Towards Scalable Compression with Universally Quantized Diffusion Models

Yibo Yang* **Justus C. Will*** **Stephan Mandt**
Department of Computer Science
University of California, Irvine
{yibo.yang, jcwill, mandt}@uci.edu

Abstract

Diffusion probabilistic models have achieved success in many generative modeling tasks, from image generation to inverse problem solving. A distinct feature of these models is that they correspond to deep hierarchical latent variable models optimizing a variational evidence lower bound (ELBO) on the data likelihood. Drawing on a basic connection between likelihood modeling and compression, we explore the potential of diffusion models for progressive coding, resulting in a sequence of bits that can be incrementally transmitted and decoded with progressively improving reconstruction quality. Unlike prior work based on Gaussian diffusion or conditional diffusion models, we propose a new form of diffusion model with uniform noise in the forward process, whose negative ELBO corresponds to the end-to-end compression cost using universal quantization. We obtain promising first results on image compression, achieving competitive rate-distortion and rate-realism results on a wide range of bit-rates with a single model.

1 Introduction

Popularized by their impressive sample quality, diffusion models (Sohl-Dickstein et al., 2015) have quickly dominated the task of likelihood estimation (Kingma et al., 2021; Nichol & Dhariwal, 2021). Given the close connection between density estimation and data compression (MacKay, 2003; Yang et al., 2022), diffusion models have been shown to naturally lead to progressive compression codecs (Ho et al., 2020; Theis et al., 2022). Such as a *progressive* codec has the advantage of enabling dynamic rate-distortion (and computation) tradeoff while achieving high realism, all with a single model. Unfortunately, such a method requires the communication of Gaussian samples across many steps, and has to date not been implemented in a practical compression algorithm. In this work, we take first steps towards making such a diffusion-based progressive codec tractable. The key idea is to replace the Gaussian distributions in the forward diffusion process with suitable *uniform* distributions, and correspondingly adjust the reverse process distributions. These modifications allow the application of universal quantization (UQ) for simulating the uniform noise channel, avoiding the intractability of Gaussian channel simulation in the method of Theis et al. (2022).

2 Background and Related Work

Diffusion models Diffusion probabilistic models learn to model data by inverting a Gaussian noising process. Following the setup of VDM (Kingma et al., 2021), the forward noising process begins with a data observation $\mathbf{x}$, and defines a sequence of increasingly noisy latent variables $\mathbf{z}_t$ with a conditional Gaussian distribution,

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad t = 0, 1, ..., T.$$

*Equal contribution

Workshop on Machine Learning and Compression, Neural Information Processing Systems 2024.

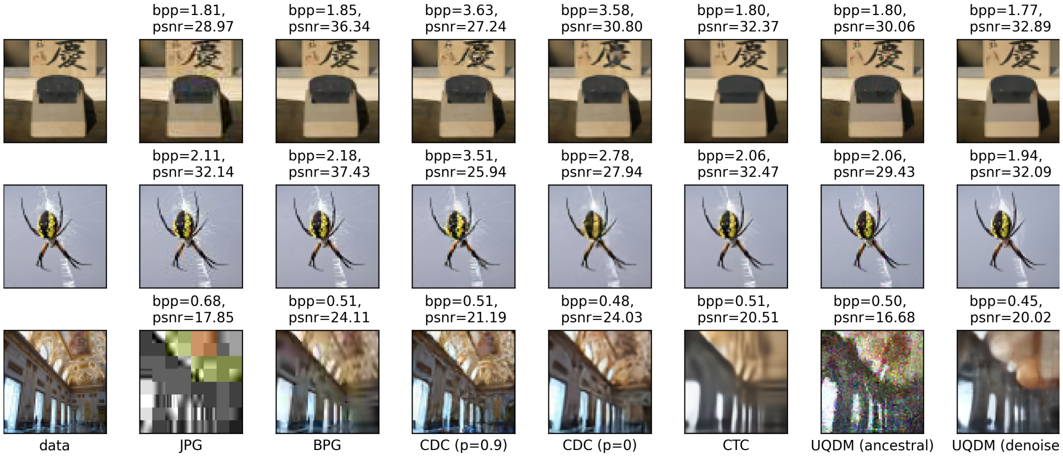

Figure 1: Example reconstructions from several traditional and neural codecs, picked to have roughly similar bitrates. At high bitrates, UQDM preserves details (e.g. shape and color pattern of the spider) better than other neural codecs. Reconstructions at low bitrates highlight the artifacts introduced by each codec. Note that only CTC and UQDM allow for progressive coding.

Here α_t and σ_t^2 are positive scalar-valued functions of time, with a strictly monotonically increasing *signal-to-noise-ratio* $\mathrm{SNR}(t) := \alpha_t^2/\sigma_t^2$. The *variance-preserving* process of (Ho et al., 2020) corresponds to the choice $\alpha_t^2 = 1 - \sigma_t^2$. The reverse-time generative model is defined by a collection of conditional distributions $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$, a prior $p(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and likelihood model $p(\mathbf{x}|\mathbf{z}_0)$. The conditional distributions $p(\mathbf{z}_{t-1}|\mathbf{z}_t) := q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t))$ are chosen to have the same distributional form as the "forward posterior" distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$, with $\mathbf{x}$ estimated from its noisy version $\mathbf{z}_t$ through the learned *denoising model* $\hat{\mathbf{x}}_\theta$. Further details on the forward and backward processes can be found in Appendix A and B. The model is trained by minimizing the negative ELBO

$$\mathcal{L}(\mathbf{x}) = \underbrace{\mathrm{KL}(q(\mathbf{z}_T|\mathbf{x}) \,\|\, p(\mathbf{z}_T))}_{:=L_T} + \underbrace{\mathbb{E}\left[-\log p(\mathbf{x}|\mathbf{z}_0)\right]}_{:=L_{\mathbf{x}|\mathbf{z}_0}} + \sum_{t=1}^{T} \underbrace{\mathbb{E}\left[\mathrm{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) \,\|\, p(\mathbf{z}_{t-1}|\mathbf{z}_t))\right]}_{:=L_{t-1}}, \quad (1)$$

where the expectations are taken with respect to the forward process $q(\mathbf{z}_{0:T}|\mathbf{x})$. Kingma et al. (2021) showed that a larger T corresponds to a tighter bound on the marginal likelihood $\log p(\mathbf{x})$, and as $T \to \infty$ the loss approaches the loss of a class of continuous-time diffusion models that includes the ones considered by Song et al. (2020).

