Fisher Adjoint Matching: Natural Gradients for Stochastic Optimal Control

Mayank Shrivastava
University of Illinois Urbana-Champaign
mayanks4@illinois.edu

Chandni Nagda
University of Illinois Urbana-Champaign
cnagda2@illinois.edu

Rohan Deb
University of Illinois Urbana-Champaign
rd22@illinois.edu

Arindam Banerjee
University of Illinois Urbana-Champaign
arindamb@illinois.edu

March 8, 2026

Abstract

Stochastic Optimal Control (SOC) problems are becoming increasingly important in machine learning, with applications in reward fine-tuning of generative models and model-based reinforcement learning (RL). A recent approach, Adjoint Matching (AM), reformulates quadratic-cost SOC as a regression problem, avoiding the high-variance importance weights of prior approaches. However, we show that the AM update performs descent on a surrogate of the SOC objective; consequently, unconstrained AM steps need not improve the SOC objective monotonically. This leads to unstable optimization dynamics. We then demonstrate that monotonic descent is guaranteed when each update is restricted to a KL trust region between successive controlled path measures. Motivated by this observation, we propose Fisher Adjoint Matching (FAM). FAM tractably approximates the KL-constrained update by replacing the trust-region problem with a Fisher-preconditioned step under the current path measure. The resulting update follows the steepest-descent direction in KL geometry (a natural-gradient step), yielding stable and efficient optimization. Finally, we empirically demonstrate the improved stability and sample efficiency of FAM relative to AM on classical control benchmarks and generative modeling tasks, underscoring the need to control distributional shift in fixed-point SOC.

1 Introduction

In many scientific and engineering applications, the evolution of a dynamical system is modeled as a stochastic process $\{X_t\}_{t\geq 0}$ governed by a stochastic differential equation (SDE), where the state evolves through a deterministic drift together with stochastic fluctuations, typically modelled as Brownian noise. Stochastic optimal control (SOC) studies how to steer such noisy dynamics to minimize a prescribed cost functional, and it has a long history of applications across science and engineering: rare-event simulation and molecular dynamics [21, 22, 50, 25], finance and economics [33, 16], stochastic filtering and data assimilation [31, 36], nonconvex optimization via PDE viewpoints [8], energy systems and markets [4, 35], and robotics [43, 18]. SOC has also shaped neighboring areas such as mean-field games [7], optimal transport [45, 46], backward stochastic differential equations (BSDEs) [6], and large deviations [14].

In machine learning, continuous-time stochastic process formalisms now play a central role in both modeling and algorithms. Deep generative modeling can be phrased in terms of probability transport and time-indexed stochastic dynamics, spanning normalizing flows [37], score-based/diffusion models [40, 24, 41], flow matching [29], and stochastic interpolants [1, 2]. Additionally SOC viewpoint has recently become an

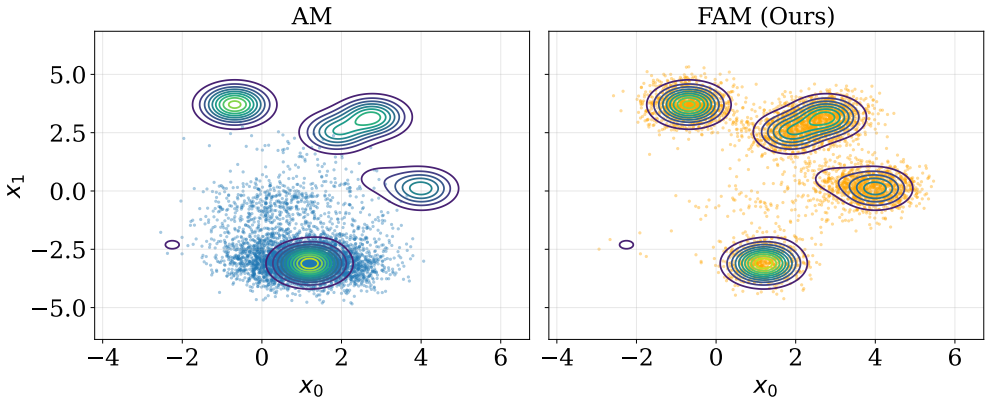


Figure 1: Qualitative comparison on a 2D GMM target. Contours show the ground truth density; dots are samples at iteration 500. AM exhibits slow convergence, while FAM successfully captures all modes after less iterations. Left: AM baseline. Right: FAM (ours).

explicit algorithmic primitive for sampling from unnormalized densities and for improving diffusion samplers [49, 44, 5, 38]. As such, it is natural to seek learning-based methods that can reliably optimize controls and solve the associated high-dimensional SOC problems.

Formally, given an SDE, SOC studies the design of an additional *control drift* u that alters the dynamics to steer the resulting trajectories toward a prescribed objective (typically expressed as the minimization of an expected cost functional). In reinforcement learning, this perspective corresponds to choosing a policy (equivalently, a control law) to maximize expected cumulative reward [13, 28, 42]. In generative modeling, related control formulations have been used for reward-based fine tuning of flow matching models [47, 12].

Recently, learning-based approaches to SOC [11, 20, 12] have gained significant attention, where the control drift $u(X_t, t) \in \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is modeled as a parametric function u_θ. The parameters θ are optimized by simulating trajectories of the controlled stochastic process and minimizing an associated objective functional $\mathcal{L}(u_\theta)$. As in deep learning, gradient-based methods are commonly employed, with gradients computed by differentiating through the controlled trajectories. In continuous-time settings, this is typically accomplished via the *adjoint-sensitivity method* [34, 9], which evaluates gradients through a backward ordinary differential equation (ODE) while avoiding storage of the entire forward trajectory.

