2023 Spring Master of Science Computer Science 3 4in 4in
Eastern Washington University

DIFFUSION MODELS FOR CROSS-DOMAIN IMAGE-TO-IMAGE TRANSLATION WITH PARTIALLY PAIRED DATASETS

A Thesis
Presented To
Eastern Washington University
Cheney, Washington

In Partial Fulfillment of the Requirements
for the Degree
Master of Science in Computer Science

by
Evan Bell

THESIS OF Evan Bell APPROVED BY

...., GRADUATE STUDY COMMITTE	Date
...., GRADUATE STUDY COMMITTE	Date
...., GRADUATE STUDY COMMITTE	Date

DIFFUSION MODELS FOR CROSS-DOMAIN IMAGE-TO-IMAGE TRANSLATION WITH PARTIALLY PAIRED DATASETS

by

Evan C. Bell

Spring 2023

todo: write abstract here

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Todo

Chapter 2

Image Generation and Translation with Neural Networks

2.1 Introduction

Generative machine learning has made waves outside of the machine learning field. While impressive neural-network-powered models for image-to-image translation, text-to-speech generation, speech-to-text generation, image generation, and much more, have been refined over the past decade, recently large-scale generative models have jumped into the public consciousness. Their results have been both highly visible, reported on in the media, and surprising to many. At the moment of writing, models like GPT-4 [7] for text generation as well as latent diffusion models [8] for image generation, to name only a couple, have being enjoying a moment of intense attention in the current zeitgeist.

2.2 Generative Machine Learning

Let's walk through the concept of generative models. A dataset contains a set of samples from a larger distribution. We model of the distribution (usually a complex, high-dimensional one) the dataset's samples are drawn from. If we have a good model, we may draw samples from the model that are highly likely assuming the dataset's distribution. Let's say we want to generate photos of dogs. We can take a dataset containing images of dogs, which are samples from the unknown, intractable distribution of all dog images. We then may train a model to generate new samples, outside of the data points, that are likely assuming the dog-image distribution. In practice, we don't know the true likelihood or distribution, so we judge the quality of the model based on other metrics, to be discussed later.

Here is a geometric interpretation of generative modeling using some concepts from mathematics. If our dataset is color (RGB) photos of dogs at a resolution of 256x256, the domain of each sample in the dataset is any 256x256 image that follows the RGB color model. An individual sample is a point in the 196,608-dimensional feature space, which contains every possible 256x256 image. If we take all possible dog photos, we have a set of points that define the surface of the dog manifold inside that space. In practice, that high-dimensional space is not mapped out exhaustively and we do not have every possible image of a dog on hand. What we can do is take a collection of samples whose features are on the surface of the dog manifold - a dataset. We don't know the whole surface of the dog manifold, but points nearby and in between samples are more likely to be from the dog manifold than points far away. We can train a model to interpolate between the data points, estimating the shape of the underlying surface. Since the arrangement of these points is highly non-linear, we might use a neural network as our model.

Our model also need to not be so specific. We could be trying to model the set of all natural photos, or even to model the distribution of all semantically meaningful (not noise) images. In language models, we could be trying to model arbitrary coherent sentences or arrangements of characters.

2.2.1 Families of Generative Models

There are multiple types of generative models in the machine learning landscape. Generative adversarial networks (GANs) [1] have been a dominant family of generative models since the mid-2010's. GAN learning is a game between two neural networks: a generator, and a discriminator. While the model's goal as a whole is to generate synthetic data, the discriminator's goal is to differentiate between synthetic and real data. The generator, which makes the synthetic samples, gets a gradient signal from mistakes that the discriminator makes during training (and optionally other sources of training loss). Ideally, during training, gradient descent will lead the system into a local Nash equilibrium such that the generator makes convincing synthetic samples and the discriminator makes accurate predictions about which samples are real and which are fake.

Note: Also add blurbs acknowledging variational autoencoders, energy based models, autoregressive models, and normalizing flows.

2.3 Diffusion Models

2.3.1 Introducing Diffusion Models

Diffusion models [12] are a family of generative machine learning models that have emerged as a leader in image generation. [2] While the framework for dif-

fusion models was laid out in 2015, in 2020 successful design of the architecture, training and sampling methods for image-generation diffusion models using neural networks was proposed. [3] The proposed neural network is a convolutional architecture with multi-scale convolution. In the following year, the architecture of this neural network was refined [6] and since then the research direction has been iterated on at a rapid pace.

Diffusion models have the advantage of being capable of high-quality sample generation. Samples from prominent diffusion models such as Imagen [9] have few artifacts and are often visually striking for their quality. The most significant drawback of diffusion models is that they require sampling in multiple steps from the model, which makes image generation slower and more computationally expensive than GAN models with similar sample quality, for example, StyleGAN-T. [11] This issue may be addressed in future research. For now, if one is willing to accept higher computational cost, diffusion models are a go-to option if one wants the highest quality of generated samples.

2.3.2 How do diffusion models work?

Diffusion models generate samples by transforming the distribution of a sample from a known distribution (typically normal) into the distribution of the dataset in multiple steps. For example, a diffusion model trained on dog images can take a sample of pure Gaussian noise and, by magic, change the sample into a realistic image of a dog. Doing the denoising in multiple steps is key for high sample quality.

