Investigating Explainability of Diffusion Model for Generation of Chinese Style Landscape Paintings

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Jiarong Li
School of Computer Science
University of Galway

Supervisor(s)
Dr. Matthias Nickles

In partial fulfillment of the requirements for the degree of
MSc in Computer Science (Artificial Intelligence)
July 30, 2023

DECLARATION I, Jiarong Li, hereby declare that this thesis, titled "Investigating Explainability of Diffusion Model for Generation of Chinese Style Landscape Paintings", and the work presented in it are entirely my own except where explicitly stated otherwise in the text, and that this work has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: Jiarong Li

Acknowledgement

I would like to take this opportunity to sincerely thank my project supervisor, Dr. Matthias Nickles, a cordial erudite lecturer. In my research and writing, his generous, cordial, patient and priceless help is indispensable to my success of completing this journey. He encourages to pursue my academic dream and generously gives me priceless support. I am so grateful to be supervised by him.

I also wish to thank the following lecturers: Dr. James McDermott, who taught us Optimisation, Programming and Tools for AI and Deep Learning; Prof. Michael Madden, who taught us Deep Learning; Dr. Colm O' Riordan, who taught us Information Retrieval; Dr. Conor Hayes, who taught us Data Visualisation; Dr. John McCrae, who taught us Introduction to Natural Language Processing; Dr. Paul Buitelaar, who taught us Introduction to Natural Language Processing and Advanced Topics in Natural Language Processing; Dr. Enda Howley, who taught us Agents, Multi-Agent Systems and Reinforcement Learning; Dr. Patrick Mannion, who taught us Agents, Multi-Agent Systems and Reinforcement Learning; Dr. Ihsan Ullah, who taught us Research Topics in AI and Machine Learning; Dr. Heike Schmidt Felzmann, who taught us Artificial Intelligence and Ethics; Dr. Brian Deegan, who taught us Digital Signal Processing and Embedded Image Processing. They are fabulous lecturers and I can never be at this point without their priceless help.

Abstract

Generative diffusion model is state-of-the-art in generating originated high-quality images. Differentiate from other generative models such as Auto-Encoder, Variational Auto-Encoder and Generative Adversarial Neural Networks which produce unoriginal images based on conditional input, diffusion model can generative images originate from latent space of probabilistic distribution rather than from conditional input. We apply a diffusion model trained from traditional historical Chinese style landscape paintings dataset to generate creative Chinese style landscape paintings. Furthermore, we dig into our diffusion model with Explainable AI tools to explain the diffusion process of our model in a human-understandable way.

Keywords: Diffusion Model, Explainable AI, Image Generation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Image generation is a crucial technique for limited artwork generation especially for valuable, historical paintings. Traditional Chinese style painting generation plays an important role in revisiting history by learning the main characteristics of traditional Chinese style landscape paintings from the perspective of the records of history. Chinese painting reflects the lights of various ages of the long Chinese history and culture and is known as one of the four arts of the Chinese scholar-official group [1].

The emergence of generative models sheds light on generating artworks. Prior works mainly leverage style transfer algorithms regarding Chinese style painting generation by recombining the style information and content information of two images or applying a text prompt as a condition to make descriptions of the image styles. However, these generative methods suffer from fail to achieve true machine originality in that they are not able to originate arts without conditional input. In light of the impressive advantages of diffusion model, the generated originated images are of good quality with high image synthesis understanding. Our research

Figure 1.1: Autumn Colors on the Que and Hua Mountains. Image from the China Online Museum (`https://www.comuseum.com/painting/landscape-painting`).

implements the Chinese landscape painting generation by leveraging the state-of-the-art diffusion model.

Explainable AI techniques are essential, especially for understanding the decision-making process by complex networks such as deep NNs. Although models such as role-based models, decision trees and linear models are intrinsically interpretable, they are less powerful to represent complex tasks such as image tasks compared with deep NNs.

The emergence of Explainable AI sheds light on making a deep model explainable from local interpreted to a global understanding of the network. Our research leverages Explainable AI tools to demystify the factors in the diffusion process within our diffusion model.

1.2 Our research

Our research implements a generation model to generate originated Chinese style landscape paintings by leveraging the state-of-the-art denoising diffusion probabilistic model. Moreover, we dig into the model to explain the generation process

with the help of Explainable AI tools.

Diffusion model exhibits state-of-the-art performance on image synthesis tasks. It consists of a tractable forward process which corrupts the clean image by gradually adding noise to the image on the previous timestamp and a denoising diffusion process which learns the reverse reconstruction. Our research questions regarding the diffusion model are:

- RQ1: Can diffusion model generate Chinese style landscape paintings?
- RQ2: Can diffusion process be visualised?
- RQ3: Can Explainable AI techniques help us to evaluate the performance?

1.3 Structure of the thesis

In Chapter 2, we introduce the background of our research including a brief history of Chinese Painting, attributes of Chinese style landscape paintings and image generation models (such as VAE, GAN, DDPM). In Chapter 3, we survey the techniques regarding image generation methods including image style transfer, generation models and explainable AI tools. In Chapter 5, we conduct a sufficient introduction of data and methodology. We address our experiments in Chapter 6, and analyse the results of our model in Chapter 7. We conclude our work and future work in Chapter 8.

Chapter 2

Background

2.1 Summary

Generation of Chinese style landscape paintings can be regarded as an image generation task by leveraging a generative model $p(x)$, which is learned from a complex input data distribution x to approximate the true distribution of the input x, then generate any related output from the learned complex distribution.

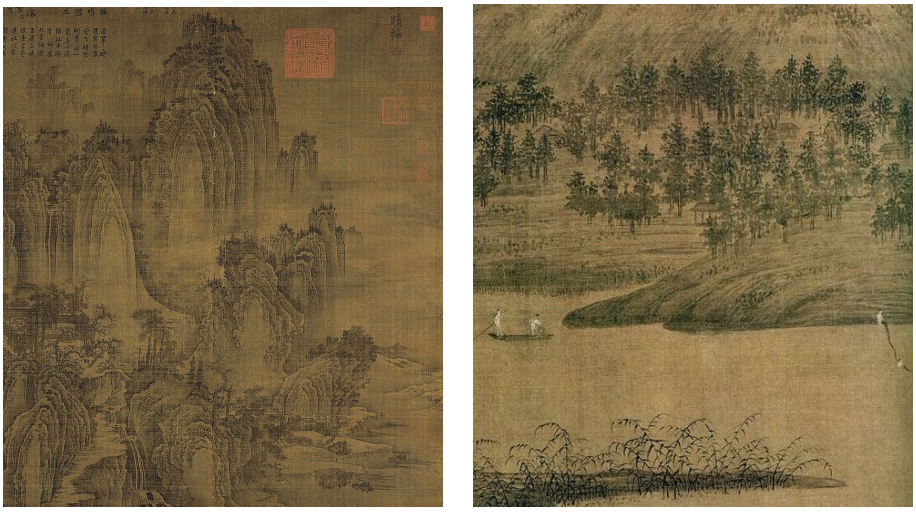
In this Chapter 2, we introduce traditional Chinese painting in terms of its history and patterns in Section 2.2, we introduce characteristics of Chinese style paintings in Section 2.3 and three classes of generative models – Variational Autoencoder in Section 2.4, Generative Adversarial Neuron Network in Section 2.5 and state-of-the-art Diffusion Model in Section 2.6 – at a high level.

2.2 A brief history of Chinese Painting

Chinese painting is important in Chinese history as it reflects the philosophy of Chinese people in understanding nature and society. The main painting technique Chinese Painting utilizes is done with a brush dipped in black ink or colourful

pigments. Bussagli characterises Chinese painting styles as follows: (1) is GongBi style which focuses on highly meticulous details of the objects and the viewings with different types of painting tools such as brushes and (2) is XieYi style which emphasises drawing the spirit of the objects, the emotions of the artists and the motion of the dynamic life.

