

# A Fourier Space Perspective on Diffusion Models

Fabian Falck<sup>1</sup> Teodora Pandeva<sup>1</sup> Kiarash Zahirnia<sup>1</sup> Rachel Lawrence<sup>1</sup>  
 Richard Turner<sup>1</sup> Edward Meeds<sup>1</sup> Javier Zazo<sup>1</sup> Sushrut Karmalkar<sup>1</sup>

## Abstract

Diffusion models are state-of-the-art generative models on data modalities such as images, audio, proteins and materials. These modalities share the property of exponentially decaying variance and magnitude in the Fourier domain. Under the standard Denoising Diffusion Probabilistic Models (DDPM) forward process of additive white noise, this property results in high-frequency components being corrupted faster and earlier in terms of their Signal-to-Noise Ratio (SNR) than low-frequency ones. The reverse process then generates low-frequency information before high-frequency details. In this work, we study the inductive bias of the forward process of diffusion models in Fourier space. We theoretically analyse and empirically demonstrate that the faster noising of high-frequency components in DDPM results in violations of the normality assumption in the reverse process. Our experiments show that this leads to degraded generation quality of high-frequency components. We then study an alternate forward process in Fourier space which corrupts all frequencies at the same rate, removing the typical frequency hierarchy during generation, and demonstrate marked performance improvements on datasets where high frequencies are primary, while performing on par with DDPM on standard imaging benchmarks.

## 1. Introduction

Diffusion models are the state-of-the-art generative model on data modalities such as images (Rombach et al., 2022; Baldridge et al., 2024), videos (Brooks et al., 2024; Ho et al., 2022; Blattmann et al., 2023), proteins (Watson et al., 2023; Lewis et al., 2024), and materials (Zeni et al., 2023). They excel on a wide range of tasks and applications on these

modalities, such as generating high-resolution images and videos given a text prompt (Rombach et al., 2022), sampling the distribution of conformational states of proteins (Lewis et al., 2024), or generating novel materials under property constraints (Zeni et al., 2023). *Why do diffusion models work so well on these modalities*, possibly even surpassing the performance and efficiency of autoregressive models?

In this work, we study this question via the forward (or noising) process of diffusion models in Fourier space. The forward process of standard diffusion models such as DDPM (Song et al., 2020a) corrupts data by progressively adding white Gaussian noise until all information is destroyed.<sup>1</sup> Diffusion models then learn a denoiser which reverses this forward process by starting from Gaussian noise and iteratively refining it to approximate the original data (Song et al., 2020b; Sohl-Dickstein et al., 2015).

While one might assume that the forward process destroys all information in data uniformly, this is not the case. In fact, for the modalities mentioned above, a DDPM forward process does *not* treat all frequency components equally. These modalities have in common that they exhibit a *power law* in their Fourier representation: low-frequency components have orders of magnitudes higher variances (and magnitudes) than high-frequency components (see Fig. 1 [left], (Van der Schaaf and van Hateren, 1996)). This data property has two important implications: The DDPM forward process noises high-frequency components both substantially earlier (Rissanen et al., 2022), and faster than low-frequency components (see Fig. 1 [centre]), which we will discuss in § 2 and theoretically characterise with the *Signal-to-Noise Ratio (SNR)* (see Definition 4). Intuitively speaking, the forward process in DDPM corrupts the high frequency information—the fine details such as edges—in fewer timesteps than low-frequency features, such as the larger structures and overall colour of an image (see Fig. 4 for an illustration). What is the inductive bias of this forward process on the learned reverse process?

In theory (i.e. with unlimited resources and arbitrarily accurate estimates of the scores of the noised distributions), diffusion models can learn to express any continuous distri-

<sup>1</sup>Microsoft Research. Correspondence to: Fabian Falck <fabian.falck@microsoft.com>, Sushrut Karmalkar <skar-malkar@microsoft.com>.

<sup>1</sup>White noise has equal variance across all frequency components.

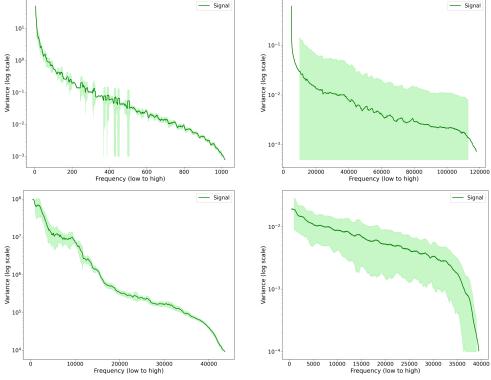


Figure 1: [Left] The Fourier power law observed in (top-left) images (Krizhevsky et al., 2009), (top-right) videos (Kay et al., 2017), (bottom-left) audio (Tzanetakis, 1999), and (bottom-right) Cryo-EM derived protein density maps (wwPDB Consortium, 2023). [Center] A DDPM forward process on these modalities noises high-frequency components substantially faster (SNR changes more per time increment), and earlier than low-frequency components. [Right] The alternate EqualSNR forward process noises all frequencies at the same rate, disrupting DDPM’s generation hierarchy. The GIFs are best viewed in Adobe Reader.

bution (see for e.g. Theorem 2 in (Chen et al., 2023)). In practice, however, diffusion models are constrained, for instance by a limited number of intermediate steps (discretisation) and the expressiveness of the neural network (score estimation), resulting in approximation errors. Since DDPM applies noise more aggressively to high-frequency components, corrupting them in fewer steps, we hypothesise that their approximation error is larger, resulting in a lower generation quality. As a result, DDPM prioritises low-frequency components within its resource constraints.

The forward process also imposes a hierarchy of the frequencies during generation: as the generative process learns to reverse the forward process (which in DDPM and on the modalities of interest noises high frequencies before low frequencies), the reverse process generates low frequencies first, and generates high frequencies conditional on low frequencies (see Fig. 4 for an illustration). Previous work has observed this phenomenon of a soft-conditioning or “approximately autoregressive” generation in DDPM diffusion models (Dieleman, 2024; Gerdes et al., 2024), drawing an important parallel with Large Language Models (LLMs). However, in spite of this ‘hidden structure’ in diffusion models, we lack understanding of the degree to which it is essential. Can diffusion models also generate all frequencies at the same rate, i.e. without any hierarchy (see Fig. 1 [right]), and how does this perform? Can we also generate high-frequency details first, then low frequencies conditional on them (see Fig. 16 in App. B)?

In this work, we study the forward process of diffusion models on data exhibiting the Fourier power law and its effect on the learned reverse process in Fourier space. We theoretically and empirically analyse the impact of noising

rate and hierarchy on the assumptions of the reverse process, and in turn on generation quality in diffusion models. We organise our *contributions* as follows:

- In § 2, we formally describe the DDPM forward process and define the SNR. We observe that this forward process results in a non-uniform rate of noising across frequencies: specifically, high-frequency components are noised faster (in the sense that the SNR decreases more per time increment) and earlier (in the sense that the SNR of high-frequency components is smaller than the SNR of low-frequency components at all timesteps).
- In § 3, we theoretically explain and empirically demonstrate that faster noising as measured by SNR of high-frequency components violates the Gaussian assumption in the backward process, leading to worse generation quality for these frequencies. We then propose a framework for forward processes based on frequency-specific SNR, including a training and sampling algorithm, a loss based on the Evidence Lower Bound (ELBO), and a mutual calibration procedure. In particular, we explore a variance-preserving forward process which noises all frequencies at the same rate (EqualSNR), disrupting the usual generation hierarchy in DDPM.
- In § 4, we demonstrate that EqualSNR performs on-par with DDPM on standard image benchmarks. Additionally, it improves the generation quality of high frequencies, and consequently the performance of diffusion models on datasets where high frequencies are dominant. In particular, we show that for DDPM, the high-frequency components of generated images can be distinguished ‘by eye’ from the high-frequency components of the original

data. In fact, a simple logistic regression classifier trained on two summary statistics of high-frequency components can discriminate DDPM-generated images from real images, while the same classifier performs poorly for Equal-SNR. This has important implications for generative modelling tasks where high-frequency details are of core interest, such as in astronomy or medical imaging, and for the generation of more realistic DeepFake data.

## 2. Background: a spectral analysis of DDPM

In this section we recall the DDPM forward process and analyse how it affects the frequencies of a signal, showing that high-frequencies are corrupted faster, and substantially earlier than low-frequency ones. This effect can be quantified in terms of the Signal-to-Noise Ratio (SNR). All proofs, further theoretical results, and an extended exposition of our theoretical framework are deferred to App. A.

**The DDPM forward process.** Given a data point  $\mathbf{x}_0 \in \mathbb{R}^d$  drawn from the data distribution whose density is given by  $p(\mathbf{x}_0)$ , the DDPM forward process (Ho et al., 2020) generates a sequence of latent variables  $\{\mathbf{x}_t\}_{t=0}^T$  which satisfy the following transitions:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \alpha_t} \mathbf{x}_{t-1}, \alpha_t \mathbf{I}), \quad (1)$$

where  $\alpha_t$  controls the amount of (white) noise added at each timestep  $t$ , which is equal for all data dimensions. The marginal distribution of  $\mathbf{x}_t$  given  $\mathbf{x}_0$  is then obtained in closed form:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}), \quad (2)$$

with  $\bar{\alpha}_t = \prod_{s=1}^t (1 - \alpha_s)$ . This can be reparameterised as

$$\mathbf{x}_t = \underbrace{\sqrt{\bar{\alpha}_t} \mathbf{x}_0}_{\text{signal}} + \underbrace{\sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}}_{\text{noise}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad (3)$$

where the first term on the right-hand side is the (scaled) signal, the second term is the (scaled) noise.

**DDPM corrupts high-frequencies faster and earlier.** In this paper, we focus on data modalities where diffusion models achieve state-of-the-art performance, such as images, video, proteins, and materials. These modalities share the property that—when viewed in their Fourier representation—their signal variance (and magnitude) decays with frequency by orders of magnitude (see Fig. 1 [left]). We call this data property the *Fourier power law* henceforth, which has for the example of images been previously noted in several works (Van der Schaaf and van Hateren, 1996; Hyvärinen et al., 2009; Rissanen et al., 2022). This property has two implications that we study in this work: fine details (high frequencies) are corrupted 1) faster, and 2) before larger structures (low frequencies) during the forward process (Kingma and Gao, 2024).

To see this, we can view the DDPM forward process equivalently under a change of basis to the Fourier space, a basis which has been of interest in the diffusion community (Dieleman, 2024; Gerdes et al., 2024) (see § 5 for a discussion). This is accomplished by applying the Fourier transform  $\mathbf{F}$  to the latent variables  $\mathbf{x}_t$  as:

$$\begin{aligned} \mathbf{y}_t &:= \mathbf{F}\mathbf{x}_t = \mathbf{F}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0) + \mathbf{F}(\sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}) \\ &= \underbrace{\sqrt{\bar{\alpha}_t} \mathbf{F}\mathbf{x}_0}_{\text{signal } \mathbf{s}} + \underbrace{\sqrt{1 - \bar{\alpha}_t} \mathbf{F}\boldsymbol{\epsilon}}_{\text{noise } \mathbf{n}} \end{aligned} \quad (4)$$

by linearity, and  $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{F}(1 - \bar{\alpha}_t) \mathbf{I}\mathbf{F}^\dagger) = \mathcal{CN}(\mathbf{0}, (1 - \bar{\alpha}_t) \mathbf{I})$ , where  $\mathbf{F}^\dagger$  is the adjoint of  $\mathbf{F}$  and noting that  $\mathbf{y}_t$  is complex-valued (see App. A for details on this calculation). Since  $\mathbf{F}$  is invertible, there is a one-to-one correspondence between a DDPM forward process in Euclidean (or pixel) space in Equation (3) and in Fourier space in Equation (4), rendering these equivalent, alternative viewpoints.

**Signal-to-Noise Ratio.** We can quantify the corruption of the signal with the *Signal-to-Noise Ratio (SNR)*.

**Definition 1** (Signal-to-Noise Ratio). Let  $s$  and  $n$  be two random variables with realisations in  $\mathbb{C}$ , and  $f(s, n) = s + n$  be a measurement process. The Signal-to-Noise Ratio (SNR) is defined as

$$\text{SNR}(f) = \frac{\text{Var}[s]}{\text{Var}[n]}, \quad (5)$$

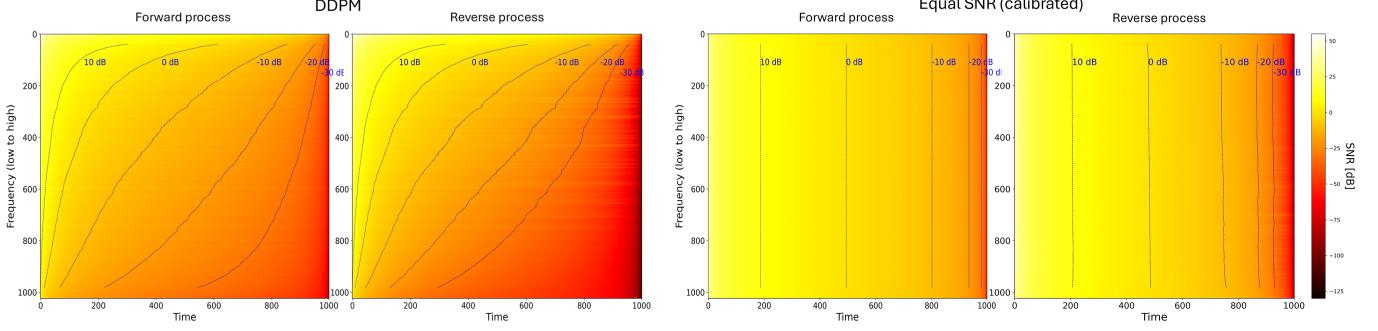
where  $\text{Var}(s) := \text{Var}(\text{Re}(s)) + \text{Var}(\text{Im}(s))$  (and likewise for  $n$ ). For  $d$ -dimensional random vectors  $\mathbf{s}, \mathbf{n}$ , we abuse notation and define  $\text{SNR}(\mathbf{f})$  entry-wise:  $\text{SNR}(\mathbf{f})_i = \text{SNR}(f_i)$ .

We can now compute the SNR of  $(\mathbf{y}_t)_i$  in Equation (4), i.e. the SNR of frequency  $i$  at timestep  $t$  as

$$s_t^{\text{DDPM}}(i) := \text{SNR}((\mathbf{y}_t)_i) = \frac{\bar{\alpha}_t \mathbf{C}_i}{1 - \bar{\alpha}_t}, \quad (6)$$

where  $\mathbf{C}_i := \text{Var}((\mathbf{y}_0)_i)$  represents the signal variance of frequency  $i$  which decays rapidly with frequency for our data modalities of interest. Equation (6) lets us formalise the two implications of the Fourier power law property: The SNR of low frequencies is orders of magnitudes higher than the SNR of high frequencies at all timesteps under a DDPM forward process (see red line in Fig. 1 [middle] for an animation) (Kingma and Gao, 2024; Dieleman, 2024). Informally speaking, a DDPM forward process does *not* ‘noise all frequencies equally’. Relative to the signal, the white noise of the DDPM forward process, which is equal for all frequencies, corrupts high-frequency information faster, i.e. in fewer timesteps. The SNR of high-frequency components changes more rapidly per time increment compared to low-frequency components.

It further imposes a hierarchy onto the forward process where high frequencies attain a low SNR much earlier than



**DDPM:** High frequencies are corrupted substantially faster (SNR changes more per time increment) than low frequencies.

**EqualSNR:** All frequencies are corrupted equally fast, achieving the same SNR of all frequencies at each timestep.

Figure 2: Comparing the SNR (dB scale) for DDPM and the alternate EqualSNR in Fourier space. In the forward process, SNR is computed as a Monte Carlo estimate of Eq. (5) on CIFAR10 (referring to App. B for details on the reverse process).

low frequencies. As the generative process of diffusion models reverses the forward process, low-frequencies are generated earlier than high-frequencies, which can be viewed as being generated conditional on the former (see Fig. 4 for an illustration). In this work, we analyse the effect of the faster and earlier noising of high frequency information in DDPM on the learned reverse process and its generative performance. In preview of our experimental results, we will demonstrate that high-frequencies generated with DDPM are of poor quality, and show that an alternate forward process (EqualSNR) alleviates this issue. This has important implications for applications where high-frequency details are the key modelling objective, such as astronomy or medical imaging, and DeepFake technology.

### 3. Fast noising disrupts high-frequency generation: alternate forward processes

In this section, we theoretically and empirically investigate the consequence of our observation in § 2 that high frequencies are noised faster than low frequencies in DDPM. Specifically, we will show that under limited resources, the Gaussian assumption of the reverse process is violated for high-frequency components. We then propose an alternate forward process in Fourier space and corresponding training and sampling algorithm which alleviates the issue. In § 4, we will experimentally investigate the effect of this violation on generation quality.

#### 3.1. Faster noising of high frequencies violates the Gaussian assumption in their reverse process.

The stochastic reverse process of DDPM, which we consider first, iteratively denoises a (Fourier-transformed) iterate  $\mathbf{y}_t$  by sampling from the learned reverse process dis-

tribution  $p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t)$  which approximates the intractable distribution  $q(\mathbf{y}_{t-1}|\mathbf{y}_t)$ . A key assumption of the DDPM reverse process is that the distribution  $q(\mathbf{y}_{t-1}|\mathbf{y}_t)$  is Gaussian, leading to the design choice that  $p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t)$  is Gaussian. In the limit of the total number of timesteps  $T$  tending to infinity,  $q(\mathbf{y}_{t-1}|\mathbf{y}_t)$  is known to converge to a Gaussian (Feller, 1954). However, in the practical setting of having a finite number of discretisation steps, this assumption only holds if the Gaussian noise added in the forward process  $q(\mathbf{y}_t|\mathbf{y}_{t-1})$  is small enough relative to the signal variance. Under a DDPM forward process on data modalities with the Fourier power law, which adds noise of equal variance to all frequencies but has orders of magnitudes smaller signal variance in the high frequencies, i.e. a faster noising of the high-frequency components (see § 2), this may lead to significant violations of the Gaussian assumption in high-frequency components.

To see this formally, we first apply Bayes rule:

$$q(\mathbf{y}_{t-1}|\mathbf{y}_t) = \frac{q(\mathbf{y}_t|\mathbf{y}_{t-1})q(\mathbf{y}_{t-1})}{q(\mathbf{y}_t)}. \quad (7)$$

We note that if the Gaussian distribution  $q(\mathbf{y}_t|\mathbf{y}_{t-1})$  has a large variance relative to  $q(\mathbf{y}_{t-1})$  and  $q(\mathbf{y}_t)$ , which is the case for high frequencies in DDPM (see § 2), fluctuations in the quantity  $\frac{q(\mathbf{y}_{t-1})}{q(\mathbf{y}_t)}$  are more apparent in the distribution  $q(\mathbf{y}_{t-1}|\mathbf{y}_t)$ . We formalise this intuition in Proposition 1. We state the informal version here, with the formal version shown in App. A.6.

**Proposition 1** ((Informal) Counterexample to normality of  $q(\mathbf{y}_{t-1}|\mathbf{y}_t)$ ). *There is a choice of sufficiently small positive constants  $\delta, \tau$  such that the following holds. Let  $D_0 = \frac{1}{2}\mathcal{N}(-1, \delta^2) + \frac{1}{2}\mathcal{N}(1, \delta^2)$ ; and  $\mathbf{x}_{t-1} \sim D_0$  and let  $\varepsilon \sim \mathcal{N}(0, 4)$ . Then for the forward update,  $\mathbf{x}_t = \mathbf{x}_{t-1} + \varepsilon$ . The corresponding reverse distribution  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  is at least a*

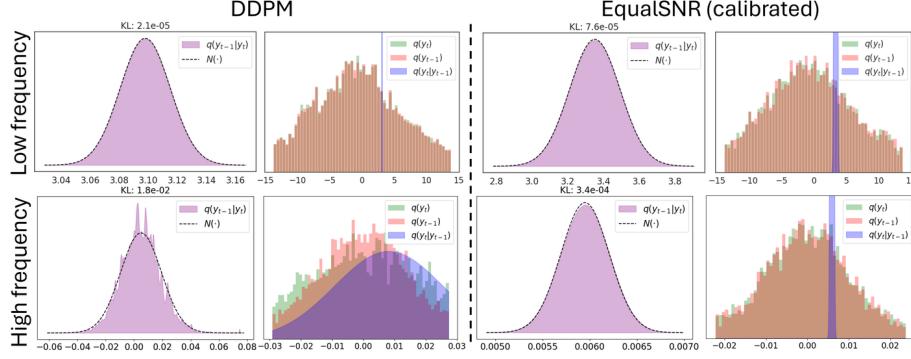


Figure 3: Fast noising of high frequencies leads to violations of normality in the DDPM reverse process. We plot Monte Carlo estimates of  $q(\mathbf{y}_t) = \mathbb{E}_{\mathbf{y}_0 \sim q(\mathbf{y}_0)} q(\mathbf{y}_t | \mathbf{y}_0)$  (and similarly for  $q(\mathbf{y}_{t-1})$ ) as histograms.

constant away in total variation distance from any Gaussian.

