Stochastic Optimal Control Matching

Carles Domingo-Enrich
NYU & FAIR, Meta
cd2754@nyu.edu

Jiequn Han
Flatiron Institute
jhan@flatironinstitute.org

Brandon Amos
FAIR, Meta
bda@meta.com

Joan Bruna
NYU & Flatiron Institute
bruna@cims.nyu.edu

Ricky T. Q. Chen
FAIR, Meta
rtqichen@meta.com

Abstract

Stochastic optimal control, which has the goal of driving the behavior of noisy systems, is broadly applicable in science, engineering and artificial intelligence. Our work introduces Stochastic Optimal Control Matching (SOCM), a novel Iterative Diffusion Optimization (IDO) technique for stochastic optimal control that stems from the same philosophy as the conditional score matching loss for diffusion models. That is, the control is learned via a least squares problem by trying to fit a matching vector field. The training loss, which is closely connected to the cross-entropy loss, is optimized with respect to both the control function and a family of reparameterization matrices which appear in the matching vector field. The optimization with respect to the reparameterization matrices aims at minimizing the variance of the matching vector field. Experimentally, our algorithm achieves lower error than all the existing IDO techniques for stochastic optimal control for three out of four control problems, in some cases by an order of magnitude. The key idea underlying SOCM is the path-wise reparameterization trick, a novel technique that may be of independent interest.

1 Introduction

Stochastic optimal control aims to drive the behavior of a noisy system in order to minimize a given cost. It has myriad applications in science and engineering: examples include the simulation of rare events in molecular dynamics [37, 36, 85, 41], finance and economics [63, 25], stochastic filtering and data assimilation [58, 68], nonconvex optimization [19], sampling [9], power systems and energy markets [8, 66], and robotics [77, 32]. Stochastic optimal has also been impactful in fields such as mean-field games [17], optimal transport [80, 81], backward stochastic differential equations (BSDEs) [14] and large deviations [24]. Recently, it has been the basis of algorithms to sample from unnormalized densities [84, 79, 9, 71].

For continuous-time problems with low-dimensional state spaces, the standard approach to learn the optimal control is to solve the Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE) by gridding the space and using classical numerical methods. For high-dimensional problems, a large number of works parameterize the control using a neural network and train it applying a stochastic optimization algorithm on a loss function. These methods are known as *Iterative Diffusion Optimization* (IDO) techniques [59] (see Subsec. 2.2).

It is convenient to draw an analogy between stochastic optimal control and *continuous normalizing flows* (CNFs), which are a generative modeling technique where samples are generated by solving an ordinary differential equation (ODE) for which the vector field has been learned, initialized at a Gaussian sample. CNFs were introduced by [20] (building on top of Rezende and Mohamed [70]),

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

and training them is similar to solving control problems because in both cases one needs to learn high-dimensional vector fields using neural networks, in continuous time.

The first algorithm developed to train normalizing flows was based on maximizing the likelihood of the generated samples [20, Sec. 4]. Obtaining the gradient of the maximum likelihood loss with respect to the vector field parameters requires backpropagating through the computation of the ODE trajectory, or equivalently, solving the *adjoint* ODE in parallel to the original ODE. Maximum likelihood CNFs (ML-CNFs) were superseded by diffusion models [75, 40, 76] and flow-matching, a.k.a. stochastic interpolant, methods [55, 1, 65, 2], which are currently the preferred algorithms to train CNFs. Aside from architectural improvements such as the UNet [73], a potential reason for the success of diffusion and flow matching models is that their *functional landscape* is convex, unlike for ML-CNFs. Namely, vector fields are learned by solving least squares regression problems where the goal is to fit a random matching vector field. Convex functional landscapes in combination with overparameterized models and moderate gradient variance can yield very stable training dynamics and help achieve low error.

Returning to stochastic optimal control, one of the best-performing IDO techniques amounts to choosing the control objective (equation 1) as the training loss (see (12)). As in ML-CNFs, computing the gradient of this loss requires backpropagating through the computation of the trajectories of the SDE (2), or equivalently, using an adjoint method. The functional landscape of the loss is highly non-convex, and the method is prone to unstable training (see green curve in the bottom right plot of Figure 3). In light of this, a natural idea is to develop the analog of diffusion model losses for the stochastic optimal control problem, to obtain more stable training and lower error, and this is what we set out to do in our work. Our contributions are as follows:

- We introduce Stochastic Optimal Control Matching (SOCM), a novel IDO algorithm in which the control is learned by solving a least-squares regression problem where the goal is to fit a random *matching vector field* which depends on a family of *reparameterization matrices* that are also optimized.
- We derive a bias-variance decomposition of the SOCM loss (Prop. 2). The bias term is equal to an existing IDO loss: the *cross-entropy loss*, which shows that both algorithms have the same landscape in expectation. However, SOCM has an extra flexibility in the choice of reparameterization matrices, which affect only the variance. Hence, we propose optimizing the reparameterization matrices to reduce the variance of the SOCM objective.
- The key idea that underlies the SOCM algorithm is the *path-wise reparameterization trick* (Prop. 1), which is a novel technique for estimating gradients of an expectation of a functional of a random process with respect to its initial value. It is of independent interest and may be more generally applicable outside of the settings considered in this paper.
- We perform experiments on four different settings where we have access to the ground-truth control. For three of these, SOCM obtains a lower L^2 error with respect to the ground-truth control than all the existing IDO techniques, with around $10\times$ lower error than competing methods in some instances.

2 Framework

2.1 Setup and Preliminaries

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathcal{P})$ be a fixed filtered probability space on which is defined a Brownian motion $B = (B_t)_{t\geq 0}$. We consider the control-affine problem

$$\min_{u\in\mathcal{U}} \mathbb{E}\big[\int_0^T \big(\tfrac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\big)\,\mathrm{d}t + g(X_T^u)\big], \tag{1}$$

$$\text{s.t. } \mathrm{d}X_t^u = (b(X_t^u,t) + \sigma(t)u(X_t^u,t))\,\mathrm{d}t + \sqrt{\lambda}\sigma(t)\mathrm{d}B_t,\ X_0^u \sim p_0 \tag{2}$$

and where $X_t^u \in \mathbb{R}^d$ is the state, $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the feedback control and belongs to the set of admissible controls $\mathcal{U}$, $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ is the state cost, $g : \mathbb{R}^d \to \mathbb{R}$ is the terminal cost, $b : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the base drift, and $\sigma : [0,T] \to \mathbb{R}^{d\times d}$ is the invertible diffusion coefficient and $\lambda \in (0,+\infty)$ is the noise level. In App. A we formally define the set $\mathcal{U}$ of admissible controls and describe the regularity assumptions needed on the control functions. In the remainder of the section we introduce relevant concepts in stochastic optimal control; we provide the most relevant proofs in App. B and refer the reader to Oksendal [60, Chap. 11] and Nüsken and Richter [59, Sec. 2] for a similar, more extensive treatment.

Cost functional and value function The *cost functional* for the control u, point x and time t is defined as $J(u; x, t) := \mathbb{E}\big[\int_t^T \big(\frac{1}{2}\|u_s(X_s^u)\|^2 + f_s(X_s^u)\big) \, \mathrm{d}t + g(X_T^u) \big| X_t^u = x \big]$. That is, the cost functional is the expected value of the control objective restricted to the times $[t, T]$ with the initial value x at time t. The *value function* or *optimal cost-to-go* at (x, t) is defined as the minimum value of the cost functional across all possible controls:

$$V(x, t) := \inf_{u \in \mathcal{U}} J(u; x, t). \tag{3}$$

Hamilton-Jacobi-Bellman equation and optimal control If we define the infinitesimal generator $L := \frac{\lambda}{2} \sum_{i,j=1}^d (\sigma\sigma^\top)_{ij}(t) \partial_{x_i} \partial_{x_j} + \sum_{i=1}^d b_i(x, t) \partial_{x_i}$, the value function solves the following Hamilton-Jacobi-Bellman (HJB) partial differential equation:

$$(\partial_t + L)V(x, t) - \tfrac{1}{2}\|(\sigma^\top \nabla V)(x, t)\|^2 + f(x, t) = 0, \qquad V(x, T) = g(x). \tag{4}$$

The *verification theorem* [62, Sec. 2.3] states that if a function V solves the HJB equation above and has certain regularity conditions, then V is the value function (3) of the problem (1)-(2). An implication of the verification theorem is that for every $u \in \mathcal{U}$,

$$V(x, t) + \mathbb{E}\big[\tfrac{1}{2} \int_t^T \|\sigma^\top \nabla V + u\|^2(X_s^u, s) \, \mathrm{d}s \, \big| \, X_t^u = x\big] = J(u, x, t). \tag{5}$$

In particular, this implies that the unique optimal control is given in terms of the value function as $u^*(x, t) = -\sigma(t)^\top \nabla V(x, t)$. Equation (5) can be deduced by integrating the HJB equation (4) over $[t, T]$, and taking the conditional expectation with respect to $X_t^u = x$. We include the proof of (5) in App. B for completeness.

