Stochastic Optimal Control for Diffusion Bridges in Function Spaces

Byoungwoo Park[1], **Jungwon Choi**[1], **Sungbin Lim**[2,3]*, **Juho Lee**[1]*
KAIST[1], Korea University[2], LG AI Research[3]
{bw.park, jungwon.choi, juholee}@kaist.ac.kr,
sungbin@korea.ac.kr

Abstract

Recent advancements in diffusion models and diffusion bridges primarily focus on finite-dimensional spaces, yet many real-world problems necessitate operations in infinite-dimensional function spaces for more natural and interpretable formulations. In this paper, we present a theory of stochastic optimal control (SOC) tailored to infinite-dimensional spaces, aiming to extend diffusion-based algorithms to function spaces. Specifically, we demonstrate how Doob's h-transform, the fundamental tool for constructing diffusion bridges, can be derived from the SOC perspective and expanded to infinite dimensions. This expansion presents a challenge, as infinite-dimensional spaces typically lack closed-form densities. Leveraging our theory, we establish that solving the optimal control problem with a specific objective function choice is equivalent to learning diffusion-based generative models. We propose two applications: 1) learning bridges between two infinite-dimensional distributions and 2) generative models for sampling from an infinite-dimensional distribution. Our approach proves effective for diverse problems involving continuous function space representations, such as resolution-free images, time-series data, and probability density functions. Code is available at https://github.com/bw-park/DBFS.

1 Introduction

Stochastic Optimal Control (SOC) is designed to steer a noisy system toward a desired state by minimizing a specific cost function. This methodology finds extensive applications across various fields in science and engineering, including rate event simulation [33, 35], stochastic filtering and data assimilation [47, 55, 72], non-convex optimization [9], modeling population dynamics [8, 43]. SOC is also related to diffusion-based sampling methods that are predominant in machine learning literature. Specifically, if we choose the terminal cost of a control problem as the log density ratio between a target distribution and a simple prior distribution, solving the optimal control reduces to learning a diffusion-based generative models [56, 73, 77] built upon the Schrödinger bridge problem [12, 54].

While SOC associated diffusion-based generative models have been well-established for finite-dimensional spaces [5, 10, 11, 44], their theoretical foundations and practical algorithms for infinite-dimensional spaces remain underexplored. There is a growing interest in developing generative models in function spaces. Examples include learning neural operators for partial differential equations (PDEs) [37, 39, 40], interpreting images as discretized functions through implicit neural representations (INRs) [21, 63], and operating in function spaces for Bayesian neural networks (BNNs) [65, 74]. Models that operate in function spaces are more parameter-efficient as they avoid resolution-specific parameterizations. In response to this demand, several extensions of

*Equal advising.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

diffusion-based generative models for infinite-dimensional function spaces have been proposed [3, 25, 32, 41, 42, 53]. However, SOC theory for building diffusion bridges in function spaces is still demanding in the community of generative modeling.

To address these challenges, this work introduces an extension of SOC for diffusion bridges in infinite-dimensional Hilbert spaces, particularly focusing on its applications in sampling problems. Specifically, we demonstrate the idea of Doob's h-transform [13, 59] can be derived from SOC theory and extend its formulation into Hilbert spaces. Due to the absence of a density with respect to the Lebesgue measure in infinite-dimensional spaces, building a diffusion bridge between function spaces is a nontrivial task separated from the finite-dimensional cases [51, 61]. To this end, we propose a Radon-Nikodym derivative relative to a specified Gaussian reference measure in Hilbert space. Leveraging the infinite-dimensional Doob's h-transform and SOC, we then formulate diffusion bridge-based sampling algorithms in function spaces. While the infinite-dimensional Doob's h-transform has already been derived in [2, 26, 31], our main goal is not merely to derive it. Instead, we aim to generalize various finite-dimensional sampling problems [51, 61, 73, 77] into the infinite-dimensional space by exploiting the theory of infinite-dimensional SOC.

To demonstrate the applicability of our theory, we present learning algorithms for two representative problems. First, we introduce an infinite-dimensional bridge-matching algorithm as an extension of previous methods [51, 61] into Hilbert spaces, which learns a generative model to bridge two distributions defined in function spaces. As an example, we show that our framework can learn smooth transitions between two image distributions in a resolution-free manner. Second, we propose a simulation-based Bayesian inference algorithm [73, 77] that operates in function space. Instead of directly approximating the target Bayesian posterior, our algorithm learns a stochastic transition from the prior to the posterior within the function space. We demonstrate the utility of this approach by inferring Bayesian posteriors of stochastic processes, such as Gaussian processes. We summarize our contributions as follows:

- Based on the SOC theory, we derive the Doob's h-transform in Hilbert spaces. We propose a h function as a Randon-Nikodym derivative with respect to a Gaussian measure in infinite-dimensional space.
- Based on the infinite-dimensional extension of the Doob's h-transform, we present the diffusion bridge and simulation-based Bayesian inference algorithm in function spaces.
- We demonstrate our method for various real-world problems involving function spaces, including resolution-free image translation and posterior sampling for stochastic processes.

Notation. Consider a real and separable Hilbert space $\mathcal{H}$ with the norm and inner product denoted by $\|\cdot\|_{\mathcal{H}}$ and $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. Throughout the paper, we consider a path measure denoted by $\mathbb{P}^{(\cdot)}$ on the space of all continuous mappings $\Omega = C([0,T],\mathcal{H})$. The stochastic processes associated with this path measure $\mathbb{P}^{(\cdot)}$ are denoted by $\mathbf{X}^{(\cdot)}$, and their time-marginal distribution at time $t \in [0,T]$ as push-forward measure $\mu_t^{(\cdot)} := (\mathbf{X}_t^{(\cdot)})_{\#}\mathbb{P}^{(\cdot)}$. Furthermore, for a function $\mathcal{V} : [0,T] \times \mathcal{H} \to \mathbb{R}$, we define $D_{\mathbf{x}}\mathcal{V}, D_{\mathbf{xx}}\mathcal{V}$ as the first and second order Fréchet derivatives with respect to the variable $\mathbf{x} \in \mathcal{H}$, respectively, and $\partial_t\mathcal{V}$ as the derivative with respect to the time variable $t \in [0,T]$.

2 A Foundation on Stochastic Optimal Control for Diffusion Bridges

In this section, we first present a brief introduction to the theory of stochastic optimal control (SOC) in infinite dimensions and the Verification Theorem (Lemma 2.1), which is the key to understanding the theoretical connection between SOC and the diffusion bridges. Then we propose Doob's h-transform in Hilbert spaces based on the SOC theory (Theorems 2.2 and 2.3).

2.1 Preliminaries

Gaussian Measure and Cameron-Martin Space. Let $(\Omega,\mathcal{F},\mathbb{Q})$ and $(\mathcal{H},\mathcal{B}(\mathcal{H}))$ be two measurable spaces and consider $\mathcal{H}$-valued random variable $\mathbf{X} : \Omega \to \mathcal{H}$ such that the push-forward measure $\mu := \mathbf{X}_{\#}\mathbb{Q}$ induced by $\mathbf{X}$ is Gaussian, i.e., a real-valued random variable $\langle u, \mathbf{X}\rangle_{\mathcal{H}}$ follows Gaussian distribution on $\mathbb{R}$ for any $u \in \mathcal{H}$. Then, there exist a unique mean $m_{\mathbf{X}} \in \mathcal{H}$ given by $\langle m_{\mathbf{X}}, u\rangle_{\mathcal{H}} = \mathbf{E}_\mu[\langle \mathbf{X}, u\rangle_{\mathcal{H}}]$ and a nonnegative, symmetric, and self-adjoint covariance operator $Q : \mathcal{H} \to \mathcal{H}$ defined by $\langle u, Qv\rangle_{\mathcal{H}} = \mathbb{E}_\mu[\langle \mathbf{X} - m_{\mathbf{X}}, u\rangle_{\mathcal{H}}, \langle \mathbf{X} - m_{\mathbf{X}}, v\rangle_{\mathcal{H}}]$ for any $u, v \in \mathcal{H}$.

We write $\mu = \mathcal{N}(m_{\mathbf{X}}, Q)$ and say $\mathbf{X}$ is centered if $m_{\mathbf{X}} = 0$. For a covariance operator Q, we assume $\mathcal{H}$-valued centered $\mathbf{X}$ follows the distribution $\mathcal{N}(0, Q)$ supported on $\mathcal{H}$ so that Q is guaranteed to be compact. Hence, there exists an eigen-system $\{(\lambda^{(k)}, \phi^{(k)}) \in \mathbb{R} \times \mathcal{H} : k \in \mathbb{N}\}$ such that $Q(\phi^{(k)}) = \lambda^{(k)}\phi^{(k)}$ holds and $\text{Tr}(Q) = \sum_{k=1}^{\infty} \lambda^{(k)} < \infty$. We define the *Cameron-Martin space* by $\mathcal{H}_0 := Q^{1/2}(\mathcal{H})$, $\mathcal{H}_0 \subseteq \mathcal{H}$ is a separable Hilbert space endowed with the inner product $\langle u, v\rangle_{\mathcal{H}_0} := \langle Q^{-1/2}u, Q^{-1/2}v\rangle_{\mathcal{H}}$.

Stochastic Differential Equations in Hilbert Spaces. The standard $\mathbb{R}^d$-valued Wiener process has independent increments $w_{t+\Delta_t} - w_t \sim \mathcal{N}(0, \Delta_t \mathbf{I}_d)$. In the case of infinite-dimensional Hilbert space $\mathcal{H}_0$, however, such identity covariance may not be a trace class. We consider a larger space $\mathcal{H}_0 \subset \mathcal{H}_1$ such that $\mathcal{H}_0$ is embedded into $\mathcal{H}_1$ with a Hilbert-Schmidt embedding $J : \mathcal{H}_0 \to \mathcal{H}_1$. Let $Q := JJ^*$ and we define $\mathcal{H}_1$-valued Q-Wiener process [17, Proposition 4.7] as $\mathbf{W}_t^Q = \sum_{k=1}^{\infty} Q^{1/2}\phi^{(k)} w_t^{(k)}$. We focus on the Cameron-Martin space $\mathcal{H}_0$ where the Wiener process has increments $\mathcal{N}(0, \Delta_t \mathbf{I}_{\mathcal{H}})$, and a larger space $\mathcal{H}_1 = \mathcal{H}$, where the Wiener process has increments $\mathcal{N}(0, \Delta_t Q)$. Then we define a path measure $\mathbb{P}$ and associated stochastic differential equation (SDE) in $\mathcal{H}$ as follows

$$d\mathbf{X}_t = \mathcal{A}\mathbf{X}_t + \sigma d\mathbf{W}_t^Q, \quad \mathbf{X}_0 \in \mathcal{H}, \quad t \in [0, T], \tag{1}$$

where $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ is a linear operator, a constant $\sigma > 0$ and a Q-Wiener process $\mathbf{W}^Q$ defined on a probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$. For a more comprehensive understanding, see [17, 29].

2.2 Stochastic Optimal Control in Hilbert Spaces

From the *uncontrolled* SDE introduced in equation (1), various sampling problems in $\mathbb{R}^d$ including density sampling [5, 73, 77] and generative modeling [10, 11] can be solved by adjusting the SDE with proper drift (control) function $\alpha \in \mathbb{R}^d$. Motivated by these approaches, introducing stochastic optimal control (SOC) to solve real-world sampling problems, we aim to introduce SOC to the infinite-dimensional Hilbert space $\mathcal{H}$. We consider that a controlled path measures $\mathbb{P}^\alpha$ is induced by following infinite-dimensional SDE defined as follows:

$$d\mathbf{X}_t^\alpha = \left[\mathcal{A}\mathbf{X}_t^\alpha + \sigma Q^{1/2}\alpha_t\right] dt + \sigma d\tilde{\mathbf{W}}_t^Q, \quad \mathbf{X}_0^\alpha = \mathbf{x}_0, \quad t \in [0, T] \tag{2}$$

where $\alpha_{(\cdot)} : [0, T] \times \mathcal{H} \to \mathcal{H}$ is infinite-dimensional Markov control (see Section A.1.2 for more details). We refer to the SDE in equation (2) as *controlled* SDE. The controlled SDE can be exploited to the various problems. In general, it can be done by finding the optimal control function that minimizes the objective functional with suitably chosen cost functionals depending on the problem:

$$\mathcal{J}(t, \mathbf{x}_t, \alpha) = \mathbb{E}_{\mathbb{P}^\alpha}\left[\int_t^T [R(\alpha_s)]\, ds + G(\mathbf{X}_T^\alpha)\middle|\mathbf{X}_t^\alpha = \mathbf{x}_t\right], \tag{3}$$

where $R : \mathcal{H} \to \mathbb{R}$ are running cost and $G : \mathcal{H} \to \mathbb{R}$ is terminal cost. The measurable function $\mathcal{J}(t, \mathbf{x}, \alpha)$, representing the total cost incurred by the control α over the interval $[t, T]$, given that the control strategy from the interval $[0, t]$ has resulted in $\mathbf{X}_t^\alpha = \mathbf{x}$. The objective is to minimize the objective functional in (3) over all admissible control policies $\alpha \in \mathcal{U}$, where $\mathcal{U}$ is the Hilbert space of all square-integrable $\mathcal{H}$-valued processes adapted to $\mathbf{W}^Q$ defined on $[0, T]$. Then we define the *value function* $\mathcal{V}(t, \mathbf{x}) = \inf_{\alpha\in\mathcal{U}} \mathcal{J}(t, \mathbf{x}, \alpha)$, the optimal costs conditioned on $(t, \mathbf{x}) \in [0, T] \times \mathcal{H}$. By using the dynamic programming [23], we can solve the Hamilton-Jacobi-Bellman (HJB) equation,

$$\partial_t \mathcal{V}_t + \mathcal{L}\mathcal{V}_t + \inf_{\alpha\in\mathcal{U}}\left[\langle \alpha, \sigma Q^{1/2} D_{\mathbf{x}}\mathcal{V}_t\rangle + R\right] = 0, \quad \mathcal{V}(T, \mathbf{x}) = G(\mathbf{x}), \tag{4}$$

where $\mathcal{L}\mathcal{V}_t := \langle \mathbf{X}_t^\alpha, \mathcal{A}D_{\mathbf{x}}\mathcal{V}_t\rangle_{\mathcal{H}} + \frac{1}{2}\text{Tr}\left[\sigma^2 Q D_{\mathbf{xx}}\mathcal{V}_t\right]$. We demonstrate that with specific choices of cost functionals R and G in (3), the optimal control $\alpha^\star$ of the minimization problem aligns with the proper drift function for sampling problems we will discuss later. To do so, we start with how the HJB equation can characterize the optimal controls.