Proposition 1 provides a counterexample to the assumption that the reverse process can be arbitrarily well-approximated by a Gaussian. It shows that starting with a mixture of two sufficiently separated Gaussians, if we add sufficiently large variance noise, then the distribution  $q(\mathbf{y}_{t-1} | \mathbf{y}_t)$  is a constant away from any Gaussian (in fact, it also looks like a mixture of two Gaussians). These deviations lead to errors in the reverse process which can accumulate across time (Li and van der Schaar, 2023). For instance, in the case of CIFAR10 data, this happens for high frequencies since the variance of the noise added to the high frequency components is much higher relative to the variance of the data. For low frequency components on the other hand, since the variance of the data is large, this phenomenon does not occur.

In Fig. 3 [left] we verify our theoretical analysis empirically for DDPM on CIFAR10. We estimate  $q(\mathbf{y}_{t-1} | \mathbf{y}_t)$  (purple) via Bayes rule through Equation (7) for a [top] low frequency and [bottom] high frequency, focusing on the real component, one image channel and time step  $t = 1$  (see App. B Figs. 9 to 14 for further plots). We observe that in DDPM, relative to the variance of the noisy signal  $q(\mathbf{y}_t)$  and  $q(\mathbf{y}_{t-1})$ , the Gaussian  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$  has low variance for the low frequency components (as can be seen by the sharp peak), and a significantly larger variance for high frequencies. This results in violations of the Gaussian assumption for the posterior  $q(\mathbf{y}_{t-1} | \mathbf{y})$  of high frequencies, while low frequencies approximately follow a Gaussian distribution. We can quantify this in terms of the KL-divergence, observing that it is orders of magnitudes higher for high-frequencies ( $1.8 \times 10^{-2}$ ) than for low-frequencies ( $3.1 \times 10^{-4}$ ). Note that while the violations are small, resulting errors may accumulate across timesteps in the reverse process, adding up to a significant distortion in the final sample (Li and van der Schaar, 2023).

### 3.2. Alternate forward processes

In the following and throughout our experiments in § 4 we study an alternate forward process which we call *EqualSNR* where the SNR of all frequencies is the same at every timestep. In App. A we also define *FlippedSNR*, a forward process where the SNR of the  $i^{\text{th}}$  frequency is the same as the SNR of the  $(d - i)^{\text{th}}$ , and which hence inverts the frequency hierarchy of DDPM during generation. EqualSNR investigates two questions: first, it noises all frequencies at the same rate. Recalling our analysis of the aggressiveness of noising in § 3.1, in ESNR we enforce the ratio of the variance of  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$  and the variance of  $\frac{q(\mathbf{y}_t)}{q(\mathbf{y}_{t-1})}$  to be *equal* for all frequencies at each timestep. We can here compute the Monte Carlo estimate of  $q(\mathbf{y}_t)$  and  $q(\mathbf{y}_{t-1})$  via the push-forward in Equation (8). Fluctuations in  $\frac{q(\mathbf{y}_t)}{q(\mathbf{y}_{t-1})}$  hence affect all frequencies equally, and the Gaussian assumption of the reverse process in high-frequency components is no longer violated with similar distribution distances (see Fig. 3 [right]; KL for low frequencies:  $1.0 \times 10^{-4}$ , compared to high frequencies:  $1.3 \times 10^{-4}$ ), overcoming this issue of DDPM. Second, both EqualSNR and FlippedSNR address the question whether low-to-high frequency generation is essential in diffusion models: EqualSNR, which we will focus on in § 4, generates samples without any hierarchy among the frequencies, while FlippedSNR inverts the hierarchy of DDPM, generating frequencies from high to low. We now formalise this.

**Definition 2** (EqualSNR process in Fourier space). Let  $C_i = \text{Var}[(\mathbf{y}_0)_i]$  represent the coordinate-wise variance in Fourier space, and let  $\epsilon \sim \mathcal{CN}(0, \Sigma)$ . Suppose the forward process in Fourier space is given by  $\mathbf{y}_t = \sqrt{\alpha_t} \mathbf{y}_0 + \sqrt{1 - \alpha_t} \epsilon_{\Sigma}$ , with the SNR at timestep  $t$  and frequency  $i$  defined as  $s_t(i) = \frac{\sqrt{\alpha_t} C_i}{(1 - \sqrt{\alpha_t}) \Sigma_{ii}}$ . This implies:

1. **DDPM:** The forward process for DDPM has SNR  $s_t^{\text{DDPM}}(i) = \frac{\sqrt{\alpha_t} C_i}{(1 - \sqrt{\alpha_t})}$ .
2. **Equal SNR:** The forward process has equal SNR across all coordinates if and only if  $\Sigma_{ii} = c C_i$ , where  $c$  is a uni-

versal constant. The process is coordinate-wise ‘variance-preserving’ (as in (Song et al., 2020b)) when  $c = 1$ .<sup>2</sup>

Note that we define the forward processes through their frequency-specific SNR  $s_t(i)$  at time  $t$ , not the mixing coefficients (e.g.  $\bar{\alpha}_t$ ), illustrated in Fig. 2 (together with a proxy-SNR for the reverse process, see App. B for details). This is advantageous as it allows us to straight-forwardly apply such a forward process to high-resolution data without modification (in contrast to related work (Hoogeboom et al., 2023) where, in order to preserve the per-timestep SNR, they have to explicitly account for an adjustment to the mixing weights when generating high-resolution images). In the following, we summarise key framework components to train diffusion model in Fourier space with these alternate forward process, referring to App. A for details.

#### Algorithm 1 Training algorithm (Fourier space noise)

**Input:** Data samples  $S := \{\mathbf{x}_0^{(i)}\}_{i=1}^N$ , number of diffusion steps  $T$ , noise schedule  $\{\bar{\alpha}_t\}_{t=1}^T$ , neural network  $f_\theta$ , number of training iterations  $M$ .

$C := \text{Diag}(\text{Cov}(\mathbf{y}))$  **for**  $M$  training iterations **do**

- Sample  $\mathbf{x}_0 \sim S, \mathbf{y}_0 = \mathbf{F}\mathbf{x}_0, t \sim \text{Uniform}(1, \dots, T)$ ,
- for every  $j \in [d/2]$ ,  $(\epsilon_C)_j = (\epsilon_C)_{d-j}$  and  $(\epsilon_C)_j \sim \frac{C_j^{1/2}}{\sqrt{2}}(\epsilon + ie')$  where  $\epsilon, \epsilon' \sim \mathcal{N}(0, 1)$ .
- Fourier forward process:  $\mathbf{y}_t = \sqrt{\bar{\alpha}_t}\mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_C$ .
- Predict sample:  $\hat{\mathbf{y}}_0 = (\mathbf{F} \circ f_\theta)(\mathbf{F}^{-1}(\mathbf{y}_t), t)$ .
- Compute loss  $\mathcal{L}_t = \|C^{-1/2}(\mathbf{y}_0 - \hat{\mathbf{y}}_0)\|^2$ .
- Update  $\theta$  using gradient descent on  $\mathcal{L}_t$ .

**Training algorithm and ELBO.** Algorithm 1 allows to train diffusion models with the alternate forward processes in Fourier space (in the variant predicting the clean sample). A key difference to standard DDPM training is that the noise and the difference in the loss  $\mathcal{L}_t$  are scaled by the signal variances  $C^{1/2}$  in Fourier space. Proposition 2 in App. A proves that  $\mathcal{L}_t$  is an ELBO. To ensure a fair comparison with standard diffusion models, we train the neural network  $f_\theta$  (typically a U-Net (Ronneberger et al., 2015) in Euclidean/pixel space to maintain its inductive bias.

**Sampling algorithm.** Algorithm 2 adapts the Denoising Diffusion Implicit Models (DDIM) sampling algorithm (Song et al., 2020a) to Fourier space. We note that while our analysis of the normality assumption in § 3.1 was for the stochastic reverse process of DDPM, even though DDIM is deterministic, a similar property holds here, which results in poor generation quality for faster noised frequencies.

<sup>2</sup>If we require that  $\Sigma = c\text{Cov}(\mathbf{y}_0)$ , the equal SNR property holds across all bases (see App. A.5).

---

**Algorithm 2** Sampling algorithm (DDIM in Fourier space)

**Input:** Input noise  $\mathbf{y}_T := \epsilon_C$  where for every  $j \in [d/2]$ ,

$$(\epsilon_C)_j = \overline{(\epsilon_C)_{d-j}}$$

and  $(\epsilon_C)_j \sim \frac{C_j^{1/2}}{\sqrt{2}}(\epsilon + ie')$  where  $\epsilon, \epsilon' \sim \mathcal{N}(0, 1)$ , neural network  $f_\theta$ , noise schedule  $\{\bar{\alpha}_t\}_{t=1}^T$ , sampling steps  $T$ .

**for**  $t$  from  $T$  to 1 **do**

- Predict sample:  $\hat{\mathbf{y}}_0^{(t)} = (\mathbf{F} \circ f_\theta)(\mathbf{F}^{-1}(\mathbf{y}_t), t)$ .
- $\mathbf{y}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{y}}_0^{(t)} + \frac{\sqrt{1 - \bar{\alpha}_{t-1}}}{\sqrt{1 - \bar{\alpha}_t}}(\mathbf{y}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{y}}_0^{(t)})$ .

**return**  $\mathbf{F}^{-1}\mathbf{y}_0^{(1)}$ .

---

**Calibration.** To compare performance across different forward processes, we calibrate them by ensuring that the average SNR across frequencies at any given timestep is the same. This means that the average amount of information destroyed across frequencies at timestep  $t$  is the same across these processes. We refer to App. A.3 for how to realise the calibration with an appropriate choice of the mixing coefficients  $\bar{\alpha}_t$ .

## 4. Experiments

In this section, we experimentally study the effect of the rate of noising in DDPM and EqualSNR on generation quality in Fourier space.

**Experiment setting.** We use three natural imaging datasets, a modality which exhibits the power law property in their Fourier representation, of different resolution: CIFAR10 ( $32 \times 32$ ), CelebA ( $64 \times 64$ ), and LSUN Church ( $128 \times 128$ ). Furthermore, we use synthetic, high-frequency datasets which are described in the corresponding sections. We use a standard U-Net architecture throughout our experiments (Ronneberger et al., 2015). App. B provides additional experimental details and results.

**The reverse process learns to mirror its forward.** To provide intuition, we begin by analysing the effect a forward (noising) process has on the reverse (generation) process in Fourier space. In Fig. 4 we illustrate the forward and backward process for a single image for DDPM and EqualSNR schedule in pixel and Fourier space between  $t = 0$  and  $t = T/2$  (see App. B Figs. 22 to 27 for the full time interval, further examples and other datasets). In DDPM, the low- and high-pass filtered images (rows 3 and 4; details on the calculation in App. B) show that the high frequency information is lost early on during the forward process. This enforces the reverse process to generate low frequencies first, and complete high frequency information only at the very end of the reverse process. In contrast, in EqualSNR the low- and high-pass images are corrupted at the same rate, which enforces the reverse process to learn to generate all frequen-

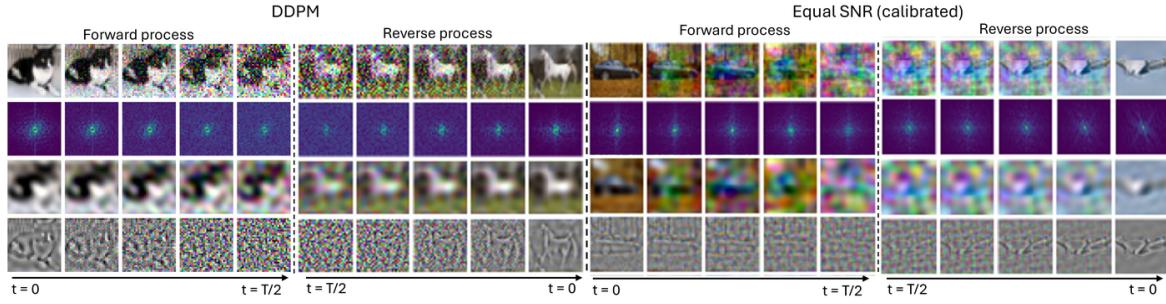


Figure 4: *The forward process controls when frequencies are generated in the reverse process.* We visualise the forward and backward process of [left] DDPM and [right] EqualSNR in pixel space (rows 1,3,4) and Fourier space (magnitudes; row 2). Rows 3 and 4 are low- and high-pass filters of the (noisy) image. DDPM noises high-frequency components first and hence generates them last, while EqualSNR noises and generates all frequencies at the same time.

Table 1: EqualSNR performs on par with DDPM. We measure performance using Clean-FID ( $\downarrow$ ) (Parmar et al., 2022).

T	CIFAR10 (32 × 32)				CelebA (64 × 64)				LSUN Church (128 × 128)			
	50	100	200	1000	50	100	200	1000	50	100	200	1000
DDPM schedule	18.63	18.01	17.68	17.7	10.10	8.72	8.30	8.62	29.36	25.36	24.03	23.22
EqualSNR (calibrated) schedule	16.00	15.91	15.76	15.73	9.45	8.79	8.62	8.56	19.42	19.75	19.90	19.80
DDPM (calibrated) schedule	16.64	14.69	14.07	13.85	12.65	7.88	6.54	6.59	40.31	26.4	22.05	20.09
EqualSNR schedule	15.44	14.56	14.13	13.63	12.99	11.64	10.96	10.37	27.13	25.68	24.81	24.05

cies simultaneously. Fig. 2 (reverse process) and Fig. 17 further illustrates this mirroring of the forward and reverse process computing the trajectory of variances of each frequency component at different timesteps (see App. B for details).

**Faster noising degrades high-frequency generation quality in DDPM, EqualSNR overcomes this.** In Sections 2 and 3 we theoretically and empirically identified that high-frequency components are noised faster than low-frequency components, and showed that this leads to violations of the Gaussian assumption in their reverse process. We here investigate the effect of such (accumulating) violations on generation performance. We build on the Fourier-based Deepfake detection framework proposed by (Dzanic et al., 2020). Intuitively, this framework trains a classifier on a subset of the frequencies to discriminate generated and real data. We use classification performance as a (frequency-specific) proxy for generation quality. All experiments were performed with  $T = 1000$  sampling steps.

Fig. 5 compares the spectral profile of magnitudes for CIFAR10 training data, and samples generated with a [top] DDPM and [bottom] EqualSNR diffusion model, respectively. We plot the mean magnitude and standard deviation for the [left] lowest and [right] highest frequencies. Our analysis reveals a systematic failure of DDPM to accurately capture the high-frequency statistics of CIFAR10: while the low-frequency magnitudes of generated samples perfectly match those of the data distribution, high-frequency magnitudes can be qualitatively distinguished, even ‘by eye’. In contrast, EqualSNR demonstrates superior generation qual-

ity for the high frequencies with matching magnitudes for both low and high frequencies.

Table 2: *High-frequency components generated with DDPM are easy to discriminate from real data, those generated with EqualSNR are not.* We report classifier accuracy averaged over 100 runs and corresponding % (out of 100) of true positives (TP) at significance levels 0.05 and 0.01.

	Freq. band	Mean Acc.	%TP at 0.05	%TP at 0.01
DDPM	5%	0.624	99%	99%
	15%	0.643	100%	99%
	25%	0.654	100%	100%
EqualSNR	5%	0.516	13%	5%
	15%	0.521	16%	1%
	25%	0.518	10%	5%

In Table 2 we measure this mismatch of the approximate and data distribution for high-frequency components quantitatively and on a per-sample basis. We isolate high frequency bands (top 5%, 15%, 25%) and fit a regression model with two parameters to the spectral magnitudes of each image following (Dzanic et al., 2020). We then train a logistic regression classifier receiving these two regression parameters per image as input which discriminates real and generated images, repeated for 100 times. We intentionally choose a simple regression model and classifier for the benefit of the interpretability of the analysis, noting that more complicated classifiers which in particular take into account all frequencies may achieve substantially better performance. To

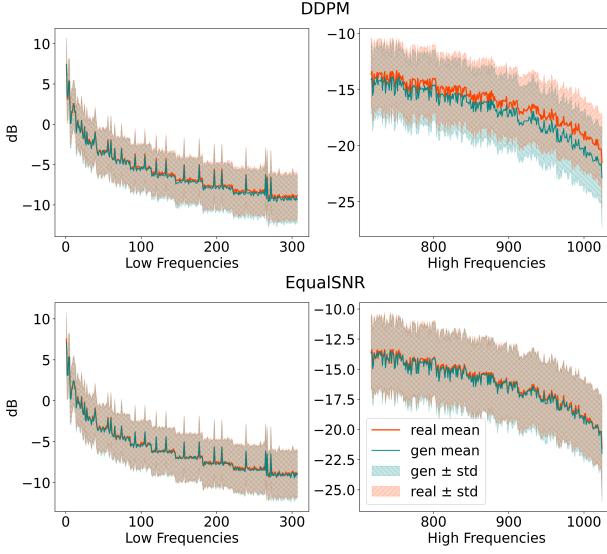


Figure 5: *EqualSNR* is superior to *DDPM* in high-frequency generation quality. We plot the spectral magnitude profile (in decibels) for low and high frequencies, comparing data generated with [top] *DDPM* and [bottom] *EqualSNR* generated (blue) and real (red) data.

assess the validity of our classifier performance results and determine whether they are statistically significant (rather than occurring by random chance), we perform the following test: we divide the data into 100 independent partitions and perform 100 hypothesis tests on the classifier’s performance. For each test, our measure of success is the fraction of correctly rejected hypotheses (true positives) (see App. B for details on the statistical test).

For standard *DDPM*, we correctly reject the null hypothesis (distributions of real and generated images are the same) almost 100% of the time for all high-frequency bands. In contrast, for *EqualSNR*, we reject it at a much lower rate (10-13%) at a significance level of 0.05, and 1-5% at a significance level of 0.01. Note when repeating the same experiment with low-frequency bands, *DDPM* and *EqualSNR* perform similarly (see Table 3 in App. B). This underlines the advantageous high-frequency generation performance of *EqualSNR* compared to *DDPM*. We will exploit this property on data where high-frequency information is the key modelling objective next.

**When high-frequency information matters: a synthetic study.** We simulate real-world experiments where high-frequency, fine-grained information is of primary interest, such as astronomy, satellite and medical imaging, with a synthetic *Dots* experiment. We generate  $32 \times 32$  images with between 46 and 50 white pixels randomly placed on a black background (see Fig. 6 [right] for generated exam-

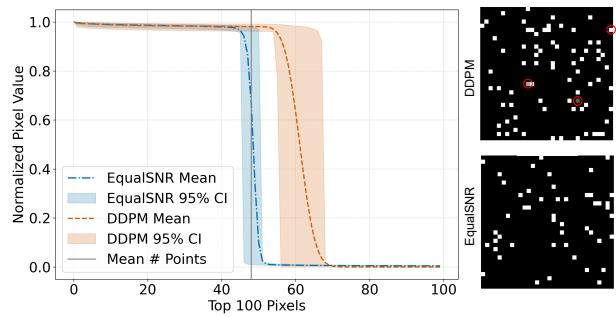


Figure 6: *EqualSNR* outperforms *DDPM* on data where high-frequency information is dominant. Pixel intensity distribution (sorted descendingly) of 1000 generated samples for *DDPM* and *EqualSNR*, and two examples.

ples). In Fig. 6 [left] we examine the average pixel intensity and the corresponding 95% confidence intervals over 1000 images for both *DDPM* and *EqualSNR*. *EqualSNR* significantly outperforms *DDPM* on this task: *EqualSNR* generates the correct amount of dots and has a steeper slope than *DDPM*. *DDPM*, however, generates more points on average than present in the real data (gray line). In conclusion, this suggests that an *EqualSNR* diffusion model is better at generating sparse, high frequency features, achieving a closer match to the original data distribution.

