Investigating Novel Methods to Perform Image-to-Image Translation

Student Name: Ishan Srivastava
Supervisor Name: Dr Chris G Willcocks
Submitted as part of the degree of M.Sc. MISCADA to the
Board of Examiners in the Department of Computer Science, Durham University.
Submitted: August 23, 2022

Abstract

An essential application of deep generative models in computer vision has been to perform image-to-image translation. Traditionally, Generative Adversarial Networks (GANs) inspired models have been employed successfully for the task, with a few limitations such as mode collapse. Recently, diffusion-based models demonstrated state-of-the-art results in high-quality sample generation and better mode coverage. However, their application has been primarily limited to unconditional sample generation. In this paper, we leverage the recent advances in generative modelling to perform image-to-image translations. We incorporate conditional generative models to perform the unpaired and paired image-to-image translation tasks. Our model begins by systematically corrupting a given dataset with a forward stochastic differential equation. Simultaneously, a score function approximated by a conditional UNet is trained to denoise the corrupted data over several iterations, given the conditional information. Then, a reverse-time stochastic differential equation using the learned score function is employed to generate conditional samples from noise and conditional images. We observe that our model performs the paired image-to-image translations at par with other existing methods by calculating the Structural Similarity Index Measure. However, our unpaired model falls short of executing its task due to memory and time limitations. We propose a few methods to remedy this and achieve better results on the unpaired image-to-image translation task.

Keywords: Generative Models, Stochastic Differential Equations (SDEs), Image-to-Image translation, Score Function, Cycle Consistency.

Contents

1 Introduction

An important class of problems in computer vision can be formulated as an image-to-image translation task, where functions are determined to translate images in a given domain A to another image domain B and vice versa. For instance, a satellite image can be translated into google style maps; a low-resolution image can be translated to a high-resolution image, etcetera. Mathematically the image-to-image translation task comprises learning the conditional distribution of outputs given the inputs. In the presence of paired dataset, conditional generative models can be employed to capture the conditional distributions. An unpaired image-to-image approach is required to approximate the conditional distribution in a more challenging setting, where supervision is unavailable.

A class of deep learning models based on the Generative Adversarial Networks (GANs) [1] have significantly been successful in performing many of the image-to-image translation tasks [11, 14–17]. In general, the GANs-based models generate high-quality samples very efficiently. However, GANs suffer from what is known as mode collapse, where they are unable to learn all the modes of a multimodal conditional distribution. Other models, such as the variational autoencoders [2] and normalizing flows [3] can cover the modes of a multimodal data distribution but perform poorly at high-quality sample generation.

Recently, a new class of models inspired by physics, called the diffusion-based models, have outperformed the GANs-based models on several of the image-to-image translation tasks [5–8, 12, 13, 18]. The diffusion-based models can generate high-quality samples while maintaining a solid mode coverage. The critical approach incorporated by the diffusion models is to corrupt a given dataset by slowly injecting noise into it and learn a denoising function to remove the injected noise. Then after training, the denoising function can generate new high-quality samples over several iterative steps. In [7], the authors proposed a score-based generative model with stochastic differential equations, which unified the existing diffusion models as limiting cases to certain stochastic differential equations. Although diffusion-based models have been remarkably successful at the paired image-to-image tasks such as super-resolution, colourization, and inpainting, their application to unpaired tasks has been minimal [18].

In this paper, we propose a new method to perform the image-to-image translation task. This method is based on the diffusion models and utilizes a score-based approach to learn the conditional distributions. We employ our model to perform both unpaired and paired image-to-image translation tasks. The key to our model for the unpaired tasks is pairing the given unpaired datasets utilizing a set of Euler-Maruyama samplers. Initially, the model generates poor-quality paired image samples. However, the model training improves the quality of samples over several training steps. The domain translation functions are further constrained by incorporating a cycle-consistent loss providing better approximations.

This paper is organized as follows. In the second section, we briefly overview the Denoising Diffusion Probabilistic Models (DDPMs) and the Score-Based Generative Models with SDEs for the generative tasks. We also define the image-to-image translation task formally in section two. Section three discusses our method for performing the unpaired image-to-image translation tasks. We also show how our model can be adapted to perform

the paired image-to-image translations. In section four, we discuss the implementation of our method in PyTorch and the choice of hyperparameters for training our model. We also discuss the results and shortcomings of our methods and suggest improvements to deal with the deficiencies.

2 Related Work

The central idea of diffusion-based generative modelling is to perturb given data with multiple noise scales and then learn a denoising function to reverse the process. Here, we provide an overview of the generative modelling process based on DDPMs and SDEs. The critical difference between the two is that in SDE-based methods, the noise scales are continuous, and in DDPMs, they are discrete.

2.1 Denoising Diffusion Probabilistic Models

DDPMs are a class of latent variable models [4], where the intermediate latent codes share the same dimensionality as that of the data $\mathbf{x}_0$. For a given data point from an i.i.d distribution $\mathbf{x}_0 \sim p_0$, a discrete Markov chain $\{\mathbf{x}_i\}_{i=0}^N$, where $i \in [0, N]$ and $\mathbf{x}_N \sim p_N$ is a tractable distribution, can be constructed by Gaussian perturbation kernel as given in [5] of form

$$p(\mathbf{x}_i|\mathbf{x}_{i-1}) = \mathcal{N}(\mathbf{x}_i; \sqrt{1-\beta_i}\mathbf{x}_{i-1}, \beta_i \mathbb{1}) \ , \tag{2.1}$$

where $0 < \beta_i < 1$ are the noise scales which are usually chosen such that $\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$. The discrete Markov chain constructed from the perturbation kernel in (2.1) is given as

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{\beta_i}\mathbf{z}_{i-1} \ , \qquad i = 1, \cdots, N \ , \tag{2.2}$$

where $\mathbf{z}_{i-1} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$.

Following [5], a reverse Markov chain can be parameterized with another Gaussian perturbation kernel of form

$$p_\theta(\mathbf{x}_{i-1}|\mathbf{x}_i) = \mathcal{N}\left(\mathbf{x}_{i-1}; \frac{1}{\sqrt{1-\beta_i}}\left(\mathbf{x}_i + \beta_i s_\theta(\mathbf{x}_i, i)\right), \beta_i \mathbb{1}\right) \ , \tag{2.3}$$

where s_θ is parameterized by a neural network whose architecture must be such that the input dims are equal to the output dims, and $\mathbf{x}_N$ is pure noise.

The parameters of s_θ can be learned by minimizing an objective function as given in [5,6]. After the training process, samples can be generated by following equation (2.2)

$$\mathbf{x}_{i-1} = \frac{1}{\sqrt{1-\beta_i}}(\mathbf{x}_i + \beta_i s_\theta(\mathbf{x}_i, i)) + \sqrt{\beta_i}\mathbf{z}_i, \qquad i = N, N-1, \cdots, 1, \tag{2.4}$$

where $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$.

The DDPMs improve the quality of samples generated by GANs [6] and other models such as Normalizing Flows, and Autoregressive models [7]. However, the sampling from diffusion models is still expensive in practice due to the thousands of evaluations required.

