Flow Calibrated Reinforcement Learning with Application in Transition Path Sampling

Zexi Fan[1], Dinghuai Zhang[2] and Yiping Lu[3]

[1]*School of Mathematics, Peking University*
[2]*Mila*
[3]*IEMS, Northwestern University*

May 20, 2025

Abstract

This paper aims to develop a comprehensive method for integrating generative flow networks (GFlowNet) into reinforcement learning (RL) for stochastic control issues. Our primary goal is to use GFlowNet to mitigate the impact of terminal cost during the initial phases of RL training, a process we term "flow calibration." Specifically, we concentrate on creating a flow-calibrated soft actor-critic (SAC) algorithm to address the transition path sampling (TPS) problem, which focuses on sampling rare transitions between two metastable states. TPS can be framed as a stochastic control problem and is generally challenging to train with traditional RL techniques in high-dimensional spaces due to the significant influence of terminal cost. Furthermore, to prevent discretization errors, we propose a continuous-time formulation of SAC and GFlowNet, grounded in the stochastic optimal control matching and Girsanov theorem. The central element linking the two models is the quantization strategy, and we also employ principal bundle theory to further utilize physics informations. Numerical results of generated transition paths for the Müller potential and Alanine dipeptide validate the effectiveness of these methods in both data-rich and data-scarce scenarios.

Keywords stochastic optimal control, reinforcement learning, generative flow network, transition path sampling, rare event sampling

1 Introduction

Rare event sampling constitutes a multidisciplinary field with applications spanning finance [14], molecular dynamics[30] , and beyond. These rare events often arise due to the metastable behavior exhibited by the dynamical systems of interest. Metastability implies that the system remains confined to specific closed regions of its state space for extended periods, rarely making transitions to other closed regions. Precise sampling of these rare events necessitates a deep understanding of this transition behavior.

Remarkably, the average waiting time for a rare event significantly exceeds the intrinsic timescale of the underlying process. This phenomenon commonly manifests itself in dynamical systems governed by Langevin dynamics and navigating potentials with multiple minima. In such scenarios, the metastable closed regions align with local minima in the potential landscape. These minima are separated by energy barriers, and the transitions between them carry crucial information such as the macroscopic properties of the studied molecules. Furthermore, empirical evidence confirms that the time required to overcome these barriers scales exponentially with their height [4]. From a sampling perspective, the variance of Monte Carlo estimators associated with these rare transitions tends to be very high.

Many efforts have been devoted to the two issues presented above. The most notable methods can be classified into two groups[23]:

- **Splitting Method:** This idea involves breaking down a rare event into a series of moderately rare events, making it more tractable to sample. If only the transition between to local minima points is concerned, this procedure can be achieved adaptively using Adaptive Multilevel Splitting algorithm, which concurrently evolves an ensemble of trajectories and eliminates those who lags behind, whilst replicates that successfully explore the paths towards the target states; see [8, 2, 6, 7].

- **Importance Sampling:** This method biases the dynamics(e.g. modifying the potential) to reduce the variance, as detailed in [13, 5] and [22, Section 6.2]. If both the starting point and the target point are given, one can formulate the problem as a Schrodinger Bridge Matching problem[17] or it can be solved via an OT scheme[11]. However, to cope with more general transtions between two metastable states where the target point is not given, one may need to choose from adapting Stochastic Optimal Control formulation[39](fixed starting point) , approximating the target distribution(unknown) via a neural network[35](need sample paths) or sampling in a latent space [29] by means of Boltzmann generator(very slow). To our knowledge, none of the aforementioned methods can efficiently deal with transitions between metastable states, which are essentially two closed regions without conditioning on either the starting point or target point.

As a fundamental machine learning paradigm with notable achievements in complex, high-dimensional tasks such as Go [33, 34], recent initiatives have explored addressing classical Stochastic Optimal Control challenges, including the aforementioned one[23, 18], through a reinforcement learning (RL) lens, and is notably benefits large-scale Markov Decision Processes where traditional methods like dynamic programming are not viable. Yet, existing studies are either missing comprehensive theoretical backing [18] or are limited to discrete-time frameworks [31], both posing risks of errors in outcomes. Beyond issues of discretization, RL tends to produce a unimodal target, even when entropy regularization is employed [21]. This could lead to the accumulation of target points at local minima when a standard approximation of terminal costs is used. On the contrary, GFlowNet[3], a well-celebrated amortize sampler, is capable of generating multimodal ones, and this capability has been verified in various scenarios[19, 40]. Furthermore, when the search space is large and the time horizon is long, RL algorithms may require an extended period to converge, as only the final step's update incorporates terminal costs. This classic dilemma of exploration-exploitation is always a core topic in RL research, such as [36, 32]. In comparison, GFlowNet equipped with trajectory balance loss[27] transmits the feedback of terminal cost to early sampling steps on each training step.

To address these challenges, we undertake the following steps: Initially, leveraging a ready-made theoretical result from transition path theory[15], we develop a novel formulation suitable for creating pathways between two metastable states, specifically from a known initial distribution to an unspecified target distribution, instead of merely from point-to-point or point-to-distribution. Subsequently, we introduce continuous-time versions of SAC and GFlowNet, with their sampling policies represented by parameterized SDEs, utilizing the stochastic optimal control matching[10] and Girsanov theorem respectively to eliminate discretization errors. To resolve the exploration-exploitation trade-off, we introduce a 'flow calibration' strategy that merges the immediate feedback benefits of RL with the foresight of GFlowNet, grounded by a quantization scheme, and further incorporates physics information via principal bundle theory, utilizing the idea from equivariant flow matching[**?**]. Notably, while our focus is on the TPS problem setting, our approaches are readily adaptable to broader SOC issues, a topic we will revisit in the discussion section.

Following this thread, the paper is structured as follows. In Section 2 we set the stage of rare event simulation, present the transition path sampling problem in section 2.1 and derive its stochastic optimal control formulation for distribution to distribution case in 2.2. Section 3 is dedicated to the introduction of the RL and GFlowNet framework. In Section 3.1, we discuss the underlying theoretical framework of reinforcement learning in SOC . In Section 3.2 we will briefly recap the key ideas of Soft Actor-Crictic(SAC) algorithm[16] employed as RL part of our design. As for section 3.3, we introduce the GFlowNet for SOC, namely CFlowNet[24], under the same frame. Section 4 is devoted to continuous time generalization for both algorithms. In Section 4.1 we show how to represent the strategy of both algorithms via SDEs. More precisely, a single forward SDE for SAC and coupled SDEs for GFlowNet. In Section 4.2 we derive the continuous version of SAC from HJB equations following recent advances in continuous time RL[20]. In Section 4.3 we analyze the limit of trajectory balance loss[27] and derive its continous time adaptation from Girsanov theorem. In Section 5, we demonstrate our "physics-informed reparameterization" starting with an introduction to equivariant flow in 5.1, followed by an exploration of the connection between reparameterization and variance in 5.2, and finally, we describe our novel reparameterization approach based on principal bundle theory in 5.3. In Section 6, we provide a brief overview of stochastic quantization in 6.1 and develop our final strategy using this concept in 6.2, which seamlessly integrates the different representations of SAC and GFlowNet strategies. In Section 7, we showcase our experimental results, highlighting their superiority over existing TPS algorithms. Finally, we conclude the paper with a discussion of our findings.

2 Stochastic Optimal Control for Transition Path Sampling

[**Yiping:** this section is all prior work, so that I can skip right?] [**Zexi:** There are some little differences... 1.

I did not directly use the original proof of these theorems but proved many of them myself since I find many do not perfectly match the literature they cited. 2. To the best of my knowledge, there are no dist-to-dist formulations in the ML community. Anyway, these theorems are not new or hard to derive, so you can skip them if you want.]

In this section, we develop a distribution-to-distribution formulation for the transition path sampling problem. First, we bias the potential landscape, which changes the drift term of SDE representing original molecule dynamics, to reweigh the sampling process. Second, we show that the distribution-to-distribution case can be reduced to the point-to-distribution case and derive that the bias potential term in point to point-to-distribution case can be solved from a stochastic control problem.

2.1 Bias Potential Method

Following the settings in [35], we will derive the bias potential method for transition path sampling in this subsection. Notice that in this section, we assume that X_t always has a fixed starting point.

2.1.1 Definition of Transition Paths

Given probability space $(\Omega, \Sigma, \varrho)$. Suppose $V(x) : \mathbb{R}^d \to \mathbb{R}$ be the potential of the system, $W_t(\omega) : [0, +\infty) \times \Omega \to \mathbb{R}^d$ be the d-dimensional Brownian motion w.r.t. measure ϱ. Let two metastable states be represented by closed closed regions $A, B \subset \mathbb{R}^d$. Their respective boundaries are $\partial A, \partial B$. In studying transition paths, it is often useful to model the governing system as an SDE. In this section, we will look at the overdamped Langevin equation, given by

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta}dW_t, \quad X_0 = x, \tag{1}$$

where β is the diffusion coefficient. We omit the matrix I_d in the SDE above and the same hereafter, if the context is clear.

Let $\mathcal{F}_t$ represent the filtration generated by the process X_t up to time t. Assume the path measure of X_t is $\mathbb{P}^x$, its marginals at stopping time τ being $\mathbb{P}^x_\tau$ respectively.

Assumption 2.1. *Assume V is regular enough such that the invariant probability distribution exists*

More precisely, if it exists, then it is given by

$$p(x) = e^{-\beta^{-1}V(x)}/Z, \qquad Z = \int_\Omega e^{-\beta^{-1}V(x)}dx. \tag{2}$$

Remark 2.2. *In transition path theory(TPT), a commonly seen term is "collective variable", which are functions of atomic coordinates used for dimension reduction. The case we considered above is the formulation under collective variable, because the Langevin dynamics is overdamped and can be represented by one SDE, instead of a couple of SDEs. In fact, the original Langevin dynamics without collective variable is given by*

$$\begin{cases} dX_t &= m^{-1}P_t dt \\ dP_t &= \left[-\gamma P_t - \nabla V(X_t)\right] dt + \sqrt{2\gamma m\epsilon}dW_t, \end{cases} \tag{3}$$

and its invariant probability distribution is given by

$$\phi(x,p) = Z_H e^{-H(x,p)/\epsilon}, \quad H(x,p) = \frac{p^2}{2m} + V(x), \quad Z_H = \int_\Omega e^{-H(x,p)/\epsilon}dxdp. \tag{4}$$

It can be shown that this collective variable-free case also has a SOC formulation[39], and can be solved using the path-integral cross-entropy method[17]. There is no inherent difficulty in extending our algorithm to the free collective variable case, but we will focus on the free collective variable case for simplicity.

Definition 2.3. *The paths that go from the boundary of A to the boundary of B without returning to A are called **transition paths** [12]. Mathematically, a realization $(X_s)_{s=0}^T$ is a transition path if $X_0 \in \partial A, X_T \in \partial B$, and $X_s \notin A \cup B, \ \forall\ s \in (0,T)$.*

2.1.2 Distribution of Transition Paths

We will now examine the distribution of transition paths as in [26].

Definition 2.4. *Given time t, the **hitting time of the process** X_t w.r.t. filtration $\mathcal{F}_\infty$ starting at time s and position x is defined as:*

$$\tau_A^{(X)} := \inf\{t \geq s : X_t \in A\}, \tag{5}$$

$$\tau_B^{(X)} := \inf\{t \geq s : X_t \in B\}. \tag{6}$$

We may omit the superscript (X) sometimes if the context is clear.

Assumption 2.5. *For given X_s starting at starting point x, we always assume the system is ergodic, that is $\tau_A^{(X)}, \tau_B^{(X)} < +\infty$.*

Remark 2.6. *This condition can be slightly relaxed to $\tau_A^{(X)}, \tau_B^{(X)} < +\infty$ almost surely with respect to $\mathbb{P}^x$, the path measure of X_t[35].*

From this definition, we introduce a key ingredient for solving TPS problem named committor

Definition 2.7. *Let E be the event that $\tau_B^{(X)} < \tau_A^{(X)}$. Given starting point x, the **committor** is defined as*

$$q(x) := \mathbb{P}^x(E) \tag{7}$$

That is, the committor gives the probability that a path starting from a particular point x reaches closed region B before closed region A.