**EqualSNR performs on par with DDPM on imaging benchmarks.** Given the advantageous performance of *EqualSNR* on high frequencies, a natural question to ask is if—in light of the No Free Lunch Theorem (Wolpert and Macready, 1997)—this harms generation quality on other data. We answer this question by training unconditional *DDPM* and *EqualSNR* diffusion models, where two runs are calibrated to each other, on standard imaging benchmarks, reporting obtained Clean-FID (Heusel et al., 2017; Parmar et al., 2022) values in Table 1. Overall, we observe that *EqualSNR* performs on par with *DDPM*. We note that FID, which correlates with human perception of fidelity, may not capture the demonstrated issues in high-frequency accuracy observed for *DDPM*, contributing to the discussion of the appropriateness of the metric (Jayasumana et al., 2024). On LSUN Church, the highest resolution dataset which contains more high frequencies, *EqualSNR* (calibrated) outperforms all *DDPM* variants. Furthermore, we observe that *EqualSNR* requires less timesteps  $T$  than *DDPM* to saturate performance. This underlines that *EqualSNR* is a competitive forward process for diffusion models with beneficial performance on high-frequency data.

## 5. Related work

**Diffusion in Fourier space and other function spaces.** Several works have studied diffusion models in Fourier space. Most notably, concurrent work by [Gerdes et al. \(2024\)](#) proposes a framework for diffusion processes on general function spaces, covering both Fourier and wavelet spaces. Furthermore, they show that the forward process controls the degree of overlap of the ‘active time’ of adjacent frequencies. They draw an important *parallel between diffusion models and autoregressive models*: at one extreme, autoregressive models such as Large Language Models (LLMs) would generate each frequency component sequentially conditional on all previous ones. The other extreme generates data without any hierarchy. The latter is implemented by Equal SNR which they propose but to the best of our knowledge do not experiment with (see Fig. 4). [Jiralerspong et al. \(2025\)](#) concurrently analyse shaping the inductive bias of the forward process in Fourier space. They demonstrate how the forward process should be chosen in relation to the approximated data distribution. [Crabbé et al. \(2024\)](#) show how to map the continuous-time formulation of diffusion models between an origin space and its Fourier transform, and apply diffusion models in the Fourier space to time series data, demonstrating superior performance. [Phillips et al. \(2022\)](#) likewise study diffusion in the spectral domain, particularly on multi-modal data.

[Guth et al. \(2022\)](#) and [Phung et al. \(2023\)](#) propose a diffusion model acting on multiple wavelet subspaces simultaneously, the former showing that time complexity increases linearly with image size. [Jiang et al. \(2023\)](#) and [Huang et al. \(2024\)](#) exploit wavelet representations of images for the inverse problems of low-light enhancement and image restoration.

**Diffusion schedules and the importance of the rate of noising.** Previous work also studied the importance of the noising schedule, also in light of how fast and early frequency information is corrupted. [Kingma et al. \(2021\)](#) learn the parameters of a variance-preserving noising schedule during training, observing that it spends more time in the high-SNR regime compared to other schedules, i.e. on generating high frequencies. [Williams et al. \(2024\)](#) provide an adaptive algorithm for finding a noising schedule which is optimal given a cost measuring the work required to transport samples along the diffusion path, and similarly observe on image experiments that this schedule spends more time on high-frequency details compared to a cosine schedule. [Ziashahabi et al. \(2024\)](#) adapt the noising schedule in Fourier space to focus the limited resources on certain frequencies ranges, observing significant speedups at inference time while maintaining image quality. [Voleti et al. \(2022\)](#), similar to our work, propose a non-isotropic covariance structure in the forward diffusion process. [Yang et al. \(2023\)](#) also

found that high-frequency generation is harmed by DDPM diffusion models, but provide different explanations for this insight.

### The low-to-high frequency hierarchy in diffusion models.

Several works either explicitly or implicitly state the hypothesis that generating data from low to high frequency, and the corresponding inductive bias on the forward process which enforces this, is crucial for the success of diffusion models. In the seminal DDPM paper, [Ho et al. \(2020\)](#) observe that when generating samples conditional on noisy iterate, the sample shares almost all features (except for high-frequency details) with the iterate when the iterate is less noisy (i.e. is from late in the reverse process), but only shares the large-scale (i.e. low frequency) features when the iterate is early in the reverse process, and refer to this as “conceptual compression”. [Rissanen et al. \(2022\)](#) observe the “implicit spectral inductive bias” of generating low frequencies before high frequencies and proposes a model which leverages this structure explicit. In this context, [Dieleman \(2024\)](#) characterises diffusion models as approximate autoregressive models in Fourier space. [Hoogeboom et al. \(2023\)](#) hypothesise that high-frequency details can be generated in few diffusion steps when conditioning on already generated low-frequency features, which they exploit to adjust the noising schedule. Our EqualSNR schedule puts this to the extreme in the sense that it spends the same time on all frequencies. Importantly, to the best of our knowledge, no previous work showed to what degree the hierarchical structure in Fourier space of generating low frequencies before high frequencies is essential for diffusion models. Several applications such as text-to-image and video generation of diffusion models also benefit from the low-to-high frequency generation of diffusion models ([Yi et al., 2024](#); [Ruhe et al., 2024](#)).

Conditioning from low-to-high frequency information is further exploited in U-Nets ([Ronneberger et al., 2015](#)), a go-to architecture for diffusion models. [Williams et al. \(2023\)](#) show that under DDPM diffusion noise, high-frequency components in a Haar wavelet basis have a substantially lower SNR than low frequencies. This is exploited by U-Nets as average pooling, a common downsampling operation, is conjugate with projection to a lower-resolution Haar wavelet subspace, an alternate, frequency-sensitive basis. This means that the noisy high-frequency components, where most information is corrupted, are nullified along the levels of the encoder, rendering U-Nets efficient denoisers in diffusion models ([Falck et al., 2022](#)).

Unrelated to the above works, [Rahaman et al. \(2019\)](#); [Wang et al. \(2024\)](#) identify a spectral bias of neural networks towards low frequencies (under strict network constraints), offering a potential alternative explanation for poor high-frequency generation.

## 6. Conclusion

In this work, we analysed the forward process of diffusion models in Fourier space, specifically the rate of noising and the induced hierarchy of frequencies, and its inductive bias on the learned reverse process. We theoretically analysed and experimentally demonstrated that faster noising of high-frequency components as commonly done in DDPM degrades their generation quality, and showed that a diffusion model with a hierarchy-free forward process (EqualSNR) alleviates this issue, performing on par with the standard (hierarchical) DDPM forward process in terms of FID on standard imaging benchmarks. We refer to App. B for a discussion of our limitations. Future work should further investigate SNR-governed, alternate forward processes and tune them to the specific (Fourier) properties of the modality and task at hand.

## Acknowledgements

We acknowledge the blog post by ‘Diffusion is spectral autoregression’ (Dieleman, 2024) by Sander Dieleman which strongly motivated this work and inspired the design of Fig. 1. We thank Sam Bond-Taylor, Heiner Kremer, Andrew Y. K. Foong, Markus Heinonen and the colleagues at Microsoft Research Cambridge for useful discussions.

## Impact Statement

While our work is of methodological nature, it has potential, yet important safety concerns for the ability to generate highly-realistic DeepFake data (Westerlund, 2019), in particular images and videos, which have the Fourier power law property. As we demonstrated in § 4, a diffusion model with the alternate EqualSNR forward process showed marked improvements in the generation quality of high-frequency components. As a result, the generated samples were significantly harder to discriminate from real data by a classifier than samples from DDPM. More generally, our SNR-governed framework for designing forward processes and the corresponding analysis may be abused to design ‘adversarial’ forward processes which either make the generated samples even more realistic, or allow the modification of real data to appear generated from a diffusion model. Future work should further investigate these safety concerns.

## References

- Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, et al. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024.
- Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis.

Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL <https://openai.com/research/video-generation-models-as-world-simulators>.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL [https://openreview.net/forum?id=zyLVMgsZOU\\_](https://openreview.net/forum?id=zyLVMgsZOU_).

Jonathan Crabbé, Nicolas Huynh, Jan Stanczuk, and Mihaela van der Schaar. Time series diffusion in the frequency domain. *arXiv preprint arXiv:2402.05933*, 2024.

Sander Dieleman. Diffusion is spectral autoregression, 2024. URL <https://sander.ai/2024/09/02/spectral-autoregression.html>.

Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier Spectrum Discrepancies in Deep Network Generated Image. *Advances in neural information processing systems*, 33:3022–3032, 2020.

Fabian Falck, Christopher Williams, Dominic Danks, George Deligiannidis, Christopher Yau, Chris C Holmes, Arnaud Doucet, and Matthew Willetts. A multi-resolution framework for U-Nets with applications to hierarchical VAEs. *Advances in Neural Information Processing Systems*, 35:15529–15544, 2022.

William Feller. Diffusion processes in one dimension. *Transactions of the American Mathematical Society*, 77:1–31, 1954. URL <https://api.semanticscholar.org/CorpusID:32048987>.

Mathis Gerdes, Max Welling, and Miranda CN Cheng. Gud: Generation with unified diffusion. *arXiv preprint arXiv:2410.02667*, 2024.

Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet score-based generative modeling. *Advances in neural information processing systems*, 35:478–491, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.
- Yi Huang, Jiancheng Huang, Jianzhuang Liu, Mingfu Yan, Yu Dong, Jiaxi Lyu, Chaoqi Chen, and Shifeng Chen. Wavedm: Wavelet-based diffusion models for image restoration. *IEEE Transactions on Multimedia*, 2024.
- Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media, 2009.
- Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315, 2024.
- Hai Jiang, Ao Luo, Haoqiang Fan, Songchen Han, and Shuaicheng Liu. Low-light image enhancement with wavelet-based diffusion models. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.
- Thomas Jiralerspong, Berton Earnshaw, Jason Hartford, Yoshua Bengio, and Luca Scimeca. Shaping inductive bias in diffusion models through frequency-based noise control. *arXiv preprint arXiv:2502.10236*, 2025.
- Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.
- Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Sarah Lewis, Tim Hempel, José Jiménez Luna, Michael Gastegger, Yu Xie, Andrew YK Foong, Victor García Satorras, Osama Abdin, Bastiaan S Veeling, Iryna Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *bioRxiv*, pages 2024–12, 2024.
- Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.
- Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.
- Angus Phillips, Thomas Seror, Michael Hutchinson, Valentin De Bortoli, Arnaud Doucet, and Emile Mathieu. Spectral diffusion processes. *arXiv preprint arXiv:2209.14125*, 2022.
- Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10199–10208, 2023.
- Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using

- nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.
- 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*, 2020b.
- George Tzanetakis. Gtzan music/speech collection, 1999. URL <http://marsyas.info/index.html>.
- van A Van der Schaaf and JH van van Hateren. Modelling the power spectra of natural images: statistics and information. *Vision research*, 36(17):2759–2770, 1996.
- Vikram Voleti, Christopher Pal, and Adam Oberman. Score-based denoising diffusion with non-isotropic gaussian noise models. *arXiv preprint arXiv:2210.12254*, 2022.
- Jing Wang, Songtao Wu, Zhiqiang Yuan, Qiang Tong, and Kuanhong Xu. Frequency compensated diffusion model for real-scene dehazing. *Neural Networks*, 175:106281, 2024.
- Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.
- Mika Westerlund. The emergence of deepfake technology: A review. *Technology innovation management review*, 9(11), 2019.
- Christopher Williams, Fabian Falck, George Deligiannidis, Chris C Holmes, Arnaud Doucet, and Saifuddin Syed. A unified framework for u-net design and analysis. *Advances in Neural Information Processing Systems*, 36:27745–27782, 2023.
- Christopher Williams, Andrew Campbell, Arnaud Doucet, and Saifuddin Syed. Score-optimal diffusion schedules. *arXiv preprint arXiv:2412.07877*, 2024.
- David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.
- The wwPDB Consortium. Emdb—the electron microscopy data bank. *Nucleic Acids Research*, 52(D1):D456–D465, 11 2023. ISSN 0305-1048. doi: 10.1093/nar/gkad1019. URL <https://doi.org/10.1093/nar/gkad1019>.
- Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 22552–22562, 2023.
- Mingyang Yi, Aoxue Li, Yi Xin, and Zhenguo Li. Towards understanding the working mechanism of text-to-image diffusion model. *arXiv preprint arXiv:2405.15330*, 2024.
- Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*, 2023.
- Amir Ziashahabi, Baturalp Buyukates, Artan Sheshmani, Yi-Zhuang You, and Salman Avestimehr. Frequency domain diffusion model with scale-dependent noise schedule. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 19–24. IEEE, 2024.

## A. Extended theoretical framework and proofs

In this appendix, we will describe theoretical results in more detail. We begin by defining our notation and setting out basic concepts.

### A.1. Preliminaries

#### A.1.1. NOTATION

Lowercase Greek letters, such as  $\alpha, \beta, \gamma, \varepsilon$  denote scalars. Lowercase bold letters, such as  $\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\epsilon}, \dots$  denote vectors. If  $\mathbf{x} \in \mathbb{C}^d$  then  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ . Uppercase bold letters, such as  $\mathbf{I}, \mathbf{C}, \mathbf{F}, \dots$  denote matrices.  $[T] := \{1, \dots, T\}$ .  $\|\cdot\|_2$  denotes the  $\ell_2$ -norm, i.e. for  $\mathbf{x} \in \mathbb{R}^d$ ,  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^d x_i^2}$ . For two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{C}^d$ , we define the inner product between  $\mathbf{x}$  and  $\mathbf{y}$  to be  $\mathbf{x} \cdot \mathbf{y} := \sum_{i=1}^d \mathbf{x}_i \mathbf{y}_i$ .  $\mathcal{N}(\mu, \Sigma)$  denotes the multivariate normal distribution with mean  $\mu$  and covariance  $\Sigma$ . Sometimes we may write  $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ , this just means that the random variable  $\mathbf{x}$  is drawn from  $\mathcal{N}(\mu, \Sigma)$ .  $\mathcal{CN}(\mu, \Sigma)$  is a complex Gaussian. If  $p(\mathbf{x}_1, \dots, \mathbf{x}_T)$  denotes the distribution (or the pdf of the distribution depending on context) over  $T$  random variables and  $S \subset [T]$ ,  $p((\mathbf{x}_i)_{i \in S})$  denotes the distribution marginalized over  $S$ . For any vector  $\mathbf{v} \in \mathbb{C}^d$ , let  $i := \sqrt{-1}$  and  $\text{Re}(\mathbf{v}) \in \mathbb{R}^d$  denote the real part of  $\mathbf{v}$ , and let  $\text{Im}(\mathbf{v}) \in \mathbb{R}^d$  denote the imaginary part.

In the following subsections, we discuss the Fourier transform and calculate the covariance of the transformed data.

#### A.1.2. FAST FOURIER TRANSFORM

We define the discrete Fourier transform operator  $\mathbf{F}_1$  acting on a vector  $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^\top \in \mathbb{C}^N$  as

$$(\mathbf{F}_1 \mathbf{x})_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{nk}{N}}, \quad k = 0, \dots, N-1.$$

Note that  $\mathbf{F}_1 \mathbf{x}$  is generally *complex-valued*.

Images (which can be viewed as 2D grids of pixel values) often use a two-dimensional version of the FFT, for a vectorized image, this mapping corresponds to the tensor power of  $\mathbf{F}$ ,  $\mathbf{F} := \mathbf{F}_1^{\otimes 2}$ . Conceptually, the two-dimensional FFT applies the 1D transform (as defined above) first along each row and then along each column (or vice versa), yielding a complex-valued frequency representation of the image.

#### A.1.3. COVARIANCE OF FOURIER TRANSFORMED DIFFUSION NOISE

Let  $\mathbf{x}_t := \mathbf{F} \mathbf{y}_t$  where  $\mathbf{y}$  is distributed according to Eq. (2). Then,

$$\begin{aligned} \mathbb{E}[\mathbf{x}_t] &= \sqrt{\bar{\alpha}_t} \mathbf{F} \mathbf{y}_0 \\ \text{Cov}[\mathbf{x}_t] &= \mathbb{E}[(\mathbf{x}_t - \mathbb{E}[\mathbf{x}_t])(\mathbf{x}_t - \mathbb{E}[\mathbf{x}_t])^\dagger] \\ &= \mathbb{E}[(\mathbf{F} \mathbf{y} - \sqrt{\bar{\alpha}_t} \mathbf{F} \mathbf{y}_0)(\mathbf{F} \mathbf{y} - \sqrt{\bar{\alpha}_t} \mathbf{F} \mathbf{y}_0)^\dagger] \\ &= \mathbb{E}[(\mathbf{F}(\mathbf{y}_t - \sqrt{\bar{\alpha}_t} \mathbf{y}_0))(\mathbf{F}(\mathbf{y}_t - \sqrt{\bar{\alpha}_t} \mathbf{y}_0))^\dagger] \\ &= \mathbf{F} \mathbb{E}[\mathbf{y}_t - \sqrt{\bar{\alpha}_t} \mathbf{y}_0](\mathbf{y}_t - \sqrt{\bar{\alpha}_t} \mathbf{y}_0)^T \mathbf{F}^\dagger \\ &= \mathbf{F} \text{Cov}(\mathbf{y}_t) \mathbf{F}^\dagger \\ &= \mathbf{F}(1 - \bar{\alpha}_t) \mathbf{I} \mathbf{F}^\dagger \\ &= (1 - \bar{\alpha}_t) \mathbf{I} \end{aligned}$$

## A.2. Forward Processes and their Derivation

**Definition 3** (Modified forward process in Fourier space). Let the random variable  $\mathbf{x}_0$  denote the vectorized signal in the data space (e.g. pixel space), and  $\mathbf{y}_0 \in \mathbb{C}^d$  denote its Fourier transform. Let  $\mathbf{C}_i := \text{Var}[(\mathbf{y}_0)_i]$  denote the coordinate-wise variance in Fourier space, which we illustrate for three imaging datasets in App. B Fig. 15, and let  $\boldsymbol{\epsilon} \sim \mathcal{CN}(0, \Sigma)$ . Then the forward process in Fourier space is written as

$$\mathbf{y}_t = \sqrt{\bar{\alpha}_t} \mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\Sigma, \tag{8}$$

which has SNR at timestep  $t$  and frequency  $i$  given by

$$s_t(i) := \frac{\bar{\alpha}_t \mathbf{C}_i}{(1 - \bar{\alpha}_t)\Sigma_{ii}}. \quad (9)$$

This implies:

1. DDPM: The forward process for DDPM has SNR given by  $s_t^{\text{DDPM}}(i) = \frac{\bar{\alpha}_t \mathbf{C}_i}{(1 - \bar{\alpha}_t)}$ .
2. Equal SNR: The forward process has equal SNR across every coordinate if and only if  $\Sigma_{ii} = c \mathbf{C}_i$ , where  $c$  is a universal constant. The process is ‘variance preserving’ (in the sense of (Song et al., 2020b)) if  $c = 1$ .<sup>3</sup>
3. Flipped SNR: The forward process has flipped SNR, i.e.,  $s_t^{\text{FLIP}}(i) := s_t^{\text{DDPM}}(d - i)$  if and only if  $\Sigma_{ii} = \mathbf{C}_i / \mathbf{C}_{d-i}$ .

We contrast the SNR of the standard DDPM noise schedule and the two alternate ones computed for CIFAR10 in Fig. 2, referring to App. B Figures 19 and 20 for the SNR profiles on CelebA and LSUN Church, respectively. EqualSNR corrupts information in all frequencies at the same rate by enforcing the same SNR for all frequencies at each timestep. This schedule removes all hierarchy among the frequencies and generates them at the same rate across diffusion time (Gerdes et al., 2024). FlippedSNR on the other hand reverses the ordering of frequencies of DDPM in terms of SNR, noising low frequencies first. This forces the reverse process to generate high frequencies first, then low frequencies.