2.2 Score-Based Generative Modelling with SDEs

In [7], the authors showed that the DDPMs are a subclass of a more general class of models known as Score-Based Generative Models. These models consist of a continuous Markov chain defined by a forward stochastic differential equation. A corresponding reverse Markov chain is defined with a reverse-time stochastic differential equation. In a discrete limit, it was shown in [12] that the forward SDE reduces to equation (2.2). Any black box numerical SDE solver can be used to solve the reverse-time SDE yielding high-quality samples from the original data distribution.

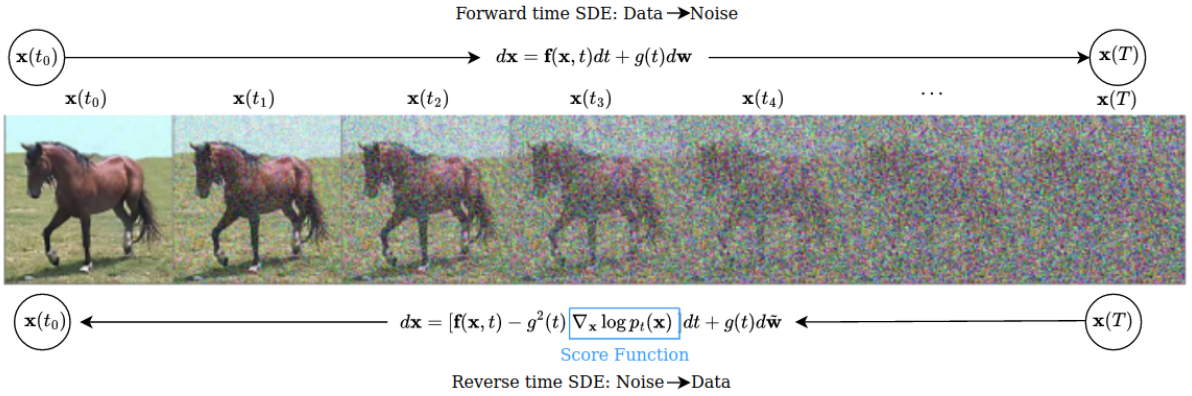

Figure 1: **Genetative modelling via SDEs.** The data distribution is continuously perturbed via a forward SDE to map it to a noise distribution. The reverse SDE is used for generating samples from the data distribution.

2.2.1 Perturbing Data with SDE

Consider a dataset of i.i.d samples $\mathbf{x}(0) \sim p_0$. A forward diffusion process $\{\mathbf{x}(t)\}_{t=0}^{T}$, where t is a continuous time variable $t \in [0, T]$ and $\mathbf{x}(T) \sim p_T$ is a tractable distribution, is defined by the solution of an Itô SDE [7]

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \tag{2.5}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued drift coefficient, $g : \mathbb{R} \to \mathbb{R}$ is a scalar diffusion coefficient, and $\mathbf{w}$ is a standard Wiener process [7]. The perturbed data $\mathbf{x}(t)$ is sampled from the intermediate distribution p_t. Typically, the drift and the diffusion coefficients are chosen such that the distribution p_T is a Gaussian distribution with a fixed mean and variance.

In practice, a sample from a forward SDE process is obtained via a Gaussian diffusion process since the coefficients of the SDE are fixed, such as the perturbation kernel is Gaussian [7]. The sample at a time t is given as

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}) \ , \tag{2.6}$$

where $\alpha(t)$ denotes the magnitude of data, and $\sigma(t)$ refers to the amount of noise $\mathbf{z}$.

2.2.2 Generating Samples with Reverse SDE

In [10], Anderson showed that for a given forward time diffusion process, there exists a corresponding reverse time diffusion which runs backwards in time and is given by the reverse-time SDE

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + g(t)d\tilde{\mathbf{w}}, \tag{2.7}$$

where p_t is the marginal probability distribution of the perturbed samples $\mathbf{x}(t)$ and $\tilde{\mathbf{w}}$ is a standard Wiener process for reversed time $t \in [T, 0]$ Wiener process for reversed time $t \in [T, 0]$. The infinitesimal timestep dt is taken to be negative. The term $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is referred to as the score function [7] of a distribution and once it is determined for each time t, samples can be generated from the original distribution p_0.

2.2.3 Estimating the Score Function

The knowledge of the terminal distribution p_T and the score-function at time $t \in [0, T]$ is required to solve the reverse-time SDE equation (2.7). The drift and the diffusion coefficients in equation (2.5) are fixed such that the terminal distribution p_T is a Normal distribution, i.e. $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbb{1})$, and the intermediate samples obey equation (2.6). The score function of p_t is usually not known analytically. But, it can be approximated by training a parametric function s_θ, referred to as a time-dependent score-based model [7,8], such that $s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

The learning objective function given the score model $s_\theta(\mathbf{x}, t)$ is given as

$$\theta^* = \underset{\theta}{argmin}\ \mathbb{E}_{t\sim\mathcal{U}[0,T]} \left\{\lambda(t)\mathbb{E}_{\mathbf{x}\sim p_0, z\sim\mathcal{N}(\mathbf{0},\mathbb{I})} \left[||\sigma_t s_\theta(\mathbf{x}, t) + \mathbf{z}||_2^2\right]\right\} \ , \tag{2.8}$$

where σ_t is the covariance of $\mathbf{x}$ at time $t \in [0, T]$, and $\lambda(t)$ is a positive weighting function. **Note**: Throughout this work, we refer to the score model as the score function interchangeably.

2.2.4 Solving the Reverse SDE Numerically

Various numerical methods exist to solve a stochastic differential equation, such as the Euler-Maruyama (EM), Milstein, and stochastic Runge-Kutta [9]. Any of these methods can be used to approximate the solution for the reverse-time SDE. Here we discuss the EM method.

Once the score-based model is trained such that $s_{\theta^*}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, the reverse-time SDE can be discretized using finite negative timesteps and finite Gaussian noise and solved iteratively. The iterations run until $t \to \epsilon$, where ϵ is the desired convergence, and each iteration follows

$$\begin{aligned} \Delta\mathbf{x}_n &\leftarrow \left[\mathbf{f}(\mathbf{x}_n, t_n) - g^2(t_n)s_{\theta^*}(\mathbf{x}_n, t_n)\right] \Delta t + g(t_n)\sqrt{|\Delta t|}\mathbf{z}_{t_n} \\ \mathbf{x}_{n-1} &\leftarrow \mathbf{x}_n + \Delta\mathbf{x}_n \\ t_{n-1} &\leftarrow t_n + \Delta t \ , \end{aligned} \tag{2.9}$$

where Δt is a small negative timestep, $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ [7]. Algorithm 1 and 2 depict the pseudocode for training and sampling procedure for score-based generative models via SDEs.

Other sampling methods exist, such as Predictor-Correlator sampling and Ancestral Sampling [7], which incorporate the score-based MCMC approaches to sample from the terminal distribution p_T. For this paper, we have employed only the Euler-Maruyama sampler as they are more intuitive to understand in the context of SDEs. Algorithms 1 and 2 depict a score-based model's training and sampling procedure.

