arXiv:2505.20984v1 [eess.IV] 27 May 2025

Generative Image Compression by Estimating Gradients of the Rate-variable Feature Distribution

Minghao Han, Weiyi You, Jinhua Zhang, Leheng Zhang, Ce Zhu, Shuhang Gu*
University of Electronic Science and Technology of China
{minghao.hmh, shuhanggu}@gmail.com

Abstract

While learned image compression (LIC) focuses on efficient data transmission, generative image compression (GIC) extends this framework by integrating generative modeling to produce photo-realistic reconstructed images. In this paper, we propose a novel diffusion-based generative modeling framework tailored for generative image compression. Unlike prior diffusion-based approaches that indirectly exploit diffusion modeling, we reinterpret the compression process itself as a forward diffusion path governed by stochastic differential equations (SDEs). A reverse neural network is trained to reconstruct images by reversing the compression process directly, without requiring Gaussian noise initialization. This approach achieves smooth rate adjustment and photo-realistic reconstructions with only a minimal number of sampling steps. Extensive experiments on benchmark datasets demonstrate that our method outperforms existing generative image compression approaches across a range of metrics, including perceptual distortion, statistical fidelity, and no-reference quality assessments. The code is available here.

1 Introduction

Image compression techniques aim to encode images into the shortest possible bit streams for efficient data transmission. Recent studies have developed learned image compression (LIC) methods [4, 5, 9, 15, 16, 27, 29, 32, 33, 35] that achieve superior rate-distortion performance compared to conventional codecs [6, 41]. However, the compression process tends to sacrifice the details in images, and the optimization objective, i.e., the rate-distortion loss function, restricts the ability of decoder to restore the lost image details, which finally results in a blurred unrealistic recovered image. The pursuit of realism has given rise to a range of generative image compression methods.

Generative image compression (GIC) methods introduce impressive generative modeling techniques, e.g., generative adversarial networks (GANs) [14], vector-quantized variational autoencoders (VQ-VAEs) [13], and diffusion-based models [18, 39], to obtain the capability of prior distribution modeling, thus improving human-perceptual performance with guaranteed fidelity. In their seminal work, perceptual loss, GANs [3, 31, 44], and its discriminator variants [34] are first utilized to finetune the basic image compression network, which allows the decoder in the autoencoder to complement the image details. For the VQ-based method [20, 26, 30], the prior distribution of compressed latent variables is modeled as classification probabilities according to the codebook. Compared to GAN-based and VQ-based methods, diffusion modeling decouples their once-through distribution transformations into asymptotic stochastic processes, which significantly enhances generation performance. The existing diffusion-based methods [8, 19, 25, 47] can be viewed as a form of "post-processing", where diffusion models are employed to enhance the compressed data—an indirect approach aimed at supplementing the lost details. However, such an indirect method may not fully harness the potential of diffusion modeling.

*Corresponding Author

Preprint. Under review.

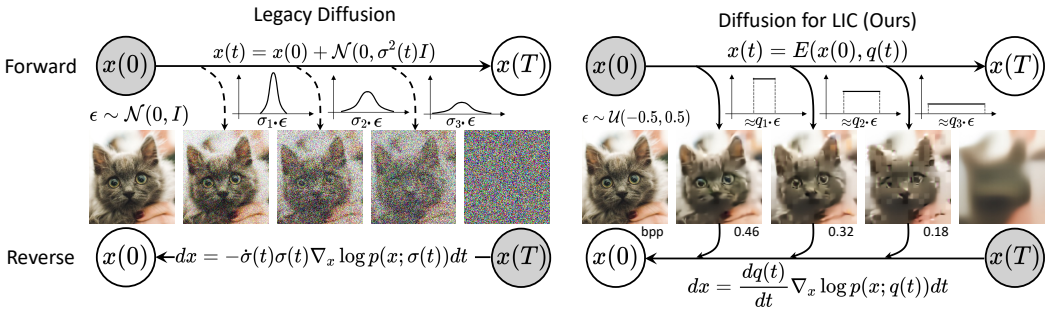


Figure 1: **Left:** The forward and reverse process of legacy diffusion for comparison. The legacy diffusion consists of transforming data to a simple noise distribution and a reverse ODE to restore the original data. **Right: The overview pipeline of our method.** The forward process is defined as the entropy model compressing the data. With the bit rates decrease, the compressed images retain less details (please *zoom in* for better visualization). We can reverse such ODE at any intermediate time to recover the data under various compression rates. This makes a full use of the benefits of diffusion modeling and an organic integration of LIC and diffusion.

Revisiting diffusion modeling in the context of image generation, a forward process is constructed by progressively corrupting data with increasing Gaussian noise, and generative modeling is achieved by training a sequence of probabilistic models to reverse the corruption process. Rate-variable quantization, i.e. quantize feature with different quantization factors, could establish a similar process of gradually distorting high quality data. As shown in Fig. 1, legacy diffusion model adds Gaussian noise to distort the data while the corruption process of rate-variable quantization is often formulated as additive noise with uniform distribution [4, 5]. Following this, we argue that the rate-variable compression forms a particular forward process with additive noise, and that the goal of restoring the details oriented to the state before compression can be achieved by reversing it. Taking inspiration from generative modeling through stochastic differential equations (SDEs) [40], we describe the aforementioned forward process (compression) and reverse process with the help of SDEs. That is, it is possible to construct intermediate probability distributions for a stochastic process using a variety of compression rates, and we train a neural network to model the score of a compression-rate-dependent marginal distribution of the training data corrupted by compressing. Moreover, with the adopted rate-variable quantization parameterizing the compression process into a single quantization factor, this exactly facilitates the formulation of our customized forward and reverse processes. The above framework establishes a sequence of distributions for generative modeling, result in a more natural integration of diffusion model and learned image compression, which could support rate-variable generative compression with only a minimal number of reverse steps.

Taken together, we propose generative image compression by estimating gradients of rate-variable feature distribution, offering a novel perspective beyond conventional paradigms. Our method constitutes a customized diffusion framework designed for image compression, featuring both **flexible rate adjustment** and **high-fidelity image reconstruction**. Our rate-variable model is able to outperform current fixed-rate state-of-the-art methods on a range of perceptual metrics. We believe that this work will inspire further innovations in broader areas, particularly in the modification of the diffusion modeling core to suit a variety of specific research scenarios.

2 Background

In this section, we present the related work and the existing technologies that we adopt in our proposed method. First, we review a brief theory of denoising score matching modeling [39] as a preliminary for our specialized diffusion models. Next, we describe the adopted rate-variable compression strategy for the sake of understanding our proposed framework. Finally, we introduce recent representative GIC methods that integrate with generative modeling.