A pair of forward and backward SDEs (FBSDEs) Consider the pair of SDEs

$$\mathrm{d}X_t = b(X_t, t) \, \mathrm{d}t + \sqrt{\lambda}\sigma(t)\mathrm{d}B_t, \qquad X_0 \sim p_0, \tag{6}$$

$$\mathrm{d}Y_t = (-f(X_t, t) + \tfrac{1}{2}\|Z_t\|^2) \, \mathrm{d}t + \sqrt{\lambda}\langle Z_t, \mathrm{d}B_t \rangle, \qquad Y_T = g(X_T). \tag{7}$$

where $Y : \Omega \times [0, T] \to \mathbb{R}$ and $Z : \Omega \times [0, T] \to \mathbb{R}^d$ are progressively measurable [1] random processes. It turns out that Y_t and Z_t defined as $Y_t := V(X_t, t)$ and $Z_t := \sigma(t)^\top \nabla V(X_t, t) = -u^*(X_t, t)$ satisfy (7). We include the proof in App. B for completeness.

An analytic expression for the value function From the forward-backward equations (6)-(7), one can derive a closed-form expression for the value function V:

$$V(x, t) = -\lambda \log \mathbb{E}\big[\exp\big(-\lambda^{-1} \int_t^T f(X_s, s) \, \mathrm{d}s - \lambda^{-1} g(X_T)\big) \big| X_t = x\big], \tag{8}$$

where X_t is the solution of the uncontrolled SDE (6). This is a classical result, but we still include its proof in App. B. Given that $u^*(x, t) = -\sigma(t)^\top \nabla V(x, t)$, an immediate, yet important, consequence of (8) is the following path-integral representation of the optimal control:

$$u^*(x, t) = \lambda \sigma(t)^\top \nabla_x \log \mathbb{E}\big[\exp\big(-\lambda^{-1} \int_t^T f(X_s, s) \, \mathrm{d}s - \lambda^{-1} g(X_T)\big) \big| X_t = x\big]. \tag{9}$$

Remark this equation involves the gradient of logarithm of a conditional expectation, which is reminiscent of the vector fields that are learned when training diffusion models. For example, the target vector field for variance-exploding score-based diffusion loss [76] can be expressed as $\nabla_x \log p_t(x) = \nabla_x \log \mathbb{E}_{Y \sim p_{\mathrm{data}}}[\frac{\exp(-\|x - Y\|^2/(2\sigma_t^2))}{(2\pi\sigma_t^2)^{d/2}}]$. Note, however, that in (9) the gradient is taken with respect to the initial condition of the process, which requires the development of novel techniques.

Conditioned diffusions Let $\mathcal{C} = C([0, T]; \mathbb{R}^d)$ be the Wiener space of continuous functions from $[0, T]$ to $\mathbb{R}^d$ equipped with the supremum norm, and let $\mathcal{P}(\mathcal{C})$ be the space of Borel probability measures over $\mathcal{C}$. For each control $u \in \mathcal{U}$, the controlled process in equation (2) induces a probability measure in $\mathcal{P}(\mathcal{C})$, as the law of the paths X_t^u, which we refer to as $\mathbb{P}^u$. We let $\mathbb{P}$ be the probability measure induced by the uncontrolled process (6), and define the *work functional*

$$\mathcal{W}(X, t) := \int_t^T f(X_s, s) \, \mathrm{d}s + g(X_T). \tag{10}$$

[1] Being progressively measurable is a strictly stronger property than the notion of being a process adapted to the filtration $\mathcal{F}_t$ of B_t (see [50]).

It turns out (Lemma 2 in App. B) that the Radon-Nikodym derivative $\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}$ satisfies $\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) = \exp\big(\lambda^{-1}\big(V(X_0,0) - \mathcal{W}(X,0)\big)\big)$. Also, a straight-forward application of the Girsanov theorem for SDEs (Cor. 1) shows that

$$\frac{d\mathbb{P}^{u}}{d\mathbb{P}^{u^*}}(X^{u^*}) = \exp\big(-\lambda^{-1/2}\int_0^T \langle u^*(X_t^{u^*},t) - u(X_t^{u^*},t), \mathrm{d}B_t\rangle - \frac{\lambda^{-1}}{2}\int_0^T \|u^*(X_t^{u^*},t) - u(X_t^{u^*},t)\|^2 \,\mathrm{d}t\big), \tag{11}$$

which means that the only control $u \in \mathcal{U}$ such that $\mathbb{P}^u = \mathbb{P}^{u^*}$ is the optimal control itself.

2.2 Existing approaches and related work

Low-dimensional case: solving the HJB equation For low-dimensional control problems ($d \leq 3$), it is possible to grid the domain and use a numerical PDE solver to find a solution to the HJB equation (4). The main approaches include *finite difference methods* [11, 57, 4], which approximate the derivatives and gradients of the value function using finite differences, *finite element methods* [47], which involve restricting the solution to domain-dependent function spaces, and semi-Lagrangian schemes [21, 13, 12], which trace back characteristics and have better stability than finite difference methods. See Greif [33] for an overview on these techniques, and Bañas et al. [4] for a comparison between them. Hutzenthaler et al. [44] introduced the multilevel Picard method, which leverages the Feynman-Kac and the Bismut-Elworthy-Li formulas to beat the curse of dimensionality in some settings [6, 46, 45, 43].

High dimensional methods leveraging FBSDEs The FBSDE formulation in equations (6)-(7) has given rise to multiple methods to learn controls. One such approach is *least-squares Monte Carlo* (see Pham [63, Chapter 3] and Gobet [28] for an introduction, and Gobet et al. [30], Zhang et al. [83] for an extensive analysis), where trajectories from the forward process (6) are sampled, and then regression problems are solved backwards in time to estimate the expected future cost in the spirit of dynamic programming. A second method that exploits FBSDEs was proposed by E et al. [22], Han et al. [35]. They parameterize the control using a neural network u_θ, and use stochastic gradient algorithms to minimize the loss $\mathcal{L}(u_\theta, y_0) = \mathbb{E}[(Y_T(y_0, u_\theta) - g(X_T))^2]$, where $Y_T(y_0, u_\theta)$ is the process in (7) with initial condition y_0 and control u_θ. This algorithm can be seen as a shooting method, where the initial condition and the control are learned to match the terminal condition. Multiple recent works have combined neural networks with FBSDE Monte Carlo methods for parabolic and elliptic PDEs [5, 18, 86], control [7, 39], multi-agent games [34, 15, 16]; see [23] for a more comprehensive review.

Many of the methods referenced above and some additional ones can be seen from a common perspective using controlled diffusions. As observed in equation (11), the key idea is that learning the optimal control is equivalent to finding a control u such that the induced probability measure $\mathbb{P}^u$ on paths is equal to the probability measure $\mathbb{P}^{u^*}$ for the optimal control. In the paragraphs below we cover several loss that fall into this framework. All the losses below can be optimized using a common algorithmic framework, which we describe in Algorithm 1. For more details, we refer the reader to Nüsken and Richter [59], which introduced this perspective and named such methods *Iterative Diffusion Optimization* (IDO) techniques. For simplicity, we introduce the losses for the setting in which the initial distribution p_0 is concentrated at a single point x_{init}; we cover the general setting in App. B.

The relative entropy loss and the adjoint method The relative entropy loss is defined as the Kullback-Leibler divergence between $\mathbb{P}^u$ and $\mathbb{P}^{u^*}$: $\mathbb{E}_{\mathbb{P}^u}[\log \frac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}]$. Upon removing constant terms and factors, this loss is equivalent to (see Lemma 3 in App. B):

$$\mathcal{L}_{\mathrm{Adj}}(u) := \mathbb{E}\big[\int_0^T \big(\tfrac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\big)\,\mathrm{d}t + g(X_T^u)\big]. \tag{12}$$

This is exactly the control objective in (1). This fact has been studied extensively [10, 31, 36, 48, 67]. Hence, the relative entropy loss is very natural and widely used; see Onken et al. [61], Zhang and Chen [84] for examples on multiagent systems and sampling.

