이학 석사 학위논문

Applying Regularized Schrödinger-Bridge-Based Stochastic Process in Generative Modeling

생성 모델링에서 정칙화된 슈뢰딩거 브리지 기반의 확률 과정의 적용

2022년 8월

서울대학교 대학원

수리과학부

송 기 웅

Applying Regularized Schrödinger-Bridge-Based Stochastic Process in Generative Modeling

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University

by

Ki-Ung Song

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

August 2022

Applying Regularized Schrödinger-Bridge-Based Stochastic Process in Generative Modeling

생성 모델링에서 정칙화된 슈뢰딩거 브리지 기반의 확률 과정의 적용

지도교수 강 명 주

이 논문을 이학 석사 학위논문으로 제출함

2022년 4월

서울대학교 대학원

수 리 과 학 부

송 기 웅

송 기 웅의 이학 석사 학위논문을 인준함

2022년 7월

위 원 장 국 웅 (인)

부 위 원 장 강 명 주 (인)

위 원 Ernest K. Ryu (인)

Abstract

Applying Regularized Schrödinger-Bridge-Based Stochastic Process in Generative Modeling

Ki-Ung Song

Department of Mathematical Sciences

The Graduate School

Seoul National University

Compared to the existing function-based models in deep generative modeling, the recently proposed diffusion models have achieved outstanding performance with a stochastic-process-based approach. But a long sampling time is required for this approach due to many timesteps for discretization. Schrödinger bridge (SB)-based models attempt to tackle this problem by training bidirectional stochastic processes between distributions. However, they still have a slow sampling speed compared to generative models such as generative adversarial networks. And due to the training of the bidirectional stochastic processes, they require a relatively long training time. Therefore, this study tried to reduce the number of timesteps and training time required and proposed regularization terms to the existing SB models to make the bidirectional stochastic processes consistent and stable with a reduced number of timesteps. Each regularization term was integrated into a single term to enable more efficient training in computation time and memory usage. Applying

this regularized stochastic process to various generation tasks, the desired translations between different distributions were obtained, and accordingly, the possibility of generative modeling based on a stochastic process with faster sampling speed could be confirmed.

Key words: Deep Learning, Generative Model, Stochastic Process, Diffusion Model, Schrödinger Bridge

Student Number: 2020-22722

Contents

Chapter 1

Introduction

As deep neural networks become essential elements in modern artificial intelligence research, various deep generative models and related neural network architectures have been proposed. One of the most widely known deep generative models is generative adversarial networks (GANs) [8]. They are based on adversarial training of generator and discriminator networks and have shown outstanding performance in various fields. Based on a log-likelihood of desired data distribution $\mathcal{P}$, variational autoencoders (VAEs) [13] and normalizing flows [12] were proposed. VAEs are trained with a lower bound of the log-likelihood designed with encoder and decoder networks. And normalizing flows are trained with an invertible design of neural network architectures for the exact computation of the log-likelihood. Although there are differences in specific ways, they all approach generative modeling as a function.

Recently, diffusion models [9, 23] have been proposed and shown outstanding performance with a stochastic-process-based approach. Since the latent space of generative modeling, $\mathcal{Z}$, is assumed to be a Gaussian noise space, diffusion models

first consider the stochastic noising process, say forward process, from $\mathcal{P}$ to $\mathcal{Z}$. Then, they consider a generation process as a corresponding backward stochastic process of the forward process. Due to their impressive performance and formulation, they are applied in various fields, including largely pre-trained multimodal models [19, 20].

Under the neural network's universal approximation property, deep generative models have achieved remarkable improvement in various generation tasks such as unconditional generation, image-to-image translation, image super-resolution, *etc.* Even though the desired generation outcome varies for each situation, every situation is to find a proper mapping between two different desired distributions $\mathcal{P}$ and $\mathcal{Q}$ with task-dependent conditions. For instance, in the case of an unconditional data generation task, $\mathcal{P}$ is the desired data distribution, and $\mathcal{Q}$ is the distribution of the latent space $\mathcal{Z}$, *e.g.* Gaussian distribution. And, in the case of an image-to-image translation task, two different image domains $\mathcal{P}$ and $\mathcal{Q}$ are given, for instance, male and female. Then the main objective is to find the proper mapping between $\mathcal{P}$ and $\mathcal{Q}$ while preserving the semantic information of the given image, *e.g.* identity of the human face. There are various studies on generative model frameworks and related neural network architecture for each generation task. In other words, the existing methods on deep generative models rely heavily on what the two specific distributions $\mathcal{P}$ and $\mathcal{Q}$ are.

And among many generative models, the two main approaches are competing for the best performance: GANs and diffusion models. For years, GANs have shown an ability to generate high-quality images, and diffusion models demonstrated that they can be better than GANs [7] in an image generation task. However, each model has its shortcomings. In the case of GANs, various models suffer from unstable

training and failure modes. Among the failure modes, there is a mode collapse problem where the trained models do not fully cover the desired data space. On the other hand, diffusion models show relatively stable training and high mode coverage performance. But the main disadvantage is its slow inference speed since it needs multiple timesteps to discretize the stochastic process. But, the success of diffusion models provides a new idea in generative modeling. Since they consider the generation process as a series of stochastic processes rather than a single function, it demonstrated that the application of stochastic processes in generative modeling could achieve both high mode coverage performance and high-quality generation.

As mentioned above, various generation tasks depend on what the desired distributions $\mathcal{P}$ and $\mathcal{Q}$ are. And in many cases, there is no need for $\mathcal{Q} = \mathcal{Z}$. Even in generation situations such as text-conditional image generation, text-to-speech translation, or image captioning, the transformation between two distributions with different modalities should be considered. Although diffusion models proposed a multi-stage stochastic-process-based generative modeling rather than a single-stage function, they depend on the forward process from $\mathcal{P}$ to $\mathcal{Z}$. Thus, they cannot construct bidirectional stochastic processes between arbitrary distributions $\mathcal{P}$ and $\mathcal{Q}$. To tackle this problem, various conditioning methods [4, 14] have been proposed. But, these diffusion-based approaches still suffer from the slow sampling speed. However, in a generative framework where stochastic processes between arbitrary distributions are constructed, it can be quite possible to improve the diffusion model's disadvantages while maintaining the advantage.

From the perspective of applied mathematics, the problem of transportation between two distributions $\mathcal{P}$ and $\mathcal{Q}$ with minimal cost can be expressed as an optimal transport (OT) problem. And based on a Schrödinger bridge (SB) problem,

which is an extension of entropy regularized OT problem, the desired bidirectional stochastic processes can be obtained. Therefore, based on the SB problem's formulation, some generative modelings [3, 6, 27] were proposed. And the recent work [3] has proposed an SB-based stochastic process as an extension to the diffusion model's stochastic process. SB models require a relatively small number of timesteps compared to the diffusion models because the bidirectional processes are learnable. However, they still need a large number of evaluation steps than function-based generative modelings such as GANs.

Therefore, by modifying the existing SB-based formulation, this study tried to construct bidirectional stochastic processes with a reduced number of timesteps compared to the previous SB-based works. Before the main discussion, the following section briefly introduces the basic concepts of OT and SB for a better understanding of this study. The presented definitions and flow of explanation mainly referred to the work of Peyré [18] and Vargas [25].

1.1 Preliminaries

Given two data spaces X and Y, let $\mathcal{M}(X)$ and $\mathcal{M}(Y)$ be the set of probability measures, respectively. The optimal transport (OT) problem aims to formulate minimal-cost transportation from one data space to another. Let $T : X \to Y$ be a continuous map, then a corresponding push-forward operator $T_{\#} : \mathcal{M}(X) \to \mathcal{M}(Y)$ exists. For discrete measure $\alpha = \sum_{i=1}^{n} a_i \delta_{x_i}$ where δ_x is Dirac-Delta function, push-forward operator $T_{\#}$ can be expressed as

$$T_{\#}\alpha = \sum_{i=1}^{n} a_i \delta_{T(x_i)}. \tag{1.1}$$

More generally, the push-forward measure $\beta = T_{\#}\alpha$ should satisfy

$$\forall f \in \mathcal{C}(Y), \quad \int_Y f(y)d\beta(y) = \int_X f(T(x))d\alpha(x), \tag{1.2}$$

where $\mathcal{C}(Y)$ is a set of continuous functions on space Y and probability measures α and β on data space X and Y respectively. Now, given a cost function c, Monge's OT problem can be formulated as

$$\begin{aligned} &\inf_T \left\{ \int_X c(x, T(x))d\alpha(x) \right\}, \\ &s.t. \quad T_{\#}\alpha = \beta, \end{aligned} \tag{1.3}$$

to find the optimal transition from X to Y.