2.1 Denoising Score Matching

Given the data distribution $p_{data}(\boldsymbol{x})$, the idea of score matching is to find the score/gradients of the data distribution $p_{data}(\boldsymbol{x})$, that is, the fastest growing direction of the log probability density of the data $s(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$. To obtain the score, Song et al. [39] proposed the denoising score matching and noise conditional score networks (NCSN): To consider the family of mollified distributions $p(\boldsymbol{x}; \sigma)$ obtained by adding i.i.d. Gaussian noise of standard deviation σ to the data, the diffusion model sequentially denoises from a pure Gaussian noise $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$ into intermediate states $\boldsymbol{x}_i \sim p(\boldsymbol{x}; \sigma_i)$ with decreasing noise levels $\sigma_i > \sigma_{i+1}$. It ends up with the target image $\boldsymbol{x}_N$ with $\sigma_N = 0$ at the place where the log probability density is maximized. The optimization objective is to minimize the L_2 denoising error for samples drawn from p_{data} separately for every σ. Defining a neural denoiser $D(\boldsymbol{x}, \sigma)$:

$$\mathbb{E}_{\boldsymbol{y} \sim p_{data}} \mathbb{E}_{\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} ||D(\boldsymbol{y} + \boldsymbol{n}, \sigma) - \boldsymbol{y}||_2^2, \tag{1}$$

where $\boldsymbol{y}$ is a training image and $\boldsymbol{n}$ is the sampled Gaussian noise. In this vein, the score of the state with noise of deviation σ can be formulated as:

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma) = (D(\boldsymbol{x}; \sigma) - \boldsymbol{x}) / \sigma^2. \tag{2}$$

For the sampling of the diffusion modeling above, Song et al. [40] present a stochastic differential equation (SDE) to unify the processes of noise removal and addition into an integral theoretical framework. An overview of legacy diffusion with a simplified form of SDEs, ordinary differential equations (ODEs), is shown in Fig. 1. To solve the ODE is to substitute Eq. 2 into it and calculate the numerical integration, i.e., taking finite steps over discrete time intervals.

2.2 Learned Lossy Image Compression

Learned lossy image compression methods are built upon a variational autoencoder (VAE) framework proposed by Ballé et al. [5]. The VAE based LIC framework mainly comprises an autoencoder and an entropy model. The autoencoder conducts nonlinear transforms between the image space, i.e., input: $\boldsymbol{x}$, reconstruction: $\hat{\boldsymbol{x}}$, and the latent representation space, i.e., latent representation: $\boldsymbol{y}$, quantized latents: $\hat{\boldsymbol{y}}$; while, the entropy model minimizes the code length by estimating the probability distribution of latent representations.

Rate-distortion optimization. In their seminal work, Ballé et al. [4] established the end-to-end rate-distortion minimization framework. It showed that the smallest average code length of latent representation is given by the Shannon cross entropy [37] between the actual marginal distribution and a learned entropy model. Thus, the optimization objective appears a rate-distortion trade-off between rate loss $R(\cdot)$ and distortion loss $D(\cdot)$:

$$\mathcal{L}_{R-D} = \mathcal{R}(\hat{\boldsymbol{y}}) + \lambda \cdot \mathcal{D}(\hat{\boldsymbol{x}}, \boldsymbol{x}), \tag{3}$$

where λ controls whether the network is more concerned about the quality of the recovery or the compression efficiency. The optimization problem under the fixed λ in Eq. 3 is employed for the fixed-rate paradigm, driving the encoder to adjust the information reserved in the latent variables $\boldsymbol{y}$.

Quantization scaling. Nevertheless, the LIC network trained under this schedule only yields a single compression rate result, which limits the design space of diffusion modeling. A general solution is to randomly sample λ during training, finetuning a vanilla fixed-rate LIC network to support rate-variable compression [10, 11, 42, 45]. In this work, we adopt quantization scaling to control the compression rate via the entropy model. Under this circumstance, the role of "information reducer" moves from the encoder to the entropy model, separating the autoencoder as a stand-alone component. This strategy focuses on the quantization operation $\lceil \boldsymbol{y} \rfloor$. Since codecs only work on integers, the entropy model quantized the latent representation for bit stream transmission. The quantization operation can be regarded as adding a uniform noise in a range of $[-0.5, 0.5]$:

$$\hat{\boldsymbol{y}} = \lceil \boldsymbol{y} \rfloor \Rightarrow \boldsymbol{y} + \mathcal{U}(-0.5, 0.5). \tag{4}$$

The idea of quantization scaling is to scale the latent representation $\boldsymbol{y}$ before quantization. Given a scale parameter q, this can be formulated as:

$$\begin{aligned} \hat{\boldsymbol{y}}_q &= \lceil \boldsymbol{y}/q \rfloor \cdot q \\ &\Rightarrow [(\boldsymbol{y}/q) + \mathcal{U}(-0.5, 0.5)] \cdot q \\ &= \boldsymbol{y} + \mathcal{U}(-0.5, 0.5) \cdot q. \end{aligned} \tag{5}$$

Referring to Eq. 4, quantization scaling can be considered as scaling the uniform noise to control the information gap between original latents $\boldsymbol{y}$ and quantized latents $\hat{\boldsymbol{y}}_q$. At this point, we obtain a rate-variable compression network with only one parameter to adjust the compression rate. This facilitates our following theory construction and its implementation.

Generative compression. Generative modeling methods, e.g., generative adversarial nets (GANs), vector-quantized variational autoencoders (VQ-VAEs), and diffusion-based models, probabilistically model real data distributions. Generative compression methods exploit them to supplement the prior distribution to the compressed data, thereby producing photo-realistic reconstructed images. In their seminal work, GANs and perceptual loss are first utilized to enhance the fidelity of the reconstructed image. Agustsson et al. [3] first introduced GANs to generate extra details for shaper decompressed images. Subsequent work has studied further the variants of discriminators, such as patch-GAN [31], local label prediction [34], and realism guidance [1]. In the context of the VQ-based method, a vector-quantized variational autoencoder is used to replace or wrap the vanilla LIC variational autoencoder. Jia et al. [20] move the lossy compression framework into the latent space of VQ-VAEs. Li et al. [26] modify the basic VQ-VAEs to obtain a rate-variable generative image compression network. Given the surprising results of diffusion modeling in the generative domain, recent work has attempted to incorporate the advantages of the diffusion model into image compression. The existing diffusion-based methods employ two primary training approaches. The first approach involves training a denoising network itself [19, 47], while the second approach involves fine-tuning a pretrained large-scale diffusion model [8, 25]. With respect to the integration of diffusion models within the LIC network, the prevailing methodologies encompass two approaches: the replacement of the decoder in the compression autoencoder with the diffusion model [47], and the subsequent addition of the denoising network following the completion of the compression procedure [19]. Existing diffusion-based methods indirectly take advantage of diffusion modeling. We expect to establish a novel framework to seamlessly combine LIC and diffusion, realizing the potential of diffusion modeling. Deriving from the nature of the diffusion modeling is a delicate way.

