Frequency Domain Diffusion Model with Scale-Dependent Noise Schedule

Amir Ziashahabi*, Baturalp Buyukates*, Artan Sheshmani†‡, Yi-Zhuang You§, and Salman Avestimehr*
*Dept. of Electrical and Computer Engineering, University of Southern California, {ziashaha, buyukate, avestime}@usc.edu
†Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts Institute of Technology, artan@mit.edu
‡Beijing Institute of Mathematical Sciences and Applications
§Dept. of Physics, University of California San Diego, yzyou@physics.ucsd.edu

***Abstract*—Diffusion models have played a crucial role in the recent advancements in generative image modeling. These models are characterized by a forward process that incrementally corrupts images. The modeling objective is to develop a reverse process capable of reconstructing the original image from degraded inputs so that the trained model can then be leveraged to generate natural images from pure noise. In this work, we introduce a novel diffusion process that operates in the frequency domain. Typically, the frequency domain representation of an image exhibits a sparse structure, with energy predominantly concentrated in low frequency components. This inherent sparsity aids us in the effective separation of signal and noise during the reverse process. We utilize this property to introduce a scale-dependent noise schedule, offering precise control over various image scales. Working in the frequency domain allows us to modify the training protocol, resulting in significant computation enhancements, achieving a speedup of 2.7-8.5$\times$ without a significant drop in generated image quality, compared to the image domain models, which operate with fixed noise schedules. Source code is available at https://github.com/Amir-zsh/FDDM.**

I. Introduction

Generative image modeling has demonstrated impressive performance in numerous areas, including image generation [1]–[4], super-resolution [5], [6], inpainting [6]–[8], image colorization [6], and medical imaging [9], [10]. Diffusion models [1], [11] are a more recent approach to generative image modeling that have taken over earlier methods, including generative adversarial networks (GANs) [12] and variational autoencoders (VAEs) [13], by offering superior image quality [14]. Inspired by non-equilibrium thermodynamics [11], diffusion models are comprised of two processes: (1) the forward process, which is a Markovian process that gradually adds noise to (corrupts) data over a specific number of steps, and (2) the reverse process which progressively restores natural data points from noisy inputs. The forward process is simple and tractable, and training is performed to fit the reverse process, resulting in a powerful generative model. In this work, we propose the Frequency Domain Diffusion Model (FDDM).

FDDM is a novel diffusion-based approach for image generation that maps data between image and frequency domains using the discrete cosine transform (DCT) [15]. Unlike the existing works that operate in the image domain, FDDM performs diffusion-based generative modeling in the frequency domain. This model is motivated by the fact that many natural images have a sparse representation in the frequency domain, which makes it easier to separate signal from noise.

During the forward process, we add noise to the image in the frequency domain, with the diffusion coefficient being determined by a novel scale-dependent noise schedule. We define this schedule using a dispersion relation, which describes the energy associated with each frequency component. The dispersion relation determines how quickly information diffuses through the image. The sparsity in the frequency domain means that most of the energy in an image is concentrated in a small number of frequency components, while the remaining components have very low energy (see Fig. 1). With this sparsity, we separate signal from noise more effectively than in image space. This is because the noise is spread out across all frequency components, while the signal is concentrated in a small number of components (unlike in the image domain).

Another advantage of the FDDM is that it can handle both large-scale (low frequency) and small-scale (high frequency) features (components) in an image. By separating these in the frequency domain, we apply different diffusion coefficients to each scale. We then leverage ideas from JPEG encoding [16]–[18] and apply FDDM on patches of images (see Fig. 2), significantly increasing the training/inference speed (compared to image domain diffusion), making FDDM suitable for time-critical applications such as medical imaging where there is a need for rapid image generation [19].

Our contributions are two-fold: (1) We introduce a novel diffusion model, FDDM, that operates in the frequency domain using a scale-dependent noise schedule. (2) We combine our process with ideas from JPEG encoding and frequency domain learning, improving speed in both training and inference, without a significant drop in generated image quality.