Lemma 2.1 (Verification Theorem)**.** *Let $\mathcal{V}$ be a solution of HJB equation (4) with $R(\alpha) := \frac{1}{2}\|\alpha\|_{\mathcal{H}}^2$ satisfying the assumptions in A.1. Then, we have $\mathcal{V}(t, \mathbf{x}) \leq \mathcal{J}(t, x, \alpha)$ for every $\alpha \in \mathcal{U}$ and $(t, \mathbf{x}) \in [0, T] \times \mathcal{H}$. Let $(\alpha^*, \mathbf{X}^{\alpha^*})$ be an admissible pair such that*

$$\alpha_s^* = \arg\inf_{\alpha\in\mathcal{U}}\left[\langle \alpha_s, \sigma Q^{1/2} D_{\mathbf{x}}\mathcal{V}_t\rangle + \frac{1}{2}\|\alpha_s\|_{\mathcal{H}}^2\right] = -\sigma Q^{1/2} D_{\mathbf{x}}\mathcal{V}(s, \mathbf{X}_s^{\alpha^*}) \tag{5}$$

for almost every $s \in [t, T]$ and $\mathbb{P}$-almost surely. Then $(\alpha^*, \mathbf{X}^{\alpha^*})$ satisfying $\mathcal{V}(t, \mathbf{x}) = \mathcal{J}(t, \mathbf{x}, \alpha^*)$.

Lemma 2.1 demonstrate that with specific choices of running costs, the solution to the HJB equation in (4) is the optimal cost of the minimization problem in (3) with the closed-form optimal control $\alpha_t^* = -\sigma Q^{1/2} D\mathcal{V}_t$. In the subsequent subsection, we reveal the connection between the optimal controlled process, characterized by the controlled SDE with optimal control $\alpha^\star$, and the conditioned SDE in $\mathcal{H}$. Additionally, we show various problems depending on the choice of the terminal cost G.

2.3 Doob's h-transform for Diffusion Bridges in Hilbert Spaces

The *conditioned* SDE is a stochastic process that is guaranteed to satisfy a set constraint defined over the interval $[0, T]$. For instance, a *Diffusion Bridge* is a stochastic process that satisfies both terminal constraints $\mathbf{X}_0 = \mathbf{x}_0$, $\mathbf{X}_T = \mathbf{x}_T$ for any $\mathbf{x}_0, \mathbf{x}_T \in \mathcal{H}$. We will show that with a proper choice of the terminal cost G, the conditioned SDE is a special case of controlled SDE. For this, let us define the function $h : [0, T] \times \mathcal{H} \to \mathbb{R}$:

$$h(t, \mathbf{x}) = \int_{\mathcal{H}} \tilde{G}(\mathbf{z}) \mathcal{N}_{e^{(T-t)\mathcal{A}}\mathbf{x}, Q_{T-t}}(d\mathbf{z}) = \mathbb{E}_{\mathbb{P}} \left[\tilde{G}(\mathbf{X}_T) | \mathbf{X}_t = \mathbf{x} \right], \tag{6}$$

where $\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x}, Q_t}$ is a Gaussian measure with mean function $e^{t\mathcal{A}}\mathbf{x}$ and covariance operator $Q_t = \int_0^t e^{(t-s)\mathcal{A}} Q e^{(t-s)\mathcal{A}} ds$ and $\tilde{G} := e^{-G}$ for the function G in (3). The function h evaluates the future states $\mathbf{X}_T$ using $\tilde{G}$ which propagated by (1) for a given initial $\mathbf{x}$ at time $t \in [0, T]$. It can be shown that the function h satisfies the Kolmogorov-backward equation [17] with terminal condition,

$$\partial_t h_t + \mathcal{L} h_t = 0, \quad h(T, \mathbf{x}) = \tilde{G}(\mathbf{x}). \tag{7}$$

Now, we employ the Hopf-Cole transformation [24] to establish an inherent connection between two classes of PDEs, the linear PDE in (7) and the HJB equation in (4), which provide us a key insight to deriving the Doob's h-transform in function spaces utilizing the SOC theory.

Theorem 2.2 (Hopf-Cole Transform)**.** *Let $\mathcal{V}_t = -\log h_t$. Then $\mathcal{V}_t$ satisfies the HJB equation:*

$$\partial_t \mathcal{V}_t + \mathcal{L}\mathcal{V}_t - \frac{1}{2} \left\| \sigma Q^{1/2} D_{\mathbf{x}} \mathcal{V}_t \right\|_{\mathcal{H}}^2 = 0, \quad \mathcal{V}(T, \mathbf{x}) = G(\mathbf{x}). \tag{8}$$

According to Theorem 2.2, the solution of linear PDE in equation (7) is negative exponential to the solution of the HJB equation in (8). Given that it already verified that the optimal control $\alpha^\star$ results the value function $\mathcal{V}_t$, which has explicit form as described in (8), we find that the relationship of two PDEs through $\mathcal{V}_t = -\log h_t$ leads to a distinct form of optimal control $\alpha^\star = -\sigma Q^{1/2} D_{\mathbf{x}} \mathcal{V} = \sigma Q^{1/2} D_{\mathbf{x}} \log h$, where $D_{\mathbf{x}} \log h := D_{\mathbf{x}} h / h$. Consequently, it yields another class of SDE as follows:

$$d\mathbf{X}_t^h = \left[\mathcal{A}\mathbf{X}_t^h dt + \sigma^2 Q D_{\mathbf{x}} \log h(t, \mathbf{X}_t^h) \right] dt + \sigma d\hat{\mathbf{W}}_t^Q, \quad \mathbf{X}_0^h = \mathbf{x}_0, \tag{9}$$

where $\hat{\mathbf{W}}_t^Q$ is a Q-Wiener process on $\mathbb{P}^h$. This representation is consistent with infinite-dimensional conditional SDE [2, 26] which induces an expansion of Doob's h-transform [59] in Hilbert space $\mathcal{H}$.

The Doob's h-transform in finite-dimensional spaces is well-established to construct the diffusion bridge process under the assumption that $h(t, \mathbf{x}) = \mathbb{P}(\mathbf{X}_T \in d\mathbf{x}_T | \mathbf{X}_t = \mathbf{x})$ has an explicit Radon-Nikodym density function [46, 60], enable to simulate the bridge SDEs. In contrast, although the choice of $\tilde{G}(\mathbf{x}) = \mathbf{1}_{d\mathbf{x}_T}(\mathbf{x}_T)$ in (6) yields same representation of $h(t, \mathbf{x}) = \mathbb{P}(\mathbf{X}_T \in d\mathbf{x}_T | \mathbf{X}_t = \mathbf{x})$, we cannot easily define the Radon-Nikodym density in $\mathcal{H}$ due to the absence of an equivalent form of Lesbesgue measure which hinder the computation of $D_{\mathbf{x}} \log h(t, \mathbf{X}_t^h)$ in (9) explicitly. Hence, to define the diffusion bridge process in $\mathcal{H}$, we need to identify an explicit density form of $h(t, \mathbf{x}) = \mathbb{P}(\mathbf{X}_T \in d\mathbf{x}_T | \mathbf{X}_t = \mathbf{x})$. The following theorem reveals the explicit form of h function and becomes a key ingredient in deriving infinite dimensional diffusion bridge processes.

Theorem 2.3 (Explicit Representation of h)**.** *For any $t > 0$ and any $\mathbf{x} \in \mathcal{H}$, the measure $\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x}, Q_t}$ and $\mathcal{N}_{0, Q_\infty}$ are equivalent, where $\mathcal{N}_{0, Q_\infty}$ is an invariant measure of $\mathbb{P}$ in (1) as $t \to \infty$ where $Q_\infty = -\frac{1}{2} Q \mathcal{A}^{-1}$. Moreover, for any $\mathbf{x}, \mathbf{y} \in \mathcal{H}$, the Radon-Nikodym density $\frac{d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x}, Q_t}}{d\mathcal{N}_{0, Q_\infty}}(\cdot) = q_t(\mathbf{x}, \cdot)$ is given by*

$$q_t(\mathbf{x}, \mathbf{y}) = det(1 - \Theta_t)^{-1/2} \exp \Bigg[-\frac{1}{2} \langle (1 - \Theta_t)^{-1} Q_\infty^{-1/2} e^{t\mathcal{A}} \mathbf{x}, Q_\infty^{-1/2} e^{t\mathcal{A}} \mathbf{x} \rangle_{\mathcal{H}} \tag{10}$$

$$+ \langle (1 - \Theta_t)^{-1} e^{t\mathcal{A}} Q_\infty^{-1/2} \mathbf{x}, Q_\infty^{-1/2} \mathbf{y} \rangle_{\mathcal{H}} - \frac{1}{2} \langle \Theta_t (1 - \Theta_t)^{-1} Q_\infty^{-1/2} \mathbf{y}, Q_\infty^{-1/2} \mathbf{y} \rangle_{\mathcal{H}} \Bigg], \tag{11}$$

where $\Theta_t = Q_\infty^{1/2} (Q_t^{-1/2} e^{t\mathcal{A}})^* (Q_\infty^{-1/2} Q_t^{1/2})^* (Q_\infty^{1/2} (Q_t^{-1/2} e^{t\mathcal{A}})^* (Q_\infty^{-1/2} Q_t^{1/2})^*)^*$, $t \geq 0$.

Theorem 2.3 states that the marginal distribution of certain classes of SDEs described in (1) has an explicit Radon-Nikodym density with respect to their invariant measure. Therefore, by the time-homogeneity of the process in (1), it allows us to define the h function explicitly:

$$h(t, \mathbf{x}) = \int_{\mathcal{H}} \tilde{G}(\mathbf{z}) \mathcal{N}_{e^{(T-t)\mathcal{A}}\mathbf{x}, Q_{T-t}}(d\mathbf{z}) = \int_{\mathcal{H}} \tilde{G}(\mathbf{z}) q_{T-t}(\mathbf{x}, \mathbf{z}) \mathcal{N}_{0, Q_\infty}(d\mathbf{z}). \tag{12}$$

This framework enables the construction of an infinite-dimensional bridge process, by selecting $\tilde{G}$ in (12) properly. Below, we demonstrate the infinite-dimensional diffusion bridge.

Example 2.4 (Diffusion Bridge in $\mathcal{H}$)**.** *Let $\{(\lambda^{(k)}, \phi^{(k)}) \in \mathbb{R} \times \mathcal{H} : k \in \mathbb{N}\}$ be an eigen-system of $\mathcal{H}$. Then for each $k \in \mathbb{N}$, the SDE system in equation (1) can be represented as:*

$$d\mathbf{X}_t^{(k)} = -a_k \mathbf{X}_t^{(k)} dt + \sigma \sqrt{\lambda^{(k)}} d\mathbf{W}_t^{(k)}, \quad \mathbf{X}^{(k)}(0) = \mathbf{x}_0^{(k)}, \tag{13}$$

where $\mathcal{A}\phi^{(k)} = -a_k \phi^{(k)}$, $Q\phi^{(k)} = \lambda^{(k)} \phi^{(k)}$, $\mathbf{X}_t^{(k)} = \langle \mathbf{X}_t, \phi^{(k)} \rangle_{\mathcal{H}}$ and $\mathbf{W}_t^{(k)} = \langle \mathbf{W}_t, \phi^{(k)} \rangle_{\mathcal{H}}$. Then, for any $\mathbf{x}_T \in \mathcal{H}$, the conditional law of $\mathbf{x}_T^{(k)}$ given $\mathbf{X}_t^{(k)}$ is a Gaussian $\mathcal{N}(\mathbf{m}_{T|t}^{(k)} \mathbf{X}_t^{(k)}, \mathbf{\Sigma}_{T|t}^{(k)})$ with

$$\mathbf{m}_{T|t}^{(k)} = e^{-a_k(T-t)}, \quad \mathbf{\Sigma}_{T|t}^{(k)} = \sigma^2 \frac{\lambda_k}{2a_k} \left(1 - e^{-2a_k(T-t)}\right). \tag{14}$$

Now, by setting the terminal condition in (6) as $\tilde{G}(\mathbf{x}) := \mathbf{1}_{\mathbf{x}_T}(\mathbf{x})$ (i.e., Dirac delta of $\mathbf{x}_T$) then $h(t, \mathbf{x}) = q_{T-t}(\mathbf{x}, \mathbf{x}_T)$. Thus, for each coordinate k, we get following representation:

$$d\mathbf{X}_t^{(k)} = \left[-a_k \mathbf{X}_t^{(k)} + \frac{2a_k e^{-a_k(T-t)}}{1 - e^{-2a_k(T-t)}} (\mathbf{x}_T^{(k)} - e^{-a_k(T-t)} \mathbf{X}_t^{(k)}) \right] dt + \sigma \sqrt{\lambda^{(k)}} d\mathbf{W}_t^{(k)}, \tag{15}$$

with two end points conditions $\mathbf{X}_0^{(k)} = \mathbf{x}_0^{(k)}$ and $\mathbf{X}_T^{(k)} = \mathbf{x}_T^{(k)}$.

2.4 Approximating path measures

Since the function h is intractable for a general terminal cost $\tilde{G}$ in (12), simulating the conditioned SDEs in (9) requires some approximation techniques. As observed in Theorem 2.2, finding the function h is equal to learning the control function α such that $\mathbb{P}^\alpha$ is equal to $\mathbb{P}^\star := \mathbb{P}^{\alpha^\star} = \mathbb{P}^h$. Therefore, with a parametrization $\alpha := \alpha(\cdot, \theta)$ the approximation can be done by neural network parameterization *i.e.*, $\alpha^\star \approx \alpha^{\theta^\star}$ with local minimum $\theta^\star = \arg\min_\theta D(\mathbb{P}^\alpha || \mathbb{P}^\star)$, where $D(\mathbb{P}^\alpha || \mathbb{P}^\star)$ is a divergence between $\mathbb{P}^\alpha$ and $\mathbb{P}^\star$. For example, the cost functional described in equation (3) can be represented as relative-entropy loss $D_{\text{rel}}(\mathbb{P}^\alpha || \mathbb{P}^\star) = \mathbb{E}_{\mathbb{P}^\alpha} \left[\log \frac{d\mathbb{P}^\alpha}{d\mathbb{P}^\star} \right]$[2]. Therefore, the training loss for θ can be estimated by first simulating the parameterized control path and then calculating (3) for a specified cost functional R, G. Moreover, if we can access to the $\mathbb{P}^\star$, one can define the variational optimization [69] where the loss is defined as cross-entropy loss $D_{\text{cross}}(\mathbb{P}^\alpha || \mathbb{P}^\star) = \mathbb{E}_{\mathbb{P}^\star} \left[\log \frac{d\mathbb{P}^\star}{d\mathbb{P}^\alpha} \right]$. See [20, 48] for more details about the approximation technique and other loss functions.

3 Simulating Diffusion Bridges in Infinite Dimensional Spaces

Leveraging the SOC theory within $\mathcal{H}$, we show how our approach generalizes existing diffusion-based sampling methods. Specifically, incorporating the relation between *controlled* SDEs (2) and *conditioned* SDEs (9), we introduce two learning algorithms that allow us to simulate various diffusion bridge-based sampling algorithms.