Relative Entropy Coding (REC) Relative Entropy Coding (REC) deals with the problem of efficiently communicating a single sample from a target distribution q using a coding distribution p. Given a shared random number generator and "prior" distribution p between two parties, a Relative Entropy Coding (REC) method (Flamich et al., 2020; Theis & Ahmed, 2022) allows the sender to transmit a sample $\mathbf{z} \sim q$ using close to $\mathrm{KL}(q \,\|\, p)$ nats, up to a logarithmic overhead. A major challenge of REC algorithms is that their computational complexity generally scales exponentially with the amount of information being communicated (Agustsson & Theis, 2020; Goc & Flamich, 2024). This difficulty can be partly remedied by performing REC on sub-problems with lower dimensions (Flamich et al., 2020, 2022), for which computationally efficient REC algorithms exist (Flamich et al., 2024; Flamich, 2024), but comes at the expense of worse bitrate efficiency.

Progressive Coding with Diffusion Given a REC algorithm, we can use a trained diffusion model to perform progressive compression (Ho et al., 2020; Theis et al., 2022) as follows: Initially, at time T, the sender transmits a sample of $q(\mathbf{z}_T|\mathbf{x})$ under the prior $p(\mathbf{z}_T)$, using L_T nats on average. At each subsequent time step t, the sender transmits a sample of $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ given the previously transmitted $\mathbf{z}_t$, under the (conditional) prior $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$, using approximately L_{t-1} nats. Finally, at $t = 0$, $\mathbf{x}$ can be losslessly transmitted given $\mathbf{z}_0$ under the entropy model $p(\mathbf{x}|\mathbf{z}_0)$ using roughly $L_{\mathbf{x}|\mathbf{z}_0}$ nats. Thus the overall cost of losslessly compressing $\mathbf{x}$ sums up to $\mathcal{L}(\mathbf{x})$ nats, as in eq. (1). Crucially, at any time t, the receiver can make use of the most-recently received $\mathbf{z}_t$ to already estimate a data reconstruction $\hat{\mathbf{x}}_t$. For this, several options are possible: Ho et al. (2020) consider using the diffusion model's denoising prediction $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$, while Theis et al. (2022) consider sampling $\hat{\mathbf{x}}_t \sim p(\mathbf{x}|\mathbf{z}_t)$, either by ancestral sampling or a probability flow ODE (Song et al., 2020). Note that if the reverse generative model captures the data distribution perfectly, then $\hat{\mathbf{x}}_t \sim p(\mathbf{x}|\mathbf{z}_t)$ follows the same marginal distribution as the data and has the desirable property of *perfect realism*, i.e., being indistinguishable from real data (Theis et al., 2022). We note that several diffusion-based neural compression methods exist, but they use conditional diffusion models (Yang & Mandt, 2023; Careil et al., 2023; Hoogeboom et al., 2023) which are less flexible and do not permit progressive decoding. Other existing progressive neural compression methods are based on hierarchical VAEs (Lu et al., 2021; Lee et al., 2022; Jeon et al., 2023; Lee et al., 2024), and do not directly target realism.

Universal Quantization We focus on the special case of REC, where the target distribution q is defined by a uniform noise channel, which is solved efficiently by Universal Quantization (UQ) (Roberts, 1962; Zamir & Feder, 1992). Specifically, suppose we (the sender) have access to a scalar r.v. $Y \sim p_Y$, and would like to communicate a noise-perturbed version of it,

$$\tilde{Y} = Y + U,$$

where $U \sim \mathcal{U}(-\Delta/2, \Delta/2)$ is an independent r.v. with a uniform distribution on the interval $[-\Delta/2, \Delta/2]$. UQ accomplishes this as follows: *Step 1.* Perturb Y by adding another independent noise $U' \sim \mathcal{U}(-\Delta/2, \Delta/2)$, and quantize the result to the closet quantization point K on a uniform grid of width Δ, i.e., computing $K := \Delta\lfloor \frac{Y+U'}{\Delta} \rceil$ where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. *Step 2.* Entropy code and transmit K under the conditional distribution of K given U'. *Step 3.* The receiver draws the same U' by using the same random number generator and obtains a reconstruction $\hat{Y} := K - U' = \Delta\lfloor \frac{Y+U'}{\Delta} \rceil - U'$. Zamir & Feder (1992) showed that $\hat{Y}$ indeed has the same distribution as $\tilde{Y}$, and the entropy coding cost of K is related to the differential entropy of $\tilde{Y}$ via

$$H[K|U'] = I(Y; \tilde{Y}) = h(\tilde{Y}) - \log(\Delta).$$

In the above, the optimal entropy coding distribution $\mathbb{P}(K|U' = u')$ is obtained by discretizing $p_{\tilde{Y}} = p_Y * \mathcal{U}(-\Delta/2, \Delta/2)$ on a grid of width Δ and offset by $U' = u'$ (Zamir & Feder, 1992). If the true $p_{\tilde{Y}}$ is unknown, we can replace it with a surrogate density model $f_\theta(y)$ during entropy coding, and incur a higher coding cost,

$$\mathbb{E}_{y \sim P_Y}[\mathrm{KL}(u(\cdot|y) \,\|\, f_\theta(\cdot))] \geq I(Y; \tilde{Y}), \tag{2}$$

where $u(\cdot|y)$ denotes the density function of the uniform noise channel $q_{\tilde{Y}|Y=y} = \mathcal{U}(y - \Delta/2, y + \Delta/2)$. It can be shown that the optimal choice of f_θ is the convolution of p_Y with $\mathcal{U}(-\Delta/2, \Delta/2)$. Therefore, as in prior work (Agustsson & Theis, 2020; Ballé et al., 2018), we will choose f_θ to have the form of another underlying density model g_θ convolved with uniform noise, i.e.

$$f_\theta(\cdot) = g_\theta(\cdot) * \mathcal{U}(\cdot\,; -\Delta/2, \Delta/2). \tag{3}$$

3 Universally Quantized Diffusion Models

We follow the same conceptual framework of progressive compression with diffusion models as in (Ho et al., 2020; Theis et al., 2022), but propose to avoid the communication of Gaussian samples by using UQ instead. We therefore introduce a new model with a modified forward process and reverse process, which we term *universally quantized diffusion model* (UQDM).