II. Background and Related works

A diffusion model gradually introduces random noise into data using a sequence of Markov chain steps and is then trained to reverse the process for image generation. Let x_0 be the data and $x_{1:T}$ be a sequence of noisy samples. The forward diffusion is defined [1] by a Markov process

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t;\ \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbb{I}), \tag{1}$$

where $\beta_{1:T}$ denote the noise variance schedule. Under reparameterization, with $\epsilon_{t|t-1} \sim \mathcal{N}(0, \mathbb{I})$ we obtain

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t|t-1}. \tag{2}$$

 DOI: 10.1109/ISIT57864.2024.10619452

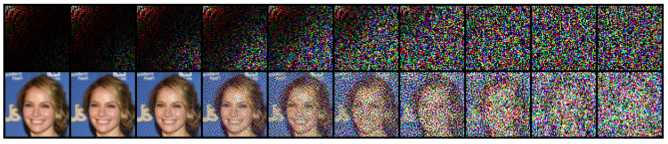

Fig. 1: Scale-dependent noise schedule in the frequency domain (top) and the corresponding image domain representation, without patching, i.e., the whole image is a single patch. The frequency representation of the original (leftmost) image exhibits a sparsity, which we leverage in the proposed FDDM.

The diffusion chain is generated auto-regressively such that

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}). \tag{3}$$

In fact, marginalization to any time step is tractable

$$q(x_t \mid x_0) = \mathcal{N}(x_t;\, \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbb{I}),$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. When conditioned on x_0, the reverse diffusion is defined by the Markov process

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1};\, \bar{\mu}(x_t, x_0), \bar{\beta}_t\mathbb{I}), \tag{4}$$

where we have

$$\bar{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0, \quad \bar{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{5}$$

The landmark study [1] demonstrated the capability of diffusion models to generate high-quality images, sparking significant interest in this area. Studies pertinent to our work can be grouped into two main categories: (1) studies aiming to provide alternative diffusion (corruption) processes to the standard Gaussian process [20]–[23] and (2) studies that focus on enhancing the performance of diffusion models [24]–[29]. This work lies at the intersection of the two, as we achieve performance improvements under the proposed FDDM.

III. Proposed Frequency Domain Diffusion Model

We denote the data in the image space as $\boldsymbol{x}$ and in the frequency space as $\tilde{\boldsymbol{x}}$ such that $\tilde{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{x}) \Leftrightarrow \boldsymbol{x} = \mathcal{F}^{-1}(\tilde{\boldsymbol{x}})$. The inverse frequency transformation $\mathcal{F}^{-1}$ maps the data back to its original domain. In this work, we use the discrete cosine transformation (DCT) [15] to ensure that real data maps to real features. DCT is similar to the Fourier transformation, using real-valued cosine functions instead of complex exponentials. It is more suitable for processing real-valued data like images.

Large-scale (small-scale) features correspond to low-frequency (high-frequency) components in the frequency domain. By leveraging this separation, we apply different diffusion coefficients to each scale to improve the denoising performance. In particular, we use momentum coordinates in the frequency domain and label the components of the transformed data by their momentum $\boldsymbol{k}$, defined as $\boldsymbol{k} = (k_1, k_2)$, where k_1 and k_2 are wave numbers that coordinate the frequency domain. Then we apply different diffusion coefficients based on the location in the frequency domain.

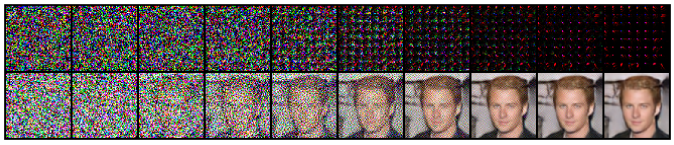

Fig. 2: Image denoising and corresponding frequency domain representation for a 8×8 patched image. FDDM generates new samples by gradually denoising in the frequency domain.