3.1 Infinite Dimensional Bridge Matching

In this section, our objective is to learn a control α that yields $\mathbb{P}^\alpha$ such that $\{\mathbf{X}_t^\alpha\}_{t \in [0,T]}$ satisfies $\mu_t^\alpha \approx \mu_t^\star$ for all pre-specified $\mu_t^\star$ over the interval $t \in [0, T]$. Specifically, we assume that the end-point marginals $\mu_0^\star$ and $\mu_T^\star$ follow the laws of two data distributions π_0 and π_T, respectively, and the intermediate marginals $\{\mu_t^\star\}_{t \in (0,T)}$ are defined as a mixture of diffusion bridge paths. This

[2] See Sec A.5 for more details

Algorithm 1 Bridge Matching of DBHS

Input: Coupling $\Pi_{0,1}$, Bridge $\mathbb{P}_{|0,T}$
for $n = 1, \cdots, N$ **do**
  Sample $(\mathbf{x}_0, \mathbf{x}_T) \sim \Pi_{0,T}$
  Simulate $\mathbf{X}^{\star}_{[0,T]} \sim \mathbb{P}_{|0,T}$ with $(\mathbf{x}_0, \mathbf{x}_T)$
  Compute $\mathcal{L}_{\text{BM}}(\theta_k)$ wtih (20)
  Update θ_{n+1} with $\nabla_{\theta_n} \mathcal{L}_{\text{BM}}(\theta_n)$
end for
Output: Approximated optimal control $\alpha^{\theta^\star}$

Algorithm 2 Bayesian Learning in $\mathcal{H}$

Input: Initial condition $\mathbf{x}_0$, energy functional $\mathcal{U}$
for $n = 1, \cdots, N$ **do**
  Simulate $\mathbf{X}^{\alpha^\theta}_{[0,T]} \sim \mathbb{P}^{\alpha^\theta}$ with $\mathbf{X}^{\alpha^\theta}_0 = \mathbf{x}_0$
  Compute $\mathcal{L}_{\text{Bayes}}(\theta_k)$ with (24)
  Update θ_{n+1} with $\nabla_{\theta_n} \mathcal{L}_{\text{Bayes}}(\theta_n)$
end for
Output: Approximated optimal control $\alpha^{\theta^\star}$

learning problem is referred to as the *Bridge Matching* (BM) algorithm [45, 50, 61] and can be expressed as a solution of the SOC problem structured as

$$\inf_{\alpha} D(\mathbb{P}^{\alpha}|\mathbb{P}^{\star}), \text{ such that } d\mathbf{X}^{\alpha}_t = \left[\mathcal{A}\mathbf{X}^{\alpha}_t + \sigma Q^{1/2}\alpha_t\right] dt + \sigma d\tilde{\mathbf{W}}^{Q}_t, \quad \mathbf{X}^{\alpha}_0 \sim \pi_0. \tag{16}$$

In (16), various divergences can be chosen for the same learning problem, as discussed in Section 2.4. Here, we will choose the cross-entropy because the relative entropy requires the appropriate selection of the terminal cost G in (3), which is intractable since we do not have access to the distributional form of π_0, π_T [44]. Furthermore, keeping the entire computational graph of $\mathbb{P}^{\alpha}$ with parameterized α can become resource-intensive, especially for higher-dimensional datasets like images [11].

Now, we specify the optimal path measure $\mathbb{P}^{\star}$ for a problem in (16). Let $\mathbb{P}_{|0,T}$ be a path measure induced by (15) and $\mu_{t|0,T}$ be a marginal distribution of $\mathbb{P}_{|0,T}$. Moreover, let $\mathbb{P}^{\star} = \mathbb{P}_{|0,T}\Pi_{0,T}$ for an independent coupling $\Pi_{0,T} = \pi_0 \otimes \pi_T$. Then the optimal path measure $\mathbb{P}^{\star}$ is defined as *Mixture of bridge*. Under regular assumptions, the optimal control $\alpha^{\star}$ that induces the optimal path measure $\mathbb{P}^{\star}$ can be constructed as a mixture of functions h in (9) by choosing $G(\mathbf{x}) = \mathbf{1}_{\mathbf{x}_T}(\mathbf{x})$.

Theorem 3.1 (Mixture of Bridges in $\mathcal{H}$). *Let us consider a marginal distribution of $\mathbb{P}^{\star}$ at $t \in [0,T]$, $\mu^{\star}_t(d\mathbf{x}_t) = \int \mu_{t|0,T}(d\mathbf{x}_t)\Pi_{0,T}(d\mathbf{x}_0, d\mathbf{x}_T)$ has density $p^{\star}_t$ with respect to some Gaussian reference measure μ_{ref} i.e., $\mu^{\star}_t(d\mathbf{x}_t)/\mu_{ref}(d\mathbf{x}_t) = p^{\star}_t(\mathbf{x}_t)$. Then the optimal path measure $\mathbb{P}^{\star}$ associated with:*

$$d\mathbf{X}^{\star}_t = \left[\mathcal{A}\mathbf{X}^{\star}_t + \mathbb{E}_{\mathbf{x}_T \sim \mathbb{P}^{\star}(d\mathbf{x}_T|\mathbf{X}^{\star}_t)}\left[\sigma^2 Q D_{\mathbf{x}} \log \mathcal{N}(\mathbf{x}_T; \mathbf{m}_{T|t}\mathbf{X}^{\star}_t, \mathbf{\Sigma}_{T|t})\right]\right] dt + \sigma d\hat{\mathbf{W}}^{Q}_t, \tag{17}$$

where $\mathbf{m}_{T|t} = e^{(T-t)\mathcal{A}}$ and $\mathbf{\Sigma}_{T|t} = \sigma^2 \int_0^{T-t} e^{(T-t-s)\mathcal{A}} Q e^{(T-t-s)\mathcal{A}} ds$ and $\mathbf{X}^{\star}_t \sim \mu_t$ for $t \in [0,T]$.

Objective Functional for Bridge Matching. With the structure of $\mathbb{P}^{\star}$ specified in (17) we can estimate the divergence between the optimal target path measure $\mathbb{P}^{\star}$ and a path measure $\mathbb{P}^{\alpha}$. To accomplish this, we first define

$$\gamma(t, \mathbf{x}; \theta) = Q^{1/2}\left[\mathbb{E}_{\mathbf{x}_T \sim \mathbb{P}^{\star}(d\mathbf{x}_T|\mathbf{x})}\left[\sigma Q^{1/2} D_{\mathbf{x}} \log \mathcal{N}(\mathbf{x}_T; \mathbf{m}_{T|t}\mathbf{X}^{\star}_t, \mathbf{\Sigma}_{T|t})\right] - \alpha(t, \mathbf{x}; \theta)\right]. \tag{18}$$

Then, by applying the Girsanov theorem[3] which provides us the Radon-Nikodym derivative between $\mathbb{P}^{\star}$ and $\mathbb{P}^{\alpha}$, we can derive the cross-entropy loss in equation (16):

$$D_{\text{cross}}(\mathbb{P}^{\alpha^\theta}|\mathbb{P}^{\star}) = \mathbb{E}_{\mathbb{P}^{\star}}\left[\log \frac{d\mathbb{P}^{\star}}{d\mathbb{P}^{\alpha^\theta}}\right] = \mathbb{E}_{\mathbb{P}^{\star}}\left[\int_0^T \frac{1}{2} \|\gamma(t, \mathbf{X}^{\star}_t; \theta)\|^2_{\mathcal{H}_0} ds\right] \tag{19}$$

Then, under the neural network parameterization of control function α^{θ}, we can reformulate the SOC problem in (16) as a learning problem with the training loss function represented by:

$$\mathcal{L}_{\text{BM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}_{0,T}} \mathbb{E}_{\mathbb{P}^{\star}(\mathbf{x}_T \sim d\mathbf{x}_T|\mathbf{X}^{\star}_t)}\left[\frac{1}{2}\left\|\sigma Q^{1/2} D_{\mathbf{x}} \log \mathcal{N}(\mathbf{x}_T; \mathbf{m}_{T|t}\mathbf{X}^{\star}_t, \mathbf{\Sigma}_{T|t}) - \alpha(t, \mathbf{X}^{\star}_t; \theta)\right\|^2_{\mathcal{H}}\right] \tag{20}$$

The $\mathcal{L}_{\text{BM}}(\theta)$ in (20) yields the infinite-dimensional BM summarized in Alg 1.

3.2 Bayesian Learning in Function Space

In the previous section, we observed that by appropriately defining a terminal cost functional G in (3), the SOC problem aligns with the sampling problem, where optimal control effectively steers

[3] See Sec A.5 for details.

the distribution from π_0 to the target distribution π_T, where we can access samples from π_0 and π_T. However, accessing samples from unknown target distribution π_T is generally not feasible. For instance, for a π_T, a posterior distribution over function. In this case, although direct samples from π_T are unattainable, its distributional representation is given as [3, 66]:

$$\frac{d\pi_T}{d\mu_{\text{prior}}}(\mathbf{X}_T) \propto \exp\left(-\mathcal{U}(\mathbf{X}_T)\right), \quad \mu_{\text{prior}} = \mathcal{N}(\mathbf{m}_{\text{prior}}, Q_{\text{prior}}), \tag{21}$$

where $\mathcal{U}$ is a energy function. Here, our primary objective is to sample from a distribution over function $\pi_T := \mu_T^\star$ by simulating the controlled diffusion process $\{\mathbf{X}_t^\alpha\}_{t\in[0,T]}$ over finite horizon $[0, T]$ with $T < \infty$. It can be represented as a solution of the following SOC problem:

$$\inf_\alpha D(\mathbb{P}^\alpha|\mathbb{P}^\star), \text{ such that } d\mathbf{X}_t^\alpha = \left[\mathcal{A}\mathbf{X}_t^\alpha + \sigma Q^{1/2}\alpha_t\right]dt + \sigma d\tilde{\mathbf{W}}_t^Q, \quad \mathbf{X}_0^\alpha = \mathbf{x}_0. \tag{22}$$

The following theorem implies that with a suitable terminal cost functional G in (3), it is possible to achieve $\mathbf{X}_T^{\alpha^\star} \sim \pi_T$ as an expansion of [54, 71] for infinite dimensional space $\mathcal{H}$.

Theorem 3.2 (Exact sampling in $\mathcal{H}$)**.** *Consider that the initial distribution μ_0 is given as the Dirac measure $\delta_{\mathbf{x}_0}$ for some $\mathbf{x}_0 \in \mathcal{H}$ and the following objective functional*

$$\mathcal{J}(\alpha) = \mathbb{E}_{\mathbb{P}^\alpha}\left[\int_0^T \frac{1}{2}\|\alpha_s\|_{\mathcal{H}}^2\, ds - \log\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T^\alpha)\right] \tag{23}$$

where $\mu_T = \mathcal{N}(e^{T\mathcal{A}}\mathbf{x}_0, Q_T)$ as a marginal distribution of $\mathbf{X}_T$ in (1) with a well-defined terminal cost $\frac{d\pi_T}{d\mu_T}$ by Theorem 2.3. Then, $\mathbf{X}_T^{\alpha^\star} \sim \pi_T$.

Objective Functional for Bayesian Learning. Unlike problem in (16) where we can access to the target path measure $\mathbb{P}^\star$ directly, it is not feasible here because $h(t, \mathbf{x}) = \mathbb{E}_{\mathbb{P}}\left[-\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T)|\mathbf{X}_t = \mathbf{x}\right]$ does not have an explicit solution. Therefore, we will use the relative-entropy loss as our training loss function:

$$\mathcal{L}_{\text{Bayes}}(\theta) = D_{\text{rel}}(\mathbb{P}^{\alpha^\theta}|\mathbb{P}^\star) = \mathbb{E}_{\mathbb{P}^{\alpha^\theta}}\left[\int_0^T \frac{1}{2}\left\|\alpha(s, \mathbf{X}_s^{\alpha^\theta}; \theta)\right\|_{\mathcal{H}}^2 ds - \log\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T^{\alpha^\theta})\right]. \tag{24}$$

The key difference from previous algorithms [73, 77] is that the Radon-Nikodym derivative $\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T^\alpha)$ may not be well-defined on $\mathcal{H}$ due to the absence of the Lesbesgue measure. However, the Theorem 2.3 suggests that by choosing certain classes of Gaussian measure *i.e.*, $\mu_{\text{prior}} := \mathcal{N}(e^{t\mathcal{A}}\mathbf{x}, Q_t)$, the Radon-Nikodym density of $\frac{d\pi_T}{d\mu_{\text{prior}}}$ and $\frac{d\mu_T}{d\mu_{\text{prior}}}$ has explicit form since μ_{prior} and $\mathcal{N}(0, Q_\infty)$ are equivalent. Thus, using the chain rule, the terminal cost for any $\mathbf{x} \in \mathcal{H}$ can be computed as follows:

$$\log\frac{d\pi_T}{d\mu_T}(\mathbf{x}) = -\mathcal{U}(\mathbf{x}) - \log\frac{d\mu_{\text{prior}}}{d\mathcal{N}(0, Q_\infty)}(\mathbf{x}) + \log\frac{d\mu_T}{d\mathcal{N}(0, Q_\infty)}(\mathbf{x}). \tag{25}$$

With $\mathcal{L}_{\text{Bayes}}(\theta)$ in (24), the infinite-dimensional bayesian learning algorithm is summarized in Alg 2.

4 Related Work

Most diffusion models operate within the framework of time-reversal [1], where the generation process is learned from its corresponding time-reversed SDEs [64]. In contrast, diffusion models based on conditioned SDEs, such as diffusion bridges, built upon the theory of Doob's h-transform, offer a conceptually simpler approach as they solely rely on a forward process. [50] proposes generative models with this concept, showing that the mixture of forward diffusion bridge processes effectively transports between couplings of two distributions. [76] introduces a family of first hitting diffusion models that generate data with a forward diffusion process at a random first hitting time based on Doob's h-transform. Combining time-reversal with the h-transform, [46] proposes a diffusion bridge process on constrained domains. Moreover, [51, 61] presented that the Schrödinger bridge problem can be solved by an iterative algorithm, which is improved by [18] to enhance efficiency. Furthermore, [44] generalizes the Schrödinger bridge matching algorithm by introducing an approximation scheme with a non-trivial running cost. Compared to prior works, which primarily focus on finite-dimensional spaces, our work extends the formulation of Doob's h-transform into Hilbert space, enabling the development of various sampling algorithms in function spaces.

5 Experiments

This section details the experimental setup and the application of the proposed Diffusion Bridges in Function Spaces **(DBFS)** for generating functional data. We interpret the data from a functional perspective, known as *field* representation [75, 79], where data are seen as a finite collection of function evaluations $\{\mathbf{Y}[\mathbf{p}_i], \mathbf{p}_i\}_i^N$. Here, a function $\mathbf{Y}$ maps points $\mathbf{p}_i$ from a coordinate space $\mathcal{X}$ to a signal space $\mathcal{Y}$, i.e., $\mathbf{Y} : \mathcal{X} \to \mathcal{Y}$. Additional experimental details are provided in Appendix A.8.

5.1 Bridge Matching

First, we present empirical results for the infinite-dimensional BM algorithm discussed in Sec 3.1, applied to 1D and 2D data. For 1D data, we consider $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$. For 2D data, we assume $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = \mathbb{R}$ for probability density or grayscale images, and $\mathcal{Y} = \mathbb{R}^3$ for RGB images.