Chinese landscape painting is mainly the Xieyi style as Figure 1.1 shows. The greatest age of Chinese landscape paintings is from the Five Dynasties period (907) to the North Song Dynasty (1127) period. The emergence of classical Chinese landscape painting styles such as the South style and the North style reflects the life and the emotions of the artists of different dynasties during such period. Figure 4.1(c) shows the North style of landscape painting which draws the landscape with strong black lines and sharp, dotted brushstrokes to suggest the rough mountains or rocks, whereas Figure 4.1(d) shows the artists in South style drawing the landscape with soft, rubbed brushwork to display the peaceful countryside.



(a) Mount Lu. Image from National Palace Museum, Taipei, Taiwan.

(b) Detail from Xiao and Xiang Rivers. Image from Palace Museum

Figure 2.1: Classical Chinese landscape painting styles. Images from the China Online Museum (`https://www.comuseum.com/painting/landscape-painting`).

2.3 Styles Characterization of Chinese Style Landscape Paintings

Styles in visual art represent a distinctive manner that permits a group of works into a related category [2]. From the perspective of art, styles are the visual appearances of the work. Such appearances of the work refer to the different groups of artists which are related to the period, the location, the training and the individual styles within the group. Paintings are a main form of visual art. Tosaki characterises that Chinese painting styles consist of (1) colour and tone, (2) non-traditional elements such as texture and materials, and (3) rhythm in form, melody and colouration allowing the creations to intervene.

From the perspective of computer vision, image style consists of high-level

attributes and low-level attributes. Low-level attributes include colours, painting materials and brushstrokes, whereas high-level attributes consider the semantic elements and precise object shapes [4].

2.4 VAE

Variational Autoencoder (VAE) [5] is a likelihood-based class of generative model as the architectural affinity of Autoencoder model (AE) [6]. VAE model has been proven with better generation performance than AE model does as it generates a distribution of latent space instead of a discrete vector as the output of the encoder and samples a vector from latent space to feed into the decoder [7]. VAE updates the model not only by learning the reconstruction loss values between input data and the output data but also updates the model by learning the regularization function, KL divergence, to calculate the similarity between the input data distribution and the output data distribution.

VAE consists of an encoder and a decoder. The encoder processes the input data by compressing the input data into a latent space distribution which contains low-dimensional vectors, whereas the decoder inversely learns the probability distribution of the input data from the samples of the latent space, and finally, the trained decoder (generator) can generate a new interpolate image from the approximate probabilistic distribution model by sampling an arbitrary latent vector from the learned latent space [5].

VAE models can generate interpolate images. Nevertheless, VAE suffers difficulty in high-resolution image generation. Generative Adversarial Network which we introduce in the next section addresses the difficulty of generating images with good quality [8].

2.5 GAN

Generative Adversarial Networks (GAN) [8] have been proven the outstanding performance in image generation tasks, especially in generating realistic images. Such a model follows an adversarial manner to improve and control the generated quality of the images during the training process.

Basic GAN architecture consists of a generating process and a discriminating process. The generating process includes a variational autoencoder that encodes the input image to a low-dimensional vector distribution, samples a latent vector from the latent space and decodes the latent vector into a predicted result which needs to be as close as the input data. The discriminator process is a classifier that binary classifies whether the generated output is the ground truth or not. The goal of the GAN model during the training process is to make the generator generate realistic images that can fool the discriminator classifying the generated images as the ground truth images [8].

GAN outperforms priors in generating high-resolution images. However, it suffers difficulties during the training process as the generator and discriminator components have to be trained simultaneously such that the training process can become very slow and hard to debug immediately when errors come out. It is also hard to handle the importance of both components as the power of the generator and the discriminator varies in different training stages.

2.6 Denoising Diffusion Probabilistic Model

Denoising Diffusion Probabilistic Model (DDPM) [9] is a state-of-the-art algorithm in generative models and it outperforms priors algorithms in generating tasks. DDPM is designed as a generative probabilistic model which consists of a forward diffusion process and a denoising diffusion process.

The goal of DDPM is to learn a noise predictor $\epsilon_\theta(x_t, t)$, where t represents the timestamp of the noise, and x_t represents the noisy image at the t timestamp. Such a noise predictor can be used to train the model and generate a probabilistic distribution of the approximate target data.

DDPM learns the noise predictor during the forward diffusion training process by calculating the loss between the added noise at a certain timestamp to a cleaned image and the predicted noise at such timestamp and applying gradient descent technique to update the noise predictor. The denoising diffusion process generates the approximate target image reversely from a pure noise image by denoising the predicted noise at the corresponding timestamp with the learned noise predictor and adding sampled uninformed noise iteratively until the timestamp becomes 1.

The aforementioned processes make diffusion model benefit not only from the advantages of an autoregressive model (forward diffusion process) but also from a non-autoregressive model (denoising diffusion process).

Therefore, we conduct an image-generation diffusion model that generates creative Chinese style landscape images thanks to the advantages of DDPM. We propose that the diffusion process can be visualized by harnessing explainable AI tools.

Chapter 3

Related Work

3.1 Summary

Image generative model gains tremendous attention in both academia and industry recently. Prior researchers propose various algorithms for solving the problems of different tasks of image generative models.

In Chapter 3, we first survey image generative models in Section 3.2. In Section 3.3, we introduce some prior works in terms of image generation tasks by harnessing image style transfer technologies from example-based image stylization to text-based image stylization to the methods with GANs and Diffusion models. In Section 3.4, we conduct the previous work regarding Explainable AI tools.

3.2 Image Generation

Image generation is crucial and gains huge attention in both academia and industry. It is one of the applications of generative models. The emergence of generative models such as VAE, GAN and Diffusion model sheds light on the evolution of generating high-quality images. Goals of such generative models are

to learn a model that can capture the entire distribution of a complex, unseen dataset.

VAE: is one of the first generative models. Standard VAE [5] encodes input as a distribution over latent space which tackles overfit issues in image generation from AutoEncoder model [6]. However, there are a number of issues associated with standard VAE such as poor approximation for image generation tasks and lack of representation of complex datasets. To circumvent these issues, Pixel-VAE [10] models details of data distribution well by learning compressed latent codes, Importance-weighted VAEs (IWAE) [11] increase the flexibility of modelling complex data distribution by using strictly tight log-likelihood lower bound summarised from important weights. Nevertheless, VAE models still have difficulty capturing details from complex data distributions which leads to failures of generating the best results.

GAN: has been popular in terms of image generation tasks thanks to its good performance in learning implicit distributions of unknown densities. It has been widely undertaken in generative tasks such as Sketchygan [12] of generating realistic photographs, Image-to-Image Translation [13], Super Resolution [14], Video Prediction [15], and 3D Object Generation [16, 17]. Although a good number of research papers have recently appeared, the application of GAN to generative tasks still lacks well defined metrics to track progress which results in difficulty working with GAN.

Diffusion Model: is a new state-of-the-art algorithm in a wide range of generative tasks and outperforms other generative models in terms of text-to-image generations. GLIDE [18] and Imagen [19] leverage CLIP [20] guidance to generate photorealism images directly from high-dimensional pixel levels. Stable Diffusion [21], VQ-diffusion [22] and DALL-E 2 [23] methods compress input into a low-dimensional latent space and train a diffusion model from such latent space.

In our thesis, we apply an end-to-end diffusion model to generate Chinese style landscape paintings without conditional human input as guidance.

3.3 Image Style Transfer

Image style transfer has been regarded as a model of artistic image generation following a certain painting style. It has been widely studied from Example-based image stylization to Text-based image stylization to the methods applying GANs and Diffusion models.

Example-based image stylization sheds light on making the artistic effect for an image by simulating the style of one image and applying the simulated style to generate a new image with different content. Prior works solve the problem of finding the mappings of the local pixel patches of the source image to those of the stylized images [24] by leveraging patch-based methods to make the content information and style information aligned with the low-level hand-crafted features [25, 24, 26, 27]. However, Example-based image stylization suffers from the restriction of generality and usability as the example image dataset is fixed.