Monge's OT problem has a deterministic nature. To relax that condition, Kantorovich proposed another form of OT problem. Given two discrete measures, $\alpha = \sum_{i=1}^n a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^m b_j \delta_{x_j}$, let probability vectors as $a = (a_i) \in \mathbb{R}^n$ and $b = (b_j) \in \mathbb{R}^m$. Then, with a cost matrix C, Kantorovich's OT problem is given as

$$\begin{aligned} &\min_{P \in U(a,b)} \langle C, P \rangle = \min_{P \in U(a,b)} \sum_{i,j} C_{i,j} P_{i,j}, \\ &s.t. \quad U(a,b) = \{P \in \mathbb{R}_+^{n \times m} : P\mathbb{1}_m = a \quad and \quad P^T \mathbb{1}_n = b\}, \end{aligned} \tag{1.4}$$

where P can be said as a policy matrix that moves measures α to β. An extension of the above 1.4 to include continuous measures can be expressed as

$$\begin{aligned} &\inf_{\pi \in U(\alpha,\beta)} \int_{X \times Y} c(x,y)d\pi(x,y), \\ &s.t. \quad U(\alpha,\beta) = \{\pi \in \mathcal{M}(X \times Y) : \text{proj}_{X\#}\pi = \alpha \; and \; \text{proj}_{Y\#}\pi = \beta\}, \end{aligned} \tag{1.5}$$

where $\text{proj}_{X\#}$ and $\text{proj}_{Y\#}$ is the push-forwards of the projections $\text{proj}_X(x,y) = x$ and $\text{proj}_Y(x,y) = y$ respectively.

From a perspective of optimization problem on OT, the corresponding dual problem of 1.4 can be considered. And the dual problem is given as

$$\begin{aligned} &\max_{(f,g)\in\mathbb{R}(C)} \langle f, a\rangle + \langle g, b\rangle, \\ s.t. \quad &\mathbb{R}(C) = \{(f,g) \in \mathbb{R}^n \times \mathbb{R}^m : \forall(i,j),\ f_i + g_j \leq C_{ij}\}. \end{aligned} \tag{1.6}$$

Similarly, the dual problem formulation of 1.5 with arbitrary probability measures α and β is given as

$$\begin{aligned} &\sup_{(f,g)\in\mathbb{R}(C)} \int_X f(x)d\alpha(x) + \int_Y g(y)d\beta(y), \\ s.t. \quad &\mathbb{R}(C) = \{(f,g) \in \mathcal{C}(X) \times \mathcal{C}(Y) : \forall(x,y),\ f(x) + g(y) \leq C(x,y)\}. \end{aligned} \tag{1.7}$$

This type of dual problem provides a different perspective on the given OT problem.

By adding entropy regularization term in 1.4 and 1.5, stochastic nature can implicitly conditioned to the OT problem. For policy matrix P, discrete entropy term is given as $H(P) = -\sum_{i,j} P_{i,j} \log P_{i,j}$, thus the entropy regularized version of 1.4 is given as

$$\min_{P\in U(a,b)} \langle C, P\rangle - \epsilon H(P). \tag{1.8}$$

Again, the entropy regularized version of 1.5 can be given as below

$$\inf_{\pi\in U(\alpha,\beta)} \int_{X\times Y} c(x,y)d\pi(x,y) + \epsilon D_{KL}(\pi|\alpha\times\beta), \tag{1.9}$$

where $D_{KL}(p|q) = \int_{X\times Y} \log\left(\frac{dp}{dq}\right)dp$ is Kullback–Leibler (KL) divergence for distributions p and q.

By refactoring the 1.9 with Gibbs distribution $\mathcal{K}$ which is given as

$$d\mathcal{K}(x, y) = \exp -\frac{c(x, y)}{\epsilon} d\alpha(x) d\beta(y), \tag{1.10}$$

the entropy regularized OT problem 1.9 can be expressed as

$$\inf_{\pi \in U(\alpha,\beta)} D_{KL}(\pi|\mathcal{K}). \tag{1.11}$$

The above form of the problem is often called a static Schrödinger problem. This is a situation where there the Gibbs distribution $\mathcal{K}$ contains information about the cost function, and path π between α and β is optimized to be close with the Gibbs distribution as a reference.

By extending this, Schrödinger bridge (SB) problem can be proposed as

$$\inf_{\pi \in \mathcal{D}(\mathcal{P},\mathcal{Q})} D_{KL}(\pi|\mathcal{W}), \tag{1.12}$$

where reference measure $\mathcal{W}$ replaces the Gibbs measure and $\mathcal{D}(\mathcal{P}, \mathcal{Q})$ is a set of path measures with marginals of desired distribution $\mathcal{P}$ and $\mathcal{Q}$. This formulates a more general situation of finding a path measure between $\mathcal{P}$ and $\mathcal{Q}$ where cost information is implicitly reflected in a choice of reference measure. From the perspective of KL divergence as a distance, it can be interpreted that the process of reducing the distance between the path measure and reference measure reflects the nature of OT since the reference measure contains cost information.

The choice of $\mathcal{W}$ as a prior knowledge enables different interpretations of the SB problem. For instance, if $\mathcal{W}$ is uniform distribution, then 1.12 becomes equivalent to 1.11 with entropy. And it was demonstrated that the SB problem is equivalent

to a stochastic control problem with a proper choice of $\mathcal{W}$ [25, 17]. Let $\mathcal{W}^\gamma$ be the Wiener measure with volatility γ, then path measure $\pi \in \mathcal{D}(\mathcal{P}, \mathcal{Q})$ can be expressed as a distribution which evolves according to the solution of stochastic differential equations: forward direction and backward direction of Ito process form as

$$\begin{aligned} dx_t &= f_t dt + \sqrt{\gamma} dw_t, \\ dx_t &= b_t dt + \sqrt{\gamma} dw_t. \end{aligned} \tag{1.13}$$

With the above forward and backward Ito process, the SB problem can be expressed as the following two alternate objectives

$$\begin{aligned} &\min_{\pi \in \mathcal{D}(\mathcal{P},\mathcal{Q})} D_{KL}(\pi|\mathcal{W}^\gamma) = \min_{f_t} \mathbf{E}_\pi \left[\int_0^1 \frac{1}{2\gamma} \|f_t\|^2 dt \right], \\ &s.t. \quad dx_t = f_t dt + \sqrt{\gamma} dw_t, \ x_0 \sim \mathcal{P}, \ x_1 \sim \mathcal{Q}, \end{aligned} \tag{1.14}$$

$$\begin{aligned} &\min_{\pi \in \mathcal{D}(\mathcal{P},\mathcal{Q})} D_{KL}(\pi|\mathcal{W}^\gamma) = \min_{b_t} \mathbf{E}_\pi \left[\int_0^1 \frac{1}{2\gamma} \|b_t\|^2 dt \right], \\ &s.t. \quad dx_t = b_t dt + \sqrt{\gamma} dw_t, \ x_1 \sim \mathcal{Q}, \ x_0 \sim \mathcal{P} \end{aligned} \tag{1.15}$$

The above objectives do not provide information about an update rule of drift f_t and b_t in a stochastic process. But it means that the SB problem can be formulated as an optimal control problem with bidirectional stochastic processes minimizing their energy.

Chapter 2

Related Works

In this chapter, more backgrounds related to this work are presented. First, some OT-related deep learning studies are briefly introduced with their formulation. Next, detailed backgrounds related to diffusion models are explained. And lastly, formulations of SB-based generative modeling are presented.

2.1 Optimal Transport in Deep Learning

The OT-based approaches for deep learning were already widely used in many places, even before SB. The most widely known result would be WGAN [1]. With the definition of Kantorovich OT problem 1.5, consider the below:

$$\inf_{\pi \in U(\mathcal{P},\mathcal{Q})} \mathbf{E}_{\pi} \left[\|x - y\|_2 \right] = \inf_{\pi \in U(\mathcal{P},\mathcal{Q})} \int_{X \times X} \|x - y\|_2 \, d\pi(x, y), \tag{2.1}$$

where $\mathcal{P}$ is a distribution of desired real data and $\mathcal{Q}$ is a distribution of generated fake data. Then by the Kantorovich duality 1.7, the equivalent dual problem can

be expressed as

$$\sup_{\|h\|_L \leq 1} \mathbf{E}_{x \sim \mathcal{P}} \left[h(x) \right] - \mathbf{E}_{y \sim \mathcal{Q}} \left[h(y) \right], \tag{2.2}$$

where $\|h\|_L \leq 1$ denotes that h is a 1-Lipschitz function. Now with a generator network g_θ and discriminator network d_ϕ, the above 2.2 can be expressed as

$$\sup_{\|d_\phi\|_L \leq 1} \mathbf{E}_{x \sim \mathcal{P}} \left[d_\phi(x) \right] - \mathbf{E}_{z \sim \mathcal{Z}} \left[d_\phi(g_\theta(z)) \right], \tag{2.3}$$

where $\mathcal{Z}$ is a latent space, *i.e.* Gaussian noise.