Bridging Field. We begin by validating our bridge matching Algorithm in Alg 1 on bridging probability density function within $\mathcal{H}$. Specifically, we set $\pi_0 := \delta_{p_0}$ with a ring-shaped density function p_0 and $\pi_T := \delta_{p_T}$ characterized by a Gaussian mixture density function p_T. The functions map each grid points $\mathbf{p}_i$ to the probability in $\mathcal{Y} = \mathbb{R}$. Therefore, both density functions can be represented as their field representations $\{p_0[\mathbf{p}_i], \mathbf{p}_i\}_i^N, \{p_T[\mathbf{p}_i], \mathbf{p}_i\}_i^N$, respectively. Figure 1 illustrates the progressive propagation of the target optimal bridge process $\mathbb{P}^\star$ from p_0 to p_T. Despite the $\alpha^\star$ is trained on the functions generated from $\mathbb{P}^\star$ which are evaluated on a coarse grid $\{\mathbf{p}_i\}_i^{32^2}$, $\mathbb{P}^{\alpha^\star}$ is capable of producing accurate functional evaluations on a finer grid $\{\mathbf{p}_i\}_i^{256^2}$. This resolution-invariance property indicates that our method is adept at learning continuous functional representations, rather than merely memorizing the discrete evaluations.

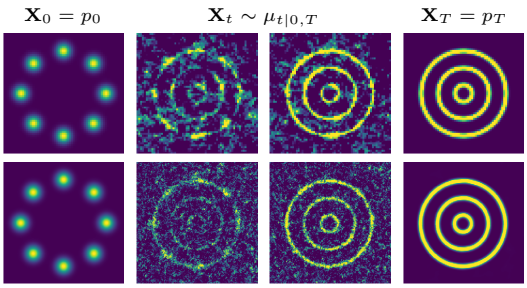
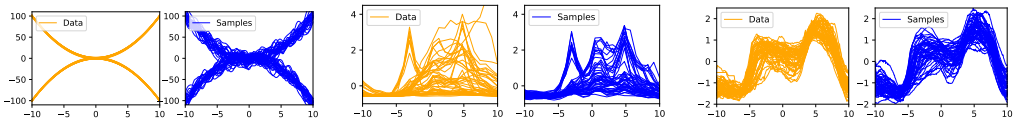


Figure 1: (Top) Diffusion Bridge $\mathbb{P}^\star$ evaluated on 32^2. (Bottom) Learned process $\mathbb{P}^{\alpha^\star}$ evaluated on 256^2



(a) Quadratic (b) Melbourne (c) Gridwatch

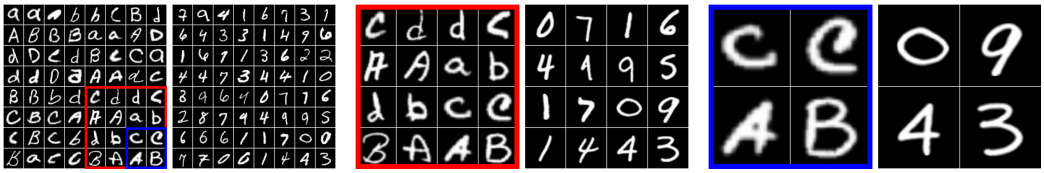
Figure 2: Results on 1D function generation. (Left) Real data and (Right) generated samples from our model.

1D function generation. We conducted an experiment on a 1D function generation task, comparing our baseline methods [22, 52] on three datasets: Quadratic, Melbourne, and Gridwatch, following the setup from [52]. For generative modeling, we set the initial distribution π_0 as $\mathcal{N}(0, Q)$ with RBF kernel for the covariance operator Q and the terminal distribution as data distribution π_T, respectively. We employing the bridge matching algorithm in Alg 1. For quantitative evaluation, we used the power of a kernel two-sample hypothesis test to distinguish between generated and ground-truth samples. Table 1 shows that our method performs comparably to baseline infinite-dimensional methods. Additionaly, The generated samples compared to the ground-truth for each dataset are provided in Figure 2.

Table 1: A Power(%) of a kernel two-sample test.

	NDP [22]	**SP-SGM** [52]	**DBFS** (Ours)
Quadratic	≥ 99.0	5.4 ± 0.7	$\mathbf{5.1 \pm 0.4}$
Melbourne	12.8 ± 0.4	$\mathbf{5.3 \pm 0.7}$	9.67 ± 0.45
Gridwatch	16.3 ± 1.8	4.7 ± 0.5	$\mathbf{3.9 \pm 0.4}$

Unpaired Image Transfer. We compare our proposed model with a finite (fixed)-dimensional baseline through an experiment on unpaired image transfer between the MNIST and EMNIST datasets at 32^2 resolution, as well as wild and cat images from the AFHQ dataset [14], downsampled to 64^2 resolution (AFHQ-64). Specifically, we evaluate the performance of [51, 61] and our DBFS model. For a fair comparison, we follow the iterative training scheme of [51] based on



(a) 32^2 (observed resolution) (b) 64^2 (unseen resolution) (c) 128^2 (unseen resolution)

(d) 64^2 (observed resolution) (e) 128^2 (unseen resolution)

Figure 3: Results on Unpaired image transfer task. **(Up)** EMNIST $\rightarrow$ MNIST **(Down)** AFHQ-64 Wild $\rightarrow$ Cat. (Left) Real data and (Right) generated samples from our model. For generation at unseen resolutions, the images within the red and blue boxed initial conditions were upsampled (using bi-linear transformation) from the observed resolution (32^2) for EMNIST and (64^2) for AFHQ-64 Wild, respectively.

the public repository[4], where two forward and backward control networks are trained alternately. For quantitative evaluation, we estimate the FID score between the generated samples and real datasets. We set $\sigma = 1$ for both [51] and our method, while FID scores for [61] are taken from [18]. Table 2 shows that our method performs comparably to the finite-dimensional method. Additionally, we provide generated samples at various unseen resolutions in Figure 3 to demonstrate the resolution-invariant property of our infinite-dimensional models. We note that our method may have slightly lower FID scores compared to finite-dimensional baselines, which may align with the observation in [79] that resolution-agnostic methods tend to have lower FID scores compared to resolution-specific ones. This could be because resolution-specific methods can incorporate domain-specific design features in their score networks. Samples generated from the reverse direction can be found in Figure A.2.

Table 2: Test FID on unpaired image transfer task. **(A)** EMNIST $\rightarrow$ MNIST, **(B)** AFHQ-64 Wild $\rightarrow$ Cat.

Method	(A)	(B)
IDBM [51]	8.2	-
DSBM† [61]	**6.0**	**25.4**
DBFS (Ours)	9.1	44.4

† result from [18].

5.2 Bayesian Learning

We validate our Bayesian learning algorithm for modeling functional data. Specifically, we will consider the temporal data as a function. We denote $\mathbf{Y}[\mathbf{O}] = \{\mathbf{Y}[\mathbf{p}_i]\}_{i=1}^{|\mathbf{O}|}$ as a collection of a function evaluation on a set of 1-dimensional observation grid $\mathbf{O} = \{\mathbf{p}_i\}_i^{|\mathbf{O}|}$ where $0 \leq \mathbf{p}_0 < \cdots < \mathbf{p}_{|\mathbf{O}|} \leq I$. We assume that each observed time series approximates a corresponding underlying continuous function $\mathbf{X} : \mathbb{R} \rightarrow \mathbb{R}^d$ as the number of observations increases *i.e.*, $\{\mathbf{Y}[\mathbf{p}_i]\}_{i=1}^{|\mathbf{O}|\rightarrow\infty} \approx \mathbf{X}$. For given observations $\mathbf{Y}[\mathbf{O}]$, our goal is to infer the posterior distribution on some set of unobserved grid $\mathbf{T} = [0, I] - \mathbf{O}$ *i.e.*, $\mathbb{P}(\mathbf{Y}[\mathbf{T}]|\mathbf{Y}[\mathbf{O}])$ and therefore modeling distribution over $\mathbf{X}$ on $[0, I]$. Please refer to Section A.8.2 for further details.

Functional Regression To verify the effectiveness of the proposed DBFS in generating functions in the 1D domain, we conducted regression experiments using synthetic data generated from the Gaussian Process (GP) by following the experimental settings in [38]. Figure 4 shows the sampled trajectories of a controlled dynamics $\mathbf{X}_t^\alpha$ for $t \in [0, \frac{T}{2}, T]$ trained on data generated from GP with RBF covariance kernel. The stochastic process begins from the deterministic function $\mathbf{X}_0^\alpha = \mathbf{x}_0$ at $t = 0$ and propagates towards the conditional posterior distribution $\mathbf{X}_T^\alpha \sim \mathbb{P}(\mathbf{Y}[\mathbf{T}]|\mathbf{Y}[\mathbf{O}])$ at $t = T$.

[4] https://github.com/stepelu/idbm-pytorch, under MIT License.

Table 4: Regression results. "context" and "target" refer to the log-likelihoods at $\mathbf{O}$ and $\mathbf{T}$, respectively.

Method	RBF		Matérn 5/2		Periodic	
	context	target	context	target	context	target
CNP	0.97 ± 0.01	0.45 ± 0.01	0.85± 0.01	0.21 ± 0.02	-0.16± 0.01	-1.75 ± 0.02
NP	0.90 ± 0.01	0.42 ± 0.01	0.77 ± 0.01	0.20 ± 0.03	-0.18 ± 0.01	-1.34 ± 0.03
DBFS	**1.02 ± 0.01**	**0.47 ± 0.01**	**0.93 ± 0.01**	**0.25 ± 0.01**	**-0.15 ± 0.01**	-1.88 ± 0.02

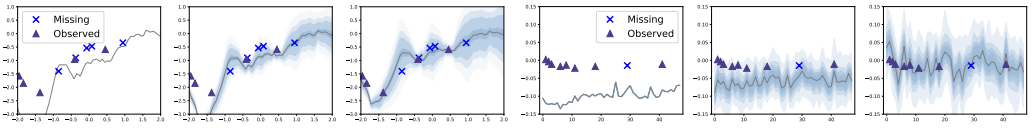


Figure 4: Sampled functions from a learned stochastic process $\mathbf{X}_t^\alpha$ evaluated on $[0, I]$ for $t \in [0, \frac{T}{2}, T]$. The grey line represents the mean function $\mathbb{E}[\mathbf{X}_t^\alpha]$ and the blue-shaded region represents the confidence interval. (Left) GP with RBF kernel. (Right) Physionet.

Imputation. We evaluate our method against recent diffusion-based imputation method where the goal is to infer the conditional distribution $p(\mathbf{Y}[\mathbf{T}]|\mathbf{Y}[\mathbf{O}])$ of unobserved grid $\mathbf{Y}[\mathbf{T}]$ give observations $\mathbf{Y}[\mathbf{O}]$. CSDI [68] utilizes DDPM [34] to learn the reverse process by treating the temporal data $\mathbf{Y}[\mathbf{O}]$ as a $\mathbb{R}^{|\mathbf{O}|\times d}$ dimensional feature. Extending this, DSDP-GP [6] enhances CSDI by incorporating noise derived from a stochastic process, instead of simple Gaussian noise. We maintained the same training setup as these models, including random seeds and the model architecture for control α^θ in (2). Consistent with their methodology, we employed the Physionet dataset [30], which comprises medical time-series data collected on an hourly rate. Since the dataset inherently contains missing values, we selected certain degrees of observed values to create an imputation test set for evaluation. We then reported the results on this test set, varying the degrees of missingness. Table 3 shows that we outperform the previous methods even though it solely relies on forward propagation of controlled SDEs in (9) without denoising procedure.

Table 3: Test imputation RMSE on Physionet.

Model	10%	50%	90%
CSDI† [68]	0.60 ± 0.27	0.66 ± 0.06	0.84 ± 0.04
DSDP-GP† [6]	0.52 ± 0.04	0.64 ± 0.05	0.81 ± 0.03
DBFS (Ours)	**0.50 ± 0.04**	**0.61 ± 0.04**	**0.77 ± 0.03**

† results from [6].

6 Conclusion and Limitation

In this work, we shed light on the application of the infinite-dimensional Doob's h-transform, exploiting SOC theory in infinite-dimensional spaces. By developing an explicit Radon-Nikodym density, we address the challenge posed by the absence of an equivalent to the Lebesgue measure. With specified cost functions for control objectives, it enables us to extend previous algorithm based on the finite-dimensional Doob's h-transform into infinite-dimensional function spaces, such as resolution-free unpaired image transfer and functional Bayesian posterior sampling.

Compared to the recent infinite-dimensional score-based diffusion model [42], our work restricts the coefficients for the stochastic dynamics to be time-independent. This limitation prevents us from defining a noise schedule for the diffusion model [78], which may hinder performance improvements. Additionally, in Bayesian learning, computing the gradient of the proposed training loss function (24) can be computationally demanding. Thus, developing a more scalable algorithm would be an interesting direction for future work. Furthermore, as our model can be applied to any functional domain, we have limited our experiments to regular 1D and 2D domains, leaving the extension to more general domains for future work.

Broader Societal Impact

Similar to other works in the literature, our proposed method holds the potential for both beneficial outcomes, such as automated data synthesis, and adverse implications, such as the deep fakes, depending on how it is used. We adhere to ethical standards for using our model in generative AI.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST), No.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics, No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2021M3E5D9025030, NRF-2022R1A5A708390812, RS-2024-00410082).

References

[1] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Elizabeth Louise Baker, Gefan Yang, Michael L Severinsen, Christy Anna Hipsley, and Stefan Sommer. Conditioning non-linear and infinite-dimensional diffusion processes. *arXiv preprint arXiv:2402.01434*, 2024.

[3] Lorenzo Baldassari, Ali Siahkoohi, Josselin Garnier, Knut Solna, and Maarten V de Hoop. Conditional score-based diffusion models for bayesian inference in infinite dimensions. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Y.I. Belopolskaya and Y.L. Dalecky. *Stochastic Equations and Differential Geometry*. Mathematics and its Applications. Springer Netherlands, 2012.

[5] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024.

[6] Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *International Conference on Machine Learning*, pages 2452–2470. PMLR, 2023.

[7] Vladimir Bogachev, Giuseppe Da Prato, and Michael Röckner. Uniqueness for solutions of fokker–planck equations on infinite dimensional spaces. *Communications in Partial Differential Equations*, 36(6):925–939, 2011.

[8] R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I: Mean Field FBSDEs, Control, and Games*. Probability Theory and Stochastic Modelling. Springer International Publishing, 2018.

[9] Pratik Chaudhari, Adam Oberman, Stanley Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5:1–30, 2018.

[10] Tianrong Chen, Jiatao Gu, Laurent Dinh, Evangelos Theodorou, Joshua M. Susskind, and Shuangfei Zhai. Generative modeling with phase stochastic bridge. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood training of schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022.

[12] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrödinger bridge. *SIAM Review*, 63(2):249–313, 2021.

[13] Raphaël Chetrite and Hugo Touchette. Nonequilibrium markov processes conditioned on large deviations. In *Annales Henri Poincaré*, volume 16, pages 2005–2057. Springer, 2015.

[14] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[15] G. Da Prato and J. Zabczyk. Differentiability of the feynman-kac semigroup and a control application. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni*, 8(3):183–188, 10 1997.

[16] G. Da Prato and J. Zabczyk. *Second Order Partial Differential Equations in Hilbert Spaces*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2002.

[17] G. Da Prato and J. Zabczyk. *Stochastic Equations in Infinite Dimensions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014.

[18] Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schr\" odinger bridge flow for unpaired data translation. *arXiv preprint arXiv:2409.09347*, 2024.

[19] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[20] Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky TQ Chen. Stochastic optimal control matching. *arXiv preprint arXiv:2312.02027*, 2023.

[21] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

[22] Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. In *International Conference on Machine Learning*, pages 8990–9012. PMLR, 2023.

