Shaping Inductive Bias in Diffusion Models through Frequency-Based Noise Control

Thomas Jiralerspong
Mila - Quebec AI Institute and Université de Montréal, Quebec
thomas.jiralerspong@mila.quebec

Berton Earnshaw, Jason Hartford
Valence Labs
{berton.earnshaw,jason.hartford}@recursion.com

Yoshua Bengio, Luca Scimeca
Mila - Quebec AI Institute and Université de Montréal, Quebec
{yoshua.bengio,luca.scimeca}@mila.quebec

Abstract

Diffusion Probabilistic Models (DPMs) are powerful generative models that have achieved unparalleled success in a number of generative tasks. In this work, we aim to build inductive biases into the training and sampling of diffusion models to better accommodate the target distribution of the data to model. For topologically structured data, we devise a frequency-based noising operator to purposefully manipulate, and set, these inductive biases. We first show that appropriate manipulations of the noising forward process can lead DPMs to focus on particular aspects of the distribution to learn. We show that different datasets necessitate different inductive biases, and that appropriate frequency-based noise control induces increased generative performance compared to standard diffusion. Finally, we demonstrate the possibility of ignoring information at particular frequencies while learning. We show this in an image corruption and recovery task, where we train a DPM to recover the original target distribution after severe noise corruption.

1 Introduction

Diffusion Probabilistic Models (DPMs) have recently emerged as powerful tools for approximating complex data distributions, finding applications across a variety of domains, from image synthesis to probabilistic modeling (Yang et al., 2024; Ho et al., 2020; Sohl-Dickstein et al., 2015; Venkatraman et al., 2024; Sendera et al., 2024). These models operate by gradually transforming data into noise through a defined diffusion process and training a denoising model (Vincent et al., 2008; Alain & Bengio, 2014) to learn to reverse this process, enabling the generation of samples from the desired distribution via appropriate scheduling. Despite their success, the inductive biases inherent in diffusion models remain largely unexplored, particularly in how these biases influence model performance and the types of distributions that can be effectively modeled.

Inductive biases are known to play a crucial role in deep learning models, guiding the learning process by favoring certain types of data representations over others (Geirhos et al., 2019; Bietti & Mairal, 2019; Tishby & Zaslavsky, 2015). A well-studied example is the Frequency Principle (F-principle) or spectral bias, which suggests that neural networks tend to learn low-frequency components of data

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Structured Probabilistic Inference & Generative Modeling (SPIGM).

before high-frequency ones (Xu et al., 2019; Rahaman et al., 2019). Another related phenomenon is what is also known as the simplicity bias, or shortcut learning (Geirhos et al., 2020; Scimeca et al., 2021, 2023b), in which models are observed to preferentially pick up on simple, easy-to-learn, and often spuriously correlated features in the data for prediction. If left implicit, it is often unclear whether these biases will improve or hurt the performance of generative model on downstream task, and they could lead to flawed approximations(Scimeca et al., 2023a). In this work, we aim to explicitly tailor the inductive biases of DPMs to better learn the target distribution of interest.

Recent studies have begun to explore the inductive biases inherent in diffusion models. For instance, Kadkhodaie et al. (2023) analyze how the inductive biases of deep neural networks trained for image denoising contribute to the generalization capabilities of diffusion models. They demonstrate that these biases lead to geometry-adaptive harmonic representations, which play a crucial role in the models' ability to generalize beyond the training data (Kadkhodaie et al., 2023). Similarly, Zhang et al. (2024) investigate the role of inductive and primacy biases in diffusion models, particularly in the context of reward optimization. They propose methods to mitigate overoptimization by aligning the models' inductive biases with desired outcomes (Zhang et al., 2024). Other methods, such as noise schedule adaptations (Sahoo et al., 2024) and the introduction of non-Gaussian noise (Bansal et al., 2022) have shown promise in improving the performance of diffusion models on various tasks. However, the exploration of frequency domain techniques within diffusion models is a relatively new area of interest. One of the pioneering studies in this domain investigates the application of diffusion models to time series data, where frequency domain methods have shown potential for capturing temporal dependencies more effectively (Crabbé et al., 2024). Similarly, the integration of spatial frequency components into the denoising process has been explored for enhancing image generation tasks (Qian et al., 2024; Yuan et al., 2023), showcasing the importance of considering frequency-based techniques as a means of refining the inductive biases of diffusion models.

In this work, we explore a new avenue, to build inductive biases in DPMs by frequency-based noise control. The main hypothesis in this paper is that the noising operator in a diffusion model has a direct influence on the model's representation of the data. Intuitively, the information erased by the noising process is the very information that the denoising model has pressure to learn, so that reconstruction is possible. Accordingly, we propose that by strategically manipulating the noising operation, we can effectively steer the model to learn particular aspects of the data distribution. We focus our attention to the generative learning of topologically structured data, and propose an approach that involves designing a frequency-based noise schedule that selectively emphasizes or de-emphasizes certain frequency components during the noising process. In this paper, we refer to our approach as *frequency diffusion*. Because the Fourier transform of a Gaussian is just another Gaussian in the frequency domain, this approach allows us to maintain the Gaussian assumptions of the diffusion process while reorienting the noising operator within the frequency domain, enabling the generation of Gaussian noise at different frequencies and thereby influencing the model's learning trajectory.

We report several findings. First, we show that when the information content in the data lies more heavily in particular frequencies, frequency diffusion yields better samplers. Furthermore, we test this in several natural datasets, and show that depending on the dataset characteristic, different settings of our frequency diffusion approach yield optimal results, often with comparable or superior performance to standard diffusion. Finally, we show that through frequency-denoising we can recover complex distributions after severe noise corruption at particular frequencies, opening interesting venues for applications within the generative landscape.

We summarize our contributions as follows:

1. We introduce a frequency-informed noising operator that can shape the inductive biases of diffusion models.

2. We empirically show that *frequency diffusion* can steer models to better approximate information at particular frequencies of the underlying data distribution.

3. We provide empirical evidence that models trained with frequency-based noise schedules can outperform traditional diffusion schedules across multiple datasets.

4. We show that through frequency-denoising we can recover complex distributions after severe noise corruption at particular frequencies.

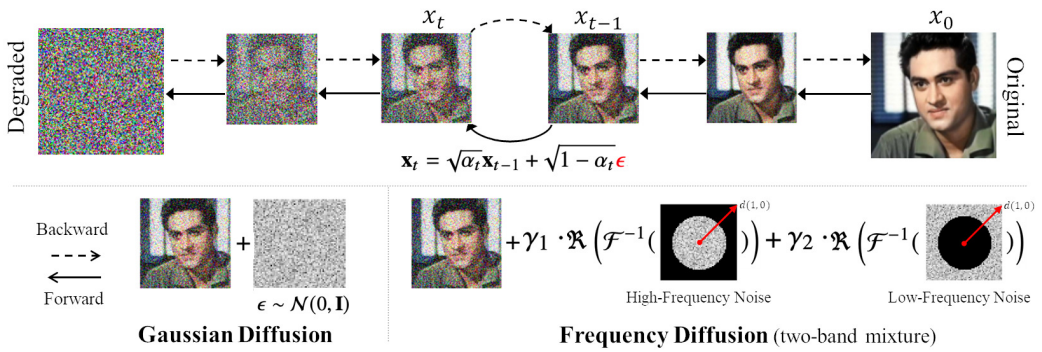

Figure 1: Frequency diffusion under a generalized framework.