3 Method

In the context of learned image compression technologies, the loss of detailed information occurs in the encoder; while the decoder decodes the broken data to images. Restricted to the training strategy and network paradigm, it is difficult for the decoder to restore the lost image details. We propose generative image compression by estimating gradients of rate-variable feature distribution to reverse the compression process, assisting the restoration of lost details. Note that we consider our proposed diffusion modeling specialized in image compression to be regarded as a form of "generalized diffusion". Consequently, our analysis lives outside the confines of the diffusion theoretical frameworks but borrows some analytical processes and ideas from these frameworks [21, 39, 40]. Building a standard diffusion model involves two key components, i.e., degraded data construction for training a denoising network (forward process) and the sampling design (reverse process). In this section, we construct the diffusion modeling specialized in image compression following the above modules.

3.1 Compression Forward Process

Given an image $\boldsymbol{x}_0$, the legacy diffusion obtains degraded data $\boldsymbol{x}_i$ by adding various levels (denoted by deviations $\sigma_N = \sigma_{max} > \cdots > \sigma_1 > \sigma_0 = 0$) of Gaussian noise so that $p(\boldsymbol{x}_i|\boldsymbol{x}_0) \sim \mathcal{N}(\boldsymbol{x}_0, \sigma_i^2\mathbf{I})$. This corrupts data to varying degrees, and the network is trained to estimate the score function $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}_i)$ to restore the original data. In essence, learning to restore from corrupted data, also known as the reverse process, enables the network to perform score matching. The training of the reverse process is relaxed and facilitated by data at various levels of corruption.

For the image compression task, we expect to equip the reverse neural network with the ability to restore the compression-corrupted data. Following the standard diffusion process, we replace the data corruption of adding noises with rate-variable compression. Defining a pretrained rate-variable entropy model E, the compression process can be formulated as:

$$\boldsymbol{x}_t = E(\boldsymbol{x}_0, q_t), \tag{6}$$

where q_t denotes the parameter of quantization scaling as we mentioned in Sec 2.2. Similar to the deviation σ in legacy diffusion, q reflects the extent of data corruption.

The reverse neural network D_θ inverts the corruption due to compression:

$$\hat{\boldsymbol{x}}_0 = D_\theta(\boldsymbol{x}_t, q_t), \tag{7}$$

where $\hat{\boldsymbol{x}}_0$ denotes the approximated recovered data produced by network D_θ. When q_t is small, $\hat{\boldsymbol{x}}_0$ should appear close to real $\boldsymbol{x}_0$ and vice versa. Thus, our optimization objective is to minimize the distance between ground truth $\boldsymbol{x}_0$ and approximated $\hat{\boldsymbol{x}}_0$:

$$\mathbb{E}_{\boldsymbol{x}_0 \sim p_{data}, q_t \sim \epsilon_q} ||\boldsymbol{x}_0 - D_\theta(E(\boldsymbol{x}_0, q_t), q_t)||^2, \tag{8}$$

where $\boldsymbol{x}_0$ is randomly drawn from a dataset and q_t is sampled from the distribution ϵ_q. Due to the equal significance of all bit rates in the context of rate-variable compression tasks, we set the distribution $\epsilon_q := \mathcal{U}(q_{min}, q_{max})$, where q_{max} is the maximum supported scale parameter for the entropy model and q_{min} is lower than the minimum support, a constant very close to 0. When the sampled q_t is not supported by the entropy model, referring to Sec. 2.2 we take a simulated quantization method as $\boldsymbol{x}_t = \boldsymbol{x}_0 + \mathcal{U}(-0.5, 0.5) \cdot q_t$.

3.2 Reverse Process Design

In their pioneering work, Song et al. [40] present a stochastic differential equation (SDE) that maintains the desired distribution p as the sample $\boldsymbol{x}$ evolves over time. Within this theoretical framework, the sampling process is defined as a reverse SDE, which facilitates the derivation of sampling formulas oriented to better generation results. Following this, we endeavor to formulate a sampling approach that aligns with the diffusion modeling of our design for image compression. To that end, we first express the aforementioned diffusion process as an ordinary differential equation (ODE). Subsequently, a reverse ODE is derived and extended to SDEs for enhancing the quality of the reconstructed images.

ODE formulation. Although our diffusion forward process is implemented by the entropy model E, we can approximate it (see Sec. 2.2) as follows:

$$\boldsymbol{x}_t \Rightarrow \lceil \boldsymbol{x}_0 / q(t) \rfloor \cdot q(t) \Rightarrow \boldsymbol{x}_0 + \mathcal{U}(-0.5, 0.5) \cdot q(t). \tag{9}$$

For the sake of representing differential equations, we move the subscript q_t into the function brackets $q(t)$. We define the ODE evolving a sample $\boldsymbol{x}_a \sim p(\boldsymbol{x}_a; q(t_a))$ from time t_a to t_b yields a sample $\boldsymbol{x}_b \sim p(\boldsymbol{x}_b; q(t_b))$, which is satisfied by

$$\mathrm{d}\boldsymbol{x} = -\frac{\mathrm{d}q(t)}{\mathrm{d}t} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t)) \mathrm{d}t, \tag{10}$$

where $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t))$ is the score function in our theoretical framework. To reverse it, we define the reverse neural network D_θ producing a result $\hat{\boldsymbol{x}}_0$ approximated to $\boldsymbol{x}_0$. Thus, the score is obtained by $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t)) = (\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_t)/q(t)$. Following the previous works, the Euler's method is adopted as the discrete solution during sampling. We substitute the equation above to Eq. 10 and use Euler's solver (see Appendix A):

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{q(t_i) - q(t_{i+1})}{q(t_i)} (\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_i). \tag{11}$$

Nevertheless, the reverse ODE in isolation remains not an optimal solution for sampling, since we find that the deterministic sampling based on the derivation of ODE produces a suboptimal image quality compared to SDE-based stochastic sampling.

