Denoising Diffusion GANs

Parth Pujari, Anish Kulkarni, Cheshta Damor, Ashwin Abraham

1 Introduction

Stable Diffusion is a deep learning, text-to-image model released in 2022. It is primarily used to generate detailed images conditioned on text descriptions, though it can also be applied to other tasks such as inpainting, outpainting, and generating image-to-image translations guided by a text prompt.

Generative Adversarial Networks (GANs) are an approach to generative modeling (for generating realistic and high-quality synthetic data) using deep learning methods. GANs consist of two neural networks, the generator, and the discriminator, engaged in a competitive training process. The generator aims to produce data that is indistinguishable from real data, while the discriminator's role is to distinguish between real and generated data. Through this adversarial training, both networks improve iteratively.

1.1 Tackling a trilemma

We plan to reproduce the paper on "TACKLING THE GENERATIVE LEARNING TRILEMMA WITH DENOISING DIFFUSION GANs" by Zhisheng Xiao, Karsten Kreis and Arash Vahdat. The gist of the paper is as follows.

Abstract
A wide variety of deep generative models have been developed in the past decade. Yet, these models often struggle with simultaneously addressing three key requirements including: high sample quality, mode coverage, and fast sampling. We call the challenge imposed by these requirements the generative learning trilemma, as the existing models often trade some of them for others. Particularly, denoising diffusion models have shown impressive sample quality and diversity, but their expensive sampling does not yet allow them to be applied in many real-world applications. In this paper, the authors argue that slow sampling in these models is fundamentally attributed to the Gaussian assumption in the denoising step which is justified only for small step sizes. To enable denoising with large steps, and hence, to reduce the total number of denoising steps, they propose to model the denoising distribution using a complex multimodal distribution. They introduce denoising diffusion generative adversarial networks (denoising diffusion GANs) that model each denoising step using a multimodal conditional GAN.

The datasets used in the paper are the CIFAR-10, the stacked MNIST and/or CelebA-HQ all of which are found on Kaggle.

1.2 Probabilistic Modelling of the DDGAN

1.3 Background

We define the Diffusion model as a *latent variable* model of the form

$$p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T}$$

where $x_1 ... x_T$ are the latents as defined earlier. Note that they have the same dimensionality as the data x_0. The joint distribution $p_\theta(x_0)$ is called the *reverse process* and is defined as a Markov chain with learned Gaussian transitions starting from $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$ and

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \qquad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t)) \tag{1}$$

The *forward process* is fixed to a Markov chain that gradually adds Gaussian noise to data as per a variance schedule $\beta_1 .... \beta_T$:

$$q(x_{1:T}) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \qquad q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t \mathbf{I}) \tag{2}$$

Training is performed (similar to variational Bayes) by minimizing the negative log likelihood;

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] = \mathbb{E}_q \left[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] =: L \tag{3}$$

The forward process parameters are held constant (as hyperparameters). The forward process admits sampling x_t at an arbitrary time step in closed form, the notation used is : $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. Then,

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0,\, (1-\bar{\alpha}_t)\mathbf{I}) \tag{4}$$

We note that the forward process is derived from the conditional probability distribution $q(x_t|x_{t-1})$, our sampling is to be estimated by the probability distribution $p_\theta(x_{t-1}|x_t)$.

Our goal is to reduce the number of denoising diffusion steps T required in the reverse process of diffusion models. Inspired by the observation above, we propose to model the denoising distribution. with an expressive multimodal distribution. Since conditional GANs have been shown to model complex conditional distributions in the image domain , we adopt them to approximate the true denoising distribution $q(x_{t-1}|x_t)$.

To set up the **adversarial training** we denote the time dependent discriminator as $D_\phi(x_{t-1}, x_t, t)$: $\mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R} \rightarrow [0, 1]$, with parameters ϕ. It takes the N dimensional images x_{t-1} and x_t and decides whether x_{t-1} is a plausible denoised version of x_t. The generator takes as input x_t and tries to generate a plausible x_0 (called x'_0) from $p_\theta(x_0|x_t)$.

1.4 Neural Network model

The idea is that now our approximation to the posterior $p_\theta(z|x)$ is going to be a neural network and the parameters θ (and consquently ϕ for the discriminator) are jointly optimized using minibatch gradient descent. We model the Discriminator as a Convolutional Neural Network with heavy downsampling and the Generator as a UNet, both with timestep embeddings. We also train a DDPM model on the Diffuser's UNet. Finally we train the models used in the paper and test them on CIFAR-10 and stacked MNIST.

