

Lossy Image Compression with Foundation Diffusion Models

Lucas Relic^{1,2}, Roberto Azevedo², Markus Gross^{1,2}, and Christopher Schroers²

¹ ETH Zürich, Switzerland

{lucas.relic, grossm}@inf.ethz.ch

² Disney Research | Studios, Zürich, Switzerland

{roberto.azevedo, christopher.schroers}@disneyresearch.com

Abstract. Incorporating diffusion models in the image compression domain has the potential to produce realistic and detailed reconstructions, especially at extremely low bitrates. Previous methods focus on using diffusion models as expressive decoders robust to quantization errors in the conditioning signals. However, achieving competitive results in this manner requires costly training of the diffusion model and long inference times due to the iterative generative process. In this work we formulate the removal of quantization error as a denoising task, using diffusion to recover lost information in the transmitted image latent. Our approach allows us to perform less than 10% of the full diffusion generative process and requires no architectural changes to the diffusion model, enabling the use of foundation models as a strong prior without additional fine tuning of the backbone. Our proposed codec outperforms previous methods in quantitative realism metrics, and we verify that our reconstructions are qualitatively preferred by end users, even when other methods use twice the bitrate.

Keywords: Image compression · Latent diffusion · Generative models

1 Introduction

In today’s digital era, multimedia content dominates global internet traffic, making the development of efficient compression algorithms increasingly important. Traditional codecs, which use handcrafted transformations [1, 42], are now outperformed by data-driven neural image compression (NIC) [7, 8, 33] methods that optimize for both rate and distortion. Nevertheless, most current methods still produce blurry and unrealistic images in extremely low bitrate settings [5, 13, 24]. This is the result of such methods being optimized for rate-distortion, where distortion is measured with pixel-wise metrics like mean-squared error (MSE) [10]. The rate-distortion-realism³ triple tradeoff [4, 10, 32] formalizes this phenomenon and states that optimizing for low distortion (*i.e.*, pixel-wise error) *necessarily*

³ We use “realism” and “perception” interchangeably, representing the similarity of the reconstructed image to other natural images.



Fig. 1: Visual examples of our proposed method and various classes of image compression codecs. Traditional (BPG [1]) and autoencoder-based (ELIC [20]) codecs suffer from blocking or blurring, and reconstructions from GAN-based (ILLM [34]) and previous diffusion-based (HFD [24]) methods contain high-frequency artifacts. Our proposal is as realistic as the original image while recovering a high level of detail. Bitrates are shown relative to our method. Best viewed digitally.

results in unrealistic images (*i.e.*, images that do not fall on the manifold of natural images). In low-bitrate scenarios, however, it can be preferable to decode realistic (thus, more perceptually pleasant) images, even if that means lower performance in pixel-wise metrics [6].

Generative compression methods [5, 32, 34] try to reconstruct realistic images by introducing GAN architectures and adversarial or perceptual losses. In image generation, however, diffusion models [15] have now emerged as a powerful alternative, outperforming GANs [15] and achieving state-of-the-art realism scores [37]. Diffusion models are thus a natural fit for generative image compression architectures targeting low-bitrate scenarios. Yet, their applicability and adoption is hindered by large model size and prohibitively expensive training times, requiring multiple GPU *years* and hundreds of thousands of dollars [14]. The introduction of open-source foundation models [11] has the potential to democratize these powerful models and provide strong priors that can be explored for feature extraction or transfer learning on a variety of domains [19], for example image generation [45], depth estimation [27], and even music generation [16].

The use of foundation diffusion models as prior for image compression, however, is still an underexplored research area. Some works address this task [12, 30] but operate at extremely low bitrates (less than 0.03 bpp) where reconstructed image content significantly differs from the original, limiting applicability. Only Careil *et al.* [13] apply foundation diffusion models in a practical compression setting. However, they modify the base model architecture and thus require fine-tuning on a large dataset containing millions of images. Other works, which train

the diffusion component from scratch, operate at relatively high bitrates [44] where low pixel-wise distortion can be achieved, or perform image enhancement [18, 24] rather than native end-to-end compression. Notably, all current work on diffusion-based image compression tasks sample the output image from pure noise, requiring the full diffusion sampling process, which can take up to one minute per image [44] due to its iterative nature.

To advance the state of the art, we propose a novel image compression codec that uses foundation latent diffusion models as a means to synthesize lost details, particularly at low bitrate. Leveraging the similarities between quantization error and noise [7], we transmit a quantized image latent and perform a subset of denoising steps at the receiver corresponding to the noise level (*i.e.*, quantization error) of the latent (similar to diffusion image editing techniques [31]). The key components of our proposal are: i) the autoencoder from a foundation latent diffusion model to transform an input image to a lower-dimensional latent space; ii) a learned *adaptive quantization* and *entropy encoder*, enabling inference-time control over bitrate within a single model; iii) a learned method to *predict the ideal denoising timestep*, which allows for balancing between transmission cost and reconstruction quality; and iv) a diffusion decoding process to synthesize information lost during quantization. Unlike previous work, our formulation requires only a fraction of iterative diffusion steps and can be trained on a dataset of fewer than 100k images. We also directly optimize a distortion objective between input and reconstructed images, enforcing coherency to the input image while maintaining highly realistic reconstructions (Fig. 1) due to the diffusion backbone.