Solving optimization problems of the form (12) has a long history that dates back to Pontryagin [64]. Note that $\mathcal{L}_{\mathrm{Adj}}(u)$ depends on u both explicitly, and implicitly through the process X^u. To compute the gradient $\nabla_\theta \hat{\mathcal{L}}_{\mathrm{Adj}}(u_{\theta_n})$ of a Monte Carlo approximation $\hat{\mathcal{L}}_{\mathrm{Adj}}(u_{\theta_n})$ of $\mathcal{L}_{\mathrm{Adj}}(u_{\theta_n})$ as required by Algorithm 1, we need to backpropagate through the simulation of the trajectories, which is why

Algorithm 1 Iterative Diffusion Optimization (IDO) algorithms for stochastic optimal control

```
Input: State cost $f(x,t)$, terminal cost $g(x)$, diffusion coeff. $\sigma(t)$, base drift $b(x,t)$, noise level $\lambda$, number of iterations $N$, batch size $m$, number of time steps $K$, initial control parameters $\theta_0$, loss $\mathcal{L} \in \{\mathcal{L}_{\mathrm{Adj}}(12), \mathcal{L}_{\mathrm{CE}}(13), \mathcal{L}_{\mathrm{Var}_v}(16), \mathcal{L}^{\log}_{\mathrm{Var}_v}(17), \mathcal{L}_{\mathrm{Mom}_v}(18)\}$
for $n \in \{0, \dots, N-1\}$ do
    Simulate $m$ trajectories of the process $X^v$ controlled by $v = u_{\theta_n}$, e.g., using Euler-Maruyama updates
    if $\mathcal{L} \neq \mathcal{L}_{\mathrm{Adj}}$ then detach the $m$ trajectories from the computational graph, so that gradients do not backpropagate;
    Using the $m$ trajectories, compute an $m$-sample Monte Carlo approximation $\hat{\mathcal{L}}(u_{\theta_n})$ of the loss $\mathcal{L}(u_{\theta_n})$
    Compute the gradients $\nabla_\theta \hat{\mathcal{L}}(u_{\theta_n})$ of $\hat{\mathcal{L}}(u_{\theta_n})$ w.r.t. $\theta_n$
    Obtain $\theta_{n+1}$ with via an Adam update on $\theta_n$ (or another stochastic algorithm)
end
Output: Learned control $u_{\theta_N}$
```

we do *not* detach them from the computational graph. One can alternatively compute the gradient $\nabla_\theta \hat{\mathcal{L}}_{\mathrm{Adj}}(u_{\theta_n})$ by explicitly solving an ODE, a technique known as the *adjoint method*. The adjoint method was introduced by Pontryagin [64], popularized in deep learning by Chen et al. [20], and further developed for SDEs in Li et al. [54].

The cross-entropy loss The cross-entropy loss is defined as the Kullback-Leibler divergence between $\mathbb{P}^{u^*}$ and $\mathbb{P}^u$, i.e., flipping the order of the two measures: $\mathbb{E}_{\mathbb{P}^{u^*}}[\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}]$. For an arbitrary $v \in \mathcal{U}$, this loss is equivalent to the following one (see Prop. 3(i) in App. B):

$$\begin{aligned}\mathcal{L}_{\mathrm{CE}}(u) := \mathbb{E}\big[\big(&-\lambda^{-1/2}\int_0^T \langle u(X_t^v,t), \mathrm{d}B_t\rangle - \lambda^{-1}\int_0^T \langle u(X_t^v,t), v(X_t^v,t)\rangle\,\mathrm{d}t + \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v,t)\|^2\,\mathrm{d}t\big) \\ &\times \exp\big(-\lambda^{-1}\mathcal{W}(X^v,0) - \lambda^{-1/2}\int_0^T \langle v(X_t^v,t), \mathrm{d}B_t\rangle - \tfrac{\lambda^{-1}}{2}\int_0^T \|v(X_t^v,t)\|^2\,\mathrm{d}t\big)\big].\end{aligned} \tag{13}$$

The cross-entropy loss has a rich literature [38, 49, 74, 85] and has been recently used in applications such as molecular dynamics [41]. Furthermore, we note that the cross-entropy loss can be significantly simplified and written in terms of the unnormalized L^2 error of the control u with respect to the optimal control u^*:

$$\mathcal{L}_{\mathrm{CE}}(u) = \tfrac{\lambda^{-1}}{2}\mathbb{E}\big[\int_0^T \|u^*(X_t^{u^*},t) - u(X_t^{u^*},t)\|^2\,\mathrm{d}t \times \exp\big(-\lambda^{-1}V(X_0^{u^*},0)\big)\big]. \tag{14}$$

This characterization, which is proven in Prop. 3(ii) in App. B, is relevant for us because a similar one can be written for the loss that we propose (see Prop. 2).

Variance and log-variance losses For an arbitrary $v \in \mathcal{U}$, the *variance* and the *log-variance losses* are defined as $\tilde{\mathcal{L}}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}(\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u})$ and $\tilde{\mathcal{L}}^{\log}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}(\log \frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u})$ whenever $\mathbb{E}_{\mathbb{P}^v}|\frac{\mathrm{d}\mathbb{P}^{u^*}}{\mathrm{d}\mathbb{P}^u}| < +\infty$ and $\mathbb{E}_{\mathbb{P}^v}|\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}| < +\infty$, respectively. Define

$$\begin{aligned}\tilde{Y}_T^{u,v} &= -\lambda^{-1}\int_0^T \langle u(X_t^v,t), v(X_t^v,t)\rangle\,\mathrm{d}t \\ &\quad - \lambda^{-1}\int_0^T f(X_t^v,t)\,\mathrm{d}t - \lambda^{-1/2}\int_0^T \langle u(X_t^v,t), \mathrm{d}B_t\rangle \\ &\quad + \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v,t)\|^2\,\mathrm{d}t.\end{aligned} \tag{15}$$

Then, $\tilde{\mathcal{L}}_{\mathrm{Var}_v}$ and $\tilde{\mathcal{L}}^{\log}_{\mathrm{Var}_v}$ are equivalent, respectively, to the following losses (see Lemma 4):

$$\mathcal{L}_{\mathrm{Var}_v}(u) := \mathrm{Var}\big(\exp\big(\tilde{Y}_T^{u,v} - \lambda^{-1}g(X_T^v)\big)\big), \tag{16}$$

$$\mathcal{L}^{\log}_{\mathrm{Var}_v}(u) := \mathrm{Var}\big(\tilde{Y}_T^{u,v} - \lambda^{-1}g(X_T^v)\big), \tag{17}$$

The variance and log-variance losses were introduced by Nüsken and Richter [59]. Unlike for the cross-entropy loss, the choice of the control v does lead to different losses. When using $\mathcal{L}_{\mathrm{Var}_v}$ or $\mathcal{L}^{\log}_{\mathrm{Var}_v}$ in Algorithm 1, the variance is computed across the m trajectories in each batch.

Moment loss For an arbitrary $v \in \mathcal{U}$, the moment loss is defined as

$$\mathcal{L}_{\mathrm{Mom}_v}(u, y_0) = \mathbb{E}[(\tilde{Y}_T^{u,v} + y_0 - \lambda^{-1}g(X_T^v))^2], \tag{18}$$

where $\tilde{Y}_T^{u,v}$ is defined in (15). Note the similarity with the log-variance loss (17); the optimal value of y_0 for a fixed u is $y_0^* = \mathbb{E}[\lambda^{-1} g(X_T^v) - \tilde{Y}_T^{u,v}]$, and plugging this into (18) yields exactly the log-variance loss. The moment loss was introduced by Hartmann et al. [39, Section III.B], and it is a generalization of the FBSDE method pioneered by E et al. [22], Han et al. [35] and referenced earlier in this subsection, which corresponds to setting $v = 0$.

3 Stochastic Optimal Control Matching

In this section we present our loss, *Stochastic Optimal Control Matching* (SOCM). The corresponding method, which we describe in Algorithm 2, falls into the class of IDO techniques described in Subsec. 2.2. The general idea is to leverage the analytic expression of u^* in (9) to write a least squares loss for u, and the main challenge is to reexpress the gradient of a conditional expectation with respect to the initial condition of the process. We do that using a novel technique which introduces certain arbitrary matrix-valued functions M_t, that we also optimize.

