Diffusing the art of Diffusion Models

Kshitiz Kumar Tanay Yadav Tanmay Goyal Tanmay Shah

May 3, 2023

Introduction

The present era is truly exciting, with the advent of tools like ChatGPT and DALL-E 2. What is even more thrilling is that better and stronger versions of the same are under wraps, waiting to be unleashed. Moreover, the interleaving of Machine and Deep Learning with techniques from Physics, Chemistry, etc., leaves much to be desired, and the technology is making headway rapidly. One example of this is Physics-informed Neural Networks, which are making waves in the Deep Learning Community. In this paper, we shall explore Diffusion models, which arrive from a similar realm of interdisciplinary Machine Learning. It also forms the backbone of some of the state-of-the-art models, such as DALL-E 2 and GLIDE.

Diffusion

- In the field of thermodynamics, diffusion is defined as *the movement of particles from a region of higher concentration to that having a lower concentration.* In physics literature, diffusion results in an increase in entropy, whereas in information theory, it leads to a loss in information.
- The idea of a diffusion model is very similar to that of Variational Auto-encoders (VAEs): to project the data into a latent space and recover the original data. However, diffusion models have several advantages over GANs and VAEs, such as little training (unlike the encoder in a VAE, training is minimal due to the presence of Markov Chains), and little to no adversarial training required.

Forward Diffusion

- At every stage, we choose a transition kernel, which depends only on the current state (Markov Chain), and none of the previous states, i.e

$$q(x_{t+1}|x_t) = T(x_{t+1}|x_t;\ \beta) \tag{1}$$

where T is the Markov Kernel, and β is the diffusion rate.

- We choose our Transition Kernel T to be the Gaussian Kernel. Thus,

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

where, $\beta_i\ ;\ 1 \leq i \leq T$ are pre-determined hyper-parameters.

Destruction of the image

- We can show the analytical form of $q(x_t|x_0)$ to be:

$$q(x_t|x_0) = \mathcal{N}(x_t;\ x_0\sqrt{\hat{\alpha}_t}, (1-\hat{\alpha}_t)\mathbf{I}) \tag{3}$$

- To sample x_t, we can use the following:

$$x_t = x_0\sqrt{\hat{\alpha}_t} + \sqrt{(1-\hat{\alpha}_t)}\,\mathcal{N}(0,\mathbf{I}) \tag{4}$$

- This has an important implication. If the diffusion rates β are chosen properly to ensure $\hat{\alpha_T} \approx 0$, we can obtain an **isotropic Gaussian** such that $q(x_T) \approx \mathcal{N}(0,\mathbf{I})$. Thus, the forward diffusion process ensures almost all structures within the image are destroyed.
- To generate new samples, we can derive an unstructured noise vector from the prior distribution and remove noise from it by running a Markov Chain in the reverse time direction.

Reverse Diffusion

- We also have a learnable kernel given by:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

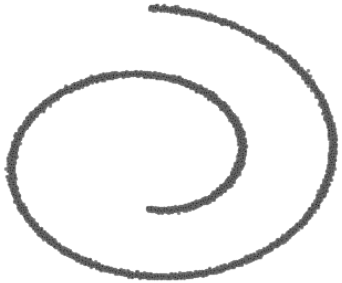
  where θ represents the model parameters, and the mean $\mu_\theta(x_t, t)$ and variance $\Sigma_\theta(x_t, t)$ are parametrized by deep neural nets.
- The eventual goal is to learn the transition kernels in the reverse Markov Chain to be very similar to the kernels used in the forward Markov Chain. Since we wish to minimize the distance between two distributions, we shall use the KL divergence joint distributions of the forward Markov Chain and the reverse Markov Chain.