2 Methods

2.1 Denoising Probabilistic Models (DPMs)

Denoising Probabilistic Models are a class of generative models that learn to reconstruct complex data distributions by reversing a gradual noising process. DPMs are characterized by a *forward* and *backward* process. The *forward process* defines how data is corrupted, typically by Gaussian noise, over time. Given a data point $\mathbf{x}_0$ sampled from the data distribution $q(\mathbf{x}_0)$, the noisy versions of the data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ are generated according to:

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

with variance schedule α_t. The *reverse process* models the denoising operation, attempting to recover $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \;=\; \mathcal{N}\big(\mathbf{x}_{t-1};\; \mu_\theta(\mathbf{x}_t, t),\; \sigma_t^2\,\mathbf{I}\big), \tag{2}$$

where $\mu_\theta(\mathbf{x}_t, t)$ is predicted by a neural network f_θ, and the variance σ_t^2 can be fixed, learned, or precomputed based on a schedule. We train the denoising model with the standard ϵ–parameterization by minimizing

$$\mathcal{L} \;=\; \mathbb{E}_{t,\mathbf{x}_0,\;\epsilon\sim\mathcal{N}(0,\mathbf{I})}\Big[\big\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\big\|^2\Big], \qquad \mathbf{x}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 \;+\; \sigma_t\,\epsilon, \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\sigma_t^2 = 1 - \bar{\alpha}_t$, ϵ is the Gaussian noise added to $\mathbf{x}_0$, and ϵ_θ is the model's prediction of this noise. To generate new samples, we start from noise and apply the learned reverse process iteratively.

2.2 Frequency Diffusion

The objective of this section is to generate spatial Gaussian noise whose frequency content can be systematically manipulated according to an arbitrary weighting function. From subsection 2.1, $\mathbf{x}_t$ is generated by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to $\mathbf{x}_{t-1}$ via

$$\mathbf{x}_t \;=\; \sqrt{\alpha_t}\,\mathbf{x}_{t-1} \;+\; \sqrt{1-\alpha_t}\,\epsilon. \tag{4}$$

Let us denote by $\mathbf{x} \in \mathbb{R}^{H\times W}$ an image (or noise field) in the spatial domain, and by $\mathcal{F}$ the two-dimensional Fourier transform operator. We let $\mathbf{N}_{\text{freq}} \in \mathbb{C}^{H\times W}$ be a complex-valued random field whose real and imaginary parts are i.i.d. Gaussian:

$$\mathbf{N}_{\text{freq}} \;=\; \mathbf{N}_{\text{real}} \;+\; i\,\mathbf{N}_{\text{imag}}, \quad \mathbf{N}_{\text{real}}, \mathbf{N}_{\text{imag}} \;\sim\; \mathcal{N}\big(0, \mathbf{I}\big), \tag{5}$$

where each pixel (or frequency bin) in $\mathbf{N}_{\text{real}}$ and $\mathbf{N}_{\text{imag}}$ is drawn independently from a standard normal distribution. We introduce a *weighting function* $w(f_x, f_y)$ that scales the amplitude of each frequency component. Let $\mathbf{f} = (f_x, f_y)$ denote coordinates in frequency space, where $f_x = \frac{k_x}{W}$, $f_y = \frac{k_y}{H}$, and k_x, k_y are integer indices (ranging over the width and height), while H and W are the image dimensions. We define the frequency-controlled noise $\mathbf{N}_{\text{freq}}^{(w)}(\mathbf{f})$ as:

$$\mathbf{N}_{\text{freq}}^{(w)}(\mathbf{f}) \;=\; \mathbf{N}_{\text{freq}}(\mathbf{f}) \;\odot\; w(\mathbf{f}), \tag{6}$$

After applying $w(\mathbf{f})$ in the frequency domain, we invert back to the spatial domain to obtain $\epsilon^{(w)}$, our *frequency-shaped* noise:

$$\epsilon^{(w)} \;=\; \Re\Big(\mathcal{F}^{-1}\big(\mathbf{N}_{\text{freq}}^{(w)}\big)\Big), \tag{7}$$

where $\Re(\cdot)$ ensures that our final noise field is purely real.[1]

In summary, any frequency weighting can be represented in this unified framework:

$$\epsilon \xrightarrow{\mathcal{F}} \mathbf{N}_{\text{freq}} \xrightarrow{w(\mathbf{f})} \mathbf{N}_{\text{freq}}^{(w)} \xrightarrow{\mathcal{F}^{-1}} \epsilon^{(w)}.$$

Note that standard white Gaussian noise is a special case of this formulation, where $w(\mathbf{f}) = 1$ for all $\mathbf{f}$. In contrast, more sophisticated weightings allow one to emphasize, de-emphasize, or even remove specific bands of the frequency domain.

Theoretical consistency We keep the forward process Gaussian while reshaping its spectrum. Let $\mathcal{F}$ be the unitary DFT and $w(\mathbf{f}) \geq 0$ a fixed spectral weight. The linear map $\mathbf{T}_w = \mathcal{F}^{-1}\mathrm{Diag}(w)\mathcal{F}$ sends white noise to *frequency-shaped* noise $\boldsymbol{\epsilon}^{(w)} = \mathbf{T}_w\,\boldsymbol{\epsilon}$ with covariance

$$\Sigma_w \;=\; \mathbf{T}_w\,\mathbf{T}_w^{*} \;=\; \mathcal{F}^{-1}\mathrm{Diag}\big(|w(\mathbf{f})|^2\big)\,\mathcal{F}. \tag{8}$$

Replacing $\boldsymbol{\epsilon}$ by $\boldsymbol{\epsilon}^{(w)}$ in the forward step yields the marginal

$$q_w(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\; \sigma_t^2\,\Sigma_w\big). \tag{9}$$

Training with the standard ℓ_2 objective on the added noise remains optimal: $\boldsymbol{\epsilon}_\theta^\star(\mathbf{x}_t, t) = \mathbb{E}[\boldsymbol{\epsilon}^{(w)} \mid \mathbf{x}_t]$. The corresponding score satisfies

$$\nabla_{\mathbf{x}_t} \log q_{w,t}(\mathbf{x}_t) \;=\; -\frac{1}{\sigma_t}\,\Sigma_w^{-1}\,\boldsymbol{\epsilon}_\theta^\star(\mathbf{x}_t, t), \tag{10}$$

so *converting* ϵ-predictions to scores simply multiplies by Σ_w^{-1}. As $t \to 0$, if $\Sigma_w \succ 0$ and q has a locally positive density with $\nabla \log q \in L^1_{\text{loc}}$, the anisotropic Gaussian smoothing collapses to a Dirac and $\nabla \log q_{w,t} \to \nabla \log q$ almost everywhere. Thus, shaping the forward spectrum preserves the *endpoint* score while altering the *path* (see Appendix for proofs and extensions).

2.3 Frequency Noise operators

In this work, the design of $w(\mathbf{f})$ is especially important. We propose a particular choice of $w(\mathbf{f})$ and propose alternative formulations of $w(\mathbf{f})$ in Appendix B.

Band-Pass Masking and Two-Band Mixture

A *band-pass mask* can be viewed as a special case of a more general weighting function:

$$w(\mathbf{f}) \;\in\; \{0, 1\}. \tag{11}$$

In this case, the frequency domain is split into a set of permitted and excluded regions, or radial thresholds. With this, we can construct several types of filters, including a low-pass filter retaining only frequencies below a cutoff (e.g., $\|\mathbf{f}\| \leq \omega_c$), a high-pass filter keeping only frequencies above a cutoff, or more generally a filter restricting $\|\mathbf{f}\|$ to lie between two thresholds $[a, b]$. We thus define a simple band-pass filter as:

$$w(\mathbf{f}) \;=\; \mathbf{M}_{[a,b]}(f_x, f_y) \;=\; \begin{cases} 1, & \text{if } a \leq d(f_x, f_y) \leq b, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

Here, $d(f_x, f_y) = \sqrt{\left(f_x - \frac{1}{2}\right)^2 + \left(f_y - \frac{1}{2}\right)^2}$ measures the radial distance in frequency space. In this special case, $w(\mathbf{f})$ is simply a *binary* mask, selecting only those frequencies within $[a, b]$.

[1] Since the DFT of a real signal has Hermitian symmetry, multiplying by a real, pointwise weight w preserves Hermitian symmetry and yields a real-valued inverse transform.