Text-based image stylization extends the application scenario by using text-guided constraints instead of an example image which provides the style information. The essential problem of text-based stylization is to find a mapping of style features in different modalities of data. Patashnik et al. propose a method called StyleClip which applies pre-trained deep convolutional neuron networks with the learned probabilistic distribution of features which helps the model in capturing the style patterns effectively. However, StyleClip suffers inefficiency problems as the model has to be retrained when there is a new text input. LDAST [29] recently proposed to jointly extract the style information and content information by two separate encoders. Nevertheless, LDAST requires

paired image-text data during the training process so it is hard to obtain the training dataset and the performance is not satisfied. ITstyler [30] alleviates the problem by converting the text input to the style space of the pre-trained VGG network and achieves a good performance.

GAN for image style transfer. Generative Adversarial Networks (GAN) have been proven the effective expressive power in image synthesis with style-based generative models. Different from the other works that focus on the enhancement of the discriminator which has been used to train a generator in GAN, Karras et al. propose StyleGAN which focuses on the understanding of the generator by extending the architecture of GAN with a mapping network generating an intermediate latent space and a noise layer. Such intermediate latent space helps to process disentangled style features. Inspired by the advantages of disentanglement properties from such work, there has been a rich amount of works utilising the pre-trained model to process a wide variety of images [32, 33, 34]. However, it is difficult for the GAN model to optimize and it struggles to capture the full data distribution.

Diffusion model for image style transfer. Diffusion model is state-of-the-art and outperforms the previous models in image synthesis. It learns the data distribution by adding noise to the original image and gradually denoise a normally distributed variable to achieve high-level sample quality [21]. However, since the diffusion model operates directly on pixel space, it suffers from large computation and GPU requirements. Latent diffusion model (LDM) tackles this problem by applying the diffusion model in the latent space of a pre-trained autoencoder to reduce the complexity and achieve quality results with less computation [21]. Nevertheless, prior studies use simple-modality input as guidance which limits the expressiveness and diversity of the model. Multimodal guided artwork diffusion (MGAD) solves this problem by leveraging multimodal prompts

as guidance with a classifier-free diffusion model [35]. However, such models are still difficult to summarize the imagery characteristics which are hard to be adequately described by normal text. To tackle this problem, an inversion-based style transfer method (InST) has been proposed by learning the key features of the images efficiently and transferring the style of an image [4].

3.4 Explainable AI

In the last few years, the application of machine learning and deep learning approaches has attracted progressive attention of researchers and practitioners which results in requirements of explainable approaches in ML and DL algorithms becoming a tremendous surge. Explainable AI methods can be regarded as local explanation methods (attempts to explain a specific decision) or global explanation methods (attempts to explain the whole model) based on the purpose of explanation.

Model-specific interpretability. Some machine learning models such as decision trees [36], rule-based models [37] and linear models [38] are intrinsically interpretable. The decision-making process of such a model is tractable and reasonable. For instance, the importance of features can be acknowledged by weights assigned to them when applying a linear model. However, the interpretable machine learning model is not powerful in solving complex problems compare with the deep learning algorithm. Furthermore, computation and complexity of the model can be huge as the problem becomes complex.

Model-agnostic explainability. In order to tackle the aforementioned problem that an interpretable machine learning model which is strongly interpretable but less powerful can not be applied to complex tasks, recent works [39, 40, 41] focus on studying explainable approaches in deep convolutional neuron

networks from various perspectives and we introduce the related works from such perspectives below.

Gradient-based methods. Given a trained model, gradient information can be accessed by back-propagating the output score (logit after the softmax layer) to the extracted features or the input and can be applied to estimate the attribution score with respect to the extracted features or the input data [42, 43]. However, generated attribution map is always noisy due to the backpropagation. Various works propose alternative approaches in tackling the noise problem generated with an attribute map. SmoothGrad [44] handles noise problems by adding random noise to the input to get different attribute maps, then averages the attribute maps. Integrated Gradients (IG) [45] approach reduces noise by averaging the element-wise manipulation of gradient and input.

Perturbation-based methods. In order to determine the importance of each pixel, these methods [46, 47, 48] perturb input features by masking arbitrary pixels or setting arbitrary pixels as different values, then record the effect of such changes on the output by multiple passes which leads to high complexity in computations. Local Interpretable Model-agnostic Explanations (LIME) [49] explain predictions of any machine learning classifiers by learning an interpretable model to locally mimic the behaviours of an uninterpretable model. Shapley Additive Explanations (SHAP) [50] increase the transparency of the model by proposing a unified framework for interpreting a prediction to determine the contribution or importance of each feature based on cooperative game theory.

Chapter 4

Data

4.1 Summary

In Chapter 4, we conduct a characterization of our dataset - historical Chinese style landscape painting - and introduce the original open-source dataset of Chinese landscape paintings in Section 4.2. We depict the principles of dataset collection and creation in Section 4.3. We demonstrate our methods of pre-processing data and create a dataset in Section 4.4. In Section 4.5, we state the specifications of our dataset.

4.2 Characterization of Chinese Style Landscape Painting

Valuable Chinese style paintings which are drawn by individual painters with good reputations [51] play an important role in Chinese style painting generation tasks. Therefore, we apply the historical Chinese landscape paintings which are drawn by famous painters with high reputations as our training and evaluation data.

Chinese style landscape painting is one of the most crucial parts of the history of Chinese painting [51]. As Figure 4.1 shows, Chinese style landscape painting usually includes different combinations of mountains, lakes, trees and people. Hence, we limit the scope of the data chosen into historical Chinese style landscape paintings instead of Chinese style paintings.

The dataset we use is chosen from an open-source dataset from [52]. Such dataset consists of $2,192$ historical Chinese style landscape paintings which are sized 512×512 with high quality. The principles we follow to create and apply for the training dataset will be mentioned in Section 4.3.

(a) Landscape with people, trees and mountains. Image from Princeton University Art Museum.

(b) Landscape with lake, trees, people and mountains. Image from Harvard University Art Museum.

(c) Landscape with mountains. Image from Harvard University Art Museum.

(d) Landscape with trees, lake and rocks. Image from Metropolitan Museum of Art.

Figure 4.1: Historical Chinese style landscape paintings. Images from the open source dataset (`https://github.com/alicex2020/Chinese-Landscape-Painting-Dataset`).

4.3 Principles of Dataset Creation

We have conducted a characterization of historical Chinese style landscape painting of our research and mentioned the open source dataset that we create the

training dataset from in Section 4.2. In order to satisfy the aforementioned characterization, we create our training dataset following principles of data collection and principles of quality training data creation.

Principles of data collection. We follow the principles from Berne Convention [53] and the regulation of Europe GDPR [54].

Principles of quality training data creation. We summarize the principles of creating our quality training dataset from the aforementioned characterization of Chinese style landscape painting based on the aforementioned open source dataset:

- **Diversity:** Diversity is crucial in creating a quality dataset. We manually remove the duplicated images and ensure the balance of the data.
- **Reliability:** We manually remove the low-quality images.
- **Consistency:** We ensure images in the training dataset with the same size and number of channels. We also normalise data to $[-1, 1]$ range of values, and we will introduce methods in detail in Section 4.4.

4.4 Methods of Processing Data

We follow the aforementioned principles in Section 4.3 to create a dataset. We set the resolution of our dataset to 64×64 pixels, due to the huge computation and the high complexity of the diffusion model we apply in the experiment. Furthermore, we convert all training images to grayscale images with one channel as the images from the original dataset have different numbers of channels.

4.5 Specifications of Our Dataset

Table 4.1: The specifications of Our Dataset.

Dataset	Chinese Style Landscape Painting Dataset
# of images	2180
# of image channels	3
# of image size	64

Chapter 5

Methodology

5.1 Summary

In this Chapter 5, we conduct a sufficient depiction of the methodology we apply in our research. We also justify why we leverage some techniques by answering the aforementioned research questions from Section 1.3 in Chapter 1.

We summarise our considerations in Section 5.2 and apply the diffusion model to answer the research question, RQ 1.2. We depict the architecture of a core component, noise predictor, in our diffusion model in Section 5.3. We introduce the training process of our diffusion model in Section 5.4. We state the sampling process of generating new images from the predicted data distribution of our diffusion model in Section 5.5. In Section 5.6, we propose an explanation process for our diffusion model to make it human understandable and we state our tricks to answer the research questions RQ 1.2 and RQ 1.2 from Section 1.3 of Chapter 1.