More recently, in self-supervised deep learning, SwAV [2] utilized an OT-based approach. In the process of learning the representation feature vector, latent codes with discrete values are proposed. Thus, with the OT problem 1.8, Sinkhorn's algorithm was used to transport one latent code to another. The Sinkhorn's algorithm can be derived by applying Lagrangian $L(P, f, g)$ to 1.8 with two dual variables $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$ as

$$L(P, f, g) = \langle C, P \rangle - \epsilon H(P) - \langle f, P \mathbb{1}_m - a \rangle - \langle g, P^T \mathbb{1}_n - b \rangle. \tag{2.4}$$

Then, with first-order derivative to each element,

$$\frac{\partial L(P, f, g)}{\partial P_{i,j}} = C_{i,j} + \epsilon \log P_{i,j} - f_i - g_j = 0. \tag{2.5}$$

Thus, the resulting solution of 2.5 is given as

$$P_{i,j} = \exp(f_i/\epsilon) \exp(-C_{i,j}/\epsilon) \exp(g_j/\epsilon). \tag{2.6}$$

And by refactoring 2.6 as $P = \mathrm{diag}(u) K \mathrm{diag}(v)$, the iterative update rule of

Sinkhorn's algorithm is given as

$$u^{k+1} = \frac{a}{Kv^k} \quad \text{and} \quad v^{k+1} = \frac{b}{K^T u^k}. \tag{2.7}$$

To solve more general OT problems such as 1.9 in iterative form, the term called iterative proportional fitting (IPF) algorithm is broadly used including Sinkhorn's algorithm.

Domain translation is a well-known application of deep generative models. For this task, it is important to maintain the semantic information of the image during the translation: *e.g.* the content of an image in image-to-image translation or the nuance of a sentence in language translation. For this purpose, the cycle-consistency loss was proposed by CycleGAN [29] and has been used widely in various domains of deep learning. Given the two desired data spaces X and Y with measures α and β respectively, the (unsupervised) domain translation tasks such as image-to-image translation can be formulated as [5]

$$\begin{aligned} &\inf_{T,S} \int_X c(x, T(x)) d\alpha(x) + \int_Y c(S(y), y) d\beta(y), \\ &s.t. \quad T_{\#}\alpha = \beta, \ S_{\#}\beta = \alpha, \ T \circ S = \mathrm{id}, \ S \circ T = \mathrm{id}, \end{aligned} \tag{2.8}$$

where T and S are the desired functions for translations. And some studies [5, 21] demonstrated that this formulation is known to be equivalent to the form of Monge's OT problem 1.3. In other words, when the domain translation tasks such as image-to-image translation are formulated from the perspective of OT, it can be interpreted as an implicit cycle-consistency conditioned problem by its nature.

2.2 Deep Generative Models

Except for diffusion models, the previously mentioned generative models aim to train a one-stage function from $\mathcal{Z}$ to $\mathcal{P}$. In GANs, a mapping $G : \mathcal{Z} \to \mathcal{P}$ is trained directly. And in VAEs, two mappings are trained: encoder $E : \mathcal{P} \to \mathcal{Z}$ and decoder $D : \mathcal{Z} \to \mathcal{P}$. In normalizing flows, invertible network $G : \mathcal{P} \to \mathcal{Z}$ is trained where inference is done by $G^{-1} : \mathcal{Z} \to \mathcal{P}$. However, diffusion models first considered a transition from $\mathcal{P}$ to $\mathcal{Z}$ as a stochastic forward process. The forward process is a noising process and it can be formulated in various ways. One can be formulated via the Markov process [9] as

$$x_{t+1} = \mathcal{N}(\sqrt{\beta_t}x_t, (1-\beta_t)\mathbf{I}). \tag{2.9}$$

Meanwhile, the forward noising process can be formulated as a continuous stochastic differential equation (SDE), unlike the above discrete Markov process. This was proposed in the score-based generative model (SGM) [23] which became the prototype of the diffusion model and its SDE form is expressed as

$$dx_t = f_t(x_t)dt + g_t dw_t. \tag{2.10}$$

With such a forward process is given, it is known that the corresponding backward stochastic process is expressed as

$$dx_t = \left[f_t(x_t) - g_t^2 \nabla_x \log p_t(x_t)\right] dt + g_t dw_t, \tag{2.11}$$

where p_t is the marginal distribution of x_t, and $\nabla_x \log p_t(x_t)$ is called score function of p_t. Since the numerical computation of the above SDEs requires discretization,

with an Euler-Maruyama scheme and discrete-time step size γ, the numerical sampling process of the above two processes 2.10 and 2.11 is given as

$$x_{t+1} = x_t + \gamma f_t(x_t) + \sqrt{\gamma} g_t z, \tag{2.12}$$

$$x_t = x_{t+1} - \gamma \left[f_{t+1}(x_{t+1}) - g_{t+1}^2 \nabla_x \log p_{t+1}(x_{t+1}) \right] + \sqrt{\gamma} g_{t+1} z, \tag{2.13}$$

where z is Gaussian noise. The drift terms f_t and g_t are derived from the discrete-time diffusion frameworks such as DDPM [9] and NCSN [22]. The previously mentioned discrete noising process 2.9 of DDPM [9] can be induced in a SDE form 2.10. It is known as VPSDE and given as

$$dx_t = -\frac{1}{2}\beta_t x_t dt + \sqrt{\beta_t} dw_t. \tag{2.14}$$

Since the forward process 2.10 of SGM has a fixed linear drift, it is possible to compute an exact transition kernel that transits x_0 to x_t. The corresponding transition kernel of VPSDE is

$$p(x_t|x_0) = \mathcal{N}\left(x_t; x_0 e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \left(1 - e^{-\int_0^t \beta(s)ds}\right)\mathbf{I}\right). \tag{2.15}$$

Another example of SGM type is VESDE. It is induced from the formulation of NCSN [22] and given as

$$dx_t = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw_t, \tag{2.16}$$

with its corresponding transition kernel is

$$p(x_t|x_0) = \mathcal{N}\left(x_0, (\sigma^2(t) - \sigma^2(0))\mathbf{I}\right). \tag{2.17}$$

Considering the numerical backward process 2.13 for generation, in addition to the information of drift f_t and g_t, the information about score function $\nabla_x \log p_t(x_t)$ is also essential. Thus, the training of diffusion models is designed to make neural network $s_\theta(x_t, t)$ approximate the non-linear drift $\nabla_x \log p_t(x_t)$ at each x_t. To achieve this, objective function for training is given as

$$\mathbf{E}_{x_0,t}\left[\|s_\theta(x_t,t) - \nabla_x \log p(x_t|x_0)\|_2^2\right]. \tag{2.18}$$

By minimizing the above objective, known as score-matching framework [26], the neural network $s_\theta(x_t, t)$ can approximate the desired score function $\nabla_x \log p_t(x_t)$ properly.

Although diffusion models have shown remarkable performance and scalability to various domains, they suffer from slow sampling speed. Thus, many studies [10, 28, 24, 11] have proposed methods to improve sampling speed. These methods 1) utilize GAN's approximation ability, 2) introduce a faster numerical algorithm for solving SDEs, or 3) consider the diffusion process in the latent space, which is equivalent to diffusion models with non-linear drift f_t and g_t in the original data space. However, even in these attempts, the diffusion models still require a large number of timesteps for inference compared to one-stage generative models such as GANs. Also, bidirectional stochastic processes between any desired distribution $\mathcal{P}$ and $\mathcal{Q}$ cannot be obtained in the framework of diffusion models. As discussed before, a modality-agnostic transformation between $\mathcal{P}$ and $\mathcal{Q}$ is required to solve various real-world generation problems. For instance, at present, text-conditional image generation and image captioning are approached as separate tasks. But if the transformation between image and text data spaces are induced together as

bidirectional processes, the two tasks can be learned as one framework. Therefore, there is a need for stochastic-process-based generative modeling between any $\mathcal{P}$ and $\mathcal{Q}$. More research is still needed for the general modality-agnostic transformation between data distributions with different dimensions. But, the SB formulation can construct desired stochastic processes between data distributions of the same dimension, and it can be seen as an attempt at a more general generative framework while showing a direction for solving the limitations of the diffusion model.