For the experiments in this paper we formulate a simple two-band mixture, where we limit ourselves to constructing noise as a linear combination of two band-pass filtered components. Specifically, as in the original band-based approach, we generate frequency-filtered noise ϵ_f via:

$$\epsilon_f \;=\; \gamma_l\, \epsilon_{[a_l, b_l]} \;+\; \gamma_h\, \epsilon_{[a_h, b_h]}, \tag{13}$$

where $\gamma_l, \gamma_h \geq 0$ denote the relative contributions of a low- and a high-frequency component ($\gamma_l + \gamma_h = 1$), each filtering noise respectively in the ranges $[a_l, b_l]$ (low-frequency range) and $[a_h, b_h]$ (high-frequency range). We uniquely refer to $\epsilon_{[a,b]}$ as the noise filtered in the $[a, b]$ frequency range following Equation 6 and Equation 7. Standard Gaussian noise emerges as a particular instance (with $\gamma_l = 0.5$, $\gamma_h = 0.5$, $a_l = 0$, $b_l = 0.5$, $a_h = 0.5$, and $b_h = 1$) of this formulation.

Selective omission. If w vanishes on a band, then Σ_w is rank-deficient and the model learns the score *projected* onto range(Σ_w); we exploit this in corruption–recovery, but it cannot recover frequencies never presented during training.

3 Datasets

For the experiments, we consider five datasets, namely: MNIST, CIFAR-10, Domainnet-Quickdraw, Wiki-Art and CelebA; providing examples of widely different visual distributions, scales, and domain-specific statistics. We provide more information in Appendix A

4 Results

All experiments involve separately training and testing DPMs with various *frequency diffusion* schedules, as well as baseline standard denoising diffusion training. We consider five datasets, namely: MNIST, CIFAR-10, Domainnet-Quickdraw, Wiki-Art and CelebA; providing examples of widely different visual distributions, scales, and domain-specific statistics. Unless stated otherwise, all runs use the same α_t schedule and *deterministic* DDIM sampling (η=0), so no step noise is injected at test time. For frequency diffusion we normalize w with C_w to match the per-step variance across settings. This keeps the effective SNR comparable and isolates the effect of the *frequency profile* of the forward noise. Across the experiments, we report FID and KID scores as similarity score estimate metrics of the generated samples with respect to a held-out set of data samples. In all relevant experiments, we compute the metrics on embeddings from block 768 of a pre-trained Inception v3 model.

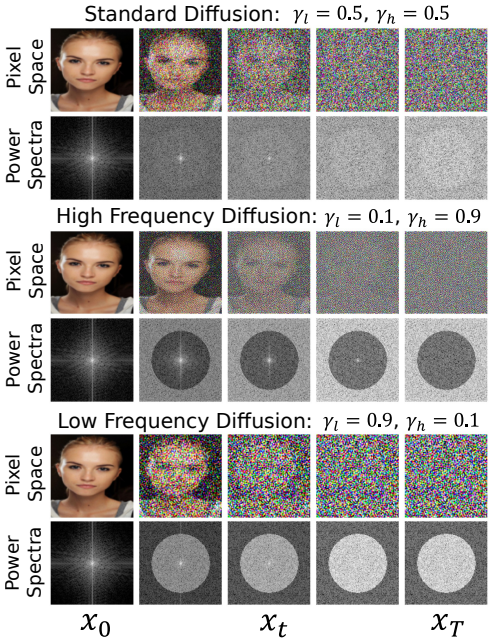

Figure 2: Power spectra and image visuals of the forward Process in standard diffusion, as compared to high and low-frequency noise settings of a two-band mixture noise parametrization.

4.1 Improved Diffusion Sampling via Frequency-Based Noise Control

In the first set of experiments, we wish to test our main hypothesis, i.e. that appropriate manipulation of the frequency components of the noise can better support the learning of the distribution of interest. We follow the formulation in Equation 13 to train and compare diffusion models with a noisy operator prioritizing different parts of the frequency distribution. In these experiments we fix $a_l = 0$, $b_2 = 1$, and $b_l = a_h = 0.5$, while performing a linear sweep of the γ_l and γ_h parameters by searching $\gamma_l \in [.1, .2, ..., .9]$ and $\gamma_h = 1 - \gamma_l$.

4.1.1 Qualitative Overview

First, we show a qualitative example of a standard linear noising schedule forward operation in Figure 2, as compared to two particular settings of our constant high and low-frequency linear schedules of the band-pass filter. With standard noise, information is uniformly removed from the image, with sample quality degrading evenly over time. In the high-frequency noising schedule, sharpness and texture are removed more prominently, while in the low-frequency noising schedule, general shapes and homogeneous pixel clusters are affected most, yielding qualitatively different information destruction operations. As discussed previously, we hypothesize that this will in turn purposely affect the statistics of the information learned by the denoiser model, effectively focusing the diffusion sampling process on different parts of the distribution.

4.1.2 Learning Target Distributions from Frequency-Bounded Information

We conduct experiments to learn the distribution of data where, by construction, the information content lies in the low frequencies. We use the CIFAR-10 dataset, and corrupt the original data with high-frequency noise $\epsilon_{[.3,1.]}$, thus erasing the high-frequency content while predominantly preserving the low-frequency details in the range $\epsilon_{[0.,.3]}$. We train 9 diffusion models, including a standard diffusion (*baseline*) model, and 8 models trained with frequency-based noise control spanning 8 combinations of γ_l ($\gamma_h = 1 - \gamma_l$). We repeat the experiment over three seeds and report the average FID and error in Figure 3. In the figure, we observe the DPMs trained with higher amounts of low-frequency noise (higher γ_l) to perform significantly better than both the baseline ($\gamma_l = 0.5$), and higher frequency denoising models (lower γ_l). Furthermore, we see a mostly monotonically descending trend in FID for increasing values of lower frequency noise in the diffusion forward schedule, supporting the original intuition of how the frequency manipulation of the noising operator can directly steer the denoiser's learning trends, and therefor how progressively higher amount of low-frequency forward noise aid in the learning of samplers for data containing mostly low-frequency information.

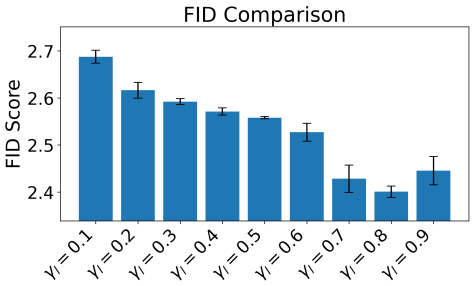

Figure 3: FID of diffusion samplers trained with various combinations of frequency noise. The settings for $\gamma_l = 0.5$ yields standard diffusion training.

4.1.3 Frequency-Based noise control in natural datasets

We further test our hypothesis by training 9 models for each of the datasets considered, inclusive of all γ-variations of our two-band mixture frequency-based noise schedule. We train these models on MNIST, CIFAR-10, Domainnet-Quickdraw, Wiki-Art and CelebA, and report the FID and KID metrics for all ablations in Table 1. In the table, we observe three out of five datasets to significantly benefit from frequency-controlled noising schedules, achieving the lowest FID and KID scores across all tested models. Interestingly, the performance trends are also mostly monotonic, which together with our previous experiments is indicative of where the learned information lies. For simple datasets, such as MNIST or CIFAR-10, most frequency denoising settings perform well, with balanced high-to-low-frequency schedules performing best overall. Denoisers for Domainnet-Quickdraw and CelebA yield better performance for slightly higher frequency noising schedules, suggesting higher frequency information content for good FID and KID approximations, while Wiki-Art shows slight biases towards lower frequency schedules.

4.2 Selective Learning: Frequency-Based Noise Control to Omit Targeted Information

Following our original intuition, a denoising model has pressure to learn the very information that is erased by the forward noising operator to achieve successful reconstruction. Conversely, when the noising operator is crafted to leave parts of the original distribution intact, no such pressure exists, and the denoising model can effectively discard the left-out statistics during generation.