A. Forward Diffusion in the Frequency Domain

Unlike in the image domain, as in (2), the forward diffusion process in FDDM gradually introduces noise from ultraviolet (UV), i.e., high frequency, to infrared (IR), i.e., low frequency, in the frequency domain (see Fig. 1), using a stochastic process $\tilde{\boldsymbol{x}}_0 \to \tilde{\boldsymbol{x}}_1 \to \cdots \to \tilde{\boldsymbol{x}}_t \to \cdots$ defined by

$$\tilde{\boldsymbol{x}}_{t+1} = \sqrt{1-\boldsymbol{\beta}_t} \odot \tilde{\boldsymbol{x}}_t + \sqrt{\boldsymbol{\beta}_t} \odot \boldsymbol{z}_t,$$

where we have $\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, I)$ and $\boldsymbol{\beta}_t$ is a scale-dependent diffusion coefficient that controls how much noise we introduce at each step. The symbol $\odot$ denotes element-wise multiplication. Alternatively, the same process can be defined using an $\bar{\boldsymbol{\alpha}}_t$ parameter that controls the SNR at each step:

$$\tilde{\boldsymbol{x}}_t = \sqrt{\bar{\boldsymbol{\alpha}}_t} \odot \tilde{\boldsymbol{x}}_0 + \sqrt{1-\bar{\boldsymbol{\alpha}}_t} \odot \boldsymbol{z}_t, \tag{6}$$

where we have $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, I)$. We specify the design of $\bar{\boldsymbol{\alpha}}_t$ in Section III-D, where we discuss the proposed scale-dependent noise schedule. The key idea of this noise schedule is to gradually add correlated noise to the image from small-scale to large-scale in the forward process. By controlling the diffusion coefficient, i.e., SNR, based on locations in the frequency space, we essentially apply different amounts of smoothing to different scales and locations in the frequency domain.

B. Backward Diffusion in the Frequency Domain

The backward diffusion process in FDDM learns to generate data from IR to UV in the frequency domain, using a noise prediction model $\boldsymbol{\phi}_\theta$ that takes the noisy frequency features $\tilde{\boldsymbol{x}}_t$ and predicts the clean image in the frequency space $\tilde{\boldsymbol{x}}_0$ such that $\boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t) \to \tilde{\boldsymbol{x}}_0$. By utilizing the predicted clean image, we can perform the denoising operation. Following [27], we define the following relationship between $\tilde{\boldsymbol{x}}_{t-1}$ and $\tilde{\boldsymbol{x}}_t$

$$\tilde{\boldsymbol{x}}_{t-1} = \underbrace{\sqrt{\bar{\boldsymbol{\alpha}}_{t-1}} \odot \boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t) + \sqrt{1-\bar{\boldsymbol{\alpha}}_{t-1} - \boldsymbol{\sigma}_t(\eta)^2} \odot \hat{\boldsymbol{z}}}_{\text{deterministic part}} + \underbrace{\boldsymbol{\sigma}_t(\eta) \odot \boldsymbol{z}_t}_{\text{stochastic part}} \tag{7}$$

with $\boldsymbol{\sigma}_t(\eta) = \eta\sqrt{\frac{1-\bar{\boldsymbol{\alpha}}_{t-1}}{1-\bar{\boldsymbol{\alpha}}_t}} \odot \sqrt{1-\frac{\bar{\boldsymbol{\alpha}}_t}{\bar{\boldsymbol{\alpha}}_{t-1}}}$. The predicted noise is

$$\hat{\boldsymbol{z}} = \frac{\tilde{\boldsymbol{x}}_t - \boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t) \odot \sqrt{\bar{\boldsymbol{\alpha}}_t}}{\sqrt{1-\bar{\boldsymbol{\alpha}}_t}}. \tag{8}$$