In this work, we focus on Adjoint Matching (AM) [12]. Earlier learning-based approaches to SOC regress directly to the optimal control by approximating expectations under the optimally controlled path measure using importance sampling from the uncontrolled (base) process. However, this suffers from high-variance importance weights, particularly in high-dimensional settings. In contrast, AM solves the following fixed point equation:

$$\theta^* = \operatorname*{argmin}_\theta \mathcal{L}_{\text{fp}}(\theta)$$
$$\mathcal{L}_{\text{fp}}(\theta) := \mathbb{E}_{\mathbb{P}^{u_\theta}} \left[\int \|u_\theta + \sigma^\top a_\theta\|^2 \right] \tag{1}$$

where a_θ is the adjoint under the current control. Under suitable regularity conditions, it can be shown the optimal control u^*, is the unique minimizer of the above objective (Lemma 6, [12]).

The starting point of this work is the observation that, AM solves (1) in an iterative manner and the AM objective at iteration i:

$$\theta_i = \operatorname*{argmin}_\theta \mathcal{L}_{\text{AM}}(\theta, \theta_{i-1})$$
$$\mathcal{L}_{\text{AM}}(\theta, \theta_{i-1}) := \mathbb{E}_{\mathbb{P}^{u_{\theta_{i-1}}}} \left[\int \|u_\theta + \sigma^\top a_{\theta_{i-1}}\|^2 \right] \tag{2}$$

is only a surrogate to the true objective (1). In fact, a gradient step on (2) simultaneously changes the objective and the distribution under which (1) is evaluated. As a result, taking a large step in the direction of the gradient of the surrogate can lead to non-monotonic behavior of the true objective. To remedy this, we propose **F**isher **A**djoint **M**atching (FAM), an iterative procedure for solving SOC with guaranteed monotonic improvement. Our contributions are fourfold:

1. We show that the AM update performs descent on a surrogate of the fixed-point objective, which *in general* does not guarantee a decrease of the underlying fixed-point objective.
2. We derive a modified AM objective that ensures monotonic decrease of the fixed-point objective by restricting each update to a KL trust region defined between successive path measures.
3. We propose Fisher Adjoint Matching (FAM), an efficient algorithm for solving the resulting KL-constrained problem via Fisher preconditioning of the gradient.
4. We empirically demonstrate improved stability and performance over Adjoint Matching on classical stochastic control benchmarks and generative sampling tasks.[1]

2 Preliminaries

Notation Vectors are denoted by X. A stochastic process is denoted by $\{X_t\}_{t\geq 0}$. The norm $\|\cdot\|$ refers to the Euclidean norm $\|\cdot\|_2$ unless stated otherwise. We use $\mathbb{P}$ to denote path measures, and p to denote probability density functions. $\mathbb{E}_{\mathbb{P}}[\cdot]$ is shorthand for $\mathbb{E}_{X\sim\mathbb{P}}[\cdot]$.

Problem Formulation We begin by describing the SOC problem. Consider a dynamical system described by a stochastic differential equation (SDE):

$$dX_t = b(X_t, t)\, dt + \sigma(t)\, dB_t, \qquad X_0 \sim p_0, \tag{3}$$

where $t \in [0,1]$, $b : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ denotes the drift, $\sigma : [0,1] \to \mathbb{R}^{d\times d}$ the diffusion coefficient, $(B_t)_{t\in[0,1]}$ a standard d-dimensional Brownian motion and p_0 is a known, fixed distribution. SOC introduces an additional *control drift* $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ to the *base process* (3) and the controlled dynamics are given by:

$$\begin{aligned} X_0^u &\sim p_0, \quad \text{and} \\ dX_t^u &= b(X_t^u, t)\, dt + \sigma(t) u(X_t^u, t)\, dt + \sigma(t)\, dB_t. \end{aligned} \tag{4}$$

Given a running cost $f(X_t, t) : \mathbb{R}^d \times [0,1] \to \mathbb{R}$ and a terminal cost $g(X_1) : \mathbb{R}^d \to \mathbb{R}$, the goal of quadratic control-cost SOC is to find the optimal control u^* that minimizes the cost functional:

$$\mathcal{L}(u) = \mathbb{E}_{\mathbb{P}^u}\left[\int_0^1 \left(\tfrac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\right) dt + g(X_1^u)\right], \tag{5}$$

where $\mathbb{P}^u$ denotes the path measure induced by the controlled SDE.

In parametric approaches to SOC [11, 12, 32], control u is restricted to a parameterized family $u_\theta(X_t, t)$, where $\theta \in \mathbb{R}^p$ denotes trainable parameters. Thus, the minimization problem is given as :

$$\theta^* = \operatorname{argmin}_\theta \mathcal{L}(u_\theta) \tag{6}$$

In practice, u_θ is implemented as a neural network. Next we describe Adjoint Matching [12], a recent method for solving the resulting parametric control problem.

Algorithm 1 Adjoint Matching

Require: Drift b, diffusion σ, state cost $f(\cdot)$, terminal cost $g(\cdot)$, initial distribution p_0, parametric control u_θ, step size h, number of time steps K, batch size N, KL threshold `maxkl`, Number of training iterations T

for $i = 0, \cdots, T-1$ **do**

Sample N trajectories $\{X_k\}_{k=1}^N$ with Euler–Maruyama on $t_j = jh$, $j = 0, \ldots, K-1$:

$$X_{k,0} \sim p_0\ , \quad \varepsilon_{k,j} \sim \mathcal{N}(0, I)$$
$$X_{k,j+1} \leftarrow X_{k,j} + \big(b(X_{k,j}, t_j) + \sigma(t_j) u_\theta(X_{k,j}, t_j)\big)h + \sigma(t_j)\sqrt{h}\,\varepsilon_{k,j}, \quad \forall k, j.$$

Compute adjoints via backward Euler discretization of (11):

$$a_{k,K} \leftarrow \nabla g(X_{k,K})$$
$$a_{k,j} \leftarrow a_{k,j+1} - \big((\nabla_x b(X_{k,j}, t_j))^\top a_{k,j+1} + \nabla_x f(X_{k,j}, t_j)\big)h, \quad \forall j = K-1, \ldots, 0.$$