Table 1: Results for FID and KID across different settings of (γ_l, γ_h) for our frequency diffusion two-band mixture schedule across different datasets (mean ± standard error across 3 seeds). The baseline runs correspond to $\gamma_l = \gamma_h = 0.5$.

Dataset →	**MNIST**		**CIFAR-10**		**Domainnet-Quickdraw**		**Wiki-Art**		**CelebA**	
Algo ↓ Metric →	FID (↓)	KID (↓)	FID (↓)	KID (↓)	FID (↓)	KID (↓)	FID (↓)	KID (↓)	FID (↓)	KID (↓)
baseline	0.0168±0.0010	0.0000±0.0000	0.1055±0.0042	0.0001±0.0000	0.0875±0.0060	1.69e-04±1.61e − 05	0.1622±0.0133	2.53e-04±1.80e − 05	0.0863±0.0094	0.0001±0.0000
$\gamma_l = 0.1, \gamma_h = 0.9$	0.2624±0.2184	7.90e-04±6.85e − 04	0.2648±0.0691	4.31e-04±1.30e − 04	0.5250±0.3907	1.46e-03±1.21e − 03	0.2673±0.0273	4.31e-04±4.56e − 05	0.1555±0.0273	2.97e-04±6.93e − 05
$\gamma_l = 0.2, \gamma_h = 0.8$	0.0432±0.0187	1.10e-04±5.24e − 05	0.2191±0.0223	3.86e-04±6.72e − 05	0.1843±0.0723	4.20e-04±2.15e − 04	0.2048±0.0063	3.43e-04±1.27e − 05	0.1024±0.0045	1.85e-04±2.72e − 06
$\gamma_l = 0.3, \gamma_h = 0.7$	0.0267±0.0029	6.40e-05±8.63e − 06	0.1506±0.0168	2.28e-04±3.34e − 05	0.1248±0.0375	2.70e-04±1.13e − 04	0.1865±0.0181	2.86e-04±2.46e − 05	0.0838±0.0107	1.44e-04±1.89e − 05
$\gamma_l = 0.4, \gamma_h = 0.6$	0.0224±0.0032	5.29e-05±8.15e − 06	0.1131±0.0079	1.64e-04±2.15e − 05	0.0799±0.0166	0.0001±0.0000	0.1597±0.0122	2.62e-04±3.23e − 05	0.0875±0.0020	1.49e-04±1.71e − 06
$\gamma_l = 0.6, \gamma_h = 0.4$	0.0253±0.0039	5.81e-05±7.63e − 06	0.1131±0.0074	1.56e-04±1.95e − 05	0.1128±0.0174	2.57e-04±5.56e − 05	0.1348±0.0126	0.0002±0.0000	0.1068±0.0039	2.04e-04±1.07e − 05
$\gamma_l = 0.7, \gamma_h = 0.3$	0.0363±0.0075	9.14e-05±2.04e − 05	0.1432±0.0203	2.19e-04±3.66e − 05	0.1353±0.0223	2.91e-04±6.08e − 05	0.1561±0.0123	2.32e-04±2.46e − 05	0.0990±0.0082	1.84e-04±2.12e − 05
$\gamma_l = 0.8, \gamma_h = 0.2$	0.0512±0.0119	1.36e-04±3.60e − 05	0.1898±0.0095	2.88e-04±1.85e − 05	0.2288±0.0737	5.85e-04±2.21e − 04	0.2256±0.0096	3.86e-04±3.08e − 05	0.1053±0.0185	1.95e-04±4.34e − 05
$\gamma_l = 0.9, \gamma_h = 0.1$	0.3403±0.1513	9.74e-04±4.47e − 04	0.3226±0.0660	5.31e-04±1.20e − 04	0.9827±0.4229	2.84e-03±1.29e − 03	0.3250±0.0270	5.57e-04±3.22e − 05	0.2291±0.0605	4.86e-04±1.52e − 04

Table 2: Resulting FID and KID between standard diffusion and frequency diffusion DPMs trained on noise-corrupted data, with respect to samples from the true uncorrupted distribution (mean ± standard error across 3 seeds). We report eight ablation experiments across different non-overlapping corruption noise schemes.

Dataset →	**Baseline**		**Ours**	
Corruption ↓	FID (↓)	KID (↓)	FID (↓)	KID (↓)
$\epsilon_{[0.1,0.2]}$	3.2273±8.50e − 03	0.0114±3.13e − 05	2.7572±3.56e − 02	0.0095±1.47e − 04
$\epsilon_{[0.2,0.3]}$	3.6601±4.43e − 03	0.0132±1.67e − 05	3.0416±4.47e − 02	0.0107±1.79e − 04
$\epsilon_{[0.3,0.4]}$	3.4771±4.79e − 03	0.0125±1.89e − 05	2.9952±3.35e − 02	0.0106±1.23e − 04
$\epsilon_{[0.4,0.5]}$	3.4281±5.46e − 03	0.0123±1.98e − 05	2.9218±2.54e − 02	0.0105±8.79e − 05
$\epsilon_{[0.5,0.6]}$	3.3638±6.31e − 03	0.0121±2.32e − 05	2.8267±2.81e − 02	0.0102±9.32e − 05
$\epsilon_{[0.6,0.7]}$	3.2444±7.10e − 03	0.0116±2.55e − 05	2.7026±3.90e − 02	0.0097±1.28e − 04
$\epsilon_{[0.7,0.8]}$	3.0442±6.32e − 03	0.0109±2.29e − 05	2.5469±6.39e − 02	0.0091±2.00e − 04
$\epsilon_{[0.8,0.9]}$	3.4660±7.90e − 03	0.0124±2.96e − 05	2.5138±9.63e − 02	0.0090±3.07e − 04

In this section, we perform experiments whereby the original data is corrupted with noise at different frequency ranges. The objective is to manipulate the inductive biases of diffusion denoisers to avoid learning the corruption noise, while correctly approximating the relevant information in the data. We formulate our corruption process as $\mathbf{x}' = A_c(\mathbf{x})$, where:

$$A_c(\mathbf{x}) = \mathbf{x} + \gamma_c \epsilon_{f[a_c, b_c]} \tag{14}$$

Here, $\epsilon_{[a_c,b_c]}$ denotes noise in the $[a_c, b_c]$ frequency range. We default $\gamma_c = 1.$ and show samples of the original and corrupted distributions in Figure 4. For any standard DPM training procedure, the denoiser would make no distinction of which information to learn, and thus would approximate the corrupted distribution presented at training time. As such, the recovery of the original, noiseless, distribution would normally be impossible. Assuming knowledge of the corruption process, we frame the frequency diffusion learning procedures as a noiseless distribution recovery process, and set $a_l = 0$, $b_h = 1$, $b_l = a_c$, and $a_h = b_c$. This formulation effectively allows for the forward frequency noising operator to omit the range of frequencies in which the noise lies. In line with our previous rationale, this would effectively put no pressure on the denoiser to learn the noise part of the distribution at hand, and focus instead on the frequency ranges where the true information lies.

We compare original and corrupted samples from MNIST, as well as samples from standard and frequency diffusion-trained models in Figure 4. In line with our hypothesis, we observe frequency diffusion DPMs trained with an appropriate frequency noise operator to be able to discard the corrupting information and recover the original distribution after severe noisy corruption. We further measure the FID and KID of the samples generated by the baseline and frequency DPMs against the original (uncorrupted) data samples in Table 2. We perform 8 ablation studies, considering noises at 0.1 non-overlapping intervals in the $[0.1, .9]$ frequency range. We observe *frequency diffusion* to outperform standard diffusion training across all tested ranges. Interestingly, we observe better performance (lower FID) for data corruption in the high-frequency ranges, and reduced performance for data corruptions in low-frequency ranges, suggesting a marginally higher information content in the low frequencies for the MNIST dataset.