The backward diffusion process learns to generate cleaner data from noisy data in the frequency space by predicting the noise configuration and using it to recover the clean signal (see Fig. 2). As in the forward process, in the backward process, we apply different amounts of smoothing to different scales and

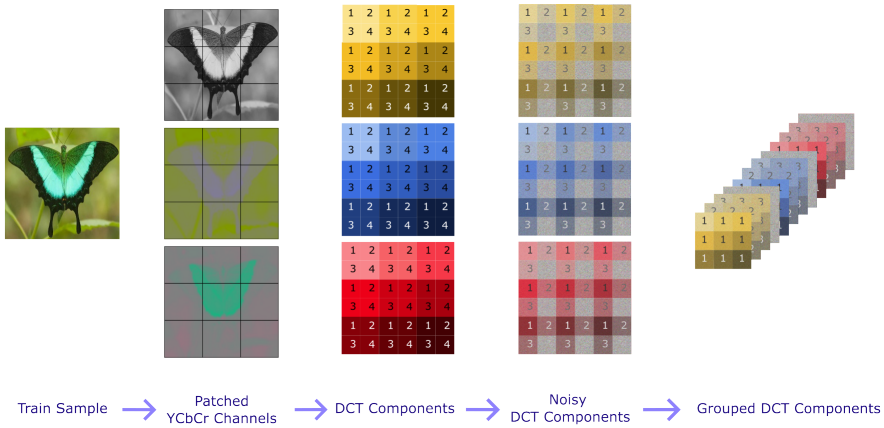

Fig. 3: The proposed frequency-based forward process. We split YCbCr input image channels into patches and take the DCT of each patch. Then, we add the scale-dependent noise to each patch in the frequency domain. Finally, we group the DCT components into output channels, with each channel containing components of the same frequency. We feed these output channels, i.e., grouped DCT components to the diffusion model during training.

locations in the frequency space by controlling the diffusion coefficient (or SNR) based on momentum coordinates.

C. Objective Function and Training Approach

The objective function for training the noise prediction model ϕ_θ in FDDM is

$$\mathcal{L}_\theta = \mathop{\mathbb{E}}_{\boldsymbol{x}_0\in\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},I)} \mathop{\mathbb{E}}_{t\sim\mathcal{U}(1,\dots,T)} \|(\tilde{\boldsymbol{x}}_0 - \boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t))\|^2, \quad (9)$$

where $\mathcal{D}$ is the training dataset, $\boldsymbol{x}_0$ is the image drawn from the training set, and $\tilde{\boldsymbol{x}}_t$ is the noisy frequency data obtained through the forward diffusion described by (6) given a noise configuration $\boldsymbol{z}$. The objective function measures the mean squared error between the DCT components of the predicted clean image $\boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t)$ and those of the true clean image $\tilde{\boldsymbol{x}}_0$ over all training images. So, the loss in (9) is computed in the frequency domain. Similarly, in FDDM, the forward and backward processes take place in the frequency domain.

We use Adam optimizer [30] to minimize the loss function in (9) by sampling a mini-batch of training images from $\mathcal{D}$ and computing the gradients with respect to the parameters of the noise prediction model. Adam updates the parameters using a learning rate and adjusts the rate for each parameter based on the first and second moments of the gradients, improving the optimization's convergence properties. By updating the parameters iteratively over many epochs, we learn a noise prediction model that denoises images in the frequency domain.