Theorem 1 (SOCM loss). *For each $t \in [0, T]$, let $M_t : [t, T] \to \mathbb{R}^{d\times d}$ be an arbitrary matrix-valued differentiable function such that $M_t(t) = \mathrm{Id}$. Let $v \in \mathcal{U}$ be an arbitrary control. Let $\mathcal{L}_{\mathrm{SOCM}} : L^2(\mathbb{R}^d \times [0, T]; \mathbb{R}^d) \times L^2([0, T]^2; \mathbb{R}^{d\times d}) \to \mathbb{R}$ be the loss function defined as*

$$\mathcal{L}_{\mathrm{SOCM}}(u, M) := \mathbb{E}\big[\tfrac{1}{T} \int_0^T \big\|u(X_t^v, t) - w(t, v, X^v, B, M_t)\big\|^2 \, \mathrm{d}t \times \alpha(v, X^v, B)\big] \,, \qquad (19)$$

where X^v is the process controlled by v (i.e., $dX_t^v = (b(X_t^v, t) + \sigma(t)v(X_t^v, t)) \, \mathrm{d}t + \sqrt{\lambda}\sigma(t) \, \mathrm{d}B_t$ and $X_0^v \sim p_0$), and

$$\begin{aligned} w(t, v, X^v, B, M_t) &= \sigma(t)^\top \big(- \textstyle\int_t^T M_t(s)\nabla_x f(X_s^v, s) \, \mathrm{d}s - M_t(T)\nabla g(X_T^v) \\ &\quad + \textstyle\int_t^T (M_t(s)\nabla_x b(X_s^v, s) - \partial_s M_t(s))(\sigma^{-1})^\top(s) v(X_s^v, s) \, \mathrm{d}s \\ &\quad + \lambda^{1/2} \textstyle\int_t^T (M_t(s)\nabla_x b(X_s^v, s) - \partial_s M_t(s))(\sigma^{-1})^\top(s) \mathrm{d}B_s\big), \\ \alpha(v, X^v, B) &= \exp\big(- \lambda^{-1} \textstyle\int_0^T f(X_t^v, t) \, \mathrm{d}s - \lambda^{-1} g(X_T^v) \\ &\quad - \lambda^{-1/2} \textstyle\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t\rangle - \frac{\lambda^{-1}}{2} \int_0^T \|v(X_t^v, t)\|^2 \, \mathrm{d}t\big). \end{aligned} \qquad (20)$$

$\mathcal{L}_{\mathrm{SOCM}}$ has a unique optimum (u^*, M^*), where u^* is the optimal control.

We refer to $M = (M_t)_{t\in[0,T]}$ as the family of *reparametrization matrices*, to the random vector field w as the *matching vector field*, and to α as the *importance weight*. We present a proof sketch of Thm. 1; the full proofs for all the results in this section are in App. C.

Proof sketch of Thm. 1 Let X be the uncontrolled process (6). Consider the loss

$$\begin{aligned} \tilde{\mathcal{L}}(u) &= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|u(X_t, t) - u^*(X_t, t)\big\|^2 \, \mathrm{d}t \, \exp\big(- \lambda^{-1} \int_0^T f(X_t, t) \, \mathrm{d}t - \lambda^{-1} g(X_T)\big)\big] \\ &= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big(\big\|u(X_t, t)\big\|^2 - 2\langle u(X_t, t), u^*(X_t, t)\rangle + \|u^*(X_t, t)\|^2\big) \, \mathrm{d}t \\ &\qquad \times \exp\big(- \lambda^{-1} \textstyle\int_0^T f(X_t, t) \, \mathrm{d}t - \lambda^{-1} g(X_T)\big)\big]. \end{aligned} \qquad (21)$$

Clearly, the only optimum of this loss is the optimal control u^*. Using the analytic expression of u^* in (9), the cross-term can be rewritten as (see Lemma 5 in App. C):

$$\begin{aligned} &\mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \langle u(X_t, t), u^*(X_t, t)\rangle \, \mathrm{d}t \, \exp\big(- \lambda^{-1} \int_0^T f(X_t, t) \, \mathrm{d}t - \lambda^{-1} g(X_T)\big)\big] \\ &= \lambda \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\langle u(X_t, t), \sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(- \lambda^{-1} \int_t^T f(X_s, s) \, \mathrm{d}s - \lambda^{-1} g(X_T)\big)\big| X_t = x\big]\big\rangle \\ &\qquad \times \exp\big(- \lambda^{-1} \textstyle\int_0^t f(X_s, s) \, \mathrm{d}s\big) \, \mathrm{d}t\big]. \end{aligned} \qquad (22)$$

It remains to evaluate the conditional expectation $\nabla_x \mathbb{E}\big[\exp\big(- \lambda^{-1} \int_t^T f(X_s, s) \, \mathrm{d}s - \lambda^{-1} g(X_T)\big)\big| X_t = x\big]$, which we do by a “reparameterization trick” that shifts the dependence on the initial value x into the stochastic processes—here we introduce a free variable M_t—and then applying Girsanov theorem. We coin this the *path-wise reparameterization trick*:

Proposition 1 (Path-wise reparameterization trick for stochastic optimal control). *For each $t \in [0, T]$, let $M_t : [t, T] \to \mathbb{R}^{d\times d}$ be an arbitrary continuously differentiable function matrix-valued function* *such that* $M_t(t) = \mathrm{Id}$. *We have that*

$$
\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big] \\
&= \mathbb{E}\big[\big(-\lambda^{-1}\textstyle\int_t^T M_t(s)\nabla_x f(X_s,s)\,\mathrm{d}s - \lambda^{-1} M_t(T)\nabla g(X_T) \\
&\qquad + \lambda^{-1/2}\textstyle\int_t^T (M_t(s)\nabla_x b(X_s,s) - \partial_s M_t(s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big) \\
&\qquad \times \exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big].
\end{aligned}
\tag{23}
$$

We prove a more general form of this result (Prop. 4) in Subsec. C.2 and also provide an intuitive derivation in Subsec. C.3. In the proof of Prop. 4, the reparameterization matrices M_t arise as the gradients of a perturbation to the process X_t. Similar ideas can potentially be applied to derive losses for generative modeling. If we plug (23) into the right-hand side of (22), and then this back into (21), and we complete the square, we obtain that for some constant K independent of u,

$$
\begin{aligned}
\tilde{\mathcal{L}}(u) = \mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\|u(X_t,t) + \sigma(t)\big(\int_t^T M_t(s)\nabla_x f(X_s,s)\,\mathrm{d}s + M_t(T)\nabla g(X_T) \\
- \lambda^{1/2}\textstyle\int_t^T (M_t(s)\nabla_x b(X_s,s) - \partial_s M_t(s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big)\big\|^2\,\mathrm{d}t \\
\times \exp\big(-\lambda^{-1}\textstyle\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1} g(X_T)\big)\big] + K.
\end{aligned}
$$

If we perform a change of process from X to X^v applying the Girsanov theorem (Cor. 1 in App. C), we obtain the loss $\mathcal{L}_{\mathrm{SOCM}}(u,M)$. □

The following result clarifies the role of reparameterization matrices, connecting the SOCM and cross-entropy losses.

Proposition 2 (Bias-variance decomposition of the SOCM loss)**.** *The SOCM loss decomposes into a bias term that only depends on u and a variance term that only depends on M:*

$$
\mathcal{L}_{\mathrm{SOCM}}(u,M) = \underbrace{\mathrm{CondVar}(w;M)}_{\substack{\textit{Unnormalized expected}\\ \textit{conditional variance of } w}} + \underbrace{\mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\|u(X_t^{u^*},t) - u^*(X_t^{u^*},t)\big\|^2\,\mathrm{d}t\, e^{-\lambda^{-1}V(X_0^{u^*},0)}\big]}_{\textit{Unnormalized bias of } u}, \tag{24}
$$

where

$$
\mathrm{CondVar}(w;M) = \mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\|w(t,v,X^v,B,M_t) - \underbrace{\tfrac{\mathbb{E}[w(t,v,X^v,B,M_t)\alpha(v,X^v,B)|X_t^v,t]}{\mathbb{E}[\alpha(v,X^v,B)|X_t^v,t]}}_{u^*(X_t^v,t)}\big\|^2\,\mathrm{d}t\,\alpha(v,X^v,B)\big]. \tag{25}
$$

Remark that the bias term in equation (24) is equal to the characterization of the cross-entropy loss in (14). In other words, the landscape of $\mathcal{L}_{\mathrm{SOCM}}(u,M)$ with respect to u is the landscape of the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}(u)$. Thus, the SOCM loss can be seen as some form of variance reduction method for the cross-entropy loss, and performs substantially better experimentally (Sec. 4). Yet, the expressions of the SOCM loss and the cross-entropy loss are very different; the former is a least squares loss and is expressed in terms of the gradients of the costs.