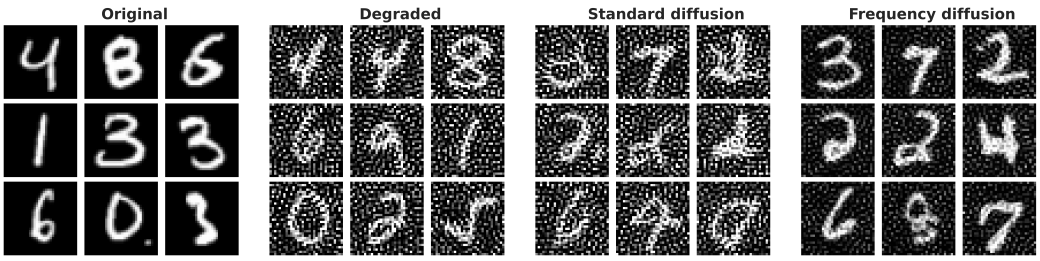

Figure 4: Samples from the original data distribution, the degraded data distribution, a standard diffusion sampler trained on the degraded data distribution, and a *frequency diffusion* sampler trained on the degraded data distribution. We generate noise for data corruption in the frequency range $[a_c = 0.5, b_c = 0.6)]$.

5 Discussion and Conclusion

In this work, we studied the potential to build inductive biases in the training and sampling of Diffusion Probabilistic Models by purposeful manipulation of the forward, noising, process. We introduced *frequency diffusion*, an approach that enables us to guide DPMs toward learning specific statistics of the data distribution. We compare *frequency diffusion* to DPS trained with standard gaussian noise on generative visual tasks set by several datasets, with significant varying structure and scales. We show several key findings. First, we show that appropriate manipulation of the forward noising process can serve as a stong inductive bias for diffusion models to better learn the information of the distribution at particular frequencies. Second, we show that this important characteristic can be readily used when training diffusion models on natural dataset, some of which may be better supported by appropriate frequency diffusion schedules, yielding higher sampling quality. Third, we show how this processes can be used to discard unwanted information at particular frequency ranges, yielding DPMs capable of extract noiseless signals from the remaining ranges.

In our approach, we have limited the results to a simple two-band pass frequency filter. We propose in Appendix B several other alternatives, which may serve as more flexible tools to inject useful inductive biases for similar tasks. Moreover, the approach can be extended beyond constant schedules. For instance, it may prove useful to introduce dynamic frequency noise strategies that shift the focus from low-frequency (general shapes) to high-frequency (sharp edges and textures) components over the time discretization of the sampling process. Such methods could more closely align with human visual processing, which progressively sharpens details over time, offering a more natural sampling process. Additionally, other domains of noise manipulation—outside of the frequency domain may also present new opportunities for further improving DPMs across various tasks.

Finally, a current limitation of this approach lies in the complexity of understanding the relationship between visual data in spatial and frequency domains. The perception of information in the frequency domain does not always translate straightforwardly to visual content, complicating the process of designing optimal noise schedules. As such, it is not trivial to design appropriate frequency schedules for a particular distribution. In practice, empirical validation may still be required to identify the best inductive biases for a given dataset. Future work could focus on refining analytical tools for frequency analysis or exploring alternative inductive bias mechanisms that extend beyond frequency-based manipulations.

Overall, this work opens the door for more targeted and flexible diffusion generative modeling by building inductive biases through the manipulation of the forward nosing process. The ability to design noise schedules that align with specific data characteristics holds promise for advancing the state of the art in generative modeling.

Acknowledgments

The authors acknowledge funding from CIFAR, and Recursion. The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (`https://alliancecan.ca`), Mila (`https://mila.quebec`), and NVIDIA.

References

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise, August 2022. URL `http://arxiv.org/abs/2208.09392`. arXiv:2208.09392 [cs].

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, 2019.

Jonathan Crabbé, Nicolas Huynh, Jan Stanczuk, and Mihaela van der Schaar. Time series diffusion in the frequency domain, 2024. URL `https://arxiv.org/abs/2402.05933`.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL `https://doi.org/10.1038/s42256-020-00257-z`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, June 2020. URL `https://arxiv.org/abs/2006.11239v2`.

Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8911–8920, June 2024.

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 *Proceedings of the 36th International Conference on Machine Learning*, pp. 5301–5310. PMLR, May 2019. URL `https://proceedings.mlr.press/v97/rahaman19a.html`. ISSN: 2640-3498.

Subham Sekhar Sahoo, Aaron Gokaslan, Chris De Sa, and Volodymyr Kuleshov. Diffusion Models With Learned Adaptive Noise, June 2024. URL `http://arxiv.org/abs/2312.13236`. arXiv:2312.13236 [cs].

Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will dnns choose? a study from the parameter-space perspective. In *International Conference on Learning Representations*, 2021.

Luca Scimeca, Alexander Rubinstein, Armand Nicolicioiu, Damien Teney, and Yoshua Bengio. Leveraging diffusion disentangled representations to mitigate shortcuts in underspecified visual tasks. In *NeurIPS 2023 Workshop on Diffusion Models*, 2023a. URL https://openreview.net/forum?id=AvUAVYRA70.

Luca Scimeca, Alexander Rubinstein, Damien Teney, Seong Joon Oh, Armand Mihai Nicolicioiu, and Yoshua Bengio. Shortcut bias mitigation via ensemble diversity using diffusion probabilistic models. *arXiv preprint arXiv:2311.16176*, 2023b.

Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Nikolay Malkin. On diffusion models for amortized inference: Benchmarking and improving stochastic control and sampling. *arXiv preprint arXiv:2402.05098*, 2024.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. URL http://arxiv.org/abs/1503.03585. arXiv:1503.03585 [cond-mat, q-bio, stat].

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain, October 2019. URL http://arxiv.org/abs/1807.01251. arXiv:1807.01251 [cs, math, stat].

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models: A Comprehensive Survey of Methods and Applications, June 2024. URL http://arxiv.org/abs/2209.00796. arXiv:2209.00796 [cs].

Xin Yuan, Linjie Li, Jianfeng Wang, Zhengyuan Yang, Kevin Lin, Zicheng Liu, and Lijuan Wang. Spatial-frequency u-net for denoising diffusion probabilistic models, 2023. URL https://arxiv.org/abs/2307.14648.

Ziyi Zhang, Sen Zhang, Yibing Zhan, Yong Luo, Yonggang Wen, and Dacheng Tao. Confronting reward overoptimization for diffusion models: A perspective of inductive and primacy biases. *arXiv preprint arXiv:2402.08552*, 2024.

A Datasets

For the experiments, we consider five datasets, namely: MNIST, CIFAR-10, Domainnet-Quickdraw, Wiki-Art and CelebA; providing examples of widely different visual distributions, scales, and domain-specific statistics.

MNIST: MNIST consists of $70,000$ grayscale images of handwritten digits (0-9) (Matthey et al., 2017). MNIST provides a simple test-bed to for the hypothesis in this work, as a well-understood dataset with well-structured, and visually coherent samples.

CIFAR-10: CIFAR-10 contains $60,000$ color images distributed across 10 object categories (Krizhevsky et al., 2009). The dataset is highly diverse in terms of object appearance, backgrounds, and colors, with the wide-ranging visual variations across classes like animals, vehicles, and other common objects.

DomainNet-Quickdraw: DomainNet-Quickdraw features $120,750$ sketch-style images, covering 345 object categories (Peng et al., 2019). These images, drawn in a minimalistic, abstract style, present a distribution that is drastically different from natural images, with sparse details and heavy visual simplifications.

WikiArt: WikiArt consists of over $81,000$ images of artwork spanning a wide array of artistic styles, genres, and historical periods (Saleh & Elgammal, 2015). The dataset encompasses a rich and varied distribution of textures, color palettes, and compositions, making it a challenging benchmark for generative models, which must capture both the global structure and fine-grained stylistic variations that exist across different forms of visual art.

CelebA: CelebA contains $202,599$ images of celebrity faces, each 178×218 pixels in resolution (Liu et al., 2015). The dataset presents a diverse distribution of human faces with variations in pose, lighting, and facial expressions.