3.1 Forward process

The forward process of a standard diffusion model is often given by the transition kernel $q(\mathbf{z}_{t+1}|\mathbf{z}_t)$ (Ho et al., 2020), which in turn determines the conditional (reverse-time) distributions $q(\mathbf{z}_T|\mathbf{x})$ and $\{q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})|t = 1, ..., T\}$ appearing in the NELBO eq. (1). As we are interested in operationalizing and optimizing the coding cost associated with eq. (1), we will directly specify these conditional distributions to be compatible with UQ, rather than deriving them from a transition kernel. We thus specify the forward process with the same factorization as in DDIM (Song et al., 2021) via $q(\mathbf{z}_{0:T}|\mathbf{x}) = q(\mathbf{z}_T|\mathbf{x}) \prod_{t=1}^{T} q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$. Specifically, we consider

$$\begin{cases} q(\mathbf{z}_T|\mathbf{x}) := \mathcal{N}(\alpha_T \mathbf{x}, \sigma_T^2 \mathbf{I}), \\ q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) := \mathcal{U}\left(b(t)\mathbf{z}_t + c(t)\mathbf{x} - \frac{\Delta(t)}{2}, b(t)\mathbf{z}_t + c(t)\mathbf{x} + \frac{\Delta(t)}{2}\right), t = 1, 2, ..., T, \end{cases} \tag{4}$$

where $b(t)$, $c(t)$, and $\Delta(t)$ are scalar-valued functions of time. Note that unlike in Gaussian diffusion, $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ is chosen to be a uniform distribution so that it can be efficiently simulated with UQ. There is freedom in other choices of the forward process, but for simplicity we base them closely on the Gaussian case: we choose the same Gaussian $q(\mathbf{z}_T|\mathbf{x})$, and set $b(t)$, $c(t)$, $\Delta(t)$ so that $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ has the same mean and variance as in the Gaussian case (see Sec. A for more details).

We note here that $q(\mathbf{z}_t|\mathbf{z}_T, \mathbf{x})$ can be written as a sum of uniform distributions, which as we increase $T \to \infty$, converges in distribution to a Gaussian by the Central Limit Theorem. Under the assumptions $\alpha_T = 0$ and $\sigma_T = 1$, the forward process $q(\mathbf{z}_t|\mathbf{x})$ therefore also converges to a Gaussian, showing that our forward process has the same underlying continuous-time limit as in VDM (Kingma et al., 2021). See Appendix A.3 for details and proof. As in VDM (Kingma et al., 2021), the forward process schedules (i.e., α_t and σ_t, as well as $b(t), c(t), \Delta(t)$) can be learned end-to-end, e.g., by parameterizing $\sigma_t^2 = \mathrm{sigmoid}(\phi(t))$, where ϕ is a monotonic neural network. We did not find this to yield significant improvements compared to using a linear noise schedule as in (Kingma et al., 2021).

3.2 Backward process

Analogously to the Gaussian case, we want to define a conditional distribution $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ that leverages a denoising model $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ and closely matches the forward "posterior" $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$. In our case, the forward "posterior" corresponds to a uniform noise channel with width $\Delta(t)$, i.e., $\mathbf{z}_{t-1} = b(t)\mathbf{z}_t + c(t)\mathbf{x} + \Delta(t)\mathbf{u}_t, \mathbf{u}_t \sim \mathcal{U}(-{}^{\mathbf{1}}\!/\!_{\mathbf{2}}, {}^{\mathbf{1}}\!/\!_{\mathbf{2}})$; to simulate it with UQ, we choose a density model for $\mathbf{z}_{t-1}$ with the same form as the convolution in eq. (3). Specifically, we let

$$p(\mathbf{z}_{t-1}|\mathbf{z}_t) = g_\theta(\mathbf{z}_{t-1}; \mathbf{z}_t, t) \star \mathcal{U}(-{}^{\Delta(t)}\!/\!_{\mathbf{2}}, {}^{\Delta(t)}\!/\!_{\mathbf{2}}), \tag{5}$$

where $g_\theta(\mathbf{z}_{t-1}; \mathbf{z}_t, t)$ is a learned density chosen to match $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$. Recall in Gaussian diffusion (Kingma et al., 2021), $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ is chosen to be a Gaussian of the form $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}}_\theta(\mathbf{z}_t; t))$, i.e., the same as $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ but with the original data $\mathbf{x}$ replaced by a denoised prediction $\mathbf{x} = \hat{\mathbf{x}}_\theta(\mathbf{z}_t; t)$. For simplicity, we base g_θ closely on the choice of $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ in Gaussian diffusion, e.g.,

$$g_\theta(\mathbf{z}_{t-1}; \mathbf{z}_t, t) = \mathcal{N}(b(t)\mathbf{z}_t + c(t)\hat{\mathbf{x}}_\theta(\mathbf{z}_t; t), \sigma_Q^2(t)\mathbf{I}), \tag{6}$$

where $\sigma_Q^2(t)$ is the variance of the Gaussian forward "posterior", and we use the same noise-prediction network for $\hat{\mathbf{x}}_\theta$ as in (Kingma et al., 2021). We found that a logistic distribution with the same mean and variance to be numerically more stable than the Gaussian, and adopt it in our experiments. Inspired by (Nichol & Dhariwal, 2021), we found that learning a per-coordinate variance in the reverse process to significantly improve the log-likelihood, which we demonstrate in Sec. 4. In practice, this is implemented by doubling the output dimension of the score network to also compute a tensor of scaling factors $\mathbf{s}_\theta(\mathbf{z}_t)$, so that the variance of g_θ is $\boldsymbol{\sigma}_\theta^2 = \sigma_Q^2(t) \odot \mathbf{s}_\theta(\mathbf{z}_t)$. We adopt the same form of categorical likelihood model $p(\mathbf{x}|\mathbf{z}_0)$ as in VDM (Kingma et al., 2021), as well as the use of Fourier features.

4 Experiments

We train UQDM end-to-end and perform compression experiments on toy swirl data, CIFAR10, and ImageNet64×64. When comparing UQDM with VDM (Kingma et al., 2021), we always use the same U-net architecture for both, except UQDM uses twice as many output dimensions for both the denoising prediction and learned reverse-process variance (see Sec. 3). We refer to Appendix Sec. C for further experiment and implementation details.