Algorithm 2: Denoising Diffusion GANs Training Algorithm

```
Input: Training dataset $\mathcal{D}$, Generator $G$, Discriminator $D$, Diffusion process parameters,
       Learning rate $\alpha$
Output: Trained Generator $G$
for each training iteration do
    Sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$;
    Sample a timestep $t$ uniformly;
    for each data point $\mathbf{x}$ in $\mathcal{B}$ do
        Sample $\mathbf{y_t} = (x_{t-1}, x_t)$ from the forward process;
        Sample $x'_0$ from $p_\theta(x_0|x_t)$ and $x'_{t-1}$ from $q(x'_{t-1}|x'_0, x_t)$;
        Let $\tilde{\mathbf{y_t}} \coloneqq (x'_{t-1}, x_t)$, $\mathbf{x} \in \mathcal{B}$;
    Update Discriminator $D$ using $\mathcal{B}$ and $\tilde{\mathcal{B}}$ according to the GAN objective:

        $\nabla_{\theta_D} \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} -\log D(\mathbf{y_t}, t) + \nabla_{\theta_D} \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} -\log(1 - D(\tilde{\mathbf{y_t}}), t)$

     Sample another $t$ uniformly and another $\tilde{\mathbf{y_t}}$;
    Update Generator $G$ using $\mathcal{B}$ to minimize the denoising diffusion GAN objective:

        $\nabla_{\theta_G} \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} -\log D(\tilde{\mathbf{y_t}})$
```

2 Results

2.1 Diffusion on MNIST and Fashion MNIST

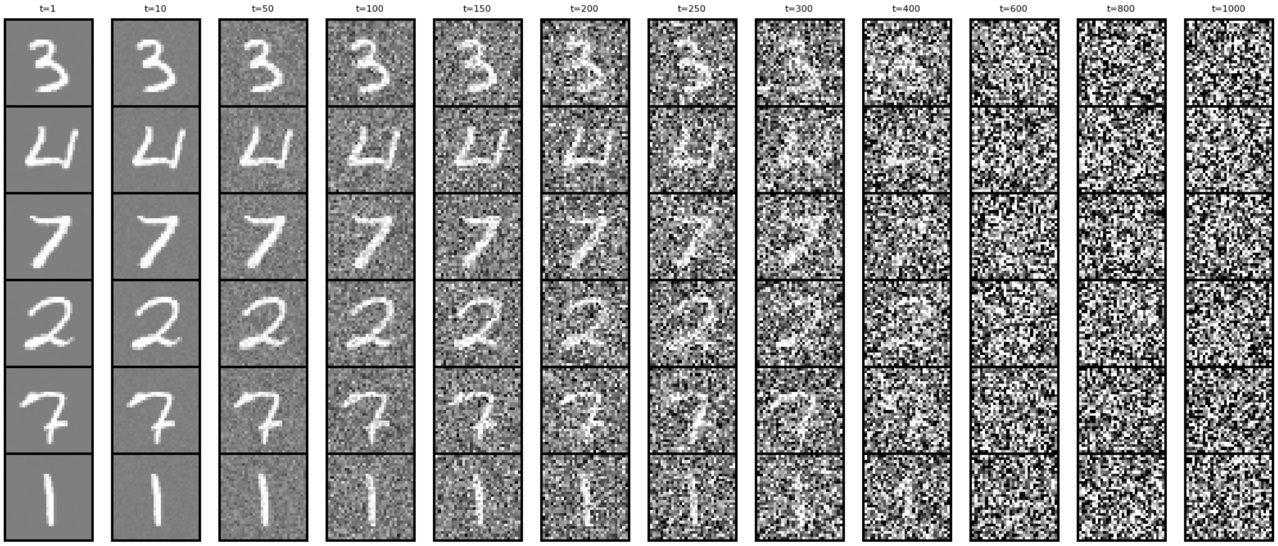
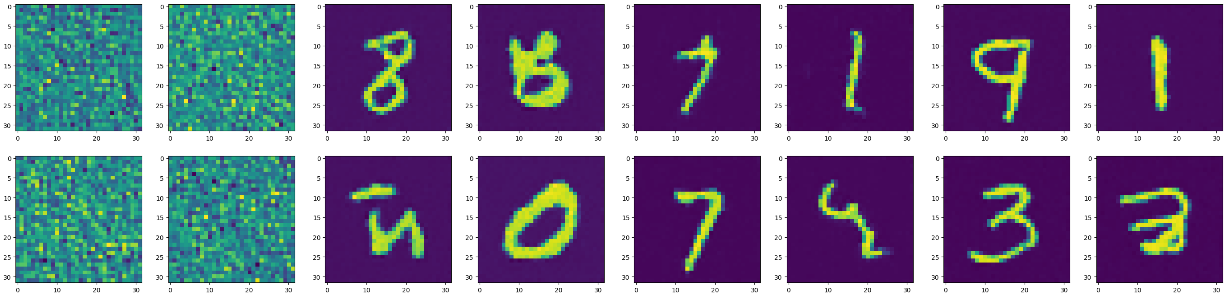
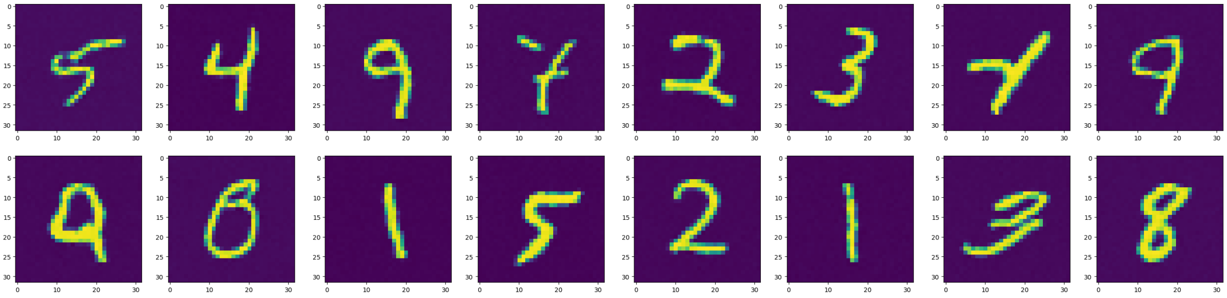