Note: Insert figure here of denoising an image.

To accomplish this, a diffusion model defines two processes: a forward and a reverse process. The forward process is the process of corrupting a data distribution into a noise distribution over time. The reverse process is the process of denoising a noise distribution back into the data distribution over time.

Here I will follow the continuous-time specification of the diffusion process [15] [13] [14] [4] [5] (time can also be specified discretely as in [3]). We denote time $t \in [0, 1]$. A diffusion model has latent variables $\mathbf{z} = \{z_t\}$. We assume a population density $\mathsf{pop}(\mathbf{x})$ for $\mathbf{x} \in R^d$ for, say, the population of images. For a given time resolution Δt we consider the *forward* process

$$z(0),\ \ z(\Delta t),\ \ z(2\Delta t),\ \ z(3\Delta t),\ \ \ldots\ \ z(T\Delta t)$$

where T is the number of total steps and $\Delta t = \frac{1}{T}$. The distribution of the latent variable $z(t)$ is a function of t where

$$z(0) \sim \mathsf{pop}(\mathbf{x})$$

$$z(1) \sim \mathcal{N}(0, I)$$

In the forward process, t is increasing; the distribution of $z(0)$ is the data distribution, and as t increases the distribution of $z(t)$ is corrupted into a noise distribution. Let $\mathbf{x}_0 \sim \texttt{pop}(\mathbf{x})$ denote training data and $\epsilon \sim \mathcal{N}(0, I)$ denote noise from a Gaussian normal distribution. Then when $t = 0, z(0) = x$ is clean training data. In the final step when $t = 1$, $z(1) = \epsilon$ is pure noise. In the intermediate steps, $z(t)$ is defined by

$$z(t) = \sqrt{\gamma(t)}\mathbf{x}_0 + \sqrt{1 - \gamma(t)}\epsilon$$

where $\gamma(t)$ is a *noise schedule* that determines the distribution of the intermediate latent variables. When $\gamma(t)$ is a monotonically decreasing function that satisfies the conditions

$$\gamma(0) = 1, \ \ \gamma(1) = 0, \ \ \int_0^1 \sqrt{1 - \gamma(t)} = 1$$

where the first condition means $z(0) = \mathbf{x}_0$. The second condition ensures that $z(1)$ is some scalar multiple of ϵ. Because a sum of Gaussians is a Gaussian with variance equal to the sum of the variances, the third condition ensures that $z(1) = \epsilon$. Then the distribution of the forward process $q(z(t) \mid \mathbf{x}_0)$ satisfies this Markovian structure:

$$q(z(t) \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{\gamma(t)}\mathbf{x}_0, 1 - \gamma(t))$$

Now we have equations instructing us much to corrupt each data point with noise at a given timestep t. In order to make diffusion models *work*, we need to be able to reverse the corruption and predict an $\mathbf{x}_0$ from ϵ. In my implementation, I choose to solve this with an angular parameterization of the reverse process, originating in [10]. Let $\gamma(t) = \cos^2 \frac{\pi}{2}t$. Then $\sqrt{1 - \gamma(t)} = \sin(\frac{\pi}{2}t)$. In this way, we can reparameterize our diffusion process with an angle $\theta_t = \frac{\pi}{2}t$, and each latent $\mathbf{z}$ is defined by

$$z(t) = \cos(\theta_t)\mathbf{x}_0 + \sin(\theta_t)\epsilon$$

With this, a geometric interpretation of the diffusion process follows. The forward process evolves the vector $z(t)$ by moving it in a circle in a space where $\mathbf{x}_0$ and ϵ form the basis. The direction of the forward process is its derivative,

$$\frac{d}{d\theta_t}z(t) = -\sin(\theta_t)\mathbf{x}_0 + \cos(\theta_t)\epsilon$$

Chapter 3

Approach

3.1 Introduction

3.2 Datasets

3.3 Network Architecture

3.4 Proposed Diffusion Models for Image Translation

3.4.1 Simple Concatenated-sample v-Diffusion Model

3.4.2 Spherical Coordinate Diffusion Model

Chapter 4

Experiments

4.1 Methods

4.1.1 Concatenated-sample, Paired-image Training with v-diffusion Model

4.1.2 Paired-image Training with Cross-Domain Spherical Coordinate Diffusion Model

4.1.3 Partially-paired-image Training with Cross-Domain Spherical Coordinate v-diffusion Model

4.2 Results

4.2.1 MSE Performance Metric

4.2.2 FID Performance Metric

4.2.3 Comparison of Methods

Chapter 5

Conclusion

Bibliography

[1] Aaron Courville and Yoshua Bengio. Generative adversarial nets. *Advanc in Neural*, 2014.

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

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

[4] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[5] David McAllester. On the mathematics of diffusion models. *arXiv preprint arXiv:2301.11108*, 2023.

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

[7] OpenAI. Gpt-4 technical report, 2023.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[9] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[10] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[11] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023.

[12] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[13] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[14] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[15] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

Appendix A

Implementation Details

A.1 Training Settings