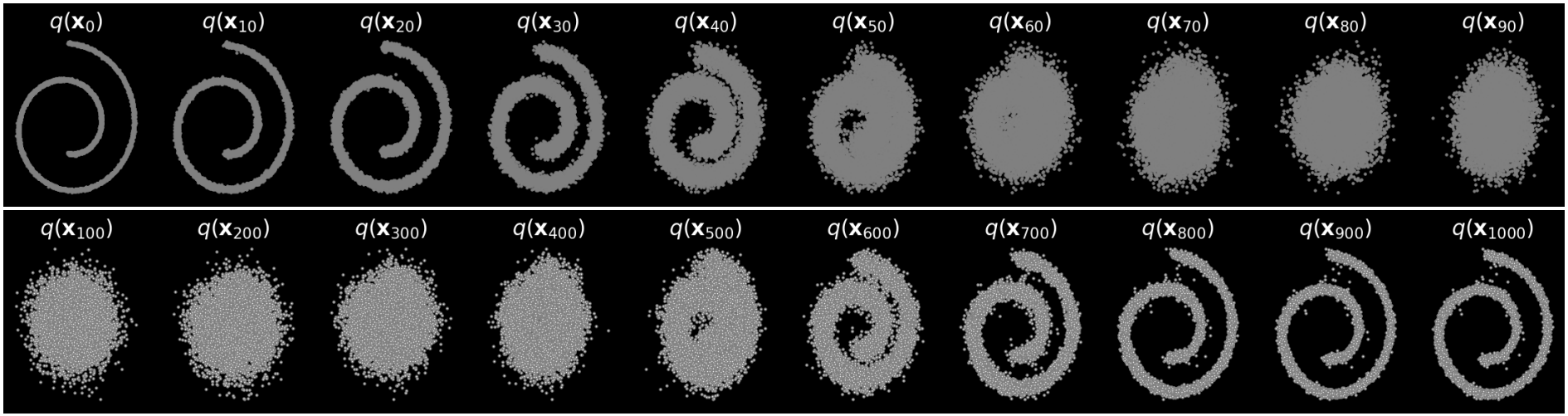
The Loss Function

$$\log p(x) = \mathbb{E}_{q(x_1|x_0)}\left[\log p(x_0|x_1)\right] + KL[q(x_{T-1}, x_T|x_0)||p(x_T)] \\ -\mathbb{E}_{q(x_{t-1},x_{t+1}|x_0)}\ KL[q(x_t|x_{t-1})||p(x_t|x_{t+1})]$$

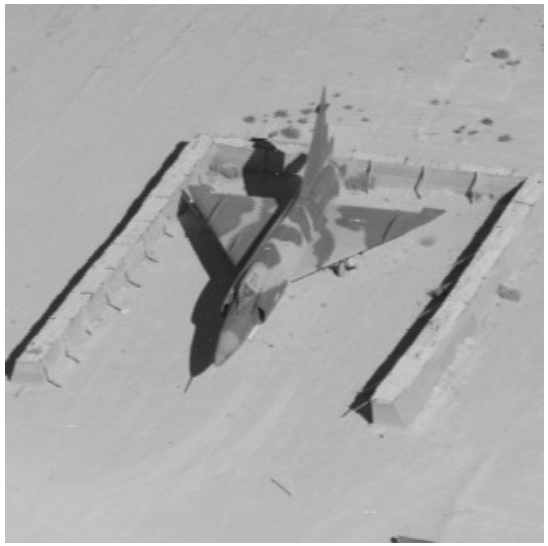
- The first term $\mathbb{E}_{q(x_1|x_0)}\left[\log p(x_0|x_1)\right]$ is called the *reconstruction term*, which predicts the probability of the original sample given the first denoised image.
- The second term $KL[q(x_{T-1}, x_T|x_0)||p(x_T)]$ is the KL divergence between the final latent distribution and the Gaussian prior. Note that there is no trainable term here, and effectively becomes zero as soon as we assume $T \to \infty$. This term is called the *prior matching term*.
- The third term is the *consistency term* which ensures that the distribution at x_t in both forward and reverse directions is consistent, i.e the denoising step from the reverse Markov chain should match the noising step in the forward chain.