D. Proposed Scale-Dependent Noise Schedule

The scale-dependent noise schedule in FDDM is a momentum-dependent function that controls the amount of noise introduced at each step of the forward process. We define it using a dispersion relation $\epsilon_{\boldsymbol{k}}$, which gives the energy associated with a frequency mode of momentum $\boldsymbol{k}$. We use a tight-binding dispersion such that $\epsilon_{\boldsymbol{k}} = -\cos\pi k_1 - \cos\pi k_2$. In particular, the scale-dependent noise schedule is given by

$$\bar{\alpha}_{t,\boldsymbol{k}} = \frac{1}{\exp\left(\frac{\epsilon_{\boldsymbol{k}} - \mu_t}{T'}\right) + 1}, \quad (10)$$

where μ_t is the Fermi level at time t, which controls the overall SNR. The noise schedule varies with momentum $\boldsymbol{k}$, and can be packed into a vector $\bar{\boldsymbol{\alpha}}_t = [\bar{\alpha}_{t,\boldsymbol{k}}]_{\boldsymbol{k}\in\mathsf{BZ}}$, where BZ denotes the Brillouin zone. The Fermi level μ_t is expected to decrease with diffusion time t, such that $\bar{\boldsymbol{\alpha}}_t$ decreases from 1 to 0 as more noise is introduced. This allows for a gradual introduction of noise from UV to IR in the frequency space. The temperature parameter T' controls how sharp or smooth this transition is, with lower values of T' leading to sharper transitions. With this scale-dependent noise schedule, we essentially apply different amounts of smoothing to different scales and locations in the frequency space. This allows for effective denoising while preserving important features in images.

IV. Description of FDDM

FDDM consists of two algorithms: frequency-based forward process and sampling, which are given in Algorithms 1 and 2, respectively. Frequency-based forward process includes pre-processing, forward process, and training steps.

Frequency-Based Forward Process. We first sample an original image from a set $\mathcal{D}$ of training images. This sample image is initially in the RGB format. Following the standard practice of JPEG encoding [17], we first convert it to YCbCr format and split it into patches to obtain enhanced training and inference speed. We note that this patch-based forward process is enabled by the FDDM, as it performs diffusion in the frequency domain. Next, we take the DCT of each patch separately. We then sample a timestep uniformly for the image and corrupt this image based on the forward process defined in (6). In particular, we apply the scale-dependent noise to each patch independently. Once the noise applied, we obtain the noisy DCT components for each patch. Final step of the pre-processing is to group the components from the same frequency across patches into separate channels. In our design, this process is performed for a batch of images and we feed the associated uniformly sampled timestep of each image of the batch, i.e., noise level, to the neural network, together with the corrupted (and grouped) DCT components, i.e., channels. Finally, we take a gradient descent step using the

L2 loss between the actual input image in frequency domain and the predicted clean image in the frequency domain. This entire patchifying and forward process pipeline is demonstrated in Fig. 3 and Algorithm 1 (which does not explicitly state grouping of the noisy DCT components for ease of exposition).

We let n denote the size of the images, i.e., input images are of size $n \times n$. Assume the patching operation is performed with a patch size of $d \times d$. Then, the resulting patchified image has a total of $P = \left(\frac{n}{d}\right)^2$ patches. We note that this operation is repeated for each input channel (YCbCr images have 3 channels). In the example in Fig. 3, the image size is $n = 6$ and the patch size is $d = 2$. Thus, we have a total of $P = 9$ patches in each image channel and the resulting tensor is of size $12 \times 3 \times 3$, where 12 is the number of output channels and 3×3 is the size of each output channel. We note that in the standard JPEG convention [16], [17], patches are of size 8×8. Also, we note that the patch size directly determines the number of DCT components we have in each patch which is $d^2 = 4$, as also demonstrated in Fig. 3.

Algorithm 1 Frequency-Based Forward Process

Require: Input distribution D, # of training timesteps T, patch size d, image size n