In sum, our contributions are:

- We propose a novel latent diffusion-based lossy image compression pipeline that is able to produce highly realistic and detailed image reconstructions at low bitrates.
- To achieve this, we introduce a novel parameter estimation module that simultaneously learns *adaptive quantization parameters* as well as the *ideal number of denoising diffusion steps*, allowing a faithful and realistic reconstruction for a range of target bitrates with a single model.
- We extensively evaluate state-of-the-art generative compression methods on several datasets via both objective metrics and a user study. To the best of our knowledge, this is the first user study that compares generative diffusion models for image compression. Our experiments verify that our method achieves state-of-the-art visual quality as measured in FID and end users subjectively prefer our reconstructions.

2 Related Work

Although diffusion models have seen significant successes in the machine learning community, their use in the image compression domain is still limited.

Yang and Mandt [44] proposed the first transform-coding-based lossy compression codec using diffusion models. They condition a diffusion model on con-

textual latent variables produced with a VAE-style encoder. Despite showing competitive testing results, their method operates in a relatively high bitrate range (0.2bpp and above). It thus leaves room for improvement, particularly at lower bitrates, where the powerful generation capabilities of diffusion models can be used to reconstruct images from lower entropy signals.

Diffusion models have also been proposed to augment existing compression architectures by adding details to images compressed with autoencoder-based neural codecs [18, 24]. These methods are sub-optimal since they address an image enhancement task decoupled from compression; in other words, they post-process a compressed image rather than train an image compression method end-to-end. Images reconstructed in this manner often contain high-frequency artifacts or entirely lose image content as the diffusion model cannot rectify artifacts introduced in the initial compression stage.

Several works develop codecs for extremely low bitrate compression (less than 0.03 bpp) via latent diffusion. These methods condition pretrained text-to-image diffusion models with text and spatial conditioning such as CLIP embeddings [6, 30] and edge [30] or color [6] maps, respectively. Most notably, Careil *et al.* [13] augment a latent diffusion model with an additional encoder and image captioner to produce vector-quantized “hyper-latents” and text signals which are used to generate an image at the receiver. However, their architectural changes require substantial fine-tuning of the underlying diffusion model, hindering the advantages of using a foundation model. Additionally, in general, the extremely low bitrate of these methods results in reconstructions that, while realistic, vary significantly in content from the original images.

Notably, all existing works focus on regenerating the image at the receiver side from low-entropy conditioning signals, requiring tens or hundreds of costly diffusion sampling steps per image. Our novel formulation of processing a quantized latent representation allows us to predict and perform the ideal number of denoising diffusion steps, typically between 2 and 7% of the full process, depending on the bitrate. Combined with a learned per-content adaptive quantization, we propose the first diffusion-based image compression codec with inference-time bitrate control and demonstrate that our method produces more realistic reconstructions and more faithfully represents the input image compared to previous works on generative image compression (see Sec. 5).

3 Background

Neural Image Compression. Lossy neural image codecs (NIC) are commonly modeled as autoencoders, in which an encoder \mathcal{E} transforms an image \mathbf{x} to a quantized latent $\hat{\mathbf{y}} = \lfloor \mathcal{E}(\mathbf{x}) \rfloor$, while a decoder \mathcal{D} reconstructs an approximation of the original image $\hat{\mathbf{x}} = \mathcal{D}(\hat{\mathbf{y}})$. Based on Shannon’s rate-distortion theory [38], during training, \mathcal{E} and \mathcal{D} are optimized to minimize the rate-distortion trade-off:

$$\mathcal{L}_{total} = \mathcal{L}_{bits}(\hat{\mathbf{y}}) + \lambda \mathcal{L}_{rec}(\mathbf{x}, \hat{\mathbf{x}}) \quad (1)$$

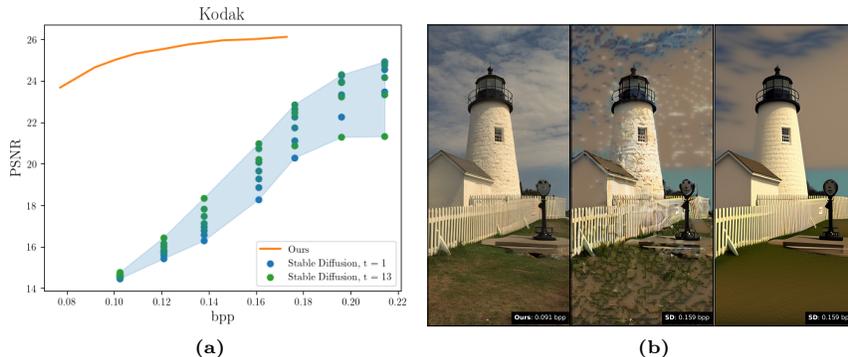


Fig. 2: Rate-distortion (Fig. 2a) and visual (Fig. 2b) comparisons of our method to naively quantizing and entropy coding the latents of a latent diffusion model (Stable Diffusion [37]). The LDM baseline requires nearly triple the bits to achieve comparable performance to our method and severely degrades the image at lower bitrates. Performing additional diffusion steps still does not produce a realistic image (Fig. 2b, right). The color gradient of the dots in Fig. 2a represents the number of denoising steps.

where \mathcal{L}_{rec} is a measure of distortion (commonly MSE), $\mathcal{L}_{bits}(\hat{\mathbf{y}})$ is an estimate of the bitrate needed to store $\hat{\mathbf{y}}$, and λ controls the trade-off between rate and distortion. According to Shannon’s theory, $\mathcal{L}_{bits}(\hat{\mathbf{y}}) = -\log_2 P(\hat{\mathbf{y}})$, where P is a probability model of $\hat{\mathbf{y}}$.