Definition 2.8. *A stopping time*

$$\tau_{AB}^{(X)} := \inf\{t \geq 0 | X_t \in A \cup B\} = \tau_A^{(X)} \wedge \tau_B^{(X)} \tag{8}$$

*is called the **reactive time** of the path X_t.*

Remark 2.9. *From the ergodic assumption, we have $\tau_{AB} < +\infty$, that is, the reactive time is always finite.*

Before we give the SDE of the transition path, we need the following lemma

Lemma 2.10. *The committor q is determined by the following boundary value problem*

$$\begin{cases} \mathcal{L}q(x) = 0 & \text{if } x \notin A \cup B \\ q(x) = 0 & \text{if } x \in A \\ q(x) = 1 & \text{if } x \in B \end{cases}, \tag{9}$$

where $\mathcal{L}$ is the generator of original diffusion(time-homogeneous), given by

$$\mathcal{L} = -\nabla V(X_t)\nabla + \beta\Delta \tag{10}$$

Proof. Suppose $h(x)$ is a solution of the boundary value problem.

For $x \in A \cup B$, apparently h coincides with q, which is 1_B.

For $x \in (A \cup B)^c$, from [28, Theorem 9.2.13], we have

$$h(x) = \mathbb{E}_{\mathbb{P}^x}[1_B(X_{\tau_{AB}})] \tag{11}$$

$$= \mathbb{P}^x(\tau_A < \tau_B) \tag{12}$$

$$= \mathbb{P}^x(E) \tag{13}$$

$$= q(x) \tag{14}$$

Hence, $h \equiv q$ for any x. Refer to [37] for more details. □

We claim that

Theorem 2.11. *The dynamics of transition paths starting at x is given by*

$$dY_t = (-\nabla V(Y_t) - 2\beta\frac{\nabla q(Y_t)}{q(Y_t)})dt + \sqrt{2\beta}dW_t, \quad Y_0 = x \in A^c \tag{15}$$

Proof. We will use Doob's h-transform, a common technique for conditional dynamic to derive the equation. Define h-function

$$h(t,x) := \mathbb{P}(\tau_B^{(X)} < \tau_A^{(X)} | X_t = x) \tag{16}$$

Denote the generator of transition paths as $\mathcal{G}^{(t)}$,i.e.

$$\begin{aligned}
\mathcal{G}^{(t)}f(x) = &\lim_{s\to 0}\frac{1}{s}\Big(\mathbb{E}\left[f(X_{t+s})|X_t = x, \tau_B^{(X)} < \tau_A^{(X)}\right] - f(x)\Big) &(17)\\
= &\lim_{s\to 0}\frac{\mathbb{E}[f(X_{t+s})h(t+s,X_{t+s})|X_t = x] - f(x)h(t,x)}{sh(t,x)} &(18)\\
= &\lim_{s\to 0}\frac{\mathbb{E}[f(X_{t+s})q(X_{t+s})|X_t = x] - f(x)q(x)}{sq(x)} &(19)
\end{aligned}$$

for arbitrary function f. From the definition above, the generator of biased diffusion(can be non-time-homogeneous in general Doob's h-transform) is given by

$$\begin{aligned}
\mathcal{G}^{(t)}f &= \frac{1}{q}(\partial_t + \mathcal{L})(qf) &(20)\\
&= \frac{1}{q}\mathcal{L}(qf), &(21)
\end{aligned}$$

because $q(x)f(x)$ apparently does not depend on time in our situation. From previous Lemma 2.10

$$\mathcal{L}q = 0 \tag{22}$$

Hence

$$\mathcal{G}^{(t)}f = \mathcal{L}f + \frac{2\beta\nabla q}{q}\cdot\nabla f \tag{23}$$

which completes our proof. See [9] for more details. □

Remark 2.12. *In fact, the theorem still holds for $x \in \partial A$, where the committor q vanishes, that is, this SDE still admits a unique strong solution. See [26, Theorem 1.1] for the proof.*

2.2 Solving Bias Potential from Stochastic Optimal Control

In this subsection, we begin with reducing the distribution to distribution case to point to distribution case, and then derive a SOC formulation from minimal action in optimal transport.

2.2.1 Reduction of initial condition

Consider Overdamped Langevin Dynamics with given initial distribution μ,i.e.

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta}dW_t, \quad X_0 \sim \mu. \tag{24}$$

Denote its corresponding path measure as $\mathbb{P}^\mu$, and its marginal at stopping time τ as $\mathbb{P}^\mu_\tau$.

Definition 2.13. *The relative entropy between two measures α, β is given by*

$$Ent(\alpha|\beta) := \begin{cases} \int_\varsigma \log\left(\frac{\mathrm{d}\alpha}{\mathrm{d}\beta}\right)\mathrm{d}\alpha & \text{if } \alpha \ll \beta, \\ +\infty & \text{otherwise,} \end{cases} \tag{25}$$

for measurable space ς.

The next theorem is related to the Schrodinger Bridge formulation of the general sampling problem, whose connection with TPS will be discussed later. We will use the proof in [15, Theorem 4.3]

Theorem 2.14. *Suppose the strictly convex terminal cost $g : \mathbb{R}^d \to [0, +\infty]$ is $\mathcal{F}_\tau$ measurable, there exists a $\mathcal{F}_\tau$ measurable set $D \subset dom\ g$,s.t. $\mathbb{P}^x_\tau(D) > 0$, and that $\int_D g d\mathbb{P}_\tau < +\infty$.*
Then, the following holds:

1. *The point-to-distribution sampling problem*

$$\inf_{\mathbb{Q}}\{\mathbb{E}_{\mathbb{Q}}[g(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}]|\mathbb{Q} \in \mathcal{P}(\Omega)\}, \tag{26}$$

   where $\mathcal{P}(\cdot)$ are all measures on certain sample space, admits unique optimal path measure $\mathbb{Q}^{\star,x}$, satisfying $\mathbb{Q}^{\star,x}_0 = \mu$

2. *The distribution-to-distribution sampling problem*

$$\inf_{\mathbb{Q}}\{\mathbb{E}_{\mathbb{Q}}[g(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^\mu}]|\mathbb{Q} \in \mathcal{P}(\Omega), \mathbb{Q}_0 = \tilde{\mu}\}, \tag{27}$$

   where $\tilde{\mu} \in \mathcal{P}(\mathbb{R}^d)$ is a given measure(not necessarily equals to μ) with $Ent(\tilde{\mu}|\mu) < +\infty$, has a minimizer taking the form $\mathbb{Q}^{\star,\tilde{\mu}} = \tilde{\mu} \otimes \mathbb{Q}^{\star,x}$ (the definition of $\otimes$ is in the proof)

Remark 2.15. *In our case,$\tilde{\mu} = \mu$. It indicates that we only need to solve the measure $\mathbb{Q}^{\star,x}$ for the point-to-distribution case on one point x. For the distribution-to-distribution case, it suffices to randomize the starting point x by $x \sim \mu$ and "shoot" with the corresponding conditional measure $\mathbb{Q}^{\star,x}$.*

Proof. We will give a proof for 1 and 2 for completeness.

1. First, we show that the set of admissible path measures is nonempty, that is, there exists at least one path measure, such that the stochastic control problem in 1 has a finite cost. To construct such a measure, we reweight the probability of $\mathbb{P}^x_\tau$ to make its support set be D,i.e.

$$\nu := \frac{1_D \mathbb{P}^x_\tau}{\mathbb{P}^x_\tau(D)}. \tag{28}$$

   Make a path measure $\mathbb{Q}$ to have marginal ν at stopping time τ, that is

$$\mathbb{Q} := \nu \otimes \mathbb{P}^x_{|X_\tau} = \int_{\mathbb{R}^d} \mathbb{P}^x_{|X_\tau=\eta} d\nu(\eta), \tag{29}$$

   where $\mathbb{P}^x_{|X_\tau=\eta}$ is measure $\mathbb{P}^x$ conditioning on $X_\tau = \eta$.
   We claim this measure is admissible, verified as follows

$$\mathbb{E}_{\mathbb{Q}}[g(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}] = \mathbb{E}_{\mathbb{Q}}[g(X_\tau) + \log \frac{d\mathbb{Q}_\tau}{d\mathbb{P}^x_\tau}(X_\tau)], \text{ since it is the only difference between } \mathbb{Q} \text{ and } \mathbb{P}^x \tag{30}$$

$$= \mathbb{E}_\nu[g(x) + \log \frac{d\nu}{d\mathbb{P}^x_\tau}(x)] \tag{31}$$

$$= \int_{\mathbb{R}^d} g d\nu + Ent(\nu|\mathbb{P}^x_\tau), \text{ where Ent is the relative entropy} \tag{32}$$

$$= \frac{\int_D g d\mathbb{P}^x_\tau}{\mathbb{P}^x_\tau(D)} - \log \mathbb{P}^x_\tau(D) \tag{33}$$

$$< +\infty \tag{34}$$

   Next, we move to the existence of minimizers by the Direct Method of the Calculus of Variations.

   Step one, construct such a candidate minimizer. From previous argument, the minimizing sequence $\mathbb{Q}^k$ exists, that is

$$\sup_{k\geq 1} \mathbb{E}_{\mathbb{Q}^k}[f(X_\tau) + \log \frac{d\mathbb{Q}^k}{d\mathbb{P}^x}] < +\infty \tag{35}$$

Since $f \geq 0$, it implies

$$\sup_{k\geq 1} Ent(\mathbb{Q}^k|\mathbb{P}^x) < +\infty \tag{36}$$

Thus, by de la Vallee Poussin theorem, the sequence $(\frac{d\mathbb{Q}^k}{d\mathbb{P}^x})_{k\geq 1}$ is uniformly integrable. Further, since the sequence is bounded in $L^1(\Omega, \mathbb{P}^x)$, Dunford-Pettis theorem provides that this sequence is also convergent weakly in $L^1(\Omega, \mathbb{P}^x)$. Denote the weak limit as Z, and let the candidate be $\mathbb{Q} := Z\mathbb{P}^x$, then we have convergence $\mathbb{Q}^k \overset{setwise}{\to} \mathbb{Q}$.

Step two, we show that $\mathbb{Q}$ is indeed the minimizer we want

$$\mathbb{E}_{\mathbb{Q}}[(g \wedge n)(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}] \leq \lim_{k\to\infty} \mathbb{E}_{\mathbb{Q}^k}[(g \wedge n)] + \liminf_{k\to\infty} Ent(\mathbb{Q}|\mathbb{P}^x), \text{ by lower semi-continuity} \tag{37}$$

$$\leq \liminf_{k\to\infty} [(g \wedge n)(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}] \tag{38}$$

$$\leq \liminf_{k\to\infty} [g(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}]. \tag{39}$$

Take $n \to \infty$. From the monotone convergence theorem, we have

$$\mathbb{E}_{\mathbb{Q}}[g(X_\tau) + \log \frac{d\mathbb{Q}}{d\mathbb{P}^x}] \leq \liminf_{k\to\infty} \mathbb{E}_{\mathbb{Q}^k}[g(X_\tau) + \log \frac{d\mathbb{Q}^k}{d\mathbb{P}^x}]. \tag{40}$$

The uniqueness of the minimizer $\mathbb{Q}$ follows directly from the strict convexity of g.