Stochastic sampling. The reverse ODE, corresponding to deterministic sampling, has been observed to result in a worse performance [38, 40] than stochastic sampling, i.e., reverse SDEs. Following the existing work [21], we extend the ODEs Eq. 10 to SDEs and append the random items from Langevin sampling:

$$\mathrm{d}\boldsymbol{x} = -\frac{\mathrm{d}q(t)}{\mathrm{d}t} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t)) \mathrm{d}t + \alpha(t)\, \mathrm{d}\omega_t \pm \alpha(t)\, \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t)), \tag{12}$$

where ω_t is the standard Wiener process to inject the randomness to the sampling, the last term is the deterministic decaying item, and $\alpha(t)$ is the hyperparameter set empirically. Generally speaking, randomness injection is implemented by adding standard Gaussian noises in the legacy diffusion model framework [39]; however, our framework lives outside the standard diffusion theory, so we discuss the form of randomness injection and hyperparameter schedules in the following section.

Algorithm 1: The procedure of our diffusion-based compression of data $\boldsymbol{x}$

Input: $\boldsymbol{x}, q_0$ **Given:** $\mathcal{E}(\boldsymbol{x}), \mathcal{D}(\bar{\boldsymbol{y}}), E_\phi(\boldsymbol{y}, q), D_\theta(\hat{\boldsymbol{y}}, q), q_{i\in\{0,\cdots,N\}}, \alpha, \omega$

Encoding

- $\boldsymbol{y}_0 \leftarrow \mathcal{E}(\boldsymbol{x})$ ▷ $\boldsymbol{x}$ to be compressed, q_0 setting the compression rate
- $\hat{\boldsymbol{y}}_q \leftarrow E_\phi(\boldsymbol{y}_0, q_0)$ ▷ Approximate $p(\hat{\boldsymbol{y}}_q)$ and compress $\boldsymbol{y}_0$ with scale q_0
- $\hat{\boldsymbol{y}}_q \xleftrightarrow{p(\hat{\boldsymbol{y}}_q)}$ bit stream ▷ Entropy code using $p(\hat{\boldsymbol{y}}_q)$

Decoding

- **for** $i \in \{0, \cdots, N-1\}$ **do** ▷ Reverse directly from $\bar{\boldsymbol{y}}_0 := \hat{\boldsymbol{y}}_q$
  - $\boldsymbol{d}_i \leftarrow (D_\theta(\bar{\boldsymbol{y}}_i, q_i) - \bar{\boldsymbol{y}}_i)/q_i$ ▷ Evaluate the score $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q_i)$ at q_i
  - $\bar{\boldsymbol{y}}_{i+1} \leftarrow \bar{\boldsymbol{y}}_i + (q_i - q_{i+1})\boldsymbol{d}_i$ ▷ Take Euler step from q_i to q_{i+1}
  - **sample** $\boldsymbol{\epsilon}_i \sim \omega$ ▷ Inject randomness for stochastic sampling
  - $\bar{\boldsymbol{y}}_{i+1} \leftarrow \alpha(\boldsymbol{\epsilon}_i - \boldsymbol{d}_i)$
- **end**
- $\hat{\boldsymbol{x}} \leftarrow \mathcal{D}(\bar{\boldsymbol{y}}_N)$

return $\hat{\boldsymbol{x}}$

Randomness injection. Previous results [38, 40] show that finding an optimal setting of stochasticity is significant and that the setting should be empirically determined with respect to specific diffusion models. In legacy diffusion models, the amount of stochasticity grows with the number of sampling steps. Recent work [21] suggests that a non-uniform growing schedule surpasses a linear one. Combining the above with the properties of compression tasks, we propose a randomness injection schedule for the hyperparameter α_t:

$$\alpha_t = \beta \cdot \sqrt{q_t - q_{min}}, \tag{13}$$

where q_t adjusts the compression rate, q_{min} is the minimus quantization scale parameter supported by the entropy model, and β controls the growth rate of stochastic injections with q_t. Moreover, due to the specialization of our model, the specific form of stochasticity is worth arguing, e.g., standard Gaussian noise, uniform noise, and noise drawn from the entropy model. Our analysis in Sec. 4.2 discuss the setting of β and the form of randomness injection ω.

3.3 Image Compression by Estimating Gradients of Rate-variable Feature Distribution

Training the rate-variable compression entropy model. To alleviate the computational overhead, we move the compression entropy model and the diffusion network into the latent space of the pretrained VAE [36], where the original image is compressed initially by a factor of four in spatial dimensions. We train the rate-variable entropy model E_ϕ via optimizing a rate-distortion loss:

$$\mathcal{L}_{R-D} = -\log p(\hat{\boldsymbol{z}}) - \log p(\hat{\boldsymbol{y}}) + \lambda \cdot ||\hat{\boldsymbol{y}} - \boldsymbol{y}||^2, \tag{14}$$

where $\boldsymbol{y}$ is the latent representation produced by VAE encoder $\mathcal{E}(\boldsymbol{x})$, $\hat{\boldsymbol{z}}$ is the hyperprior latents, and $\hat{\boldsymbol{y}}$ is quantized by $E_\phi(\boldsymbol{y}, q)$. For multi-rate training, we randomly sample λ and obtain the corresponding q. The network architecture of the entropy model is established in accordance with [15], the most recent SOTA work that attains a great trade-off between inference latency and rate-distortion performance. Further elaborations about such compression paradigms can be found in the literatures [4, 5, 15].

Training the reverse neural network. Inspired by the recent work [21], we borrow the denoising U-net architecture from it. Following Sec. 3.1, in every training iteration a quantization scale q_t is sampled from the uniform distribution $q_t \sim \mathcal{U}(q_{min}, q_{max})$. Given a latent representation $\boldsymbol{y}_0$ extracted from a sampled image $\boldsymbol{x}$, the compressed latent representation $\boldsymbol{y}_t$ is obtained by the entropy model $\boldsymbol{y}_t = E_\phi(\boldsymbol{y}_0, q_t)$. The reverse neural network D_θ is trained using a single L_2 distance:

$$\mathcal{L}_{\text{diff}} = ||\boldsymbol{y}_0 - D_\theta(\boldsymbol{y}_t, q_t)||^2. \tag{15}$$

In order to maintain the purity of the theoretical framework, this work incorporates no generative adversarial networks (GANs) or perceptual loss finetuning stage. Nonetheless, the capacity to generate photo-realistic images of the proposed approach is evidenced by its exceptional performance on a range of perceptual metrics (see Experiment Sec. 4.3).