B Frequency Noise operators

In this section, we propose two alternatives to the design of $w(\mathbf{f})$, which can be considered as viable alternatives for frequency manipulations of the noise.

Power-Law Weighting

We implement a radial power-law frequency noising operator that imposes a *linear slope in the log–log power spectrum*. Let $\mathbf{f} = (f_x, f_y)$ denote normalized frequency coordinates on $[-\frac{1}{2}, \frac{1}{2}]^2$, and define the radial frequency

$$r(\mathbf{f}) \;=\; \sqrt{f_x^2 + f_y^2}.$$

Given white spatial Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \mathbf{I})$, we form its Fourier transform $\mathbf{N}_{\text{freq}} = \mathcal{F}(\mathbf{n})$ and scale each frequency bin by

$$w_\alpha(\mathbf{f}) \;=\; \big(r(\mathbf{f}) + \varepsilon\big)^{\alpha}, \qquad \varepsilon = 10^{-10}, \tag{15}$$

where $\alpha \in \mathbb{R}$ controls the slope and ε prevents the DC singularity. The shaped spectrum and spatial noise are

$$\mathbf{N}^{(\alpha)}_{\text{freq}}(\mathbf{f}) \;=\; \mathbf{N}_{\text{freq}}(\mathbf{f}) \cdot w_\alpha(\mathbf{f}), \qquad \boldsymbol{\epsilon}^{(\alpha)} \;=\; \Re\big(\mathcal{F}^{-1}[\mathbf{N}^{(\alpha)}_{\text{freq}}]\big), \tag{16}$$

which we use in the forward step $\mathbf{x}_t = \sqrt{\alpha_t}\,\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}^{(\alpha)}$.

Effect on spectrum. Because power spectra scale with $|w_\alpha|^2$, the radially averaged power spectral density (RAPSD) obeys

$$\log \mathrm{PSD}(r) \;\approx\; \underbrace{(2\alpha)}_{\text{slope}}\; \log r \;+\; \text{const.}$$

Thus $\alpha > 0$ emphasizes high frequencies (sharper textures), $\alpha < 0$ emphasizes low frequencies (coarser structure), and $\alpha = 0$ recovers white noise.

Discretization and batching. In code, we construct the grid with $f_x(k) = \frac{k}{W} - \frac{1}{2}$ and $f_y(\ell) = \frac{\ell}{H} - \frac{1}{2}$ for $k \in \{0, \dots, W-1\}$, $\ell \in \{0, \dots, H-1\}$ (equivalently, `np.linspace(-0.5,0.5, W)` and `H`). The weight w_α is broadcast across batch (and channels, if present). For convenience, one may multiply in the `fftshift`-centered domain and undo the shift before the inverse FFT; this is equivalent to multiplying in the unshifted domain because w_α is radial.

Optional variance calibration. To keep $\mathbb{E}\|\boldsymbol{\epsilon}^{(\alpha)}\|_2^2$ roughly constant across α, an energy-preserving scalar

$$C_\alpha \;=\; \left(\frac{1}{HW} \sum_{u,v} \left| w_\alpha(f_{uv}) \right|^2 \right)^{-\frac{1}{2}} \tag{17}$$

can be applied in equation 16, i.e., $\mathbf{N}_{\text{freq}}^{(\alpha)} \leftarrow C_\alpha \, \mathbf{N}_{\text{freq}} \cdot w_\alpha$. (Our experiments omit this by default, matching the implementation in the main text.)

Exponential Decay Weighting

Another alternative is an exponential decay function, defined as as:

$$w(\mathbf{f}) \;=\; \exp\!\big(-\beta \, \|\mathbf{f}\|^2\big), \tag{18}$$

where $\beta > 0$, and frequencies with larger norms $\|\mathbf{f}\|$ are exponentially suppressed. This weighting effectively imposes spatial correlations, e.g. for β close to 0 the function induces the retention of more high-frequency components, while for large β, the function quickly damps out high frequencies, resulting in a smoothing of the spatial domain.

C Noise Parameterization, Scores, and Frequency-Shaped Dynamics

In this section, we wish to formalize the role of frequency diffusion in correctly learning the gradient of the log probability density of the data distribution at various noise levels (the score function). We model frequency-shaped corruption as an *anisotropic* Gaussian forward process, derive the score–ϵ relation for this general case, and prove that as $t \to 0$ the learned score converges to the true data score whenever all frequencies are represented (full-rank covariance). We also derive the reverse/posterior formulas and discuss how shaping the forward covariance changes the *path to the score*, shifting the information burden across frequencies. Finally, we formalize the selective-omission case when some bands are removed.

Setup and notation. Let $\alpha_t \in (0, 1)$ be the per-step scaling, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\sigma_t^2 = 1 - \bar{\alpha}_t$. In standard DDPM, the forward marginal is

$$q(\mathbf{x}_t \,|\, \mathbf{x}_0) \;=\; \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0,\; \sigma_t^2 \, \mathbf{I}\big), \tag{19}$$

and one trains an ϵ-predictor $\epsilon_\theta(\mathbf{x}_t, t)$ by minimizing

$$\mathcal{L}_{\text{simple}} \;=\; \mathbb{E}_{t, \mathbf{x}_0,\; \epsilon \sim \mathcal{N}(0, \mathbf{I})} \big[\; \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2 \;\big], \qquad \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sigma_t \, \epsilon. \tag{20}$$

The optimal predictor is $\epsilon_\theta^\star(\mathbf{x}_t, t) = \mathbb{E}[\epsilon \,|\, \mathbf{x}_t]$ and the true score relates to it via

$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) \;=\; -\frac{1}{\sigma_t} \, \epsilon_\star(\mathbf{x}_t, t), \quad \text{where} \quad \epsilon_\star(\mathbf{x}_t, t) = \mathbb{E}[\epsilon \,|\, \mathbf{x}_t]. \tag{21}$$

C.1 Frequency-Shaped Forward Process as Anisotropic Gaussian

Let $w(\mathbf{f}) > 0$ be a (time-independent) radial spectral weight and let $\mathcal{F}$ denote the discrete Fourier transform (unitary). The linear operator

$$\mathbf{T}_w \;:=\; \mathcal{F}^{-1} \circ \operatorname{Diag}\big(w(\mathbf{f})\big) \;\circ\; \mathcal{F} \tag{22}$$

maps spatial white noise to *frequency-shaped* noise. Writing $\xi \sim \mathcal{N}(0, \mathbf{I})$ and $\epsilon^{(w)} = \mathbf{T}_w \, \xi$, we have $\epsilon^{(w)} \sim \mathcal{N}(0,\; \Sigma_w)$ with

$$\Sigma_w \;=\; \mathbf{T}_w \mathbf{T}_w^\top \;=\; \mathcal{F}^{-1} \operatorname{Diag}\big(|w(\mathbf{f})|^2\big)\, \mathcal{F}, \tag{23}$$

i.e., Σ_w is circulant and diagonalized by the Fourier basis, with eigenvalues given by the power spectrum $|w|^2$.[2]

Our forward process uses this shaped noise at each step:

$$\mathbf{x}_t \;=\; \sqrt{\alpha_t}\,\mathbf{x}_{t-1} \;+\; \sqrt{1-\alpha_t}\,\epsilon_t^{(w)}, \qquad \epsilon_t^{(w)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\Sigma_w). \tag{24}$$

A simple induction gives the marginal

$$q_w(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; \mathcal{N}\big(\sqrt{\bar\alpha_t}\,\mathbf{x}_0,\; \sigma_t^2\,\Sigma_w\big). \tag{25}$$

Hence, relative to equation 19, we have replaced the isotropic covariance by Σ_w, while $\bar\alpha_t$ and σ_t^2 remain unchanged.

Support condition. If $w(\mathbf{f}) > 0$ for all $\mathbf{f}$, then $\Sigma_w \succ 0$ (full rank) and the forward kernels have full support in $\mathbb{R}^{HW}$. If w vanishes on a band, Σ_w is singular and the forward kernels are supported on a strict subspace (Section C.5). In practice, adding a small DC floor (e.g., $r(\mathbf{f}) \mapsto r(\mathbf{f}) + \varepsilon$ with $\varepsilon > 0$) ensures $w(0) > 0$ and thus $\Sigma_w \succ 0$.