Compute gradient $\nabla_\theta \mathcal{L}(\theta, \theta_i)$ using the AM loss (13) and update:

$$\theta_{i+1} \leftarrow \theta_i - \eta\, \nabla_\theta \mathcal{L}(\theta_i)$$

end for

return θ_T

2.1 Adjoint Matching (AM)

AM belongs to the broad family of adjoint-based methods [9, 20] which rely on forward simulation of trajectories followed by differentiation through the dynamics. Given simulations $X^u \sim \mathbb{P}^{u_\theta}$, we can define the trajectory loss as:

$$\mathcal{L}_{\text{traj}}(\theta, X^u) := \int_0^1 \frac{1}{2}\|u_\theta(X_t^u, t)\|^2\, dt + \int_0^1 f(X_t^u, t)dt + g(X_1^u). \tag{7}$$

Then the gradient of the objective w.r.t θ can be written as (Proposition 6, [12]):

$$\frac{d}{d\theta}\mathcal{L}_{\text{traj}}(\theta, X^u) = \frac{1}{2}\int_0^1 \frac{\partial}{\partial\theta}\|u(X_t^u, t)\|^2\, dt + \int_0^1 \frac{\partial u(X_t^u, t)^\top}{\partial\theta}\, \sigma(t)^\top a(X_t^u, u, t)\, dt$$

where the adjoint $a(X_t^u, u, t)$ is defined by ODE,

$$\frac{d}{dt}a(X_t^u, u, t) = -\left[a(X_t^u, u, t)^\top \nabla_X \big(b(X_t^u, t) + \sigma(t)u(X_t^u, t)\big) + \nabla_X \left(f(X_t^u, t) + \tfrac{1}{2}\|u(X_t^u, t)\|^2\right)\right] \tag{8}$$

solved backwards in time starting from $a(X_1^u, 1) = \nabla_X g(X_1^u)$. Note that in Eq . **??**, the first term refers to the gradient of the quadratic cost and the second term is the gradient through the process. While optimization in Eq. 6 can be performed using an algorithm like gradient descent, a known result [27] in SOC is that the optimal control u^* satisfies the following equation:

$$u^*(X, t) = -\sigma(t)^\top \mathbb{E}_{\mathbb{P}^{u^*}}[a(X, u^*, t)|X_t = x] \tag{9}$$

While u^* is unknown, inspired by this, Domingo-Enrich et al. [12] propose an iterative optimization algorithm enforcing Eq. 9 with respect to the current control u^θ, by solving the fixed point equation :

$$\theta^* = \operatorname*{argmin}_\theta \mathcal{L}_{\text{fp}}(\theta)$$
$$\mathcal{L}_{\text{fp}}(\theta) = \mathbb{E}_{\mathbb{P}^u}\left[\int_0^1 \frac{1}{2}\left\|u_\theta(X_t^u, t) + \sigma(t)^\top \tilde{a}(X_t^u, t)\right\|^2 dt\right], \tag{10}$$

[1] Code is available here.

where $\tilde{a}$ is the so-called "lean-adjoint" ODE solved backward in time from $t = 1$:

$$\begin{aligned} \tilde{a}(X_1, 1) &= \nabla g(X_1), \\ \frac{d}{dt}\tilde{a}(X_t, t) &= -\Big(\nabla b(X_t, t)^\top \tilde{a}(X_t, t) + \nabla f(X_t, t)\Big). \end{aligned} \tag{11}$$

Note that minimizing Eq. 13 is equivalent to solving a fixed-point equation, and at optimality the solution satisfies

$$u^*(X, t) = -\mathbb{E}_{\mathbb{P}^{u^*}}\left[\sigma(t)^\top \tilde{a}(X, t) | X_t = X\right], \tag{12}$$

i.e., the optimum of the fixed point objective is a unique minimizer of Eq. 5. (See Proposition 2 Domingo-Enrich et al. [12]). To solve the fixed-point equation, AM proposes a iterative optimization, where at each iteration i trajectories $X^u \sim \mathbb{P}^{u_{\theta_i}}$ are sampled *on-policy* using the current control u_{θ_i}. A stochastic gradient step is then performed on the "one-step" AM objective given as :

$$\begin{aligned} \theta_{i+1} &= \operatorname*{argmin}_\theta \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i) \\ \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i) &= \mathbb{E}_{\mathbb{P}^{u_{\theta_i}}}\left[\int_0^1 \frac{1}{2}\left\|u_\theta(X_t^u, t) + \sigma(t)^\top \tilde{a}(X_t^u, t)\right\|^2 dt\right]. \end{aligned} \tag{13}$$

The complete procedure is summarized in Alg. 1. Although the one-step objective shares the same fixed point as the true objective, the latter defines the quantity of primary interest. It is therefore necessary to understand whether descent on the surrogate objective guarantees descent of the true objective. We investigate this relationship in the next section.

Algorithm 2 Fisher Adjoint Matching

Require: Drift b, diffusion σ, state cost $f(\cdot)$, terminal cost $g(\cdot)$, initial distribution p_0, parametric control u_θ, step size h, number of time steps K, batch size N, KL threshold `maxkl`, Number of training iterations T
for $i = 0, \cdots, T-1$ **do**

Sample N trajectories $\{X_k\}_{k=1}^N$ with Euler–Maruyama on $t_j = jh$, $j = 0, \ldots, K-1$:

$$\begin{aligned} &X_{k,0} \sim p_0\ , \quad \varepsilon_{k,j} \sim \mathcal{N}(0, I) \\ &X_{k,j+1} \leftarrow X_{k,j} + \big(b(X_{k,j}, t_j) + \sigma(t_j) u_\theta(X_{k,j}, t_j)\big)h + \sigma(t_j)\sqrt{h}\,\varepsilon_{k,j}, \quad \forall k, j. \end{aligned}$$

Compute adjoints via backward Euler discretization of (11):

$$\begin{aligned} &a_{k,K} \leftarrow \nabla g(X_{k,K}) \\ &a_{k,j} \leftarrow a_{k,j+1} - \big((\nabla_x b(X_{k,j}, t_j))^\top a_{k,j+1} + \nabla_x f(X_{k,j}, t_j)\big)h, \quad \forall j = K-1, \ldots, 0. \end{aligned}$$