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim p(\mathbf{x}_0)$
  $t \sim \mathcal{U}[0, 1]$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
  $\sigma_t \leftarrow \sigma(\mathbf{x}_0, t)$
  Take gradient step on
    $\nabla_\theta \left[||\sigma_t s_\theta(\mathbf{x}, t) + \mathbf{z}||_2^2 \right]$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
for $t = 1, \cdots, \epsilon$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ if $t > \epsilon$, $else$ $\mathbf{z} = 0$
  $\mathbf{x}_{t-1} = \mathbf{x}_t + \left[f(\mathbf{x}_t, t) - g^2(t) s_\theta(\mathbf{x}_t, t) \right] \Delta t + g(t) \sqrt{|\Delta t|} \mathbf{z}$
end for
return $\mathbf{x}_0$

Unlike earlier diffusion-based models [5, 6], the more general score-based generative models provide more flexibility in sample generation. We are not restricted to using the EM method to solve the reverse-time SDE numerically [7] and any general black-box numerical SDE solver can be utilized for the task [9].

2.3 Image-to-Image Translation

The DDPMs and the Score-Based Models have been utilized extensively for unconditional image generation [5–8]. However, these methods can easily be adapted to the conditional setting by simply modifying the architecture of the UNet model, which approximates the score function [12, 13, 23]. This way, these can be utilized to perform the image-to-image translation tasks. The task of image-to-image translation pertains to learning a mapping function from a domain X to another domain Y and visa versa. Figure 2 illustrates an image-to-image translation task, where domain X corresponds to google map satellite photos and domain Y corresponds to street maps.

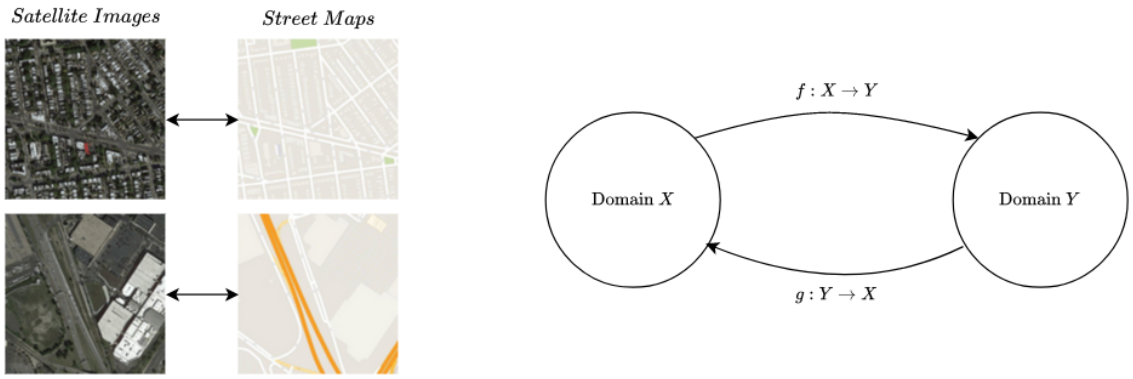

Figure 2: **Left:** Google map satellite photos and corresponding street map. **Right:** Image-to-Image translation task.

Consider images $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. Then the two dataset is paired iff there exists images $\tilde{\mathbf{y}} \in Y$ and $\tilde{\mathbf{x}} \in X$ corresponding to every $\mathbf{x}$ and $\mathbf{y}$ respectively. Otherwise, the dataset is referred to as unpaired. The paired dataset is represented as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$, where N is the size of the dataset. Whereas the unpaired dataset is represented as $\{\mathbf{x}_i\}_{i=1}^{N_X}$ and $\{\mathbf{y}_i\}_{i=1}^{N_Y}$, where N_X and N_Y are the length of the X and Y datasets respectively. The mapping function $f : X \to Y$ and $g : Y \to X$ can be learned using a training set of paired or unpaired images.

Existing Methods for Image-to-Image Translations

Paired Tasks

In the presence of a paired dataset, additional information is bestowed in the form of the other image domain. In [11], the authors successfully incorporated this additional information by using Conditional GANs to perform paired image-to-image translation tasks [11]. More specifically, the authors used a UNet architecture for a generator to pass in the conditional information. An unconditional generator inputs only a noise vector and produces an image. However, a conditional generator with a UNet architecture inputs an image and a random noise vector to create another image. The generator is paired with a discriminator, which acts as an adversary assisting the generator in producing more "real" images.

More recently, in [12] and [13], the authors utilized a DDPM-based approach to perform image-to-image translations on paired datasets. The key to their approach was incorporating a conditional UNet architecture in the denoising step to achieve conditional image generation. Essentially, by modifying the UNet architecture, the authors were able to input the conditional information in the model, which restricted the vector space of the reverse Markov chain as given in equation (2.4) to a subspace of the joint distribution $p(\mathbf{x}, \mathbf{y})$. The subspace corresponds to either of the two domains X or Y depending on the conditioning. In our work, we have incorporated a similar strategy of using a conditional UNet architecture to pass in the conditioning at each step of the reverse diffusion.

Unpaired Tasks

Unpaired image-to-image translations generalize the paired task and seek to learn the domain translation functions without paired datasets. The seminal approach of CycleGANs [14], proposed using a set of generators f and g which learns to map $X \to Y$ and $Y \to X$. The generators are paired with corresponding discriminators, which assists them in getting better at the translation task. The translation task is reinforced by introducing cycle consistency losses, ensuring that the functions f and g are such that $f \circ g : X \to X$ and $g \circ f : Y \to Y$. CycleGANs, while extremely efficient at high-quality sample generation, suffer from mode collapse and are highly dependent on the architecture choice.

CycleGANs was later improved by another approach referred to as UNIT [15], which uses a set of Coupled GANs to perform the unpaired image-to-image task. The authors of UNIT made an important assumption that a pair of corresponding images $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $\mathbf{x} \in X$ and $\mathbf{y} \in Y$, can be mapped to a shared-latent space $\mathcal{Z}$. This is referred to as the shared-latent space assumption. Further, the authors incorporated two encoders with shared weights to map the $\mathbf{x}$ and $\mathbf{y}$ to a shared latent space. They then used a set of generators and corresponding discriminators to map the latent codes in $\mathcal{Z}$ to images in the other domain. The UNIT is generalized to address multimodal datasets using MUNIT [16], and DRIT [17], which uses the disentangled representations to map a paired set of images to a shared latent space and a non-shared content space.

A recent approach, UNIT-DDPM [18], improves on the previous methods by leveraging the Denoising Diffusion Probabilistic Models (DDPMs) to perform the unpaired image-to-image translations. The authors use auxiliary functions to pair up the unpaired dataset. Then, as given in equation (2.2), a set of forward diffusion equations maps the dataset to shared latent space. The training is performed to learn the parameters of the pairing functions and the reverse diffusion process. The pairing functions are only used during the training process. After the training, the reverse diffusion process, as in equation (2.4), with conditioning on the input image from either domain, is used to generate the samples from the other domain. Our approach to the unpaired task is strongly inspired by the method presented in this work. The critical difference between the authors' work and ours is that we do not employ pairing-up functions but instead utilize a set of reverse diffusion processes to pair up the unpaired images. Another difference is that we utilized a more general SDE-based approach discussed in section (2.2) instead of the DDPMs, to perform the forward diffusion and sampling.