Algorithm 2 Stochastic Optimal Control Matching (SOCM)

```
Input: State cost $f(x,t)$, terminal cost $g(x)$, diffusion coeff. $\sigma(t)$, base drift $b(x,t)$, noise level $\lambda$, number
    of iterations $N$, batch size $m$, number of time steps $K$, initial control parameters $\theta_0$, initial matrix
    parameters $\omega_0$, loss $\mathcal{L}_{\mathrm{SOCM}}$ in (19)
for $n \in \{0,\dots,N-1\}$ do
    Simulate $m$ trajectories of the process $X^v$ controlled by $v = u_{\theta_n}$, e.g., using Euler-Maruyama updates
    Detach the $m$ trajectories from the computational graph, so that gradients do not backpropagate
    Using the $m$ trajectories, compute an $m$-sample Monte-Carlo approximation $\hat{\mathcal{L}}_{\mathrm{SOCM}}(u_{\theta_n}, M_{\omega_n})$ of the
      loss $\mathcal{L}_{\mathrm{SOCM}}(u_{\theta_n}, M_{\omega_n})$ in (19)
    Compute the gradients $\nabla_{(\theta,\omega)}\hat{\mathcal{L}}_{\mathrm{SOCM}}(u_{\theta_n}, M_{\omega_n})$ of $\hat{\mathcal{L}}_{\mathrm{SOCM}}(u_{\theta_n}, M_{\omega_n})$ at $(\theta_n, \omega_n)$
    Obtain $\theta_{n+1}, \omega_{n+1}$ with via an Adam update on $\theta_n, \omega_n$, resp.
end
Output: Learned control $u_{\theta_N}$
```

For good training performance, it is critical that the gradients have high signal-to-noise ratio. Looking at the SOCM loss, a good proxy for low gradient variance is to have low variance for $\frac{1}{T}\int_0^T \big\|u(X_t^v,t) - w(t,v,X^v,B,M_t)\big\|^2\,\mathrm{d}t \times \alpha(v,X^v,B)$, and this holds when both $\alpha(v,X^v,B)$ and $w(t,v,X^v,B,M_t)$ have low variance. Next, we present strategies to lower the variance of these two objects.

Minimizing the variance of the importance weight α We want to use a vector field v such that $\mathrm{Var}[\alpha(v, X^v, B)]$ is as low as possible. As shown by the following lemma, which is well-known in the literature, setting v to be the optimal control u^* actually achieves variance zero when we condition on the starting point of the controlled process X^v. The proof of this result can be found in Hartmann et al. [38], but we include it in Subsec. C.5 for completeness.

Lemma 1. *When we set $v = u^*$, the conditional variance $\mathrm{Var}[\alpha(v, X^v, B)|X_0^v = x_{\mathrm{init}}]$ is zero for any $x_{\mathrm{init}} \in \mathbb{R}^d$.*

Of course, we do not have access to the optimal control u^*, but it is still a good idea to set v as the closest vector field to u^* that we have access to, which is typically the currently learned control. In some instances, one may benefit from using a warm-started control parameterized as $u_{\mathrm{WS}}(x,t) + u_\theta(x,t)$, where the warm-start u_{WS} is a reasonably good control obtained via a different strategy (see App. E).

Minimizing the variance of the matching vector field w We are interested in finding the family $M = (M_t)_{t\in[0,T]}$ that minimizes the variance of $w(t, v, X^v, B, M_t)$ conditioned on t and X_t. Note that this is exactly the term $\mathrm{CondVar}(w; M)$ in the right-hand side of equation (24). Since $\mathrm{CondVar}(w; M)$ does not depend on the specific v, the optimal M does not depend on v either. And since the second term in the right-hand side of equation (24) does not depend on $M = (M_t)_{t\in[0,T]}$, minimizing $\mathrm{CondVar}(w; M)$ is equivalent to minimizing $\mathcal{L}(u)$ with respect to M.

Parameterizing the matrices M_t vs solving for the optimal matrices In practice, we parameterize the matrices $(M_t)_{t\in[0,T]}$ using a function M_ω with two arguments (t, s). To enforce that $M_\omega(t,t) = \mathrm{Id}$, we set $M_\omega(t,s) = e^{-\gamma(s-t)}\mathrm{Id} + (1 - e^{-\gamma(s-t)})\tilde{M}_{\tilde{\omega}}(t,s)$, where $\omega = (\gamma, \tilde{\omega})$, and $\tilde{M}_{\tilde{\omega}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{d\times d}$ is an unconstrained neural network. Alternatively, Thm. 4 in App. D shows that the optimal family $M^* = (M_t^*)_{t\in[0,T]}$ can be characterized as the solution of a linear equation in infinite dimensions (a Fredholm equation of the first kind). The discretized linear system has d^2K equations and variables, K being the number of discretization time points. However, since the optimal M^* does not depend on v (see Remark 1), this is a computation that must be done only once and that may be affordable in some settings. We did not test this approach experimentally.

4 Experiments

We consider four experimental settings that we adapt from Nüsken and Richter [59]: QUADRATIC ORNSTEIN UHLENBECK (EASY), QUADRATIC ORNSTEIN UHLENBECK (HARD), LINEAR ORNSTEIN UHLENBECK and DOUBLE WELL. We describe them in detail in App. F. For all of them, we have access to the ground-truth optimal control, which means that we are able to estimate the L^2 error incurred by the learned control u. In Figure 2 we plot the control L^2 error for each IDO algorithm described in Subsec. 2.2, and for the SOCM algorithm (Algorithm 2), for the QUADRATIC OU (EASY) and (HARD) settings. We also include two ablations of SOCM: *(i)* a version of SOCM where the reparameterization matrices M_t are set fixed to the identity I, *(ii)* SOCM-Adjoint, where we estimate the conditional expectation in equation (23) using the adjoint method for SDEs instead of the path-wise reparameterization trick (see Subsec. C.4). Code can be found at `https://github.com/facebookresearch/SOC-matching`.

At the end of training, SOCM obtains the lowest L^2 error, improving over all existing methods by a factor of around ten. The two SOCM ablations come in second and third by a substantial difference, which underlines the importance of the path-wise reparameterization trick. The best among existing methods is the adjoint method (the relative entropy loss). In Figure 2 (*bottom*) we show the squared norm of the gradient of each loss with respect to the parameters θ of the control: algorithms with small noise variance have low error values.

In Figure 3, we plot the control L^2 error for LINEAR ORNSTEIN UHLENBECK and DOUBLE WELL. For LINEAR OU, the error is around five times smaller for SOCM than for any existing method. For DOUBLE WELL, the SOCM algorithm achieves the third smallest error, slightly behind the variance loss and the adjoint method, but the latter shows instabilities. As we show in Figure 9 in App. F, these instabilities are inherent to the adjoint method and they do not disappear for small learning rates. Both in Figure 2 and Figure 3, we observe that learning the reparameterization matrices is critical to obtain gradient estimates with high signal-to-noise ratio. DOUBLE WELL is a particularly interesting and challenging setting because its solution is highly multimodal: g has 1024 modes. Multimodality is a feature observed in realistic settings, and is hard to handle because it involves learning the control correctly in each mode.

The costs f and g and the base drift b for QUADRATIC OU (HARD) are five times those of QUADRATIC OU (EASY). Consequently, the factor $\alpha(v, X^v, B)$ initially has a much larger variance for the SOCM methods, and for cross-entropy. As training progresses, u_{θ_n} gets closer to u^*, and consequently the variance of $\alpha(v, X^v, B)$ decreases, which in turn makes learning easier. This explains the initial slow decrease in the control error, followed by a fast drop that places SOCM well below existing algorithms. In App. E, we showcase a control warm-start strategy that can help and speed up convergence.