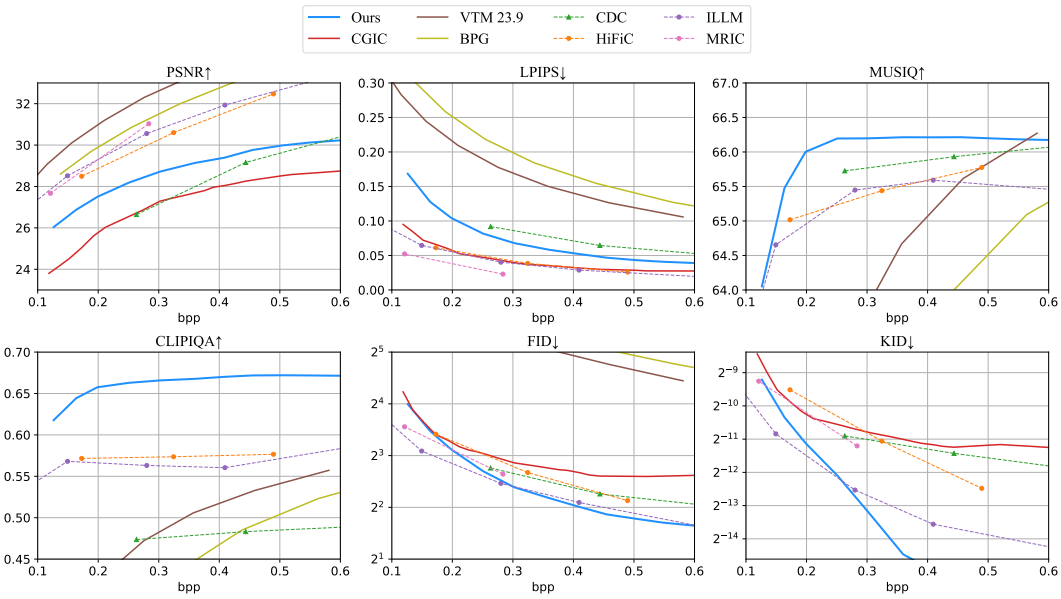


Figure 2: Comparisons of methods across various distortion and statistical fidelity metrics for the DIV2K dataset. The continuous lines represent rate-variable methods (one model for all bit rates). Circular markers denote GAN-based methods and triangular markers denote diffusion-based methods (every marker corresponds to a separate model respectively).

Putting it together. The whole process of the proposed Algorithm 1 can be elaborated as: Given a source image vector $\boldsymbol{x}$, the autoencoder contains a parametric analysis transform $\mathcal{E}$ to obtain the latent representation $\boldsymbol{y}_0$ from $\boldsymbol{x}$ and a parametric synthesis transform $\mathcal{D}$ for reconstruction. $\boldsymbol{y}$ is then quantized and compressed by the entropy model E_ϕ with a quantization scale of q_0 to latents $\hat{\boldsymbol{y}}_q$ for storage or transmission. When decoding to the reconstructed image $\hat{\boldsymbol{x}}$, a reverse network D_θ is utilized to reverse directly from the compressed data $\hat{\boldsymbol{y}}_q$ for minimal steps. The reversed latent variable $\bar{\boldsymbol{y}}_N$ is then fed to the parametric synthesis transform $\mathcal{D}$ for reconstruction $\hat{\boldsymbol{x}}$.

4 Experiments

4.1 Experimental Settings

Datasets. For the first training stage, we follow the previous compression work [15] and train the entropy model on the Open Images [24] dataset. The randomly selected Open Images dataset contains 300k images with short edge no less than 256 pixels. For the second training stage, we follow the previous diffusion work [21] and train the reverse neural network on the training set of ImageNet [12]. For evaluation, three benchmarks, i.e., DIV2K dataset [2], Kodak image set [23], and CLIC2020 test set [43], are utilized to evaluate the proposed network.

Implementation details. The detailed architecture and hyperparameter settings of the entropy model refer to the previous work [15]. For the reverse neural network, we employ the U-Net structure from EDM [21]. Our experiments and evaluations are carried out on Intel Xeon Platinum 8375C and Nvidia RTX 4090 graphics cards. By default, our proposed networks are trained using the AdamW optimizer [28]. The weight decay and momenta for AdamW are 0.02 and (0.9, 0.95). For the entropy model, we randomly crop 256×256 sub-blocks from the Open Images dataset [24] with a batchsize of 8. We train the entropy model in two stages. In the first (fixed-rate) stage, the model is trained for 0.75M steps using a constant learning rate of $1e-4$. In the second (multi-rate) stage, training continues for another 0.75M steps and then decreases the learning rate to $1e-5$ for 0.375M steps. The network is optimized with MSE metric, which represents the last term in Eq. 14. For multi-rate training, the multiplier λ are $\{0.05, 0.025, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$. As for the reverse neural network, we crop 256×256 center blocks from the ImageNet dataset [12] with a batchsize of 128. We optimize the network with the initial learning rate $1e-4$ for 0.5M steps and then decrease the

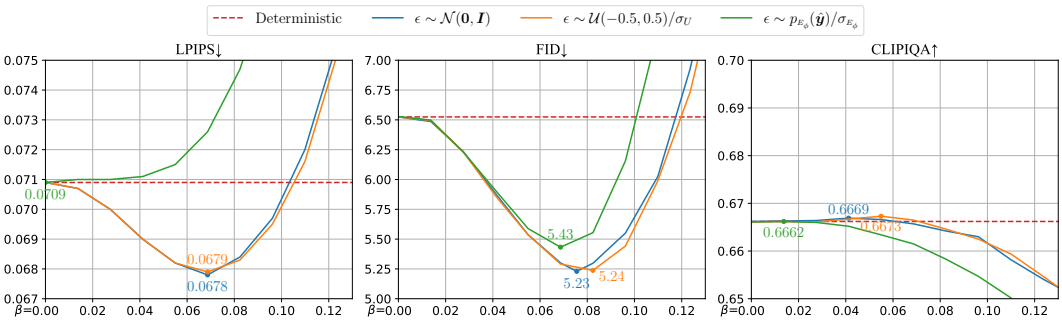


Figure 3: Evaluation of randomness injection schedules when the scale parameter q_0 is set as 0.7 (i.e., bpp $= 0.3024$), test on DIV2K with LPIPS, FID, and CLIPIQA. The dashed red lines correspond to deterministic sampling, equivalent to setting $\beta = 0$. The blue, orange, and green curves correspond to drawing a noise from a standard normal distribution $\mathcal{N}$, a uniform distribution $\mathcal{U}$, and a probability distribution p_{E_ϕ} estimated by the entropy model E_ϕ, respectively. Note that the latter two distributions are normalized by dividing the statistical deviations σ. The dots indicate the best observed results.

learning rate to $5e-5$ for other 0.5M steps. Since the compression task provides a strong prior (the compressed data), minimal sampling steps are required for the reverse network during the decoding process. We only use 2 reverse steps for all bit rates, which improves the efficiency of our method.