5.2 Considerations

Diffusion model can generate new quality images from the distribution of latent space by sampling a noisy image from the distribution, then denoising it and adding a weighted random noise to the weighted denoised image. We depict our considerations of harnessing such a model in this Section 5.2 to justify the choice of algorithms and tools instead of directly introducing algorithms of our model.

We leverage the state-out-the-art diffusion model in this thesis to generate new Chinese style landscape paintings. We mentioned that a diffusion model consists of adding noise process and denoising process which correspond to the forward process and diffusion backward process as Figure 5.1 shows. The idea behind our diffusion model is that:

- **Forward process:** We feed a clean image without noise into our diffusion model gradually adding noise at each timestamp t until we get a pure noisy image and store the timestamp.
- **Diffusion backward process:** We reverse the timestamp and reduce the noise corresponding to each timestamp to obtain the original clean image.

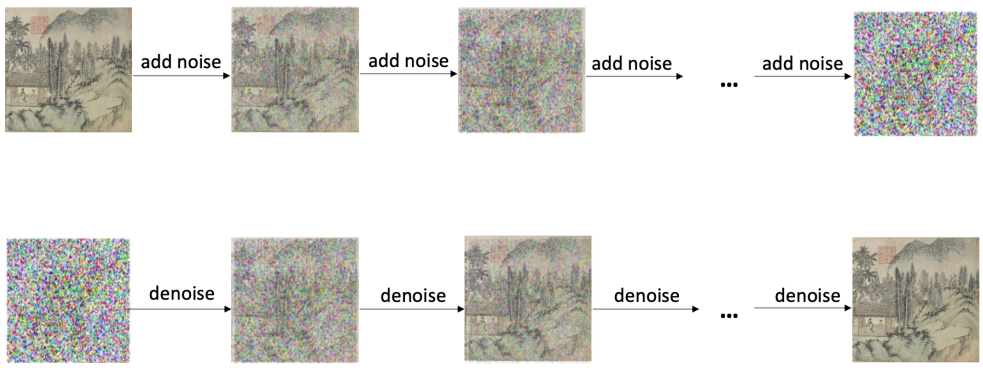


Figure 5.1: The directed graphic consideration of our diffusion model in the training process. The first row shows a forward process of adding noise to the previous image until we get a pure noise image. The second row shows a diffusion backward process of denoising from a noisy image.

We summarise the considerations and tricks we apply in this thesis in the forward diffusion process and the backward diffusion work process to obtain satisfied performance of Chinese style landscape painting generation:

- **Noise predictor;**
- **Position Embeddings;**
- **Attention layer;**
- **Wide ResNet block with weight-standardized convolutional layer.**

Noise predictor: Noise predictor is a core component in our diffusion model. We can generate a desired image from a pure noise image by reducing the weighted predicted noise output from the noise predictor. The resolution of the output noise requires identity of the resolution of the input image. Hereby, we apply U-Net[55] architecture as the noise predictor.

Position embeddings: Position embeddings are fundamental parameters of a diffusion model. We have mentioned that each noise corresponds to a certain timestamp in the diffusion model. Inspired by Transformer[56], we apply the tricks of sinusoidal position embeddings to encode each timestamp.

Attention layer: Attention layer is also essential in our diffusion model and is placed between convolutional blocks of the U-Net noise predictor. The attention layer can help the neural network focus on relevant positions of input. We employ two variants of attention techniques which are multi-head attention[56] and linear attention[57].

Wide ResNet block with weight standardized convolutional layer: Diffusion model is slow in sampling as the high complexity of mapping from a simple Gaussian noise distribution to a multimodal data distribution[58]. Our diffusion model tackles this problem by employing wide Resnet blocks[59] in the

U-Net model. We also apply a weight-standardized convolutional layer[60] in the wide Resnet block to improve the performance in combination.

5.3 Architecture of Noise Predictor

Noise predictor plays an important role in our diffusion model as it predicts noise, with respect to a corresponding timestamp, which is a core component in both training process and sampling process. We employ U-Net architecture for noise predictor as predicted noise should have the same size as the input image and the residual connection of U-Net between encoder and decoder can solve the problem of gradient vanishing[61].

Figure 5.2 shows the graphic architecture of our U-Net noise predictor. Table 5.1 shows the configuration of our U-Net noise predictor. Table 5.2 shows the configuration of the Residual function. Table 5.3 shows the configuration of the Resnet Block.

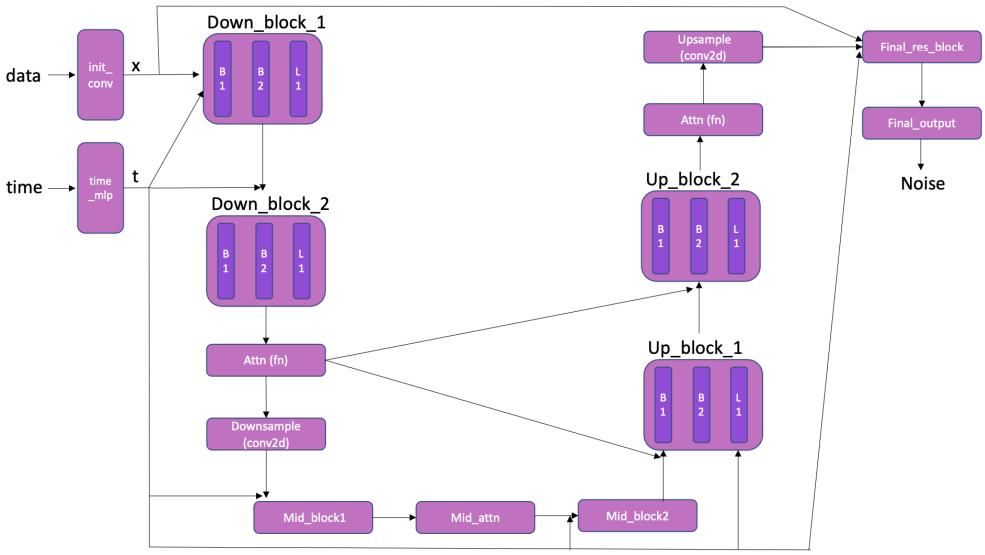


Figure 5.2: The overview of U-Net architecture. The pipeline consists of normalization, residual, position embedding, attention layer, down sampling and up sampling. This U-Net model predicts noise for training and sampling in the diffusion process.

Module Name	Module Type	Sub-Module Type	Layers and Parameters
init_conv	Conv2d		$Conv2d(3, 16, kernel\_size = (7, 7), stride = (1, 1), padding = (3, 3))$
time_mlp	**Sequential**		$SinusoidalPosEmb()$
			$Linear(in\_features = 16, out\_features = 64, bias = True)$
			$GELU(approximate =' none')$
			$Linear(in\_features = 64, out\_features = 64, bias = True)$
downs	**ModuleList**	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Sequential	$Rearrange('b\, c\, (h\, p_1)(w\, p_2) \to b\, (c\, p_1\, p_2)\, h\, w', p_1 = 2, p_2 = 2)$
			$Conv2d(64, 16, kernel\_size = (1, 1), stride = (1, 1))$
	ModuleList	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Sequential	$Rearrange('b\, c\, (h\, p_1)(w\, p_2) \to b\, (c\, p_1\, p_2)\, h\, w', p_1 = 2, p_2 = 2)$
			$Conv2d(64, 32, kernel\_size = (1, 1), stride = (1, 1))$
	ModuleList	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Conv2d	$Conv2d(32, 64, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
mid_block1	**ResnetBlock**		
mid_attn	**Residual**		
mid_block2	**ResnetBlock**		
ups	**ModuleList**	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Sequential	$Upsample(scale\_factor = 2.0, mode =' nearest')$
			$Conv2d(64, 32, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
	ModuleList	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Sequential	$Upsample(scale\_factor = 2.0, mode =' nearest')$
			$Conv2d(32, 16, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
	ModuleList	**ResnetBlock**	
		ResnetBlock	
		Residual	
		Conv2d	$Conv2d(16, 16, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
final_res_block	**ResnetBlock**		
final_conv	Conv2d		$Conv2d(16, 3, kernel\_size = (1, 1), stride = (1, 1))$

Table 5.1: The configuration of the Unet.