2. Notice that for any $\mathbb{Q} \in \mathcal{P}(\Omega), \mathbb{Q}_0 = \tilde{\mu}$, we have:$\mathbb{Q}_0 \ll \mu$(Since $Ent(\tilde{\mu}|\mu) < +\infty$).
Moreover, by disintegration theorem, we have:

$$\frac{d\mathbb{Q}}{d\mathbb{P}^\mu}(\omega) = \frac{d\mathbb{Q}_0}{d\mathbb{P}_0^\mu}(X_0(\omega)) \frac{d\mathbb{Q}_{|X_0}}{d\mathbb{P}^\mu_{|X_0}}(\omega), \text{ for } \mathbb{Q}\text{-almost } \omega \in \Omega \tag{41}$$

This induces the disintegration of relative entropy, which writes:

$$Ent(\mathbb{Q}|\mathbb{P}^\mu) = Ent(\tilde{\mu}|\mu) + \int_{\mathbb{R}^d} Ent(\mathbb{Q}_{|X_0=x}|\mathbb{P}^x)\mathbb{Q}_0(dx) \tag{42}$$

The problem now becomes:

$$Ent(\tilde{\mu}|\mu) + \inf_{\mathbb{Q}} \{\int_{\mathbb{R}^d} \mathbb{E}_{\mathbb{Q}_{|X_0=x}}[f(X_\tau) + \log \frac{d\mathbb{Q}_{|X_0=x}}{d\mathbb{P}^x}]\tilde{\mu}(dx)\} \tag{43}$$

Therefore, if $\mathbb{Q}^{\star,x}$ is the minimizer of the problem conditioning on initial point x, then the optimal path measure is given by $\mathbb{Q}^{\star,\tilde{\mu}} = \tilde{\mu} \otimes \mathbb{Q}^{\star,x}$

□

Since the Theorem 2.11 and Remark 2.12 only holds for $x \in (int\ A)^c$, we need to clarify the movement of the particle starting at a point within A to make the boundary distribution of Y_t on ∂A coincides with the boundary distribution of X_t.

Definition 2.16. *Suppose the initial distribution on A is given by $\phi(x)$. Define **reactive exiting distribution** of ∂A as $\mu := \phi|_{\partial A}$. Moreover, define the **empirical reactive exiting distribution** of ∂A as $\mu_N := \frac{1}{N}\sum_{k=0}^{N-1} \delta_{X_{\tau_{A,k}}(z)}$, where $z \in A$ and k is the label for N iid samples.*

Next, we extend the definition of Y_t to $x \in A$ case simply by

$$Y_t := -\nabla V(Y_t)dt + \sqrt{2\beta}dW_t, 0 \leq t \leq \tau_A, x \in A \tag{44}$$

The following proposition can justify the definition

Proposition 2.17. *$\mu_N \overset{weak}{\to} \mu$ as $N \to +\infty$,i.e.*

$$\lim_{N\to\infty} \int_{\partial A} f(x)\mathrm{d}\mu_N(x) = \int_{\partial A} f(x)\mathrm{d}\mu(x), \mathbb{P}^x - a.s. \tag{45}$$

for any continuous bounded function $f : \partial A \to \mathbb{R}$.

Proof. See [26, Proposition 1.4]. □

This proposition allows us to equalize the hitting distribution of Y_t on ∂A with μ. Combining it Theorem 2.14, it suffices to construct SOC formulation for the point-to-distribution case with $x \in \partial A$.

2.2.2 Deriving SOC formulation

Define terminal cost as follows

$$g(x) = \begin{cases} 0, x \in B \\ +\infty, otherwise \end{cases} \tag{46}$$

Pick sufficiently small $\epsilon > 0$. Generate a starting point $x \in A^c$ satisfying $d(x, \partial A) < \epsilon$, such as taking one more small step without bias after Y_t hits the boundary ∂A to go outside the closed region A.

Parameterize the path measure $\mathbb{Q}$ by setting it as the law of the following SDE

$$dY_t = \left[-\nabla V(Y_t) + 2\beta v_F^\theta(Y_t)\right] dt + \sqrt{2\beta} dW_t, \quad Y_0 = x \in A^c. \tag{47}$$

Remark 2.18. *Different from the next section, we temporarily assume that θ parameterizes v_F^θ, a deterministic mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$ here.*

To apply Girsanov's theorem to the entropy term above, we need

Assumption 2.19. $\mathbb{E}_\mathbb{Q} \log \frac{d\mathbb{Q}}{d\mathbb{P}^x} < +\infty \iff \mathbb{E}_\mathbb{Q}[\int_0^{\tau_{AB}} |v_F^\theta(Y_t)|^2 dt] < +\infty$ *(Novikov condition, derived from the following lines),*

which gives

$$\begin{aligned} \mathbb{E}_\mathbb{Q}[\log \frac{d\mathbb{Q}}{d\mathbb{P}^x}] &= \mathbb{E}_\mathbb{Q}[-\log \frac{d\mathbb{P}^x}{d\mathbb{Q}}] && (48) \\ &= \mathbb{E}_\mathbb{Q}[\frac{1}{\sqrt{2\beta}} \int_0^{\tau_{AB}} v_F^\theta(Y_t) d\breve{W}_t + \frac{1}{4\beta} \int_0^{\tau_{AB}} |v_F^\theta(Y_t)|^2 dt] && (49) \\ &= \frac{1}{4\beta} \mathbb{E}_\mathbb{Q} \int_0^{\tau_{AB}} |v_F^\theta(Y_t)|^2 dt, && (50) \end{aligned}$$

where we denote the Brownian motion w.r.t. $\mathbb{Q}$ as $\breve{W}_t$ temporarily. Thus the original SOC problem is equivalent to

$$\inf_\theta \mathbb{E}_\mathbb{Q}[\frac{1}{4\beta} \int_0^{\tau_{AB}} |v_F^\theta(Y_t)|^2 dt + g(Y_{\tau_{AB}}) | Y_t \sim \mathbb{Q}, Y_0 = x]. \tag{51}$$

The next theorem shows that the stochastic optimal control problem presented above gives the transition path measure

Corollary 2.20. *For $x \in A^c$, the optimal control for the problem*

$$\inf_\theta \mathbb{E}_\mathbb{Q}[\frac{1}{4\beta} \int_0^{\tau_{AB}} |v_F^\theta(Y_t)|^2 dt + g(Y_{\tau_{AB}}) | Y_t \sim \mathbb{Q}, Y_0 = x], \tag{52}$$

is given by

$$v_F^{\star,\theta}(x) = \nabla \log q(x) = \frac{\nabla q(x)}{q(x)}. \tag{53}$$

That is, the optimal path measure for the original problem is the measure of the transition path Y_t.

Proof. See [39, Corollary 3.1.1]. □

Here is a brief summary of the procedure for our sampling

1. Choose a point on ∂A, and take a sufficiently small step following X_t to generate a starting point $x \in A^c$
2. For this fixed x, solve the SOC problem in corollary 2.20 for $v_F^{\star,\theta}(x)$
3. Pick any fixed point x_{in} in *int* A(normally the local minima)
4. Simulate empirical exiting measure μ_N starting from x_{in}
5. Solve $v_F^{\star,\theta}$ for all point measures in μ_N and denote the corresponding measures as $\mathbb{Q}^{\star,x}$ following 1-2
6. Generate paths from $\mathbb{Q}^\star{}_N = \mu_N \otimes \mathbb{Q}^{\star,x}$

Thus, we will focus on solving SOC problem for a fixed starting point in the following sections.

3 Discretized RL and GFN for Stochastic Optimal Control

3.1 Discretized RL for stochastic optimal control

First, to avoid an infinite time horizon, pick a sufficiently large constant $\tilde{T}$, and set stopping time $T = \tilde{T} \wedge \tau_{AB}$.

Discretize the objective function and the dynamics with time step Δt [**Yiping:** you are minimizing respect to θ, but where is the θ in your objective. I guess you are minimizing over v_t] [**Zexi:** Yes and no. v_t represents the value of v_F^θ we detect on the sampled trajectory, while v_F^θ is the randomized bias term we are really interested in.]

$$\inf_\theta \mathbb{E}_{\hat{\mathbb{Q}}}\left[\frac{1}{4\beta}\sum_{t=0}^{T-1}\Delta t|v_t|^2 + g(Y_T)|Y_0 = x\right]$$
$$\text{s.t. } Y_{t+\Delta t} = Y_t + [-\nabla V(Y_t) + 2\beta v_t]\,\Delta t + \sqrt{2\beta}\Delta W_t,$$

where T is the time step when $Y_{1:T-1} \in (A \cup B)^c$,$\Delta W_t \sim \mathcal{N}(0, \Delta t)$, and $\hat{\mathbb{Q}}$ is the discretized adaptation of path measure $\mathbb{Q}$.

Second, we introduce regular elements in RL problem and clarify some concept in the equation above.

Let the state space $\mathscr{S} := \mathbb{R}^d$, the policy space $\mathscr{P} := \{\pi_F^\theta | \pi_F^\theta : Y_t \in \mathbb{R}^d \to \pi_F^\theta(\cdot|Y_t) \in \mathcal{P}(\mathbb{R}^d)\}$, namely all the mixed strategies on $\mathbb{R}^d$(we may choose different parameterized policy space $\mathscr{P}_\theta$ according to the specific RL algorithm we use in practice). Denote the selected action at time t as $v_t \in \mathscr{A} = \mathbb{R}^d$, where $\mathscr{A}$ stands for action space.

Remark 3.1. *In fact, due to the deterministic $\nabla \log q(x)$, we should have used the deterministic action space. However, in practice, the agent may benefit from training with stochastic strategy when the search space is very large, such as avoiding suboptimal policies, obtaining more robust learning outcomes, etc, especially for SAC algorithm[16], which is the case we will be concerned about.*

Define the intermediate reward to be $r(v_t) := \frac{1}{4\beta}\Delta t|v_t|^2$, and let the terminal reward be $g(Y_T)$. This formulation allows us to solve the discretized problem with RL, which is Soft Actor-Crictic in our situation.

3.2 Soft Actor Crictic for Discretized SOC Problem

The key features of SAC algorithm are summarized as follows

1. Use entropy regularization to make the agent more explorative [**Yiping:** will this make the solution biased] [**Zexi:** I am not very sure. SAC is not guaranteed to converge to something in my understanding. It only guarantees that the value function will not get worse when you take gradient.]
2. Use clipped double $Q(s, a)$ function [**Yiping:** what does this mean] [**Zexi:** see the pseudocode] to avoid overparameterization [**Yiping:** why] [**Zexi:** I guess this is because that there would be an additional network parameterizing the value function if we do not do so.] and overestimation
3. Use Gaussian policy, where the variance term is also parameterized, to make a bigger hypothesis set

Since we focus on continuous cases, we will not give a full introduction to the original SAC algorithm. Here is a complete pseudo-code implementation of SAC adapted from [1] for the discretized problem

Algorithm 1 Soft Actor-Critic (SAC) algorithm for SOC problem

Initialize policy parameters θ, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$, tradeoff constant γ, target parameter $\phi_{targ,1} \leftarrow \phi_1, \phi_{targ,2} \leftarrow \phi_2$, and smoothing parameter ρ.
repeat
  Observe state Y_t and select action v_t from distribution $\pi_F^\theta(\cdot|Y_t)$.
  Execute action v_t in the environment.
  Observe next state Y_{t+1}, reward r_{t+1}, and done signal d to indicate $Y_{t+1} \in A \cup B$.
  if d is True **then** $r_{t+1} = r_{t+1} - g(Y_{t+1})$ and reset the environment state.
  Store transition $(Y_t, v_t, r_{t+1}, Y_{t+1}, d)$ in replay buffer $\mathcal{D}$.
  if it's time to update **then**
    for each gradient step **do**
      Randomly sample a batch B of transitions $(Y_t, v_t, r_{t+1}, Y_{t+1}, d)$ from $\mathcal{D}$.
      Compute targets for the Q functions

12: $$y_q = r_t + \gamma(1-d)\min_{i=1,2} Q_{\phi_{targ,i}}(Y_{t+1}, v_{t+1}) - \gamma \log \pi_F^\theta(v_{t+1}|Y_{t+1}), \quad v_{t+1} \sim \pi_F^\theta(\cdot|Y_{t+1})$$

13: Update Q-functions by one step of gradient descent using

14: $$\nabla_{\theta_i} \frac{1}{|B|} \sum_{(Y_t, v_t, r_{t+1}, Y_{t+1}, d) \in B} \left(Q_{\phi_i}(Y_t, v_t) - y_q\right)^2, \text{ for } i \in \{1, 2\}.$$

15: Update policy by one step of gradient ascent using

16: $$\nabla_\phi \frac{1}{|B|} \sum_{Y_t \in B} \left(\min_{i=1,2} Q_{\phi_i}(Y_t, v_t) - \gamma \log \pi_F^\theta(v_t|Y_t) \right),$$

17: where the reparametrization trick is used, that is

$$v_t = \mu_\theta(Y_t) + \sigma_\theta(Y_t) \cdot \xi, \quad \xi \sim \mathcal{N}(0, I_d). \tag{54}$$

18: Update target network

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1-\rho)\phi_i \quad \text{for} \quad i = 1, 2 \tag{55}$$

19: **end for**
20: **end if**
21: **until** convergence

At test time, we will output the mean policy μ_θ to improve the behavior.