Comparison methods and metrics. We compare our method with the hand-crafted coding standards VVC [41], BPG [6] and recent state-of-the-art methods [1, 26, 31, 34, 47]. Note that our method should be classified with CGIC [26], a category of **rate-variable generative image compression**. The rate-variable methods obtain compressed results at all bit rates using only one model, while the other compared methods are fixed-rate, i.e., multiple separate models are required to be retrained for various compression rates. CDC [47] is the most recent state-of-the-art **diffusion-based method**. Other methods: HiFiC [31], MRIC [1] and ILLM [34] are **GAN-based approaches**. We use bits per pixel (bpp) value to indicate the compression ratio. In addition to the basic distortion metric PSNR, a range of perceptual metrics are used as evaluation: perceptual distortion: LPIPS [48], non-reference measure: MUSIQ [22], CLIPIQA [46] and statistical fidelity: FID [17], KID [7]. For the calculation of FID and KID, we follow the previous work [31] to patchify the high-resolution images.

4.2 Model Analysis

Form of randomness ω. We regard three forms of randomness: Gaussian noise is the general form adopted in legacy diffusion modeling, uniform noise simulates the quantization operation, and the noise drawn from the entropy model estimated probability distribution of latent variables. Fig. 3 shows the impacts of various types of randomness. We find that standard Gaussian noise and uniform noise are comparable, while, the noise from the entropy model performs worse. Consequently, we adopt uniform and Gaussian forms of stochasticity as our final randomness injection schedule.

Hyperparameter β. We further study the amount of stochasticity. Following Sec. 3.2, β controls the growth rate of the amount of stochastic injections with q_t. Fig. 3 demonstrates that there exists a trade-off among perceptual distortion (LPIPS), statistical fidelity (FID), and non-reference metric (CLIPIQA). The results of Gaussian and uniform curves are analogous, and finally we select $\beta = 0.075$ as the setting of our SOTA model.

4.3 Comparison with State-of-the-art Methods

Rate-distortion comparison. We evaluate the rate-distortion performance of our proposed models by drawing the rate-distortion curves Fig. 2. As DIV2K [2] is one of the most commonly used benchmark datasets in the field of low-level vision, we mainly compare our proposed network with the aforementioned SOTA methods on DIV2K dataset. Reference models (VTM 23.9 and BPG) achieve the best PSNR scores, but display poor perceptual distortion and statistical fidelity. GAN-based methods tend to obtain a better LPIPS metric, while, compared to the diffusion-based SOTA

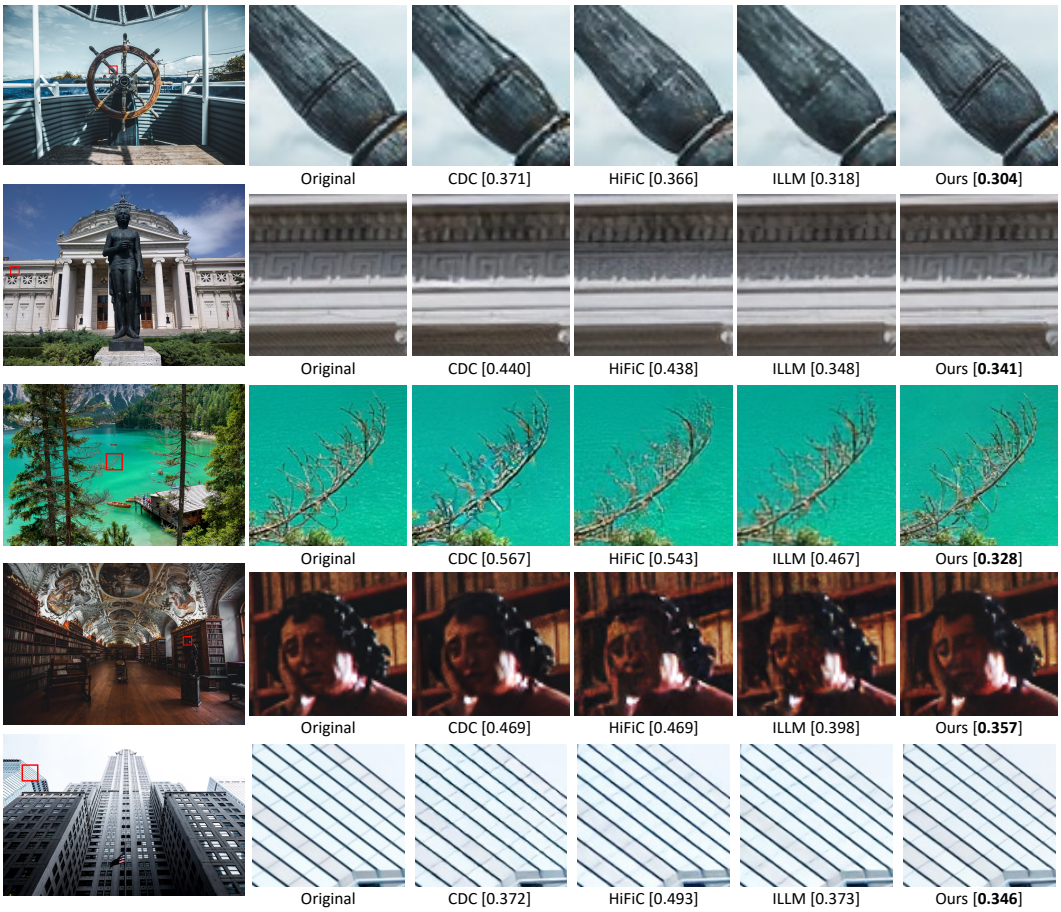


Figure 4: Visualization of the reconstructed images (top to bottom: *0824*, *0812*, *0807*, *0841*, and *0846*) from DIV2K dataset. The titles under the sub-figures are represented as "method [bpp]".

method (CDC), ours achieves superior performance. For the non-reference metrics (MUSIQ and CLIPIQA), our method is able to acquire a clear advantage over the other models. In the context of rate-variable image compression, our method has uniformly better statistical fidelity than CGIC as measured by FID and KID. Except for the lower bpp range (below 0.25), our method performs comparably to the fixed-rate SOTA model ILLM on FID metric, and demonstrates better statistical fidelity evaluated by KID. We also investigate the effectiveness of our method on the Kodak and CLIC2020 dataset, where the results are provided in our supplementary materials.

Visualization analysis. Thanks to our well-designed framework, our method achieves superior restoration of fine image details. Fig. 4 presents five comparison sets against recent state-of-the-art reconstruction models [31, 34, 47], with results generated at comparable bit rates on the DIV2K dataset. These visualizations demonstrate that our method faithfully reconstructs image details aligned with the original content, rather than introducing artificial or irrelevant textures.