3 Solution

In this paper, we propose a new formalism to perform the unpaired image-to-image translation task using score-based generative modelling with SDEs. We are given an unpaired dataset $\mathcal{D}$ consisting of images $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\mathbf{y}_i\}_{i=1}^N$. We want to determine the mapping functions $f : X \to Y$ and $g : Y \to X$ that generates the corresponding pairs $\{\mathbf{x}_i, \tilde{\mathbf{y}}_i\}_{i=1}^N$ and $\{\tilde{\mathbf{x}}_i, \mathbf{y}_i\}_{i=1}^N$. The mapping functions applied to the dataset approximates the conditional distributions $p(\tilde{\mathbf{x}}|\mathbf{y})$ and $p(\tilde{\mathbf{y}}|\mathbf{x})$, respectively. To determine the mapping functions, we adopt a score-based model [7,8], as discussed in section (2.1) for conditional

image generation [12,13]. We also show that our model can easily be adapted to perform paired image-to-image translations in the limiting case of paired datasets.

Unpaired: We begin by utilizing the EM samplers, as in equation (2.9), where the score function s_θ is initialized with a random set of weights, to generate the image pairs $\{\mathbf{x}_i, \tilde{\mathbf{y}}_i\}_{i=1}^N$ and $\{\tilde{\mathbf{x}}_i, \mathbf{y}_i\}_{i=1}^N$. Initially, $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{x}}_i$ are very noisy images, but as the model is trained with the objective function as in equation (2.8), the quality of the images generated by the EM samplers improves. To ensure that a given image $\mathbf{x}_i \in X$ is mapped only to a corresponding image $\tilde{\mathbf{y}}_i \in Y$, we incorporate a conditional score function within the EM samplers to input the domain-specific information. This limits the solutions to the reverse-time SDE in either of the two domains depending on the conditional information. The conditional score model is represented as

$$s_\theta = s_\theta(\cdot\ , \mathbf{y}, t), \tag{3.1}$$

where the second argument represents the conditioning, and the last argument represents the time.

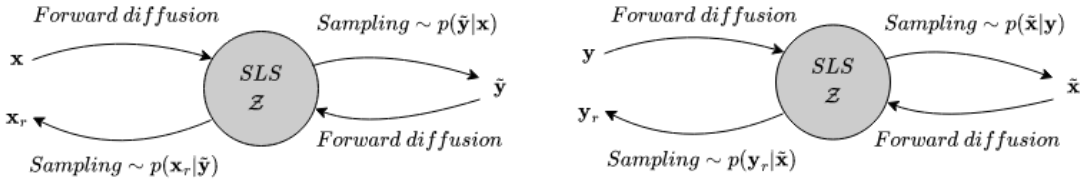

Figure 3: Unpaired image-to-image with SDEs. SLS denotes the shared latent space. The sampling denotes the reverse diffusion process.

Paired: Since the dataset is already paired in the paired task, the sampling step is not required during the training steps and only during inference. The conditional score function, as given in equation (2.1), is utilized to pass in the domain-specific knowledge. The rest of the procedure is similar to an unpaired translation task.

Throughout both the tasks, we made an implicit assumption of a shared latent space where we assumed that perturbing images in both the domains X and Y with the same noise levels $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ maps them to distinct points in a shared space of latent variables $\mathcal{Z}$ known as the shared latent space (SLS). This is a valid assumption for a paired set of images $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ since the distributions $p(\mathbf{x})$ and $p(\mathbf{y})$ can be considered to be approximately similar. Further, the forward SDE can always be designed by choosing appropriate diffusion and drift coefficients such that the images in both the domains are mapped to the terminal distribution, which is a Normal distribution.

3.1 Perturbing the Data

In order to learn the conditional distributions $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$, we initially perturb the images using the forward-time SDE

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}\ , \tag{3.2}$$

Algorithm 3 Training (Unpaired Task)

repeat
  $\mathbf{x}_0 \sim p_x(\mathbf{x}_0),\ \mathbf{y}_0 \sim p_y(\mathbf{y}_0)$
  $t \sim \mathcal{U}[0,1]$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
  $\tilde{\mathbf{y}} \sim\ p(\tilde{\mathbf{y}}|\mathbf{x}_0),\ \tilde{\mathbf{x}} \sim\ p(\tilde{\mathbf{x}}|\mathbf{y}_0)$
  $\mathbf{y}_r \sim\ p(\mathbf{y}_r|\tilde{\mathbf{x}}),\ \mathbf{x}_r \sim\ p(\mathbf{x}_r|\tilde{\mathbf{y}})$
  $\sigma_{\tilde{y}_t} \leftarrow \sigma(\tilde{\mathbf{y}}, t),\ \sigma_{y_t} \leftarrow \sigma(\mathbf{y}, t)$
  $\sigma_{\tilde{x}_t} \leftarrow \sigma(\tilde{\mathbf{x}}, t),\ \sigma_{x_t} \leftarrow \sigma(\mathbf{x}, t)$
  Take gradient step on

$$\nabla_{\theta_x,\theta_y} \Big[||\sigma_{\tilde{y}_t} s_{\theta_y}(\tilde{\mathbf{y}}, \mathbf{x}, t) + \mathbf{z}_y||_2^2 + ||\sigma_{y_t} s_{\theta_y}(\mathbf{y}, \tilde{\mathbf{x}}, t) + \mathbf{z}_y||_2^2 \\ ||\sigma_{\tilde{x}_t} s_{\theta_x}(\tilde{\mathbf{x}}, \mathbf{y}, t) + \mathbf{z}_x||_2^2 + ||\sigma_{x_t} s_{\theta_x}(\mathbf{x}, \tilde{\mathbf{y}}, t) + \mathbf{z}_x||_2^2 \\ \lambda_{cyc} L_{cyc}(\mathbf{x}, \mathbf{x}_r) + \lambda_{cyc} L_{cyc}(\mathbf{y}, \mathbf{y}_r)]$$

until converged

Algorithm 4 Sampling (Unpaired Task)

$\mathbf{x}_0 \sim p_x(\mathbf{x}_0),\ \mathbf{y}_0 \sim p_y(\mathbf{y}_0)$
$\tilde{\mathbf{x}}_T, \tilde{\mathbf{y}}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
for $t = 1, \cdots, \epsilon$ **do**
  $\mathbf{z}_x, \mathbf{z}_y \sim \mathcal{N}(\mathbf{0}, \mathbb{1})\ if\ t > \epsilon,\ else\ \mathbf{z}_x, \mathbf{z}_y = 0$
  $\tilde{\mathbf{x}}_{t-1} = \tilde{\mathbf{x}}_t + [f(\tilde{\mathbf{x}}_t, t) - g^2(t) s_{\theta_x}(\tilde{\mathbf{x}}_t, \mathbf{y}_0, t)]\,\Delta t + g(t)\sqrt{|\Delta t|}\mathbf{z}_x$
  $\tilde{\mathbf{y}}_{t-1} = \tilde{\mathbf{y}}_t + \Big[f(\tilde{\mathbf{y}}_t, t) - g^2(t) s_{\theta_y}(\tilde{\mathbf{y}}_t, \mathbf{x}_0, t)\Big]\,\Delta t + g(t)\sqrt{|\Delta t|}\mathbf{z}_y$
end for
return $\tilde{\mathbf{x}}_0, \tilde{\mathbf{y}}_0$

which gradually adds Gaussian noise according to a set noise schedule to the input images. The $\beta(t)$ corresponds to the noise-schedule function and is chosen to be $\beta(t) = \bar{\beta}_{min} + t(\bar{\beta}_{max} - \bar{\beta}_{min})$ for $t \sim \mathcal{U}[0,1]$ [7].