Remark 3.2. *Three key issues for the original SAC*

1. *Discretization error*
2. *Only consider terminal cost at the final step*
3. *may not generate very diverse trajectories or target points in sampling applications*

3.3 GFlowNet for Discretized SOC Problem

The key difference between GFlowNet and SAC is that it does not directly compute the reward at each intermediate step [**Yiping:** we have estimation of Q function, why you claim this, any referneces?] [**Zexi:** I don't see your point. SAC computes intermediate rewards but GFlowNet typically not. Any issues?], instead, it collects all the reward at the final step and learns to construct consistent probability flow(both forward and backward) to match the distribution of the final reward, which is better regarding exploration ability[3] and is particularly suitable for sparse signal RL scenario.

Likewise, since we focus on continuous cases, we will only give the pseudo-code of it based on [24] and leave more space for continuous case

Remark 3.3. *Two key issues for GFN*

1. *Not very suitable for problems with intermediate cost[24]*
2. *Trajectory balance is very costly to train since they are evaluated on the entire trajectory at a time compared to SAC, especially when T is large.*

4 Continuous Generalization of SAC and GFN

[**Yiping:** jianfeng already done continouous SAC? I think focus on condintouus GFN is better] [**Zexi:** He uses TD3. Even Xunyu Zhou's paper uses discretization to compute values at last. I think there might be inherent computational difficulties if we do not use SOCM.]

Algorithm 2 Generative Flow Network (GFlowNet) algorithm for SOC Problem

Initialize: Policy network θ for π_F^θ, a pre-trained retrieval network Υ to find parent, empty buffer $\mathcal{D}$ and $\mathcal{P}$, clip constant ϵ, reward weight λ

repeat

Set $t = 0$, $Y_0 = x$

while $Y_t \neq$ *terminal* and $t < \tilde{T}$ **do**

Sample M actions v_t from action space $\mathscr{A}$ according to $\pi_F^\theta(\cdot|Y_t)$

Execute v_t in the environment to obtain r_{t+1} and s_{t+1}

$t = t + 1$

end while

Store episodes $\{(Y_t, v_t, r_{t+1}, Y_{t+1})\}_{t=1}^T$ in replay buffer $\mathcal{D}$

[Optional] Fine-tuning retrieval network G_Υ based on $\mathcal{D}$

Sample a random batch B of episodes from $\mathcal{D}$

Uniformly sample K actions $\{v_t^k\}_{k=1}^K$ from action space $\mathscr{A}$ for each state in B

Compute parent states according to $\{G_\Upsilon(Y_t^k, v_t^k)\}_{k=1}^K$ for each state in B

Update π_F^θ according to

$$\mathcal{L}_\theta(\tau) = \sum_{t=1}^{T} \left\{ \log \left[\epsilon + \sum_{k=1}^{K} \hat{\mathbb{Q}}(G_\Upsilon(Y_t^k, v_t^k) \overset{v_t^k}{\to} Y_t^k) \right] - \log \left[\epsilon + \sum_{k=1}^{K} (\hat{\mathbb{Q}}(Y_t^k \overset{v_t^k}{\to} Y_{t+1}^k) + \lambda(r(v_t) - 1_{t=T} g(Y_T))) \right] \right\}^2, \tag{56}$$

until convergence

4.1 Representation of Continuous Strategies via SDEs

4.1.1 Representation of SAC Strategy via Forward SDE

Since the transition dynamic Y_t for SAC is already determined $v_t \sim \pi_F^\theta(\cdot|Y_t)$, a feedback stochastic control policy, the SAC strategy can be simply given by

$$v_t = \mu_\theta(Y_t) + \sigma_\theta(Y_t) \cdot \xi, \quad \xi \sim \mathcal{N}(0, \sqrt{h} I_d) \tag{57}$$

$$dY_t = \left[-\nabla V(Y_t) + 2\beta v_t \right] dt + \sqrt{2\beta} dW_t \tag{58}$$

where $\mu_\theta, \sigma_\theta$ are some suitable NNs, and h being some suitable discretization step size. The unique strong solution does exist for arbitrary fixed $(v_t)_{t=0}^T$ from [38, Chapter 1, Theorem 6.16] under mild assumptions, as long as we assume g is already approximated by some smooth functions, which is usually the case in practice[39, 18]. More rigorously, we need to write the variables as v_t^π, Y_t^π, but we will stick to the original notation if the context is clear, for consistency.

Remark 4.1. *In this light, we are dealing with a model-based case in TPS problem. However, our method can be extended to model-free cases without inherent difficulty.*

4.1.2 Representation of GFlowNet Strategy via coupled SDEs

Next, we will derive the coupled SDE for GFlowNet. Let us establish the theorem for a more general case. Notice, that all the ∇, Δ, div is only taken to the spatial component in the following theorem and its proof.

Theorem 4.2. *Suppose we are given a general diffusion process*

$$dX_t = f(X_t, t)dt + \sigma(X_t, t)dW_t. \tag{59}$$

where $f : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}, \sigma : \mathbb{R}^{d+1} \to \mathbb{R}^{(d+1)\times(d+1)}$. Its time reversal is given by the following equation:

$$dX_{\tilde{t}} = [f(X_{\tilde{t}}, \tilde{t}) - div(\sigma(X_{\tilde{t}}, \tilde{t})\sigma^T(X_{\tilde{t}}, \tilde{t})) \tag{60}$$

$$- \sigma(X_{\tilde{t}}, \tilde{t})\sigma^T(X_{\tilde{t}}, \tilde{t})\nabla \log p(X_{\tilde{t}}, \tilde{t})]d\tilde{t} + \sigma(X_{\tilde{t}}, \tilde{t})d\tilde{W}_{\tilde{t}}. \tag{61}$$

Here, we denote the time with a negative increment with $\tilde{t}$ and the backward Brownian motion with $\tilde{W}_{\tilde{t}}$.

Proof. Write the forward FPK equation

$$\frac{\partial p_t}{\partial t}(X_t,t) = -div\left\{f(X_t,t)p(X_t,t) - \frac{1}{2}div[\sigma(X_t,t)\sigma^T(X_t,t)p(X_t,t)]\right\}. \tag{62}$$

Notice

$$\begin{aligned} & div[\sigma(X_t,t)\sigma^T(X_t,t)p(X_t,t)] && (63)\\ =& p(X_t,t)div[\sigma(X_t,t)\sigma^T(X_t,t)] + \sigma(X_t,t)\sigma^T(X_t,t)\nabla p(X_t,t) && (64)\\ =& p(X_t,t)div[\sigma(X_t,t)\sigma^T(X_t,t)] + p(X_t,t)\sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t). && (65)\end{aligned}$$

We have

$$\begin{aligned} \frac{\partial p}{\partial t}(X_t,t) =& - div\{[f(X_t,t) && (66)\\ & - \frac{1}{2}div(\sigma(X_t,t)\sigma^T(X_t,t)) && (67)\\ & - \frac{1}{2}\sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)] \cdot p(X_t,t)\}. && (68)\end{aligned}$$

It corresponds to forward ODE

$$\begin{aligned} dX_t =& [f(X_t,t) - \frac{1}{2}div(\sigma(X_t,t)\sigma^T(X_t,t)) && (69)\\ & - \frac{1}{2}\sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)]dt. && (70)\end{aligned}$$

[**Yiping:** I havn't read the proof in the textbook on this. Is this the standard way to express the proof of this in textbook/papers?] [**Zexi:** Neither do I. But I checked a few lecture notes and they both give very similar proofs.] It gives the reverse ODE with a positive time increment (t starts from the beginning).

$$\begin{aligned} dX_t =& - [f(X_t,t) - \frac{1}{2}div(\sigma(X_t,t)\sigma^T(X_t,t)) && (71)\\ & - \frac{1}{2}\sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)]dt. && (72)\end{aligned}$$

The FPK equation for reverse ODE with a positive time increment is

$$\begin{aligned} \frac{\partial p}{\partial t}(X_t,t) =& - div\{-[f(X_t,t) - \frac{1}{2}div(\sigma(X_t,t)\sigma^T(X_t,t)) && (73)\\ & - \frac{1}{2}\sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)] \cdot p(X_t,t)\} && (74)\\ =& - div\{-[f(X_t,t) - div(\sigma(X_t,t)\sigma^T(X_t,t)) && (75)\\ & - \sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)] \cdot p(X_t,t)\} && (76)\\ & + \frac{1}{2}\Delta(\sigma(X_t,t)\sigma^T(X_t,t)p(X_t,t)). && (77)\end{aligned}$$

Thus, the corresponding SDE with a positive time increment is

$$\begin{aligned} dX_t =& - [f(X_t,t) - div(\sigma(X_t,t)\sigma^T(X_t,t)) && (78)\\ & - \sigma(X_t,t)\sigma^T(X_t,t)\nabla \log p(X_t,t)]dt + \sigma(X_t,t)dW_t. && (79)\end{aligned}$$

Reverse the time, we have SDE with a negative time increment ($\tilde{t}$ starts from the end)

$$\begin{aligned} dX_{\tilde{t}} =& [f(X_{\tilde{t}},\tilde{t}) - div(\sigma(X_{\tilde{t}},\tilde{t})\sigma^T(X_{\tilde{t}},\tilde{t})) && (80)\\ & - \sigma(X_{\tilde{t}},\tilde{t})\sigma^T(X_{\tilde{t}},\tilde{t})\nabla \log p(X_{\tilde{t}},\tilde{t})]d\tilde{t} + \sigma(X_{\tilde{t}},\tilde{t})d\tilde{W}_{\tilde{t}}, && (81)\end{aligned}$$

as desired. □

Unlike SAC, we assume the strategy v_t is also evolving in an SDE form, that is

$$dv_t = \mu_\theta(Y_t)dt + \sigma_\theta(Y_t)d\breve{W}_t, \tag{82}$$

where $\breve{W}_t$ is the noise independent of W_t, $\mu_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is the mean strategy and $\sigma_\theta : \mathbb{R}^d \to \mathbb{R}$ is the noise level of the strategy.

Remark 4.3. *We can choose $\sigma_\theta : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ if the world model is very complex. However, this is not likely the case in TPS, since the dynamics is explicitly given. We will only use a parameterized constant to control the noise level, which, to some extent, presumes an isotropy of the strategy.*

We want to derive an explicit reverse SDE system corresponding to

$$\begin{cases} dY_t = [-\nabla V(Y_t) + 2\beta v_t]dt + \sqrt{2\beta}dW_t \\ dv_t = \mu_\theta(Y_t)dt + \sigma_\theta(Y_t)d\breve{W}_t, \end{cases} \tag{83}$$

[**Yiping:** why the v dyamic has a brownian motion?] [**Zexi:** This SDE system matches the "half-discretized" one of SAC. The aim is to train them jointly.] which can be rewritten as

$$d\begin{pmatrix} Y_t \\ v_t \end{pmatrix} = \begin{pmatrix} -\nabla V(Y_t) + 2\beta v_t \\ \mu_\theta(Y_t) \end{pmatrix} dt + \begin{pmatrix} \sqrt{2\beta}I_d & 0 \\ 0 & \sigma_\theta(Y_t) \end{pmatrix} \cdot \begin{pmatrix} dW_t \\ d\breve{W}_t \end{pmatrix}. \tag{84}$$

Applying theorem 4.2, we immediately have

Corollary 4.4. *The reverse representation of the system is given by*

$$\begin{cases} dY_{\tilde{t}} = [-\nabla V(Y_{\tilde{t}}) + 2\beta v_{\tilde{t}} - 2\beta\nabla \log p^{(1)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})]d\tilde{t} + \sqrt{2\beta}dW_{\tilde{t}} \\ dv_{\tilde{t}} = [\mu_\theta(Y_{\tilde{t}}) - \sigma_\theta(Y_{\tilde{t}})\nabla\sigma_\theta(Y_{\tilde{t}}) - \sigma_\theta^2(Y_{\tilde{t}})\nabla \log p^{(2)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})]d\tilde{t} + \sigma_\theta(Y_{\tilde{t}})d\breve{W}_{\tilde{t}} \end{cases}, \tag{85}$$

where $\nabla \log p^{(j)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t}), j = 1, 2$ stands for jth component of $\nabla \log p(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})$, each of which settles in $\mathbb{R}^d$.