2.3 Schrödinger Bridge in Generative Modeling

From the SB problem 1.12 perspective, the work of SB-FBSDE [3] consists of two learnable generative processes: forward and backward stochastic processes. Similar to the SGM's forward and backward process, 2.10 and 2.11, SB-based formulation from the work of [3] is given as

$$dx_t = \left[f_t(x_t) + g_t^2 \nabla_x \log \psi_t(x_t)\right] dt + g_t dw_t, \tag{2.19}$$

$$dx_t = \left[f_t(x_t) - g_t^2 \nabla_x \log \hat{\psi}_t(x_t)\right] dt + g_t dw_t, \tag{2.20}$$

where $\nabla_x \log \psi_t(x_t)$ and $\nabla_x \log \hat{\psi}_t(x_t)$ are non-linear drift terms. If $\nabla_x \log \psi_t(x_t) = 0$ holds, it can be readily confirmed that it is equivalent to that of 2.10 and 2.11. With discretization step size γ and Euler-Maruyama scheme, the sampling process for numerical computation is given as

$$x_{t+1} = x_t + \gamma \left[f_t(x_t) + g_t^2 \nabla_x \log \psi_t(x_t)\right] + \sqrt{\gamma} g_t z, \tag{2.21}$$

$$x_t = x_{t+1} - \gamma \left[f_{t+1}(x_{t+1}) - g_{t+1}^2 \nabla_x \log \hat{\psi}_{t+1}(x_{t+1})\right] + \sqrt{\gamma} g_{t+1} z. \tag{2.22}$$

The similarity between the forward and backward processes of SB-FBSDE and those of SGM is straightforward. And since SB-FBSDE has non-linear drift in both bidirectional processes unlike SGM, SB-based stochastic processes are a generalization of that of SGM. While the drift term f_t and g_t are derived from discrete-time diffusion models in SGM's formulation, they are not induced theoretically in the SB-FBSDE formulation. Since there is no information to construct f_t and g_t in SB-FBSDE, they were manually set as those of VESDE or simply as $f_t = 0$ and $g_t = 1$. And unlike diffusion models, the above SB formulation cannot be trained in the form of score-matching, so two processes 2.19 and 2.20 are transformed into an equivalent SDE problem to construct a loss objective. In this process, non-linear drift terms of SB-FBSDE have the following relationship with score function $\nabla_x \log p_t^{\mathrm{SB}}(x_t)$ of path measure p_t^{SB} of SB problem:

$$\nabla_x \log \psi_t(x_t) + \nabla_x \log \hat{\psi}_t(x_t) = \nabla_x \log p_t^{\mathrm{SB}}(x_t). \tag{2.23}$$

There is another SB-based formulation, Diffusion Schrödinger Bridge (DSB) [6], which was induced as Markov processes. It is defined as

$$p^f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}; x_t + \gamma \mathbf{f}_t(x_t), 2\gamma \mathbf{I}) = \mathcal{N}(x_{t+1}; F_t(x_t), 2\gamma \mathbf{I}), \tag{2.24}$$

$$p^b(x_t|x_{t+1}) = \mathcal{N}(x_t; x_{t+1} + \gamma \mathbf{b}_{t+1}(x_{t+1}), 2\gamma \mathbf{I}) = \mathcal{N}(x_t; B_{t+1}(x_{t+1}), 2\gamma \mathbf{I}), \tag{2.25}$$

for $t \in \{1, \ldots, T-1\}$ and T is the number of total time step. By letting the approximation of $p^f(x_t|x_{t+1})$ as

$$p^f(x_t|x_{t+1}) \approx \mathcal{N}(x_t; x_{t+1} - \gamma \mathbf{f}_t(x_{t+1}) + 2\gamma \nabla_x \log p_{t+1}^f(x_{t+1}), 2\gamma \mathbf{I}), \tag{2.26}$$

if the two processes $p^f(x_t|x_{t+1})$ and $p^b(x_t|x_{t+1})$ corresponds, the drift term of p^b can be expressed with the drift term and score function of p_t^f. Thus, it leads to an iterative update rule for the backward drift as

$$\mathbf{b}_{t+1}^n(x_{t+1}) = -\mathbf{f}_t^n(x_{t+1}) + 2\gamma\nabla_x \log p_{t+1}^{f,n}(x_{t+1}). \tag{2.27}$$

And the update rule for the forward drift can be derived similarly as

$$\mathbf{f}_t^{n+1}(x_t) = -\mathbf{b}_{t+1}^n(x_t) + 2\gamma\nabla_x \log p_t^{b,n}(x_t). \tag{2.28}$$

With basic calculus and dominated convergence theorem, the followings can be derived:

$$\begin{aligned}
p^{f,n}(x_{t+1}|x_t) &= \mathcal{N}(x_{t+1}; F_t^n(x_t), 2\gamma\mathbf{I}), \\
p_{t+1}^{f,n}(x_{t+1}) &= \mathbf{E}_{p_t^{f,n}}[p^{f,n}(x_{t+1}|x_t)] \\
&= (4\pi\gamma)^{-d/2}\ \mathbf{E}_{p_t^{f,n}}\left[\exp[-\|F_t^n(x_t) - x_{t+1}\|^2]/4\gamma\right], \\
\nabla_x \log p_{t+1}^{f,n}(x_{t+1}) &= \mathbf{E}_{p_{t|t+1}^{f,n}}\left[F_t^n(x_t) - x_{t+1}\right]/2\gamma, \\
B_{t+1}^n(x_{t+1}) &= \mathbf{E}_{p_{t|t+1}^{f,n}}\left[x_{t+1} + F_t^n(x_t) - F_t^n(x_{t+1})\right].
\end{aligned} \tag{2.29}$$

Based on this result, as proof of other directions is similar, the following iterative objectives can be derived:

$$\mathbf{E}_{p_{t,t+1}^{f,n}}[\|B_{t+1}(x_{t+1}) - x_{t+1} - (F_t^n(x_t) - F_t^n(x_{t+1}))\|^2], \tag{2.30}$$

$$\mathbf{E}_{b_{t,t+1}^{b,n}}[\|F_t(x_t) - x_t - (B_{t+1}^n(x_{t+1}) - B_{t+1}^n(x_t))\|^2], \tag{2.31}$$

where B_{t+1}^n and F_t^{n+1} can be trained with the objectives 2.30 and 2.31 respectively.

In the DSB framework, the drift term for Brownian motion cannot be addressed

and there is no guarantee that each process p_t^f and p_t^b corresponds to a path measure p_t^{SB} of SB problem as in SB-FBSDE. However, the non-linear terms $\mathbf{f}_t$ and $\mathbf{b}_t$ are only core drift terms, reducing the need for manual selection of the linear drift terms, unlike FBSDE. In other words, since the non-linear drifts of DSB are trained through the neural networks without manual selection, the approximation property can be maximized. Thus, the DSB framework can be valid with fewer timesteps compared to SB-FBSDE. Both SB-FBSDE and DSM can be trained iteratively with IPF recursion. And since the number of required iterations for each IPF step is quite large, the training takes a long time for convergence compared to diffusion models.

Chapter 3

Proposed Method

Based on the SB-based formulations, this study tried to construct discrete-time stochastic processes between any two distributions with smaller timesteps required. While maximizing the use of non-linear drift as in the DSB framework, the formulation of SB-FBSDE was added as a regularization to make the two different forward and backward processes coincide as a path measure of the SB problem. The existing SB-based generative models' training is unstable with a smaller number of timesteps and iterations. Thus, to construct stable stochastic processes while reducing the number of iterations and timesteps required for convergence, the concept of cycle-consistency proposed by CycleGAN [29] was introduced as a regularization to SB-based formulation.