Since equation (3.2) adds Gaussian noise to the dataset, we can determine the mathematical form of the Gaussian perturbation kernel and calculate the mean and the variance $\sigma(t)$ of the given dataset. The calculated variance can then be utilized to determine the latent codes corresponding to the Markov chain $\{\mathbf{x}(t)\}_{t=0}^{1}$, which the score-model learns to denoise over several iterations. The forward diffusion equation essentially maps the image space of the two domains X and Y to points in a shared latent space $\mathcal{Z}$, as depicted in figure 3.

Note that the SDE in the above equation (eq. 3.2) corresponds to a forward process with a fixed variance as $t \to \infty$ and is therefore referred to as Variance Preserving SDE. Also, it was shown in [7] that in the discrete limit, the above equation yields the discrete Markov Chain of the DDPM as given in equation (2.2); therefore, the score-based diffusion models with SDEs are more general.

3.2 Sampling

To encode the conditional information of an image domain, we modify the reverse-time SDE for conditional sampling. This is achieved by modifying the UNet architecture to approximate the score function s_θ [12, 13]. For the unpaired task, we incorporate two samplers, one translating from $X \to Y$ and another from $Y \to X$, to generate the image pairings given the conditional information $\mathbf{x}_0$ and $\mathbf{y}_0$. We represent the score-model incorporated by sampler for domain translation $Y \to X$ and $X \to Y$ as s_{θ_x} and s_{θ_y} respectively.

The sampler begins with pure Gaussian noise $\tilde{\mathbf{x}}_T, \tilde{\mathbf{y}}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ and the conditionals $\mathbf{x}_0 \sim p_x(\mathbf{x}_0)$, $\mathbf{y}_0 \sim p_y(\mathbf{y}_0)$. It then iteratively refines the noise to generate samples $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}$ from the conditional distributions $p(\tilde{\mathbf{y}}|\mathbf{x}_0)$ and $p(\tilde{\mathbf{x}}|\mathbf{y}_0)$ respectively using the equations

$$\begin{aligned} \tilde{\mathbf{x}}_{t-1} &= \tilde{\mathbf{x}}_t - \beta(t)\left[\frac{\tilde{\mathbf{x}}_t}{2} + s_{\theta_x}(\tilde{\mathbf{x}}_t, \mathbf{y}_0, t)\right]\Delta t + \sqrt{\beta(t)|\Delta t|}\mathbf{z}_x \ , \\ \tilde{\mathbf{y}}_{t-1} &= \tilde{\mathbf{y}}_t - \beta(t)\left[\frac{\tilde{\mathbf{y}}_t}{2} + s_{\theta_y}(\tilde{\mathbf{y}}_t, \mathbf{x}_0, t)\right]\Delta t + \sqrt{\beta(t)|\Delta t|}\mathbf{z}_y \ , \end{aligned} \tag{3.3}$$

where Δt is an infinitesimal negative time step and $t \in [1, \cdots, \epsilon]$.

Now, the generated images $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are again passed as conditionals to the respective EM samplers to generate the reconstructed images represented as $\mathbf{y}_r$ and $\mathbf{x}_r$ respectively. This corresponds to sampling from the conditional distributions $p(\mathbf{y}_r|\tilde{\mathbf{x}})$ and $p(\mathbf{x}_r|\tilde{\mathbf{y}})$ respectively. The samplers essentially begin with points in the shared latent space $\mathcal{Z}$ and conditional images and, over several iterations, map it to the image space X or Y as per the conditioning, as illustrated in figure 3. The complete procedure followed by the EM samplers to generate samples from the conditional distributions both in the unpaired and the paired settings is presented in Algorithms 4 and 6, respectively.

3.3 Objective Function

In order to train the EM samplers to generate more accurate image pairings $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{y}}_0$ given the conditional images $\mathbf{y}_0$ and $\mathbf{x}_0$ respectively, we define an objective function for our conditional score function. Let L represent the objective of a conditional score-model with conditional $\mathbf{x}$. Then the objective function is defined as

$$L(s_\theta(\tilde{\mathbf{y}}, \mathbf{x}, t)) \propto \left[||\sigma_{\tilde{y}_t} s_\theta(\tilde{\mathbf{y}}; \mathbf{x}, t) + \mathbf{z}||_2^2\right], \tag{3.4}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ and $\sigma_{\tilde{y}_t} = \sigma(\tilde{\mathbf{y}}, t)$ corresponds to the variance calculated using equation (3.2).

Now for the given unpaired images $\mathbf{x}_0$ and $\mathbf{y}_0$, we generate four set of images $\tilde{\mathbf{x}}$, $\tilde{\mathbf{y}}$, and $\mathbf{x}_r$, $\mathbf{y}_r$ using equation (3.3) (procedure depicted in figure 3). Therefore, the complete objective function has four terms corresponding to the conditional score model, one for each image conditional $\mathbf{x}_0$, $\mathbf{y}_0$, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ respectively. In addition, we add a cycle-consistency loss to constrain the domain translation functions further. The cycle-consistency loss enforces that $\mathbf{x}_r \sim \mathbf{x}_0$ and $\mathbf{y}_r \sim \mathbf{y}_0$. The complete objective function for the unpaired task is given as

$$
\begin{aligned}
L_{unpaired} = & \; L(s_{\theta_y}(\tilde{\mathbf{y}}_0, \mathbf{x}_0, t)) + L(s_{\theta_y}(\mathbf{y}_0, \tilde{\mathbf{x}}_0, t)) \\
& + L(s_{\theta_x}(\tilde{\mathbf{x}}_0, \mathbf{y}_0, t)) + L(s_{\theta_x}(\mathbf{x}_0, \tilde{\mathbf{y}}_0, t)) \\
& + \lambda_{cyc} L_{cyc}(\mathbf{x}_0, \mathbf{x}_r) + \lambda_{cyc} L_{cyc}(\mathbf{y}_0, \mathbf{y}_r) \; ,
\end{aligned} \tag{3.5}
$$

where λ_{cyc} denotes a constant and L_{cyc} denotes the cycle-consistency loss. Algorithms 3 and 4 depict the pseudocode for the training and sampling using our proposed method.

Since, for the paired task, we are equipped with additional information in the form of the paired dataset, we can relax the cycle-consistency constraint. Also, we do not require to generate the image pairs, and therefore the objective function in (3.5) can be simplified to

$$
L_{paired} = L(s_{\theta_x}(\mathbf{x}, \mathbf{y}, t)) + L(s_{\theta_y}(\mathbf{y}, \mathbf{x}, t)) \; , \tag{3.6}
$$

where $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is a paired dataset. Th Algorithms 5 and 6 depict the pseudocode for training and sampling for a paired task.