Both EqualSNR and FlippedSNR have a base distribution and forward process which are data-dependent and match the coordinate-wise variance of the Gaussian noise added to the data. In fact, we propose that the SNR at timestep  $t$  is a better choice of parameterization of the corruption process for any frequency  $i$ . That is, rewriting Equation (9) as  $\bar{\alpha}_t = \frac{s_t(i)\Sigma_{ii}}{\mathbf{C}_i + s_t(i)\Sigma_{ii}}$ , we can derive the mixing coefficients as a function of the variances of the data, noise, and  $s_t(i)$ .

### A.3. Calibration

Since we are comparing different noising schemes which corrupt the signal at different rates (in the sense of having different SNR values at differing time steps) we try to ensure that the average SNR across frequencies is the same for any pair of forward processes that we are comparing. We illustrate how this can be done for the case of DDPM and EqualSNR below.

Let  $A_{\text{ddpm}}(t)$  and  $A_{\text{eq}}(t)$  denote the average SNR for DDPM and EqualSNR at timestep  $t$ , let  $d$  denote the number of frequencies, and suppose we fix  $\{\bar{\alpha}_t^{\text{ddpm}}\}_t$ . We would like to solve for  $\{\bar{\alpha}_t^{\text{eq}}\}_t$  such that  $A_{\text{eq}}(t) = A_{\text{ddpm}}(t)$  for all  $t$ . Using the formulae derived in App. A.2, we see that this is the same as:

$$\frac{\bar{\alpha}_t^{\text{eq}}}{1 - \bar{\alpha}_t^{\text{eq}}} = \frac{1}{d} \sum_i \frac{\bar{\alpha}_t^{\text{ddpm}} \mathbf{C}_i}{1 - \bar{\alpha}_t^{\text{ddpm}}} = \frac{\bar{\alpha}_t^{\text{ddpm}} \frac{1}{d} \sum_i \mathbf{C}_i}{1 - \bar{\alpha}_t^{\text{ddpm}}}$$

Solving for  $\bar{\alpha}_t^{\text{eq}}$  gives us

$$\bar{\alpha}_t^{\text{eq}} = \frac{\bar{\alpha}_t^{\text{ddpm}} \left( \frac{1}{d} \sum_i \mathbf{C}_i \right)}{\left( 1 - \bar{\alpha}_t^{\text{ddpm}} \right) + \bar{\alpha}_t^{\text{ddpm}} \left( \frac{1}{d} \sum_i \mathbf{C}_i \right)}.$$

On the other hand, if we choose to parameterize or schedule in terms of the SNR, then having decided on the SNR schedule for DDPM, the SNR schedule for EqualSNR is just the average value across frequencies of the DDPM SNR.

### A.4. Our Training Algorithm and Connection to the ELBO

We denote the corrupted signal by  $\mathbf{y}_t := \mathbf{F} \mathbf{x}_t \in \mathbb{C}^d$ , where  $\mathbf{F}$  is a linear operator and  $\mathbf{x}_t$  is the signal at timestep  $t$ . We train a model  $f_\theta(\mathbf{y}_t, t)$  by minimizing the MSE loss  $\mathcal{L}_t(\theta) := \mathbb{E}[\|\mathbf{C}^{-\frac{1}{2}}(\mathbf{y}_0 - f_\theta(\mathbf{y}_t, t))\|^2]$ , where  $\mathbf{C} \in \mathbb{C}^{d \times d}$  is Hermitian positive semidefinite. As in (Ho et al., 2020; Song et al., 2020b), minimizing this loss at each  $t$  is equivalent to maximizing a standard ELBO (evidence lower bound) on  $\log p_\theta(\mathbf{y}_0)$ .

<sup>3</sup>Note that if we insist that  $\Sigma = c \text{Cov}(\mathbf{y}_0)$  then the equal SNR property holds in *all bases* (this is shown in App. A.5).

**Proposition 2** (Loss Minimization as ELBO Maximization). *Consider a forward corruption process  $q(\mathbf{y}_1, \dots, \mathbf{y}_T \mid \mathbf{y}_0)$  and a reverse reconstruction process  $p_\theta(\mathbf{y}_0, \dots, \mathbf{y}_T)$  with  $p_\theta(\mathbf{y}_{t-1} \mid \mathbf{y}_t) := \mathcal{CN}(\mathbf{y}_{t-1}; \mu_\theta(\mathbf{y}_t, t), \frac{\alpha_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{C})$ . Minimizing the MSE loss  $\mathbb{E}[(\mathbf{y}_0 - \mu_\theta(\mathbf{y}_t, t))^\dagger \mathbf{C}^{-1} (\mathbf{y}_0 - \mu_\theta(\mathbf{y}_t, t))]$  is (up to constants) equivalent to maximizing the usual ELBO on  $\log p_\theta(\mathbf{y}_0)$ .*

*Proof Sketch.* By the standard argument from diffusion-based generative models (Ho et al., 2020; Song et al., 2020b), we have  $\log p_\theta(\mathbf{y}_0) \geq \mathbb{E}_q[\log \frac{p_\theta(\mathbf{y}_0, \dots, \mathbf{y}_T)}{q(\mathbf{y}_1, \dots, \mathbf{y}_T \mid \mathbf{y}_0)}]$ , which expands into a sum of KL divergences plus a prior term. Hence maximizing the ELBO is equivalent to minimizing  $\sum_{t=1}^T \mathbb{E}_q[D_{KL}(q(\mathbf{y}_{t-1} \mid \mathbf{y}_t, \mathbf{y}_0) \parallel p_\theta(\mathbf{y}_{t-1} \mid \mathbf{y}_t))]$ . Both  $q(\mathbf{y}_{t-1} \mid \mathbf{y}_t, \mathbf{y}_0)$  and  $p_\theta(\mathbf{y}_{t-1} \mid \mathbf{y}_t)$  are (circularly) complex Gaussian distributions with the same covariance structure (up to a scaling factor), so their KL divergence is proportional to  $\|\mathbf{y}_0 - \mu_\theta(\mathbf{y}_t, t)\|_{\mathbf{C}^{-1}}^2$  (this follows from Fact 1 below). Summing over  $t$  matches the MSE loss  $\mathcal{L}_t(\theta)$ , implying that minimizing  $\mathcal{L}_t(\theta)$  at all timesteps maximizes the ELBO.  $\square$

Let  $\det(\cdot)$  denote the determinant,  $\text{tr}(\cdot)$  denote the trace and  $(\cdot)^\dagger$  denote the conjugate transpose. We use the following fact above:

**Fact 1** (KL Divergence Between Complex Gaussian Distributions). *Let  $p(\mathbf{z}) = \mathcal{CN}(\mathbf{z}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$  and  $q(\mathbf{z}) = \mathcal{CN}(\mathbf{z}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$  be two  $d$ -dimensional complex Gaussian distributions, where  $\boldsymbol{\mu}_p, \boldsymbol{\mu}_q \in \mathbb{C}^d$  are the mean vectors,  $\boldsymbol{\Sigma}_p, \boldsymbol{\Sigma}_q \in \mathbb{C}^{d \times d}$  are the covariance matrices, which are Hermitian ( $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^\dagger$ ) and positive semidefinite.*

The Kullback-Leibler (KL) divergence  $D_{KL}(p \parallel q)$  between  $p$  and  $q$  is given by:

$$D_{KL}(p \parallel q) = \frac{1}{2} \left[ \log \frac{\det(\boldsymbol{\Sigma}_q)}{\det(\boldsymbol{\Sigma}_p)} + \text{tr}(\boldsymbol{\Sigma}_q^{-1} \boldsymbol{\Sigma}_p) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^\dagger \boldsymbol{\Sigma}_q^{-1} (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) - d \right],$$

This confirms that with identical (or proportionally scaled) covariances, the KL depends only on the mean mismatch  $(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)$  and is thus proportional to the MSE term. Consequently, the loss  $\mathcal{L}_t(\theta)$  directly aligns with the KL terms in the ELBO decomposition.

## A.5. Signal to Noise Ratio in High Dimensions

In this section, we define the signal-to-noise ratio for high-dimensional vectors and note some properties.

**Definition 4** (SNR). Let  $s$  (signal) and  $\epsilon$  (noise) be random variables in  $\mathbb{C}$ , and let  $f(s, \epsilon) = s + \epsilon$  represent a measurement process. The **signal-to-noise ratio** is defined as:

$$\text{SNR}(f) = \frac{\text{Var}[s]}{\text{Var}[\epsilon]},$$

where for any random variable  $x$  taking values in  $\mathbb{C}$ ,  $\text{Var}(x) = \text{Var}(\text{Re}(x)) + \text{Var}(\text{Im}(x))$ .

For a multivariate random variable, we define the signal-to-noise ratio in a particular direction  $v$  below.

**Definition 5** (Multivariate SNR). Let  $\mathbf{s}, \mathbf{\epsilon}$  be two random vectors in  $\mathbb{C}^d$  denoting the signal and noise respectively. Let  $f(\mathbf{s}, \mathbf{\epsilon}) = \mathbf{s} + \mathbf{\epsilon}$  be a measurement process. Then, for  $\mathbf{v} \in \mathbb{C}^d$ , we define the signal-to-noise ratio of  $f$  in the direction  $v$  to be

$$\text{SNR}_v(f) := \frac{\text{Var}(\mathbf{s} \cdot \mathbf{v})}{\text{Var}(\mathbf{\epsilon} \cdot \mathbf{v})}.$$

In this paper, we often consider the case where the SNR is constant across different frequencies of our signal in the diffusion corruption process defined. Let  $\mathbf{F}$  denote the fourier transform matrix. This corresponds to having equal SNR in the directions  $\{\mathbf{F}_i \mid i \in [d]\}$ .

An alternative way to define a constant-rate corruption process is one in which the SNR is identical *in all directions*. In Lemma 1 we show that a measurement process achieves equal SNR in all directions if and only if the covariance of the signal and the noise are proportional to each other.

A consequence of this is that it suffices to simply add covariance-matching noise in the standard pixel space to achieve constant SNR across all frequencies. Note that, since we only match the diagonal entries of the covariance, i.e. only aim for equal SNR over  $\{\mathbf{F}_i \mid i \in [d]\}$ , this does not apply to our noise schedule in § 3.

**Lemma 1** (SNR and Covariance). *Let  $\mathbf{s}$  and  $\boldsymbol{\epsilon}$  be random vectors taking values in  $\mathbb{C}^d$  such that for every  $\mathbf{v} \in \mathbb{C}^d \setminus \{0\}$ ,  $0 < \text{Var}(\mathbf{v} \cdot \mathbf{s}), \text{Var}(\mathbf{v} \cdot \boldsymbol{\epsilon}) < \infty$ . Let  $f(\mathbf{s}, \boldsymbol{\epsilon}) := \mathbf{s} + \boldsymbol{\epsilon}$  be a measurement process, and let the  $(p, q)$ -th entry of the covariance matrix of a complex random variable  $\mathbf{x}$  be given by  $[\text{Cov}(\mathbf{x})]_{p,q} := \mathbb{E}_{\mathbf{x}}[\mathbf{x}_p \mathbf{x}_q^*]$ , where the asterisk denotes the conjugate. Then, the following are equivalent:*

1. For every  $\mathbf{v}, \mathbf{v}' \in \mathbb{C}^d \setminus \{0\}$ ,  $\text{SNR}_{\mathbf{v}}(f) = \text{SNR}_{\mathbf{v}'}(f)$ .
2. For some positive constant  $c$ ,  $\text{Cov}(\mathbf{s}) = c \text{Cov}(\boldsymbol{\epsilon})$ .

*Proof.* Since for every  $\mathbf{v} \in \mathbb{C}^d \setminus \{0\}$ , the variances of  $\mathbf{v} \cdot \mathbf{s}$  and  $\mathbf{v} \cdot \boldsymbol{\epsilon}$  are in  $(0, \infty)$ , all quantities below are strictly positive. Observe that for any  $\mathbf{v}, \mathbf{v}' \in \mathbb{C}^d \setminus \{0\}$ ,

$$\frac{\text{SNR}_{\mathbf{v}'}(f)}{\text{SNR}_{\mathbf{v}}(f)} = \frac{\text{Var}(\mathbf{v}' \cdot \mathbf{s})/\text{Var}(\mathbf{v}' \cdot \boldsymbol{\epsilon})}{\text{Var}(\mathbf{v} \cdot \mathbf{s})/\text{Var}(\mathbf{v} \cdot \boldsymbol{\epsilon})} = \frac{\mathbf{v}'^T \text{Cov}(\mathbf{s})(\mathbf{v}')^*/\mathbf{v}'^T \text{Cov}(\boldsymbol{\epsilon})(\mathbf{v}')^*}{\mathbf{v}^T \text{Cov}(\mathbf{s})\mathbf{v}^*/\mathbf{v}^T \text{Cov}(\boldsymbol{\epsilon})\mathbf{v}^*} = \frac{\mathbf{v}'^T \text{Cov}(\mathbf{s})(\mathbf{v}')^*}{\mathbf{v}'^T \text{Cov}(\boldsymbol{\epsilon})(\mathbf{v}')^*} \cdot \left( \frac{\mathbf{v}^T \text{Cov}(\mathbf{s})\mathbf{v}^*}{\mathbf{v}^T \text{Cov}(\boldsymbol{\epsilon})\mathbf{v}^*} \right)^{-1}. \quad (10)$$

To see that Item 2 implies Item 1, substitute  $\text{Cov}(\mathbf{s}) = c \text{Cov}(\boldsymbol{\epsilon})$  in Equation (10).

To see that Item 1 implies Item 2, substitute  $\text{SNR}_{\mathbf{v}}(f) = \text{SNR}_{\mathbf{v}'}(f)$  in Equation (10). Rearranging, we see that for any  $\mathbf{v}, \mathbf{v}' \in \mathbb{C}^d \setminus \{0\}$ ,

$$\frac{\mathbf{v}^T \text{Cov}(\mathbf{s})\mathbf{v}^*}{\mathbf{v}^T \text{Cov}(\boldsymbol{\epsilon})\mathbf{v}^*} = \frac{\mathbf{v}'^T \text{Cov}(\mathbf{s})(\mathbf{v}')^*}{\mathbf{v}'^T \text{Cov}(\boldsymbol{\epsilon})(\mathbf{v}')^*} = c, \quad (11)$$

where  $c$  is some constant. Rearranging again and ignoring the equation involving  $\mathbf{v}'$ , we see that for any  $\mathbf{v} \in \mathbb{C}^d \setminus \{0\}$ ,  $\mathbf{v}^T \text{Cov}(\mathbf{s})(\mathbf{v})^* = c \mathbf{v}^T \text{Cov}(\boldsymbol{\epsilon})(\mathbf{v})^*$ . The conclusion then follows from Fact 2 below.

**Fact 2** (Equal Quadratic Forms). *Let  $\mathbf{A}$  and  $\mathbf{B}$  be conjugate-symmetric matrices in  $\mathbb{C}^{d \times d}$ . If  $\mathbf{v}^T \mathbf{A} \mathbf{v}^* = \mathbf{v}^T \mathbf{B} \mathbf{v}^*$  for all  $\mathbf{v} \in \mathbb{C}^d \setminus \{0\}$ , then  $\mathbf{A} = \mathbf{B}$ .*

*Proof.* Let  $\mathbf{C} = \mathbf{A} - \mathbf{B}$ . Then the condition is equivalent to having  $\mathbf{v}^T \mathbf{C} \mathbf{v}^* = 0$  for all  $\mathbf{v}$ . Suppose towards a contradiction that  $\mathbf{C}$  is not the zero matrix. Then there is some entry of  $\mathbf{C}$  that is nonzero. Let this be the entry indexed by  $(p, q)$ . Consider  $\mathbf{v}$  where  $\mathbf{v}_t = 0$  for all  $t \in [d] \setminus \{p, q\}$ . Then, we see

$$\mathbf{v}^T \mathbf{C} \mathbf{v}^* = |\mathbf{v}_p|^2 \mathbf{C}_{p,p} + |\mathbf{v}_q|^2 \mathbf{C}_{q,q} + \mathbf{v}_p \mathbf{v}_q^* \mathbf{C}_{p,q} + \mathbf{v}_p^* \mathbf{v}_q \mathbf{C}_{q,p}. \quad (12)$$

If  $p = q$ , then set  $\mathbf{v}_p = \mathbf{v}_q = 1$  to get a contradiction. Hence all the diagonal entries of  $\mathbf{C}$  are 0. If  $p \neq q$ , then taking into account the fact that  $\mathbf{C}_{p,q} = \mathbf{C}_{q,p}^*$ , we see that Equation (12) reduces to,

$$\mathbf{v}^T \mathbf{C} \mathbf{v}^* = \mathbf{v}_p \mathbf{v}_q^* \mathbf{C}_{p,q} + \mathbf{v}_p^* \mathbf{v}_q \mathbf{C}_{p,q}^*.$$

If  $\mathbf{C}_{p,q}$  has a nonzero real component, then it suffices to set  $\mathbf{v}_p = \mathbf{v}_q = 1$  to see a contradiction, since  $\mathbf{v}_p \mathbf{v}_q^* \mathbf{C}_{p,q} + \mathbf{v}_p^* \mathbf{v}_q \mathbf{C}_{p,q}^* = 2\text{Re}(\mathbf{C}_{p,q})$ . If  $\mathbf{C}_{p,q}$  has a nonzero imaginary component, then it suffices to set  $\mathbf{v}_p = 1, \mathbf{v}_q = i$  to see a contradiction, since  $\mathbf{v}_p \mathbf{v}_q^* \mathbf{C}_{p,q} + \mathbf{v}_p^* \mathbf{v}_q \mathbf{C}_{p,q}^* = -i\mathbf{C}_{p,q} + i\mathbf{C}_{p,q}^* = 2\text{Im}(\mathbf{C}_{p,q})$ .  $\square$

$\square$

### A.6. Counterexample: Breaking the Gaussian Assumption

In this section we give a simple example showing that—even if each transition in the forward noising process is Gaussian—the corresponding reverse conditional distribution  $q(x_{t-1} | x_t)$  need not itself be close (in total variation) to *any* single Gaussian. Intuitively, if the marginal  $q(x_t)$  arises from adding noise to a mixture of two well-separated Gaussians, then the reverse conditional remains bimodal and cannot collapse into a unimodal Gaussian.

**Proposition 3.** *Let  $\tau \in (0, 0.5)$  and  $\delta \in (0, \tau^{20})$  be constants. Set*

$$D_0 = \frac{1}{2} \mathcal{N}(-1, \delta^2) + \frac{1}{2} \mathcal{N}(1, \delta^2),$$

Suppose

$$x_{t-1} \sim D_0, \quad \varepsilon \sim \mathcal{N}(0, 4), \quad x_t = x_{t-1} + \varepsilon.$$

Then with probability  $1 - \tau$  over the draw of  $x_t$ , the reverse kernel  $q(x_{t-1} | x_t)$  satisfies

$$\inf_{\mu \in \mathbb{R}, \sigma > 0} D_{\text{TV}}(q(x_{t-1} | x_t), \mathcal{N}(\mu, \sigma^2)) \geq \Omega(\tau^{18})$$

In other words, no single Gaussian can approximate  $q(x_{t-1} | x_t)$  to an accuracy beyond  $O(\tau^{18})$  in total variation. We think of  $\tau$  as being a constant, and so no Gaussian can approximate  $q(x_{t-1} | x_t)$  to an arbitrary accuracy.

**Proof idea.** We will prove Proposition 3 in three stages:

1. **Compute  $q(x_{t-1} | x_t)$  explicitly.** By Bayes' rule, we show it is proportional to the original mixture times a shifted Gaussian,

$$q(x_{t-1} = x | x_t = y) \propto D_0(x) \cdot \exp\left(-\frac{(x-y)^2}{8}\right),$$

This exhibits two well-separated modes and symmetric, sub-Gaussian tails.