Proof.

$$d\begin{pmatrix} Y_{\tilde{t}} \\ v_{\tilde{t}} \end{pmatrix} = [\begin{pmatrix} -\nabla V(Y_{\tilde{t}}) + 2\beta v_{\tilde{t}} \\ \mu_\theta(Y_{\tilde{t}}) \end{pmatrix} - \tag{86}$$

$$div \begin{pmatrix} 2\beta & 0 \\ 0 & \sigma_\theta^2(Y_{\tilde{t}}) \end{pmatrix} - \begin{pmatrix} 2\beta & 0 \\ 0 & \sigma_\theta^2(Y_{\tilde{t}}) \end{pmatrix} \nabla \log p(\begin{pmatrix} Y_{\tilde{t}} \\ v_{\tilde{t}} \end{pmatrix}, \tilde{t})]d\tilde{t} \tag{87}$$

$$+ \begin{pmatrix} \sqrt{2\beta}dW_{\tilde{t}} \\ \sigma_\theta(Y_{\tilde{t}})d\breve{W}_{\tilde{t}}) \end{pmatrix} \tag{88}$$

$$= [\begin{pmatrix} -\nabla V(Y_{\tilde{t}}) + 2\beta v_{\tilde{t}} \\ \mu_\theta(Y_{\tilde{t}}) \end{pmatrix} - \tag{89}$$

$$div \begin{pmatrix} 2\beta & 0 \\ 0 & \sigma_\theta^2(Y_{\tilde{t}}) \end{pmatrix} - \begin{pmatrix} 2\beta & 0 \\ 0 & \sigma_\theta^2(Y_{\tilde{t}}) \end{pmatrix} \begin{pmatrix} \nabla \log p^{(1)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t}) \\ \nabla \log p^{(2)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t}) \end{pmatrix}]d\tilde{t} \tag{90}$$

$$+ \begin{pmatrix} \sqrt{2\beta}dW_{\tilde{t}} \\ \sigma_\theta(Y_{\tilde{t}})d\breve{W}_{\tilde{t}} \end{pmatrix} \tag{91}$$

$$= \begin{pmatrix} -\nabla V(Y_{\tilde{t}}) + 2\beta v_{\tilde{t}} - 2\beta\nabla \log p^{(1)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t}) \\ \mu_\theta(Y_{\tilde{t}}) - 2\sigma_\theta(Y_{\tilde{t}})\nabla\sigma_\theta(Y_{\tilde{t}}) - \sigma_\theta^2(Y_{\tilde{t}})\nabla \log p^{(2)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t}) \end{pmatrix} d\tilde{t} \tag{92}$$

$$+ \begin{pmatrix} \sqrt{2\beta}dW_{\tilde{t}} \\ \sigma_\theta(Y_{\tilde{t}})d\breve{W}_{\tilde{t}} \end{pmatrix}, \tag{93}$$

which completes our proof. □

Parameterize $\nabla \log p(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})$ by $s_\varpi(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})$, we have

$$\begin{cases} dY_{\tilde{t}} = [-\nabla V(Y_{\tilde{t}}) + 2\beta v_{\tilde{t}} - 2\beta s_\varpi^{(1)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})]d\tilde{t} + \sqrt{2\beta}dW_{\tilde{t}} \\ dv_{\tilde{t}} = [\mu_\theta(Y_{\tilde{t}}) - 2\sigma_\theta(Y_{\tilde{t}})\nabla\sigma_\theta(Y_{\tilde{t}}) - \sigma_\theta^2(Y_{\tilde{t}})s_\varpi^{(2)}(v_{\tilde{t}}, Y_{\tilde{t}}, \tilde{t})]d\tilde{t} + \sigma_\theta(Y_{\tilde{t}})d\breve{W}_{\tilde{t}} \end{cases}, \tag{94}$$

which will be trained by flow matching[25] or trajectory balance[27].

4.2 Continuous Generalization of SAC and GFN

4.2.1 Continuous Generalization of SAC

We derive the continuous generalization of SAC from the framework of q-learning[20], using the HJB equation.

The value function from time t is given by

$$J(t,y;\pi)=\mathbb{E}_{\mathbb{H}}\bigg[\int_t^T\bigg[\frac{1}{4\beta}|v_s|^2-\gamma\log\pi(v_s|Y_s)\bigg]\,\mathrm{d}s+g(Y_T)\bigg|Y_t=y\bigg], \tag{95}$$

where γ is the exploration-exploitation ratio, and $\mathbb{E}_{\mathbb{H}}$ is taken concerning both Y_t and π.

Next, we define $Q_{\Delta t}(t,y,v;\pi)$ function as the following

$$Q_{\Delta t}(t,y,v;\pi)=\mathbb{E}_{\mathbb{H}}[\int_t^{t+\Delta t}\frac{1}{4\beta}|v|^2ds+\int_{t+\Delta t}^T[\frac{1}{4\beta}|v_s|^2-\gamma\log\pi(v_s|Y_s^v)]ds+g(Y_T^v)|Y_t^v=y], \tag{96}$$

which represents the value of action v lasts Δt from time t at state y. Here, we use superscript v to indicate variables perturbed by v.

Taking limit, we have[20, Propostion 3]

Proposition 4.5. *The continuous generalization of Q function, namely q function in the literature, is given by*

$$\begin{aligned}q(t,y,v;\pi)&:=\lim_{\Delta t\to0}\frac{Q_{\Delta t}(t,y,v;\pi)-J(t,y;\pi)}{\Delta t} &(97)\\ &=\frac{\partial J}{\partial t}(t,y;\pi)+H(t,y,v,\frac{\partial J}{\partial y}(t,y;\pi),\frac{\partial^2J}{\partial y^2}(t,y;\pi)), &(98)\end{aligned}$$

where H is the Hamiltonian, defined as

$$H(y,v,\mathbf{p},\mathbf{q})=\big[-\nabla V(y)+2\beta v\big]\cdot\mathbf{p}+\beta\cdot\mathbf{q}+\frac{1}{4\beta}|v|^2. \tag{99}$$

Proof.

$$\begin{aligned}Q_{\Delta t}(t,y,v;\pi)=&\mathbb{E}_{\mathbb{H}}[\int_t^{t+\Delta t}\frac{1}{4\beta}|v|^2ds+\int_{t+\Delta t}^T[\frac{1}{4\beta}|v_s|^2-\gamma\log\pi(v_s|Y_s^v)]ds+g(Y_T^v)|Y_t^v=y] &(100)\\ =&\mathbb{E}_{\mathbb{H}}[\int_t^{t+\Delta t}\frac{1}{4\beta}|v|^2ds+\mathbb{E}_{\mathbb{H}}[\int_{t+\Delta t}^T[\frac{1}{4\beta}|v_s|^2-\gamma\log\pi(v_s|Y_s^v)]ds+g(Y_T^v)|Y_{t+\Delta t}^v]|Y_t^v=y] &(101)\\ =&\mathbb{E}_{\mathbb{Q}}[\int_t^{t+\Delta t}\frac{1}{4\beta}|v|^2ds+J(t+\Delta t,Y_{t+\Delta t}^v;\pi)|Y_t^v=y] &(102)\\ =&\mathbb{E}_{\mathbb{Q}}[\int_t^{t+\Delta t}\frac{1}{4\beta}|v|^2ds+J(t+\Delta t,Y_{t+\Delta t}^v;\pi)-J(t,Y_t^v;\pi)|Y_t^v=y]+J(t,Y_t^v;\pi) &(103)\\ =&\mathbb{E}_{\mathbb{Q}}[\int_t^{t+\Delta t}[\frac{\partial J}{\partial t}(s,Y_s^v;\pi)+H(s,Y_s^v,v,\frac{\partial J}{\partial y}(s,Y_s^v;\pi),\frac{\partial^2J}{\partial y^2}(s,Y_s^v;\pi))]ds|Y_t^v=y]+J(t,y;\pi) &(104)\\ =&J(t,y;\pi)+[\frac{\partial J}{\partial t}(s,Y_s^v;\pi)+H(s,Y_s^v,v,\frac{\partial J}{\partial y}(s,Y_s^v;\pi),\frac{\partial^2J}{\partial y^2}(s,Y_s^v;\pi))]\Delta t+o(\Delta t). &(105)\end{aligned}$$

which completes the proof. □

The following theorem establishes that the value function J will be improved by minimizing the relative entropy between π and $\exp(\frac{1}{\gamma}H)$

Theorem 4.6. *Given $(t,y)\in[0,T]\times\mathbb{R}^d$, if two policies $\pi,\pi'\in\mathscr{P}$, if*

$$Ent(\pi'(\cdot|t,y)|\exp(\frac{1}{\gamma}H(t,y,\cdot,\frac{\partial J}{\partial y}(t,y;\pi),\frac{\partial^2J}{\partial y^2}(t,y;\pi))))\le Ent(\pi(\cdot|t,y)|\exp(\frac{1}{\gamma}H(t,y,\cdot,\frac{\partial J}{\partial y}(t,y;\pi),\frac{\partial^2J}{\partial y^2}(t,y;\pi)))), \tag{106}$$

then

$$J(t,y;\pi')\ge J(t,y;\pi). \tag{107}$$

Proof. See [20, Theorem 10]. □

Remark 4.7. *This theorem does not guarantee the policy π to converge to the optimal one $\pi^\star$, which is also the case in discretized SAC algorithm[16].*

When we reweigh the probability of actions(at a time) given t, y, π, the value of $\frac{\partial J}{\partial t}(t, y; \pi)$ is fixed, we have

$$\exp(\frac{1}{\gamma}H(t, y, \cdot, \frac{\partial J}{\partial y}(t, y; \pi), \frac{\partial^2 J}{\partial y^2}(t, y; \pi)))) \propto \exp(\frac{1}{\gamma}q(t, y, v; \pi))) \tag{108}$$

Hence, an alternative is to update by

$$Ent(\pi'(\cdot|t, y)|\exp(\frac{1}{\gamma}q(t, y, v; \pi))) \leq Ent(\pi(\cdot|t, y)|\exp(\frac{1}{\gamma}q(t, y, v; \pi))). \tag{109}$$

However, we will use H to avoid the additional computational cost induced by $\frac{\partial J}{\partial t}$.