5 Conclusion

In this work, we propose a novel diffusion modeling framework for generative image compression. We establish an organic integration of learned image compression and diffusion, building a complete diffusion framework from the forward process to the reverse process with the help of SDE theory. Our proposed method takes full advantage of the capacity of diffusion modeling, thus achieving state-of-the-art performance on a range of perceptual metrics. Furthermore, we believe this work will spark further innovations across a wide range of domains, especially by encouraging adaptations of the core diffusion modeling framework to adapt to diverse research needs.

References

[1] Agustsson, E., Minnen, D., Toderici, G., Mentzer, F.: Multi-realism image compression with a conditional generator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22324–22333 (2023) 4, 8

[2] Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017) 7, 8

[3] Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Gool, L.V.: Generative adversarial networks for extreme learned image compression. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 221–231 (2019) 1, 4

[4] Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. arXiv preprint arXiv:1611.01704 (2016) 1, 2, 3, 6

[5] Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436 (2018) 1, 2, 3, 6

[6] Bellard, F.: Bpg image format (2015), https://bellard.org/bpg 1, 8

[7] Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. arXiv preprint arXiv:1801.01401 (2018) 8

[8] Careil, M., Muckley, M.J., Verbeek, J., Lathuilière, S.: Towards image compression with perfect realism at ultra-low bitrates. In: The Twelfth International Conference on Learning Representations (2023) 1, 4

[9] Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7939–7948 (2020) 1

[10] Choi, Y., El-Khamy, M., Lee, J.: Variable rate deep image compression with a conditional autoencoder. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3146–3154 (2019) 3

[11] Cui, Z., Wang, J., Gao, S., Guo, T., Feng, Y., Bai, B.: Asymmetric gained deep image compression with continuous rate adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10532–10541 (2021) 3

[12] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 7

[13] Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021) 1

[14] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) 1

[15] Han, M., Jiang, S., Li, S., Deng, X., Xu, M., Zhu, C., Gu, S.: Causal context adjustment loss for learned image compression. arXiv preprint arXiv:2410.04847 (2024) 1, 6, 7

[16] He, D., Yang, Z., Peng, W., Ma, R., Qin, H., Wang, Y.: Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5718–5727 (2022) 1

[17] 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) 8

[18] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 1

[19] Hoogeboom, E., Agustsson, E., Mentzer, F., Versari, L., Toderici, G., Theis, L.: High-fidelity image compression with score-based generative models. arXiv preprint arXiv:2305.18231 (2023) 1, 4

[20] Jia, Z., Li, J., Li, B., Li, H., Lu, Y.: Generative latent coding for ultra-low bitrate image compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26088–26098 (2024) 1, 4

[21] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems **35**, 26565–26577 (2022) 4, 5, 6, 7

[22] Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5148–5157 (2021) 8

[23] Kodak, E.: Kodak lossless true color image suite (photocd pcd0992) (1993), http://r0k.us/graphics/kodak 7

[24] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International journal of computer vision **128**(7), 1956–1981 (2020) 7

[25] Lei, E., Uslu, Y.B., Hassani, H., Bidokhti, S.S.: Text+ sketch: Image compression at ultra low rates. arXiv preprint arXiv:2307.01944 (2023) 1, 4

[26] Li, A., Li, F., Liu, Y., Cong, R., Zhao, Y., Bai, H.: Once-for-all: Controllable generative image compression with dynamic granularity adaption. arXiv preprint arXiv:2406.00758 (2024) 1, 4, 8, 13

[27] Liu, J., Sun, H., Katto, J.: Learned image compression with mixed transformer-cnn architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14388–14397 (2023) 1

[28] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 7

[29] Lu, J., Zhang, L., Zhou, X., Li, M., Li, W., Gu, S.: Learned image compression with dictionary-based entropy model. arXiv preprint arXiv:2504.00496 (2025) 1

[30] Mao, Q., Yang, T., Zhang, Y., Wang, Z., Wang, M., Wang, S., Jin, L., Ma, S.: Extreme image compression using fine-tuned vqgans. In: 2024 Data Compression Conference (DCC). pp. 203–212. IEEE (2024) 1

[31] Mentzer, F., Toderici, G.D., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. Advances in neural information processing systems **33**, 11913–11924 (2020) 1, 4, 8, 9, 13

[32] Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. Advances in neural information processing systems **31** (2018) 1

[33] Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3339–3343. IEEE (2020) 1

[34] Muckley, M.J., El-Nouby, A., Ullrich, K., Jégou, H., Verbeek, J.: Improving statistical fidelity for neural image compression with implicit local likelihood models. In: International Conference on Machine Learning. pp. 25426–25443. PMLR (2023) 1, 4, 8, 9, 13

[35] Qian, Y., Lin, M., Sun, X., Tan, Z., Jin, R.: Entroformer: A transformer-based entropy model for learned image compression. arXiv preprint arXiv:2202.05492 (2022) 1

[36] 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) 6

[37] Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948) 3

[38] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 5, 6

[39] Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019) 1, 2, 3, 4, 5

[40] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020) 2, 3, 4, 5, 6

[41] Team, J.V.E.: Versatile video coding reference software version 23.9(vtm-23.9) (2025), https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tags/VTM-23.9 1, 8, 13

[42] Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085 (2015) 3

[43] Toderici, G., Shi, W., Timofte, R., Theis, L., Ballé, J., Agustsson, E., Johnston, N., Mentzer, F.: Workshop and challenge on learned image compression (clic2020). In: CVPR (2020) 7

[44] Tschannen, M., Agustsson, E., Lucic, M.: Deep generative models for distribution-preserving lossy compression. Advances in neural information processing systems **31** (2018) 1

[45] Wang, G.H., Li, J., Li, B., Lu, Y.: Evc: Towards real-time neural image compression with mask decay. arXiv preprint arXiv:2302.05071 (2023) 3

[46] Wang, J., Chan, K.C., Loy, C.C.: Exploring clip for assessing the look and feel of images. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 2555–2563 (2023) 8

[47] Yang, R., Mandt, S.: Lossy image compression with conditional diffusion models. Advances in Neural Information Processing Systems **36**, 64971–64995 (2023) 1, 4, 8, 9, 13

[48] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 8

A Derivation of Discrete ODE Solver

A discrete ODE solver is to use numerical methods to compute integration of ordinary differential equations. In our framework, Given $\boldsymbol{x}_i$ at a compression ratio of q_i, we aim to obtain $\boldsymbol{x}_{i+1}$ at q_{i+1}. Using the first-order Euler's solver is to exploit a differential approximation:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \Delta t \cdot \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}. \tag{16}$$