4.1 Swirl Toy Data

We obtain initial insights into the behavior of our proposed UQDM by experimenting on toy swirl data and comparing with VDM (Kingma et al., 2021).

First, we train UQDM with various T, and ablate on learning the reverse process variance. For comparison, we also train a single VDM with $T = 1000$, but compute the progressive-coding NELBO eq. (1) for various values of T. Fig. 2 plots the resulting NELBO, corresponding to the bits-per-dimension of lossless compression. We observe that for UQDM, learning the reverse-process variance significantly improved the NELBO across all T, and a higher T is not necessarily better.

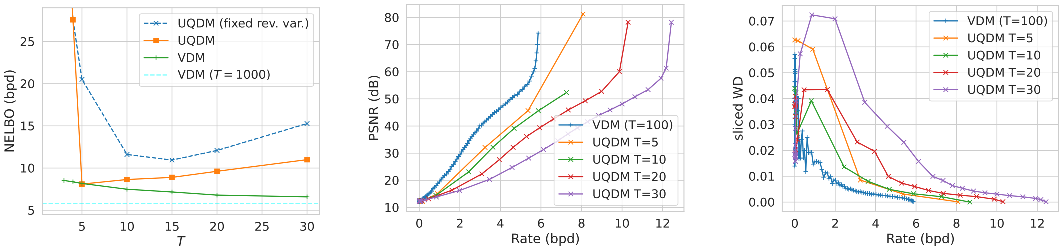

Figure 2: Results on swirl data. **Left**: Lossless compression rates v.s. the choice of T, for UQDM with/without learned reverse-process variance (blue/orange) and VDM (green). For UQDM, learning the reverse-process variance significantly improved the NELBO, and an optimal $T \approx 5$. **Middle, Right**: Progressive lossy compression performance for VDM (hypothetical) and UQDM, measured in fidelity (PSNR) v.s. bit-rate (middle), or realism (sliced Wasserstein distance) v.s. bit-rate (right).

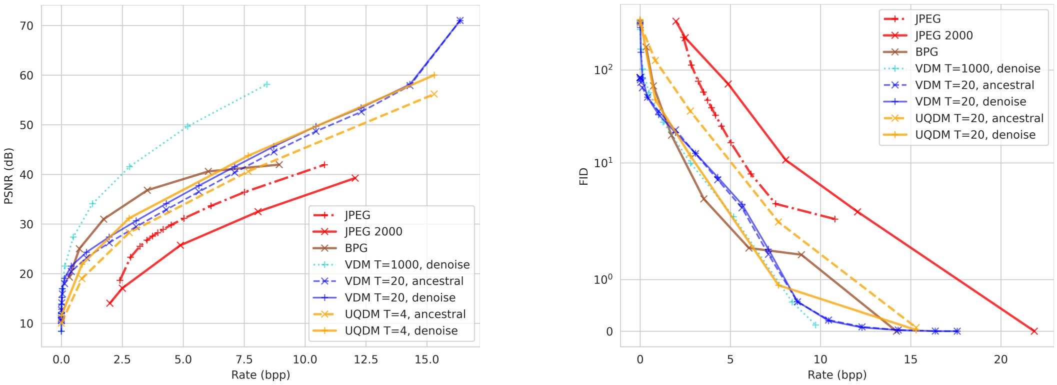

Figure 3: Progressive lossy compression performance of UQDM on the CIFAR10 dataset, comparing reconstruction quality (PSNR) and realism (FID) with bit-rate per pixel (bpp).

In fact, we find the optimal $T \approx 5$, yielding a bpd of around 8. The performance of VDM, by comparison, monotonically improves with T (green), until it converges to a bpd of 5.8 at $T = 1000$.

We then examine the lossy compression performance of progressive coding. Here we train UQDM end-to-end with learned reverse-process variances, and perform progressive reconstruction by ancestral sampling. Fig. 2 plots the results in fidelity v.s. bit-rate and realism v.s. bit-rate. For reference, we also show the theoretical performance of VDM using $T = 100$ discretization steps, assuming a hypothetical REC algorithm that operates with no overhead. The results are consistent with those on lossless compression, with a similar performance ranking for T among UQDM, and a gap remains to the hypothetical performance of VDM.

4.2 CIFAR10

Next, we apply our method to CIFAR10 images. For our UQDM model we empirically note that $T = 4$ yields the best trade-off between bit-rates and reconstruction quality. We train end-to-end on the progressive coding NELBO eq. (1) with learned reverse-process variances. We compare against the wavelet-based codecs JPEG, JPEG2000, and BPG (Bellard, 2018). For JPEG and BPG we use a fixed set of quality levels and encode the images independently, for JPEG2000 we use its progressive compression mode to obtain a rate-distortion curve from one bitstream. As shown in Figure 3, we consistently outperform both JPEG and JPEG2000 over all bitrates and metrics. Even though BPG, achieves better reconstruction fidelity (as measured in PSNR) in the low bit-rate regime, our method closely matches BPG in realism (as measured in FID) and even beats BPG in PSNR at higher bit-rates. The theoretical performance of compression with Gaussian diffusion (e.g., VDM) (Theis et al., 2022), especially with a high number of steps such as $T = 1000$, is computationally infeasible, both due to the the large number of neural function evaluations required, and due the intractable runtime of REC algorithms in the Gaussian case. Still, for reference we report theoretical results both for $T = 1000$