3.1 Regularization for Schrödinger Bridge

Consider the SB-FBSDE formulation with fixed Brownian motion drift $g_t = 1$ and let the both linear and non-linear drift of SB-FBSDE as a single non-linear drift

as in DSB,

$$dx_t = \left[f_t(x_t) + \nabla_x \log \psi_t(x_t)\right] dt + dw_t = \mathbf{f}_t(x_t)dt + dw_t, \tag{3.1}$$

$$dx_t = \left[f_t(x_t) - \nabla_x \log \hat{\psi}_t(x_t)\right] dt + dw_t = -\mathbf{b}_t(x_t)dt + dw_t. \tag{3.2}$$

When considering the following forward and backward processes where only the degree of variance differs from 2.24 and 2.25 of DSM,

$$\begin{aligned} p^f(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}; x_t + \gamma \mathbf{f}_t(x_t), \gamma\mathbf{I}) = \mathcal{N}(x_{t+1}; F_t(x_t), \gamma\mathbf{I}), \\ \rightarrow \quad x_{t+1} = x_t + \gamma \mathbf{f}_t(x_t) + \sqrt{\gamma} z \end{aligned} \tag{3.3}$$

$$\begin{aligned} p^b(x_t|x_{t+1}) = \mathcal{N}(x_t; x_{t+1} + \gamma \mathbf{b}_{t+1}(x_{t+1}), \gamma\mathbf{I}) = \mathcal{N}(x_t; B_{t+1}(x_{t+1}), \gamma\mathbf{I}), \\ \rightarrow \quad x_t = x_{t+1} - \gamma\left(-\mathbf{b}_{t+1}(x_{t+1})\right) + \sqrt{\gamma} z. \end{aligned} \tag{3.4}$$

Note that with different degree of variance, the 3.3 and 3.4 can have the same objective functions 2.30 and 2.31 respectively through the same process of DSM. Interpreting the above discrete stochastic processes as the Euler-Maruyama scheme with discretization step size γ, the corresponding continuous-time SDEs with forward and backward directions are given as

$$dx_t = \mathbf{f}_t(x_t)dt + dw_t, \tag{3.5}$$

$$dx_t = -\mathbf{b}_t(x_t)dt + dw_t, \tag{3.6}$$

Since these expressions 3.5 and 3.6 are equivalent to 3.1 and 3.2 respectively, it demonstrates that DSB and SB-FBSDE can be related as continuous-time SDEs.

Note that with the property of SB-FBSDE formulation 2.23, the following holds:

$$\mathbf{f}_t(x_t) + \mathbf{b}_t(x_t) = \nabla_x \log p_t^{\mathrm{SB}}(x_t). \tag{3.7}$$

And in ideal scenario, the path measure p_t^{SB} of SB problem should coincide with marginal measure of the forward and backward processes, 3.5 and 3.6. In this case, the following relationships hold:

$$\nabla_x \log p_t^f(x_t) = \nabla_x \log p_t^{\mathrm{SB}}(x_t) = \nabla_x \log p_t^b(x_t), \tag{3.8}$$

where p_t^f and p_t^b corresponds to the marginal distribution of forward and backward processes, 3.5 and 3.6 respectively. Now, through the same process as the work of DSM, the following holds:

$$\nabla_x \log p_{t+1}^{f,n}(x_{t+1}) = \mathbf{E}_{p_{t|t+1}^{f,n}} \left[F_t^n(x_t) - x_{t+1} \right] / \gamma. \tag{3.9}$$

Combining the results of 3.7, 3.9, and 3.9, the followings can be derived:

$$\begin{aligned}
\nabla_x \log p_{t+1}^{\mathrm{SB},n}(x_{t+1}) &= \nabla_x \log p_{t+1}^{f,n}(x_{t+1}), \\
\mathbf{f}_{t+1}^n(x_{t+1}) + \mathbf{b}_{t+1}^n(x_{t+1}) &= \mathbf{E}_{p_{t|t+1}^{f,n}} \left[F_t^n(x_t) - x_{t+1} \right] / \gamma, \\
\gamma \mathbf{b}_{t+1}^{n+1}(x_{t+1}) &= \mathbf{E}_{p_{t|t+1}^{f,n}} \left[F_t^n(x_t) - F_{t+1}^n(x_{t+1}) \right], \\
B_{t+1}^n(x_{t+1}) &= \mathbf{E}_{p_{t|t+1}^{f,n}} \left[x_{t+1} + F_t^n(x_t) - F_{t+1}^n(x_{t+1}) \right].
\end{aligned} \tag{3.10}$$

Based on this, as a case of other directions is similar, the following regularization

objectives are given as

$$\mathbf{E}_{p^{f,n}_{t,t+1}}[\|B_{t+1}(x_{t+1}) - x_{t+1} - (F^n_t(x_t) - F^n_{t+1}(x_{t+1}))\|^2], \tag{3.11}$$

$$\mathbf{E}_{b^{b,n}_{t,t+1}}[\|F_t(x_t) - x_t - (B^n_{t+1}(x_{t+1}) - B^n_t(x_t))\|^2], \tag{3.12}$$

where 3.11 holds for $t \in \{0, \ldots, T-2\}$ and 3.12 holds for $t \in \{1, \ldots, T-1\}$. And B^n_{t+1} and F^{n+1}_t can be iteratively trained with the objectives 3.11 and 3.12 respectively. This objective can be thought as an additional regularization term for the DSB's objective function 2.30 and 2.31 since it cannot applied for all time step $t \in \{0, \ldots, T-1\}$.

Putting objective term of 2.30 as $\mathcal{L}_{DSB}$ and 3.11 as $\mathcal{L}_{reg}$, the objective of regularized SB-based formulation can be expressed as

$$\alpha\mathcal{L}_{DSB} + (1-\alpha)\mathcal{L}_{reg}, \tag{3.13}$$

where α is a hyperparameter. Note that the $\mathcal{L}_{DSB}$ and $\mathcal{L}_{reg}$ are very similar, and learning these two objectives as separate terms requires more GPU memory by storing two similar computational graphs. With the convexity of $\|\cdot\|^2$, the memory-efficient objective can be attained as

$$\begin{gathered}\alpha\mathcal{L}_{DSB} + (1-\alpha)\mathcal{L}_{reg} \geq \mathcal{L}_{memory} = \\ \mathbf{E}_{p^{f,n}_{t,t+1}}[\|B(x_{t+1}) - x_{t+1} - (F^n_t(x_t) - \alpha F^n_t(x_{t+1}) - (1-\alpha)F^n_{t+1}(x_{t+1}))\|^2].\end{gathered} \tag{3.14}$$

Note that stable forward and backward processes in an ideal scenario should

satisfy the following

$$B_{t+1}(F_t(x_t)) = x_t, \quad F_t(B_{t+1}(x_{t+1})) = x_{t+1}, \quad t \in \{0, \dots, T-1\}. \tag{3.15}$$

Note that the above relation can be considered as a cycle-consistency constraint proposed by CycleGAN [29]. It is essential relation that must hold for entire stochastic processes, but SB-based processes with small timesteps may not attain this. Therefore, to reduce the number of iterations and timesteps required for convergence while maintaining the stability of the constructed stochastic process, the above cycle-consistency relation can be used as an additional regularization term to the objective function explicitly. For this, the above relation 3.15 can be expressed as

$$\begin{gathered}\mathbf{E}_{x_{t+1}\sim\, p^f_{t+1|t}}[B_{t+1}(x_{t+1})] = x_t, \quad \mathbf{E}_{x_t\sim\, p^b_{t|t+1}}[F_t(x_t)] = x_{t+1}, \\ t \in \{0, \dots, T-1\}.\end{gathered} \tag{3.16}$$

And again, it can be formulated as the following regression problem:

$$\mathcal{L}_{cyc} = \mathbf{E}_{p^{f,n}_{t,t+1}}[\|B_{t+1}(x_{t+1}) - x_t\|^2]. \tag{3.17}$$

This objective, $\mathcal{L}_{cyc}$, is an additional term of $\mathcal{L}_{memory}$. Note that all objectives $\mathcal{L}_{DSB}$, $\mathcal{L}_{reg}$, and $\mathcal{L}_{cyc}$ have $B_{t+1}(x_{t+1})$ term, this is the only term that is evaluated during training. Thus, for efficient training, $B_{t+1}(x_{t+1})$ should be evaluated once. If $\mathcal{L}_{DSB}$, $\mathcal{L}_{reg}$, and $\mathcal{L}_{cyc}$ terms are used separately, evaluation of $B_{t+1}(x_{t+1})$ should occur multiple times, which is inefficient. By the convexity of $\|\cdot\|^2$ and with the proper setting of the weight β, the memory-efficient loss objective can be obtained

as

$$\mathcal{L}_B = \mathbf{E}_{p^{f,n}_{t,t+1}}\Big[\|B_{t+1}(x_{t+1}) - \frac{1}{\beta+1}\left(x_{t+1} - (F^n_t(x_t) - \alpha F^n_t(x_{t+1}) - \right.$$
$$\left.(1-\alpha)F^n_{t+1}(x_{t+1}))\right) - \frac{\beta}{\beta+1}x_t\|^2\Big] \leq \frac{1}{\beta+1}\mathcal{L}_{memory} + \frac{\beta}{\beta+1}\mathcal{L}_{cyc}. \tag{3.18}$$

Similarly, the loss objective for updating F^{n+1}_t can be obtained as

$$\mathcal{L}_F = \mathbf{E}_{p^{b,n}_{t,t+1}}\Big[\|F_t(x_t) - \frac{1}{\beta+1}\left(x_t - (B^n_{t+1}(x_{t+1}) - \alpha B^n_{t+1}(x_t) - \right.$$
$$\left.(1-\alpha)B^n_t(x_t))\right) - \frac{\beta}{\beta+1}x_{t+1}\|^2\Big]. \tag{3.19}$$

Thus, the regularized SB-based model, **RSB**, is trained for the desired bidirectional stochastic processes by alternating between $\mathcal{L}_B$ and $\mathcal{L}_F$. Since the objective was set to memory-efficient style, model evaluation $B_{t+1}(x_{t+1})$ or $F_t(x_t)$ proceeds only one for each update. The remaining term of the objectives can be effectively obtained by a replay-memory [16].