Function Name	Function Type	Layer Name	Inner Layers	Parameters
PreNorm	**fn**	**LinearAttention**	to_qkv	$Conv2d(16, 384, kernel\_size = (1, 1), stride = (1, 1), bias = False)$
			to_out	$Conv2d(128, 16, kernel\_size = (1, 1), stride = (1, 1))$
				LayerNorm()
	norm	LayerNorm()		

Table 5.2: The configuration of the Residual function.

Name	Type	Layers and Parameters
mlp	Sequential	$SiLU()$
		$Linear(in\_features = 64, out\_features = 32, bias = True)$
block1	Block	$WeightStandardizedConv2d(16, 16, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
		$GroupNorm(8, 16, eps = 1e-05, affine = True)$
		$SiLU()$
block2	Block	$WeightStandardizedConv2d(16, 16, kernel\_size = (3, 3), stride = (1, 1), padding = (1, 1))$
		$GroupNorm(8, 16, eps = 1e-05, affine = True)$
		$SiLU()$
res_conv		$Identity()$

Table 5.3: The configuration of the Resnet Block.

5.4 Training of Diffusion Model

Training process of the diffusion model is to mainly train our noise predictor, as Figure 5.1 shows. Figure 5.3 displays the workflow of the training process of our diffusion model. The training process consists of a data loader, a normalization layer, a function of forward sampling, a U-Net noise predictor and a loss function.

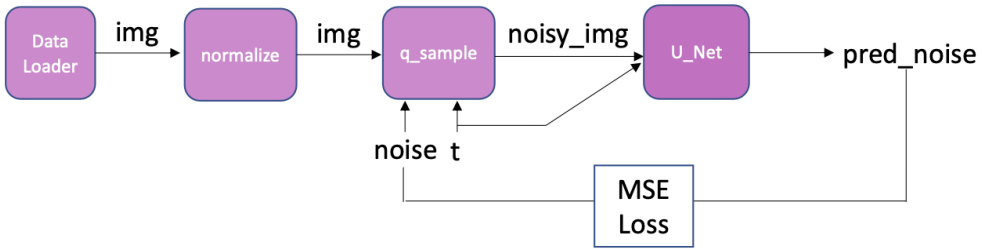


Figure 5.3: The overview of training workflow during the diffusion process. The pipeline consists of a data loader, normalization, noisy image sampling, U-Net and MSE loss function. The workflow uses MSE loss functions to minimize the distance between predicted noise and real noise for the training process.

Data Loader: We harness the data loader package from PyTorch. We set the arguments of the data loader as:

- *dataset = OurDataset:* where to load the data[62];
- *batch_size = 16:* how many samples per batch to load[62];
- *shuffle = True:* to have the data reshuffled at every epoch[62];

- *pin_memory = True:* the data loader will copy Tensors into CUDA pinned memory before returning them[62];
- *num_workers = cpu_count:* how many subprocesses to use for data loading[62].

Normalization: We normalize the values of each image to $[-1, 1]$ before sampling and feeding into the U-Net.

Forward Sample function: We sample a noisy image based on the previous image and corresponding timestamp. The equation shows the forward sampling process below:

$$\sqrt{\bar{a}_t}x_0 + \sqrt{1 - \bar{a}_t}\epsilon$$

where x_0 denotes a clean image without noise, t denotes timestamp, and ϵ represents noise.

Loss function: We identify the MSE loss function to minimize the distance between predicted noise from the U-Net model and the real noise at a particular timestamp during the training process.

5.5 Sampling Process

Sampling process of our diffusion model can be identified as a process of new image generation. Figure 5.4 shows the main idea of such sampling process. We first sample a pure noise image from a Gaussian distribution with standard deviation $= I$. Then we literately feed the noise image and a particular timestamp t into the U-Net model to predict corresponding noise and reduce such predicted noise from the image at the previous timestamp until we obtain an image which looks like sampling from the real dataset distribution. It is well noted that we also add a random noise at each sampling to generate quality new images.

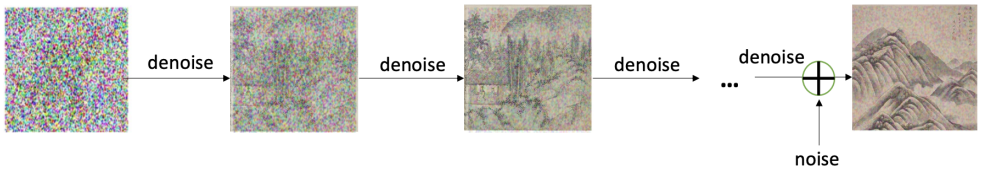


Figure 5.4: The directed graphic consideration of our diffusion model in the image generation process.

As Figure 5.5 shows, the sampling process consists of a U-Net model, a function of predicting a start image from a noise image, a function of finding a distribution of our training data and a function of sampling a generated image from such distribution.

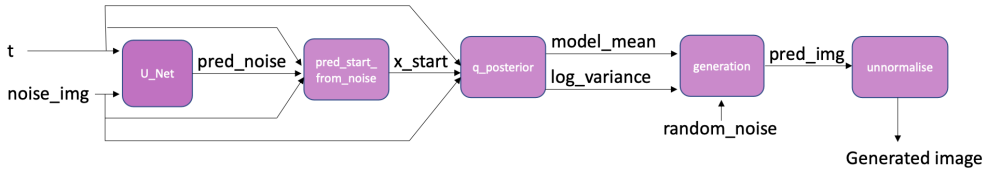


Figure 5.5: The overview of sampling workflow during the diffusion process. The pipeline consists of a U-Net, a function of predicting a start image from noise, a function of predicting noise from a start image, a function of calculating the mean and variance of model distribution and a function of generating a new image from the distribution. The workflow of sampling during the diffusion process delivers state-of-the-art performance.

Function of predicting a start image from noise: This function is to predict a start image by asking the arguments which include a particular timestamp, the predicted noise and the random image sampled from the Gaussian distribution. Such function can be formulated as follows:

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}}$$

where x_t represents a noise image, t represents a particular timestamp and ϵ denotes noise.

Function of finding a distribution of training data: This function predicts a data distribution which is as close to the real data distribution as possible.

We calculate the mean of this distribution and employ fixed variance for this distribution. Calculation of mean of the predicted distribution can be formulated as follows:

$$\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t}{1-\bar{\alpha}_t}$$

where t represents a particular timestamp, x_t represents a noise image and x_0 denotes a start image.

Function of sampling new images: This function samples new images from the predicted data distribution by reducing the predicted noise from noise image conditioned with a particular timestamp to get the mean of the distribution and adding a random noise sampled from the Gaussian distribution to generate a new quality image from the predicted data distribution. The sampling process can be formulated as follows:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$$

where x_t represents the sampled image from the predicted distribution, x_t represents a noise image, ϵ_θ denotes a noise predictor, t represents a particular timestamp, and z denotes a random noise sampled from Gaussian distribution.

5.6 Explanation Process – Saliency Map

We conduct an explanation process in our research to make the generation process human-understandable. We leverage the idea of a saliency map, which is one of explainable AI tools. Saliency maps highlight important pixels which can largely affect decisions with lease changes by calculating gradients with respect to input images[42].

Noise predictor, as aforementioned, is the core component of our diffusion model and plays an important role in predicting data distribution and generating new images from such distribution. Based on such noise predictor, our diffusion model calculate the loss between the noise predicted by the noise predictor and the real noise with a MSE loss function and leverage such calculated loss values for the training process. Furthermore, our diffusion model obtains the predicted distribution and samples new images from such distribution in the sampling process.

We probe into the loss function of our diffusion model with our explanation function in this research. Tricks we apply for show as follows:

Normalization: We first normalise the input image for speeding computations.