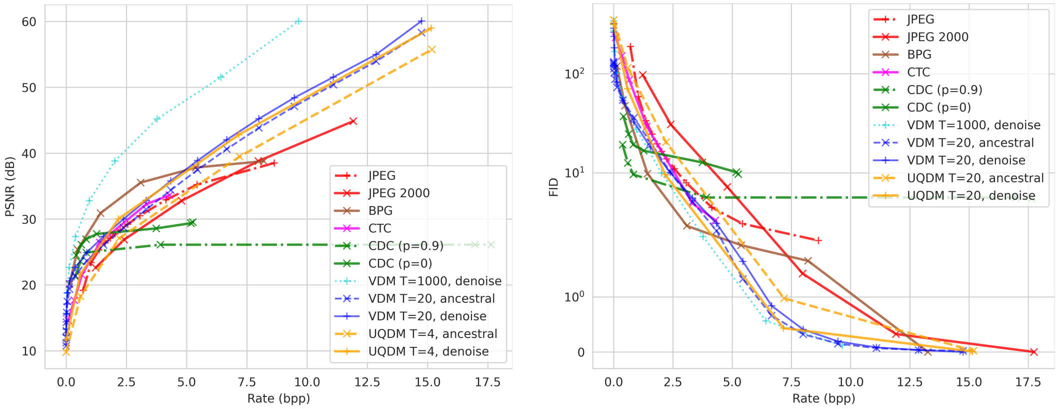

Figure 4: Progressive lossy compression performance of UQDM on the Imagenet dataset, comparing reconstruction quality (PSNR) and realism (FID) with bit-rate per pixel (bpp). While the reconstruction quality of other codecs plateaus at higher bitrates, our method continues to gradually improve quality and realism even at higher bitrates.

and $T = 20$, where the latter uses a smaller and more practical number of diffusion/progressive reconstruction steps.

4.3 ImageNet 64 × 64

Finally, we present result on the ImageNet 64 × 64 dataset. We train a baseline VDM model with the same architecture as in (Kingma et al., 2021), reproducing their reported BPD of around 3.4, and a UQDM of the same architecture with learned reverse-process variances and $T = 4$. In addition to the baselines described in the previous section we also compare with CTC (Jeon et al., 2023), a recent progressive neural codec, and CDC (Yang & Mandt, 2023), a non-progressive neural codec based on a conditional diffusion model that can trade-off between distortion and realism via a hyperparameter p. We separately report results for both $p = 0$, which only optimizes PSNR, and $p = 0.9$, which prioritizes more realistic reconstructions. For CTC we use pre-trained model checkpoints from the official implementation (Jeon et al., 2023); for CDC we fix the architecture but train a new model for each bit-rate v.s. reconstruction quality/realism trade-off. The results are shown in Figure 4. When obtaining progressive reconstructions from denoised predictions, UQDM again outperforms both JPEG and JPEG2000. Our results are comparable to, if not slightly better than, CTC and even though the reconstruction quality of other codecs plateaus at higher bitrates, our method continues to gradually improve quality and realism even at higher bitrates. Refer to Fig.1 and 5 for qualitative results demonstrating progressive coding and comparison across codecs.

5 Discussion

In this paper, we presented a new progressive coding scheme based on a novel adaptation of the standard diffusion model. Our universally quantized diffusion model (UQDM) implements the idea of progressive compression with an unconditional diffusion model (Theis et al., 2022), but bypasses the intractability of Gaussian channel simulation by using universal quantization (Zamir & Feder, 1992) instead. We present promising first results that match or outperform classic and neural compression baselines, including a recent progressive neural image compression method (Jeon et al., 2023). Given the practical advantages of a progressive neural codec – allowing for dynamic trade-offs between rate, distortion and computation, support for both lossy and lossless compression, and potential for high realism, all in a single model – our approach brings neural compression a step closer towards real-world deployment.

Future work may close the performance gap between our method and that of Gaussian diffusion (Theis et al., 2022), by e.g., considering improved reconstruction schemes (e.g., based on an ODE as in DiffC-F (Theis et al., 2022)), alternative forward/reverse process specification than ours, or investigate further efficiency improvements based on ideas such as latent diffusion (Rombach et al., 2022), distillation (Sauer et al., 2024), or consistency models (Song et al., 2023).

References

E. Agustsson and L. Theis. Universally Quantized Neural Compression. In *Neural Information Processing Systems*, 2020.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational Image Compression with a Scale Hyperprior. *ICLR*, 2018.

Johannes Ballé, Sung Jin Hwang, Nick Johnston, and David Minnen. Tensorflow-compression: Data compression in tensorflow. URL https://github.com/tensorflow/compression.

F. Bellard. Bpg image format, 2018. https://bellard.org/bpg/.

Marlène Careil, Matthew J Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International Conference on Learning Representations*, 2023.

Gergely Flamich. Greedy poisson rejection sampling. *Advances in Neural Information Processing Systems*, 36, 2024.

Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. Compressing images by encoding their latent representations with relative entropy coding. *Advances in Neural Information Processing Systems*, 33:16131–16141, 2020.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Fast relative entropy coding with a* coding. In *International Conference on Machine Learning*, pp. 6548–6577. PMLR, 2022.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Faster relative entropy coding with greedy rejection coding. *Advances in Neural Information Processing Systems*, 36, 2024.

Daniel Goc and Gergely Flamich. On channel simulation with causal rejection samplers. *arXiv preprint arXiv:2401.16579*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Eirikur Agustsson, Fabian Mentzer, Luca Versari, George Toderici, and Lucas Theis. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023.

Seungmin Jeon, Kwang Pyo Choi, Youngo Park, and Chang-Su Kim. Context-based trit-plane coding for progressive image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14348–14357, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Neural Information Processing Systems*, 34:21696–21707, 2021.

Jae-Han Lee, Seungmin Jeon, Kwang Pyo Choi, Youngo Park, and Chang-Su Kim. Dpict: Deep progressive image compression using trit-planes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16113–16122, 2022.

Jooyoung Lee, Se Yoon Jeong, and Munchurl Kim. Deephq: Learned hierarchical quantizer for progressive deep image coding. *arXiv preprint arXiv:2408.12150*, 2024.

Yadong Lu, Yinhao Zhu, Yang Yang, Amir Said, and Taco S Cohen. Progressive neural image compression with nested quantization and latent ordering. In *2021 IEEE International conference on image processing (ICIP)*, pp. 539–543. IEEE, 2021.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171, 2021.

Lawrence Roberts. Picture coding using pseudo-random noise. *IRE Transactions on Information Theory*, 8(2):145–154, 1962.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:257280191.

Lucas Theis and Noureldin Y Ahmed. Algorithms for the communication of samples. In *International Conference on Machine Learning*, pp. 21308–21328, 2022.

Lucas Theis, Tim Salimans, Matthew D Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Neural Information Processing Systems*, 36, 2023.

Yibo Yang, Stephan Mandt, and Lucas Theis. An introduction to neural data compression. *arXiv preprint arXiv:2202.06533*, 2022.