We also present experimental results on two-mode Gaussian mixture sampling in increasing dimension, using the Path Integral Sampler [84]. We take Gaussians with means that are 2 units apart, and identity variance. Figure 1 shows control objective estimates obtained after running the Adjoint, SOCM, and Cross-entropy algorithms for 40000 iterations, at dimensions $d = 2, 8, 16, 32, 64$, and error bars show standard errors. By Theorem 4 of [84], we know that the optimal value of the control objective is zero; Figure 1 shows the suboptimality gaps incurred by each algorithm. Cross-entropy, which uses the same importance weight as SOCM, performs worse than the other two losses for all dimensions, and its results are particularly poor for dimension 64, because the variance of α is too large for learning to happen. In this case, we see that SOCM has better variance reduction than cross-entropy, despite both using importance weighted objectives for training. We observe that the values for SOCM are slightly below that of Adjoint for most dimensions, which confirms that our method is better for this range of dimensions. If we keep increasing the dimension, SOCM also fails due to higher variance of α: for $n = 128$, the control objective estimates for the Adjoint, SOCM, and Cross-Entropy losses are 0.146 ± 0.001, 7.49 ± 0.01, and 12.61 ± 0.02, respectively.

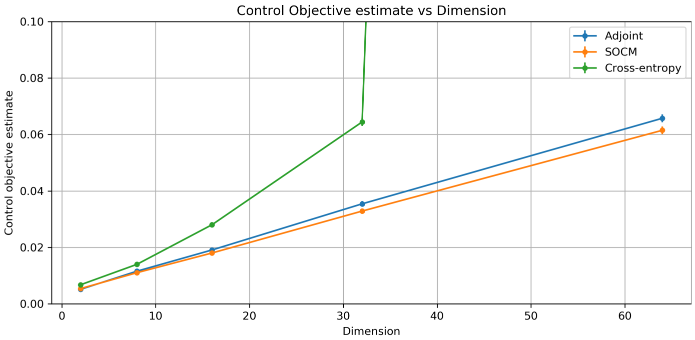

Figure 1: This plot shows the control objective values for different algorithms (Adjoint, SOCM, and Cross-entropy) across multiple dimensions, with error bars indicating the standard deviations. The y-axis is restricted to [0, 0.1] for better visibility of the lower range values; cross-entropy takes value 2.915 ± 0.008 at $d = 64$.

5 Conclusion

Our work introduces Stochastic Optimal Control Matching, a novel Iterative Diffusion Optimization technique for stochastic optimal control that stems from the same philosophy as the conditional score matching loss for diffusion models. That is, the control is learned via a least-squares problem by trying to fit a matching vector field. The training loss is optimized with respect to both the control function and a family of reparameterization matrices which appear in the matching vector field. Optimizing the reparameterization matrices reduces the variance of the matching vector field. Experimentally, our algorithm achieves lower error than all existing IDO techniques in four settings.

One of the key ideas for deriving the SOCM algorithm is the path-wise reparameterization trick, a novel technique to obtain low-variance estimates of the gradient of the conditional expectation of a functional of a random process with respect to its initial value. An interesting future direction is to use the path-wise reparameterization trick to decrease the variance of the matching vector field for diffusion models. The main roadblock when we try to apply SOCM to more challenging problems is that the variance of the factor $\alpha(v, X^v, B)$ explodes when f and/or g are large, or when the dimension d is high. The large variance of α is due to the mismatch between the probability measures induced by the learned control and the optimal control, and it decreases as the learned control approaches the optimal control.

The research presented is foundational, but it may serve as the basis of algorithms that improve the quality of generative models.

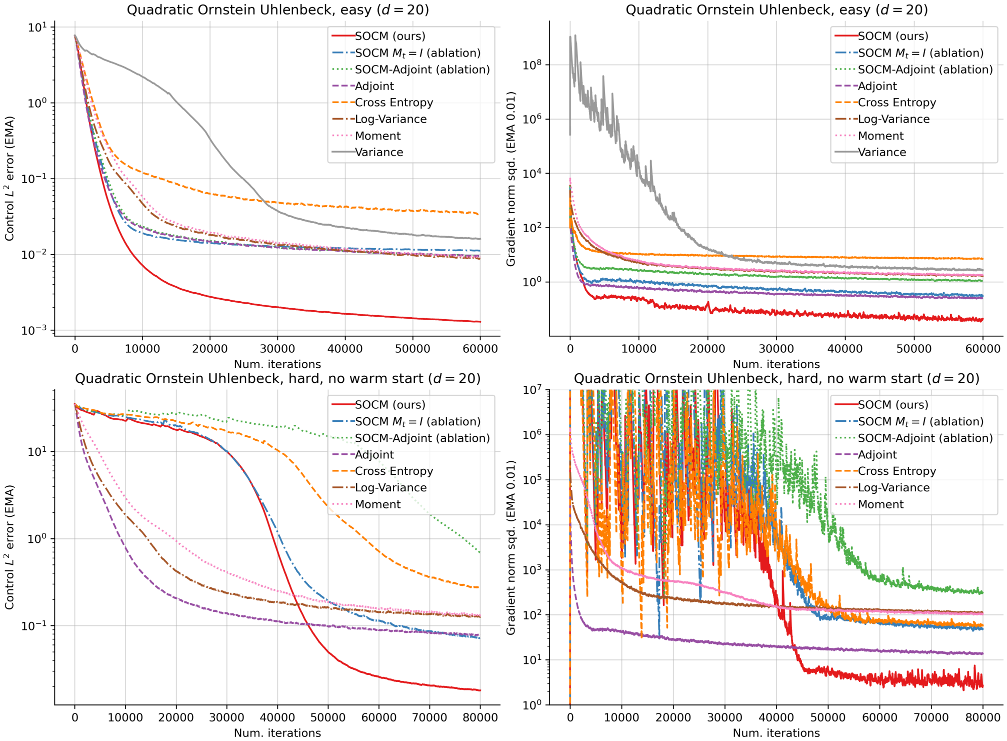

Figure 2: Plots of the L^2 error incurred by the learned control (*left*), and the norm squared of the gradient with respect to the parameters θ of the control (*right*), for the QUADRATIC ORNSTEIN UHLENBECK (EASY) (*top*) and (HARD) (*bottom*) settings and for each IDO loss. Both plots show exponential moving averages.

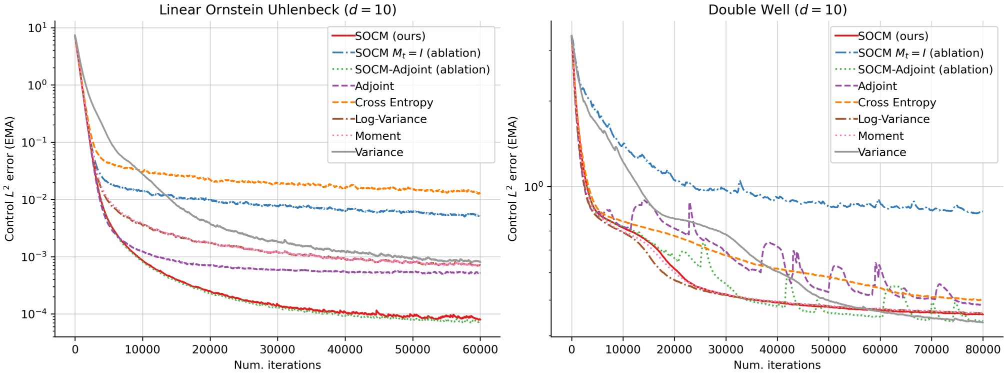

Figure 3: Plots of the L^2 error of the learned control for the LINEAR ORNSTEIN UHLENBECK and DOUBLE WELL settings.

Funding disclosure Funded by respective author affiliations.

References

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

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

[3] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[4] L'ubomír Baňas, Herbert Dawid, Tsiry Avisoa Randrianasolo, Johannes Storn, and Xin Wen. Numerical approximation of a system of hamilton–jacobi–bellman equations arising in innovation dynamics. *Journal of Scientific Computing*, 92, 2022.

[5] Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, and Arnulf Jentzen. Solving stochastic differential equations and Kolmogorov equations by means of deep learning. *arXiv:1806.00421*, 2018.

[6] Christian Beck, Fabian Hornung, Martin Hutzenthaler, Arnulf Jentzen, and Thomas Kruse. Overcoming the curse of dimensionality in the numerical approximation of Allen-Cahn partial differential equations via truncated full-history recursive multilevel Picard approximations. *arXiv:1907.06729*, 2019.