2. **Show sub-Gaussian decay around each mode.** With probability  $1 - \tau$  for  $\tau < 0.8$ ,  $y \in [-1 - 4\sqrt{\log(1/\tau)}, 1 + 4\sqrt{\log(1/\tau)}]$ , which will imply that in the high probability regions,  $\exp\left(-\frac{(x-y)^2}{8}\right)$  remains bounded.
3. **Lower-bound the total variation.** For any candidate  $\mathcal{N}(\mu, \tau^2)$ , we split the argument into two cases, either  $\tau^2 \leq \delta$  (i.e. the Gaussian is too narrow to cover one of the peaks) or  $\tau^2 > \delta$  (i.e. the Gaussian is too flat to have substantial overlap with either piece) and apply standard concentration/anticoncentration bounds to obtain a gap of at least  $\Omega(\tau^{18})$ .

*Proof.* **Step 1: Exact form via Bayes' Rule.** Bayes' rule asserts:

$$p(x | y) = \frac{\underbrace{p(y | x)}_{\text{likelihood}} \underbrace{p(x)}_{\text{prior}}}{\underbrace{p(y)}_{\text{evidence}}}.$$

Here:

- $p(x) = D_0(x)$  is the prior density of  $x_{t-1}$ .
- $p(y | x) = q(x_t = y | x_{t-1} = x) = \mathcal{N}(y - x; 0, 4)$  is the forward-noise likelihood.
- $p(y) = D_1(y) = \int p(y | x) p(x) dx$  is the marginal of  $x_t$ .

Since convolution of Gaussians yields another Gaussian,  $D_1 = D_0 * \mathcal{N}(0, 4) = \frac{1}{2} \mathcal{N}(-1, \delta^2 + 4) + \frac{1}{2} \mathcal{N}(1, \delta^2 + 4)$ . Thus for any  $y$ :

$$q(x_{t-1} = x | x_t = y) = \frac{\exp\left(-\frac{(y-x)^2}{8}\right) \left[ \frac{1}{2} \exp\left(-\frac{(x+1)^2}{2\delta^2}\right) + \frac{1}{2} \exp\left(-\frac{(x-1)^2}{2\delta^2}\right) \right]}{\frac{1}{2} \exp\left(-\frac{(y+1)^2}{2(4+\delta^2)}\right) + \frac{1}{2} \exp\left(-\frac{(y-1)^2}{2(4+\delta^2)}\right)}.$$

Since we want to view this as a distribution over  $x$  for a fixed value of  $y$ , we may drop the denominator (since it is not a function of  $x$ ), to see that

$$q(x_{t-1} = x \mid x_t = y) \propto \exp\left(-\frac{(y-x)^2}{8}\right) D_0(x)$$

Next, we would like to show that with reasonable probability over  $x$  and  $y$ ,  $A(x, y) := \exp\left(-\frac{(y-x)^2}{8}\right)$  is bounded between two constants for  $x, y$  lying in the high probability region.

**Step 2: Uniform control on  $A(x, y)$ .** For any  $0.5 > \tau > 0$ , define

$$I = [-1 - 8\sqrt{\log(1/\tau)}, 1 + 8\sqrt{\log(1/\tau)}].$$

Since  $\delta < 2$ , by standard Gaussian tail bounds,  $\Pr_{y \sim D_1}[y \notin I] \leq \tau$ . We now check that for  $y \in I$ ;  $x \in [-1 - 4\delta\sqrt{\log(1/\tau)}, -1 + 4\delta\sqrt{\log(1/\tau)}] \cup [1 - 4\delta\sqrt{\log(1/\tau)}, 1 + 4\delta\sqrt{\log(1/\tau)}]$  (the high probability region for  $D_0$ ) and  $\delta \in (0, 1)$ ,

$$\Theta(\tau^{4^2 \cdot 3^2 / 8}) < \exp\left(-\frac{(2 + 4 \cdot 3\sqrt{\log(1/\tau)})^2}{8}\right) < |A(x, y)| < 1$$

This means that one of the modes is scaled down by at least  $\Theta(\tau^{18})$ .

**Step 3: Lower-bounding total variation.** Let  $N = \mathcal{N}(\mu, \sigma^2)$  be arbitrary. We consider two regimes. First, observe that we may think of  $q(x_{t-1} = x \mid x_t = y) \propto A(x, y) \cdot D_0$ , where  $A(x, y) \in (\Theta(\tau^{18}), 1)$ .  $A$  will serve as a re-weighting of the mixture components; in the extreme case, the ratio of the mass of the smaller component to the mass of the larger component is  $\Theta(\tau^{18})$ .

- (a) If  $\sigma^2 \leq \delta$ :  $N$  is too narrow. Align its mean with the mode having a larger mass, since this is the maximum overlap we can achieve; say WLOG this is  $\mu = 1$ . Outside a window of width  $O(\delta)$  around  $x = -1$ , the overlap is  $\geq \Theta(\tau^{18}) - \exp(-\Omega(1)/\delta^2)$  (mass of the smaller component under  $q$  minus overlap of the tail of  $N$ ).
- (b) If  $\sigma^2 > \delta$ :  $N$  is too flat. Consider the tail beyond  $10\delta$  of the peak at  $+1$ , when viewed as a mixture component of  $q$ , and when  $\delta$  is sufficiently small:

$$q(|x - 1| \geq 10\delta) \geq (1 - \Theta(\tau^{18})) \cdot (1 - \exp(-10^2/2))$$

However, the mass that  $N$  places in this region is small:

$$N(|x - 1| \geq 10\delta) \leq O(\sqrt{\delta})$$

Hence  $D_{TV}(q, N) \geq 0.9 - \Theta(\tau^{18}) - O(\sqrt{\delta})$ .

In both cases we obtain

$$\inf_{\mu, \tau} D_{TV}(q(x_{t-1} \mid x_t), \mathcal{N}(\mu, \tau^2)) \geq \Theta(\tau^{18}) - O(\sqrt{\delta}).$$

Which, for a sufficiently small choice of  $\delta$ , completes the proof.  $\square$

## B. Additional experimental details and results

In this appendix we provide further details and results on the four experimental analyses in § 4 (App. B.1 to App. B.4), further describe the implementation of figures in the main text (App. B.5), and present additional results beyond the main text (App. B.6). We also briefly present the limitations of our work.

**Limitations.** Our work has three major limitations: 1) While we saw marked improvements for higher resolution experiments, our largest imaging dataset is of resolution  $128 \times 128$ , well beyond state-of-the-art generative models. 2) While the FlippedSNR forward process did not succeed in numerous experiments, underlining the importance of low-to-high generation, we cannot prove that such a forward process cannot be learned. 3) Even though many other modalities have the Fourier power law property, our real-world experiments investigate standard imaging benchmarks only.

### B.1. Analysis 1: The reverse process learns to mirror its forward

**On Figures 2, 19 and 20.** Fig. 2 illustrates the SNR of the variance of the Fourier coefficients of the forward and reverse processes for DDPM and EqualSNR on the CIFAR10 dataset (Krizhevsky et al., 2009). Frequencies are sorted by Manhattan distance to the corner of Fourier space (ascending order), resulting in low frequencies arranged at the top, and high frequencies at the bottom of the heatmap, and are averaged across channels. Figures show that DDPM noises high-frequencies at very early time steps, while EqualSNR noises all frequencies at a constant rate. The forward process shows how the Fourier frequency values are corrupted in time, and the reverse process shows how a trained model denoises the corrupted frequencies. We observe that both models learn to denoise the Fourier coefficients at the same rate as the forward process dictates. Every plot shows the contour curves on 10 dB, 0 dB, -10 dB, -20 dB and -30 dB to facilitate visualize the trend of corruption across frequencies.

To draw the forward process figures, we follow equation (4) and compute SNR for all timesteps  $t$ , given a schedule  $\bar{\alpha}_t$ . Frequencies are sorted from lowest (top) to highest (down); time runs between the data distribution ( $t = 0$ ) and the base distribution ( $t = T$ ) with  $T = 1000$  steps. Explicitly, we sample an image and white noise, compute their Fourier representations and combine them together following equation (4). Noise is weighted with the schedule in both cases, and normalized with  $C^{1/2}$  in the case of EqualSNR as in Algorithm 1. After computing the variances of the Fourier coefficients across a dataset, values are converted to decibels following the formula  $dB = 10 \log_{10}(\cdot)$  and displayed in a heatmap. Contour curves are running averages over the values that approximate best the dB threshold across frequencies.

To draw the reverse process figures, we start with a corrupted image at time  $t = T$  and do a step by step denoising process until  $t = 1$  following Algorithm 2. Our parametrized UNet predicts  $\hat{y}_t$  after Fourier transformation, and from it we estimate the noise  $\hat{\epsilon}_0$  at time  $t = 0$ :

$$\hat{\epsilon}_0 = \frac{1}{\sqrt{\alpha_t}} (\mathbf{y}_t - \sqrt{\bar{\alpha}_t} \mathbf{y}_0). \quad (13)$$

We compute the SNR by weighting the predicted signal and noise appropriately, as signal and noise are now distinct:

$$SNR(\mathbf{y}_{t-1}) = \frac{\sqrt{\bar{\alpha}_{t-1}} \text{Var}[\hat{\mathbf{y}}_0]}{\sqrt{1 - \bar{\alpha}_{t-1}} \text{Var}[\hat{\epsilon}_0]} \quad (14)$$

We compute the variances of the coefficients across the whole dataset and convert the SNR values to decibels same as in the forward process for representation purposes.

Figures 19 and 20 illustrate the forward process on the CelebA and LSUN datasets, computed on 64x64 and 128x128 resolutions, respectively. The plotted figures retain the same schedule as the one from Fig. 2 and we notice that the high frequencies are noised excessively very early on. This observation agrees with the analysis from (Hoogeboom et al., 2023) that requires the schedules to be rescaled depending on the resolution of the images. Our proposed visualization provides a useful tool to study the correct planning of the forward process.

**On Fig. 4.** This figure presents the forward and backward pass trajectory of our diffusion models trained on standard imaging benchmarks. For the forward pass, we sample using the push-forward distributions Equation (2) for DDPM, and  $\mathbf{y}_t = \sqrt{\bar{\alpha}_t} \mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\Sigma$  (see Definition 3) for EqualSNR and FlippedSNR, respectively. For the backward, we draw trajectories using standard DDIM for the model trained with a DDPM forward process, and using Algorithm 2 for EqualSNR and FlippedSNR, respectively.

The low- and high-pass filtered images are computed by setting a subset of the Fourier space coefficients, specifically those within (low-pass) or outside (high-pass) a certain distance to the center of the Fourier space representation, to 0 while using identity for all others. This is achieved by masking the Fourier space representation of the (noisy) images (see Fig. 7 for an illustration).

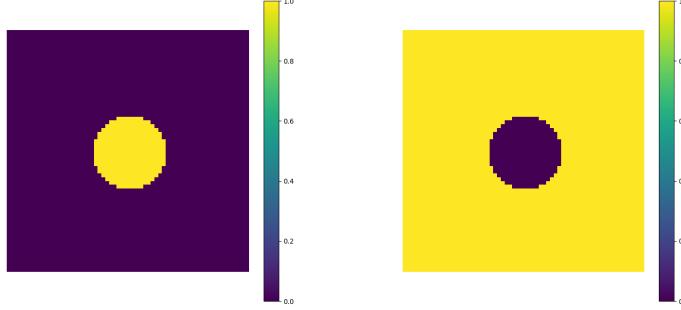


Figure 7: Visualisation of the [left] low- and [right] high-pass filter masks.

Figures 22 to 27 present further examples of the forward and backward process of DDPM, EqualSNR and FlippedSNR for the complete time interval  $t = 0$  to  $t = T$ , and also show the phase of the Fourier space representation as an additional row.

## B.2. Analysis 2: Faster noising degrades high-frequency generation in DDPM; EqualSNR overcomes this

**High Frequency/Low Frequency Classifier and Statistical Testing Procedure.** The main idea is to differentiate between real and generated data by focusing on the high-frequency magnitudes of the images. This difference can be quantified using a classifier trained to distinguish these features. High classifier accuracy indicates low generation quality of the high frequencies. Following Dzanic et al. (2020), instead of using all or a portion of the high-frequency magnitudes as classifier inputs, we fit a regression line to the log-magnitudes and use the slope and intercept as input parameters. Specifically, we find  $b$  and  $a$  such that:

$$m_i = a \left( \frac{f_i}{\tau} \right)^b,$$

where  $m_i$  is the magnitude of the  $i$ -th frequency and  $f_i = \frac{i}{F}$ , with  $F$  being the number of frequencies. Furthermore,  $\tau$  is the proportion of frequencies considered, such that  $f_i \in [\tau, 1]$ . For each image, we obtain parameters  $a$  and  $b$ , which are used as features for a logistic regression model with targets indicating whether the image is real or generated. To quantify the significance of the obtained accuracy, we split the datasets into equal-sized chunks. For example, with 50,000 generated and real images, we create 100 splits, each containing 500 images per class. We then use each batch to compute a  $p$ -value via permutation testing that tests whether the accuracy is close to a random chance i.e. 0.5. If the  $p$ -value is less than the chosen significance (in this case 0.05 or 0.01), we can reject the null hypothesis that the two distributions are the same. Then, we count how many out of the 100 independent tests are correctly rejected, i.e. how many are the true positives. A large number of true positives indicates that we can differentiate the two classes with a high probability.

Similarly, we can construct a classifier for the low-frequency components. Table 3 shows that the mean accuracies for both cases are close to chance. The number of true positives is similar between them, with standard DDPM performing slightly better.

## B.3. Analysis 3: When high-frequency information matters: a synthetic study

**The Dots Experiment.** We sampled 50,000 images with 48 to 50 randomly placed white pixels (see Fig. 8) and conducted experiments using 10,000 training steps with a cosine noise schedule, utilizing the same U-Net architecture as in the CIFAR10 experiment. In this experiment, we compare EqualSNR and standard DDPM calibrated to EqualSNR. In the main paper, we argued that EqualSNR generates samples with higher pixel intensity compared to DDPM. Examples of this issue can be seen in Fig. 8.

Table 3: Low frequency generation of DDPM and EqualSNR: Classifier mean accuracy from 100 runs and the corresponding number of true positives at significance levels 0.05 and 0.01 (correctly rejected statistical two-sample tests based on the classifier’s performance).

Freq. band	DDPM			EqualSNR		
	Mean Acc.	# TP at 0.05	# TP at 0.01	Mean Acc.	# TP at 0.05	# TP at 0.01
5%	0.492	3%	2%	0.512	14%	2%
15%	0.501	5%	2%	0.509	7%	2%
25%	0.508	13%	4%	0.511	13%	5%

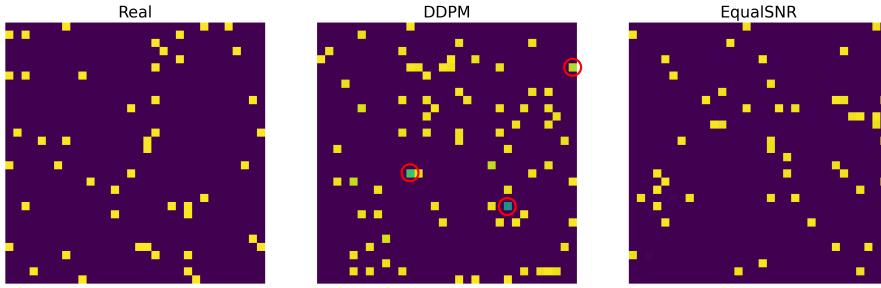


Figure 8: Dots Examples in the DDPM and EqualSNR case. Yellow corresponds to ‘white pixels’ (highest intensity) on this colour scale. The lower intensity images are marked by a red circle.

#### B.4. Analysis 4: EqualSNR performs on par with DDPM on imaging benchmarks.

In our implementation we used a UNet architecture derived from HuggingFace Diffusers’ library to model our denoiser. We feed the UNet input images and train our models by parametrizing the loss with the output of the UNet corresponding to images of the data distribution ( $t = 0$ ) in the case of DDPM. For EqualSNR we convert the output of the UNet to Fourier domain and use its values as input into the loss  $\mathcal{L}_t = \|\mathbf{C}^{-1/2}(\mathbf{y}_0 - \hat{\mathbf{y}}_0)\|^2$  (see Algorithm 1) against the input image in Fourier domain. Note, that the loss using Fourier coefficients is a novel parametrization providing competitive performance. We train all models for 800k steps using a cosine schedule and a learning rate of 0.0002. We report results on CIFAR10 ((Krizhevsky et al., 2009);  $32 \times 32$ ), CelebA ( $64 \times 64$ ), and LSUN Church ( $128 \times 128$ ) datasets.

Table 1 shows Clean-FID scores (Parmar et al., 2022), where the values we provide are on par with values provided in their repository from other models they evaluate. We chose to present this implementation in the main paper because it fixes multiple issues when comparing FID scores used by the community. We further provide in Table 4 PyTorch-FID implementation presenting values comparable to DDIM scores (Song et al., 2020a). These values are significantly lower, highlighting inconsistencies between package implementations.

We tried feeding Fourier values directly into the input of the UNet but this parametrization did not seem to provide competitive results.

#### B.5. Further details on other figures

We here provide further details on computing the figures in the main text and corresponding extended figures in the Appendix which have not yet been discussed above.

**On Figures 1 and 16.** We acknowledge (Dieleman, 2024) for inspiration on these figures. The [center] and [right] figures are GIFs best viewed in Adobe Reader.

[Left]. We present four data modalities and corresponding datasets exhibiting the Fourier power law property: images (CIFAR10 (Krizhevsky et al., 2009)), videos (Kinetic600 (Kay et al., 2017)), audio (GTZAN Music Speech (Tzanetakis, 1999)) and Cryo-EM derived protein density maps (EMDB (wwPDB Consortium, 2023)). The Fourier transform is applied to a different number of dimensions (D) for each of these datasets: 2D for images, 3D (spatial and time) for videos, 1D

Table 4: EqualSNR performs on par with DDPM on standard imaging benchmarks. We measure performance using torcheval FID ( $\downarrow$ ).

T	CIFAR10 (32 × 32)				CelebA (64 × 64)				LSUN Church (128 × 128)			
	50	100	200	1000	50	100	200	1000	50	100	200	1000
DDPM schedule	7.7	6.1	5.35	4.81	4.93	2.60	2.05	1.87	28.72	20.23	17.99	16.63
EqualSNR (calibrated) schedule	6.5	5.1	4.57	4.27	3.48	2.56	2.36	2.19	17.67	15.4	14.43	13.61
DDPM (calibrated) schedule	9.95	6.17	4.73	3.71	10.93	4.00	2.22	1.86	56.88	27.9	18.61	15.3
EqualSNR schedule	8.36	5.65	4.68	3.9	5.83	3.71	3.08	2.71	27.55	21.60	18.99	17.27

(time) for audio, and 3D for protein density maps. As the protein density maps are of different size between samples, we interpolate them to fixed-size tensors of size [200, 200, 200], equal for all dimensions. Audio and video samples are trimmed across the time dimension to 2 seconds (44100 values) and 1 second (30 frames), respectively, such that all items in a dataset are of equal dimension. We compute the signal variance on (a subset of) the Fourier-transformed dataset. We plot running averages and running standard deviations, which illustrates the overall trend. Frequencies are sorted by Manhattan distance to the center of Fourier space (ascending order, i.e. low to high frequency).

[Center] and [Right]. We plot the signal and noise variance at different timesteps of the forward process for DDPM and EqualSNR, respectively. The signal variance is computed over the entire CIFAR10 dataset. We refer to Fig. 2 for details on the calculation.

**On Fig. 3.** Further to our explanation in § 3 we provide the following details. While the distributions  $q(\mathbf{y}_t)$  and  $q(\mathbf{y}_{t-1})$  are in general intractable, we can approximate them as Monte Carlo estimates of  $q(\mathbf{x}_t) = \mathbb{E}_{\mathbf{y}_0 \sim q(\mathbf{y}_0)} q(\mathbf{y}_t | \mathbf{y}_0)$  where  $q(\mathbf{y}_t | \mathbf{y}_0)$  is the push-forward distribution Equation (2) (and similarly for  $q(\mathbf{y}_{t-1})$ , and  $q(\mathbf{y}_0)$  is the Fourier-transformed data distribution. We use 5000 samples to estimate  $q(\mathbf{y}_t)$  and  $q(\mathbf{y}_{t-1})$ . We plot histograms of these estimates (green, red) and the Gaussian  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$  (blue) on the right. Since  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$  is datapoint-specific, we arbitrarily choose the 70%-quantile of  $q(\mathbf{y}_{t-1})$  as the mean of  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$ , a representative point. The smooth posterior distribution  $q(\mathbf{y}_{t-1} | \mathbf{y}_t)$  are computed using Kernel Density Estimation (KDE) with the same bandwidth hyperparameter across all frequencies and timesteps. Figures 9 to 14 present further timesteps and frequencies, and additional plots visualising the marginals estimated via Kernel Density Estimation, and the ratio of the marginals with the Gaussian  $q(\mathbf{y}_t | \mathbf{y}_{t-1})$  overlayed, for DDPM and EqualSNR.