Chapter 4

Experiments

Based on the regularized SB-based formulation, the proposed RSB experimentally demonstrated that it can train stochastic processes between any two data spaces with relatively small timesteps compared to the previous SB models. And its training was more stable and faster. Since the SB-based stochastic process is free from the constraints of starting at $\mathcal{Z}$, the experiments were conducted on both unconditional and conditional generation tasks. And for both types of tasks, the proposed RSB confirmed its effectiveness.

4.1 Dataset

2D Toy, MNIST, and CelebA were used as datasets for qualitative performance evaluation of RSB. The 2D Toy dataset consists of intuitive 2-dimensional data, including 8-Gaussian, Checkerboard, 25-Gaussian, and Circles. And the MNIST is one of the most widely known datasets in deep learning research and consists of digits ranging from 0 to 9. Lastly, the CelebA dataset, where various male and

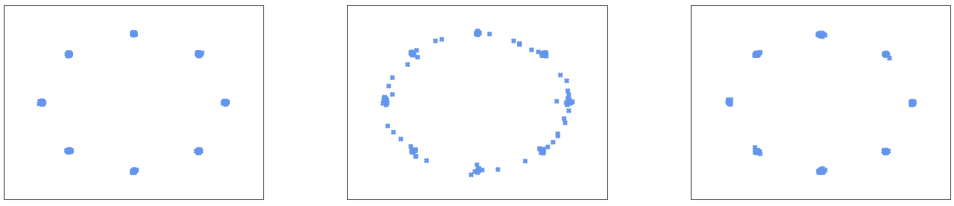
female faces exist at various ages, was also used to determine whether the RSB can handle the relatively high-resolution image domain.

4.2 Training

A vanilla GAN model with a gradient penalty (GP) of the form [15] was trained on the 2D Toy to compare with the proposed RSB. And for other datasets except for the 2D Toy dataset, the NCSN++ architecture of SGM [23] was used. In addition, since RSB requires multiple steps of model evaluation, the training can be very slow if the model is evaluated for each update iteration. To mitigate this issue, the replay memory was generated by inferencing with a large batch size at once and the generated replay memory was used for multiple iterations. Also, for faster training, in the case of unconditional generation, since the forward noising process from $\mathcal{P}$ to $\mathcal{Z}$ is relatively easy to be trained, half of the number of iterations was used at each IPF stage compared to training of the backward process.

4.3 Results

The existing SB models, DSB and SB-FBSDE, can be compared to RSB for performance evaluation. Since SB-FBSDE depends heavily on continuous-time SDEs, SB-FBSDE requires 100 steps even for a 2D Toy dataset. Thus it could not be trained for small timesteps such as $T = 4, 8$. And since DSB was formulated in a relatively discrete-time setting compared to SB-FBSDE, DSB showed better performance when the number of timesteps is restricted to be small. Thus, the proposed RSB was compared to DSB.



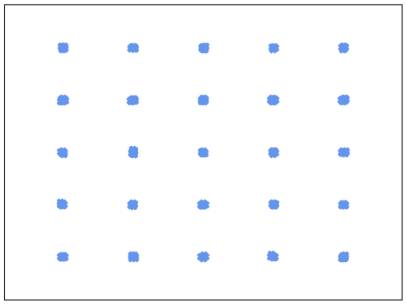
(a) Ground Truth (b) DSB (c) **RSB**

Figure 4.1: Qualitative results on 8-Gaussians of 2D Toy

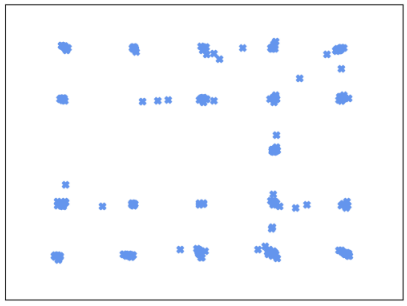
4.3.1 Results with 2D Toy

Firstly, RSB was trained for unconditional generation of 8-Gaussians of 2D Toy. Both RSB and DSB were trained with 8 timesteps, 10K iterations for the forward process, and 20K iterations for the backward process. And hyperparameters for RSB was set to $\alpha = 0.5$ and $\beta = 2.5$. See Figure 4.1 for comparison. While DSB didn't converge to the desired data space, RSB almost converged.

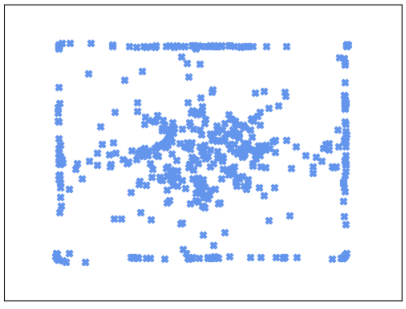
Next, an unconditional generation of 25-Gaussians which is much more complicated than 8-Gaussians was testified. Both RSB and DSB were trained with 8 timesteps, 20K iterations for the forward process, and 40K iterations for the backward process. And hyperparameters for RSB was set to $\alpha = 0.5$ and $\beta = 5$. And GAN was trained for 60K iterations. See Figure 4.2 for comparison between DSB, RSB and GAN. While DSB showed instability where the training did not progress significantly after the intermediate stage, RSB showed fast convergence in the intermediate stage, and the rest of the training stayed stable. This indicates that the existing SB-based methods are not suitable for a small number of discretized timesteps while RSB is. And although GAN was trained for a sufficient training time, mode collapse occurred that it could not cover all modes of 25-Gaussians. The



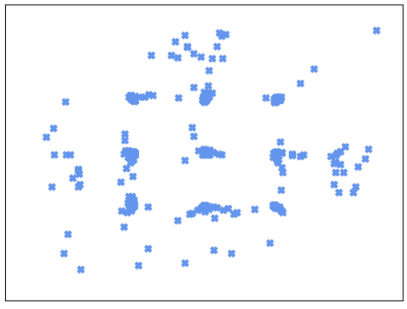
(a) Ground Truth



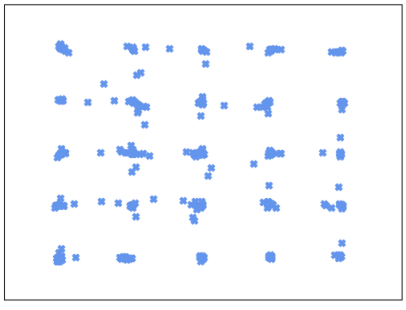
(b) GAN



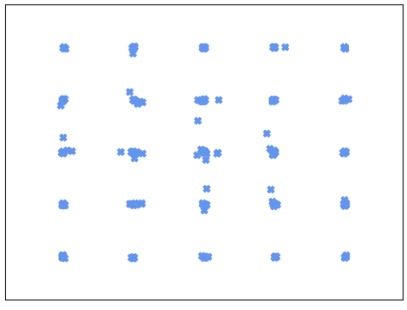
(c) DSB(intermediate)



(d) DSB



(e) **RSB**(intermediate)



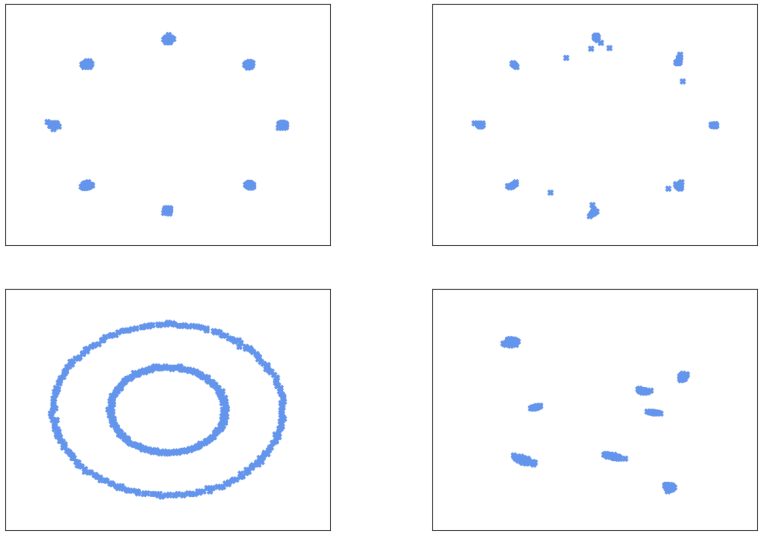
(f) **RSB**

Figure 4.2: Qualitative results on 25-Gaussians of 2D Toy

result shows that stochastic-process-based generative modeling can complement the existing function-based one.