As \mathcal{L}_{rec} is often formulated as a pixel-wise difference between images [10], especially at low bitrate, it can lead to unnatural artifacts in the reconstructed images (*e.g.*, blurring). In such scenarios, it is interesting to design \mathcal{D} such that the distribution of reconstructed images closely follows the distribution of natural images (*i.e.*, that it produces realistic images), even though this results in lower pixel-wise distortion [10]. Generative image compression methods, such as our proposal, focus on optimizing this rate-distortion-realism trade-off [4].

Diffusion. Diffusion models (DMs) [23, 39] are a class of generative models that define an iterative process $q(x_t|x_{t-1})$ that gradually destroys an input signal as t increases, and try to model the reverse process $q(x_{t-1}|x_t)$. Empirically, the forward process is performed by adding Gaussian noise to the signal; thus, the reverse process becomes a denoising task. The diffusion model \mathcal{M}_θ approximates the reverse process by estimating the noise level ϵ_θ of the image and using it to predict the previous step of the process:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\tilde{\mathbf{x}}_0 + \sqrt{1 - \alpha_{t-1}}\epsilon_\theta, \quad \text{with} \quad \tilde{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon_\theta}{\sqrt{\alpha_t}} \quad (2)$$

Here, t is the current timestep of the diffusion process, \mathbf{x}_t and α_t represent the sample and variance of the noise at timestep t , respectively, and $\tilde{\mathbf{x}}_0$ is the

predicted fully denoised sample from any given t . Eqs. 2 can be simplified to

$$\mathbf{x}_{t-1} = \mathcal{M}_\theta(\mathbf{x}_t, t) \quad (3)$$

where $\mathcal{M}_\theta(\cdot)$ is one forward pass of the diffusion model. It is therefore possible to sample from a DM by initializing $\mathbf{x}_T = \mathcal{N}(0, 1)$ and performing T forward passes to produce a fully denoised image.

Latent diffusion models (LDMs) [37] improve memory and computational efficiency of DMs by moving the diffusion process to a spatially lower dimensional latent space, encoded by a pre-trained variational autoencoder (VAE) [28]. Such a latent space provides similar performance of the corresponding pixel-space DMs while requiring less parameters (and memory) [9]. These types of DMs are trained in a VAE latent space where $\mathbf{y} = \mathcal{E}_{vae}(\mathbf{x})$ and a sampled latent \mathbf{y}_0 can be decoded back to an image $\hat{\mathbf{x}} = \mathcal{D}_{vae}(\mathbf{y}_0)$.

Since LDMs are based on VAEs, they can also be considered a type of compression method. However, their applicability in lossy image compression is hindered by inherent challenges. LDMs lack explicit training to produce discrete representations, resulting in highly distorted reconstructions when used for lossy compression [24], and cannot navigate the rate-distortion tradeoff. To highlight such issues, Fig. 2 shows the performance of the same LDM used by our method (without modifications) as a compression codec compared to our approach optimized for lossy image compression. In this experiment, we manually sweep over a range of quantization and diffusion timestep parameters, encoding the images under the different configurations. Specifically, we encode to the latent space, quantize according to the chosen parameters, compress with zlib [2], run the chosen number of denoising diffusion steps, and decode back to image space. As shown, the unmodified LDM requires nearly 3x the bits to achieve comparable performance to our method and cannot produce realistic images at low bitrates, regardless of the number of diffusion steps performed (Fig. 2b). Thus, deploying LDMs for compression requires thoughtful consideration to maximize their effectiveness.

4 Method

Fig. 3 shows the high-level architecture of our method. It is composed of a variational autoencoder (containing an encoder, \mathcal{E}_{vae} , and a decoder, \mathcal{D}_{vae}) a *quantization* and *diffusion timestep* parameter estimation network (\mathcal{P}_ϕ), an entropy model, and a latent diffusion model (\mathcal{M}_θ).

Our encoding process is performed as follows: First, the image \mathbf{x} is encoded into its latent representation $\mathbf{y} = \mathcal{E}_{vae}(\mathbf{x})$. Then, \mathbf{y} is quantized by an adaptive quantization method parameterized by γ (*i.e.*, $\hat{\mathbf{z}} = \mathcal{Q}(\mathbf{y}, \gamma)$). Finally, $\hat{\mathbf{z}}$ is entropy encoded and stored or transmitted.

During decoding, the inverse quantization transformation computes $\hat{\mathbf{y}}_t = \mathcal{Q}^{-1}(\hat{\mathbf{z}}, \gamma)$, which is then used as input to the generative LDM process over t denoising steps to recover an approximation $\hat{\mathbf{y}}_0$ of the original latent representation \mathbf{y} . Finally, $\hat{\mathbf{y}}_0$ is decoded by the VAE decoder into a reconstructed image $\hat{\mathbf{x}} = \mathcal{D}_{vae}(\hat{\mathbf{y}}_0)$. Algorithm 1 shows the complete encoding/decoding process.