Data processing: We load data onto a GPU device to make computations fast. We also call *requires_grad_()* on the data as we need to find the gradient with respect to input data.

Load Diffusion model: We load our trained diffusion model and feed the input data into the trained model to obtain the loss values of predicted noise and real noise.

Gradients calculations: We call the backward function on loss values with respect to input data.

Create saliency maps: We create saliency maps which are gradients with respect to the images. We take the maximum magnitude of all channels to create a single saliency value for each pixel as the input images are with three channels.

Visualise and save derived saliency map: We visualise and save 1,000 saliency maps with respect to the input images.

Chapter 6

Experiments

6.1 Summary

We depict our experiments in detail in this Chapter 6. We state our experimental settings in Section 6.2. We present training details of our diffusion model in Section 6.3. We introduce sampling details of our diffusion model in Section 6.4. We propose our explanation details of the core components, the noise predictor, which is fundamental in both training and sampling process in Section 6.5. We justify the performance of our diffusion model with Visual Turing Test which is introduced in Section 6.6.

6.2 Test Bed

Our experimental settings are shown in Table 6.1. We conduct training, sampling and explanation on a Kaggle platform which is an online data science platform providing powerful resources to achieve data science goals. We develop our models on a Kaggle Notebook which equips with Intel Skylake CPU which has 4 CPU cores and 30 GB of RAM, and 1 Nvidia Telsa P100 GPU which has 2 CPU

cores and 13 GB of RAM. Our experimental environments runs with Python version 3.8.8. Our deep learning framework inside the experimental environment is PyTorch with version 1.12.1.

We follow the principles of scientific reproducibility and set the random seeds to 4096 to make our experiments reproducible.

Operating System Name	GNU/Linux
CPU Specifications	4 CPU cores
	30 GB of RAM
GPU Specifications	1 Nvidia Telsa P100 GPU
	2 CPU cores
	13 GB of RAM
Memory	20GB
Python version	3.8.8
PyTorch version	1.12.1
Random seed	4096

Table 6.1: Experiment environment

6.3 Training Details

Training of our diffusion model is implemented by literately calculating the gradient of loss values which is the distance between the predicted noise value and ground-truth noise value with respect to input images until the gradient convergences or corresponding timestamp becomes zero.

We set each input image size $64 * 64$ pixels for training. Our diffusion model is trained for $10,0000$ epochs with batch size equals to 16, learning rate equals to $1e-3$, gradient accumulation steps equals to 1, and exponential moving average decay equals to 0.995. Furthermore, We set the value of timestamp which is the number of steps of adding noise equals to 100. Moreover, for the noise scheduler, we choose a linear beta schedule function to generate timestamps which is an

essential parameter used in training.

6.4 Sampling Details

Sampling originated images from the predicted probabilistic distribution of our diffusion model is a crucial process in justifying the performance of our diffusion model. Our experiments of the sampling process set the total number of images we sample from the diffusion model equals to 1000 with 5 iterations of sampling process.

6.5 Explanation Details

Predicted noise is crucial in our diffusion model. It plays an important role in determining the predicted probabilistic distribution of training data. To answer the research question RQ2 and the research question RQ3, we implement the code below to explain the predicted noise in a human understandable way.

```
x = images
x = x.cuda()
x.requires_grad_()
loss = trainer.model(x)
loss.backward()

# We take a maximum magnitude value across all channels to derive a single saliency value for each pixel
saliencies, _ = torch.max(x.grad.data.abs().detach().cpu(),dim=1)
```

```
9
10  saliencies = torch.stack([normalize(item) for item in
    ↪  saliencies])
```

6.6 Visual Turing Test

We have 36 participants for taking a Virtual Turing Test. These 36 test-takers are local Chinese who acknowledge Chinese style landscape paintings. Our test consists of a part of visual quality for generated images and a part of visual quality for explanation images. The first part of the test contains three questions and the second part of the test contains two questions.

6.6.1 Visual Quality for Generated Images

Each test-taker is given 10 images generated from our Chinese style landscape paintings generation diffusion model. They are asked the following questions:

- Was this image a Chinese style landscape painting? (Yes, No)
- If yes, was this image painted by a human or a computer? (human, computer)
- How certain were you about your answer? (Scale of 1-10)

6.6.2 Visual Quality for Explanation Images

Each test-taker is given 10 pairs of images of original Chinese style landscape paintings and corresponding saliency images with the following questions:

- Did you agree with the explanation images regarding the characteristics of the original image? (Yes, No)
- How much did you agree? (Scale of 1-10)

Chapter 7

Results

7.1 Summary

In this Chapter 7, we conduct an introduction of our results generated from the aforementioned experiments from Chapter 6, and conduct a sufficient interpretation and analysis regarding our results.

In Section 7.2, we show sampled images generated from our diffusion model and analyse the results of our sampling process. In Section 7.3, we visualize explanation images in pairs with corresponding original input images. We interpret the visualization results. In Section 7.4, we report the results of the Visual Turing Test in regard to the aforementioned results in Section 7.2 and Section 7.3. We analyse the performance of our diffusion model and the efficiency of our explanation according to the results of the Visual Turing Test.

7.2 Sampled Images from Our Diffusion Model

Figure 7.1 shows the results of sampling originated Chinese style landscape paintings from the predicted probabilistic distribution of Chinese style landscape paint-

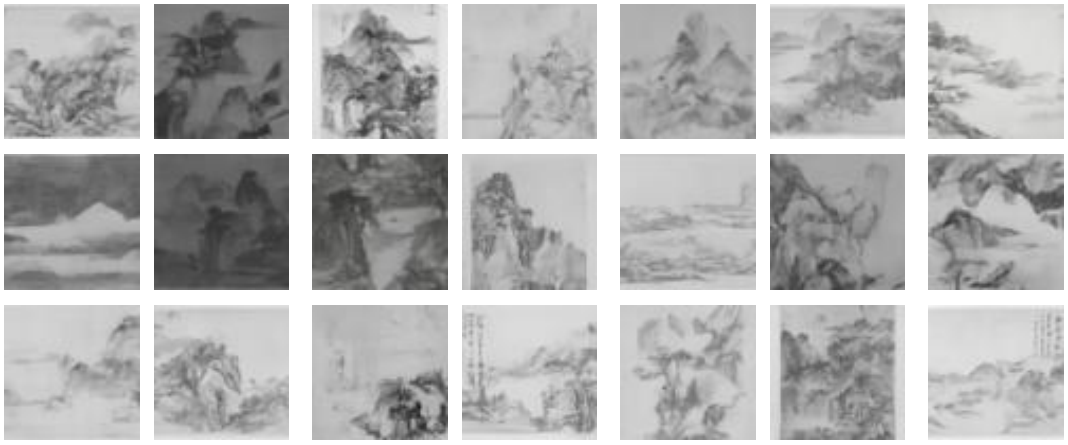


Figure 7.1: Sampled Chinese style landscape paintings from our diffusion model.

ings generated by our diffusion model.

Outputs of sampling are monochromatic with each size 64×64 pixels. The reasons that we obtain the results in such format are: (1) we pre-process our training dataset of Chinese style landscape paintings into grayscale images and resize the training data to maximise 64×64 pixels due to high complexity and large computations of training a diffusion model, (2) we apply U-Net architecture in diffusion model, which makes output the same size as input.

The sampled originated Chinese style landscape images as shown in Figure 7.1 well mimic traditional Chinese style landscape paintings regarding realism and artistic composition. The sampled outputs show a well-defined semblance of landscape such as high-level shapes of mountains, lakes, and low-level detailed depiction of rocks and trees. Such output samples show our diffusion model learns to paint distance mountains and mountain terrains surrounded with mist with light colours to establish a fade perspective. Our results pleasingly represent characteristics of traditional Chinese style landscape paintings both in realism and artistic composition, and well answered our first research question that our diffusion model can generate Chinese style landscape paintings.

7.3 Saliency Images Generated for Explanation

We proceed further with our diffusion model to identify predicted noise, the core element of our diffusion model. Visualization of predicted noise and original training images are shown in pairs in Figure 7.2. Original training images are placed on the top of each row and corresponding explanations, saliency maps, are placed on the bottom of each row.