Compute gradient $\nabla_\theta \mathcal{L}(\theta, \theta_i)$ using the AM loss (13)
Compute the natural-gradient direction $\Delta\theta$ by solving the linear system $F(\theta_i)\Delta\theta = \nabla_\theta \mathcal{L}(\theta_i)$ using CG
Perform backtracking line search along $\Delta\theta$ to find η such that $\mathrm{KL}\big(P^{\theta_i - \eta\Delta\theta} \,\|\, P^{\theta_i}\big) \le$ `maxkl`

$$\theta_{i+1} \leftarrow \theta_i - \eta\,\Delta\theta$$

end for
return θ_T

3 Monotone Descent

After one iteration of Adjoint Matching with update $\theta_{i+1} = \operatorname{argmin}_\theta \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i)$, the "improvement" in the true objective can be defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{fp}}(\theta_{i+1}) - \mathcal{L}_{\mathrm{fp}}(\theta_i) \\ = \mathbb{E}_{\mathbb{P}^{u_{\theta_{i+1}}}}\left[\Phi(X;\theta_{i+1})\right] - \mathbb{E}_{\mathbb{P}^{u_{\theta_i}}}\left[\Phi(X;\theta_i)\right],\end{aligned}$$

with $\Phi(X;\theta) := \int_0^1 \|u_\theta(X_t,t) + \sigma(t)^\top a(X_t,t)\|^2 dt$. Note the *two* simultaneous changes: (i) the integrand changes with θ, and (ii) the sampling distribution changes because the path measure $\mathbb{P}^{u_\theta}$ depends on θ. As a result, even if we update parameters by $\theta_{t+1} \leftarrow \theta_t - \eta \nabla_\theta \mathcal{L}_{\mathrm{AM}}(\theta_t)$, monotone decrease of $\mathcal{L}_{\mathrm{AM}}$ may require very small η to control the drift in the sampling measure.

This is different from say, smooth convex optimization, where for an L-smooth f, for a sufficiently small step size $\eta \leq 1/L$, gradient descent guarantees monotone decrease,

$$f(\theta_{i+1}) \leq f(\theta_i) - \eta \|\nabla f(\theta_i)\|^2,\ \theta_{i+1} = \theta_i - \eta \nabla f(\theta_i),$$

This motivates the question: *how large of a step can we take while still guaranteeing consistent improvement in the true objective?* To address this, we follow the trust-region literature in RL [39] and show that we can bound the true objective $\mathcal{L}_{\mathrm{fp}}(\theta_{i+1})$. Before we state the lemma, we make the following assumptions:

Assumption 3.1 ***[Bounded AM Residual]*** *There exists a constant $B < \infty$ such that for all admissible controls u, almost surely,*

$$\sup_{t\in[0,1]} \|u(X_t,t) + \sigma(t) a(X_t,t)\|^2 \leq B.$$

This assumption is imposed for analytical convenience. In practice, it may suffice to require the bound only along the sequence of controls $\{u_{\theta_i}\}_{i\geq 0}$ generated by the optimization procedure, starting from a suitably initialized control u_{θ_0}. We leave a formal treatment of this relaxation to future work. Under this assumption, we can state the following lemma:

Lemma 3.1 *[Upper Bound via surrogate] Under assumption 3.1, for any $\theta, \theta' \in \mathbb{R}^d$, we have the following relation:*

$$\mathcal{L}_{\mathrm{fp}}(\theta') \ \leq\ \mathcal{L}_{\mathrm{AM}}(\theta', \theta) \ +\ B\sqrt{\mathrm{KL}(\mathbb{P}^{u_\theta} \,\|\, \mathbb{P}^{u_{\theta'}})}. \tag{14}$$

Setting $\theta' = \theta_{i+1}$ and $\theta = \theta_i$, we get:

$$\mathcal{L}_{\mathrm{fp}}(\theta_{i+1}) \leq \mathcal{L}_{\mathrm{AM}}(\theta_{i+1}, \theta_i) + B \cdot \sqrt{\mathrm{KL}(P^{u_{\theta_{i+1}}} || P^{u_{\theta_i}})} \tag{15}$$

Note that while single AM update based on (13) minimizes the first term on the right, in general it is not sufficient to guarantee a monotone decrease in $\mathcal{L}_{\mathrm{fp}}(\theta)$. The above analysis, infact, motivates the upper bound of the "true" objective function as a more suitable loss, i.e.

$$\begin{aligned}\theta_{i+1} &= \operatorname{argmin}_\theta \mathcal{L}^{\mathcal{M}}_{\mathrm{AM}}(\theta, \theta_i) \\ \mathcal{L}^{\mathcal{M}}_{\mathrm{AM}}(\theta, \theta_i) &:= \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i) + B \cdot \sqrt{\mathrm{KL}(P^{u_{\theta_i}} || P^{u_\theta})}\end{aligned} \tag{16}$$

Under the modified AM objective defined in (16), we obtain the following descent guarantee:

$$\mathcal{L}_{\mathrm{fp}}(\theta_{i+1}) \ \leq\ \mathcal{L}^{\mathcal{M}}_{\mathrm{AM}}(\theta_{i+1}, \theta_i) \ \leq\ \mathcal{L}^{\mathcal{M}}_{\mathrm{AM}}(\theta_i, \theta_i) = \mathcal{L}_{\mathrm{fp}}(\theta_i)\,. \tag{17}$$

This ensures monotonic descent of the true objective.