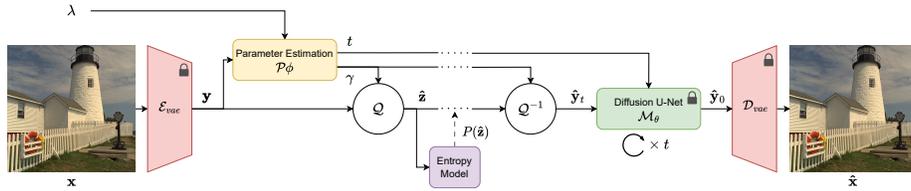


Fig. 3: Overview of our approach. The input image \mathbf{x} is encoded into latent space and transformed according to predicted parameters γ before quantization and entropy coding. The quantized representation $\hat{\mathbf{z}}$ is transmitted with γ and predicted diffusion timestep t as side information. At the receiver the latent is inverse transformed, diffused over t steps, and decoded back to image space.

A key feature of our method is that both the quantization parameters γ and the number of denoising steps t can be adapted in a per-content and per-target-bitrate manner (controlled by the rate-distortion trade-off parameter λ). To achieve this, we train a neural network $\mathcal{P}_\phi(\mathbf{y}, \lambda)$ that predicts both t and γ .

Intuitively, our method learns to discard information (through the quantization transformation) that can be synthesized during the diffusion process. Because errors introduced during quantization are similar to adding noise [7, 8, 33] and diffusion models are functionally denoising models, they can be used to remove the quantization noise introduced during coding.

4.1 Latent Diffusion Model Backbone

Aiming at avoiding extensive training time, we use Stable Diffusion v2.1 [37] for certain modules of our architecture, particularly, \mathcal{E}_{vae} , \mathcal{D}_{vae} , and \mathcal{M}_θ . Note that our method works independently of the base model. We select Stable Diffusion as it is one of the only foundation latent diffusion models with publicly available code and model weights.

4.2 Parameter Estimation

As aforementioned, the quantization parameters, γ , and the optimal number of denoising steps the diffusion network should perform, t , are predicted by a neural network $\mathcal{P}_\phi(\mathbf{y}, \lambda)$, which takes as input the latent \mathbf{y} and the rate-distortion trade-off λ .

Adaptive Quantization. Our adaptive quantization function \mathcal{Q} is defined as an affine transformation \mathcal{T} for each channel of the latent \mathbf{y} , parameterized by γ , before applying standard integer quantization, *i.e.*,

$$\hat{\mathbf{z}} = Q(\mathbf{y}, \gamma) = \lfloor \mathcal{T}(\mathbf{y}, \gamma) \rfloor \quad (4)$$



Fig. 4: Intermediate states of the sequential denoising process in our decoder. Our method predicts the optimal number of denoising steps, highlighted in red, to produce the most perceptually pleasing output. Best viewed digitally.

γ is transmitted as side information in order for the decoder to perform the inverse transform at the client side, *i.e.*,

$$\hat{\mathbf{y}}_t = Q^{-1}(\hat{\mathbf{z}}, \gamma) = \mathcal{T}^{-1}(\hat{\mathbf{z}}, \gamma) \quad (5)$$

Timestep Prediction. Contrary to an image generation task, which begins diffusion from “pure noise”, in our compression task, we start the diffusion process from a quantized latent, which already contains structural and semantic information of the content. In such a scenario, performing the entire range of denoising steps during decoding is both wasteful and results in over-smoothed images. Therefore, we learn to predict the subset of denoising diffusion steps that produces optimal decoded images. Fig. 4 illustrates how the decoded image quality changes based on the number of diffusion steps performed by the decoder, where too few or too many steps result in noisy or over-smoothed images, respectively. Because the number of optimal denoising steps depends on the amount of noise in the latent (and therefore the severity of quantization), and vice versa, we predict t and γ jointly in the parameter estimation module.

Architecture. The parameter estimation neural network $\mathcal{P}_\phi(\mathbf{y}, \lambda)$ employs a fully convolutional architecture. We stack alternating downsampling (*i.e.*, stride 2) and standard convolutional layers, increasing filter count as depth increases. In the last layer, we reduce the output filter count to correspond with the total number of parameters estimated⁴ and apply mean average pooling on each output channel to produce a single scalar for each parameter. We use SiLU [21] activation between each convolutional layer. No activation is applied after the final convolutional layer. However, we apply sigmoid activation on the scalar

⁴ We use 2 quantization parameters per latent channel plus one additional parameter for timestep prediction. As the image latent has 4 channels we predict a total of 9 parameters per image.

Algorithm 1 Encoding and Decoding process

```

Given: image  $\mathbf{x}$ 
 $\mathbf{y} \leftarrow \mathcal{E}_{vae}(\mathbf{x})$ 
 $\gamma, t \leftarrow \mathcal{P}_\phi(\mathbf{y})$ 
 $\hat{\mathbf{z}} \leftarrow \mathcal{Q}(\mathbf{y}, \gamma)$ 
bitstream  $\longleftrightarrow \hat{\mathbf{z}}$  ▷ Entropy code using  $P(\hat{\mathbf{z}})$ 
 $\hat{\mathbf{y}}_t \leftarrow \mathcal{Q}^{-1}(\hat{\mathbf{z}}, \gamma)$ 
for  $n = t$  to 1 do
     $\hat{\mathbf{y}}_{n-1} \leftarrow \mathcal{M}_\theta(\hat{\mathbf{y}}_n, n)$  ▷ Eq. 3
end for
 $\hat{\mathbf{x}} \leftarrow \mathcal{D}_{vae}(\hat{\mathbf{y}}_0)$ 

```

predicted timestep to guarantee a range of $[0, 1]$. To condition \mathcal{P}_ϕ on the target bitrate, we expand λ to the same spatial dimension as the latent sample and concatenate them along the channel dimension before processing with the parameter estimation network.