As Figure 7.2 shows, each saliency map sketches out the high-level contour such as mountains, lakes and trees, as well as low-level details such as the texture of mountains, especially the saliency maps in the third column of each row show the evidence of details depiction in regards to angular structures of mountains, trees and rocks.

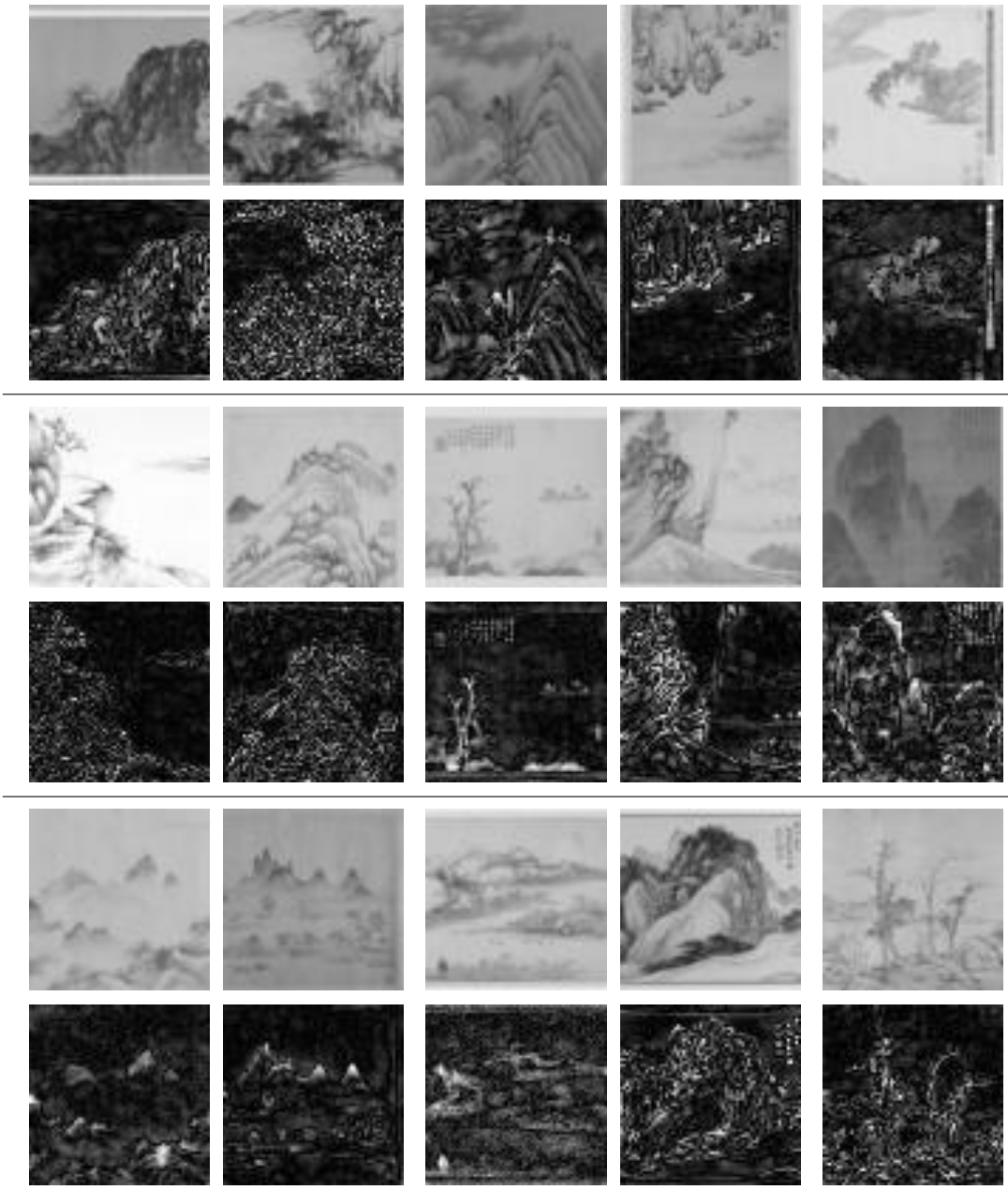


Figure 7.2: Saliency images with respect to original input for an explanation of our diffusion model. Saliency maps are on the bottom of each row with a black background. Original training images are on the top of each row.

7.4 Results of Visual Turing Test

7.4.1 Visual Quality for Generated Images

There are 33 test-takers identifying our sampled originated images are Chinese style landscape paintings among the total of 36 participants.

	Average	Stddev
Human Arts	0.67	0.48

Table 7.1: Frequency results of recognizing computer paintings as human paintings by the Virtual Turing Test.

Table 7.1 shows the frequency mistaken of human arts among 33 participants who consider our sampled images as Chinese style landscape paintings. Our Diffusion model paintings are recognized as human arts instead of computer paintings with a 67% frequency.

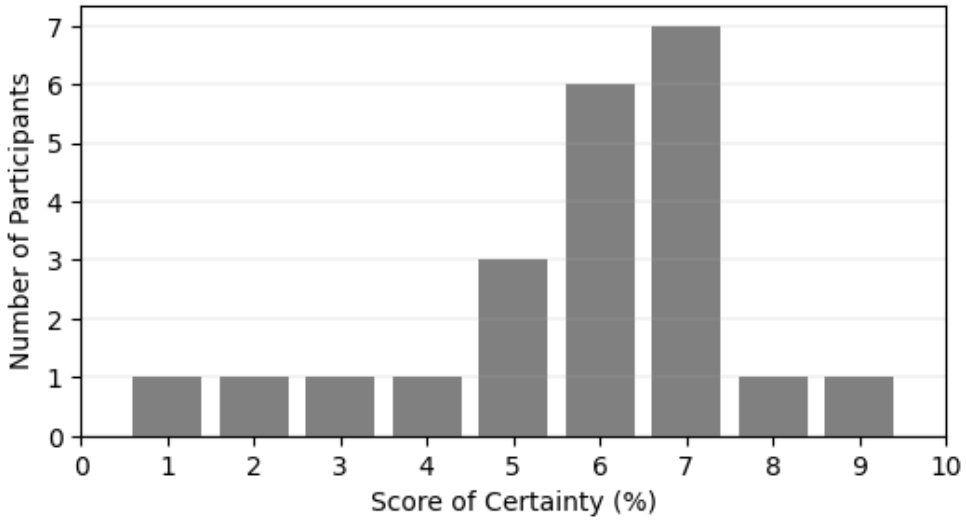


Figure 7.3: Score distribution of certainty that each test-taker considers the sampled image as human arts by the Visual Turing Test. (Average = 59%)

Figure 7.3 shows the score distribution that how certain 33 participants consider about the sampled paintings are painted by a human instead of a computer.

The average score of certainty is 59%.

The results of Table 7.1 and Figure 7.3 clearly show the answer to the research question RQ2 that the diffusion model can generate Chinese style landscape paintings.

7.4.2 Visual Quality for Explanation Images

	Average	Stddev
Agreements	0.58	0.49

Table 7.2: Agreements frequency by the Virtual Turing Test. The participants are asked if they agree with the explanation images regarding the characteristics of the original image.

Table 7.2 shows the frequency of agreements that the explanation images depict the main characteristic of corresponding original images among 36 participants. Our explanation method achieves alignments between human understanding and computer understanding in terms of Chinese style landscape paintings generated from our diffusion model.

