F.

C.3. Conditional Image Generation

We present additional qualitative results of class-conditional image generation experiments and comparisons with BigGAN-deep [7], VQ-VAE-2 [44], VQGAN [16] and MaskGIT [8] in Figure I and Figure L.

C.4. Image Inpainting

We present additional image inpainting results with different mask patterns in Figure N.

C.5. Nearest Neighbours

To further show that our model is generating novel samples instead of overfitting to the training datasets, we compare the generated images with the corresponding training datasets using LPIPS, and visualize the top-10 nearest neighbours in Figure O.

Figure B. Additional unconditional image generation results and comparisons at 256×256 with the LSUN Churches dataset.

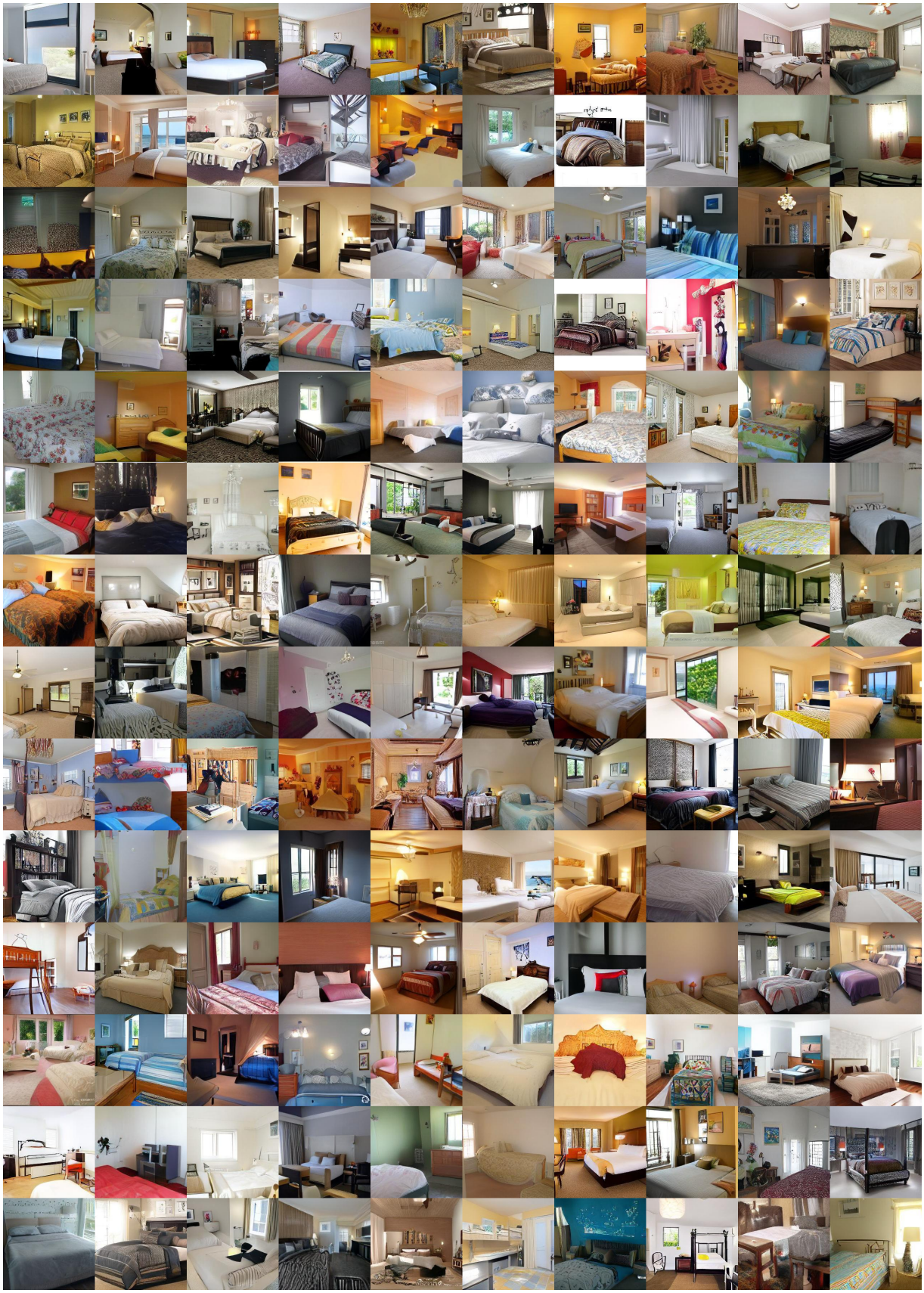


Figure C. Additional unconditional image generation results and comparisons at 256×256 with the LSUN Bedrooms dataset.

(a) Absorbing diffusion.

(b) Latent diffusion.

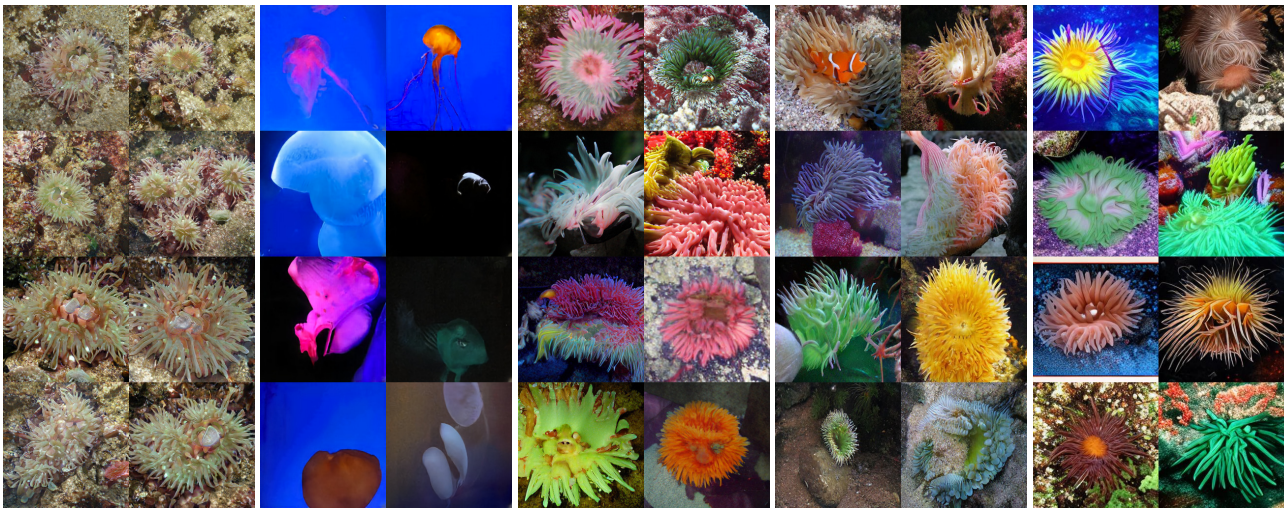
(c) Ours.

Figure D. Additional unconditional image generation results and comparisons at 256×256 with the FFHQ dataset.

Figure E. Additional high-resolution image generation results at 1024×1024 with the FFHQ dataset ($\tau = 0.8$).

Figure F. Additional high-resolution image generation results at 1024×1024 with the CelebA-HQ dataset ($\tau = 0.8$).

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.



(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

Figure I. Qualitative comparisons on class-conditional image generation with ImageNet class IDs: 22, 108, and 11.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

Figure L. Qualitative comparisons on class-conditional image generation with ImageNet class IDs: 141, 154, and 1.

Figure M. Label-conditioned image generation with different scales of classifier-free guidance (ω). Larger ω improves the sample quality at the cost of lower diversity.

Figure N. Conditional image inpainting with different masking patterns.

Figure O. Top-10 nearest neighbours in the training datasets of our generated samples. Results show that our model is not overfitting to the training datasets.