Since the SB-based formulation enables a stochastic process between any data spaces, the proposed RSB was tested on the data translation, *i.e.* the case of conditional generation, between 8-Gaussians and Circles data space. Both RSB and DSB were trained with 8 timesteps, 10K iterations for the forward process, and 20K iterations for the backward process. And hyperparameters for RSB was set to $\alpha = 0.5$ and $\beta = 2.5$. And GAN was trained for 30K iterations. See Figure 4.3 for comparison. An interesting result is that when the conventional GAN was trained to generate the Circles data space from the 8-Gaussians data space, not the latent space, the training was not done properly. And it can be confirmed that RSB obtained better translation performance than DSB. In addition, it can be visually confirmed that the trajectories of the trained stochastic process by RSB and DSB are different. See 4.4 and 4.5. When comparing the forward process from Circles to 8-Gaussians, the outer circle gathers in the form of 8-Gaussians and the central circle scatters toward the edge in DSB. In the case of RSB, the outer circle and the central circle are scattered and gathered to create 8 modes and they move to the desired place. Therefore, it can be concluded that applying regularization to the existing SB-based formulation changes the trajectories drawn by stochastic processes leading to fast training with smaller timesteps required.

These results from the 2D Toy show the possibility that the failure modes existing in both unconditional and conditional generation of GANs can be improved through stochastic-process-based generative models.



(a) Ground Truth (b) GAN

(c) DSB (d) **RSB**

Figure 4.3: Qualitative results on data translation task between 8-Gaussians and Circles of 2D Toy

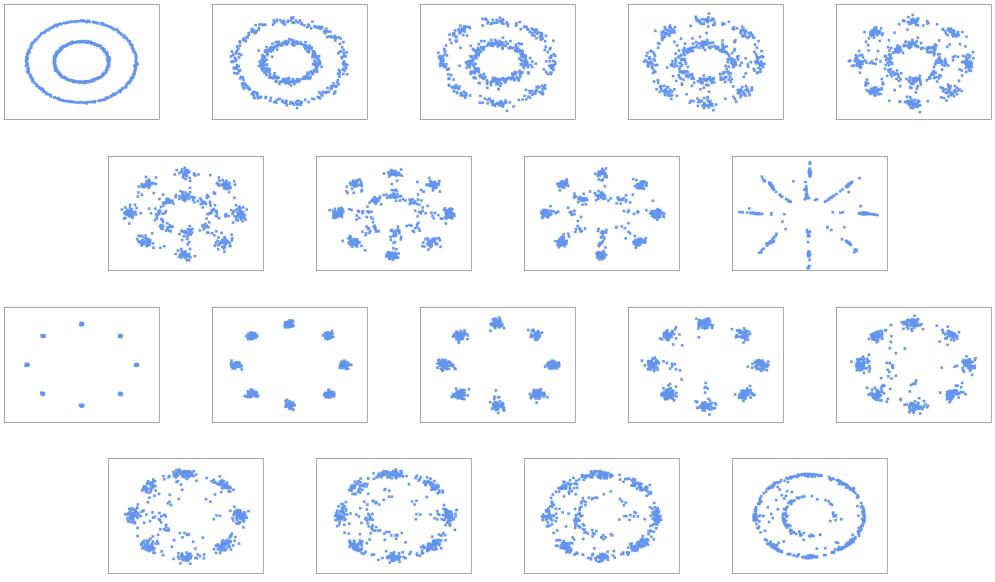


Figure 4.4: Detailed qualitative results of DSB on data translation between 8-Gaussians and Circles of 2D Toy. The top two rows illustrate the forward process from Circles to 8-Gaussians. And other rows illustrate the backward process from 8-Gaussians to Circles.

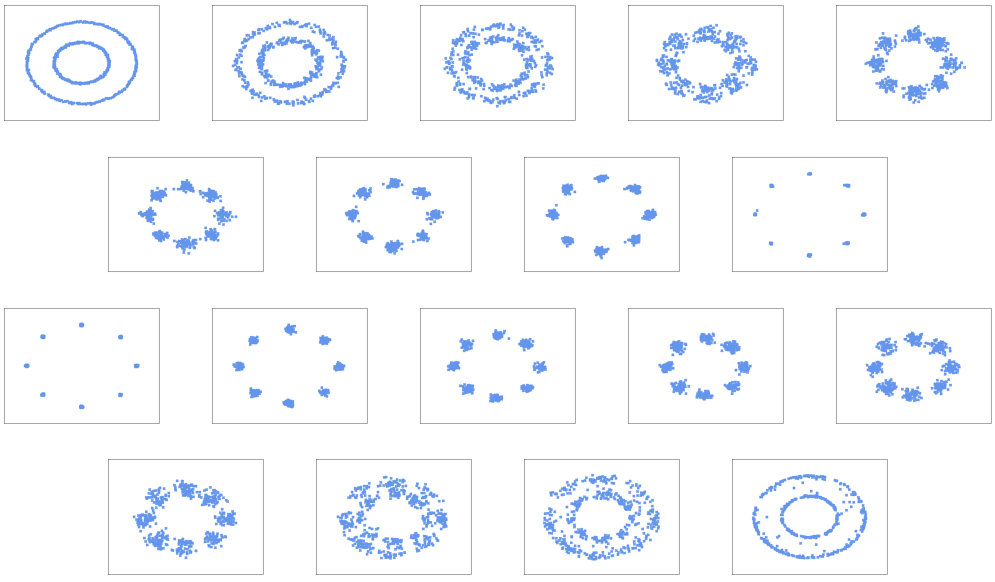


Figure 4.5: Detailed qualitative results of RSB on data translation between 8-Gaussians and Circles of 2D Toy. The top two rows illustrate the forward process from Circles to 8-Gaussians. And other rows illustrate the backward process from 8-Gaussians to Circles.

(a) Ground Truth (b) DSB (c) **RSB**

Figure 4.6: Qualitative results of MNIST generation

4.3.2 Results with MNIST

With the MNIST dataset, RSB and DSB were compared for an unconditional generation task. Both RSB and DSB were trained with 16 timesteps, 8K iterations for the forward process, and 16K iterations for the backward process. And hyperparameters for RSB was set to $\alpha = 0.5$ and $\beta = 5$. The total iterations spent for training are relatively insufficient for training the existing SB models. See Figure 4.6 for results. In the case of DSB, the training has not progressed significantly, but in the case of RSB, the training state has significantly progressed. It indicates that by adding regularization to the SB-based formulation, RSB reduces the number of iterations required for sufficient training compared to the existing SB models.

4.3.3 Results with CelebA

Recall that the advantage of SB-based formulation is that it can construct bidirectional stochastic processes between any two data spaces, eliminating the need for conditioning the generation starting from the latent space. Therefore, the proposed RSB explored this possibility through the task of image-to-image translation and the single image super-resolution. The previous SB models [3, 6] mainly demonstrated their performance on unconditional generation tasks, while the image size stayed in 32x32 size. But, since the RSB reduced the required number of timesteps and training time, it experimentally confirmed that the SB-based process could play a role in the relatively high-resolution data space of the CelebA dataset with 128x128 size.