[23] Giorgio Fabbri, Fausto Gozzi, and Andrzej Swiech. Stochastic optimal control in infinite dimension. *Probability and Stochastic Modelling. Springer*, 2017.

[24] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[25] Giulio Franzese, Giulio Corallo, Simone Rossi, Markus Heinonen, Maurizio Filippone, and Pietro Michiardi. Continuous-time functional diffusion processes. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Marco Fuhrman. A class of stochastic optimal control problems in hilbert spaces: Bsdes and optimal control laws, state constraints, conditioned processes. *Stochastic processes and their applications*, 108(2):263–298, 2003.

[27] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018.

[28] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes, 2018.

[29] L. Gawarecki and V. Mandrekar. *Stochastic Differential Equations in Infinite Dimensions: with Applications to Stochastic Partial Differential Equations*. Probability and Its Applications. Springer Berlin Heidelberg, 2010.

[30] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[31] Ben Goldys and Bohdan Maslowski. The ornstein–uhlenbeck bridge and applications to markov semigroups. *Stochastic processes and their applications*, 118(10):1738–1767, 2008.

[32] Paul Hagemann, Sophie Mildenberger, Lars Ruthotto, Gabriele Steidl, and Nicole Tianjiao Yang. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation. *arXiv preprint arXiv:2303.04772*, 2023.

[33] Carsten Hartmann, Omar Kebiri, Lara Neureither, and Lorenz Richter. Variational approach to rare event simulation using least-squares regression. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6), 2019.

[34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[35] Lars Holdijk, Yuanqi Du, Priyank Jaini, Ferry Hooft, Bernd Ensing, and Max Welling. Path integral stochastic optimal control for sampling transition paths. In *ICML 2022 2nd AI for Science Workshop*, 2022.

[36] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022.

[37] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[38] Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping neural processes. *Advances in neural information processing systems*, 33:6606–6615, 2020.

[39] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023.

[40] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[41] Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023.

[42] Sungbin Lim, Eunbi Yoon, Taehyun Byun, Taewon Kang, Seungwoo Kim, Kyungjae Lee, and Sungjoon Choi. Score-based generative modeling through stochastic evolution equations in hilbert spaces. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[43] Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized schrödinger bridge. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[44] Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching. In *The Twelfth International Conference on Learning Representations*, 2024.

[45] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

[46] Xingchao Liu, Lemeng Wu, Mao Ye, and qiang liu. Learning diffusion bridges on constrained domains. In *The Eleventh International Conference on Learning Representations*, 2023.

[47] S.K. Mitter. Filtering and stochastic control: a historical perspective. *IEEE Control Systems Magazine*, 16(3):67–76, 1996.

[48] Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2:1–48, 2021.

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[50] Stefano Peluchetti. Non-denoising forward-time diffusions, 2022.

[51] Stefano Peluchetti. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023.

[52] Angus Phillips, Thomas Seror, Michael Hutchinson, Valentin De Bortoli, Arnaud Doucet, and Emile Mathieu. Spectral diffusion processes. *arXiv preprint arXiv:2209.14125*, 2022.

[53] Jakiw Pidstrigach, Youssef Marzouk, Sebastian Reich, and Sven Wang. Infinite-dimensional diffusion models for function spaces. *arXiv e-prints*, pages arXiv–2302, 2023.

[54] Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied Mathematics and Optimization*, 23:313–329, 1991.

[55] Sebastian Reich. Data assimilation: the schrödinger perspective. *Acta Numerica*, 28:635–711, 2019.

[56] Lorenz Richter. *Solving high-dimensional PDEs, approximation of path space measures and importance sampling of diffusions*. PhD thesis, BTU Cottbus-Senftenberg, 2021.

[57] Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

[58] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023.

[59] L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press, 2000.

[60] S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[61] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[62] Isabel Simão. Regular transition densities for infinite dimensional diffusions. *Stochastic Analysis and Applications*, 11(3):309–336, 1993.

[63] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[64] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[65] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. FUNCTIONAL VARIATIONAL BAYESIAN NEURAL NETWORKS. In *International Conference on Learning Representations*, 2019.

[66] Taiji Suzuki. Generalization bound of globally optimal non-convex neural network training: Transportation map estimation by infinite dimensional langevin dynamics. *Advances in Neural Information Processing Systems*, 33:19224–19237, 2020.

[67] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[68] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, 2021.

[69] Evangelos A. Theodorou, George I. Boutselis, and Kaivalya Bakshi. Linearly solvable stochastic optimal control for infinite-dimensional systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4110–4116, 2018.

[70] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. Expert Certification.

[71] Belinda Tzen and Maxim Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *COLT*, 2019.

[72] Ramon Van Handel. Stochastic calculus, filtering, and stochastic control. *Course notes., URL http://www. princeton. edu/rvan/acm217/ACM217. pdf*, 14, 2007.

[73] Francisco Vargas, Andrius Ovsianas, David Fernandes, Mark Girolami, Neil D Lawrence, and Nikolas Nüsken. Bayesian learning via neural schrödinger–föllmer flows. *Statistics and Computing*, 33(1):3, 2023.

[74] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

[75] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41(2):641–676, 2022.

[76] Mao Ye, Lemeng Wu, and qiang liu. First hitting diffusion models for generating manifold, graph and categorical data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[77] Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

[78] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024.

[79] Peiye Zhuang, Samira Abnar, Jiatao Gu, Alex Schwing, Joshua M. Susskind, and Miguel Ángel Bautista. Diffusion probabilistic fields. In *The Eleventh International Conference on Learning Representations*, 2023.

A Appendix

A.1 Verification Theorem and Markov control

For the derivation, we will use the following assumption

Assumption A.1. *The function $\mathcal{V} : [0,T] \times \mathcal{H} \to \mathbb{R}$ and its derivatives $D_{\mathbf{x}}\mathcal{V}, D_{\mathbf{xx}}\mathcal{V}, \partial_t\mathcal{V}$ are uniformly continuous on bounded subsets of $[0,T] \times \mathcal{H}$ and $(0,T) \times \mathcal{H}$, respectively. Moreover, for all $(t,\mathbf{x}) \in (0,T) \times \mathcal{H}$, there exists $C_1, C_2 > 0$ such that*

$$|\mathcal{V}(t,\mathbf{x})| + |D_{\mathbf{x}}\mathcal{V}(t,\mathbf{x})| + |\partial_t\mathcal{V}(t,\mathbf{x})| + \|D_{\mathbf{xx}}\mathcal{V}(t,\mathbf{x})\| + |\mathcal{A}^\star D_{\mathbf{x}}\mathcal{V}(t,\mathbf{x})| \leq C_1(1+|\mathbf{x}|)^{C_2}, \quad \text{(A.1)}$$

where $\mathcal{A}^\star$ is adjoint operator of $\mathcal{A}$.

The HJB equation (4) can be derived by the following theorem.

Theorem A.2. *Let assumptions A.1 hold and let the function $\mathcal{V}$ with $\mathcal{V}(T,\mathbf{x}) = G(\mathbf{x})$ satisfying the dynamic programming principle for every $0 < t < t' < T$, $x \in \mathcal{H}$*

$$\mathcal{V}(t,\mathbf{x}) = \mathbb{E}_{\mathbb{P}^\alpha}\left[\int_t^{t'} [l(s,\mathbf{X}_t^\alpha) + \psi(\alpha_s)]\, ds + \mathcal{V}(t',\mathbf{X}_{t'})|\mathbf{X}_t^\alpha = \mathbf{x}\right]. \quad \text{(A.2)}$$

Then $\mathcal{V}$ is a solution of the following equation:

$$\partial_t\mathcal{V} + \mathcal{L}\mathcal{V} + \inf_{\alpha\in\mathcal{U}}\left[\langle\sigma Q_{\mathbf{x}}^{1/2}\mathcal{V},\alpha\rangle + \frac{1}{2}\|\alpha\|^2\right] = 0, \quad \mathcal{V}(T,\mathbf{x}) = G(\mathbf{x}), \quad \text{(A.3)}$$

Proof. The proof can be found in [23, Theorem 2.34]. □

A.1.1 Proof of Lemma 2.1

Proof. To begin the proof, we can formally compute the minimum of $F(D_{\mathbf{x}}\mathcal{V})$. Since [15]

$$F(\mathbf{x}) = \inf_{\alpha\in\mathcal{U}}\left[\langle\mathbf{x},\alpha\rangle + \frac{1}{2}\|\alpha\|^2\right] = -\frac{1}{2}\|\mathbf{x}\|^2. \quad \text{(A.4)}$$

Therefore, $F(\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V})$ takes the infimum at $\alpha^* = -\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}$. Next, applying Itô's formula [23, Proposition 1.165] to $\mathcal{V}$ and taking expectation on both sides, we get

$$\mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}[\mathcal{V}(T,\mathbf{X}_T^\alpha)] = \mathcal{V}(t,\mathbf{x}) + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \left(\partial_t\mathcal{V}(s,\mathbf{X}_s^\alpha) + \mathcal{L}\mathcal{V}(s,\mathbf{X}_s^\alpha) + \langle\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}(s,\mathbf{X}_s^\alpha),\alpha_s\rangle\right) ds\right], \quad \text{(A.5)}$$

where we denote $\mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}[\cdot] = \mathbb{E}_{\mathbb{P}^\alpha}[\cdot|\mathbf{X}_t = \mathbf{x}]$ By incorporating the fact that $\mathcal{V}$ satisfies the equation in (A.3), we can derive the following by adding $\mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \frac{1}{2}\|\alpha_s\|^2 ds\right]$ to both terms. The LHS of the equation (A.5) becomes:

$$\mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\underbrace{\mathcal{V}(T,\mathbf{X}_T^\alpha)}_{=G(\mathbf{X}_T^\alpha)}\right] + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \frac{1}{2}\|\alpha_s\|^2 ds\right] = \mathcal{J}(t,\mathbf{x},\alpha). \quad \text{(A.6)}$$

And for the RHS of the equation (A.5):

$$\begin{aligned}
&\mathcal{V}(t,\mathbf{x}) + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \frac{1}{2}\|\alpha_s\|^2 ds\right] + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \left(\partial_t\mathcal{V}(s,\mathbf{X}_s^\alpha) + \mathcal{L}\mathcal{V}(s,\mathbf{X}_s^\alpha) + \langle\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}(s,\mathbf{X}_s^\alpha),\alpha_s\rangle\right) ds\right] \\
&= \mathcal{V}(t,\mathbf{x}) + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \left(\partial_t\mathcal{V}(s,\mathbf{X}_s^\alpha) + \mathcal{L}\mathcal{V}(s,\mathbf{X}_s^\alpha) + \left[\langle\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}(s,\mathbf{X}_s^\alpha),\alpha_s\rangle + \frac{1}{2}\|\alpha_s\|^2\right]\right) ds\right] \\
&= \mathcal{V}(t,\mathbf{x}) + \mathbb{E}_{\mathbb{P}^\alpha}^{t,\mathbf{x}}\left[\int_t^T \left(\left[\langle\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}(s,\mathbf{X}_s^\alpha),\alpha_s\rangle + \frac{1}{2}\|\alpha_s\|^2\right] - F(\sigma Q^{1/2}D_{\mathbf{x}}\mathcal{V}(s,\mathbf{X}_s^\alpha))\right) ds\right],
\end{aligned} \quad \text{(A.7)}$$

where the last equation can be derived by adding and subtracting $\mathbb{E}^{t,\mathbf{x}}_{\mathbb{P}^\alpha}\left[\int_t^T F(D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^\alpha_s)ds\right]$ and incorporating the fact that $\mathcal{V}$ satisfies the equation in (A.3) again. Hence, we get the following equation:

$$\mathcal{J}(t,\mathbf{x},\alpha)=\mathcal{V}(t,\mathbf{x})+\mathbb{E}^{t,\mathbf{x}}_{\mathbb{P}^\alpha}\left[\int_t^T\left(\left[\langle\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^\alpha_s),\alpha_s\rangle+\frac{1}{2}\|\alpha_s\|^2\right]-F(\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^\alpha_s))\right)ds\right] \tag{A.8}$$

Since, by definition

$$\left[\langle\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^\alpha_s),\alpha_s\rangle+\frac{1}{2}\|\alpha_s\|^2\right]-F(D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^\alpha_s))\geq 0 \tag{A.9}$$

Therefore, by taking the infimum over $\alpha\in\mathcal{U}$ in the RHS of (A.8),

$$\mathcal{J}(t,\mathbf{x},\alpha)\geq\mathcal{V}(t,\mathbf{x}) \tag{A.10}$$

Moreover, since we already verified that the $F(\sigma Q^{1/2}D_\mathbf{x}\mathcal{V})$ has infimum at $\alpha^*=-\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}$. Therefore, by choosing $u=\alpha^*$, we have

$$\left[\langle\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^u_s),u_s\rangle+\frac{1}{2}\|u_s\|^2\right]-F(\sigma Q^{1/2}D_\mathbf{x}\mathcal{V}(s,\mathbf{X}^u_s))=0 \tag{A.11}$$

Thus, we get

$$\mathcal{J}(t,\mathbf{x},u)=\mathcal{V}(t,\mathbf{x}). \tag{A.12}$$

Therefore, together with (A.8), this implies that $(\alpha^*,\mathbf{X}^{\alpha^*})$ is optimal at $(t,\mathbf{x})\in[0,T]\times\mathcal{H}$ This concludes the proof. □

A.1.2 Markov Control Formulation

Now, we introduce the following corollary that states the Markov control formulation.

Corollary A.3 (Markov Control [23])**.** *Let us consider the measurable function $\phi_t:(t,T)\times\mathcal{H}\to\mathcal{U}$ which admit a mild solution $\mathbf{X}^{\phi_t}$ of the following closed-loop equation:*

$$d\mathbf{X}^{\phi_t}_s=\left[\mathcal{A}\mathbf{X}^{\phi_s}_s+\sigma Q^{1/2}\phi_s(s,\mathbf{X}^{\phi_s}_s)\right]ds+d\mathbf{W}^Q_t,\ \mathbf{X}_t=\mathbf{x} \tag{A.13}$$

Then the pair $(\alpha^{\phi_t},\mathbf{X}^{\phi_t})$, where the control α^{ϕ_t} is defined by the Markov feedback law $\alpha^{\phi_t}_s=\phi(s,\mathbf{X}^{\phi_t}_s)$ is admissible and it is optimal at $(t,\mathbf{x})$ for all $s\in[t,T]$.

Therefore, in the context of the initial value problem, such as in our case, we consider the form of the Markov control $\alpha_s:=\alpha^{\phi_0}_s=\phi(s,\mathbf{X}^{\phi_0}_s)$ for $s\in[0,T]$. The proof and details can be found in [23, Chap 2.5.1].