[7] Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20, 2019.

[8] Andrea Belloni, Luigi Piroddi, and Maria 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.

[9] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling, 2023.

[10] Joris Bierkens and Hilbert J Kappen. Explicit solution of relative entropy weighted control. *Systems & Control Letters*, 72:36–43, 2014.

[11] J. Bonnans, Elisabeth Ottenwaelter, and Hasnaa Zidani. A fast algorithm for the two dimensional hjb equation of stochastic control. *M2AN. Mathematical Modelling and Numerical Analysis. ESAIM, European Series in Applied and Industrial Mathematics*, 38, 07 2004.

[12] Elisa Calzola, Elisabetta Carlini, Xavier Dupuis, and Francisco Silva. A semi-Lagrangian scheme for Hamilton–Jacobi–Bellman equations with oblique derivatives boundary conditions. *Numerische Mathematik*, page 153, 2022.

[13] E. Carlini, A. Festa, and N. Forcadel. A semi-Lagrangian scheme for Hamilton–Jacobi–Bellman equations on networks. *SIAM J. Numer. Anal.*, 58(6):3165–3196, 2020.

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

[15] René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: The ergodic case. *SIAM Journal on Numerical Analysis*, 59(3):1455–1485, 2021.

[16] René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: Ii—the finite horizon case. *The Annals of Applied Probability*, 32(6):4065–4105, 2022.

[17] René Carmona, François Delarue, et al. *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer, 2018.

[18] Quentin Chan-Wai-Nam, Joseph Mikael, and Xavier Warin. Machine learning for semilinear PDEs. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.

[19] Pratik Chaudhari, Adam Oberman, Stanley Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):30, 2018.

[20] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[21] Kristian Debrabant and Espen R. Jakobsen. Semi-lagrangian schemes for linear and fully non-linear diffusion equations. *Mathematics of Computation*, 82(283):1433–1462, 2013.

[22] W. E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[23] Weinan E, Jiequn Han, and Arnulf Jentzen. Algorithms for solving high dimensional pdes: from nonlinear monte carlo to machine learning. *Nonlinearity*, 35(1):278, 2021.

[24] Jin Feng and Thomas G Kurtz. *Large deviations for stochastic processes*. Number 131. American Mathematical Soc., 2006.

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

[26] Paul Glasserman and David D. Yao. Some guidelines and guarantees for common random numbers. *Manage. Sci.*, 38(6):884–908, jun 1992.

[27] Peter W. Glynn. Stochastic approximation for monte carlo optimization. In *Proceedings of the 18th Conference on Winter Simulation*, page 356–365. Association for Computing Machinery, 1986.

[28] Emmanuel Gobet. *Monte-Carlo methods and stochastic processes: from linear to non-linear*. CRC Press, 2016.

[29] Emmanuel Gobet and Rémi Munos. Sensitivity analysis using Itô–Malliavin calculus and martingales, and application to stochastic optimal control. *SIAM Journal on control and optimization*, 43(5):1676–1713, 2005.

[30] Emmanuel Gobet, Jean-Philippe Lemor, Xavier 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.

[31] Vicenç Gómez, Hilbert J Kappen, Jan Peters, and Gerhard Neumann. Policy search for path integral control. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 482–497. Springer, 2014.

[32] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. High-dimensional stochastic optimal control using continuous tensor decompositions. *International Journal of Robotics Research*, 37(2-3), 3 2018.

[33] Constantin Greif. Numerical methods for hamilton-jacobi-bellman equations. 2017.

[34] Jiequn Han and Ruimeng Hu. Deep fictitious play for finding markovian nash equilibrium in multi-agent games. In *Mathematical and scientific machine learning*, pages 221–245. PMLR, 2020.

[35] Jiequn Han, Arnulf Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

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

[37] Carsten Hartmann, Ralf Banisch, Marco Sarich, Tomasz Badowski, and Christof Schütte. Characterization of rare events in molecular dynamics. *Entropy*, 16(1):350–376, 2014.

[38] Carsten Hartmann, Lorenz Richter, Christof Schütte, and Wei Zhang. Variational characterization of free energy: Theory and algorithms. *Entropy*, 19(11), 2017.

[39] Carsten Hartmann, Omar Kebiri, Lara Neureither, and Lorenz Richter. Variational approach to rare event simulation using least-squares regression. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6):063107, 2019.

[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

[41] 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.

[42] James E. Hutton and Paul I. Nelson. Interchanging the order of differentiation and stochastic integration. *Stochastic Processes and their Applications*, 18(2):371–377, 1984.

[43] Martin Hutzenthaler and Thomas Kruse. Multilevel picard approximations of high-dimensional semilinear parabolic differential equations with gradient-dependent nonlinearities. *SIAM Journal on Numerical Analysis*, 58(2):929–961, 2020.

[44] Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, et al. Multilevel picard iterations for solving smooth semilinear parabolic heat equations. *arXiv preprint arXiv:1607.03295*, 2016.

[45] Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, Tuan Anh Nguyen, and Philippe von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *arXiv:1807.01212*, 2018.

[46] Martin Hutzenthaler, Arnulf Jentzen, and Thomas Kruse. Overcoming the curse of dimensionality in the numerical approximation of parabolic partial differential equations with gradient-dependent nonlinearities. *arXiv:1912.02571*, 2019.

[47] Max Jensen and Iain Smears. On the convergence of finite element methods for hamilton–jacobi–bellman equations. *SIAM Journal on Numerical Analysis*, 51(1):137–162, 2013.

[48] Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

[49] Hilbert Johan Kappen and Hans Christian Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162(5):1244–1266, 2016.

[50] I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Graduate Texts in Mathematics (113) (Book 113). Springer New York, 1991.

[51] Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning*, 2021.

[52] Harold Kushner and Paul G Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2013.

[53] Pierre L'Ecuyer and Gaétan Perron. On the convergence rates of ipa and fdc derivative estimators. *Operations Research*, 42(4):643–656, 1994.

[54] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

[55] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2022.

[56] Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos A. Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching, 2023.

[57] Jingtang Ma and Jianjun Ma. Finite difference methods for the hamilton-jacobi-bellman equations arising in regime switching utility maximization. *J. Sci. Comput.*, 85(3):55, 2020.

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

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

[60] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[61] Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach for high-dimensional optimal control applied to multiagent path finding. *IEEE Transactions on Control Systems Technology*, 31(1):235–251, jan 2023.

[62] Grigorios A Pavliotis. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014.

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

[64] L.S. Pontryagin. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.

[65] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching with optimal transport couplings. In *International Conference on Machine Learning*, 2023.

[66] Warren B. Powell and Stephan Meisel. Tutorial on stochastic optimization in energy—part i: Modeling and policies. *IEEE Transactions on Power Systems*, 31(2):1459–1467, 2016.

[67] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

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

[69] Martin I. Reiman and Alan Weiss. Sensitivity analysis for simulations via likelihood ratios. *Oper. Res.*, 37:830–844, 1989.

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

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

[72] Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[73] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[74] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[75] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

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

[77] Evangelos Theodorou, Freek Stulp, Jonas Buchli, and Stefan Schaal. An iterative path integral stochastic optimal control approach for learning robotic tasks. *IFAC Proceedings Volumes*, 44 (1):11594–11601, 2011. 18th IFAC World Congress.

[78] Ramon Van Handel. Stochastic calculus, filtering, and stochastic control. *Course notes, URL http://www. prince- ton. edu/rvan/acm217/ACM217*, 2007.

[79] Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023.

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

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

[82] Jichuan Yang and Harold J. Kushner. A monte carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems. *SIAM Journal on Control and Optimization*, 29(5):1216–1249, 1991.

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

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

[85] Wei Zhang, Han Wang, Carsten Hartmann, Marcus Weber, and Christof 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.

[86] Mo Zhou, Jiequn Han, and Jianfeng Lu. Actor-critic method for high dimensional static Hamilton–Jacobi–Bellman partial differential equations based on neural networks. *SIAM Journal on Scientific Computing*, 43(6):A4043–A4066, 2021.