C.2 Score–ϵ Relation under Anisotropic Covariance

From equation 25,

$$\nabla_{\mathbf{x}_t} \log q_w(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; -\frac{1}{\sigma_t^2}\,\Sigma_w^{-1}\big(\mathbf{x}_t - \sqrt{\bar\alpha_t}\,\mathbf{x}_0\big). \tag{26}$$

Taking the posterior expectation over $q_w(\mathbf{x}_0 \mid \mathbf{x}_t)$ and using $\mathbf{x}_t - \sqrt{\bar\alpha_t}\,\mathbf{x}_0 = \sigma_t\,\epsilon^{(w)}$, we obtain the marginal score

$$\nabla_{\mathbf{x}_t} \log q_{w,t}(\mathbf{x}_t) \;=\; -\frac{1}{\sigma_t}\,\Sigma_w^{-1}\,\underbrace{\mathbb{E}\big[\epsilon^{(w)} \,\big|\, \mathbf{x}_t\big]}_{:=\,\epsilon_\star^{(w)}(\mathbf{x}_t,t)}. \tag{27}$$

Training with the natural generalization of equation 20,

$$\mathcal{L}_{\text{simple}}^{(w)} \;=\; \mathbb{E}_{t,\mathbf{x}_0,\;\epsilon^{(w)}\sim\mathcal{N}(0,\Sigma_w)}\big[\;\|\epsilon^{(w)} - \epsilon_\theta(\mathbf{x}_t,t)\|_2^2\;\big], \qquad \mathbf{x}_t = \sqrt{\bar\alpha_t}\,\mathbf{x}_0 + \sigma_t\,\epsilon^{(w)}, \tag{28}$$

the optimal predictor is $\epsilon_\theta^\star(\mathbf{x}_t,t) = \mathbb{E}[\epsilon^{(w)} \mid \mathbf{x}_t]$. Therefore, a *consistent score estimator* is

$$s_\theta(\mathbf{x}_t,t) \;:=\; \nabla_{\mathbf{x}_t} \log q_{w,t}(\mathbf{x}_t) \;\approx\; -\frac{1}{\sigma_t}\,\Sigma_w^{-1}\,\epsilon_\theta(\mathbf{x}_t,t). \tag{29}$$

Equation equation 29 reduces to equation 21 when $\Sigma_w = \mathbf{I}$. Since the corruption covariance Σ_w is fixed, the ℓ_2 objective needs no reweighting—the optimal ϵ_θ remains the conditional mean; Σ_w^{-1} appears only when converting ϵ_θ to the score via Eq. equation 29.

C.3 Tweedie's Identity and the Limit $t \to 0$

Write the marginal as a (scaled) Gaussian smoothing of the data:

$$q_{w,t}(\mathbf{x}) \;=\; \int q(\mathbf{x}_0)\;\mathcal{N}\big(\mathbf{x}\,;\;\sqrt{\bar\alpha_t}\mathbf{x}_0,\;\sigma_t^2\Sigma_w\big)\;\mathrm{d}\mathbf{x}_0. \tag{30}$$

Let $\mathbf{z}_t := \mathbf{x}_t/\sqrt{\bar\alpha_t}$; then $\mathbf{z}_t = \mathbf{x}_0 + \tilde\sigma_t\,\epsilon^{(w)}$ with $\tilde\sigma_t^2 = \sigma_t^2/\bar\alpha_t$. The *anisotropic Tweedie identity* gives

$$\mathbb{E}\big[\mathbf{x}_0 \,\big|\, \mathbf{z}_t\big] \;=\; \mathbf{z}_t \;+\; \tilde\sigma_t^2\,\Sigma_w\,\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t), \qquad p_t = \mathrm{law}(\mathbf{z}_t). \tag{31}$$

Equivalently, in the original variable,

$$\nabla_{\mathbf{x}_t} \log q_{w,t}(\mathbf{x}_t) \;=\; \frac{\sqrt{\bar\alpha_t}}{\sigma_t^2}\,\Sigma_w^{-1}\Big(\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] - \frac{\mathbf{x}_t}{\sqrt{\bar\alpha_t}}\Big). \tag{32}$$

[2] With the usual Hermitian pairing in the discrete Fourier basis, $\epsilon^{(w)}$ is real-valued.

As $t \to 0$, $\bar{\alpha}_t \to 1$, $\sigma_t \to 0$, and $q_{w,t} \Rightarrow q$. If $\Sigma_w \succ 0$ and q admits a locally positive C^1 density with $\nabla \log q \in L^1_{\text{loc}}$, the anisotropic Gaussian mollifier is an approximate identity and

$$\lim_{t \to 0} \nabla_{\mathbf{x}_t} \log q_{w,t}(\mathbf{x}_t) \;=\; \nabla_{\mathbf{x}} \log q(\mathbf{x}) \quad \text{for a.e. } \mathbf{x}. \tag{33}$$

Intuitively, the anisotropic Gaussian kernel in equation 30 shrinks to a Dirac as $\sigma_t \to 0$ regardless of its orientation, so the smoothed score converges to the true data score. Combining equation 27–equation 33, the ϵ-parameterization with frequency-shaped noise yields a *correct score at* $t = 0$, provided $\Sigma_w \succ 0$.

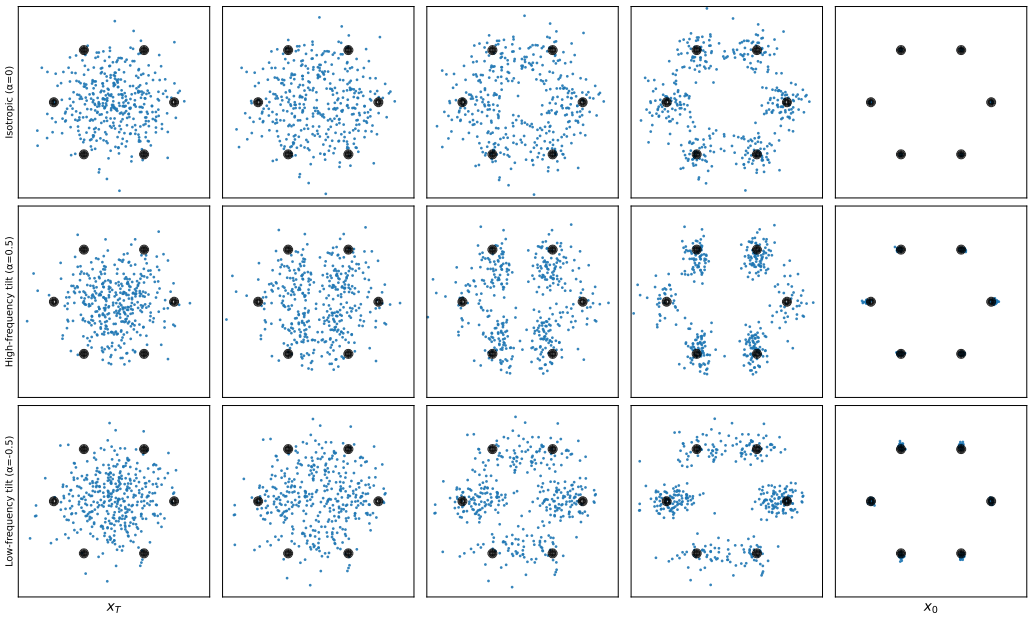

Figure C.1: Particle trajectories under the probability–flow ODE from a Gaussian prior to a mixture-of-Gaussians target (black contours), visualized at five equally spaced times (left to right). Rows: (top) isotropic noise (α=0), (middle) high-frequency tilt (α=0.1), (bottom) low-frequency tilt (α=−0.1). Frequency noising alters the path by reweighting modes via Σ_w while keeping the endpoint consistent under full support (cf. Sec. C).