4.3 Entropy Coding

We use a joint contextual and hierarchical entropy model to encode the quantized latent to a bitstream. Such models have been extensively researched in the literature [8, 33, 36]. We select Entroformer [36] as our entropy model as it yields the best performance in our experiments. It consists of a transformer-based hyperprior and bidirectional context model to estimate the latent’s distribution $P(\hat{\mathbf{z}})$, which is used to encode it to a bitstream via arithmetic encoding [35].

4.4 Optimization

Following Eq. 1, we jointly optimize the tradeoff between the estimated coding length of the bitstream and the quality of the reconstruction:

$$\mathcal{L} = -\log_2 P(\hat{\mathbf{z}}) + \lambda \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2. \quad (6)$$

We train our model on the **Vimeo-90k** [43] dataset and randomly crop the images to 256×256 px in each epoch. Our model is optimized for 300,000 steps with learning rate $1e-4$. We randomly sample $\lambda \in [1, 5, 10, 20]$ at each gradient update to train for multiple target bitrates within a single model.

While our main motivation was to utilize foundation models without significant modification, we do make minor adjustments in our pipeline to allow for optimization of trainable modules upstream. During training, it is prohibitively expensive to backpropagate the gradient through multiple passes of the diffusion model as it runs during DDIM [40] sampling. Therefore, we perform only one DDIM sampling iteration and directly use $\tilde{\mathbf{x}}_0$ as the fully denoised data (see Eq. (2) and Appendix B). For the low timestep range our model operates in, we observe that the difference between $\tilde{\mathbf{x}}_0$ and the true fully denoised data \mathbf{x}_0 is minimal and sufficient for the optimization of the parameter estimation module. At inference time, we perform the standard iterative DDIM process. Additionally, during the diffusion sampling process, the timestep t is

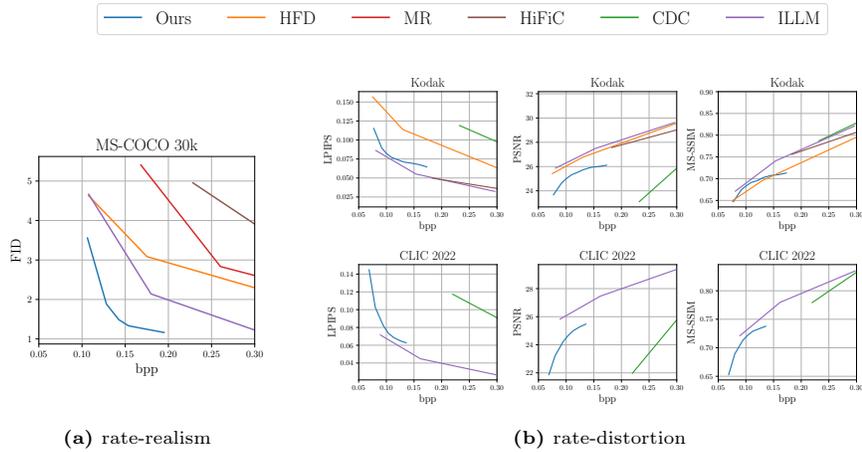


Fig. 5: Quantitative comparison of our method with other baselines. We outperform all methods in (a) rate-realism while remaining competitive with the best performing generative codecs in (b) pixel-wise distortion metrics.

used to index an array of required precomputed values (*e.g.*, variance schedule α_t). This discretization prevents optimization of the parameter estimation network. Therefore, we implement continuous functions for each required value and evaluate them with the predicted timestep during training.

5 Experiments

We compare our method to state-of-the-art generative and diffusion-based image compression codecs via objective metrics and a subjective user study. Fig. 7 also shows qualitative comparisons of ours to different methods.

Datasets. We conduct experiments on the following datasets: i) **Kodak** [29], which consists of 24 images 768×512 px (or inverse). ii) the **CLIC2022** [3] test set, which contains 30 high-resolution images resized such that the longer side is 2048px. We center crop each image to 768×768 px due to the high memory consumption required to process large images with Stable Diffusion; and iii) **MS-COCO 30k**, which has recently been used for evaluating the realism of the reconstruction of compression methods [4, 24]. The dataset is preprocessed as stated in [4], resulting in 30,000 images of size 256×256 px each.

Metrics. We evaluate our proposal and baselines using: **PSNR** and **MS-SSIM**, as measures of pixel-wise distortion; **LPIPS**, as a more perceptually-oriented distortion metric; and **FID** [22], to evaluate the realism of the reconstructed images. FID measures the similarity between the distributions of source and distorted images, and thus has been widely used as a measure of realism [4, 32],

particularly for image generation tasks. Since FID requires a higher number of samples of both source and distorted images, we focus the FID comparison only on MS-COCO 30k.

Baselines. We compare our method against GAN- and diffusion-based image codecs. **HiFiC** [32] is a well-known generative neural image compression codec and remains a strong GAN-based baseline, while the recently proposed **ILLM** [34] improves upon the HiFiC architecture and is available at more comparable bitrates to ours. For diffusion-based approaches, we use **CDC** [44] and **HFD** [24], the only two other practical works in this field. Both HFD and Careil *et al.* [13] are trained on large proprietary datasets and therefore cannot be reproduced for comparison. For this reason, we compare to HFD only on the Kodak dataset and FID score on MS-COCO 30k as these are the only results available, and cannot compare to Careil *et al.* For all other methods, we use released pretrained model weights and run with default parameters. However, for CDC we increase the number of denoising sampling steps to 1000 to produce higher quality reconstructions.