4 Fisher Adjoint Matching (FAM)

While eq. 16 ensures monotonic improvement, in practice it can lead to very small step sizes. Thus, we instead propose to solve the following objective :

$$\min_{\theta} \quad \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i) \quad \text{s.t.} \quad \mathrm{KL}(P^{u_{\theta_i}} \parallel P^{u_\theta}) \leq \delta, \tag{18}$$

To solve the constrained problem in Eq. 18, we follow TRPO [39]. For a small update $\Delta\theta$, we *linearize* the one step loss $\mathcal{L}_{\mathrm{AM}}(\theta, \theta_i)$ via a first-order Taylor expansion around θ_i:

$$\mathcal{L}_{\mathrm{AM}}(\theta_i + \Delta\theta, \theta_i) \approx \mathcal{L}_{\mathrm{AM}}(\theta_i, \theta_i) + \nabla_\theta \mathcal{L}_{\mathrm{AM}}(\theta, \theta_i)\big|_{\theta=\theta_i}^{\top} \Delta\theta. \tag{19}$$

For the constraint, we have from Girsanov's theorem (Theorem 2, Domingo-Enrich et al. [12]):

$$\begin{aligned} &KL\big(P^{u_\theta} || P^{u_{\theta'}}\big) \\ &= \frac{1}{2}\,\mathbb{E}_{P^{u_\theta}}\left[\int_0^1 \left\|\sigma(t)^{-1}(u_{\theta'}(X_t,t) - u_\theta(X_t,t))\right\|^2 dt\right], \end{aligned}$$

Again, for small $\Delta\theta$ we have:

$$u_{\theta_i+\Delta\theta}(X_t,t) - u_{\theta_i}(X_t,t) \approx J_{\theta_i}(X_t,t)\Delta\theta \tag{20}$$

where,

$$J_{\theta_i}(X_t,t) \in \mathbb{R}^{d\times p}$$

denotes the Jacobian of the control with respect to the parameters, defined as

$$J_{\theta_i}(X_t,t) := \left.\frac{\partial u_\theta(X_t,t)}{\partial\theta}\right|_{\theta=\theta_i}.$$

Thus,

$$KL(P^{\theta_i+\Delta\theta}||P^{\theta_i}) = \mathbb{E}_{P^{u_{\theta_i}}}\frac{1}{2}\int \left\|\sigma(t)^{-1}J_\theta(X_t,t)\Delta\theta\right\|^2 dt \tag{21}$$

Thus, for a small $\Delta\theta$:, we can rewrite Eq. 16 as

$$\begin{aligned} \min_{\Delta\theta} \quad & \nabla_\theta \mathcal{L}^{\mathrm{surr}}_{\mathrm{AM}}(\theta,\theta_i)\Big|_{\theta=\theta_i}^{\top}\Delta\theta \\ \text{s.t.} \quad & \mathbb{E}_{\mathbb{P}^{u_{\theta_i}}}\left[\frac{1}{2}\int_0^1 \left\|\sigma(t)^{-1}J_{\theta_i}(X_t,t)\Delta\theta\right\|^2 dt\right] \leq \varepsilon. \end{aligned} \tag{22}$$

The above trust-region subproblem is a quadratic program with a linear objective and admits the closed-form solution

$$\Delta\theta^* = -\sqrt{\frac{2\varepsilon}{g^\top F^{-1} g}}\,F^{-1}g, \tag{23}$$

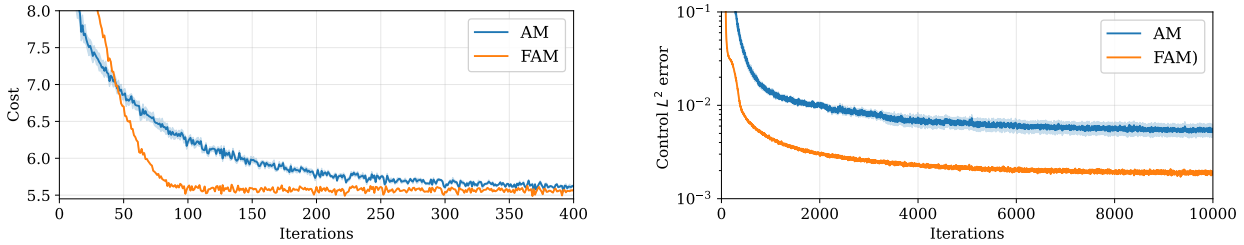
where

$$g := \nabla_\theta \mathcal{L}_{\mathrm{AM}}(\theta,\theta_i)\Big|_{\theta=\theta_i},$$

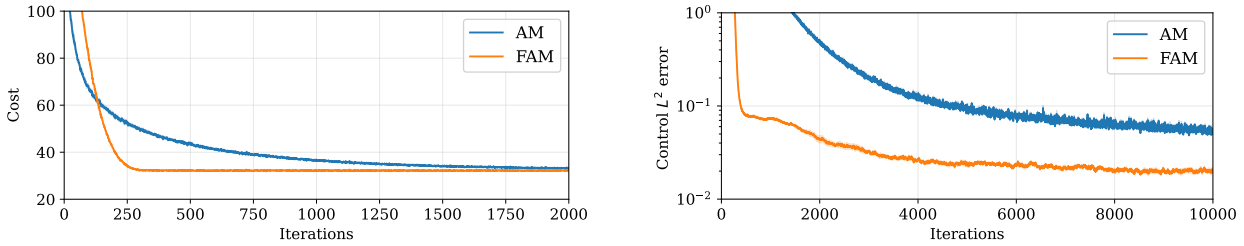
and F is given by:

$$F := \mathbb{E}_{\mathbb{P}^{u_{\theta_i}}}\left[\int_0^1 J_{\theta_i}(X_t,t)^\top \sigma(t)^{-1\top}\sigma(t)^{-1}J_{\theta_i}(X_t,t)\,dt\right].$$

The matrix F can be interpreted as a Fisher information matrix associated with the path measure induced by u_{θ_i}. It measures the local sensitivity of the trajectory distribution to infinitesimal parameter perturbations and defines the natural Riemannian metric on parameter space. The step $\Delta\theta^*$ corresponds to the *steepest descent direction under the specified KL constraint*.



(a) Easy OU setting. Left: control cost vs. iterations. Right: control L^2 error (log-scale).



(b) Hard OU setting. Left: control cost vs. iterations. Right: control L^2 error (log-scale).

Figure 2: Comparison of Adjoint Matching (AM) and Fisher Adjoint Matching (FAM) on quadratic Ornstein–Uhlenbeck control problems. Solid curves show the mean over three seeds and shaded regions indicate one standard deviation. FAM achieves faster convergence and lower control error, particularly in the harder regime.