## B.6. Further experimental illustrations and results

In this section we provide additional experimental results augmenting those presented in the main text.

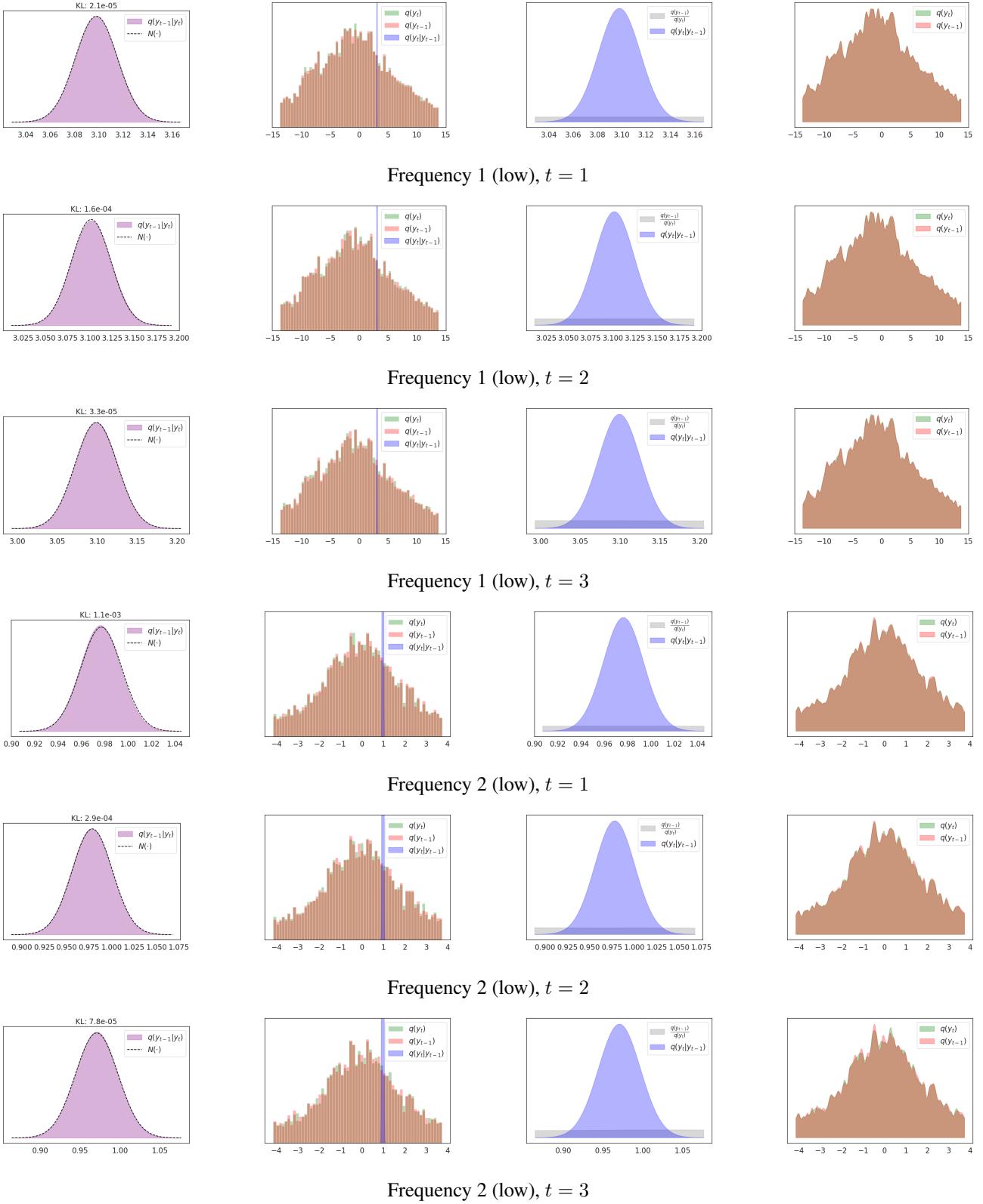


Figure 9: Analysis of violations of the Gaussian assumption in DDPM (1 of 3).

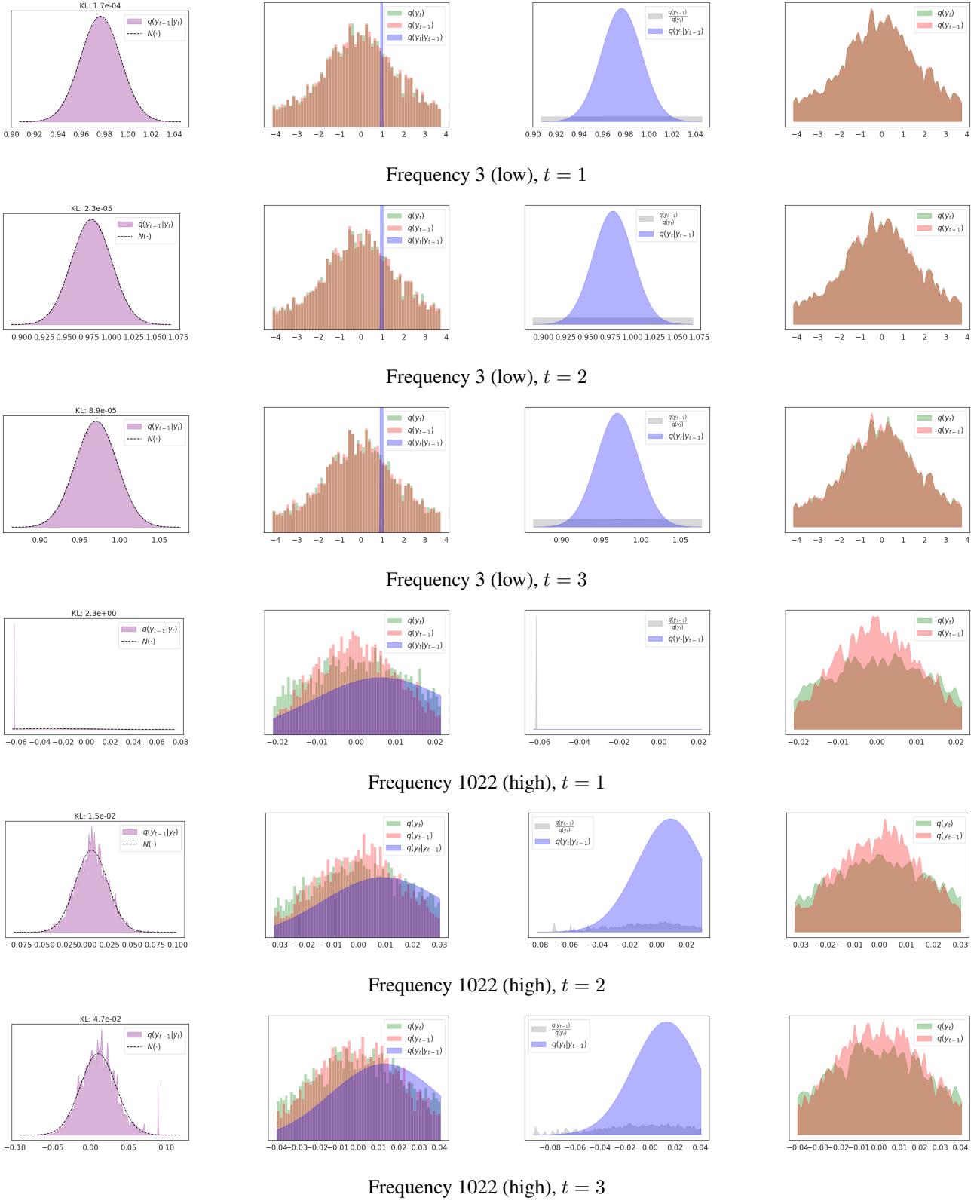


Figure 10: Analysis of violations of the Gaussian assumption in DDPM (2 of 3).

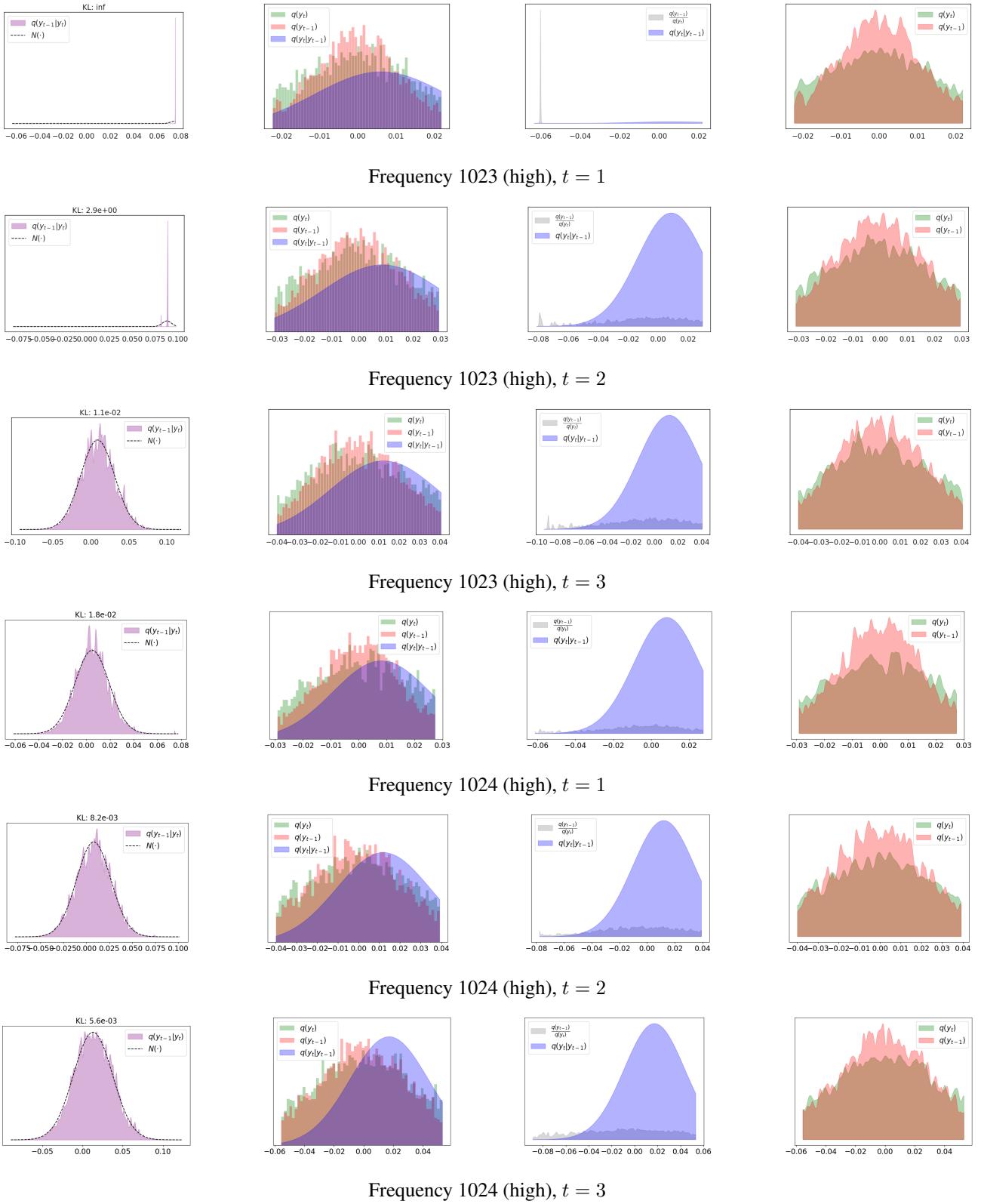


Figure 11: Analysis of violations of the Gaussian assumption in DDPM (3 of 3).

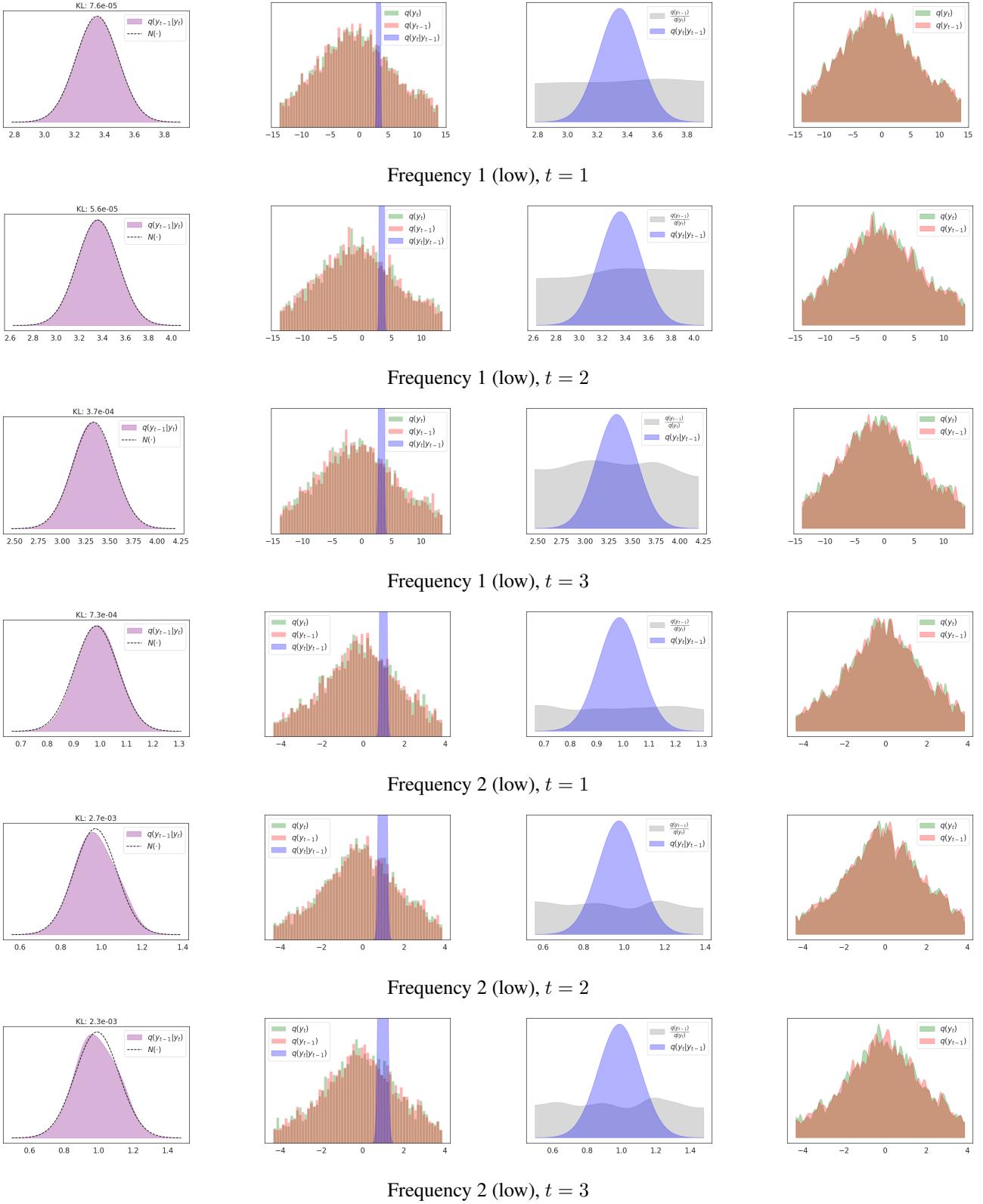


Figure 12: Analysis of violations of the Gaussian assumption in EqualSNR (1 of 3).

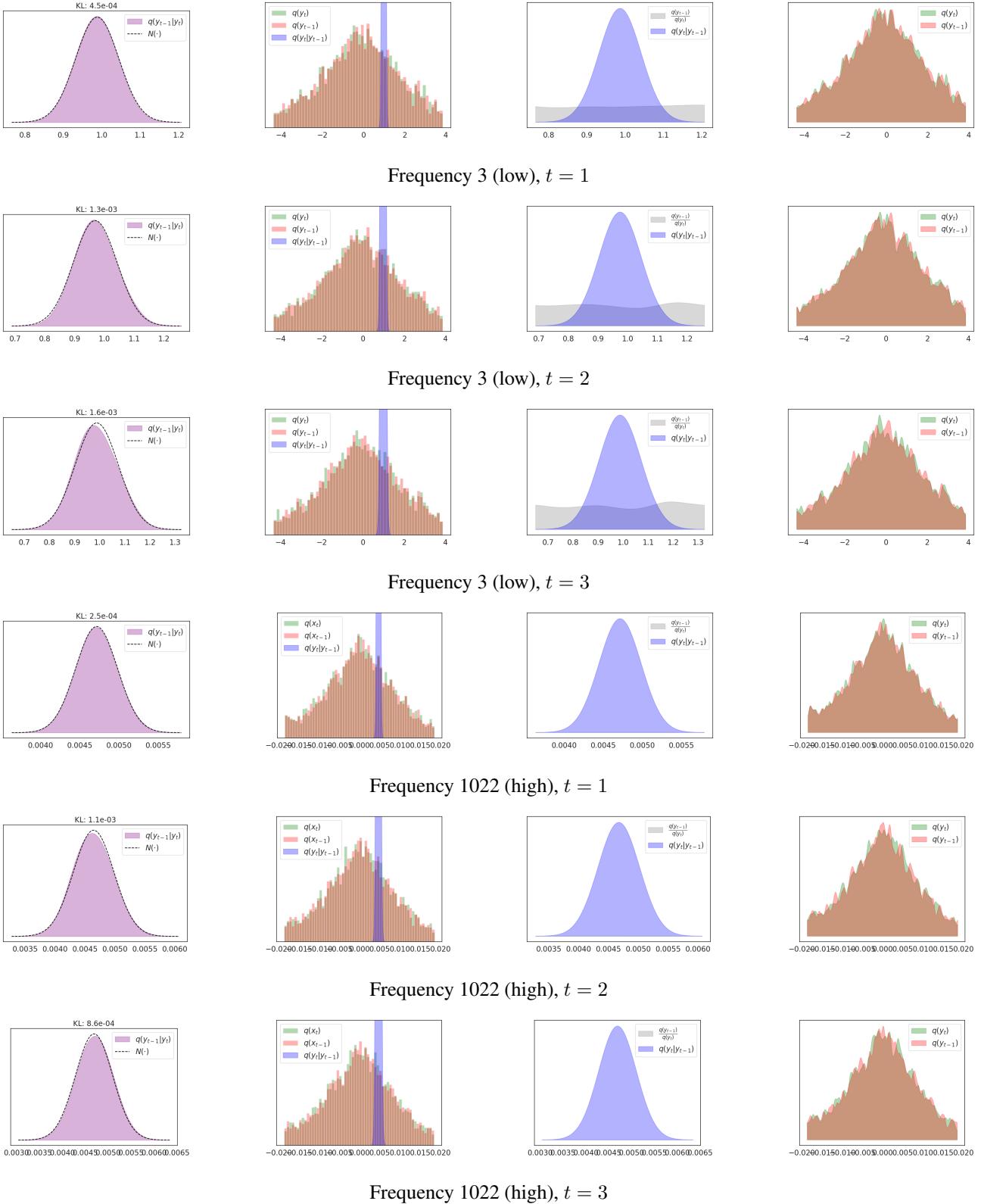


Figure 13: Analysis of violations of the Gaussian assumption in EqualSNR (2 of 3).