A.2 Proof of Theorem 2.2

Proof. Let us consider the function $\mathcal{V}(t,v)=-\log h(t,v)$.

$$\partial_t h=-h\partial_t\mathcal{V},\quad Dh=-hD_\mathbf{x}\mathcal{V},\quad D^2h=hD_\mathbf{x}\mathcal{V}\otimes D_\mathbf{x}\mathcal{V}-hD_{\mathbf{xx}}\mathcal{V}, \tag{A.14}$$

Recall that h satisfy the KBE in equation (7):

$$\partial_t h+\mathcal{L}h=0,\quad h_T=\tilde{G}. \tag{A.15}$$

Since $h=e^{-\mathcal{V}}$, hence $\partial_t h=-\mathcal{L}h=\partial_t e^{-\mathcal{V}}=-\partial_t\mathcal{V}h$. Then,

$$\partial_t\mathcal{V}h=\mathcal{L}h \tag{A.16}$$

$$=\langle D_\mathbf{x}h,\mathcal{A}\mathbf{X}_t\rangle+\frac{1}{2}\mathrm{Tr}\left[\sigma^2D_{\mathbf{xx}}hQ\right] \tag{A.17}$$

$$=-\langle hD_\mathbf{x}\mathcal{V},\mathcal{A}\mathbf{X}_t\rangle+\frac{1}{2}\mathrm{Tr}\left[\sigma^2\left(hD_\mathbf{x}\mathcal{V}\otimes D_\mathbf{x}\mathcal{V}-hD_{\mathbf{xx}}\mathcal{V}\right)Q\right] \tag{A.18}$$

$$=-\langle hD_\mathbf{x}\mathcal{V},\mathcal{A}\mathbf{X}_t\rangle+\frac{1}{2}\mathrm{Tr}\left[\sigma^2\left(hD_\mathbf{x}\mathcal{V}\otimes D_\mathbf{x}\mathcal{V}\right)Q\right]-\frac{1}{2}\mathrm{Tr}\left[\sigma^2hD_{\mathbf{xx}}\mathcal{V}Q\right]. \tag{A.19}$$

We can simplify the last equation as

$$\partial_t \mathcal{V} = -\langle D_{\mathbf{x}}\mathcal{V}, \mathcal{A}\mathbf{X}_t\rangle + \frac{1}{2}\mathrm{Tr}\left[\sigma^2 \left(D_{\mathbf{x}}\mathcal{V} \otimes D_{\mathbf{x}}\mathcal{V}\right) Q\right] - \frac{1}{2}\mathrm{Tr}\left[\sigma^2 D_{\mathbf{xx}}\mathcal{V} Q\right] \tag{A.20}$$

$$= -\mathcal{L}\mathcal{V} + \frac{1}{2}\mathrm{Tr}\left[\sigma^2 \left(D_{\mathbf{x}}\mathcal{V} \otimes D_{\mathbf{x}}\mathcal{V}\right) Q\right]. \tag{A.21}$$

Following [69], the second term of RHS can be derived as follows

$$\frac{1}{2}\mathrm{Tr}\left[\sigma^2 \left(D_{\mathbf{x}}\mathcal{V} \otimes D_{\mathbf{x}}\mathcal{V}\right) Q\right] = \frac{1}{2}\sum_{k\in\mathbb{N}} \langle \sigma^2 (D_{\mathbf{x}}\mathcal{V} \otimes D_{\mathbf{x}}\mathcal{V}) Q\phi^{(k)}, \phi^{(k)}\rangle \tag{A.22}$$

$$= \frac{1}{2}\sum_{k\in\mathbb{N}} \langle \sigma^2 D_{\mathbf{x}}\mathcal{V} \langle D_{\mathbf{x}}\mathcal{V}, Q\phi^{(k)}\rangle, \phi^{(k)}\rangle \tag{A.23}$$

$$= \frac{1}{2}\sum_{k\in\mathbb{N}} \langle \sigma^2 D_{\mathbf{x}}\mathcal{V}, Q\phi^{(k)}\rangle \langle D_{\mathbf{x}}\mathcal{V}, \phi^{(k)}\rangle \tag{A.24}$$

$$= \frac{1}{2}\sum_{k\in\mathbb{N}} \langle \sigma^2 Q D_{\mathbf{x}}\mathcal{V}, \phi^{(k)}\rangle \langle D_{\mathbf{x}}\mathcal{V}, \phi^{(k)}\rangle \tag{A.25}$$

$$= \frac{1}{2}\langle \sigma^2 D_{\mathbf{x}}\mathcal{V}, Q D_{\mathbf{x}}\mathcal{V}\rangle \tag{A.26}$$

$$= \frac{1}{2}\left\|\sigma Q^{1/2} D_{\mathbf{x}}\mathcal{V}\right\|_{\mathcal{H}}^2 \tag{A.27}$$

Therefore, combining the above results, we have

$$\partial_t \mathcal{V} + \mathcal{L}\mathcal{V} - \frac{1}{2}\left\|\sigma Q^{1/2} D_{\mathbf{x}}\mathcal{V}\right\|_{\mathcal{H}}^2, \quad \mathcal{V}_T = G. \tag{A.28}$$

Since (A.28) coincides with (4) with $\psi(\cdot) := \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$, this concludes the proof. □

A.3 Proof of Theorem 2.3

Proof. A proof of Theorem 2.3 is based on [16, Chap. 10.3]. Since $Q_\infty = -\frac{1}{2}Q\mathcal{A}^{-1}$ is a trace class, we define a trace class operator Θ_t as follows:

$$\Theta_t = Q_\infty^{1/2}(Q_t^{-1/2}e^{t\mathcal{A}})^*(Q_\infty^{-1/2}Q_t^{1/2})^*(Q_\infty^{1/2}(Q_t^{-1/2}e^{t\mathcal{A}})^*(Q_\infty^{-1/2}Q_t^{1/2})^*)^*, \tag{A.29}$$

for all $t \geq 0$. Since $Q_t = Q_\infty - e^{t\mathcal{A}}Q_\infty e^{t\mathcal{A}^*}$, we can rewrite Q_t in terms of Θ_t,

$$Q_t = Q_\infty - e^{t\mathcal{A}}Q_\infty e^{t\mathcal{A}^*} \tag{A.30}$$

$$= Q_\infty^{1/2}\left[1 - (Q_\infty^{-1/2}e^{t\mathcal{A}})Q_\infty(Q_\infty^{-1/2}e^{t\mathcal{A}})^*\right]Q_\infty^{1/2} \tag{A.31}$$

$$= Q_\infty^{1/2}(1-\Theta_t)Q_\infty^{1/2}. \tag{A.32}$$

Thus, we have $(1-\Theta_t)\mathbf{x} = Q_\infty^{-1/2}Q_tQ_\infty^{-1/2}\mathbf{x}$ for all $\mathbf{x} \in \mathcal{H}_0$. It implies that $\langle(1-\Theta_t)\mathbf{x}, \mathbf{x}\rangle_{\mathcal{H}_0} \geq 0$, the non-negativity of $(1-\Theta_t)$. Moreover it also implies that $(1-\Theta_t)^{-1}$ is invertible:

$$(1-\Theta_t)^{-1} = (Q_t^{-1/2}Q_\infty^{1/2})^*Q_t^{-1/2}Q_\infty^{1/2}. \tag{A.33}$$

Consequently, it yields the following formula [16, Proposition. 1.3.11]

$$q_t(0,\mathbf{y}) = det(1-\Theta)^{-1/2}\exp\left[-\frac{1}{2}\langle\Theta_t(1-\Theta_t)^{-1}Q_\infty^{-1/2}\mathbf{y}, Q_\infty^{-1/2}\mathbf{y}\rangle_{\mathcal{H}}\right]. \tag{A.34}$$

Now, for the general case, by using the chain rule, we have:

$$q_t(\mathbf{x},\mathbf{y}) = \frac{d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x},Q_t}}{d\mathcal{N}_{0,Q_t}}\frac{d\mathcal{N}_{0,Q_t}}{d\mathcal{N}_{0,Q_\infty}}(\mathbf{y}) = \frac{d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x},Q_t}}{d\mathcal{N}_{0,Q_t}}(\mathbf{y})q_t(0,\mathbf{y}), \tag{A.35}$$

and utilizing Cameron-Martin theorem [16, Theorem. 1.3.6], we get:

$$\frac{d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x},Q_t}}{d\mathcal{N}_{0,Q_t}}(\mathbf{y}) = \exp\left[\langle Q_t^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_t^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} - \frac{1}{2}\left\|Q_t^{-1/2}e^{t\mathcal{A}}\mathbf{x}\right\|_{\mathcal{H}}^2\right] \tag{A.36}$$

$$= \exp\left[\langle Q_t^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_t^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} - \frac{1}{2}\langle Q_t^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_t^{-1/2}e^{t\mathcal{A}}\mathbf{x}\rangle_{\mathcal{H}}\right] \tag{A.37}$$

$$= \exp\left[\langle Q_\infty^{1/2}Q_t^{-1}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} - \frac{1}{2}\langle Q_\infty^{1/2}Q_t^{-1}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}\rangle_{\mathcal{H}}\right] \tag{A.38}$$

$$\overset{(i)}{=} \exp\left[\langle (1-\Theta_t)^{-1}Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} - \frac{1}{2}\langle (1-\Theta_t)^{-1}Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}\rangle_{\mathcal{H}}\right] \tag{A.39}$$

where (i) follows from (A.33), $(1-\Theta_t)^{-1}Q_\infty^{-1/2} = (Q_t^{-1/2}Q_\infty^{1/2})^* Q_t^{-1/2} = Q_\infty^{1/2}Q_t^{-1}$. Thus, by substituting (A.34) and (A.39) into (A.35), we obtain the following result:

$$q_t(\mathbf{x},\mathbf{y}) = det(1-\Theta)^{-1/2}\exp\Bigg[-\frac{1}{2}\langle \Theta_t(1-\Theta_t)^{-1}Q_\infty^{-1/2}\mathbf{y}, Q_\infty^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} \tag{A.40}$$

$$+ \langle (1-\Theta_t)^{-1}Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}\mathbf{y}\rangle_{\mathcal{H}} - \frac{1}{2}\langle (1-\Theta_t)^{-1}Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}, Q_\infty^{-1/2}e^{t\mathcal{A}}\mathbf{x}\rangle_{\mathcal{H}}\Bigg]. \tag{A.41}$$

It concludes the proof. □

A.4 Derivation of Example 2.4

For a diffusion bridge process, let us define the h function as:

$$h(t,T,\mathbf{x}_t,\mathbf{x}_T) = \mathbb{E}_{\mathbb{P}}\left[\tilde{G}(\mathbf{X}_T,\mathbf{x}_T)|\mathbf{X}_t = \mathbf{x}_t\right] = \int \tilde{G}(\mathbf{z},\mathbf{x}_T)d\mathcal{N}_{e^{(T-t)\mathcal{A}}\mathbf{x}_t,Q_{T-t}}(\mathbf{z}) \tag{A.42}$$

If we choose $\tilde{G}(\mathbf{x},\mathbf{y}) = \mathbf{1}_{d\mathbf{y}}(\mathbf{x})$. Then, for any $\mathbf{y}\in\mathcal{H}$ and $t\in[0,T]$, Theorem 2.3 implies that

$$h(0,t,\mathbf{x}_0,\mathbf{y}) = \int \tilde{G}(\mathbf{z},\mathbf{y})d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x}_0,Q_t}(\mathbf{z}) \tag{A.43}$$

$$= \int \tilde{G}(\mathbf{z},\mathbf{y})\frac{d\mathcal{N}_{e^{t\mathcal{A}}\mathbf{x}_0,Q_t}}{d\mathcal{N}_{0,Q_\infty}}(\mathbf{z})d\mathcal{N}_{0,Q_\infty}(\mathbf{z}) \tag{A.44}$$

$$= \int \tilde{G}(\mathbf{z},\mathbf{y})q_t(\mathbf{x},\mathbf{z})d\mathcal{N}_{0,Q_\infty}(\mathbf{z}) \tag{A.45}$$

$$= q_t(\mathbf{x},\mathbf{y}). \tag{A.46}$$

Moreover, with an eigen-system of $\mathcal{H}$, $\{(\lambda^{(k)},\phi^{(k)})\in\mathbb{R}\times\mathcal{H} : k\in\mathbb{N}\}$, q_t can be represented as [62]:

$$q_t(\mathbf{x},\mathbf{y}) = \prod_{k\in\mathbb{N}} q_t^{(k)}(\mathbf{x}^{(k)},\mathbf{y}^{(k)}), \tag{A.47}$$

where for each coordinated k, $q_t^{(k)}(\mathbf{x}^{(k)},\mathbf{y}^{(k)})$ has following representation:

$$q_t^{(k)}(\mathbf{x}^{(k)},\mathbf{y}^{(k)}) = \left(\frac{\lambda^{(k)}}{2a_k}(1-e^{-2a_kt})\right)^{-1/2}\exp\left[-\frac{(\mathbf{y}^{(k)}-e^{-a_kt}\mathbf{x}^{k)})^2}{2\lambda^{(k)}(1-e^{-2a_kt})} + \frac{(\mathbf{y}^{(k)})^2}{2\lambda^{(k)}}\right] \tag{A.48}$$

$$\mathcal{A}\phi^{(k)} = -a_k\phi^{(k)}, \quad Q\phi^{(k)} = \lambda^{(k)}\phi^{(k)}, \quad \mathbf{x}^{(k)} = \langle \mathbf{x},\phi^{(k)}\rangle_{\mathcal{H}}, \quad \mathbf{y}_k = \langle \mathbf{y},\phi^{(k)}\rangle_{\mathcal{H}}. \tag{A.49}$$

Therefore, since $D_{\mathbf{x}}\log h(t,T,\mathbf{x},\mathbf{x}_T) = D_{\mathbf{x}}\log q_{T-t}(\mathbf{x}_t,\mathbf{x}_T)$, by projecting $D_{\mathbf{x}}\log q_{T-t}(\mathbf{x}_t,\mathbf{x}_T)$ to each coordinate $\phi^{(k)}$, we obtain the following results:

$$\frac{d}{d\mathbf{x}^{(k)}}\log q_{T-t}^{(k)}(\mathbf{x}_t^{(k)},\mathbf{x}_T^{(k)}) = \frac{2a_ke^{-a_k(T-t)}}{\lambda^{(k)}(1-e^{-2a_k(T-t)})}(\mathbf{x}_T^{(k)} - e^{-a_k(T-t)}\mathbf{x}^{(k)}) \tag{A.50}$$

A.5 Deriving Divergence Between Path Measures

Here we present an infinite-dimensional generalization of Girsanov's theorem [17, Theorem 10.14], which plays a crucial role in estimating the divergence between two path measures discussed in Sec 2.4. The theorem is formulated as follows:

Theorem A.4 (Girsanov's Theorem in $\mathcal{H}$)**.** *Let γ be a $\mathcal{H}_0$-valued $\mathcal{F}_t$-predictable process such that*

$$\mathbb{P}\left(\int_0^T \|\gamma_s\|_{\mathcal{H}_0}^2 \, ds < \infty\right) = 1, \quad \mathbb{E}\left[\exp\left(\int_0^t \langle \gamma_s, d\mathbf{W}_t^Q \rangle_{\mathcal{H}_0} - \frac{1}{2}\int_0^T \|\gamma_s\|_{\mathcal{H}_0}^2 \, dt\right)\right] = 1. \tag{A.51}$$

Then the process $\tilde{\mathbf{W}}_t^Q = \mathbf{W}_t^Q - \int_0^T \gamma_s ds$ is a Q-Wiener process with respect to $\{\mathcal{F}_t\}_{t\geq 0}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{Q})$ where

$$d\mathbb{Q} = \exp\left(\int_0^t \langle \gamma_s, d\mathbf{W}_t^Q \rangle_{\mathcal{H}_0} - \frac{1}{2}\int_0^T \|\gamma_s\|_{\mathcal{H}_0}^2 \, dt\right) d\mathbb{P}. \tag{A.52}$$