According to our previous parameterization

$$\pi_F^\theta(v_t|Y_t) = \frac{1}{\sqrt{(2\pi h\sigma_\theta^2(Y_t))^d}} \exp\left(-\frac{1}{2h\sigma_\theta^2(Y_t)}(v_t - \mu_\theta(Y_t))^T(v_t - \mu_\theta(Y_t))\right). \tag{110}$$

We derive the optimality condition as the following

$$\begin{aligned}
\frac{\partial}{\partial\theta}Ent(\pi_F^\theta|\exp(\frac{1}{\gamma}H)) &= \frac{\partial}{\partial\theta}\int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\pi_F^\theta(v|t, y)dv && (111)\\
&= \int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\frac{\partial\pi_F^\theta(v|t, y)}{\partial\theta}dv && (112)\\
&+ \int_{\mathscr{A}}\frac{\partial}{\partial\theta}(\log\pi_F^\theta(v|t, y))\pi_F^\theta(v|t, y)dv && (113)\\
&= \int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\frac{\partial\pi_F^\theta(v|t, y)}{\partial\theta}dv && (114)\\
&+ \int_{\mathscr{A}}\frac{\partial}{\partial\theta}\pi_F^\theta(v|t, y)dv && (115)\\
&= \int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\frac{\partial\pi_F^\theta(v|t, y)}{\partial\theta}dv && (116)\\
&+ \frac{\partial}{\partial\theta}\int_{\mathscr{A}}\pi_F^\theta(v|t, y)dv && (117)\\
&= \int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\frac{\partial\pi_F^\theta(v|t, y)}{\partial\theta}dv && (118)\\
&= \int_{\mathscr{A}}[\log\pi_F^\theta(v|t, y) - \frac{1}{\gamma}H(t, y, v, \frac{\partial J}{\partial y}(t, y; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, y; \pi_F^\theta))]\frac{\partial}{\partial\theta}(\log\pi_F^\theta(v|t, y))\pi_F^\theta(v|t, y)dv. && (119)
\end{aligned}$$

We have

$$\begin{aligned}
\log\pi_F^\theta(v_t|Y_t) &= -\frac{d}{2}\log(2\pi h) - d\log\sigma_\theta(Y_t) - \frac{1}{2\sigma_\theta^2(Y_t)}(v_t - \mu_\theta(Y_t))^T(v_t - \mu_\theta(Y_t))\\
\Longrightarrow\frac{\partial}{\partial\theta}\log\pi_F^\theta(v_t|Y_t) &= -\frac{d}{\sigma_\theta(Y_t)}\frac{\partial\sigma_\theta(Y_t)}{\partial\theta} + \frac{(v_t - \mu_\theta(Y_t))^T(v_t - \mu_\theta(Y_t))}{3\sigma_\theta^3(Y_t)}\frac{\partial\sigma_\theta(Y_t)}{\partial\theta} + \frac{1}{\sigma_\theta^2(Y_t)}(v_t - \mu_\theta(Y_t))^T\frac{\partial\mu_\theta(Y_t)}{\partial\theta}
\end{aligned}$$

If an accurate approximation of $H(t, y, \cdot, \frac{\partial J}{\partial y}(t, y; \pi), \frac{\partial^2 J}{\partial y^2}(t, y; \pi))$ is available, we can improve current policy π_F^θ by sample random actions v_t from it and take gradient steps, either offline or online

$$\theta \leftarrow \theta - [\log\pi_F^\theta(v_t|t, Y_t) - \frac{1}{\gamma}H(t, Y_t, v_t, \frac{\partial J}{\partial y}(t, Y_t; \pi_F^\theta), \frac{\partial^2 J}{\partial y^2}(t, Y_t; \pi_F^\theta))]\frac{\partial}{\partial\theta}(\log\pi_F^\theta(v_t|t, Y_t))^T \tag{120}$$

where the explicit formula of $\frac{\partial}{\partial\theta}\log\pi_F^\theta(v_t|Y_t)$ is given above.

Next, we derive how to get an satisfying approximation of $H(t, Y_t, v_t, \frac{\partial J}{\partial y}(t, Y_t; \pi), \frac{\partial^2 J}{\partial y^2}(t, Y_t; \pi))$. Instead of approximating q from Q by discretization or defining adjoint state , we directly compute Hamiltonian using the Girsanov reparameterization trick in [10, Appendix C.2, Proposition 4]. We will slightly generalize the diffusion term of the original proposition in the diffusion term from time-dependent-only to time-space-denpendent. The idea of proof is roughly the same.

Theorem 4.8. *Consider diffusion process*

$$dX_s = b(X_s, s)ds + \sigma(X_s, s)dW_s, X_0 = x \in \mathbb{R}^d \tag{121}$$

Given reparameterization flow $Z : \mathbb{R}^d \times [0,T] \xrightarrow{C^2} \mathbb{R}^d$*, satisfying* $Z(z,0) = z, \forall z \in \mathbb{R}^d$ *and* $Z(0,s) = 0, \forall s \in [0,T]$*. Let* $F : C([0,T];\mathbb{R}^d) \to \mathbb{R}^d$ *be a Frechet-differentiable functional, we have*

$$\nabla_x \mathbb{E}[F(X)|X_0 = x] = \mathbb{E}[\nabla_z F(X + Z(z,\cdot))|_{z=0} \tag{122}$$

$$+ F(X) \exp \int_0^T (\nabla_z Z(z,s)|_{z=0} \nabla_x b(X_s^x, s) - \nabla_z \partial_s Z(z,s)|_{z=0})(\sigma(X_s^x, s)^{-1})^T dW_t] \tag{123}$$

where we use capital X *without time subscript and* $Z(z,\cdot)$ *to denote one complete trajectory generated by the corresponding process.*

Remark 4.9. *The key idea of Girsanov reparameterization is similar to* $\frac{d}{dx} f(x)|_{x=x_0} = \frac{d}{dt} f(x_0 + t)|_{t=0}$*, that is, shift the gradient of* x *to a new parameter* t*. Integrating by time,* t *becomes a family of reparameterization* C^2 *section* $Z(z,s)$*, which connects to bundle theory and Lie algebra in differential geometry. We will further develop this idea in sections later.*

Proof. Observe that

$$\nabla_x \mathbb{E}[F(X)|X_0 = x] = \nabla_z \mathbb{E}[F(X)|X_0 = x + z]|_{z=0}. \tag{124}$$

Our next target is to shift the perturbed process X_t to its original position to separate the effect of perturbation, which is exactly the reparameterization component.

Define shifted process

$$dX_s^{x+z} := b(X_s^{x+z}, s)ds + \sigma(X_t^{x+z}, t)dW_t, X_0^{x+z} = x + z, \tag{125}$$

and shifted-back process

$$d\check{X}_s^x := [b(\check{X}_s^x + Z(z,s), s) - \partial_s Z(z,s)]ds + \sigma(\check{X}_s^x + Z(z,s))dW_t, d\check{X}_0^x = x \tag{126}$$

We have

$$d(\check{X}_s^x + Z(z,s)) = b(\check{X}_s^x + Z(z,s), s)ds + \sigma(\check{X}_s^x + Z(z,s), s)dW_t, \check{X}_0^x + Z(z,0) = x + z, \tag{127}$$

by the uniqueness of the strong solution[28, Theorem 5.2.1], this formula explicitly reparameterizes the shifted process for any $z \in \mathbb{R}^d$. Thus

$$\mathbb{E}[F(X)|X_0 = x + z] = \mathbb{E}[F(X^{x+z})] \tag{128}$$

$$= \mathbb{E}[F(\check{X}^x + Z(z,\cdot))] \tag{129}$$

$$= \mathbb{E}[F(X + Z(z,\cdot)) \exp(\int_0^T \sigma(X_s^x, s)^{-1}(b(X_s^x + Z(z,s), s) - b(X_s^x, s) - \partial_s Z(z,s))dW_s \tag{130}$$

$$- \frac{1}{2} \int_0^T |\sigma(X_s^x, s)^{-1}(b(X_s^x + Z(z,s), s) - b(X_s^x, s) - \partial_s Z(z,s))|^2 ds)]. \tag{131}$$

Plug it in the representation of the gradient

$$\nabla_z \mathbb{E}[F(X)|X_0 = x + z] = \nabla_z \mathbb{E}[F(X + Z(z,\cdot)) \exp(\int_0^T \sigma(X_s^x, s)^{-1}(b(X_s^x + Z(z,s), s) - b(X_s^x, s) - \partial_s Z(z,s))dW_s \tag{132}$$

$$- \frac{1}{2} \int_0^T |\sigma(X_s^x, s)^{-1}(b(X_s^x + Z(z,s), s) - b(X_s^x, s) - \partial_s Z(z,s))|^2 ds)] \tag{133}$$

We have

$$\nabla_z \mathbb{E}[F(X)|X_0 = x+z] = \mathbb{E}[\nabla_z F(X+Z(z,\cdot)) \exp(\int_0^T \sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))dW_s \tag{134}$$

$$- \frac{1}{2}\int_0^T |\sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))|^2 ds) \tag{135}$$

$$+ F(X+Z(z,\cdot)) \exp(\int_0^T \sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))dW_s \tag{136}$$

$$- \frac{1}{2}\int_0^T |\sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))|^2 ds) \tag{137}$$

$$\cdot (\nabla_z \int_0^T \sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))dW_s \tag{138}$$

$$- \frac{1}{2}\nabla_z \int_0^T |\sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))|^2 ds)] \tag{139}$$

Notice

$$\nabla_z \int_0^T |\sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))|^2 ds|_{z=0} \tag{140}$$

$$= \int_0^T \nabla_z |\sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))|^2|_{z=0} ds \tag{141}$$

$$= \int_0^T 2(b(X_s^x+Z(0,s),s) - b(X_s^x,s) - \partial_s Z(0,s))^T (\sigma(X_s^x,s)^{-1})^T \nabla_z(\cdots)|_{z=0} ds \tag{142}$$

$$= 0, \tag{143}$$

and

$$\nabla_z \int_0^T \sigma(X_s^x,s)^{-1}(b(X_s^x+Z(z,s),s) - b(X_s^x,s) - \partial_s Z(z,s))dW_s \tag{144}$$

$$= \int_0^T (\nabla_z Z(z,s)|_{z=0} \nabla_x b(X_s^x,s) - \nabla_z \partial_s Z(z,s)|_{z=0})(\sigma(X_s^x,s)^{-1})^T dW_t \tag{145}$$

Plug them in, we have

$$\nabla_x \mathbb{E}[F(X)|X_0 = x] = \nabla_z \mathbb{E}[F(X)|X_0 = x+z]|_{z=0} \tag{146}$$

$$= \mathbb{E}[\nabla_z F(X+Z(z,\cdot))|_{z=0} \tag{147}$$

$$+ F(X) \exp \int_0^T (\nabla_z Z(z,s)|_{z=0} \nabla_x b(X_s^x,s) - \nabla_z \partial_s Z(z,s)|_{z=0})(\sigma(X_s^x,s)^{-1})^T dW_t], \tag{148}$$

which completes the proof. □

Apply this formula on the $\frac{\partial J}{\partial y}$, we have

$$J(t,y;\pi_F^\theta) = \mathbb{E}_{\mathbb{H}}\left[\int_t^T \left[\frac{1}{4\beta}|v_s|^2 - \gamma \log \pi(v_s|Y_s)\right] \mathrm{d}s + g(Y_T)\middle| Y_t = y\right]$$

$$\Longrightarrow \frac{\partial J}{\partial y}(t,y;\pi_F^\theta) = \frac{\partial}{\partial y}\mathbb{E}_{\pi_F^\theta}[\mathbb{E}_{\mathbb{Q}}[\int_t^T [\frac{1}{4\beta}|v_s|^2 - \gamma \log \pi(v_s|Y_s)]\mathrm{d}s + g(Y_T)\Big| Y_t = y]]$$

$$= \mathbb{E}_{\pi_F^\theta}[\nabla_y \mathbb{E}_{\mathbb{Q}}[\int_t^T [\frac{1}{4\beta}|v_s|^2 - \gamma \log \pi(v_s|Y_s)]\mathrm{d}s + g(Y_T)\Big| Y_t = y]].$$

From reparameterization trick

$$\nabla_z \int_t^T [\frac{1}{4\beta}|\mu_\theta(Y_s + Z(z,s)) + \sigma_\theta(Y_s + Z(z,s))\xi|^2 - \gamma(-\frac{d}{2}\log(2\pi h)) - d\log\sigma_\theta(Y_s + Z(z,s)) - \frac{1}{2}\xi^T\xi]ds \tag{149}$$

$$+ \nabla_z g(Y_T + Z(z,T)) \tag{150}$$

$$= \int_t^T (\frac{1}{2\beta}(\mu_\theta(Y_s + Z(z,s)) + \sigma_\theta(Y_s + Z(z,s))\xi)^T (\frac{\partial \mu_\theta}{\partial y}(Y_s + Z(z,s))\frac{\partial Z}{\partial z}(z,s) \tag{151}$$

$$+ \xi\frac{\partial \sigma_\theta}{\partial y}(Y_s + Z(z,s))\frac{\partial Z}{\partial z}(z,s)) + \frac{d\gamma}{\sigma_\theta(Y_s + Z(z,s))}\frac{\partial \sigma_\theta}{\partial y}(Y_s + Z(z,s))\frac{\partial Z}{\partial z}(z,s))ds \tag{152}$$

$$+ \frac{\partial g}{\partial y}(Y_T + Z(z,T))\frac{\partial Z}{\partial z}(z,T). \tag{153}$$