User study. As qualitative metrics often fail to capture the perceptual quality of image reconstructions [10, 41], we further perform a user study to assess the visual quality of our results. The study is set up as a two-alternative forced choice (2AFC), where each participant is shown the source image and reconstructions from two methods and is asked to choose the reconstruction they prefer. We select 10 samples from the Kodak dataset with the smallest difference in bitrate between our method and the other generative model baselines, namely CDC, ILLM, and HFD.⁵ Thus, in a session, a participant is requested to do 60 pairwise comparisons. Each sample is center-cropped to 512×512 px so that all images being compared are shown side-by-side at native resolution (*i.e.*, without resampling).⁶ Participants can freely zoom and pan the images in a synchronized way. However, because the methods locally provide different types of reconstructions, we ask the participants to inspect the images in their entirety before rating.

Following [32], we use the Elo [17] rating system to rank the methods. Elo matches can be organized into tournaments, where ranking updates are applied only at the end of the tournament. We perform two separate experiments, where a tournament is considered to be *1.* a single comparison or *2.* all image comparisons from the same user. As Elo scores depend on game order, a Monte Carlo simulation is performed over 10,000 iterations, and we report the median score for each method.

⁵ In a pilot study, we have also considered to include BPG and HiFiC. However, since it was clear that they always performed worse than other methods in our target bitrate range, and to avoid fatigue with long rating sessions (we target a session time around 15–20min) we removed them from the study.

⁶ The images used in the study and respective bitrates can be found in the Supplementary Materials.

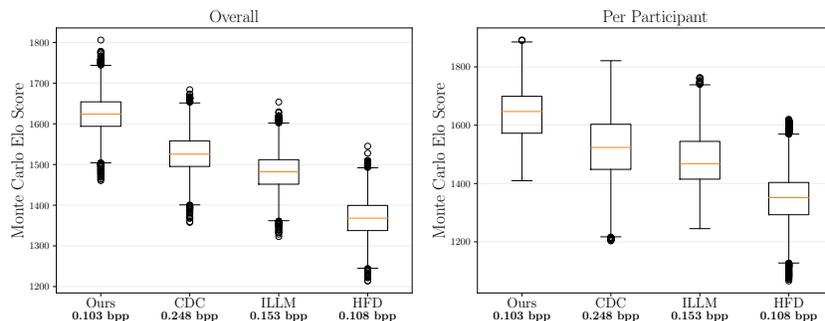


Fig. 6: Computed Elo ratings from the user study with Elo tournaments set for each comparison (left) or for each participant (right). Higher is better. The box extends to the first and third quartiles and the whiskers $1.5 \times IQR$ further.

5.1 Results

Quantitative results. Fig. 5 shows the rate-distortion (as measured by PSNR, MS-SSIM, and LPIPS) and rate-realism (as measured by FID) curves of our methods and baselines. Our method sets a new state-of-the-art in realism of reconstructed images, outperforming all baselines in FID-bitrate curves. In some distortion metrics (namely, LPIPS and MS-SSIM), we outperform all diffusion-based codecs while remaining competitive with the highest-performing generative codecs. As expected, our method and other generative methods suffer when measured in PSNR as we favor perceptually pleasing reconstructions instead of exact replication of detail (see Sec. 3).

User study. Fig. 6 shows the outcome of our user study. The methods are ordered by human preference according to Elo scores. The average bitrate of the images for each model are shown below the name of the methods. As can be seen in the Elo scores, our method significantly outperforms all the others, even compared to CDC, which uses on average double the bits of our method. This remains true regardless of Elo tournament strategy used.

Visual results. Fig. 7 qualitatively compares our method to generative neural image compression methods. Our approach can consistently reconstruct fine details and plausible textures while maintaining high realism. HFD often synthesizes incorrect content (door in row 1, flowers in row 5, and mural in row 6) or produces smooth reconstructions (flower petal in row 2, face in row 3, red barn in row 4, and face and hat in row 7). CDC and ILLM introduce unnatural blurry or high-frequency generative artifacts (flower bud in row 2 and tree in row 4) even in cases where they use 2x the bitrate of our method.

Complexity. To assess the practicality and efficiency of our method, we compare the runtime and size of our model with CDC and ILLM. We report the