$P \leftarrow (n/d)^2$ ▷ Total number of patches
repeat
  $\boldsymbol{x}_0 \sim \mathcal{D}$
  $t \sim \mathcal{U}(1, \dots, T)$
  Split $\boldsymbol{x}_0$ into $d \times d$ patches $\{\boldsymbol{x}_0^p\}_{p \in \{1,\dots,P\}}$
  for $p \in \{1, \dots, P\}$ **do**
    $\boldsymbol{z} \leftarrow \mathcal{N}(\mathbf{0}, \boldsymbol{I_d})$
    $\tilde{\boldsymbol{x}}_0^p \leftarrow \mathcal{F}(\boldsymbol{x}_0^p)$
    $\tilde{\boldsymbol{x}}_t^p \leftarrow \sqrt{\bar{\boldsymbol{\alpha}}_t} \odot \tilde{\boldsymbol{x}}_0^p + \sqrt{1 - \bar{\boldsymbol{\alpha}}_t} \odot \boldsymbol{z}$
  end for
  $\tilde{\boldsymbol{x}}_0 \leftarrow [\tilde{\boldsymbol{x}}_0^0, \tilde{\boldsymbol{x}}_0^1, \dots, \tilde{\boldsymbol{x}}_0^P]$ ▷ Concatenate patches
  $\tilde{\boldsymbol{x}}_t \leftarrow [\tilde{\boldsymbol{x}}_t^0, \tilde{\boldsymbol{x}}_t^1, \dots, \tilde{\boldsymbol{x}}_t^P]$
  Take gradient descent step on $\nabla_\theta \|\tilde{\boldsymbol{x}}_0 - \boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_t, t)\|^2$
until converged

Sampling (Inference). Sampling is essentially the backward process (denoising). We start by sampling pure noise in the frequency domain and gradually denoise it using the trained model, following (7). Algorithm 2 shows the denoising operation, which is also shown in Fig. 2. We note that, as in the forward process, we work on patches in the backward process. That is, we perform the denoising operation described by (7) on the grouped DCT components, which are omitted in Algorithm 2 for ease of exposition. In particular, using the model prediction at timestep t_i, $\boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_{t_i}, t_i)$, we recover the less noisy DCT components from the previous timestep t_{i-1}, and these steps are repeated until timestep t_1. Once the denoising is completed, we take the inverse DCT of the patches, combine the results (unpatch the image), and return the generated image. Here, t_i denotes the inference time steps for $i \in \{1, \dots, I\}$, with I being the total number of inference steps. Unlike the forward process, where the timesteps are consecutive, in the backward process, we can sparsely sample. That is, t_i and t_{i-1} are not necessarily consecutive for any $i \in \{1, \dots, I\}$. Ideally, the resulting generated image should look as if it is from the original dataset.

Algorithm 2 Sampling (Inference)

Require: Inference time steps $\{t_i\}_{i \in \{1,\dots,I\}}$, total # of inference steps I, patch size d, image size n, η