Figure 1: Forward Diffusion on MNIST



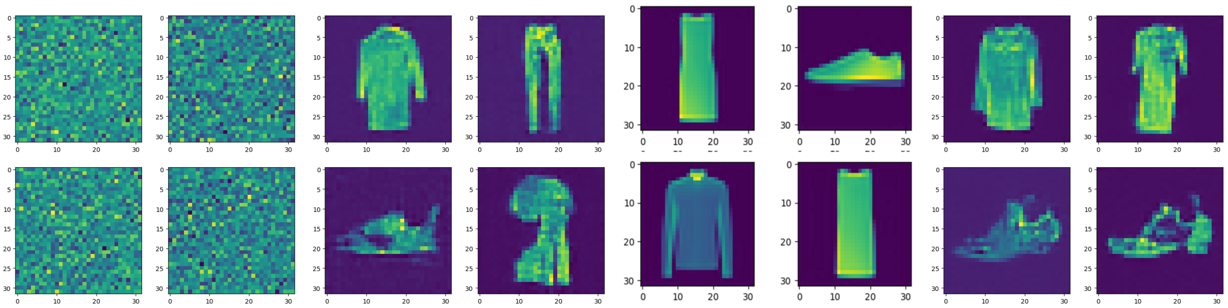
(a) (b) (c) (d)

Figure 2: Left to right, epochs 1 through 4



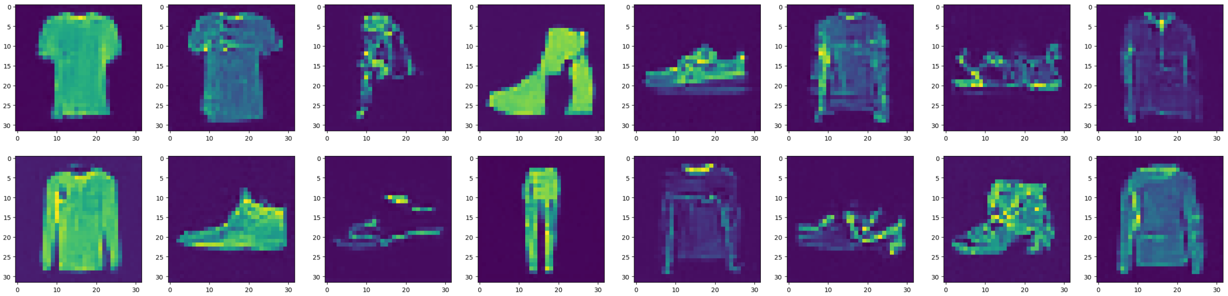
(a) (b) (c) (d)