R. Zamir and M. Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, 1992.

Appendix

A Forward process details

A.1 Gaussian (DDPM/VDM)

For completeness and reference, we restate the forward process and related conditionals given in (Kingma et al., 2021). The forward process is defined by

$$q(\mathbf{z}_t|\mathbf{x}) \coloneqq \mathcal{N}(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}),$$

where α_t and σ_t^2 are positive scalar-valued functions of t. As in (Kingma et al., 2021), we define the following notation shorthand which are used in the rest of the appendix: for any $s < t$, let

$$\alpha_{t|s} \coloneqq \frac{\alpha_t}{\alpha_s}, \quad \sigma_{t|s}^2 \coloneqq \sigma_t^2 - \frac{\alpha_t^2}{\alpha_s^2}\sigma_s^2, \quad b_{t|s} \coloneqq \frac{\alpha_t}{\alpha_s}\frac{\sigma_s^2}{\sigma_t^2}, \quad c_{t|s} \coloneqq \sigma_{t|s}^2 \frac{\alpha_s}{\sigma_t^2}, \quad \beta_{t|s} \coloneqq \sigma_{t|s}\frac{\sigma_s}{\sigma_t}.$$

By properties of the Gaussian distribution, it can be shown that for any $0 \leq s < t \leq T$,

$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\alpha_{t|s}\mathbf{x}, \sigma_{t|s}^2 \mathbf{I}),$$
$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \mathcal{N}(b_{t|s}\mathbf{z}_t + c_{t|s}\mathbf{x}, \beta_{t|s}^2 \mathbf{I}),$$

In particular,

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) = \mathcal{N}(b_{t|t-1}\mathbf{z}_t + c_{t|t-1}\mathbf{x}, \beta_{t|t-1}^2 \mathbf{I}),$$
$$q(\mathbf{z}_t|\mathbf{z}_T, \mathbf{x}) = \mathcal{N}(b_{T|t}\mathbf{z}_t + c_{T|t}\mathbf{x}, \beta_{T|t}^2 \mathbf{I}),$$

and we can use the reparameterization trick to write

$$\mathbf{z}_{t-1} = b_{t|t-1}\,\mathbf{z}_t + c_{t|t-1}\,\mathbf{x} + \beta_{t|t-1}\,\boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{7}$$
$$\mathbf{z}_t = b_{T|t}\,\mathbf{z}_T + c_{T|t}\,\mathbf{x} + \beta_{T|t}\,\boldsymbol{\epsilon}_T,\ \boldsymbol{\epsilon}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8}$$

A.2 Uniform (Ours)

Our forward process is specified by $q(\mathbf{z}_T|\mathbf{x})$ and $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ for each t, and closely follows that of the Gaussian diffusion. We set $q(\mathbf{z}_T|\mathbf{x})$ to be the same as in the Gaussian case, i.e.,

$$q(\mathbf{z}_T|\mathbf{x}) \coloneqq \mathcal{N}(\alpha_T \mathbf{x}, \sigma_T^2 \mathbf{I}),$$

and $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ to be a uniform with the same mean and variance as in the Gaussian case, s.t.

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) \coloneqq \mathcal{U}(b_{t|t-1}\mathbf{z}_t + c_{t|t-1}\mathbf{x} - \sqrt{3}\beta_{t|t-1}, b_{t|t-1}\mathbf{z}_t + c_{t|t-1}\mathbf{x} + \sqrt{3}\beta_{t|t-1}),$$

or in other words,

$$\mathbf{z}_{t-1} = b_{t|t-1}\mathbf{z}_t + c_{t|t-1}\mathbf{x} + \sqrt{12}\beta_{t|t-1}\mathbf{u}_t, \quad \mathbf{u}_t \sim \mathcal{U}(-\mathbf{1}/\mathbf{2}, \mathbf{1}/\mathbf{2}).$$

In the notation of eq. (4) this corresponds to letting $b(t) = b_{t|t-1}$, $c(t) = c_{t|t-1}$, $\Delta(t) = \sqrt{12}\beta_{t|t-1}$. It follows by algebraic manipulation that

$$\mathbf{z}_t = b_{T|t}\,\mathbf{z}_T + c_{T|t}\,\mathbf{x} + \underbrace{\sum_{v=t+1}^{T} \sqrt{12}\delta_{v|t}\mathbf{u}_v}_{\coloneqq \boldsymbol{\omega}_t},$$

where

$$\mathbf{u}_v \sim \mathcal{U}(-\mathbf{1}/\mathbf{2}, \mathbf{1}/\mathbf{2}), v = t+1, ..., T$$

are independent uniform noise variables, and

$$\delta_{v|t} = \beta_{v|v-1} \prod_{j=t+1}^{v-1} b_{j|j-1} = \frac{\sigma_t^2}{\alpha_t}\sqrt{\mathrm{SNR}(v-1) - \mathrm{SNR}(v)}.$$

We can show that

$$\mathrm{Var}\,(\boldsymbol{\omega}_t) = \sum_{v=t+1}^{T} \delta_{v|t}^2 = \frac{\sigma_t^4}{\alpha_t^2}[\mathrm{SNR}(t) - \mathrm{SNR}(T)] = \beta_{T|t}^2,$$

or in other words, our forward-process "posterior" distribution $q(\mathbf{z}_t|\mathbf{z}_T, \mathbf{x})$ at any step t has the same mean and variance as in the Gaussian case.

A.3 Convergence to the Gaussian case

We show that both parameterizations are equivalent in the continuous-time limit. To allow comparison across different number of steps T, we suppose that α_t and σ_t are obtained from continuous-time schedules $\alpha(\cdot) : [0,1] \to \mathbb{R}^+$ and $\sigma(\cdot) : [0,1] \to \mathbb{R}^+$ (which were fixed ahead of time), such that $\alpha_t := \alpha(t/T)$ and $\sigma_t := \sigma(t/T)$ for $t = 0, \dots, T$, for any choice of T. We further assume that the continuous-time signal-to-noise ratio $\operatorname{snr}(\cdot) = \alpha(\cdot)^2/\sigma(\cdot)^2$ is strictly monotonically decreasing.

Theorem A.1.
For every t, $q(\mathbf{z}_t|\mathbf{z}_T, \mathbf{x}) \xrightarrow{d} \mathcal{N}(b_{T|t}\, \mathbf{z}_T + c_{T|t}\, \mathbf{x}, \beta^2_{T|t}\, \mathbf{I})$ as $T \to \infty$.