Algorithm 5 Training (Paired Task)

repeat
  $\mathbf{x}_0 \sim p_x(\mathbf{x}_0)$, $\mathbf{y}_0 \sim p_y(\mathbf{y}_0)$
  $t \sim \mathcal{U}[0, 1]$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
  $\sigma_{x_t} \leftarrow \sigma(\mathbf{x}_0, t)$, $\sigma_{y_t} \leftarrow \sigma(\mathbf{y}_0, t)$
  Take gradient step on
    $\nabla_{\theta_x, \theta_y} \left[||\sigma_{x_t} s_{\theta_x}(\mathbf{x}, \mathbf{y}_0, t) + \mathbf{z}||_2^2 + ||\sigma_{y_t} s_{\theta_y}(\mathbf{y}, \mathbf{x}_0, t) + \mathbf{z}||_2^2 \right]$
until converged

Algorithm 6 Sampling (Paired Task)

$\mathbf{x}_0 \sim p_x(\mathbf{x}_0)$, $\mathbf{y}_0 \sim p_y(\mathbf{y}_0)$
$\tilde{\mathbf{x}}_T, \tilde{\mathbf{y}}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$
for $t = 1, \cdots, \epsilon$ **do**
  $\mathbf{z}_x, \; \mathbf{z}_y \sim \mathcal{N}(\mathbf{0}, \mathbb{1}) \; if \; t > \epsilon, \; else \; \mathbf{z}_x, \mathbf{z}_y = 0$
  $\tilde{\mathbf{x}}_{t-1} = \tilde{\mathbf{x}}_t + [f(\tilde{\mathbf{x}}_t, t) - g^2(t) s_{\theta_x}(\tilde{\mathbf{x}}_t, \mathbf{y}_0, t)] \, \Delta t + g(t) \sqrt{|\Delta t|} \mathbf{z}_x$
  $\tilde{\mathbf{y}}_{t-1} = \tilde{\mathbf{y}}_t + \left[f(\tilde{\mathbf{y}}_t, t) - g^2(t) s_{\theta_y}(\tilde{\mathbf{y}}_t, \mathbf{x}_0, t) \right] \Delta t + g(t) \sqrt{|\Delta t|} \mathbf{z}_y$
end for
return $\tilde{\mathbf{x}}_0, \tilde{\mathbf{y}}_0$

4 Results and Evaluation

4.1 Network Architecture

The UNet architecture incorporated in this paper to approximate the score function s_θ was adapted from [12,13]. The conditional information $\mathbf{y}_0$ is concatenated with the perturbed

images $\mathbf{x}_t$ along the channel dimensions. A single UNet model has 58M parameters, 2 ResBlock and channel multipliers of $1, 2, 4$, and 8. An attention block was also utilized following the first ResBlock to allow the spatial positions to attend to each other. Figure 4 depicts the architecture of the UNet model utilized in this work.

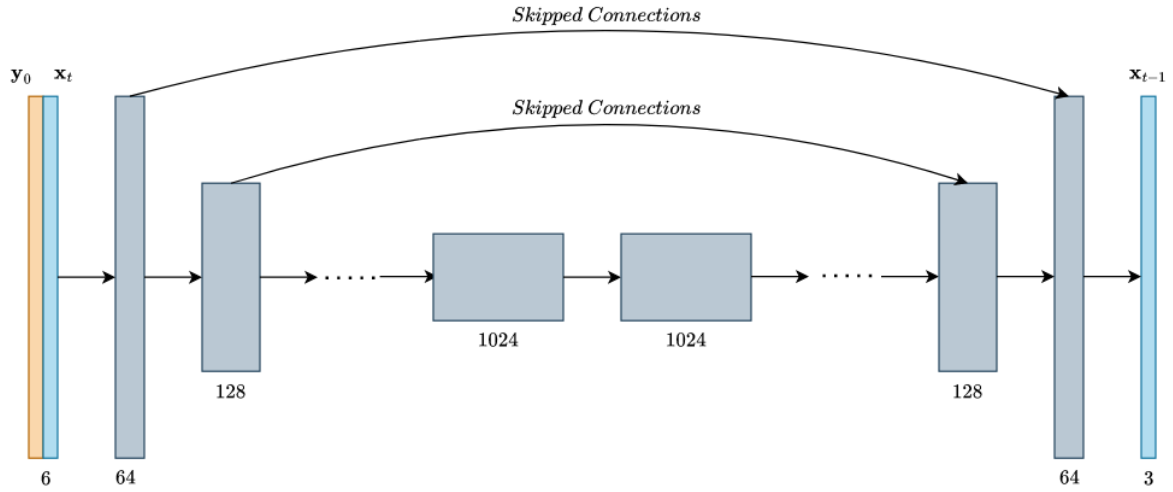

Figure 4: UNet Architecture with skip connections. The conditional image $\mathbf{y}_0$ is concatenated along the channel dimension. Therefore, the input to the UNet is an image with six channels.

4.2 Training details

Our unpaired and paired image-to-image translation models were trained on distinct paired and unpaired datasets with domains X and Y, respectively. Each domain consists of 128 images which were resized to $3 \times 64 \times 64$ and normalized to input to the model. Adam optimizer was utilized for training the model with an initial learning rate (LR) of $\eta = 10^{-7}$ found using grid search. The LR initially increases according to a warmup schedule for the first $n_{epochs} = 1000$ and then remains constant at a value $\eta = 10^{-4}$ as was suggested in [19]. The exponential decay rates of the Adam optimizer were chosen to be $(\beta_1, \beta_2) = (0.9, 0.999)$ as in [7]. The batch size was set to one, where a single batch consisted of two images in either domain. We utilized an NVIDIA GeForce RTX 2080 Ti GPU with 12GB memory available at the NCC, Durham University, to perform the training and inference. The parameters of the noise schedule function $\beta(t) = \bar{\beta}_{min} + t(\bar{\beta}_{max} - \bar{\beta}_{min})$, were fixed to be $\bar{\beta}_{min} = 0.1$ and $\bar{\beta}_{max} = 20$ as given in [7].

During the training, the EM samplers were utilized with gradient tracking for the unpaired task to allow for backpropagation through the sampling steps. The number of steps was chosen empirically to generate paired images and was set to $N = 10$. During the inference, the number of steps for the EM sampler was increased to $N = 1500$, and gradient tracking was disabled. An L2 loss, in the objective function (3.5), was chosen for the cycle consistency loss L_{cyc} with $\lambda = 10$ as suggested in [14]. For the paired task, the EM samplers were only utilized to generate the samples given the conditioning. The number of steps was chosen to be $n_{steps} = 1500$. Also, for numerical stability, the

computation of t was restricted to $t \in [\epsilon, 1]$, where ϵ was fixed to be $\epsilon = 10^{-3}$ as was suggested in [7].

4.3 Results & Discussion

In order to evaluate the performance of our unpaired image-to-image translation model, we use the unpaired horse2zebra dataset to train the model training. Here, the aim was to learn the domain translation from horse images to zebra images and vice versa. We compare our results against the baseline model of CycleGANs [14], which also utilizes an unpaired dataset for training using human evaluation. A qualitative comparison between the different methods is depicted in figure 5. The first column from the left represents the inputs for the two models. The second and the third columns depict the domain-translated images generated by CycleGANs and our method, respectively.

Figure 5: A qualitative comparison of different methods for the unpaired image-to-image translation on the horse2zebra dataset. The baseline models perform better at the task than our unpaired image-to-image translation model.

To validate our paired image-to-image translation model, we use the paired maps dataset, where the aim was to learn the domain translation between the satellite images and the street maps and vice versa. A qualitative comparison between our method and Pix2Pix [11], which uses conditional GANs, is depicted in figure 6. The leftmost column represents the input images to the model, and the rightmost column depicts the actual images in the other domains, respectively. The second and the third columns display the domain-translated images generated by Pix2Pix and our model, respectively. We also

report the Structural Similarity Index Measure (SSIM) [20], which measures the degree of structural similarity between two samples of images, between the images generated by our model and the actual ground truth images in figure 7 (left).