4.1 Implementation

To solve (22), we use the Conjugate Gradient (CG) method to compute the search direction by approximately solving the linear system

$$(F + \lambda I)\Delta\theta = -g.$$

Importantly, this does not require explicitly forming the matrix F, since Fisher-vector products can be computed efficiently using Jacobian–vector and vector–Jacobian products (JVP/VJP) in PyTorch. We additionally use a damping term λ to improve numerical stability. The resulting direction defines the natural-gradient descent step. To determine the step size, we perform a backtracking line search to ensure that the KL constraint is satisfied. Implementation details and hyperparameters are provided in Appendix A.2.

5 Experiments

In this section, we present numerical experiments comparing Fisher Adjoint Matching (FAM) with Adjoint Matching (AM) on two representative settings. First, we consider classical stochastic control benchmarks based on Ornstein–Uhlenbeck processes [32, 11], where the optimal control admits a closed-form solution; second, we evaluate performance on sampling from unnormalized target densities following the adjoint sampling framework of [23]. We first describe the experimental setup and then detail the two evaluation settings.

5.1 Control Parameterization and Optimization

We follow the architectural setup of SOCM [11]. In all experiments, the control u^θ is parameterized by a fully connected U-Net. Specifically, we employ a simplified variant in which both the down- and up-sampling blocks are implemented using fully connected layers with ReLU activations, and the skip connections are

realized via linear projections. The network consists of three down- and three up-sampling stages with hidden widths 256, 128, and 64, respectively. For AM, we optimize using the Adam optimizer and select the best learning rate via a hyperparameter sweep. For FAM, we solve the resulting linear system using conjugate gradient (CG), as described in Section 4.1, and perform ablations to tune the KL constraint and damping parameters. Further details on all hyperparameters are provided in Appendix A.2.

5.2 Quadratic Ornstein–Uhlenbeck (OU) Processes

We consider a controlled Ornstein–Uhlenbeck (OU) process with linear drift $b(X_t, t) = AX_t$ for a fixed $A \in \mathbb{R}^{d\times d}$ and $\sigma(t) = \sigma_0$. The running cost $f(X_t, t) = X_t^\top P X_t$ and the terminal cost is $g(X_1) = X_1^\top Q X_1$ where $P, Q \in \mathbb{R}^{d\times d}$ are positive definite matrices. The OU process with quadratic cost admits a closed-form optimal control given by

$$u_t^*(x) = -2\,\sigma_0^\top F_t x, \tag{24}$$

where F_t solves the Riccati differential equation integrated backward in time:

$$\frac{dF_t}{dt} + A^\top F_t + F_t A - 2F_t\sigma_0\sigma_0^\top F_t + P = 0,\ F_1 = Q. \tag{25}$$

We consider two settings: (a) *Easy*: $d = 10$, $A = 0.2I$, $P = 0.2I$, $Q = 0.1I$, $\sigma_0 = 1$; and (b) *Hard*: $d = 20$, $A = I$, $P = I$, $Q = 2I$. The hard setting increases the difficulty of the optimization problem through both higher dimensionality and stronger drift, requiring larger control inputs to stabilize the system. The initial state is sampled from $p_0 = \mathcal{N}(0, 0.5I_d)$. We discretize the SDE using 50 time steps. To evaluate performance, we report the L^2-error with respect to the optimal control:

$$L^2\text{-error} = \mathbb{E}_{X\sim P^{u^*}}\left[\int_0^1 \|u(X_t, t) - u^*(X_t, t)\|^2\ dt\right]$$

Our results are shown in Fig. 2. FAM consistently attains the lower control L^2 error using fewer target evaluations across both the easy and hard OU settings, indicating improved sample efficiency. Next,we evaluate our method on the task of sampling from unnormalized target densities.

5.3 Adjoint Sampling

Adjoint Sampling [23] adapts AM to the task of sampling from an unnormalized target density $p^\star(x) \propto \exp(-E(x))$. In Adjoint Sampling, the base process is taken to have zero drift, i.e., $b(\cdot, \cdot) = 0$. For this uncontrolled process, the terminal distribution is

$$p_{\text{base}}(X_1) = \mathcal{N}(0, \sigma^2). \tag{26}$$

Setting the running cost $f = 0$ and the terminal cost $g(x) = \log p_{\text{base}}(x) + E(x)$, one can show that the terminal distribution of the optimally controlled process satisfies $P^{u^\star}(X_1) = p^\star(X_1)$. For additional details, see [12, 23].

For our evaluation, we consider $p^\star(x)$ to be a mixture of Gaussians (MoG). Specifically, we use a mixture with 10 components, random means, and random mixing weights. The component means are sampled from a Gaussian distribution with variance 2. As before, we set the number of discretization steps to 50. We evaluate performance in dimensions $d \in \{2, 5, 10, 50\}$. As in the OU experiments, we perform a hyperparameter sweep to select the best learning rate for each method.

We compare FAM (Forward Adjoint Matching) against AM (Adjoint Matching) using two evaluation metrics: (i) Sinkhorn distance [2], defined as the entropy-regularized optimal transport distance between generated samples and target samples, and (ii) the negative log-likelihood (NLL), computed as the negative log-density of generated samples under the target distribution $p^\star(x)$.

[2]Sinkhorn distance is computed using the `GeomLoss` library [15] via `SamplesLoss("sinkhorn", p=2, blur=0.05)`.

Table 1: Adjoint Sampling results on Mixture-of-Gaussians targets. We report mean ± standard deviation over three seeds at iteration 500. Lower is better for all metrics.