Proof.
As $\operatorname{snr}(t)$ by assumption is both continuous, strictly monotone, and defined on a compact domain, it has finite range and is thus uniformly continuous. For $\sigma^2_{ni} := 12\sigma_0^4/\alpha_0^2(\operatorname{snr}((i-1)/n) - \operatorname{snr}(i/n))$ the latter implies $\max_{i\in\{1,\dots,n\}} \sigma^2_{ni} \to 0$ as $n \to \infty$. Let $X_{ni} := \sigma_{ni}\, \mathbf{u}_{ni}, \mathbf{u}_{ni} \sim \mathcal{U}(-\mathbf{1/2}, \mathbf{1/2})$ iid, then X_{ni} is a triangular array with independent rows, $\mathbb{E}\left[X_{ni}\right] = 0$, and $\operatorname{Var}\left(X_{ni}\right) = \sigma^2_{ni} < \infty$. Thus, we can apply the Lindeberg-Feller CLT which yields that $Z_n := \sum_{i=1}^n X_{ni} \xrightarrow{d} \mathcal{N}(0, s)$ if

$$\frac{1}{s}\sum_{i=1}^{n} \mathbb{E}\left[X^2_{ni}\mathbf{1}\{|X_{ni}| \geq \epsilon\}\right] \xrightarrow{n\to\infty} 0$$

holds for all $\epsilon > 0$. In this case, $s = \operatorname{Var}\left(Z_n\right) = \sigma_0^4/\alpha_0^2(\operatorname{snr}(0) - \operatorname{snr}(1)) = \beta^2_{T|0}$. The condition holds trivially as by construction $P(|X_{ni}| \geq \sqrt{3}\,\sigma_{ni}) = 0$ and for every $\epsilon > 0$ there exists N_ϵ with $\epsilon > \sqrt{3}\,\sigma_{ni}$ for all i and $n > N_\epsilon$ as $\max_{i\in\{1,\dots n\}} \sigma^2_{ni} \to 0$. The statement follows for $t = 0$ as $Z_n \sim \boldsymbol{\omega}_t|_{T=n}$, and analogously for arbitrary t by considering $\sigma^2_{ni} := 12\sigma_t^4/\alpha_t^2(\operatorname{snr}(t + (i-1)(1-t)/n) - \operatorname{snr}(t + i(1-t)/n))$. □

Corollary A.1.1.
If we assume $\sigma_T = 1$ and $\alpha_T = 0$, then for every t, $q(\mathbf{z}_t|\mathbf{x}) \xrightarrow{d} \mathcal{N}(\alpha_t\mathbf{x}, \sigma_t^2\mathbf{I})$ as $T \to \infty$, that is, our forward model approaches the Gaussian forward process of VDM with an increasing number of diffusion steps.

Proof. As $q(\mathbf{z}_T|x) = \mathcal{N}(\alpha_T\mathbf{x}, \sigma_T^2\mathbf{I}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ does not depend on T, the joint distribution $q(\mathbf{z}_t, \mathbf{z}_T|x) = q(\mathbf{z}_t|\mathbf{z}_T, \mathbf{x})q(\mathbf{z}_T|x)$ converges in distribution, which in turn implies convergence of $q(\mathbf{z}_t|x)$. The statement follows from $\mathcal{N}(\mathbf{z}_t; \alpha_t\mathbf{x}, \sigma_t^2\mathbf{I}) = \int \mathcal{N}(\mathbf{z}_t; b_{T|t}\, \mathbf{z}_T + c_{T|t}\, \mathbf{x}, \beta^2_{T|t}\, \mathbf{I})\mathcal{N}(\mathbf{z}_T; \alpha_T\mathbf{x}, \sigma_T^2\mathbf{I})d\mathbf{z}_T$. □

B Backward process details

B.1 Gaussian (DDPM/VDM)

Kingma et al. (2021) set $p(\mathbf{z}_{t-1}|\mathbf{z}_t) := q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}}_t) = \mathcal{N}(b_{t|t-1}\, \mathbf{z}_t + c_{t|t-1}\, \hat{\mathbf{x}}_t, \beta_{t|t-1}\mathbf{I})$ which yields

$$\begin{aligned} L_t &= \operatorname{KL}(\mathcal{N}(b_{t|t-1}\, \mathbf{z}_t + c_{t|t-1}\, \mathbf{x}, \beta_{t|t-1}\mathbf{I}) \,\|\, \mathcal{N}(b_{t|t-1}\, \mathbf{z}_t + c_{t|t-1}\, \hat{\mathbf{x}}_t, \beta_{t|t-1}\mathbf{I})) \\ &= \frac{1}{2}\frac{c^2_{t|t-1}}{\beta^2_{t|t-1}} \left\|\mathbf{x} - \hat{\mathbf{x}}_0\right\|_2^2 = \frac{1}{2}(\operatorname{SNR}(t-1) - \operatorname{SNR}(t)) \left\|\mathbf{x} - \hat{\mathbf{x}}_0\right\|_2^2 . \end{aligned}$$

We have that $L_t \to 0$ as $T \to \infty$, due to the continuity of $\operatorname{SNR}(\cdot\,/T) = \operatorname{snr}(\cdot) = \alpha(\cdot)^2/\sigma(\cdot)^2$.

B.2 Uniform (Ours)

Recall that we choose each coordinate of the reverse-process model $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$ to have the density

$$\begin{aligned} p(\mathbf{z}_{t-1}|\mathbf{z}_t)_i &:= g_t(z) \star \mathcal{U}(z; -\Delta/2, \Delta/2) \\ &= \frac{1}{\Delta_t}\int_{z-\Delta_t/2}^{z+\Delta_t/2} g_t(z)\, dz = \frac{1}{\Delta_t}(G_t(z + \Delta_t/2) - G_t(z - \Delta_t/2)). \end{aligned}$$