$P \leftarrow (n/d)^2$
$\{\tilde{\boldsymbol{x}}_0^p \leftarrow \mathcal{N}(\mathbf{0}, \boldsymbol{I_d})\}_{p \in \{1,\dots,P\}}$
$\tilde{\boldsymbol{x}}_{t_I} \leftarrow [\tilde{\boldsymbol{x}}_{t_I}^0, \tilde{\boldsymbol{x}}_{t_I}^1, \dots, \tilde{\boldsymbol{x}}_{t_I}^P]$
for $i = I$ **downto** 1 **do**
  $[\tilde{\boldsymbol{x}}_0^0, \tilde{\boldsymbol{x}}_0^1, \dots, \tilde{\boldsymbol{x}}_0^P] \leftarrow \boldsymbol{\phi}_\theta(\tilde{\boldsymbol{x}}_{t_i}, t_i)$
  if $i > 1$ **then**
    for $p \in \{1, \dots, P\}$ **do**
      $\boldsymbol{z} \leftarrow \mathcal{N}(\mathbf{0}, \boldsymbol{I_d})$
      $\boldsymbol{\sigma}_{t_i}(\eta) = \eta \sqrt{\frac{1 - \bar{\boldsymbol{\alpha}}_{t_{i-1}}}{1 - \bar{\boldsymbol{\alpha}}_{t_i}}} \odot \sqrt{1 - \frac{\bar{\boldsymbol{\alpha}}_{t_i}}{\bar{\boldsymbol{\alpha}}_{t_{i-1}}}}$
      $\hat{\boldsymbol{z}} = \frac{\tilde{\boldsymbol{x}}_{t_i}^p - \tilde{\boldsymbol{x}}_0^p \odot \sqrt{\bar{\boldsymbol{\alpha}}_{t_i}}}{\sqrt{1 - \bar{\boldsymbol{\alpha}}_{t_i}}}$
      $\boldsymbol{x}_{t_{i-1}}^p = \sqrt{\bar{\boldsymbol{\alpha}}_{t_{i-1}}} \odot \tilde{\boldsymbol{x}}_0^p + \boldsymbol{\sigma}_{t_i}(\eta) \odot \boldsymbol{z} + \sqrt{1 - \bar{\boldsymbol{\alpha}}_{t_{i-1}} - \boldsymbol{\sigma}_{t_i}(\eta)^2} \odot \hat{\boldsymbol{z}}$
    end for
  end if
  $\tilde{\boldsymbol{x}}_{t_{i-1}} \leftarrow [\tilde{\boldsymbol{x}}_{t_{i-1}}^0, \tilde{\boldsymbol{x}}_{t_{i-1}}^1, \dots, \tilde{\boldsymbol{x}}_{t_{i-1}}^P]$
end for
$\{\boldsymbol{x}_0^p \leftarrow \mathcal{F}^{-1}(\tilde{\boldsymbol{x}}_0^p)\}_{p \in \{1,\dots,P\}}$
return combined (unpatched) $[\boldsymbol{x}_0^0, \boldsymbol{x}_0^1, \dots, \boldsymbol{x}_0^P]$

Fig. 4: Uncurated generated samples on Fashion-MNIST.

V. Experiment Results

The Fermi level μ_t is linearly decreasing over time t from 4 to -4. The momentum coordinates in the frequency domain $\boldsymbol{k} = (k_1, k_2)$ take values between 0 and 1 with linear steps, where the step size is determined by the input image size. For example, for an image from the CelebA-64 [31] dataset of size 64×64, the precision of momentum coordinate increments is $\frac{1}{64}$. By using these momentum coordinates, we effectively vary the applied noise intensity on the input image. We set the temperature parameter T' to 0.5.

In their seminal diffusion work in [1], authors utilize a U-Net structure based on ResNet blocks. Inspired by [17], we modify the U-Net structure in [1] to match the output of our frequency-based forward process. First, we change the number

Fig. 5: Uncurated samples on CelebA-64 dataset.

Fig. 6: Curated samples on CelebA-64 dataset.

of input channels from 3 (RGB channels) to $3\times d\times d$ (number of frequency channels). Second, we keep the resolution constant throughout the model. Since the neural network parameters are shared across time, we use sinusoidal position embedding to encode time so that the neural network knows at which noise level it operates during the denoising process. In addition, since in FDDM the noise is scale-dependent, by inputting the time, we implicitly feed the noise-scale to the neural network as well.

To report the performance of the proposed FDDM, we perform experiments on Fashion-MNIST [32] (resolution 28×28) and CelebA-64 (resolution 64×64) [31]. We configure the forward and reverse process using $\eta = 1$ and $T = I = 1000$. We use a batch size of 128 and train for a total of 800k steps. We use the Adam optimizer in training, with the learning rate set to 2×10^{-4}, without any sweeping. We carry out the experiments using NVIDIA A100 40GB.

We first present a set of uncurated samples generated by FDDM, using Fashion-MNIST with patch size $d = 7$ in Fig. 4, demonstrating its image generation capabilities in the frequency domain. Next, we consider a more realistic CelebA-64 dataset and present the curated and uncurated generated samples in Figs. 5 and 6 for $d = 4$. Steps of the image denoising process and the corresponding frequency domain representation are as in Fig. 2, showcasing denoising in UV and IR parts as a function of timesteps. Overall, these results for both datasets demonstrate that the proposed FDDM successfully generates compelling images.