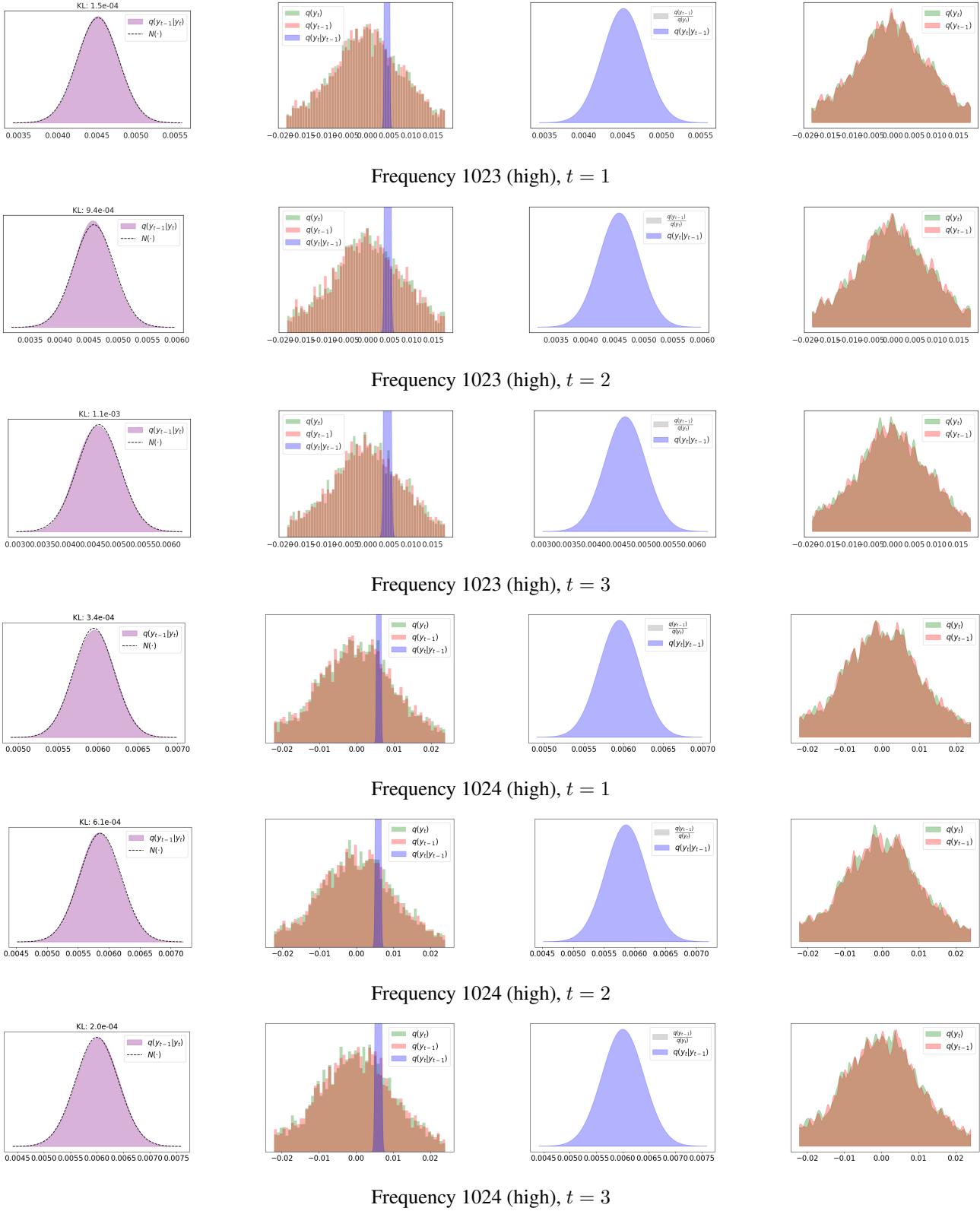


Figure 14: Analysis of violations of the Gaussian assumption in EqualSNR (3 of 3).

**Training a FlippedSNR diffusion model is challenging.** As part of our experiments we tried to train with an inverted or *flipped* SNR frequency profile as compared to DDPM (FlippedSNR), see Fig. 21. The idea of this schedule is to generate high frequencies before low frequencies in the reverse process, by having a noising process that corrupts the high frequencies gently at early timesteps, and corrupts low frequencies at a higher rate. This noising process was accomplished by computing the Fourier variance of the data, and inverting the forward process by adding noise that *flipped* the heatmap (see Figs. 21 and 24 for an illustration). However, in spite of numerous attempts, we did not manage to learn the reverse process appropriately (see Fig. 27 for a visualisation of generated samples), and hence failed to approximate the data distribution. In the following, we describe these attempts further.

We tried naive approaches such as multiple beta schedules, including cosine, linear, as well as more complicated variants which change the SNR linearly and with a cosine function, and different start and end values for the SNR. In particular, we calculated the highest SNR values at  $t = 1$  in standard DDPM (about 40 dB for low frequencies) with a standard cosine schedule. Similarly, lowest SNR values on high frequencies were around 5 dB also at  $t = 1$ . This disparity corroborates that frequencies are not noised at the same rate.

We speculated that if we flipped the heatmap profile and increased the frequencies with lowest SNRs (in the flipped regime low frequencies would be very quickly corrupted), we could possibly train successfully. We shifted the start SNR value to 60 dB (to improve the lowest SNR profile), and we kept the end SNR value the same at -40 dB. We also tried shifting to -20 dB to maintain the same range. We also trained with different number of discretization steps, i.e.,  $T \in \{1000, 5000, 10000\}$ . However, all these approaches failed to train. These experiments provide partial evidence that generating high before low frequencies may be challenging or not efficient, but further experiments are required to confirm this hypothesis.

Fig. 15 presents the  $\mathbf{C}$  matrix for the three standard imaging benchmarks used in § 4.

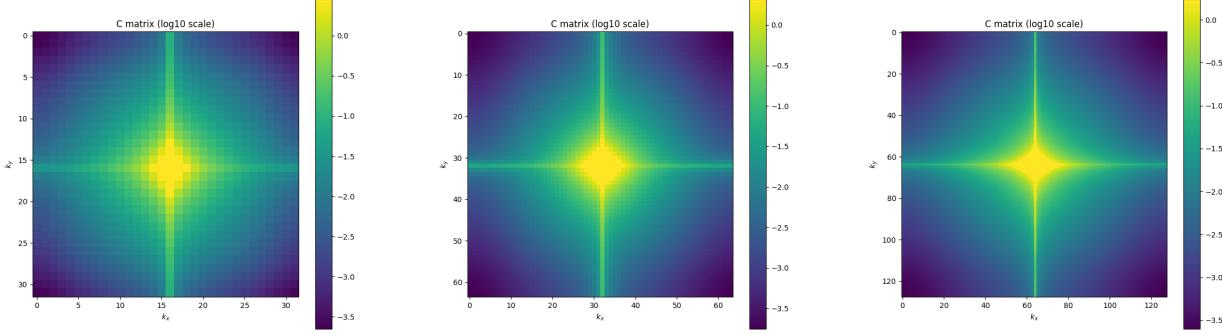


Figure 15: The signal variance  $\mathbf{C}$  in Fourier space (magnitude) on a log scale (with basis 10) for CIFAR10 [left], CelebA [centre] and LSUN Church [right]. For visualisation purposes, the largest value plotted in bright yellow corresponds to a value larger or equal to the .95-quantile of  $\mathbf{C}$ .

Fig. 16 illustrates the SNR trajectory of the FlippedSNR forward process, discussed in App. A, corresponding to the respective illustrations of DDPM and EqualSNR in Fig. 1.

Figure 16: The alternate FlippedSNR forward process noises *low-frequency* components before and faster than high-frequency components, flipping the SNR profile of DDPM. This figure complements the corresponding figures for DDPM and EqualSNR in Fig. 1. The GIF is best viewed in Adobe Reader.

The goal of Fig. 17 is—augmenting the view of Fig. 2—to characterise that the learned reverse process mirrors the forward process in diffusion models. This figure computes the per-frequency variances of samples generated via the respective push-forward distributions (in the forward process), and by sampling with DDIM or Algorithm 2 (in the reverse process), respectively. It hence provides a notion of per-frequency variability at a given diffusion timestep. It can also be interpreted relatively to the base variability at  $t = 0$ , which indicates when certain frequencies stabilise.

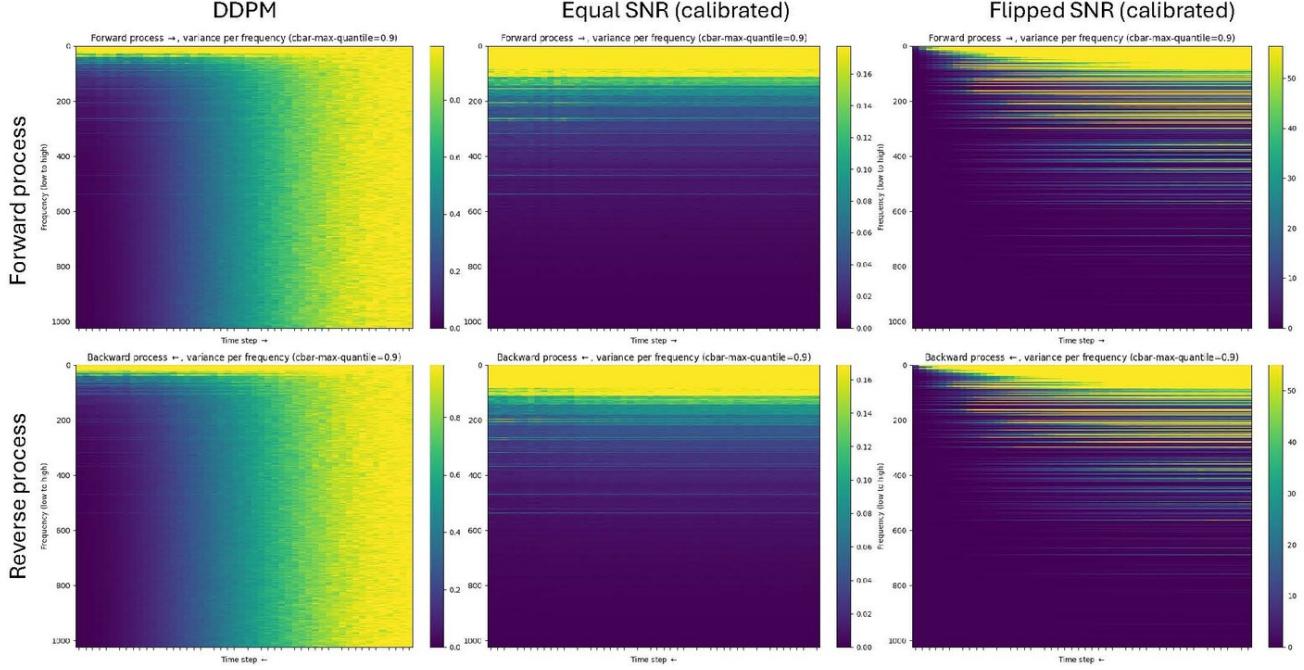


Figure 17: *The reverse process learns to mirror the forward process.* We show the variances at time  $t$  of the [top] forward and [bottom] backward process in Fourier space, for [left] DDPM, [centre] ESNR, and [right] FlippedSNR.

In Fig. 18 we illustrate the  $\mathbf{C}$  matrix, i.e. the dimension-wise variance in pixel space (as opposed to in Euclidean space), illustrating the how the variance of the data spreads spatially across an image. In particular, we can see that the variance deviates from 1, rendering DDPM not a variance-preserving forward process, in the sense that the variance changes throughout diffusion time.

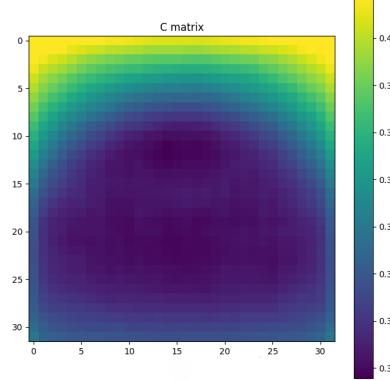


Figure 18: *DDPM on images normalised to  $[-1, 1]$  is not variance-preserving*, i.e. the variance changes throughout the forward process. To see this, we plot the coordinate-wise signal variances  $\text{diag}(\text{Cov}(\mathbf{x}_0))$  in *pixel space* (in contrast to Fig. 15), computed on the CIFAR10 dataset whose images have been normalised to the range  $[-1, 1]$ .

In Figures 19 and 20 we present SNR heatmaps for the forward process on the higher-resolution datasets CelebA and LSUN Church datasets, corresponding to Fig. 2. In Fig. 21 we also present the SNR heatmap for the forward process of FlippedSNR on CIFAR10.

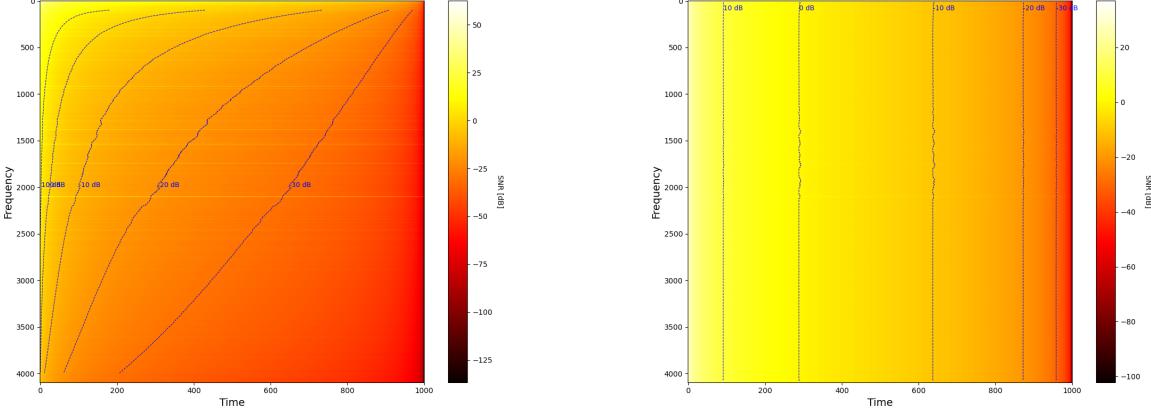


Figure 19: SNR of the [left] DDPM and [right] EqualSNR forward processes on CelebA ( $64 \times 64$ ).

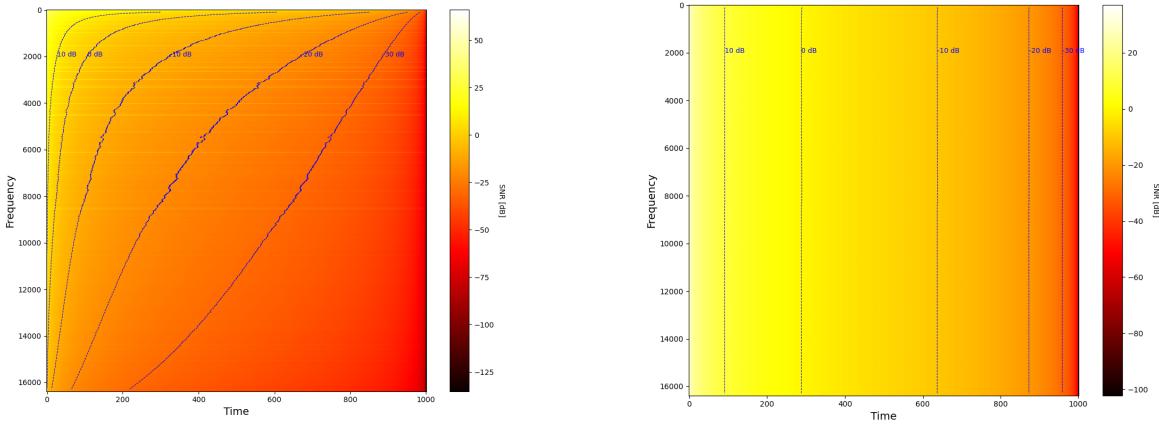


Figure 20: SNR of the [left] DDPM and [right] EqualSNR forward processes on LSUN Churches ( $128 \times 128$ ).

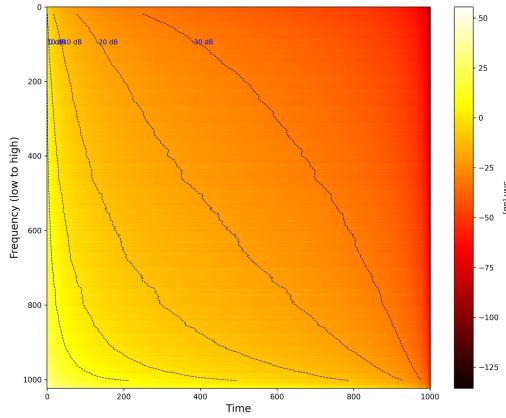


Figure 21: Flipped SNR forward process on CIFAR10 ( $32 \times 32$ ).

Figures 22 to 24 present further examples illustrating the *forward* process of DDPM, EqualSNR and FlippedSNR, augmenting Fig. 4 in the main text.

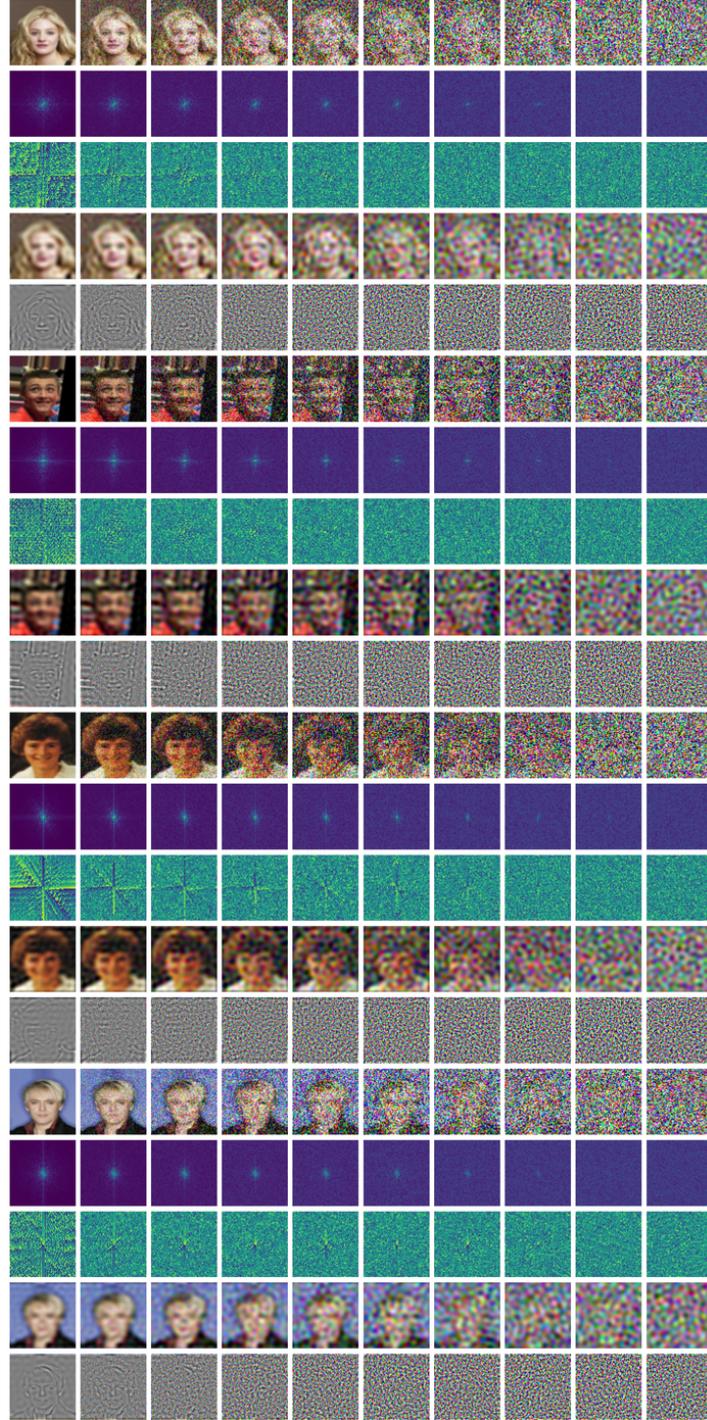


Figure 22: Forward process of DDPM, at uniformly-spaced timesteps between  $t = 0$  and  $t = T$  (left to right). Each block of five rows visualises the noising process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).