Using the shorthand $\mu_t = b_{t|t-1}z + c_{t|t-1}x$ (here, $z := (\mathbf{z}_t)_i$ and $x := (\mathbf{x})_i$ extract the ith coordinate), we can derive the rate associated with the ith coordinate

$$\begin{aligned} L_t &= \mathrm{KL}(\mathcal{U}(z; \mu_t - {}^{\Delta_t}\!/2, \mu_t + {}^{\Delta_t}\!/2) \,\|\, g_t(z) \star \mathcal{U}(z; -\Delta_t, \Delta_t)) \\ &= \frac{1}{\Delta_t} \int_{\mu_t - \Delta_t/2}^{\mu_t + \Delta_t/2} \log \frac{\frac{1}{\Delta} \mathbf{1}_{[\mu_t - \Delta_t/2, \mu_t + \Delta_t/2]}(z)}{\frac{1}{\Delta}(G_t(z + {}^{\Delta_t}\!/2) - G_t(z - {}^{\Delta_t}\!/2))} \, dz \\ &= \frac{1}{\Delta} \int_{-\Delta_t/2}^{\Delta_t/2} - \underbrace{\log(G_t(z + \mu_t + {}^{\Delta_t}\!/2) - G_t(z + \mu_t - {}^{\Delta_t}\!/2)}_{:= h(z)} \, dz \end{aligned}$$

C Additional experimental results

C.1 Training

We train UQDM end-to-end by directly optimizing the NELBO loss eq. (1), summing up L_t across all time steps. This can lead to high memory cost for a large T, but can be avoided by using a Monte-Carlo estimate based on a single L_t as in the diffusion literature. Our current experiments found a small T to give the best compression performance, and therefore leave the investigation of a single time-step Monte-Carlo objective to future work. Note that this would require sampling from the marginal distribution $q(\mathbf{z}_t|\mathbf{x})$, which becomes approximately Gaussian for large t (see Sec. 3.1).

C.2 Progressive coding with UQDM

Given a UQDM trained on the NELBO eq. (1), we can use it for progressive compression similarly to (Ho et al., 2020; Theis et al., 2022) (see Sec. 2).

The initial step $t = T$ involves transmitting a Gaussian $\mathbf{z}_T$. Since we do not assume access to an efficient REC scheme for the Gaussian channel, we will instead draw the same $\mathbf{z}_T \sim p(\mathbf{z}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ on both the encoder and decoder side, with the help of a shared pseudo-random seed.[2] To avoid a train/compression mismatch, we therefore always ensure $q(\mathbf{z}_T|\mathbf{x}) \approx p(\mathbf{z}_T)$ and hence $L_T \approx 0$. At any subsequent step t, instead of sampling $\mathbf{z}_{t-1} = b(t)\mathbf{z}_t + c(t)\mathbf{x} + \Delta(t)\mathbf{u}_t$ as in training, we apply UQ to compress the prior mean vector $\boldsymbol{\mu}_Q := b(t)\mathbf{z}_t + c(t)\mathbf{x}$. Specifically the sender draws $\mathbf{u}' \sim \mathcal{U}(-{}^{\mathbf{1}}\!/\mathbf{2}, {}^{\mathbf{1}}\!/\mathbf{2})$, computes $\mathbf{k}_t = \lfloor \frac{\boldsymbol{\mu}_Q}{\Delta(t)} + \mathbf{u}' \rceil$, entropy codes/transmits $\mathbf{k}_t$ under the discretized $p(\mathbf{z}_{t-1}|\mathbf{z}_t)$; the receiver recovers $\mathbf{k}_t$, draws the same $\mathbf{u}' \sim \mathcal{U}(-{}^{\mathbf{1}}\!/\mathbf{2}, {}^{\mathbf{1}}\!/\mathbf{2})$, and sets $\mathbf{z}_{t-1} = \Delta(t)(\mathbf{k}_t - \mathbf{u}')$. Finally, having transmitted $\mathbf{z}_0$, $\mathbf{x}$ is losslessly compressed using the entropy model $p(\mathbf{x}|\mathbf{z}_0)$.

We implemented the progressive codec using `tensorflow-compression` (Ballé et al.), and found the actual file size to be close to the theoretical NELBO. With our naive entropy coding implementation, it takes about 5 minutes to compress or decompress a 32 x 32 CIFAR image, with a file size overhead of $\leq 3\%$ of the theoretical NELBO. The bulk of the computation time is spent on a single CPU core, where a CDF table is built for each latent dimension for entropy coding (since we use a learned per-coordinate variance in the reverse model). This is implemented in a for-loop in the `tensorflow-compression` library and is embarrassingly parallelizable. Thus we expect the coding speed to be dramatically faster with a parallel implementation.

C.3 Swirl data

We use the same denoisng network $\hat{\mathbf{x}}_\theta$ as in the official implementation, which consists of 2 hidden layers with 512 units each.

[2] This corresponds to a trivial REC problem where a sample from $q = p$ can be transmitted using $KL(q\|p) = 0$ bits.

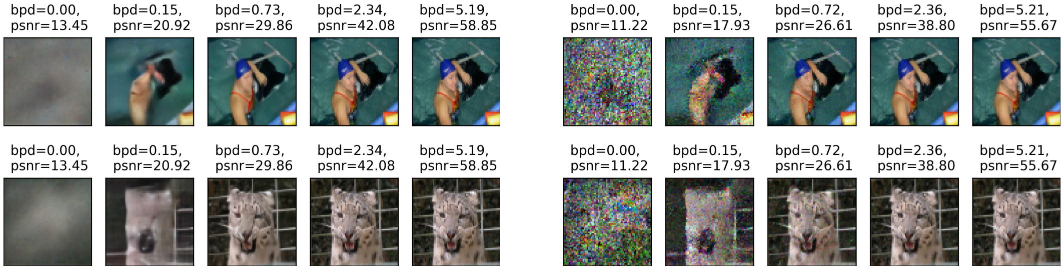

Figure 5: Example progressive reconstructions from UQDM trained with $T = 4$, obtained with denoised prediction (left) or ancestral sampling (right). The latter avoids blurriness but introduces graininess at low bit-rates, likely because the UQDM is unable to completely capture the data distribution and achieve perfect realism.

C.4 CIFAR10

We use a scaled-down version of the denoising network from the VDM paper (Kingma et al., 2021) for faster experimentation. We use a U-Net of depth 8, consisting of 8 ResNet blocks in the forward direction and 8 ResNet blocks in the reverse direction, with a single attention layer and two additional ResNet blocks in the middle. We keep the number of channels constant throughout at 128.

C.5 ImageNet 64×64

We use the same denoising network from the VDM paper (Kingma et al., 2021) – a U-Net of depth 32.