Or we can derive alternative formulation, by substituting $\mathbf{W}_t^Q = \tilde{\mathbf{W}}_t^Q + \int_0^T \gamma_s ds$ to (A.52),

$$d\mathbb{Q} = \exp\left(\int_0^t \langle \gamma_s, d\tilde{\mathbf{W}}_t^Q \rangle_{\mathcal{H}_0} + \frac{1}{2}\int_0^T \|\gamma_s\|_{\mathcal{H}_0}^2 \, dt\right) d\mathbb{P}. \tag{A.53}$$

Now, we will apply Girsanov's theorem to path measures related to (1) and (9). Let $\mathbb{Q} := \mathbb{P}^\alpha$ in (A.53). Then $\gamma_s := Q^{1/2}\alpha_s$ and we get

$$\frac{d\mathbb{P}^\alpha}{d\mathbb{P}} = \exp\left(\int_0^t \langle \alpha, d\tilde{\mathbf{W}}_t^Q \rangle_{\mathcal{H}_0} + \frac{1}{2}\int_0^T \left\|Q^{1/2}\alpha_s\right\|_{\mathcal{H}_0}^2 \, dt\right). \tag{A.54}$$

Proof. Proof can be founded in [17, Theorem 10.14] □

Since our goal is find an optimal control $\alpha^\star$ such that $\mathbf{X}_T^{\alpha^\star}$ satisfying terminal constraints which is represented by function $\tilde{G}$ in (6), we may define our target path measure $\mathbb{P}^\star$ as $\frac{d\mathbb{P}^\star}{d\mathbb{P}} = \frac{1}{\mathcal{Z}}\tilde{G}(\cdot)$ [57], where $\mathcal{Z} = \mathbb{E}_{\mathbb{P}}\left[\tilde{G}(\mathbf{X}_T)\right]$ Then, we can compute the logarithm of Radon Nikodym derivative as

$$\log \frac{d\mathbb{P}^\alpha}{d\mathbb{P}^\star} = \log \frac{d\mathbb{P}^\alpha}{d\mathbb{P}} + \log \frac{d\mathbb{P}}{d\mathbb{P}^\star} - \mathcal{Z} \tag{A.55}$$

$$\approx \int_0^t \langle \alpha, d\tilde{\mathbf{W}}_t^Q \rangle_{\mathcal{H}_0} + \frac{1}{2}\int_0^T \left\|Q^{1/2}\alpha_s\right\|_{\mathcal{H}_0}^2 \, dt + G(\cdot). \tag{A.56}$$

Since $\tilde{\mathbf{W}}_t^Q$ is Q-Wiener process on $\mathbb{P}^\alpha$, we can compute the relative entropy loss in Sec 2.4:

$$D_{\text{ref}}(\mathbb{P}^\alpha|\mathbb{P}^\star) = \mathbb{E}_{\mathbb{P}^\alpha}\left[\log \frac{d\mathbb{P}^\alpha}{d\mathbb{P}^\star}\right] = \mathbb{E}_{\mathbb{P}^\alpha}\left[\frac{1}{2}\int_0^T \|\alpha_s\|_{\mathcal{H}}^2 \, dt + G(\mathbf{X}_T^\alpha)\right], \tag{A.57}$$

where we denote $\|(\cdot)\|_{\mathcal{H}_0}^2 = \left\|Q^{-1/2}(\cdot)\right\|_{\mathcal{H}}^2$ This representation matches with (3) when $R(\cdot) := \frac{1}{2}\|\cdot\|_{\mathcal{H}}^2$. Similarly, the cross entropy loss in (19) can be derived by set $\mathbb{Q} := \mathbb{P}^\star$ and $\mathbb{P} := \mathbb{P}^\alpha$ in (A.53). Then γ_s can be defined as (18) and $\hat{\mathbf{W}}_t^Q = \tilde{\mathbf{W}}_t^Q + \int_0^T \gamma_s(\theta) ds$ is a Q-Wiener process on $\mathbb{P}^\star$ where $\mathbb{P}^\alpha$ satisfies the Radon-Nikodym derivative:

$$\frac{d\mathbb{P}^\star}{d\mathbb{P}^\alpha} = \exp\left(\int_0^t \langle \gamma_s(\theta), d\hat{\mathbf{W}}_t^Q \rangle_{\mathcal{H}_0} + \frac{1}{2}\int_0^T \|\gamma_s(\theta)\|_{\mathcal{H}_0}^2 \, dt\right). \tag{A.58}$$

Hence, the cross-entropy loss can be computed as in (19).

A.6 Proof of Theorem 3.1

Proof. Let us consider that the marginal distributions $\{\mu_t^\star\}_{t\in 0,T}$ satisfying the following relation in a weak sense [4]:

$$\partial_t \int f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t) = \int f(\mathbf{x}_t)\mathcal{L}_t^\star\mu_t^\star(d\mathbf{x}_t) = \int \mathcal{L}_t f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t) \tag{A.59}$$

where $\mathcal{L}_t^\star$ is adjoint operator of $\mathcal{L}_t$. Now, let us denote $\mu_{t|0,T}(\mathbf{x}_t) = \mu_t^\star(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)$ and consider factorizable marginal distribution $\mu_t^\star(d\mathbf{x}_t) = \int_\Pi \mu_t^\star(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)\Pi(d\mathbf{x}_0,d\mathbf{x}_T)$, where $\mu_t^\star(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)$ satisfying the following relation:

$$\begin{aligned}
&\partial_t \int_{\mathcal{H}} f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T) = \int_{\mathcal{H}} \mathcal{L}_{t|0,T} f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)\\
&= \int_{\mathcal{H}} \left[\langle \mathcal{A}\mathbf{x}_t, D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \langle h_{t|0,T}(\mathbf{x}_t), D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \frac{1}{2}\mathrm{Tr}\left[\sigma^2 Q D_{\mathbf{xx}} f(\mathbf{x}_t)\right]\right]\mu_t^\star(d\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T),
\end{aligned}$$

where we denote $h_{t|0,T}(\mathbf{x}_t) := \sigma^2 Q D_{\mathbf{x}} \log h(t,\mathbf{x_t})$. Now, we can obtain the Kolmogorov operator associated with the diffusion process associated with the marginal distributions $\mu_t^\star(d\mathbf{x}_t)$ as follows

$$\partial_t \int_{\mathcal{H}} f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t) = \partial_t \int_\Pi \int_{\mathcal{H}} f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)\Pi(d\mathbf{x}_0,d\mathbf{x}_T) \tag{A.60}$$

$$= \int_{\mathcal{H}} \left[\langle \mathcal{A}\mathbf{x}_t, D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \langle h_t^\star(\mathbf{x}_t), D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \frac{1}{2}\mathrm{Tr}\left[\sigma^2 Q D_{\mathbf{xx}} f(\mathbf{x}_t)\right]\right]\mu_t^\star(d\mathbf{x}_t) \tag{A.61}$$

$$= \int_{\mathcal{H}} \mathcal{L}_t f(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t) = \int f(\mathbf{x}_t)\mathcal{L}_t^\star\mu_t^\star(d\mathbf{x}_t), \tag{A.62}$$

where we have defined $\int_{\mathcal{H}} h_t^\star(\mathbf{x}_t)\mu_t^\star(d\mathbf{x}_t) := \int_\Pi \int_{\mathcal{H}} h_{t|0,T}\mu_t^\star(d\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)\Pi(d\mathbf{x}_0,d\mathbf{x}_T)$. It implies that the diffusion dynamics $d\mathbf{X}_t^\star$ associated with the Kolmogorov operator $\mathcal{L} := \langle \mathcal{A}\mathbf{x}_t, D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \langle h_t^\star(\mathbf{x}_t), D_{\mathbf{x}} f(\mathbf{x}_t)\rangle_{\mathcal{H}} + \frac{1}{2}\mathrm{Tr}\left[\sigma^2 Q D_{\mathbf{xx}} f(\mathbf{x}_t)\right]$ also associated with Fokker-Planck equation [7] $\mathcal{L}_t^*\mu_t^\star(d\mathbf{x}_t) = 0$, meaning $\mathbf{X}_t^\star \sim \mu_t^\star$ for all $t \in [0,T]$. Now, for some reference measure μ_{ref} where the Radon-Nikodym derivatives $\frac{d\mu_t^\star}{d\mu_{\text{ref}}}(\mathbf{x}_t) = p_t(\mathbf{x}_t)$, $\frac{d\mu_{t|0,T}}{d\mu_{\text{ref}}}(\mathbf{x}_t) = p_{t|0,T}(\mathbf{x}_t)$ exist. Then, under μ_{ref}, $h_t^\star$ can be defined as follows:

$$h_t^\star(\mathbf{x}_t) = \frac{\int_\Pi h_{t|0,T}(\mathbf{x}_t)p_{t|0,T}(\mathbf{x}_t)\Pi(d\mathbf{x}_0,d\mathbf{x}_T)}{p_t(\mathbf{x}_t)}. \tag{A.63}$$

Therefore, the diffusion process associated with marginal distributions $\mu_t^\star(\mathbf{x}_t)$ has following representation:

$$d\mathbf{X}_t^\star = \left[\mathcal{A}\mathbf{X}_t^\star + \sigma^2 Q\mathbb{E}_{\mathbf{x}_T\sim\mathbb{P}^h(d\mathbf{x}_T|\mathbf{X}_t^h)}\left[q_{T-t}(\mathbf{X}_t^\star,\mathbf{x}_T)\right]\right]dt + \sigma d\mathbf{W}_t^Q, \quad \mathbf{X}_0^\star \sim \mu_0^\star. \tag{A.64}$$

This concludes the proof. □

A.7 Proof of Theorem 3.2

Proof. Let us assume the initial condition is fixed to deterministic point $\mathbf{x}_0 \in \mathcal{H}$ and define a reference measure $\mu_{\text{ref}} = \mathcal{N}(0, Q_\infty)$. Since our goal is $\mathbf{X}_T^\alpha$ satisfying the terminal condition G, define $\mathbb{P}^\star$ as $\frac{d\mathbb{P}^\star}{d\mathbb{P}} = \frac{1}{\mathcal{Z}}\tilde{G}(\cdot)$, where $\mathcal{Z} = \mathbb{E}_{\mathbb{P}}[\tilde{G}(\mathbf{X}_T)]$. Therefore, we have the following relation for marginal distributions $d\mu_T^\star = \frac{1}{Z}\tilde{G}(\mathbf{X}_T)d\mu_T$. Moreover, since we have defined $G(\mathbf{X}_T) = -\log\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T)$ in (23), then it result $h(t,\mathbf{x}) = \mathbb{E}_{\mathbb{P}}\left[\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T)|\mathbf{X}_t = \mathbf{x}\right]$ by following Theorem 2.2, we get $\tilde{G}(\mathbf{X}_T) = h(T,\mathbf{X}_T)$. Hence, we have for any Borel set $B \in \mathcal{B}(\mathcal{H})$,

$$\int_B d\mu_T^\star(v) = \int_B \frac{h(T,\mathbf{X}_T)}{h(0,\mathbf{x}_0)}d\mu_T \tag{A.65}$$

$$= \int_B \frac{\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T)}{\int_{\mathcal{H}}\frac{d\pi_T}{d\mu_T}(\mathbf{X}_T)d\mu_T}d\mu_T = \int_B d\pi_T \tag{A.66}$$

Now, given that we have confirmed that the conditioned SDE in (9) correspond to the controlled process (9) with optimal control, we can establish the result $\mathbb{P}(\mathbf{X}_T^{\alpha^\star} \in B) = \pi(B)$. This concludes the proof. □

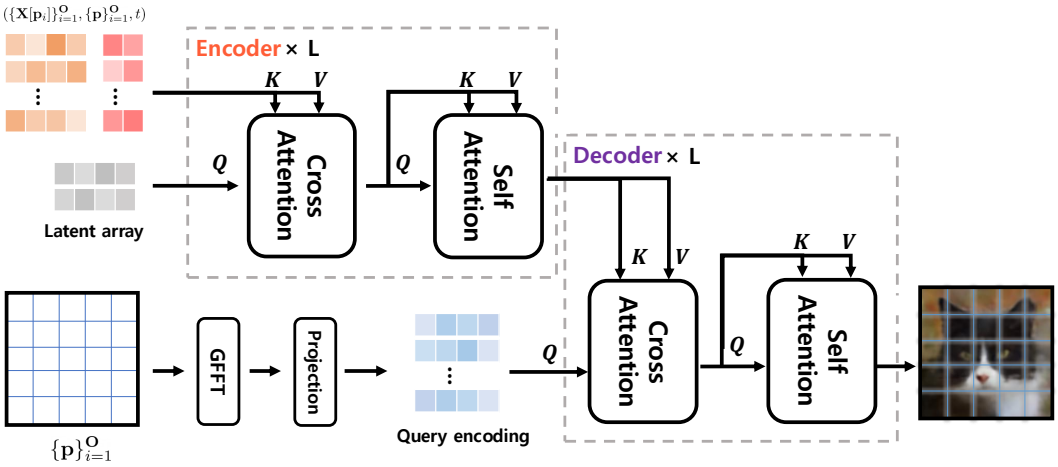


Figure A.1: Transformer-based network architecture.

A.8 Experimental Details

A.8.1 Experiment on 2D-Domain

Synthetic Experiment. In a synthetic experiment, we computed the log-probability of p_0 and p_T across a uniformly sampled grid of 64^2 points, with each point $\mathbf{p}_i$ ranging within $[-7, 7]^2$. For p_0, the log-probability was generated using an 8-Gaussian mixture model as specified in [70]. For p_T, we employed the following log-density function:

$$\log p_T(\mathbf{p}) = -\frac{\min(\|\mathbf{p}-1\|^2, \|\mathbf{p}-3\|^2, \|\mathbf{p}-5\|^2)}{0.05}. \tag{A.67}$$

Training was conducted using the Adam optimizer with a learning rate of $1e-3$. The network was trained with a batch size of 24 for a total of 1000 iterations. We set $\sigma = 0.2$ in (1) for this experiment and set 100 discretization steps. We use a single A6000 GPU for this experiment.

The control function was parameterized using a 4-layer FNO-2D [40], with the cutoff number of Fourier modes set at 8 and each convolution layer having a width of 32.