Set $z = 0$, we have

$$\nabla_z \int_t^T [\frac{1}{4\beta}|\mu_\theta(Y_s + Z(z,s)) + \sigma_\theta(Y_s + Z(z,s))\xi|^2 - \gamma(-\frac{d}{2}\log(2\pi h)) - d\log\sigma_\theta(Y_s + Z(z,s)) - \frac{1}{2}\xi^T\xi]ds \tag{154}$$

$$+ \nabla_z g(Y_T + Z(z,T))\Big|_{z=0} \tag{155}$$

$$= \int_t^T (\frac{1}{2\beta}(\mu_\theta(Y_s) + \sigma_\theta(Y_s)\xi)^T (\frac{\partial \mu_\theta}{\partial y}(Y_s)\frac{\partial Z}{\partial z}(0,s) \tag{156}$$

$$+ \xi\frac{\partial \sigma_\theta}{\partial y}(Y_s)\frac{\partial Z}{\partial z}(0,s)) + \frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s)\frac{\partial Z}{\partial z}(0,s))ds \tag{157}$$

$$+ \frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T) \tag{158}$$

$$= \int_t^T (\frac{1}{2\beta}(\mu_\theta(Y_s) + \sigma_\theta(Y_s)\xi)^T \frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi\frac{\partial \sigma_\theta}{\partial y}(Y_s) + \frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s))\frac{\partial Z}{\partial z}(0,s)ds \tag{159}$$

$$+ \frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T) \tag{160}$$

The second term is easier to compute

$$(\int_t^T (\frac{1}{4\beta}|\mu_\theta(Y_s) + \sigma_\theta(Y_s)\xi|^2 - \gamma(-\frac{d}{2}\log(2\pi h)) - d\log\sigma_\theta(Y_s) - \frac{1}{2}\xi^T\xi)ds \tag{161}$$

$$+ g(Y_T))\exp\int_0^T (\frac{\partial Z}{\partial z}(0,s)(-\nabla^2 V(Y_s) + 2\beta(\frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi^T\frac{\partial \sigma_\theta}{\partial y}(Y_s))) - \frac{\partial^2 Z}{\partial z \partial s}(0,s))dW_t \tag{162}$$

Combining all of them

$$\nabla_y \mathbb{E}_{\mathbb{Q}}[\int_t^T [\frac{1}{4\beta}|v_s|^2 - \gamma\log\pi(v_s|Y_s)]\mathrm{d}s + g(Y_T)\Big|Y_t = y] \tag{163}$$

$$= \mathbb{E}_{\mathbb{Q}}[\int_t^T (\frac{1}{2\beta}(\mu_\theta(Y_s) + \sigma_\theta(Y_s)\xi)^T \frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi\frac{\partial \sigma_\theta}{\partial y}(Y_s) + \frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s))\frac{\partial Z}{\partial z}(0,s)ds \tag{164}$$

$$+ \frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T) + (\int_t^T (\frac{1}{4\beta}|\mu_\theta(Y_s) + \sigma_\theta(Y_s)\xi|^2 - \gamma(-\frac{d}{2}\log(2\pi h)) - d\log\sigma_\theta(Y_s) - \frac{1}{2}\xi^T\xi)ds \tag{165}$$

$$+ g(Y_T))\exp\int_0^T (\frac{\partial Z}{\partial z}(0,s)(-\nabla^2 V(Y_s) + 2\beta(\frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi^T\frac{\partial \sigma_\theta}{\partial y}(Y_s))) - \frac{\partial^2 Z}{\partial z \partial s}(0,s))dW_t\Big|Y_t = y] \tag{166}$$

Thus

$$\frac{\partial J}{\partial y}(t,y;\pi_F^\theta) =\mathbb{E}_{\mathbb{H}}[\int_t^T (\frac{1}{2\beta}(\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi)^T\frac{\partial \mu_\theta}{\partial y}(Y_s)+\xi\frac{\partial \sigma_\theta}{\partial y}(Y_s)+\frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s))\frac{\partial Z}{\partial z}(0,s)ds \tag{167}$$

$$+\frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T)+(\int_t^T(\frac{1}{4\beta}|\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi|^2-\gamma(-\frac{d}{2}\log(2\pi h))-d\log\sigma_\theta(Y_s)-\frac{1}{2}\xi^T\xi)ds \tag{168}$$

$$+g(Y_T))\exp\int_0^T(\frac{\partial Z}{\partial z}(0,s)(-\nabla^2 V(Y_s)+2\beta(\frac{\partial \mu_\theta}{\partial y}(Y_s)+\xi^T\frac{\partial \sigma_\theta}{\partial y}(Y_s)))-\frac{\partial^2 Z}{\partial z\partial s}(0,s))dW_t\Big|Y_t=y] \tag{169}$$

Suppose the reparameterization section Z is already selected. To distinguish the original expression of $\frac{\partial J}{\partial y}$ from its reparameterized form, we denote the latter one $L_Z J$, and parameterize it as $L_Z^\psi J$(this notation is taken from that of Lie derivative in differential geometry).

Consider L_2 loss

$$\mathcal{L}_\psi(t,y,\theta):=|L_Z^\psi J(t,y,\theta)-L_Z J(t,y;\pi)|^2 \tag{170}$$

$$\leq \mathbb{E}_{\mathbb{H}}|L_Z^\psi J(t,y,\theta)-\int_t^T(\frac{1}{2\beta}(\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi)^T\frac{\partial \mu_\theta}{\partial y}(Y_s)+\xi\frac{\partial \sigma_\theta}{\partial y}(Y_s)+\frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s))\frac{\partial Z}{\partial z}(0,s)ds \tag{171}$$

$$+\frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T)+(\int_t^T(\frac{1}{4\beta}|\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi|^2-\gamma(-\frac{d}{2}\log(2\pi h))-d\log\sigma_\theta(Y_s)-\frac{1}{2}\xi^T\xi)ds \tag{172}$$

$$+g(Y_T))\exp\int_0^T(\frac{\partial Z}{\partial z}(0,s)(-\nabla^2 V(Y_s)+2\beta(\frac{\partial \mu_\theta}{\partial y}(Y_s)+\xi^T\frac{\partial \sigma_\theta}{\partial y}(Y_s)))-\frac{\partial^2 Z}{\partial z\partial s}(0,s))dW_t|^2, \tag{173}$$

which can be computed by simulation in online setting or by replaying in offline setting. The expectation $\mathbb{E}_{\mathbb{H}}$ can be approximated batch-wise, and we will denote the corresponding estimator $\hat{\mathcal{L}}_\psi(t,y,\theta)$.

Remark 4.10. *Note that the value function does not depend on any specific v, but on the average behavior of π along the trajectories sampled.*

This gives us the Hamiltonian on approximated $L_Z J$

$$H(t,y,v,L_Z^\psi J(t,y,\theta),\nabla_y L_Z^\psi J(t,y,\theta))=\left[-\nabla V(y)+2\beta v\right]\cdot L_Z^\psi J(t,y,\theta)+\beta\cdot\nabla_y L_Z^\psi J(t,y,\theta)+\frac{1}{4\beta}|v|^2, \tag{174}$$

and the optimization step of policy becomes

$$\theta\leftarrow\theta-[\log\pi_F^\theta(v_t|t,Y_t)-\frac{1}{\gamma}H(t,Y_t,v_t,L_Z^\psi J(t,Y_t,\theta),\nabla_y L_Z^\psi J(t,Y_t,\theta))]\frac{\partial}{\partial\theta}(\log\pi_F^\theta(v_t|t,Y_t))^T \tag{175}$$

Remark 4.11. *The advantage of Girsanov reparameterization is that it does not involve derivatives of the initial condition, which effectively avoids costly backpropagation along paths when the network is on a large scale, at the expense of affecting the variance of the result by reparameterization section Z[10, Proposition 2,Theorem 2], a new issue for achieving lower mean square error. We will tackle this problem by injecting additional physics information via principal bundle theory.*

Following the structure of its discretized counterparts in algorithm 1, we derive continuous time Soft Actor Crictic without any discretization step.

Algorithm 3 Continuous Soft Actor-Critic (Conti-SAC) algorithm for SOC problem

Initialize policy parameters θ, Hamiltonian parameters ψ_1, ψ_2, empty replay buffer $\mathcal{D}$, tradeoff constant γ, target parameter $\psi_{targ,1}\leftarrow\psi_1,\psi_{targ,2}\leftarrow\psi_2$, smoothing parameter ρ, and selected reparameterization section Z.

repeat

Observe state Y_t and select action v_t from distribution $\pi_F^\theta(\cdot|Y_t)$ throughout $0\leq t\leq T$.

Store trajectories $(Y_t,v_t)_{0\leq t\leq T}$ in replay buffer $\mathcal{D}$.

if it's time to update **then**

for each gradient step **do**

Randomly sample a batch B of trajectories $(Y_t, v_t)_{0\le t\le T}$ from $\mathcal{D}$.

For $t_j \sim Unif[0,T], 1 \le j \le |B|$, compute the targets for $L_Z^\psi J$

$$
\begin{aligned}
y_L^{t_j} := &\int_{t_j}^T (\frac{1}{2\beta}(\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi)^T \frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi \frac{\partial \sigma_\theta}{\partial y}(Y_s) + \frac{d\gamma}{\sigma_\theta(Y_s)}\frac{\partial \sigma_\theta}{\partial y}(Y_s))\frac{\partial Z}{\partial z}(0,s)ds \\
&+ \frac{\partial g}{\partial y}(Y_T)\frac{\partial Z}{\partial z}(0,T) + (\int_{t_j}^T (\frac{1}{4\beta}|\mu_\theta(Y_s)+\sigma_\theta(Y_s)\xi|^2 - \gamma(-\frac{d}{2}\log(2\pi h)) - d\log\sigma_\theta(Y_s) - \frac{1}{2}\xi^T\xi)ds \\
&+ g(Y_T))\exp\int_0^T (\frac{\partial Z}{\partial z}(0,s)(-\nabla^2 V(Y_s) + 2\beta(\frac{\partial \mu_\theta}{\partial y}(Y_s) + \xi^T \frac{\partial \sigma_\theta}{\partial y}(Y_s))) - \frac{\partial^2 Z}{\partial z \partial s}(0,s))dW_t
\end{aligned}
$$

Update $L_Z^\psi J$ by one step of gradient descent

$$
\nabla_{\psi_i}\hat{\mathcal{L}}_{\psi_i} = \nabla_{\psi_i}\frac{1}{|B|}\sum_{1\le j\le |B|}\left(L_Z^{\psi_i}(Y_t,v_t,\theta) - y_L^{t_j}\right)^2, \text{ for } i\in\{1,2\}.
$$

Compute Hamiltonian and update policy by one step of gradient descent

$$
\theta \leftarrow \theta - [\log\pi_F^\theta(v_t|t,Y_t) - \frac{1}{\gamma}H(t,Y_t,v_t,L_Z^\psi J(t,Y_t,\theta),\nabla_y L_Z^\psi J(t,Y_t,\theta))]\frac{\partial}{\partial\theta}(\log\pi_F^\theta(v_t|t,Y_t))^T \tag{176}
$$

where the reparametrization trick is used, that is

$$
v_t = \mu_\theta(Y_t) + \sigma_\theta(Y_t)\cdot\xi, \quad \xi\sim\mathcal{N}(0,\sqrt{h}I_d). \tag{177}
$$

Update target network

$$
\psi_{\text{targ},i} \leftarrow \rho\psi_{\text{targ},i} + (1-\rho)\psi_i \quad \text{for} \quad i=1,2 \tag{178}
$$

end for

end if

until convergence

4.2.2 Continuous Generalization of GFlowNet

[**Yiping:** GFlowNet is sampling over distribution but not maximizing the reward, how does this relat to transition path?] [**Zexi:** Reward is a unnormalized target distribution. It suffices to add a learnable partition.]

[**Yiping:** what is θ, ϖ in the Transition Path setting] [**Zexi:** θ is still the parameter of v_F^θ, whilst ϖ is the parameter of score. These notations have already appeared before. Note that if I did not define them explicitly then they coincide with their definitions in previous sections.] We could consider two generalizations to generalize GFlowNet to continuous adaptation: evaluate the differences in measure along given paths through the Girsanov theorem, a natural extension of trajectory balance to continuous cases, or directly match the flow via L_2 distance through flow matching.