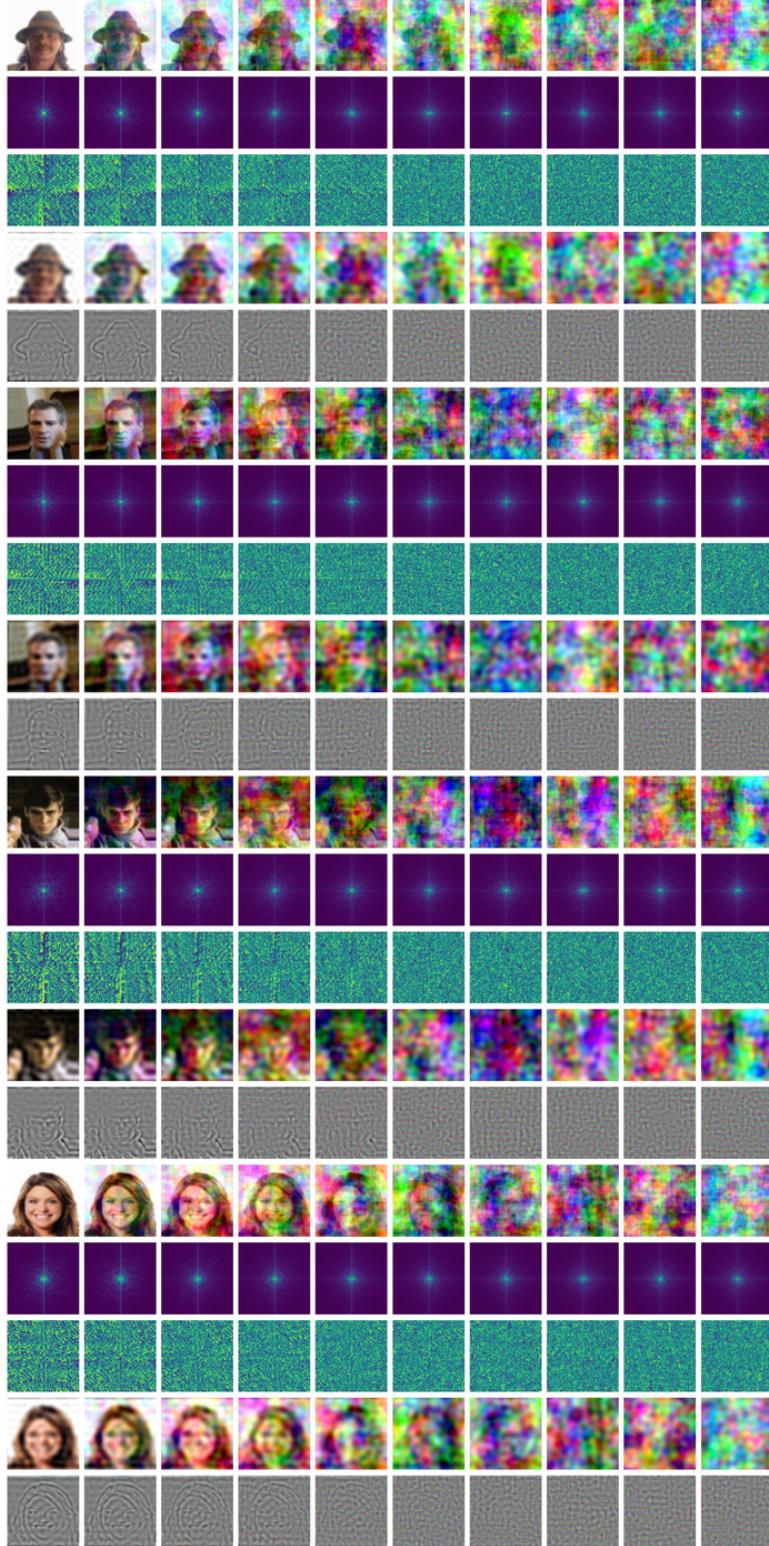


Figure 23: Forward process of EqualSNR, at uniformly-spaced timesteps between  $t = 0$  and  $t = T$  (left to right). Each block of five rows visualizes the noising process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).



Figure 24: Forward process of FlippedSNR, at uniformly-spaced timesteps between  $t = 0$  and  $t = T$  (left to right). Each block of five rows visualises the noising process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).

Figures 25 to 27 present further examples illustrating the *reverse* process of DDPM, EqualSNR and FlippedSNR, augmenting Fig. 4 in the main text. Note that the FlippedSNR diffusion model could not be trained successfully, resulting in a deteriorated sample quality.

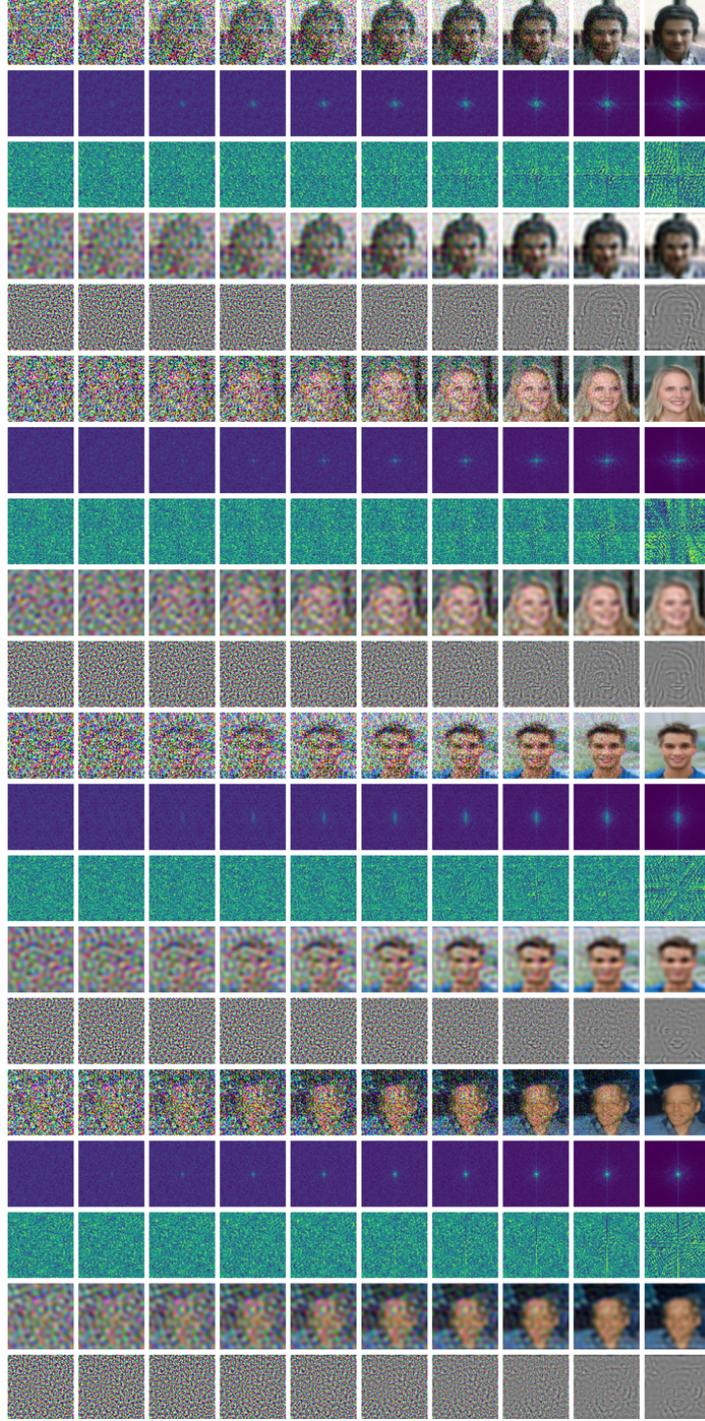


Figure 25: Reverse process of DDPM, at uniformly-spaced timesteps between  $t = T$  and  $t = 0$  (left to right). Each block of five rows visualises the generative process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).

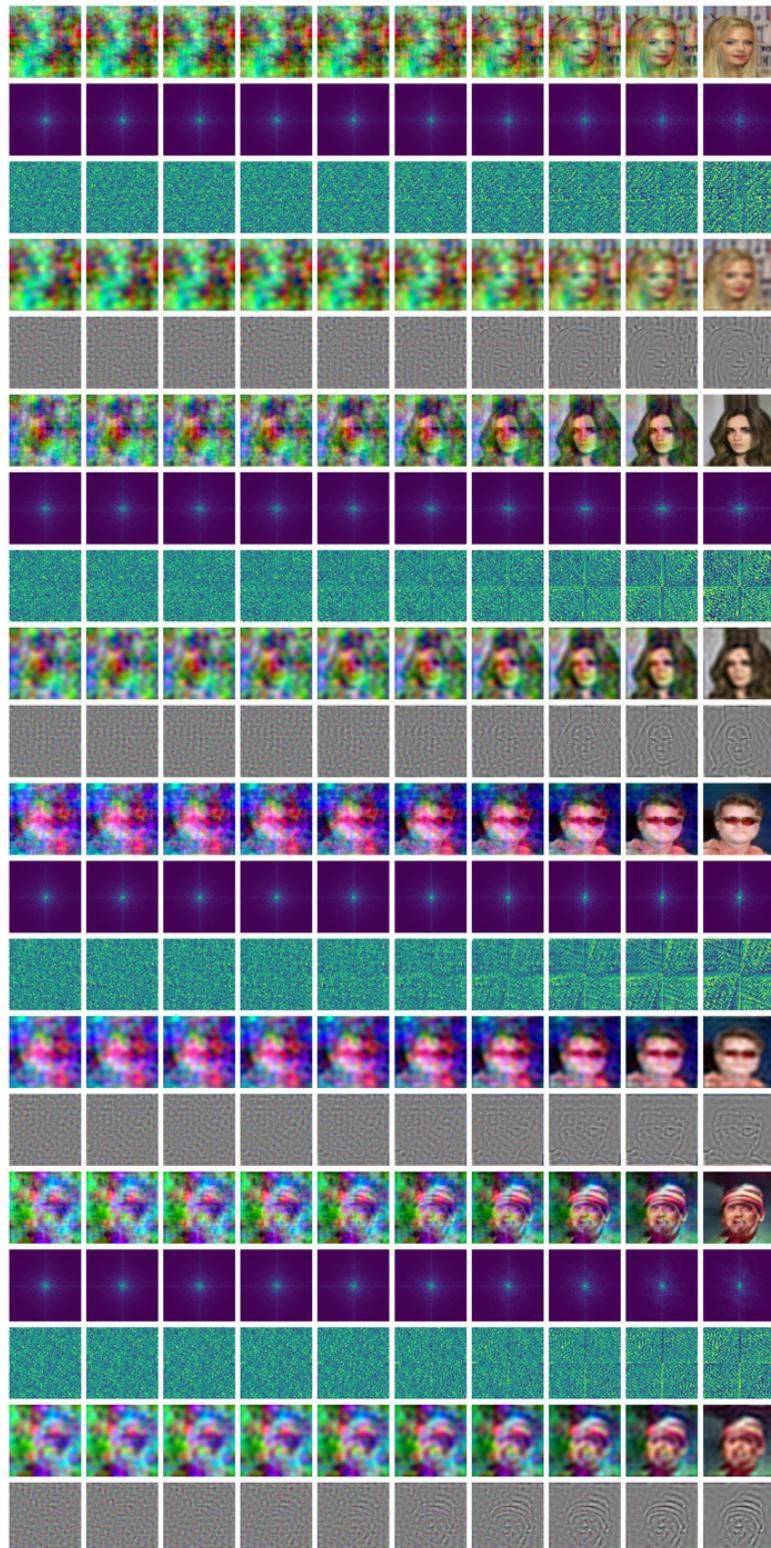


Figure 26: Reverse process of EqualSNR, at uniformly-spaced timesteps between  $t = T$  and  $t = 0$  (left to right). Each block of five rows visualises the generative process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).

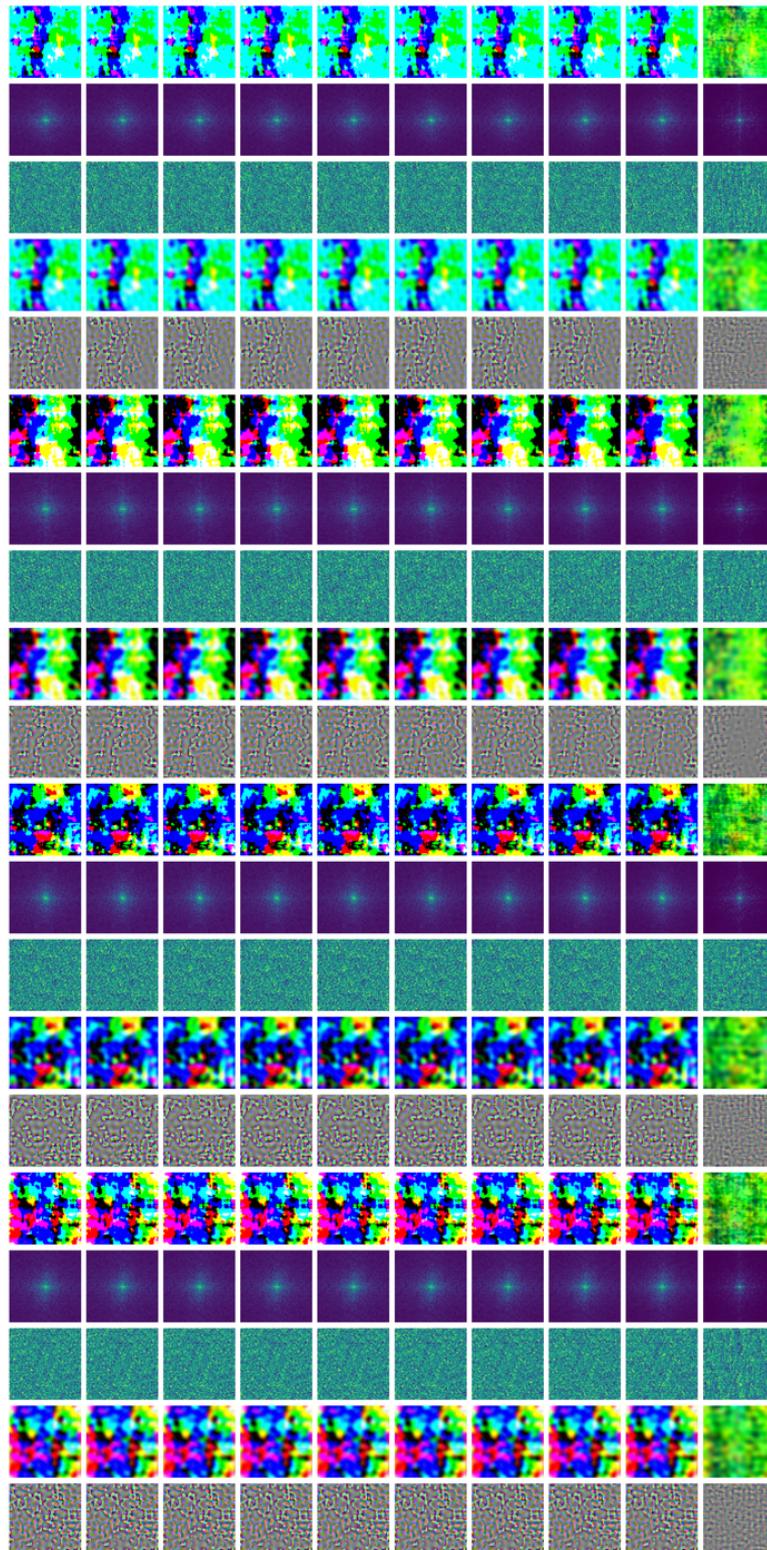


Figure 27: Reverse process of FlippedSNR, at uniformly-spaced timesteps between  $t = T$  and  $t = 0$  (left to right). Each block of five rows visualises the generative process in pixel space (row 1) and in Fourier space (magnitude and phase, rows 2 and 3), and the low- and high-pass filtered images from row 1 (rows 4 and 5).

Figures 28 to 30 present samples comparing a DDPM and EqualSNR diffusion model trained with the exact same hyperparameters and neural architecture, respectively.

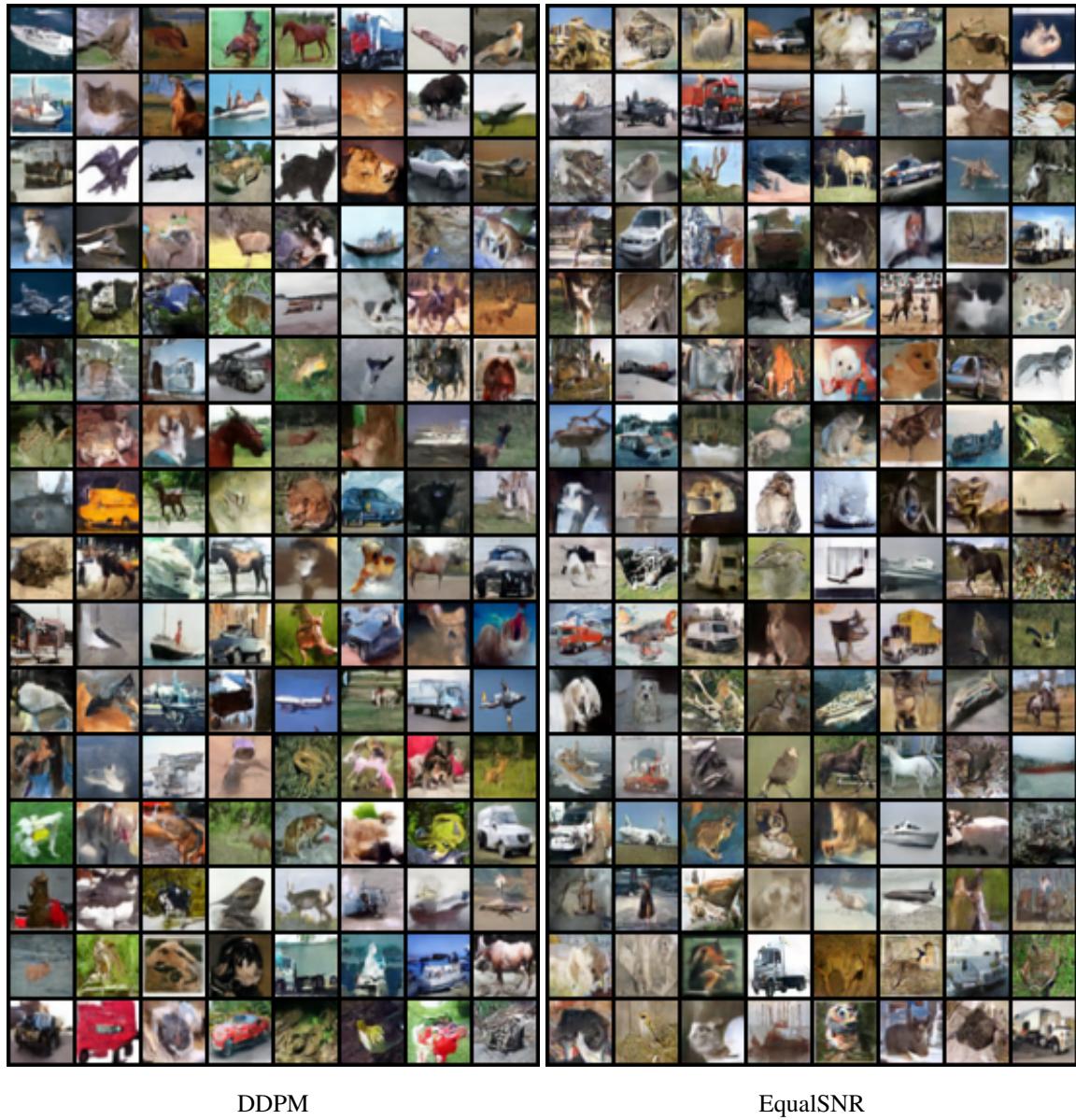


Figure 28: Samples of a diffusion model trained on CIFAR10 with a [left] DDPM and [right] EqualSNR (calibrated) forward process, using  $T = 200$  steps at inference time.



Figure 29: Samples of a diffusion model trained on CelebA with a [left] DDPM and [right] EqualSNR (calibrated) forward process, using  $T = 200$  steps at inference time.



Figure 30: Samples of a diffusion model trained on LSUN Church with a [left] DDPM and [right] EqualSNR (calibrated) forward process, using  $T = 200$  steps at inference time.