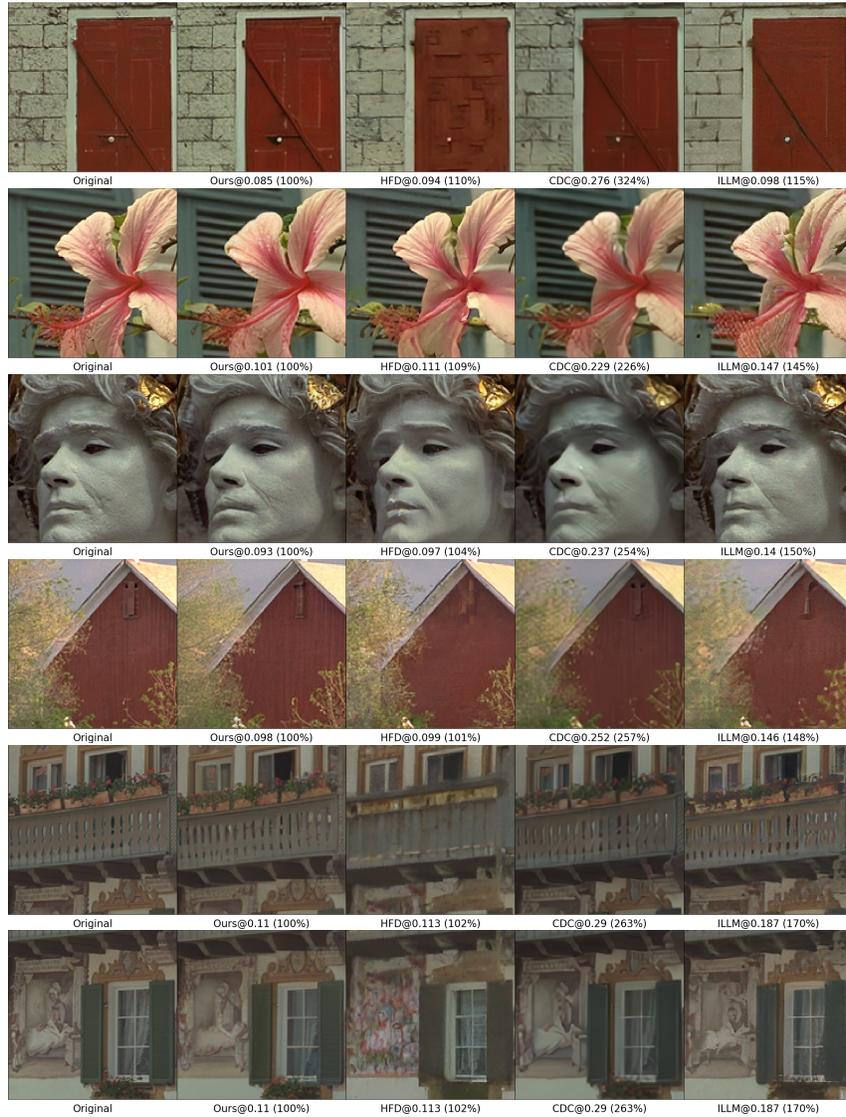


Fig. 7: Qualitative comparison of our method to the baselines. Images are labeled as Method@bpp (bpp is also shown as a percentage of our method). Best viewed digitally.

average encoding/decoding time (excluding entropy coding) for all images from the Kodak dataset. All benchmarks were performed on an NVIDIA RTX 3090 GPU. Our method processes an image in 3.49 seconds, nearly twice as fast as CDC, which requires 6.87 seconds. ILLM processes an image in 0.27 seconds. However, it is important to note that diffusion-based methods are in general slower than other codecs due to their iterative denoising nature. Due to the Sta-

ble Diffusion backbone, our method is more complex than CDC (1.3B vs. 53.6M parameters, respectively), while ILLM contains 181.5M parameters. However, the large majority of our parameters come from the diffusion backbone. Our trained modules (*e.g.*, \mathcal{P}_ϕ and the entropy model) contain only 36M parameters.

Reducing the computational burden of diffusion models is an active research area [25, 26], parallel to ours. Our method is fundamentally independent of the chosen foundation model, thus advances on reducing the complexity of Stable Diffusion can ultimately also improve our proposal.

Limitations. Similar to other generative approaches, our method can discard certain image features while synthesizing similar information at the receiver side. In specific cases, however, this might result in inaccurate reconstruction, such as bending straight lines or warping the boundary of small objects. These are well-known issues of the foundation model we build upon, which can be attributed to the relatively low feature dimension of its VAE. Despite this, our generated content is still closer to the original content than the other diffusion compression methods, as confirmed by our subjective study, and can be qualitatively compared on Fig. 7.

Ethical concerns. A core challenge of generative machine learning is the misgeneration of content. Specifically at very low bitrates, identities, text, or lower-level content can vary from the original image, and thus may raise ethical concerns in specific scenarios.

6 Conclusion

Via our proposed novel lossy image compression codec based on foundation latent diffusion, we produce realistic image reconstructions at low to very low bitrates, outperforming previous generative codecs in both perceptual metrics and subjective user preference. By combining the denoising capability of diffusion models with the inherent characteristics of quantization noise, our method predicts the ideal number of denoising steps to produce perceptually pleasing reconstructions over a range of bitrates with a single model. Our formulation has faster decoding time than previous diffusion codecs and, due to reusing a foundation model backbone, a much lower training budget. Potential future work includes the integration of more efficient backbone models [25, 26] and the support for user control to navigate the rate-distortion-realism trade-off.