Let use begin with trajectory balance situation. In discrete case, for fixed starting point, the trajectory balance loss of GFN is given by

$$
\mathcal{L}_{\text{TB}}(\theta,\varpi,\chi) = \left(\log\frac{\chi\prod_{t=0}^{T-1}P_F(Y_{t+\Delta t}|Y_t;\theta)}{R(Y_T)\prod_{\tilde{t}=0}^{T-1}P_B(Y_{\tilde{t}+\Delta\tilde{t}}|Y_{\tilde{t}};\theta,\varpi)}\right)^2 \tag{179}
$$

[**Yiping:** what is χ] [**Zexi:** learnable partition] [**Yiping:** then it should be $\mathcal{L}_{\text{TB}}(\theta,\varpi,\chi)$] where P_F and P_B are given by the transition probability of the SDE system representing the sampling strategy, $R(Y_T)$ is the reaward, satisfying

$$
R(Y_T) = \frac{1}{4\beta}\sum_{t=0}^{T-1}\Delta t|v_t|^2 + g(Y_T) \tag{180}
$$

and χ is the normalizing constant of R to be learnt.

Consider

$$\log \frac{\chi \prod_{t=0}^{T-1} P_F(Y_{t+\Delta t}|Y_t;\theta)}{R(Y_T)\prod_{\tilde{t}=0}^{T-1} P_B(Y_{\tilde{t}+\Delta\tilde{t}}|Y_{\tilde{t}};\theta,\varpi)} \tag{181}$$

$$= \log \frac{\chi \cdot \exp(\sum_{t=0}^{T-1} \log P_F(Y_{t+\Delta t}|Y_t;\theta))}{R(Y_T)\cdot \exp(\sum_{\tilde{t}=0}^{T-1} \log P_B(Y_{\tilde{t}+\Delta\tilde{t}}|Y_{\tilde{t}};\theta,\varpi))}. \tag{182}$$

[**Yiping:** the reward here only depend on Y_T but also the trajectory. Why the GFN theory can still be applied?] [**Zexi:** Theoretically not. But empirically true sometimes. See the paper I have cited in discretized GFN solver for control problems.]

Notice that $\tilde{t} = T - t$, we have

$$dY_{T-t} = [-\nabla V(Y_{T-t}) + 2\beta v_{T-t} - 2\beta s_\varpi(v_{T-t}, Y_{T-t}, T-t)]d(T-t) + \sqrt{2\beta}dW_{T-t} \tag{183}$$

$$\implies dY_{T-t} = [\nabla V(Y_{T-t}) - 2\beta v_{T-t} + 2\beta s_\varpi(v_{T-t}, Y_{T-t}, T-t)]dt + \sqrt{2\beta}dW_{T-t} \tag{184}$$

followed by

$$\log\chi - \log R(Y_T) + \sum_{t=0}^{T-1} \log \frac{P_F(Y_{t+\Delta t}|Y_t;\theta)}{P_B(Y_t|Y_{t+\Delta t};\theta,\varpi)} \tag{185}$$

$$= \log\chi - \log R(Y_T) - \sum_{t=0}^{T-1} \log \frac{P_B(Y_t|Y_{t+\Delta t};\theta,\varpi)}{P_F(Y_{t+\Delta t}|Y_t;\theta)} \tag{186}$$

Notice, when we perform gradient descent, the $Y_t(\omega)$ here are specified trajectory in the replay buffer.

Taking limit, the equation above correspond to

$$\mathcal{L}_{TB} := \mathbb{E}_{\mathbb{H}}(\log\chi - \log \frac{d\mathbb{M}}{d\mathbb{Q}}(Y(\omega)) - \log R(Y_T))^2 \tag{187}$$

$$\text{where: } dY_t = [-\nabla V(Y_t) + 2\beta v_t]dt + \sqrt{2\beta}dW_t \sim \mathbb{Q}, \tag{188}$$

$$dY_{T-t} = [\nabla V(Y_{T-t}) - 2\beta v_{T-t} + 2\beta s^{(1)}_\varpi(v_{T-t}, Y_{T-t}, T-t)]dt + \sqrt{2\beta}dW_{T-t} \sim \mathbb{M}. \tag{189}$$

Suppose a trajectory $(Y_t, v_t), 0 \le t \le T$ is taken from the replay buffer, we have

$$\frac{d\mathbb{M}}{d\mathbb{Q}} = \exp(-\int_0^T \frac{2\nabla V(Y_t) - 4\beta v_t + 2\beta s^{(1)}_\varpi(v_t, Y_t, T-t)}{\sqrt{2\beta}} dW_s \tag{190}$$

$$- \frac{1}{2}\int_0^T \frac{(2\nabla V(Y_t) - 4\beta v_t + 2\beta s^{(1)}_\varpi(v_t, Y_t, T-t))^2}{2\beta} ds), \tag{191}$$

from Girsanov's theorem.

Finally, we have

$$\mathcal{L}_{TB} = \mathbb{E}_{\mathbb{H}}(\log\chi + \int_0^T \frac{2\nabla V(Y_t) - 4\beta v_t + 2\beta s^{(1)}_\varpi(v_t, Y_t, T-t)}{\sqrt{2\beta}} dW_s \tag{192}$$

$$+ \frac{1}{2}\int_0^T \frac{(2\nabla V(Y_t) - 4\beta v_t + 2\beta s^{(1)}_\varpi(v_t, Y_t, T-t))^2}{2\beta} ds - \log R(Y_T))^2. \tag{193}$$

Remark 4.12. *Although we use experience replay in offline training, we do not detach v_F^θ from that graph,that is, θ, ϖ, χ are updated simultaneously.*

The flow matching loss is easier to compute, but requires additional backward trajectory samples.

$$\mathcal{L}_{FM} = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{\mathbb{H}}|Y_t^{forward} - Y_t^{backward}|^2. \tag{194}$$

We present the pseudocode for continuous GFN under the flow matching case.

Algorithm 4 Continuous Generative Flow Network (Conti-GFN) algorithm for SOC problem(flow matching version)

Initialize GFlowNet θ,ϖ , partition function χ and empty replay buffer $\mathcal{D}$.
repeat
  Collect a set of paths $(Y_t^{forward})_{0\leq t\leq T}$ by interaction with forward policy v_F^θ.
  Collect a set of paths $(Y_t^{backward})_{0\leq t\leq T}$ by interaction with backward policy v_F^θ, s_ϖ.
  if it's time to update **then**
    Get mini-batch $\mathcal{B}$ sampled from $\mathcal{D}$
    Sample a batch of update time $t \sim [0, T]$
    Update GFlowNet

$$\hat{\mathcal{L}}_{FM} = \frac{1}{|B|}\sum_{k=1}^{|B|} |Y_t^{forward,k} - Y_t^{backward,k}|^2. \tag{195}$$

  end if
until convergence

5 Variance Reduction via Principal Bundle Theory

In this section, our main focus lies on how to select a reparameterization flow $Z(z,t)$ that fully leverages physics information available by principal bundle theory. We present our "physics-informed reparameterization" beginning with an overview of equivariant flow in section 5.1, then we delve into the relationship between reparameterization and variance in section 5.2, and ultimately, we outline our innovative reparameterization method grounded in principal bundle theory in 5.3.

5.1 Equivariant Flow

References

[1] Josh Achiam. Sac. https://github.com/openai/spinningup/blob/master/docs/algorithms/sac.rst, January 2020. Accessed on May 3, 2024.

[2] D. Aristoff, T. Lelièvre, C.G Mayne, and I. Teo. Adaptive multilevel splitting in molecular dynamics simulations. *ESAIM: Proceedings and Surveys*, 48:215–225, 2015.

[3] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2023.

[4] N. Berglund. Kramers' law: Validity, derivations and generalisations. *Markov Processes and Related fields*, 19(3):459–490, 2013.

[5] J. A. Bucklew. *Introduction to Rare Event Simulation*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[6] F. Cérou, A. Guyader, T. Lelièvre, and D. Pommier. A multiple replica approach to simulate reactive trajectories. *J. Chem. Phys.*, 134(5):054108, 2011.

[7] F. Cérou, A. Guyader, and M. Rousset. Adaptive Multilevel Splitting: Historical perspective and recent results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(4):043108, 2019.

[8] F. Cérou and A. Guyader. Adaptive Multilevel Splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007.

[9] Martin V Day. Conditional exits for small noise diffusions with characteristic boundary. *The Annals of Probability*, pages 1385–1419, 1992.

[10] Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky T. Q. Chen. Stochastic optimal control matching, 2024.

[11] Chenru Duan, Guan-Horng Liu, Yuanqi Du, Tianrong Chen, Qiyuan Zhao, Haojun Jia, Carla P. Gomes, Evangelos A. Theodorou, and Heather J. Kulik. React-ot: Optimal transport for generating transition state in chemical reactions, 2024.

[12] Weinan E and Eric Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual review of physical chemistry*, 61:391–420, 2010.

[13] W. H. Fleming. Exit probabilities and optimal stochastic control. *Appl. Math. Optim.*, 4(4):329–346, 1977/78.

[14] E. Fournié, J.-M. Lasry, P.-L. Lions, J. Lebuchoux, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics*, 3(4):391–412, 1999.

[15] Yuan Gao, Jian-Guo Liu, and Oliver Tse. Optimal control formulation of transition path problems for markov jump processes, 2023.

[16] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2019.

[17] Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Bernd Ensing, and Max Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths, 2023.

[18] Xinru Hua, Rasool Ahmad, Jose Blanchet, and Wei Cai. Accelerated sampling of rare events using a neural network bias potential, 2024.

[19] Moksh Jain, Emmanuel Bengio, Alex-Hernandez Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ekbote, Jie Fu, Tianyu Zhang, Micheal Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with gflownets, 2023.

[20] Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61, 2023.

[21] Elaine Lau, Stephen Zhewen Lu, Ling Pan, Doina Precup, and Emmanuel Bengio. Qgfn: Controllable greediness with action values, 2024.

[22] T. Lelièvre and G. Stoltz. Partial differential equations and stochastic methods in molecular dynamics. *Acta Numerica*, 25:681–880, 2016.

[23] Tony Lelièvre, Geneviève Robin, Inass Sekkat, Gabriel Stoltz, and Gabriel Victorino Cardoso. Generative methods for sampling transition paths in molecular dynamics, 2023.

[24] Yinchuan Li, Shuang Luo, Haozhi Wang, and Jianye Hao. Cflownets: Continuous control with generative flow networks, 2023.

[25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[26] Jianfeng Lu and James Nolen. Reactive trajectories and the transition path process. *Probability Theory and Related Fields*, 161(1-2):195–244, 2015.

[27] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets, 2023.

[28] B. Øksendal. *Stochastic Differential Equations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[29] Michael Plainer, Hannes Stärk, Charlotte Bunne, and Stephan Günnemann. Transition path sampling with boltzmann generator-based mcmc moves, 2023.

[30] J. Quer, L. Donati, B. G. Keller, and M. Weber. An automatic adaptive importance sampling algorithm for molecular dynamics in reaction coordinates. *SIAM Journal on Scientific Computing*, 40(2):A653–A670, 2018.

[31] Jannes Quer and Enric Ribera Borrell. Connecting stochastic optimal control and reinforcement learning, 2024.

[32] Lior Shani, Yonathan Efroni, and Shie Mannor. Exploration conscious reinforcement learning revisited, 2019.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[34] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

[35] Luke Triplett and Jianfeng Lu. Diffusion methods for generating transition paths, 2023.

[36] Iñigo Urteaga and Chris H. Wiggins. Bayesian bandits: balancing the exploration-exploitation tradeoff via double sampling, 2018.

[37] Eric Vanden-Eijnden et al. Towards a theory of transition paths. *Journal of statistical physics*, 123(3):503–523, 2006.

[38] Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 2012.

[39] Jiaxin Yuan, Amar Shah, Channing Bentz, and Maria Cameron. Optimal control for sampling the transition path process and estimating rates, 2023.

[40] Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial optimization problems with gflownets, 2023.