The image-to-image translation task is to translate the image of the source data space to that of the target data space while maintaining the semantic information of the source image. There are various possible scenarios in image-to-image trans-

(a) Source Image

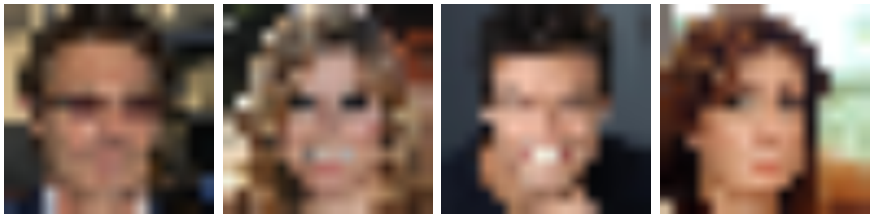
(b) Translated Image

Figure 4.7: Qualitative results on image-to-image translation task between male and female of CelebA. The top row illustrates the source images. And the bottom row illustrates the translated images.

lation, and in this experiment, the translation between male and female faces was considered. The RSB trained the discrete-time stochastic processes between the data space of the male face and the female face. And the forward and backward processes were trained with $\alpha = 0.5$, $\beta = 10$, 4 timesteps, and 48K total iterations each.

The qualitative results from RSB are in Figure 4.7. And Figure 4.8 shows the

Figure 4.8: Detailed translation process from male to female with 4 timesteps



(a) LR(×8) Input

(b) SR Output

Figure 4.9: Qualitative results on super-resolution task of CelebA. The top row illustrates the LR images. And the bottom row illustrates the SR outputs.

detailed translation process trained by the regularized stochastic process with 4 timesteps only. The results show that the face of the source domain can be translated to the target domain while maintaining semantic information such as identity, facial expression, and pose. Note that masculinity or femininity was properly changed without a large change in hairstyles. It is seen as a result of SB-based formulation as an OT problem with implicit cycle-consistency. And it seems that implicit cycle-consistency was strongly applied.

Similarly, the RSB was trained to handle the single image super-resolution task. In this task, RSB constructed the stochastic processes between the data space of low-resolution images and super-resolution images. And it was trained with $\alpha = 0.5$, $\beta = 10$, 8 timesteps, and 48K total iterations for each forward and backward process. The qualitative results from RSB are in Figure 4.9. And Figure 4.10 shows the detailed super-resolution process of trained RSB. The desired super-resolution

Figure 4.10: Detailed super-resolution process from LR(×8) to SR

(SR) space was set to be 128x128 size and the low-resolution (LR) input was downsampled with ×8 scale. The results show that the desired SR outputs were attained while maintaining the semantic information of LR input properly.

Chapter 5

Conclusion

This study tried to utilize bidirectional stochastic processes based on the Schrödinger bridge (SB) problem for deep generative modeling. The existing SB-based generative models have been proposed to improve the slow sampling speed of diffusion models and showed their potential. However, compared to generative models such as GANs, a large number of timesteps and a long training time are still required. This study aimed to reduce the number of timesteps and training time required by the existing SB models. In the existing SB-based framework, the bidirectional stochastic processes become unstable with a small number of discretization timesteps because they are not consistent with each other. Therefore, this work proposed regularization terms to maintain the consistency between the bidirectional stochastic processes. By applying this regularized SB-based process to both conditional and unconditional generation tasks, it was possible to properly train stochastic processes between two arbitrary distributions even with smaller timesteps.

Bibliography

[1] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein generative adversarial networks*, in International conference on machine learning, PMLR, 2017, pp. 214–223.

[2] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, *Unsupervised learning of visual features by contrasting cluster assignments*, Advances in Neural Information Processing Systems, 33 (2020), pp. 9912–9924.

[3] T. Chen, G.-H. Liu, and E. A. Theodorou, *Likelihood training of schrödinger bridge using forward-backward sdes theory*, in International Conference on Learning Representations, 2022.

[4] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, *Ilvr: Conditioning method for denoising diffusion probabilistic models*, arXiv preprint arXiv:2108.02938, (2021).

[5] E. de Bézenac, I. Ayed, and P. Gallinari, *Optimal unsupervised domain translation*, arXiv preprint arXiv:1906.01292, (2019).

[6] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, *Diffusion schrödinger bridge with applications to score-based generative modeling*, Advances in Neural Information Processing Systems, 34 (2021).

[7] P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, Advances in Neural Information Processing Systems, 34 (2021).

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 27 (2014).

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

[10] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, *Gotta go fast when generating data with score-based models*, arXiv preprint arXiv:2105.14080, (2021).

[11] D. Kim, B. Na, S. J. Kwon, D. Lee, W. Kang, and I.-C. Moon, *Maximum likelihood training of implicit nonlinear diffusion models*, arXiv preprint arXiv:2205.13699, (2022).

[12] D. P. Kingma and P. Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions*, Advances in neural information processing systems, 31 (2018).

[13] D. P. Kingma and M. Welling, *An introduction to variational autoencoders*, arXiv preprint arXiv:1906.02691, (2019).

[14] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, *Sdedit: Guided image synthesis and editing with stochastic differential equations*, in International Conference on Learning Representations, 2021.

[15] L. Mescheder, A. Geiger, and S. Nowozin, *Which training methods for gans do actually converge?*, in International conference on machine learning, PMLR, 2018, pp. 3481–3490.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing atari with deep reinforcement learning*, arXiv preprint arXiv:1312.5602, (2013).

[17] M. Pavon and A. Wakolbinger, *On Free Energy, Stochastic Control, and Schrödinger Processes*, Birkhäuser Boston, Boston, MA, 1991, pp. 334–348.

[18] G. Peyré, M. Cuturi, et al., *Computational optimal transport*, Center for Research in Economics and Statistics Working Papers, (2017).

[19] 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).

[20] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al., *Photorealistic text-to-image diffusion models with deep language understanding*, arXiv preprint arXiv:2205.11487, (2022).

[21] F. Santambrogio, *Optimal transport for applied mathematicians*, Birkäuser, NY, 55 (2015), p. 94.

[22] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, Advances in Neural Information Processing Systems, 32 (2019).

[23] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, arXiv preprint arXiv:2011.13456, (2020).

[24] A. Vahdat, K. Kreis, and J. Kautz, *Score-based generative modeling in latent space*, Advances in Neural Information Processing Systems, 34 (2021), pp. 11287–11302.

[25] F. Vargas, *Machine-learning approaches for the empirical schrödinger bridge problem*, tech. rep., University of Cambridge, Computer Laboratory, 2021.

[26] P. Vincent, *A connection between score matching and denoising autoencoders*, Neural computation, 23 (2011), pp. 1661–1674.

[27] G. Wang, Y. Jiao, Q. Xu, Y. Wang, and C. Yang, *Deep generative learning via schrödinger bridge*, in International Conference on Machine Learning, PMLR, 2021, pp. 10794–10804.

[28] Z. Xiao, K. Kreis, and A. Vahdat, *Tackling the generative learning trilemma with denoising diffusion gans*, arXiv preprint arXiv:2112.07804, (2021).

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

국문초록

기존의 심층 생성 모델링의 함수 기반 모델들과 비교하여, 최근 제안된 확산 생성 모델은 확률 과정 기반의 접근을 통해 우수한 성능을 달성했다. 그러나 이 접근 방식은 이산화를 위한 많은 수의 타임스텝으로 인해 긴 샘플링 시간을 필요로 한다. 슈뢰딩거 브리지 기반 모델은 분포 간의 양방향 확률 과정을 학습하여 이러한 문제를 해결하려고 시도한다. 그러나 이 역시 생성적 적대 모델과 같은 생성 모델들에 비하면 샘플링 속도가 여전히 느리다. 그리고 양방향 확률 과정의 학습으로 인해 상대적으로 긴 학습 시간을 필요로 한다. 따라서 본 연구는 필요한 타임스텝 수와 학습 시간을 줄이는 것을 시도하였고 기존의 슈뢰딩거 브리지 모델에 정칙화 항들을 제안하여 감소된 타임스텝에서도 양방향 확률 과정을 일관적이고 안정적으로 만들었다. 각 정칙화 항들은 계산 시간과 메모리 사용에서 보다 효율적인 훈련을 가능하게 하기 위해 하나의 항으로 통합되었다. 이렇게 정칙화된 확률 과정을 다양한 생성 문제에 적용하여 서로 다른 분포 간에 원하는 변환들을 얻을 수 있었고 이에 더 빠른 샘플링 속도를 가지는 확률과정 기반의 생성 모델링의 가능성이 확인될 수 있었다.

주요어휘: 딥러닝, 생성 모델, 확률 과정, 분산 생성 모델, 슈뢰딩거 브리지

학번: 2020-22722