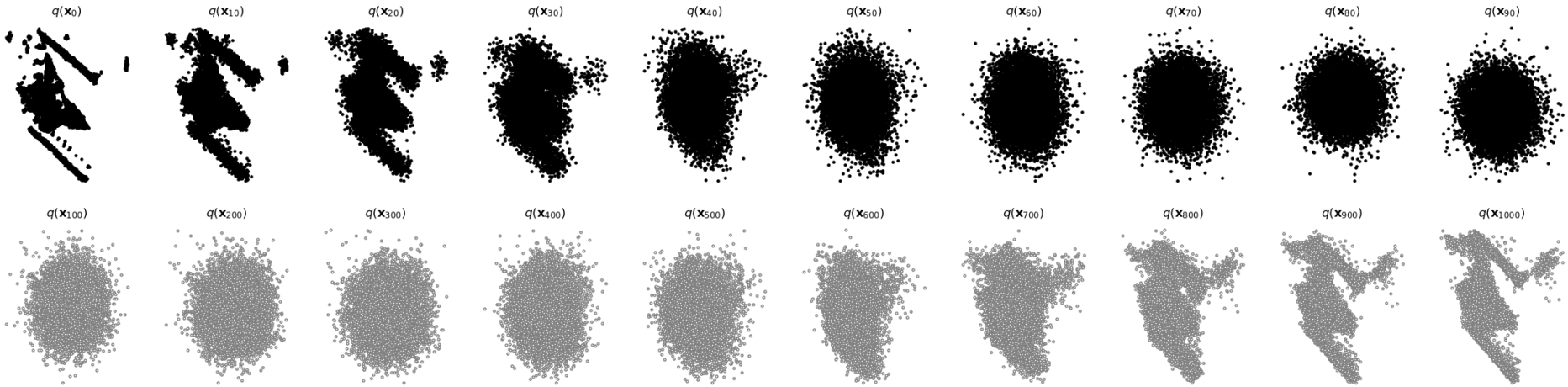
Results



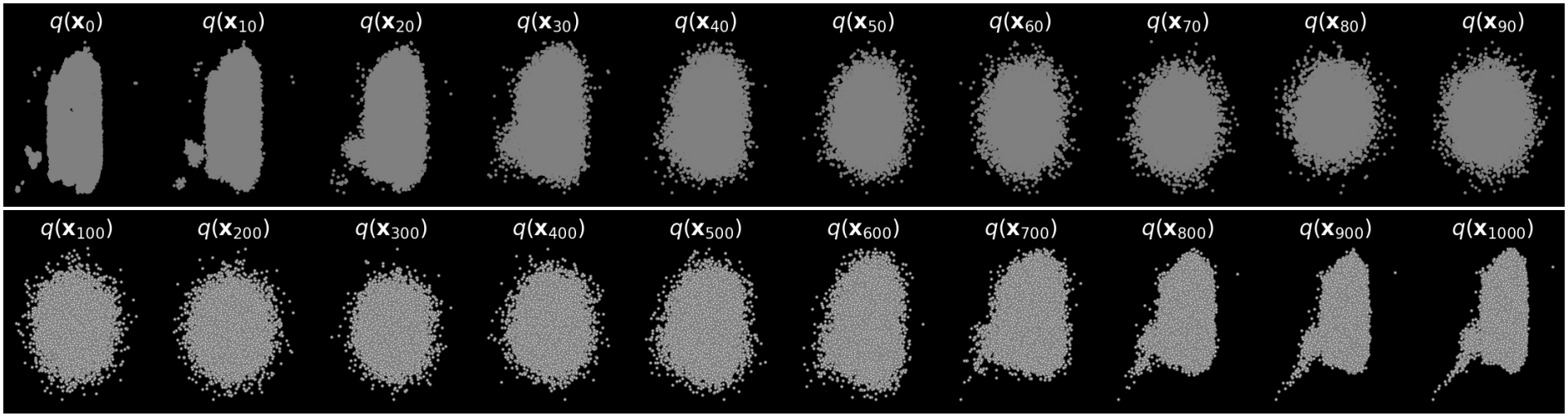

$q(\mathbf{x}_0)$
$q(\mathbf{x}_{10})$
$q(\mathbf{x}_{20})$
$q(\mathbf{x}_{30})$
$q(\mathbf{x}_{40})$
$q(\mathbf{x}_{50})$
$q(\mathbf{x}_{60})$
$q(\mathbf{x}_{70})$
$q(\mathbf{x}_{80})$
$q(\mathbf{x}_{90})$
$q(\mathbf{x}_{100})$
$q(\mathbf{x}_{200})$
$q(\mathbf{x}_{300})$
$q(\mathbf{x}_{400})$
$q(\mathbf{x}_{500})$
$q(\mathbf{x}_{600})$
$q(\mathbf{x}_{700})$
$q(\mathbf{x}_{800})$
$q(\mathbf{x}_{900})$
$q(\mathbf{x}_{1000})$

Results




$q(\mathbf{x}_0)$
$q(\mathbf{x}_{10})$
$q(\mathbf{x}_{20})$
$q(\mathbf{x}_{30})$
$q(\mathbf{x}_{40})$
$q(\mathbf{x}_{50})$
$q(\mathbf{x}_{60})$
$q(\mathbf{x}_{70})$
$q(\mathbf{x}_{80})$
$q(\mathbf{x}_{90})$
$q(\mathbf{x}_{100})$
$q(\mathbf{x}_{200})$
$q(\mathbf{x}_{300})$
$q(\mathbf{x}_{400})$
$q(\mathbf{x}_{500})$
$q(\mathbf{x}_{600})$
$q(\mathbf{x}_{700})$
$q(\mathbf{x}_{800})$
$q(\mathbf{x}_{900})$
$q(\mathbf{x}_{1000})$

Results


$q(\mathbf{x}_0)$
$q(\mathbf{x}_{10})$
$q(\mathbf{x}_{20})$
$q(\mathbf{x}_{30})$
$q(\mathbf{x}_{40})$
$q(\mathbf{x}_{50})$
$q(\mathbf{x}_{60})$
$q(\mathbf{x}_{70})$
$q(\mathbf{x}_{80})$
$q(\mathbf{x}_{90})$
$q(\mathbf{x}_{100})$
$q(\mathbf{x}_{200})$
$q(\mathbf{x}_{300})$
$q(\mathbf{x}_{400})$
$q(\mathbf{x}_{500})$
$q(\mathbf{x}_{600})$
$q(\mathbf{x}_{700})$
$q(\mathbf{x}_{800})$
$q(\mathbf{x}_{900})$
$q(\mathbf{x}_{1000})$

DALL-E 2

- DALL-E 2 is the amalgamation of multiple models made by Open AI. It first takes the caption and generates text embeddings from them using a model called CLIP (Contrastive Language-Image Pre-training). These text embeddings are then passed through a "prior." This prior is a diffusion model. Finally, the output of the prior as well as the text embeddings are passed to the decoder, which is Open AI's famous model GLIDE(Guided Language to Image Diffusion for Generation and Editing). The final image is then upsampled from 64×64 to 1024×1024 to get the final result.
- In brief, the CLIP model is trained by generating embeddings from the caption and the text and minimizing the cosine distance between the two.
- Diffusion is one of the most important concepts going into the prior and the decoder GLIDE. The prior is crucial because this is where most variations are generated. GLIDE simply goes above and beyond and uses both text and image embeddings to generate an image.

Stable Diffusion

Stable diffusion helps overcome a major computational drawback of the Diffusion Process by shifting the process from the image space to the pixel space. It does so by compressing the image into a latent space and then restoring the image from the latent space. So, instead of generating a noisy image, it shall generate a random tensor in the latent space and corrupt it with noise. Finally, the decoder aims to convert this latent space representation to the normal image space.

THANK YOU

May 3, 2023