Figure 3: Epoch 4



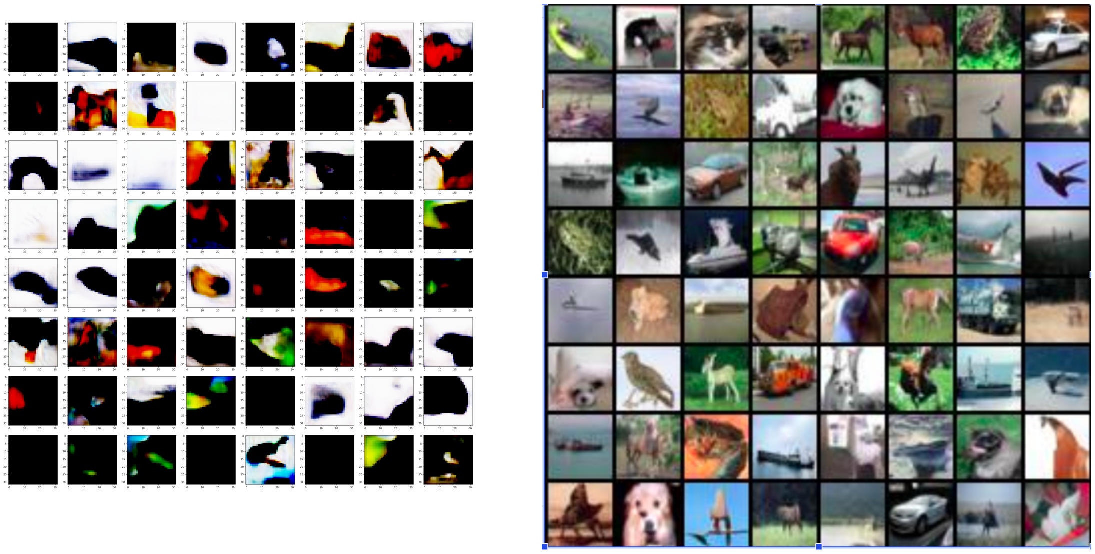
(a) (b) (c) (d)

Figure 4: Left to right, epochs 1 through 4



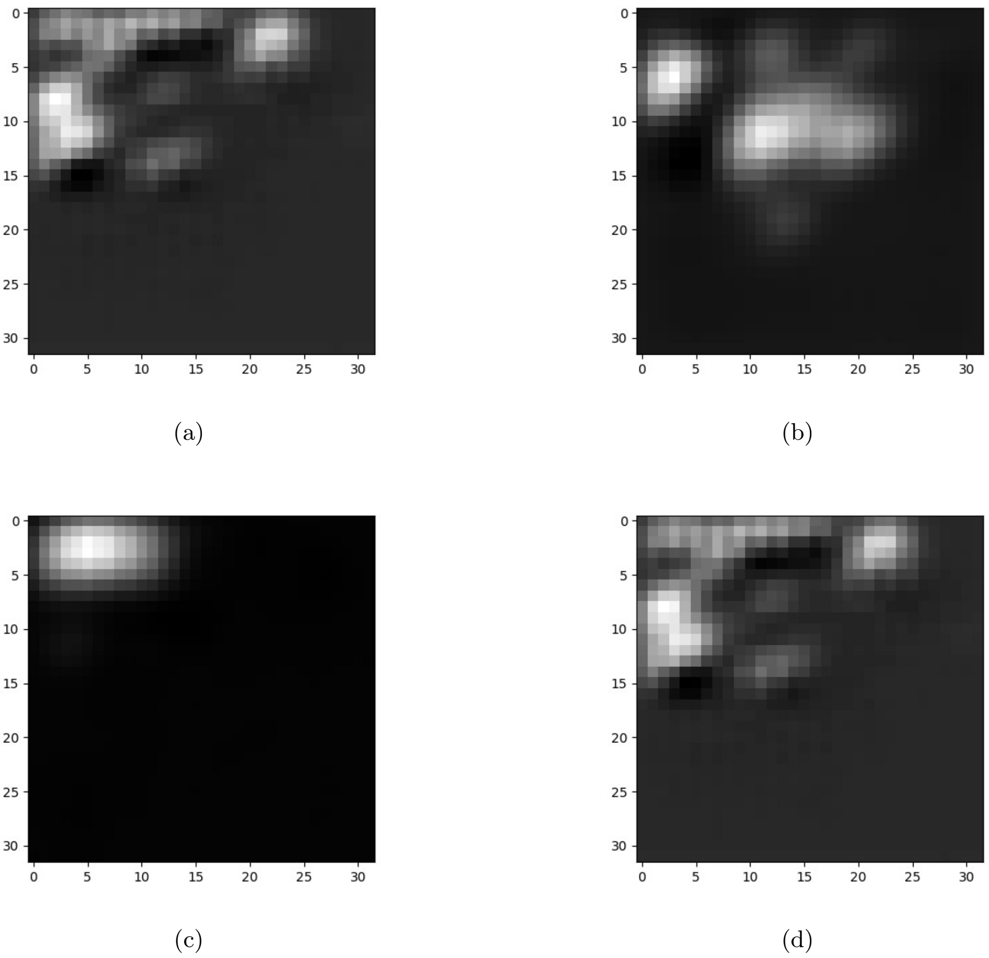
(a) (b) (c) (d)

Figure 5: Epoch 4



(a) (b)

Figure 6: Epoch 20 and 1000 for CIFAR10



(a)

(b)

(c)

(d)

Figure 7: Very strong generator and weak discriminator on MNIST