Simulation of DBFS. We follow the simulation scheme introduced in [42, 58]. For $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, we use the discrete cosine transformation (DCT) for projection. Specifically, the eigenvector $\phi^{(k)}(\mathbf{x})$ and eigenvalue $\lambda^{(k)}$ of the negative Laplacian operator $-\Delta$, which is positive definite and Hermitian, and satisfies the zero Neumann boundary condition, are given by:

$$-\Delta\phi^{(k)}(\mathbf{x}) = \lambda^{(k)}\phi^{(k)}(\mathbf{x}) \tag{A.68}$$

$$\frac{\partial\phi^{(k)}(\mathbf{x})}{\partial x_1} = \frac{\partial\phi^{(k)}(\mathbf{x})}{\partial x_2} = 0, . \tag{A.69}$$

It implies that the orthonormality of $\phi^{(k)}(\cdot)$ with respect to the associated inner product, thereby enables computations in (A.47-A.50). Now, considering a rectangular domain with Cartesian coordinates, where pixels in the image are sampled from an underlying regular domain, the eigenbasis is given as a separable cosine basis:

$$\phi^{(n,m)}(x_1, x_2) \sim \cos\left(\frac{\pi n x_1}{W}\right)\cos\left(\frac{\pi m x_2}{H}\right) \tag{A.70}$$

$$\lambda^{(n,m)} = \pi^2\left(\frac{n^2}{W^2} + \frac{m^2}{H^2}\right). \tag{A.71}$$

Then, the negative Laplacian $-\Delta$ can then by represented by an eigen decomposition $-\Delta = \mathbf{E}\mathbf{D}\mathbf{E}^T$, where $\mathbf{E}^T$ is the projection matrix for the DCT *i.e.*, $\tilde{\mathbf{X}}_t = \mathbf{E}^T\mathbf{X}_t = \texttt{DCT}(\mathbf{X}_t)$, and $\mathbf{D}$ is a digonal

Algorithm 3 DBFS sampling for Bridge Matching

Input: Linear operator $\mathcal{A}$, trace-class operator Q, learned control α, initial distribution π_0, discrete and inverse discrete cosine transforms $\texttt{DCT}$, $\texttt{iDCT}$, target resolution grid $\mathbf{T}^2$, trained resolution grid $\mathbf{O}^2$.

Sample the initial condition $\mathbf{X}_0^\alpha = \mathbf{x}_0 \sim \pi_0$
if $\mathbf{T}^2 \neq \mathbf{O}^2$ **then**
  Upsample $\mathbf{X}_0^\alpha = \{\mathbf{X}_0^\alpha[\mathbf{p}_i]\}_{i=1}^{\mathbf{O}^2}$ to $\{\mathbf{X}_0^\alpha[\mathbf{p}_i]\}_{i=1}^{\mathbf{T}^2}$
end if
for $t = 0, \cdots, T$ **do**
  Estimate the control $\alpha_t^\star = \alpha(t, \mathbf{X}_t^\alpha; \theta^\star)$
  Sample Gaussian noise $\xi \sim \mathcal{N}(0, \mathbf{I})$
  Discrete cosine transform the initial condition, estimated control, Gaussian noise
  $\{\tilde{\mathbf{X}}_t^{(k)}\}_{k=1}^{\mathbf{T}^2} = \texttt{DCT}(\mathbf{X}_t^\alpha), \quad \{\tilde{\alpha}_t^{(k)}\}_{k=1}^{\mathbf{T}^2} = \texttt{DCT}(\alpha_t^\star), \quad \{\tilde{\xi}^{(k)}\}_{k=1}^{\mathbf{T}^2} = \texttt{DCT}(\xi)$
  for $k = 1, \cdots, ^{\mathbf{T}^2}$ **do in parallel**
    $\tilde{\mathbf{X}}_{t+\Delta_t}^{(k)} = \left[-a_k \tilde{\mathbf{X}}_t^{(k)} + \sigma\sqrt{\lambda_k}\tilde{\alpha}_t^{(k)}\right]\Delta_t + \sigma\sqrt{\lambda_k \Delta_t}\tilde{\xi}^{(k)}$
  end for
  Inverse discrete cosine transform, $\mathbf{X}_{t+\Delta_t}^\alpha = \texttt{iDCT}(\{\tilde{\mathbf{X}}_{t+\Delta_t}^{(k)}\}_{k=1}^{\mathbf{T}^2})$
  $\mathbf{X}_t^\alpha = \mathbf{X}_{t+\Delta_t}^\alpha$
end for
Output: $\mathbf{X}_T^\alpha \sim \pi_T$

matrix containing the eigenvalues $\lambda^{(n,m)}$. Hence the controlled SDEs (2) with $Q = \mathbf{I}$ can be rewritten as

$$d\tilde{\mathbf{X}}_t^\alpha = \left[-\mathbf{D}\tilde{\mathbf{X}}_t^\alpha + \sigma\tilde{\alpha_t}\right]dt + \sigma d\tilde{\mathbf{W}}_t, \quad t \in [0, T], \tag{A.72}$$

where $\tilde{\mathbf{X}}_t = \mathbf{E}^T\mathbf{X}_t$,$\tilde{\alpha}_t = \mathbf{E}^T\alpha_t$, and $\tilde{\mathbf{W}}_t \overset{d}{=} \mathbf{W}_t$ for all $t \in [0, T]$.

Sampling Algorithm. The sampling algorithm for DBFS in bridge matching, discussed in Section 5.1, is provided in detail in Algorithm 3.

Unpaired dataset Transfer Experiment. For the experiment involving transfer between dataset, we followed the setup described in [51].

(A) For the EMNIST and MNIST datasets, the initial distribution, π_0, was set as the MNIST dataset, while for the terminal distribution, π_T, we used the EMNIST dataset with the first five lowercase and uppercase characters, as outlined by [19]. The iterative training scheme proposed by [51]. was adopted, which involved two neural networks, $\alpha_t(t, \mathbf{x}, \theta)$ and $\alpha_t(t, \mathbf{x}, \psi)$, each with around 20.7 million parameters. These networks approximate mixtures of bridges for the forward ($\pi_0 \to \pi_T$) and reverse ($\pi_T \to \pi_0$) directions, respectively. The SDE was discretized into 30 steps without a noise schedule. The DBFS model was trained for 60 iterations, with each iteration comprising 5,000 gradient updates. Additionally, 2,560 cached images were used for training each network, updated every 250 steps. We used the Adam optimizer with a learning rate of 1e-4 and a batch size of 128, with the EMA rate set to 0.999. The complete DBFS training for the MNIST experiment took approximately 15 hours on a single A6000 GPU.

(B) For the AFHQ dataset [14][5], we evaluated DBFS between the `wild` and `cat` classes on a 64^2 grid, with each class containing approximately 5,000 samples. The control networks each contained about 120.3 million parameters. We discretized the SDE into 100 steps without using a noise schedule. The Adam optimizer was used with a learning rate of 1e-4, and the EMA rate was set to 0.999. We used a batch size of 64 and trained for 20 iterations, with a total of 400,000 gradient steps. Additionally, 2,560 cached images were used for training each network, updated every 1,000 steps. The training took approximately 8 days, using 8 A6000 GPUs for the experiment.

For each control network α, we used a transformer network architecture inspired by PerceiverIO [36] from public repository[6] to model functional representation. This transformer architecture was chosen for its efficiency in evaluating field representations over a large number of grid points and its

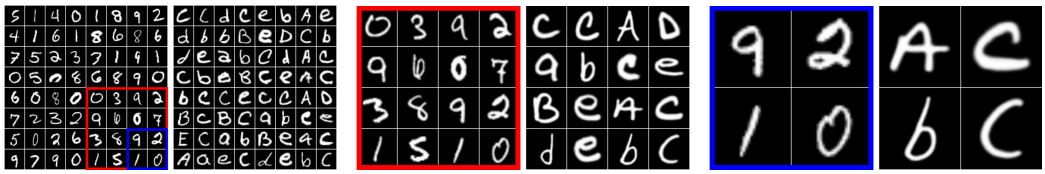
[5] https://github.com/clovaai/stargan-v2, under CC BY-NC 4.0 License.
[6] https://github.com/lucidrains/perceiver-pytorch, under MIT License.

Table A.1: Network Hyper-parameters

Dataset	Latent dim	Position dim	#heads	#enc blocks	#dec blocks	#self attn. per block	# of parameters
MNIST	256	256	4	6	2	1	20.7M
AFHQ	512	512	4	6	2	2	120.3M

ability to map arbitrary input arrays to arbitrary output arrays in a domain-agnostic way. Additionally, we adopted the attention mechanism proposed in [49]. The decoding cross-attention mechanism was also modified, inspired by [39], with the output query set to the target grid points, which were transformed as Gaussian Fourier features [67].

In practice, we start by passing the evaluations $\mathbf{X}[\mathbf{p}]$ through a single MLP layer and combine it with the Fourier feature embeddings of the grid points $\mathbf{p}$. This combined representation is then input into the encoder blocks as keys and values, where QKV attention is first applied with the latent array as the query, followed by self-attention for each block. We implement the time-modulated attention block proposed by [49], embedding the time t into latent space. Next, the target grid points, represented as Fourier feature embeddings of the grid points $\mathbf{p}$, are fed into the decoder blocks as queries. Here, QKV attention is applied with the encoded latent array serving as keys and values, followed by self-attention in each decoder block. Finally, the decoded array is mapped to grey scale or RBF channels using a single linear layer. Conceptual illustrations of the proposed network are presented in Figure A.1, and detailed network hyperparameters are listed in Table A.1.



(a) 32^2 (observed resolution) (b) 64^2 (unseen resolution) (c) 128^2 (unseen resolution)

(d) 64^2 (observed resolution) (e) 128^2 (unseen resolution)

Figure A.2: Results on Unpaired image transfer task. **(Up)** MNIST → EMNIST **(Down)** AFHQ-64 Cat → Wild. (Left) Real data and (Right) generated samples from our model. For generation at unseen resolutions, the images within the red and blue boxed initial conditions were upsampled (using bi-linear transformation) from the observed resolution (32^2) for EMNIST and (64^2) for AFHQ-64 Cat, respectively.

(a) 64^2 (observed resolution) (b) 128^2 (unseen resolution)

Figure A.3: Results on Unpaired image transfer task. AFHQ-64 Cat → Dog.

A.8.2 Experiment on 1D-Domain

For the experiments on 1D-domain, we consistently set $\mathcal{A} := -\frac{1}{2}$ and $Q := \exp(-\left\|\mathbf{p} - \mathbf{p}'\right\|^2/\gamma)$ and set $\gamma = 0.2$ and $\gamma = 0.02$ for GP regression and imputaion, respectively. The choice of γ is hyper-parameter, we search over the set $[0, 01, 0.1, 0.2, 0.5, 1.0]$ and find optimal value for GP regression. For imputation, we set $\gamma = 0.02$ by following [6].

Terminal Cost Computation For all experiments conducted in the 1D domain, we implemented a parameterized initial condition which takes as input the observed sequences $\mathbf{X}_0^\theta = \mathbf{x}^\theta(\mathbf{Y}[\mathbf{O}])$. We employed the energy functional $\mathcal{U}$ as the Gaussian negative log-likelihood (NLL). For each evaluation point on $\mathbf{T}$, $\mathcal{U}$ can be computed as follows:

$$\mathcal{U}(\mathbf{X}_T[\mathbf{T}]) = -\log \mathcal{N}(\mathbf{X}_T[T]|\mathbf{Y}, \sigma_\theta) = \sum_{i=1}^{|\mathbf{T}|} \frac{\left\|\mathbf{X}_T[\mathbf{p}_i] - \mathbf{Y}[\mathbf{p}_i]\right\|^2}{2\sigma_\theta^2}, \tag{A.73}$$

where σ_θ^2 is set as an output from the neural network in accordance with [38] for GP regression, and fixed as $\sigma_\theta^2 = 0.5$ for imputation, to establish a loss function analogous to [6]. Additionally, we specified a learnable prior distribution $\mu_{\text{prior}} = \mathcal{N}(e^{-\frac{T}{2}\mathbf{x}^\theta}, Q_T)$. Consequently, the terminal cost retains only the NLL term, simplifying the computation.

GP regression For the GP regression, we borrow the experiment setting from [38]. The model trained with curves generated from GP with RBF kernel and tested in various settings such as data generated from GP with other type of kernel (Matérn 5/2, Periodic). We generated $\mathbf{p}$ uniformly on the interval $[-2, 2]$ and generated $\mathbf{Y}[\mathbf{p}]$ from using RBF kernel $\kappa(\mathbf{p}_i, \mathbf{p}_j) = l_1^2 \exp(-\left\|\mathbf{p}_i - \mathbf{p}_j\right\|^2/l_2^2)$ with $l_1 \sim \text{Unif}(0.1, 1.0)$ and $l_2 \sim \text{Unif}(0.1, 0.6)$ and the white noise $\xi \sim \mathcal{N}(0, 1e-2)$ is added. We set $|\mathbf{O}|$ randomly from $\text{Unif}(3, 37)$ and $|\mathbf{T}|$ from $\text{Unif}(3, 50 - |\mathbf{O}|)$. For the other test data, we define $\kappa(\mathbf{p}_i, \mathbf{p}_j) = l_1^2(1 + \sqrt{5}d/l_2 + 5d^2/(3l_2^2))exp(-\sqrt{5}d/l_2)$ with $d = (\left\|\mathbf{p}_i - \mathbf{p}_j\right\|)$, $l_1 \sim \text{Unif}(0.1, 1.0)$ and $l_2 \sim \text{Unif}(0.1, 0.6)$ for Matérn kernel and $\kappa(\mathbf{p}_i, \mathbf{p}_j) = l_1^2 \exp(-2\sin^2(\pi\left\|\mathbf{p}_i - \mathbf{p}_j\right\|^2/p)/l_2)$ with $l_1 \sim \text{Unif}(0.1, 1.0)$, $l_2 \sim \text{Unif}(0.1, 0.6)$ and $p \sim \text{Unif}(0.1, 0.5)$ for periodic kernel.

We set batch size of 100 and trained for $100,000$ iterations. The Adam optimizer is used, the initial learning rate 5e-4 decayed with cosine annealing scheme. For testing, we evaluated the trained models using 3,000 batches, each consisting of 16 samples. We report the mean and standard deviation for five runs. We a single A6000 GPU for this experiment.

The architectures for NP [28] and CNP [27], we use the same setting as described in [38][7]. In our approach, we adapted the CNP architecture to incorporate a parameterized initial condition $\mathbf{x}^\theta$ (add one linear layer to output $\mathbf{x}^\theta$). The total number of parameters is similar across all three models.

Physionet Imputation The Physionet [30] contains 4000 clinical time series with 35 variables for 48 hours from intensive care unit. Following [68], we preprocess this datasets to hourly time-series which have 48 time steps. Since the dataset already contains around 80% of missingness, we randomly choose $10/50/90\%$ of observed values as test data.

In the imputation experiments, we employed the same experimental setup as [6][8] which is slight modification of original CSDI model. In this setup, the Gaussian noise of the DDPM model was replaced with GP noise employing an RBF kernel. Additionally, we adjusted the measurement approach to record the actual elapsed time rather than rounding to the nearest hour, better capturing the inherent timing characteristics of the Physionet dataset. We use a single A6000 GPU for this experiment.

[7] https://github.com/juho-lee/bnp, under MIT License.

[8] https://github.com/morganstanley/MSML/tree/main/papers/Stochastic_Process_Diffusion, under Apache 2.0 License.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction are consistent with the scope of the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the conclusion section.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, we believe our proof and assumptions are both sufficient and correct.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we believe we have provided all the necessary information to reproduce our results in Appendix A.8.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the codebase used for our experiments, particularly for unpaired image transfer.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we have provided the experimental details.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, when available, we conducted the experiments five times and reported the mean and standard deviations.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Yes, we provided the required resources in the experimental details section.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We support the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts are discussed in the conclusion section.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We adhere to ethical standards for using our model in generative AI.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, the license and terms of use are noted.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing or research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing or research with human subjects.