Next, we compare the performance of FDDM with the seminal image domain diffusion approach of [1] named DDPM, as the FDDM architecture is based on DDPM and our goal is to offer a performance-utility trade-off for this design. For comparisons, we use MACs (Multiply-accumulate operations) [24], number of inference steps per second, and Fréchet Inception Distance (FID) [33] as our metrics. FID measures the similarity between two sets of images and is shown the correlate with human judgement. In general, a low FID score indicates a good generated sample quality.

First, we take a look at the number of inference steps per second and MACs of the two schemes. Results in Table I show that inference in our proposed FDDM is significantly faster than DDPM (around 2.7 to 8.5$\times$ faster). This is attributed to our method of patching and forming frequency-based channels (as explained in section IV), which effectively reduces the computations in the U-Net. The performance enhancement is evident from the MACs column (lower the better) in Table I, demonstrating that our model requires significantly fewer MACs to generate 128 samples.

We compute the FID scores for the CelebA-64 dataset using 200000 samples and show the results in Table I. As expected, our FID scores are higher than the baseline, considering our faster and more time-efficient design. These results indicate a possible trade-off between the sample quality and training runtime efficiency. One observation is that lower FID scores around 3 are achieved when the finer details of the images are present. This means that our FID score around 18 does not mean $6\times$ worse images, and in fact the images generated by the proposed FDDM may be suitable for certain downstream tasks, for which finer details are not as critical. For example, the images we generate in Fig. 6 can be part of a synthetic dataset for training a classifier model that groups people according to their certain physical characteristics, e.g., hair color, glasses vs no glasses, and so on). For this task, finer details such as the background objects may not be as critical.

Model	MACs (G) ↓	# inference steps/s ↑	FID score ↓
DDPM [1]	3099	5.344	3.26
Ours (4x4 patch)	1781	14.655	18.17
Ours (8x8 patch)	449	45.76	20.65

TABLE I: Performance comparison of the proposed FDDM with DDPM [1] for $T = I = 1000$ on CelebA-64.

VI. Conclusion

We propose the Frequency Domain Diffusion Model (FDDM) as an alternative to image domain diffusion models. FDDM leverages the natural separation of components in the frequency domain and utilizes a scale-dependent noise schedule to intelligently add/remove noise during the diffusion process for efficient image generation. Combined with a JPEG-inspired design, FDDM achieves a computational speedup of $2.7-8.5\times$, with a modest impact on image quality.

REFERENCES

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

[2] 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.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[5] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[6] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[7] Guanhua Zhang, Jiabao Ji, Yang Zhang, Mo Yu, Tommi Jaakkola, and Shiyu Chang. Towards coherent image inpainting using denoising diffusion implicit models. In *International Conference on Machine Learning*, pages 41164–41193. PMLR, 2023.

[8] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[9] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.

[10] Batu Ozturkler, Chao Liu, Benjamin Eckart, Morteza Mardani, Jiaming Song, and Jan Kautz. Smrd: Sure-based robust mri reconstruction with diffusion models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 199–209. Springer, 2023.

[11] 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.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[15] K Ramamohan Rao and Ping Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[16] Gregory K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[17] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1740–1749, 2020.

[18] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. *Advances in Neural Information Processing Systems*, 31, 2018.

[19] Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis*, page 102846, 2023.

[20] Eliya Nachmani, Robin San Roman, and Lior Wolf. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

[21] 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. *arXiv preprint arXiv:2208.09392*, 2022.

[22] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alex Dimakis, and Peyman Milanfar. Soft diffusion: Score matching with general corruptions. *Transactions on Machine Learning Research*, 2023.

[23] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022.

[24] 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.

[25] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

[26] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 33:12438–12448, 2020.

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

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

[29] 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.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] 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.

[32] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[33] 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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.