Figure 6: A qualitative comparison of different methods for the paired image-to-image translation on the maps dataset. Our model performs on par with Pix2Pix on the paired task.

To generate the samples for the paired task, we set $N = 1500$ EM sampling steps as discussed in section 4.2. The SSIM score was calculated to 0.98, which is comparable with the performance of Pix2Pix as was shown in [11]. Moreover, from figure 7 (left), we observe that the EM sampler can generate actual-looking images with much fewer sampling steps. To assess the performance of our paired model with fewer sampling steps, we calculate the percentage change in the SSIM by varying the number of sampling steps. The SSIM was calculated to be 0.94 and 0.98 for $N = 250$ and $N = 500$ steps, respectively. This corresponds to a percentage change of $\approx 4\%$. Therefore, conditional sampling can be performed with fewer steps without losing the overall structure of the generated images. This is in contrast with the unconditional sampling, which typically requires $\sim 1.5\text{k} - 2\text{k}$ steps to generate high-quality samples [21].

From figure 5, we observe that our model for the unpaired translation task falls short in performing the unpaired image-to-image translation task compared to the baseline models of CycleGANs. We observe that our model is unable to learn domain-specific knowledge. For example, on the rightmost column in figure 5, we observe that our model has captured the structure of the conditional images (inputs) but does not display any characteristics of the original domain (for instance, stripes in zebra). This is primarily due to the fewer steps ($N = 10$) chosen for the EM samplers during training. Further, from the SSIM calculated for the paired task (figure 7 (left)), we observe that with $N = 10$ steps, the SSIM score is negative, implying that the generated images are unable to capture the structure of the ground truth images. This implies that choosing $N = 10$

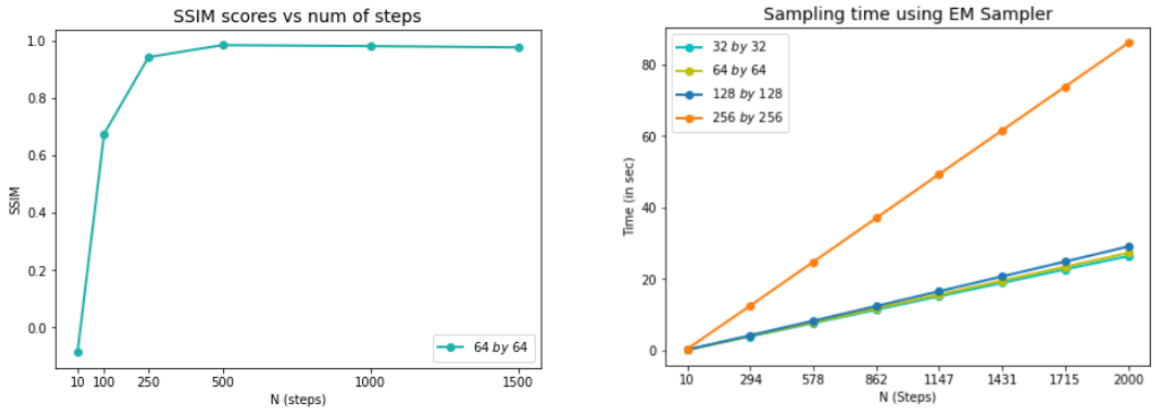

Figure 7: (**Left**) Plot depicting the SSIM score between the ground truth and the generated images as we increase the number of sampling steps. (**Right**) Plot depicting the sampling time for images of different sizes vs the number of sampling steps.

steps for training the unpaired model is insufficient for performing the image-to-image translations. When increasing N during training, the significant limitations were out of memory issues and time limitations.

From figure 7 (right), we observe that the sampling time increases linearly with an increasing number of steps and image size. Further, with the hyperparameters fixed as in section 4.2, a single training epoch takes ~ 5.5 minutes of wall time. Therefore, using samplers in the training step is a significant bottleneck for training the unpaired model. In the paired task, the inference step utilizes the samplers with gradient tracking disabled and, therefore, can sample images faster. However, the sampling is still slower than other methods, such as Pix2Pix [22].

Further, from figure 8 (left), we observe that the samplers consume more GPU memory for a given image size as the number of steps increases. The memory consumption quickly quadruples as the number of sampling steps increases from $N = 10$ to $N = 50$. For instance, sampling a 64×64 image with $N = 50$ steps inside a training loop allocates $\approx$ 800MB of GPU memory. The high memory consumption is primarily due to the gradient tracking required to backpropagate through all the sampling steps. For reference, Figure 8 (right) depicts the memory consumption when gradient tracking is disabled.

In addition, the paired image-to-image translation model provides a test for the shared latent space assumption made earlier in this work. The latent codes generated by the forward diffusion process for both the image domains X and Y inhabit a shared latent space $\mathcal{Z}$. From the shared latent space $\mathcal{Z}$, images in either domain can be generated by inputting the conditional information in the EM samplers. A key advantage of using the score-based conditional generative models with SDEs is that we do not have to perform auxiliary steps, such as weight sharing constraints to relate two VAEs, to enforce the shared latent space assumption [15]. The design of the forward SDE can always be fixed to map the images in the two domains to shared latent space. Only the perturbing noise levels must be chosen to be the same for both the image domains.

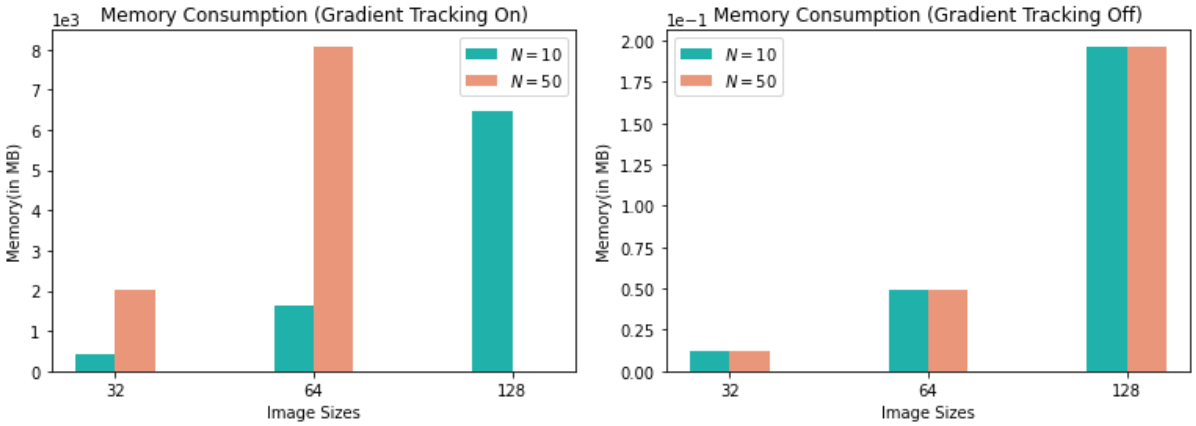

Figure 8: The memory consumption of an EM sampler with gradient tracking enabled **(left)** and disabled **(right)**. The blue bars depict ten steps, and the orange bars depict fifty steps.

4.4 Future Work

Analyzing the memory consumption and sampling time for the EM samplers, we observe that our model's sampling steps in the training loop are the primary bottleneck. Therefore, it would be imperative to alter our model such that the sampling functions are not called during training steps.