Dimension	Method	Sinkhorn ↓	NLL ↓
$d = 2$	AM	1.588 ± 0.597	3.931 ± 0.288
	FAM	$\mathbf{0.043 \pm 0.028}$	$\mathbf{3.248 \pm 0.276}$
$d = 5$	AM	$\mathbf{2.388} \pm 0.069$	7.273 ± 0.441
	FAM	2.465 ± 0.337	$\mathbf{5.913} \pm 0.073$
$d = 10$	AM	12.139 ± 4.764	12.020 ± 1.665
	FAM	$\mathbf{10.088} \pm 0.041$	$\mathbf{9.249} \pm 0.061$
$d = 50$	AM	$\mathbf{86.171} \pm 0.975$	54.097 ± 3.217
	FAM	86.954 ± 0.384	$\mathbf{43.442} \pm 0.050$

Qualitative results for $d = 2$ are shown in Fig. 1, and quantitative comparisons across dimensions are in Table 1. All results are averaged over three independent runs and reported at iteration 500 for both AM and FAM. FAM achieves lower Sinkhorn distance and lower NLL across dimensions, demonstrating improved sample efficiency over AM. Both methods degrade in high dimensions, which we attribute to the known difficulty of training diffusion models on high-dimensional Gaussian mixture targets, as observed by Vargas et al. [44].

6 Related Work

SOC. The Hamilton–Jacobi–Bellman (HJB) equation characterizes the solution to stochastic optimal control (SOC) problems. However, solving the associated nonlinear PDE in continuous time is computationally intractable in high dimensions, restricting classical numerical approaches such as finite-difference schemes to low-dimensional settings [30].

To address these limitation, Forward–Backward Stochastic Differential Equation (FBSDE) formulations have enabled learning-based approaches to SOC [17, 48, 20]. Recent work [10] provides a taxonomy of objective functions for SOC, showing that computing the optimal control can be formulated as minimizing a KL functional between path measures, where the target is a tilted (optimal) process.

Stochastic Optimal Control Matching (SOCM) [11] approaches this problem via a forward KL objective. However, it requires importance sampling from the optimally controlled process, which leads to high-variance estimators, particularly in high dimensions. Adjoint Matching (AM) [12] instead optimizes a reverse KL objective, thereby avoiding explicit importance sampling.

Natural Gradients. Natural gradient methods precondition the Euclidean gradient by the Fisher information (or an information-geometric metric), yielding parameterization-invariant steepest descent in distribution space and often improving stability under ill-conditioning [3]. In reinforcement learning, this yields the natural policy gradient update [26], and forms the conceptual basis for trust-region style methods that constrain the KL divergence between successive policies, such as TRPO [39]. For large neural networks, practical natural-gradient approximations (e.g., Kronecker-factored approximate curvature, K-FAC) make Fisher preconditioning computationally feasible and are widely used as scalable second-order surrogates [19].

7 Conclusion

In this work, we improve the optimization and sample efficiency of the Adjoint Matching (AM) framework for stochastic optimal control (SOC). We show that constraining each AM update to a trust region defined with respect to the current path measure guarantees monotonic decrease of the underlying fixed-point objective. To solve the resulting constrained problem, we propose an efficient algorithm that yields non-trivial step sizes while approximately enforcing the KL constraint. Empirically, Fisher Adjoint Matching (FAM) consistently outperforms standard AM across classical control and generative sampling tasks. Our approach adapts trust-region ideas from reinforcement learning to the SOC setting; extending other RL-based techniques to SOC remains an interesting direction for future work.

References

[1] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2022.

[2] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[3] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

[4] A. Belloni, L. Piroddi, and M. Prandini. A stochastic optimal control solution to the energy management of a microgrid with storage and renewables. In *2016 American Control Conference (ACC)*, pages 2340–2345, 2016.

[5] J. Berner, L. Richter, and K. Ullrich. An optimal control perspective on diffusion-based generative modeling, 2023.

[6] R. Carmona. *Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications*, volume 1. SIAM, 2016.

[7] R. Carmona, F. Delarue, et al. *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer, 2018.

[8] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):30, 2018.

[9] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

[10] C. Domingo-Enrich. A Taxonomy of Loss Functions for Stochastic Optimal Control, Oct. 2024. URL http://arxiv.org/abs/2410.00345. arXiv:2410.00345 [cs].

[11] C. Domingo-Enrich, J. Han, B. Amos, J. Bruna, and R. T. Q. Chen. Stochastic optimal control matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wfU2CdgmWt.

[12] C. Domingo-Enrich, M. Drozdzal, B. Karrer, and R. T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *ICLR*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

[13] R. J. Elliott. Stochastic optimal control: The discrete time case (dimitri p. bertsekas and steven f. shreve). *SIAM Review*, 22(2):237–238, 1980. doi: 10.1137/1022042. URL https://doi.org/10.1137/1022042.

[14] J. Feng and T. G. Kurtz. *Large Deviations for Stochastic Processes*, volume 131 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2006.

[15] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.

[16] W. H. Fleming and J. L. Stein. Stochastic optimal control, international finance and debt. *Journal of Banking & Finance*, 28(5):979–996, 2004.

[17] E. Gobet, J.-P. Lemor, X. Warin, et al. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.

[18] A. Gorodetsky, S. Karaman, and Y. Marzouk. High-dimensional stochastic optimal control using continuous tensor decompositions. *International Journal of Robotics Research*, 37(2-3), 3 2018.

[19] R. Grosse and J. Martens. A kronecker-factored approximate fisher matrix for convolution layers. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 573–582, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/grosse16.html.

[20] J. Han and W. E. Deep learning approximation for stochastic control problems, 2016. URL https://arxiv.org/abs/1611.07422.

[21] C. Hartmann and C. Schütte. Efficient rare event simulation by optimal nonequilibrium forcing. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(11):P11004, 2012.

[22] C. Hartmann, R. Banisch, M. Sarich, T. Badowski, and C. Schütte. Characterization of rare events in molecular dynamics. *Entropy*, 16(1):350–376, 2014.

[23] A. J. Havens, B. K. Miller, B. Yan, C. Domingo-Enrich, A. Sriram, D. S. Levine, B. M. Wood, B. Hu, B. Amos, B. Karrer, X. Fu, G.-H. Liu, and R. T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6Eg1OrHmg2.

[24] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[25] L. Holdijk, Y. Du, F. Hooft, P. Jaini, B. Ensing, and M. Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths, 2023. Preprint.

[26] S. M. Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

[27] H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. *AIP Conference Proceedings*, 887(1):149–181, 02 2007. ISSN 0094-243X. doi: 10.1063/1.2709596. URL https://doi.org/10.1063/1.2709596.