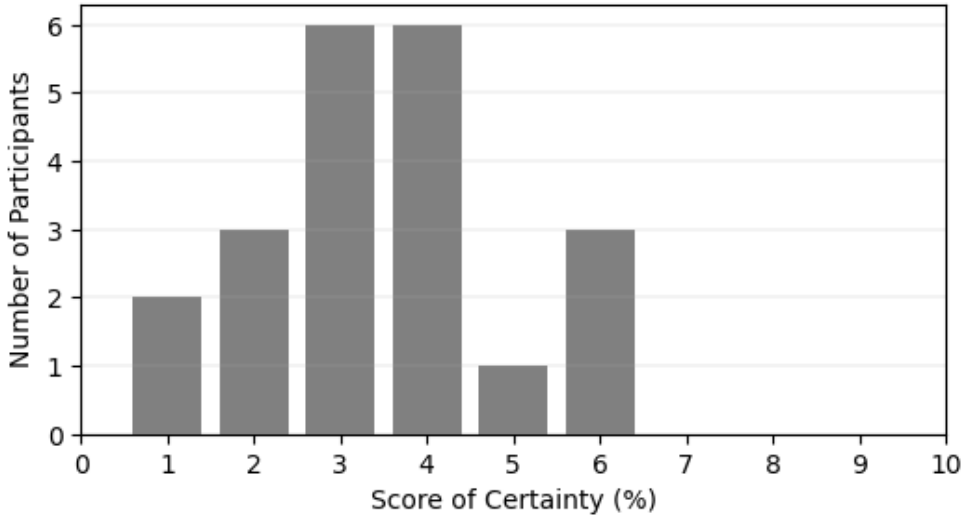


Figure 7.4: Score distribution of certainty that each test-taker agrees with each explanation image by the Visual Turing Test. (Average = 35%)

Figure 7.4 shows the score distribution of certainty by the Visual Turing Test. Participants are asked how certain they agree with the explanation images with respect to the original images. Such distribution demonstrates the participants agree with the explanation images reflect the main characteristics of the original image with average certainty of 35%.

We interview the participants in terms of low certainty of agreement with explanation images. There are two main reasons: (1) all participants agree with the structural composition of explanation images which aligns with the fundamental characteristics of traditional Chinese style landscape paintings, (2) whereas it is difficult to look into details to justify their certainty due to low-resolution of both original image and explanation image.

The results of Table 7.2 and Figure 7.4 answer our research questions RQ2 and RQ3. Our explanation images visualize the noise with respect to the corresponding input images. Furthermore, our explanation images can help us to evaluate the performance of our diffusion model.

Chapter 8

Conclusion

We leverage the diffusion model in a Chinese style landscape painting generation task. We also propose an explanation method as a probe to visualise the core elements of the diffusion model and adopt such explanation method to evaluate the performance of the diffusion model. We justify our explanation images align the considerations of humans and the diffusion process of our diffusion model regarding the main characteristics of traditional Chinese style landscape paintings.

Visual quality assessments find that: (1) our diffusion model can generate originated Chinese style landscape paintings which answer the first research question RQ1, (2) the core component, predicted noise, of our diffusion model can be visualised by saliency maps which answer our second research question RQ2, and (3) the alignment in terms of considerations of main characteristics of Chinese style landscape paintings between human and our diffusion model, such alignment can be further adopted to evaluate the performance of our diffusion model which answer our third research question RQ3.

Visual quality assessments also show us the low certainty of agreeing with explanation images. As we mentioned in Section 7.4 from Chapter 7, One main reason for such problem is the low resolution of the image limited by computation

power during the training process.

Future work may substitute different diffusion models to generate high-resolution images and generate different styles of images by an end-to-end architecture without text encoding. Importantly, apart from adopting visual quality assessments, we would like to study quantitative evaluation methods in regard to explaining the generative model.

References

[1] M. Bussagli, *Chinese painting*. Paul Hamlyn, 1969. 1, 5

[2] D. Preziosi, *The art of art history: a critical anthology*. OUP Oxford, 2009. 6

[3] E. Tosaki, *Mondrian's Philosophy of Visual Rhythm*. Springer, 2017. 6

[4] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, and C. Xu, "Inversion-based style transfer with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 146–10 156. 7, 14

[5] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 7, 11

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015. 7, 11

[7] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta vae," *arXiv preprint arXiv:1804.03599*, 2018. 7

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley,

S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. 7, 8

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020. 8

[10] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "Pixelvae: A latent variable model for natural images," *arXiv preprint arXiv:1611.05013*, 2016. 11

[11] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015. 11

[12] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425. 11

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 11

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. 11

[15] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *Advances in neural information processing systems*, vol. 29, 2016. 11

[16] M. Gadelha, S. Maji, and R. Wang, "3d shape induction from 2d views of multiple objects," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 402–411. 11

[17] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," *Advances in neural information processing systems*, vol. 29, 2016. 11

[18] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021. 11

[19] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022. 11

[20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763. 11

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695. 11, 13

[22] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10 696–10 706. 11

[23] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 11

[24] W. Zhang, C. Cao, S. Chen, J. Liu, and X. Tang, "Style transfer via image component analysis," *IEEE Transactions on multimedia*, vol. 15, no. 7, pp. 1594–1601, 2013. 12

[25] B. Wang, W. Wang, H. Yang, and J. Sun, "Efficient example-based painting and synthesis of 2d directional texture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 3, pp. 266–277, 2004. 12

[26] Y. Chang, S. Saito, K. Uchikawa, and M. Nakajima, "Example-based color stylization of images," *ACM Trans. Appl. Percept.*, vol. 2, no. 3, pp. 322–345, 2005. 12

[27] J. Fišer, O. Jamriška, D. Simons, E. Shechtman, J. Lu, P. Asente, M. Lukáč, and D. Sỳkora, "Example-based synthesis of stylized facial animations," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017. 12

[28] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2085–2094. 12

[29] T.-J. Fu, X. E. Wang, and W. Y. Wang, "Language-driven artistic style transfer," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI.* Springer, 2022, pp. 717–734. 12

[30] Y. Bai, J. Liu, C. Dong, and C. Yuan, "Itstyler: Image-optimized text-based style transfer," *arXiv preprint arXiv:2301.10916*, 2023. 13

[31] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410. 13

[32] E. Collins, R. Bala, B. Price, and S. Susstrunk, "Editing in style: Uncovering the local semantics of gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5771–5780. 13

[33] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9243–9252. 13

[34] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9841–9850, 2020. 13

[35] N. Huang, F. Tang, W. Dong, and C. Xu, "Draw your art dream: Diverse digital art synthesis with multimodal guided diffusion," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1085–1094. 14

[36] J. R. Quinlan, "Learning decision tree classifiers," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 71–72, 1996. 14

[37] L. Chiticariu, Y. Li, and F. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 827–832. 14

[38] J. G. Ibrahim, M.-H. Chen, S. R. Lipsitz, and A. H. Herring, "Missing-data methods for generalized linear models: A comparative review," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 332–346, 2005. 14

[39] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital signal processing*, vol. 73, pp. 1–15, 2018. 14

[40] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural network see the world-a survey of convolutional neural network visualization methods," *arXiv preprint arXiv:1804.11191*, 2018. 14

[41] G. Vilone and L. Longo, "Explainable artificial intelligence: a systematic review," *arXiv preprint arXiv:2006.00093*, 2020. 14

[42] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013. 15, 29

[43] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014. 15

[44] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017. 15

[45] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328. 15

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833. 15

[47] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017. 15

[48] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018. 15

[49] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144. 15

[50] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017. 15

[51] J. Cahill, *Hills beyond a river: Chinese painting of the Yüan Dynasty, 1279-1368*. Weatherhill, 1976. 16, 17

[52] A. Xue, "End-to-end chinese landscape painting creation using generative adversarial networks," in *Proceedings of the IEEE/CVF Winter conference on applications of computer vision*, 2021, pp. 3863–3871. 17

[53] P. Burger, "The berne convention: Its history and its key role in the future," *JL & Tech.*, vol. 3, p. 1, 1988. 19

[54] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017. 19

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241. 23

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 23

[57] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, "Efficient attention: Attention with linear complexities," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3531–3539. 23

[58] A. Vahdat. (2022) Improving diffusion models as an alternative to gans, part 2. [Online]. Available: https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/ 23

[59] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016. 23

[60] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 491–507. 24

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 24

[62] P. Contributor. (2023) Pytorch document. [Online]. Available: https://pytorch.org/docs/stable/data.html.torch.utils.data.DataLoader 26, 27