To visualize how frequency noising alters trajectories and score geometry through time, Fig. C.1 shows particle flows under the probability–flow ODE for isotropic noise (top), high-frequency tilt (α=0.1, middle), and low-frequency tilt (α=−0.1, bottom), while Fig. C.2 shows the corresponding score fields $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ at five equally spaced times. Frequency noising changes the path deterministically by reweighting modes through Σ_w, while preserving the $t \to 0$ endpoint score under full support (see Sec. C).

C.4 Reverse/Posterior with Frequency-Shaped Noise

Since all covariances are proportional to the same Σ_w, linear-Gaussian posteriors retain the standard *scalar* coefficients while the covariances inherit Σ_w as a factor. In particular,

$$q_w(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \;=\; \mathcal{N}\Big(\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0),\; \tilde{\beta}_t\, \Sigma_w\Big), \tag{34}$$

with

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\,(1 - \alpha_t), \tag{35}$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t \;-\; \frac{1 - \alpha_t}{1 - \bar{\alpha}_t} \big(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 \big) \right). \tag{36}$$

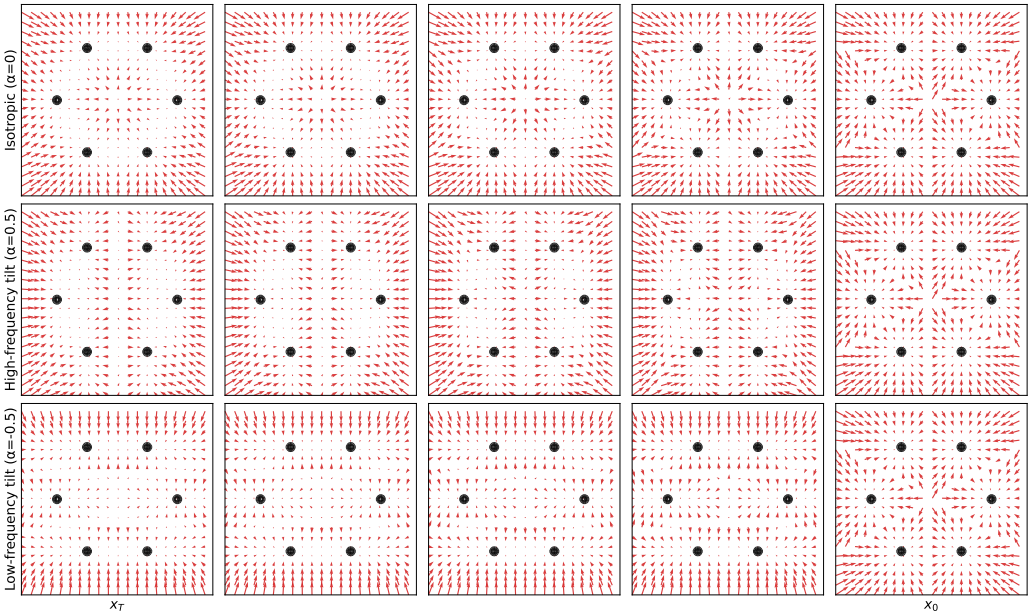

Figure C.2: Evolving score fields $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ for the same three settings as Fig. C.1. Arrows indicate the instantaneous score on a grid; black contours show the target density. Anisotropic shaping stretches/compresses the field along principal modes, biasing the trajectory toward frequencies emphasized by Σ_w.

Replacing $\mathbf{x}_0$ by $\hat{\mathbf{x}}_0$ in equation 36 yields the usual mean update. For the ϵ-parameterization we recover an estimate of $\mathbf{x}_0$ via

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) \;=\; \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sigma_t\, \epsilon_\theta(\mathbf{x}_t, t)\right). \tag{37}$$

Stochastic DDPM sampling: if one samples stochastically, the injected noise should be drawn as $\eta_t^{(w)} \sim \mathcal{N}(0, \Sigma_w)$ (not $\mathcal{N}(0, \mathbf{I})$) for consistency with the forward process.

DDIM / probability-flow ODE: if one uses the deterministic sampler (i.e., $\eta = 0$), *no* step noise is injected. In continuous time, the associated probability-flow ODE with frequency-shaped forward noise reads

$$\frac{d\mathbf{x}}{dt} \;=\; -\tfrac{1}{2}\,\beta(t)\,\mathbf{x} \;-\; \tfrac{1}{2}\,\beta(t)\,\Sigma_w\,\nabla_{\mathbf{x}} \log p_t(\mathbf{x}), \tag{38}$$

which reduces to the standard probability–flow ODE when $\Sigma_w = \mathbf{I}$. In practice with the ϵ-parameterization, one uses $\hat{\mathbf{x}}_0$ from equation 37 in the standard DDIM deterministic update; no extra noise term appears.

C.5 Selective Omission and Rank-Deficient Σ_w

If w vanishes on a measurable band, then $\Sigma_w \succeq 0$ is singular. The forward kernels in equation 25 are supported on an affine subspace determined by range(Σ_w), and the smoothed marginals $q_{w,t}$ are not strictly positive in $\mathbb{R}^{HW}$. The score $\nabla \log q_{w,t}$ exists only on that subspace and is undefined along the null space. Training with equation 28 then recovers the *projected* score, i.e., the model learns to ignore the omitted bands by construction (this is the mechanism exploited in our corruption-recovery experiments).

C.6 How Frequency Noising Changes the "Path to the Score"

Even though the $t \to 0$ limit recovers the true data score under $\Sigma_w \succ 0$, the *evolution* of the score with t changes substantially:

1. **Geometry of the score.** From equation 27, the conversion from ϵ-prediction to score multiplies by Σ_w^{-1}. In the Fourier basis (where Σ_w is diagonal), modes with larger variance (large $|w|^2$) are *downweighted* in the score, while low-variance modes are *amplified*. Thus, shaping the forward spectrum changes the relative gradient magnitudes across frequencies during training and sampling.
2. **Signal-to-noise during supervision.** The target $\epsilon^{(w)}$ has covariance Σ_w, so its per-mode variance follows $|w|^2$. The ℓ_2 loss in equation 28 therefore exposes the model to larger target amplitudes (and larger gradients) in bands where $|w|$ is large, shifting the inductive bias toward fitting those modes sooner/more accurately.
3. **Reverse dynamics.** The reverse posterior covariance in equation 34 is $\tilde{\beta}_t \Sigma_w$, so the stochasticity injected at each reverse step is anisotropic. This changes the trajectory taken from t down to 0, biasing the generation process to consolidate structure along directions favored by Σ_w. Under DDIM ($\eta = 0$), no step noise is instead injected, so the anisotropic *stochastic* effect disappears. However, the *drift* in the probability-flow ODE equation 38 still carries Σ_w through the term $-\beta(t)\,\Sigma_w\,\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. Hence trajectories remain frequency-biased deterministically: modes emphasized by Σ_w contribute more strongly to the drift, reshaping the path from t=T to t=0 even without randomness.

Collectively, these effects explain why different datasets benefit from different w: the *endpoint* score is consistent (under full rank), but the *path*—and thus the optimization landscape and sample trajectories—is reshaped by frequency weighting.

Time-varying weights. If one uses a schedule $w_t(\mathbf{f})$, the t-step marginal covariance becomes a scalar-weighted sum of commuting matrices:

$$\mathrm{Cov}(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; \sum_{s=1}^{t} \Big(\beta_s \prod_{k=s+1}^{t} \alpha_k\Big)\, \Sigma_{w_s}, \qquad \beta_s := 1 - \alpha_s. \tag{39}$$

When all Σ_{w_s} are diagonal in the Fourier basis (true for any per-frequency diagonal weight, not necessarily radial), the analysis carries through modewise with eigenvalues replaced by the corresponding *positive weighted sums* $\sum_s w_s\,|w_s(\mathbf{f})|^2$ (which form a convex combination after normalization by $\sigma_t^2 = \sum_s w_s$).