[28] S. Levine and V. Koltun. Guided policy search. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1–9, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/levine13.html.

[29] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling, 2022.

[30] J. Ma and J. Ma. Finite difference methods for the hamilton–jacobi–bellman equations arising in regime switching utility maximization. *J. Sci. Comput.*, 85(3), Dec. 2020. ISSN 0885-7474. doi: 10.1007/s10915-020-01352-4. URL https://doi.org/10.1007/s10915-020-01352-4.

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

[32] N. Nüsken and L. 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, 06 2021. doi: 10.1007/s42985-021-00102-x.

[33] H. Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

[34] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York, NY, 1961. URL https://www.google.ca/books/dfdfedition/Mathematical_Theory_of_Optimal_Processes/l3dZDwAAQBAJ. translated by K. N. Triruguf, edited by T. W. Neustadt.

[35] W. B. Powell and S. Meisel. Tutorial on stochastic optimization in energy: Part I: Modeling and policies. *IEEE Transactions on Power Systems*, 31(2):1459–1467, 2016.

[36] S. Reich. Data assimilation: The Schrödinger perspective. *Acta Numerica*, 28:635–711, 2019.

[37] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[38] L. Richter and J. Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

[40] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

[41] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

[42] E. Theodorou, J. Buchli, and S. Schaal. Learning policy improvements with path integrals. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 828–835, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/theodorou10a.html.

[43] E. Theodorou, F. Stulp, J. Buchli, and S. Schaal. An iterative path integral stochastic optimal control approach for learning robotic tasks. In *IFAC Proceedings Volumes*, volume 44, pages 11594–11601, 2011.

[44] F. Vargas, W. S. Grathwohl, and A. Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

[45] C. Villani. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003.

[46] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[47] L. Wang, C. Cheng, Y. Liao, Y. Qu, and G. Liu. Training free guided flow-matching with optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=61ss5RA1MM`.

[48] J. Zhang et al. A numerical scheme for BSDEs. *The annals of applied probability*, 14(1):459–488, 2004.

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

[50] W. Zhang, H. Wang, C. Hartmann, M. Weber, and C. Schütte. Applications of the cross-entropy method to importance sampling and optimal control of diffusions. *SIAM Journal on Scientific Computing*, 36(6): A2654–A2672, 2014.

A Appendix

A.1 Proofs

Before providing the proofs, we restate the assumptions.

Assumption 3.1 ***[Bounded AM Residual]*** *There exists a constant $B < \infty$ such that for all admissible controls u, almost surely,*

$$\sup_{t\in[0,1]} \|u(X_t,t) + \sigma(t)a(X_t,t)\|^2 \le B.$$

Lemma 3.1 *[Upper Bound via surrogate] Under assumption 3.1, for any $\theta, \theta' \in \mathbb{R}^d$, we have the following relation:*

$$\mathcal{L}_{\text{fp}}(\theta') \;\le\; \mathcal{L}_{\text{AM}}(\theta',\theta) \;+\; B\,\sqrt{\text{KL}(\mathbb{P}^{u_\theta} \,\|\, \mathbb{P}^{u_{\theta'}})}. \tag{14}$$

Proof: Fix θ (current iterate) and θ' (candidate). Consider the quantity

$$\mathcal{L}_{\text{fp}}(\theta') = \mathbb{E}_{X^u\sim P^{u_{\theta'}}}\left[\int_0^1 \left\|u_{\theta'}(X_t^{u_{\theta'}}) + \sigma(t)^\top \tilde{a}(X_t^{u_{\theta'}},t)\right\|^2 dt\right] \tag{27}$$

Define $f(X) = \int_0^1 \left\|u_{\theta'}(X_t,t) + \sigma(t)^\top \tilde{a}(X_t,t)\right\|^2 dt$. Using Assumption 3.1, we can bound this using TV distance:

$$|\mathbb{E}_{X\sim P^{u_{\theta'}}}[f(X^u)] - \mathbb{E}_{X\sim P^{u_\theta}}[f(X^u)]| \le B\dot{TV}(P^{u^\theta}, P^{u^{\theta'}}) \le C\sqrt{KL(P^{u^\theta}||P^{u^{\theta'}})}. \tag{28}$$

where the final step is via Pinsker's inequality. Thus,

$$\mathcal{L}_{\text{fp}}(\theta') \le \mathcal{L}_{\text{AM}}(\theta',\theta) + C\cdot\sqrt{KL(P^{u_\theta}||P^{u_{\theta'}})} \tag{29}$$

A.2 Experimental Details

A.2.1 Conjugate Gradient (CG) Settings

For the conjugate gradient (CG) solver, we use `torch.autograd.functional` primitives to apply the Gauss–Newton matrix

$$F \triangleq \frac{1}{K}\sum_{i=1}^{K} J_i^\top J_i$$

via Jacobian–vector and vector–Jacobian products, without explicitly forming J_i. We solve the damped system

$$(F + \lambda I)\Delta\theta = -g,$$

where the damping parameter is swept over

$$\lambda \in \{10^{-4}, 10^{-2}, 10^{-1}\}.$$

For backtracking line search, we sweep the KL threshold `maxkl` over

$$\{10^{-4}, 10^{-2}, 10^{-1}, 1, 5\}.$$

We use 20 CG iterations for all experiments.

A.2.2 OU Settings

For the OU experiments, the best-performing KL threshold is

$$\texttt{maxkl} = 1e-4.$$

For the Adjoint Matching (AM) baseline, we use Adam with learning rate

$$\texttt{lr} = 1e-3.$$

A.2.3 Diffusion Sampling Settings

For the generative-model (diffusion sampling) experiments, the best-performing KL threshold is

$$\texttt{maxkl} = 1e-2,$$

and the AM baseline is optimized using Adam with

$$\texttt{lr} = 1e-3.$$

For the GMM sampling experiments, we generate 10 Gaussian mixture models with random means and random mixture weights. The component means are drawn from a Gaussian distribution with mean zero and variance 2 for $d = 2$, and variance 1.5 for $d \in \{5, 10, 20, 50\}$.