References

1. BPG Image format. <https://bellard.org/bpg/>
2. Zlib. <https://www.zlib.net/>
3. Challenge on Learned Image Compression (2022)
4. Agustsson, E., Minnen, D., Toderici, G., Mentzer, F.: Multi-Realism Image Compression with a Conditional Generator (Mar 2023). <https://doi.org/10.48550/arXiv.2212.13824>
5. 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)
6. Bachard, T., Bordin, T., Maugey, T.: Coclico: Extremely low bitrate image compression based on clip semantic and tiny color map. In: Picture Coding Symposium 2024 (2024)
7. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end Optimized Image Compression (Mar 2017). <https://doi.org/10.48550/arXiv.1611.01704>
8. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior (May 2018). <https://doi.org/10.48550/arXiv.1802.01436>
9. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align Your Latents: High-Resolution Video Synthesis With Latent Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22563–22575 (2023)
10. Blau, Y., Michaeli, T.: Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff. In: Proceedings of the 36th International Conference on Machine Learning. pp. 675–685. PMLR (May 2019)
11. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
12. Bordin, T., Maugey, T.: Semantic based generative compression of images for extremely low bitrates. In: MMSP 2023 - IEEE 25th International Workshop on MultiMedia Signal Processing. pp. 1–6. IEEE, Poitiers, France (Sep 2023), <https://hal.science/hal-04231421>
13. Careil, M., Muckley, M.J., Verbeek, J., Lathuilière, S.: Towards image compression with perfect realism at ultra-low bitrates (Oct 2023). <https://doi.org/10.48550/arXiv.2310.10325>
14. Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., Li, Z.: PixArt- α : Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis (Oct 2023). <https://doi.org/10.48550/arXiv.2310.00426>
15. Dhariwal, P., Nichol, A.Q.: Diffusion Models Beat GANs on Image Synthesis. In: Advances in Neural Information Processing Systems (Nov 2021)
16. Forsgren, S., Martiros, H.: Riffusion - Stable diffusion for real-time music generation (2022), <https://riffusion.com/about>
17. Glickman, M.E.: A Comprehensive Guide to Chess Ratings. American Chess Journal **3**(1), 59–102 (1995)
18. Goose, N.F., Petersen, J., Wiggers, A., Xu, T., Sautière, G.: Neural Image Compression with a Diffusion-Based Decoder (Jan 2023). <https://doi.org/10.48550/arXiv.2301.05489>

19. Graikos, A., Malkin, N., Jovic, N., Samaras, D.: Diffusion Models as Plug-and-Play Priors. *Advances in Neural Information Processing Systems* **35**, 14715–14728 (Dec 2022)
20. 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)
21. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus) (2023)
22. 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)
23. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: *Advances in Neural Information Processing Systems*. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020)
24. Hoogeboom, E., Agustsson, E., Mentzer, F., Versari, L., Toderici, G., Theis, L.: High-Fidelity Image Compression with Score-based Generative Models (May 2023). <https://doi.org/10.48550/arXiv.2305.18231>
25. Hoogeboom, E., Heek, J., Salimans, T.: Simple diffusion: End-to-end diffusion for high resolution images. In: *Proceedings of the 40th International Conference on Machine Learning. ICML'23*, vol. 202, pp. 13213–13232. JMLR.org, Honolulu, Hawaii, USA (Jul 2023)
26. Jabri, A., Fleet, D.J., Chen, T.: Scalable adaptive computation for iterative generation. In: *Proceedings of the 40th International Conference on Machine Learning. ICML'23*, vol. 202, pp. 14569–14589. JMLR.org, Honolulu, Hawaii, USA (Jul 2023)
27. Ke, B., Obukhov, A., Huang, S., Metzger, N., Dautt, R.C., Schindler, K.: Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation (Dec 2023). <https://doi.org/10.48550/arXiv.2312.02145>
28. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: *2nd International Conference on Learning Representations, ICLR* (2014)
29. Kodak: PhotoCD PCD0992 (1993)
30. Lei, E., Uslu, Y.B., Hassani, H., Bidokhti, S.S.: Text + Sketch: Image Compression at Ultra Low Rates (Jul 2023). <https://doi.org/10.48550/arXiv.2307.01944>
31. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In: *International Conference on Learning Representations* (2022)
32. Mentzer, F., Toderici, G., Tschannen, M., Agustsson, E.: High-Fidelity Generative Image Compression (Oct 2020). <https://doi.org/10.48550/arXiv.2006.09965>
33. Minnen, D., Ballé, J., Toderici, G.: Joint Autoregressive and Hierarchical Priors for Learned Image Compression (Sep 2018). <https://doi.org/10.48550/arXiv.1809.02736>
34. Muckley, M.J., El-Nouby, A., Ullrich, K., Jegou, H., Verbeek, J.: Improving Statistical Fidelity for Neural Image Compression with Implicit Local Likelihood Models. In: *Proceedings of the 40th International Conference on Machine Learning*. pp. 25426–25443. PMLR (Jul 2023)
35. Pasco, R.C.: Source coding algorithms for fast data compression (1976)
36. Qian, Y., Lin, M., Sun, X., Tan, Z., Jin, R.: Entroformer: A Transformer-based Entropy Model for Learned Image Compression (Mar 2022). <https://doi.org/10.48550/arXiv.2202.05492>
37. 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)

38. Shannon, C.E.: A Mathematical Theory of Communication. *The Bell System Technical Journal* **27**, 379–423 (1948)
39. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: *Proceedings of the 32nd International Conference on Machine Learning*. pp. 2256–2265. PMLR (Jun 2015)
40. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: *International Conference on Learning Representations* (Jan 2021)
41. Stein, G., Cresswell, J.C., Hosseinzadeh, R., Sui, Y., Ross, B.L., Villecroze, V., Liu, Z., Caterini, A.L., Taylor, J.E.T., Loaiza-Ganem, G.: Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models (Oct 2023). <https://doi.org/10.48550/arXiv.2306.04675>
42. Wallace, G.: The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* **38**(1), xviii–xxxiv (1992). <https://doi.org/10.1109/30.125072>
43. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video Enhancement with Task-Oriented Flow. *International Journal of Computer Vision* **127**(8), 1106–1125 (Aug 2019). <https://doi.org/10.1007/s11263-018-01144-2>
44. Yang, R., Mandt, S.: Lossy Image Compression with Conditional Diffusion Models. *Advances in Neural Information Processing Systems* **36**, 64971–64995 (Dec 2023)
45. Zhang, L., Rao, A., Agrawala, M.: Adding Conditional Control to Text-to-Image Diffusion Models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 3836–3847 (2023)