A possible method which does not require sampling within the training step can be achieved through controllable generation [7, 24]. Using Bayes' theorem, given a forward SDE as in equation (2.5), the corresponding reverse SDE in equation (2.7) can be modified to input conditioning. The conditional reverse-time SDE is given as

$$\begin{aligned} d\mathbf{x} &= \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|\mathbf{y})\right] dt + g(t)d\tilde{\mathbf{w}} \\ &= \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\left(\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x})\right)\right] dt + g(t)d\tilde{\mathbf{w}}, \end{aligned} \tag{4.1}$$

where $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ can be approximated by a trained score-based model $s_{\theta^*}(\mathbf{x}, t)$ for an unconditional reverse-time SDE as in equation (2.7). The term $\nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x})$ can be thought as providing a guidance during the conditional sampling to guide the reverse process to generate the corresponding image of $\mathbf{y} \in Y$ in the domain X. This term can be approximated by a separate model which can be trained to infuse domain-specific knowledge. For instance, in [23], the authors used the image labels to train a classifier $p(y|\mathbf{x}_t)$ to guide the sampling process to generate images with class labels y. However, the implementation detail of such models is intended for future work.

5 Conclusion

In summary, in this paper, we discussed a novel method to perform the image-to-image translation task with paired and unpaired datasets. The procedure is based on diffusion models where a forward-time SDE perturbs a given distribution of a set of images

at various noise levels. Simultaneously, a score function is trained to learn to denoise the perturbed images. Then a corresponding reverse-time SDE with the learned score function is employed to generate samples from the original data distribution. The score function approximates the gradient of the log probability of the intermediate distributions of the perturbed images and is parameterized by a UNet architecture.

For the image translation tasks, we utilized a conditional UNet architecture to adapt the reverse-time SDE for conditional sample generation. The conditional information is passed to the model by concatenating it along the channel dimensions. For the unpaired task, we began by pairing the unpaired images using instances of the EM samplers. Initially, the EM samplers generate poor-quality paired images; however, as the model is trained, the sampler gets better at the pairings, essentially learning the domain translation functions $f : X \to Y$ and $g : Y \to X$. The pairing procedure is not required for the paired task, yielding a more straightforward approach to learning the conditional distributions.

Further, we observed that our model for the unpaired task had a few shortcomings due to which it felt short in performing the translations from domain $X \to Y$ and vice versa. From our analysis, we observed that this was primarily due to the sampling steps within the training loops, which occupied a considerable size of the GPU memory, throttling the training. However, we also contemplate that the shared latent space assumption for the unpaired task might not be sufficient to constrain the domain translation functions. Further, we demonstrate our model on the paired task on the maps dataset. Our paired model achieves an SSIM score $\approx 98\%$ with $N = 1500$ sampling steps. We also suggest a few improvements that can be incorporated to improve the model training for the unpaired task by utilizing controllable generation and detaching the sampling steps within the training loops. However, the implementation of these remains the subject of future work.

Acknowledgements

I am thankful to my supervisor Dr Chris G. Willcocks, for the valuable comments, remarks and enormously fruitful discussions on the topic. This work has used Durham University's NCC cluster. NCC has been purchased through Durham University's strategic investment funds and is installed and maintained by the Department of Computer Science.

Supplementary material

Online supplementary material associated to the dissertation is available from the following sources:
horse2zebra dataset: http://efrosgans.eecs.berkeley.edu/cyclegan/datasets/horse2zebra.zip.
maps dataset: http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz.
PyTorch: https://pytorch.org/.
Conditional UNet architecture implementation: https://github.com/Janspiry/Palette-Image-to-Image-Diffusion-Models/tree/main/models.
The code in this paper is available on

- Paired Image-to-Image https://github.com/Ishan-phys/paired-via-sde.git.
- Unpaired Image-to-Image https://github.com/Ishan-phys/Unpaired-via-SDE.git.

References

[1] Ian J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *"Generative Adversarial Nets"*, Advances in neural information processing systems, 2014.

[2] D.P. Kingma and M. Welling, *"Auto-encoding variational bayes"*, International Conference on Learning Representations, 2014

[3] D.P. Kingma and P. Dhariwal, *"Glow: Generative flow with invertible 1x1 convolutions"*, Advances in neural information processing systems, 2018.

[4] J. Sohl-Dickstein, . Weiss, N. Maheswaranathan, and S. Ganguli. *"Deep unsupervised learning using nonequilibrium thermodynamics"*, International Conference on Machine Learning, pages 2256–2265, 2015.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel, *"Denoising Diffusion Probabilistic Models"*, Advances in Neural Information Processing Systems, 2020.

[6] Alexander Quinn Nichol and Prafulla Dhariwal, *"Improved Denoising Diffusion Probabilistic Models"*, Proceedings of the 38th International Conference on Machine Learning, 2021.

[7] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon and Ben Poole, *"Score-Based Generative Modeling through Stochastic Differential Equations"*, International Conference on Learning Representations, 2021.

[8] Yang Song and Stefano Ermon, *"Improved techniques for training score-based generative models."*, International Conference on Learning Representations, 2021.

[9] Peter E. Kloeden and Eckhard Platen, *"Numerical solution of stochastic differential equations."*, volume 23. Springer Science & Business Media, 2013.

[10] Brian D O Anderson, *"Reverse-time diffusion equation models."*, Stochastic Process. Appl., 12(3): 313–326, 1982.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, *"Image-to-Image Translation with Conditional Adversarial Networks."*, CVPR, 2017.

[12] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi, *"Image Super-Resolution via Iterative Refinement."*, arXiv:2104.07636, 2021.

[13] Chitwan Saharia, William Chan, Huiwen Chang, Chris A Lee, Jonathan Ho, Tim Salimans, David J Fleet, and Mohammad Norouzi, *"Palette: Image-to-Image Diffusion Models."*, arXiv:2111.05826v2, 2021.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, *"Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks."*, Computer Vision (ICCV), 2017 IEEE International Conference, 2017.

[15] Ming-Yu Liu, Thomas Breuel, and Jan Kautz, *"Unsupervised Image-to-Image Translation Networks."*, Advances in neural information processing systems, 2017.

[16] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz, *"Multimodal Unsupervised Image-to-Image Translation."*, Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[17] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang, *"Diverse Image-to-Image Translation via Disentangled Representations."*, Int J Comput Vis 128, 2402–2417, 2020.

[18] Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon, *"UNIT-DDPM: UNpaired Image Translation with Denoising Diffusion Probabilistic Models."*, arXiv:2104.05358, 2021.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *"Attention Is All You Need."*, Advances in neural information processing systems, 2017.

[20] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, *"Image Quality Assessment: From Error Visibility to Structural Similarity."*, IEEE Transactions on Image Processing, Vol. 13, no. 4, April 2004.

[21] Tim Salimans an Jonathan Ho, *"Progressive Distillation for Fast Sampling of Diffusion Models."*, International Conference on Learning Representations, 2022.

[22] Z. Xiao, K. Kreis, and A. Vahdat, *"Tackling the Generative Learning Trilemma with Denoising GANs"*, arXiv:2112.07804v2, 2022.

[23] Prafulla Dhariwal and Alex Nichol, *"Diffusion Models Beat GANs on Image Synthesis."*, Advances in neural information processing systems, 2021.

[24] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon, *"Solving Inverse Problems in Medical Imaging with Score-Based Generative Models."*, arXiv:2111.08005, 2021.