Substitute Eq. 10 to Eq. 16:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t - \Delta t \cdot \frac{\mathrm{d}q(t)}{\mathrm{d}t} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t)). \tag{17}$$

We obtain $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; q(t))$ through the neural network D_θ:

$$\begin{gathered} \boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t - \Delta t \cdot \frac{\mathrm{d}q(t)}{\mathrm{d}t} \cdot \frac{\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_t}{q(t)}, \\ \text{with } \hat{\boldsymbol{x}}_0 = D_\theta(\boldsymbol{x}_t, q(t)). \end{gathered} \tag{18}$$

To obtain the same form as the main text, we define $t_{i+1} = t_i + \Delta t$:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - (t_{i+1} - t_i) \cdot \frac{\mathrm{d}q(t)}{\mathrm{d}t} \cdot \frac{\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_i}{q(t_i)}. \tag{19}$$

For simplicity and continuous sampling during training, we set $q(t) := t$:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{q(t_i) - q(t_{i+1})}{q(t_i)} (\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_i). \tag{20}$$

B Compression Latency

Table 1: Comparison of coding latency evaluated on Kodak dataset. All the models are evaluated on the same platform. The second line of **Model** describes the categories of the compared methods.

Model	Ours	CDC [47]	ILLM [34]	CGIC [26]
	Diffusion-based	Diffusion-based	GAN-based	VQ-based
Encoding Time (ms)	123	23	60	85
Decoding Time (ms)	280	824	71	32
Total Time (ms)	403	847	131	117

We compare the coding efficiency of our methods with recent state-of-the-art methods [26, 34, 47]. These methods are classified into diffusion-based, GAN-based, and VQ-based approaches. As Table 1 shows, thanks to the minimal sampling steps required for our method, we achieve coding efficiency superior to that of the most recent diffusion-based SOTA work CDC. However, diffusion models generally exhibit slower coding speeds than GAN-based and VQ-based methods. The reason for this is that GAN-based methods only require a one-through transformation, whereas VQ-based methods abandon the entropy model to estimate the probability distribution, which is replaced by transmitting the index of codebook. How to further promote the inference latency of diffusion-based learned image compression methods is worth exploring in the future.

C Image Reconstruction Visualization

We compare the reconstruction results on *0854* (Fig. 5), *1c55* (Fig. 6), and *0884* (Fig. 7) of our model with those of CDC [47], HiFiC [31], ILLM [34] and hand-crafted method VVC [41].



Figure 5: Visualization of the reconstructed images (*0854*) from DIV2K dataset. The titles under the sub-figures are represented as "method [bpp]".

zoom in for better visualization



Figure 6: Visualization of the reconstructed images (*1c55*) from CLIC2020 dataset. The titles under the sub-figures are represented as "method [bpp]".

zoom in for better visualization



Figure 7: Visualization of the reconstructed images (*0884*) from DIV2K dataset. The titles under the sub-figures are represented as "method [bpp]".

zoom in for better visualization

D Further Experimental Results

We also compare our methods on the CLIC2020 and Kodak datasets. For the Kodak dataset, because too few images are contained, the statistical fidelity metrics (FID and KID) are invalid to evaluate the reconstructed results. The RD curves are revealed in Fig. 8 and Fig. 9.

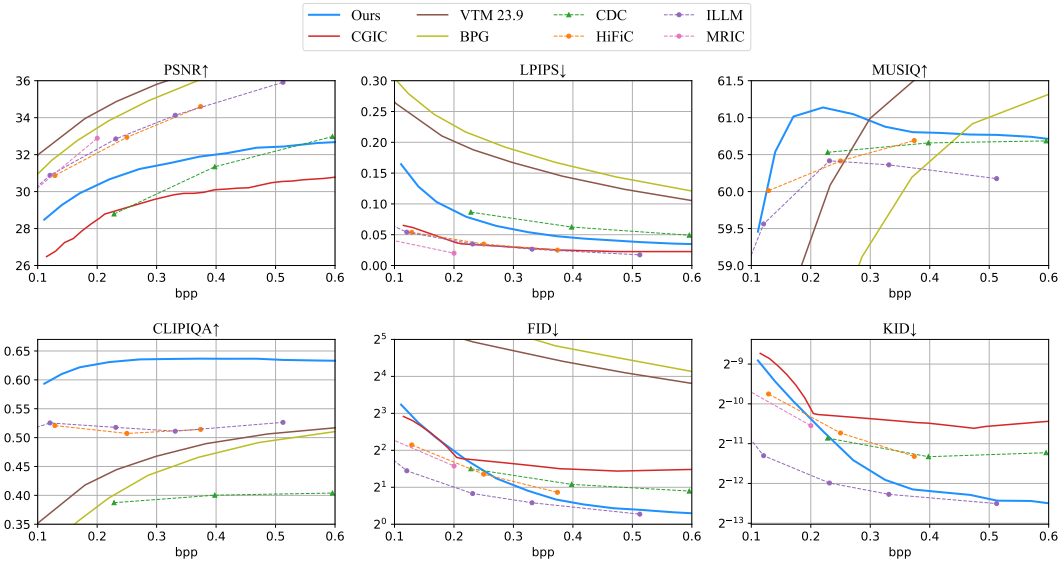


Figure 8: Comparisons of methods across various metrics on the CLIC2020 dataset.

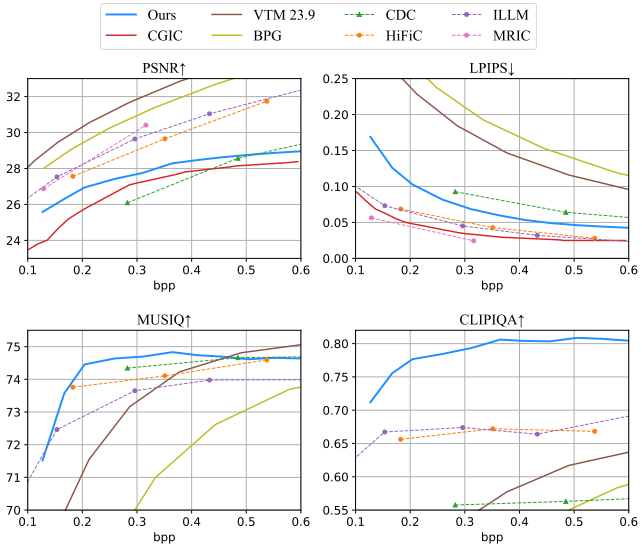


Figure 9: Comparisons of methods across various metrics on the Kodak dataset.