Contents

1 Introduction **1**

2 Framework **2**
2.1 Setup and Preliminaries 2
2.2 Existing approaches and related work 4

3 Stochastic Optimal Control Matching **6**

4 Experiments **8**

5 Conclusion **9**

A Technical assumptions **16**

B Proofs of Sec. 2 **16**

C Proofs of Sec. 3 **21**
C.1 Proof of Thm. 1 and Prop. 2 21
C.2 Proof of the path-wise reparameterization trick (Prop. 1) 24
C.3 Informal derivation of the path-wise reparameterization trick 26
C.4 SOCM-Adjoint: replacing the path-wise reparameterization trick with the adjoint method 28
C.5 Proof of Lemma 1 30

D Optimal reparameterization matrices **30**

E Control warm-starting **32**

F Experimental details and additional plots **34**
F.1 Experimental details 34
F.2 Model architectures 35
F.3 Additional tables and plots 35

A Technical assumptions

Throughout our work, we make the same assumptions as [59], which are needed for all the objects considered to be well-defined. Namely, we assume that:

(i) The set $\mathcal{U}$ of *admissible controls* is given by

$$\mathcal{U} = \{u \in C^1(\mathbb{R}^d \times [0,T]; \mathbb{R}^d) \,|\, \exists C > 0,\, \forall (x,s) \in \mathbb{R}^d \times [0,T],\, b(x,s) \leq C(1+|x|)\}.$$

(ii) The coefficients b and σ are continuously differentiable, σ has bounded first-order spatial derivatives, and $(\sigma\sigma^\top)(x,s)$ is positive definite for all $(x,s) \in \mathbb{R}^d \times [0,T]$. Furthermore, there exist constants $C, c_1, c_2 > 0$ such that

$$\|b(x,s)\| \leq C(1+\|x\|), \qquad \text{(linear growth)}$$
$$c_1\|\xi\|^2 \leq \xi^\top(\sigma\sigma^\top)(x,s)\xi \leq c_2\|\xi\|^2, \qquad \text{(ellipticity)}$$

for all $(x,s) \in \mathbb{R}^d \times [0,T]$ and $\xi \in \mathbb{R}^d$.

B Proofs of Sec. 2

Proof of (5) By Itô's lemma, we have that

$$\begin{aligned} V(X_T^u,T) - V(X_t^u,t) = \textstyle\int_t^T \big(&\partial_s V(X_s^u,s) + \langle b(X_s^u,s) + \sigma(X_s^u,s)u(X_s^u,s), \nabla V(X_s^u,s)\rangle \\ &+ \tfrac{\lambda}{2}\textstyle\sum_{i,j=1}^d (\sigma\sigma^\top)_{ij}(X_s^u,s)\partial_{x_i}\partial_{x_j}V(X_s^u,s)\big)\,\mathrm{d}s + S_t^u, \end{aligned}$$

where $S_t^u = \sqrt{\lambda} \int_t^T \nabla V(X_s^u, s)^\top \sigma(X_s^u, s) \, \mathrm{d}B_s$. Note that by (4),

$$\begin{aligned}
&\partial_s V(X_s^u, s) + \langle b(X_s^u, s) + \sigma(X_s^u, s) u(X_s^u, s), \nabla V(X_s^u, s) \rangle \\
&\quad + \tfrac{\lambda}{2} \textstyle\sum_{i,j=1}^d (\sigma \sigma^\top)_{ij}(X_s^u, s) \partial_{x_i} \partial_{x_j} V(X_s^u, s) \\
&= \tfrac{1}{2} \|(\sigma^\top \nabla V)(X_s^u, s)\|^2 - f(X_s^u, s) + \langle \sigma(X_s^u, s) u(X_s^u, s), \nabla V(X_s^u, s) \rangle \\
&= \tfrac{1}{2} \|(\sigma^\top \nabla V)(X_s^u, s) + u(X_s^u, s)\|^2 - \tfrac{1}{2} \|u(X_s^u, s)\|^2 - f(X_s^u, s),
\end{aligned}$$

and this implies that

$$g(X_T^u) - V(X_t^u, t) = \textstyle\int_t^T \big(\tfrac{1}{2} \|(\sigma^\top \nabla V)(X_s^u, s) + u(X_s^u, s)\|^2 - \tfrac{1}{2} \|u(X_s^u, s)\|^2 - f(X_s^u, s) \big) \, \mathrm{d}s + S_t^u \tag{26}$$

Since $\mathbb{E}[S_t^u \mid X_t^u = x] = 0$, rearranging (26) and taking the conditional expectation with respect to X_t^u yields the final result.

Proof of (6)-(7) By Itô's lemma, we have that

$$\begin{aligned}
dV(X_s, s) &= \big(\partial_s V(X_s, s) + \langle b(X_s, s), \nabla V(X_s, s) \rangle \\
&\quad + \tfrac{\lambda}{2} \textstyle\sum_{i,j=1}^d (\sigma \sigma^\top)_{ij}(X_s, s) \partial_{x_i} \partial_{x_j} V(X_s, s) \big) \, \mathrm{d}s + \sqrt{\lambda} \nabla V(X_s^u, s)^\top \sigma(X_s^u, s) \, \mathrm{d}B_s,
\end{aligned} \tag{27}$$

Note that by (4),

$$\begin{aligned}
&\partial_s V(X_s, s) + \langle b(X_s, s), \nabla V(X_s, s) \rangle + \tfrac{\lambda}{2} \textstyle\sum_{i,j=1}^d (\sigma \sigma^\top)_{ij}(X_s, s) \partial_{x_i} \partial_{x_j} V(X_s, s) \\
&= \tfrac{1}{2} \|(\sigma^\top \nabla V)(X_s, s)\|^2 - f(X_s, s).
\end{aligned}$$

Plugging this into (27) concludes the proof.

Proof of (8) Since $Y_s = V(X_s, s)$ and $Z_s = \sigma^\top(s) \nabla V(X_s, s) = -u^*(X_s, s)$ satisfy (7), we have that

$$g(X_T) = Y_T = Y_t - \textstyle\int_t^T (f(X_s, s) - \tfrac{1}{2} \|u^*(X_s, s)\|^2) \, \mathrm{d}s - \sqrt{\lambda} \int_t^T \langle u^*(X_s, s), \mathrm{d}B_s \rangle.$$

Hence, recalling the definition of the work functional in (10), we have that

$$\mathcal{W}(X, t) = Y_t + \tfrac{1}{2} \textstyle\int_t^T \|u^*(X_s, s)\|^2 \, \mathrm{d}s - \sqrt{\lambda} \int_t^T \langle u^*(X_s, s), \mathrm{d}B_s \rangle. \tag{28}$$

By Novikov's theorem (Thm. 2), we have that

$$\begin{aligned}
&\mathbb{E}[\exp(-\lambda^{-1} \mathcal{W}(X, t)) | X_t] \\
&= e^{-\lambda^{-1} Y_t} \mathbb{E}\big[\exp\big(\lambda^{-1/2} \textstyle\int_t^T \langle u^*(X_s, s), \mathrm{d}B_s \rangle - \tfrac{\lambda^{-1}}{2} \int_t^T \|u^*(X_s, s)\|^2 \, \mathrm{d}s \big) \big| X_t \big] = e^{-\lambda^{-1} Y_t},
\end{aligned}$$

which concludes the proof of (8).

Theorem 2 (Novikov's theorem). *Let θ_s be a locally-$\mathcal{H}_2$ process which is adapted to the natural filtration of the Brownian motion $(B_t)_{t \geq 0}$. Define*

$$Z(t) = \exp\big(\textstyle\int_0^t \theta_s \, \mathrm{d}B_s - \tfrac{1}{2} \int_0^t \|\theta_s\|^2 \, \mathrm{d}s \big). \tag{29}$$

If for each $t \geq 0$,

$$\mathbb{E}\big[\exp\big(\textstyle\int_0^t \|\theta_s\|^2 \, \mathrm{d}s \big) \big] < +\infty,$$

then for each $t \geq 0$,

$$\mathbb{E}[Z(t)] = 1. \tag{30}$$

Moreover, the process $Z(t)$ is a positive martingale, i.e. if $(\mathcal{F}_t)_{t \geq 0}$ is the filtration associated to the Brownian motion $(B_t)_{t \geq 0}$, then for $t \geq s$, $\mathbb{E}[Z_t | \mathcal{F}_s] = Z_s$.

Theorem 3 (Girsanov theorem)**.** *Let $W = (W_t)_{t\in[0,T]}$ be a standard Wiener process, and let $\mathbb{P}$ be its induced probability measure over $C([0,T];\mathbb{R}^d)$, known as the Wiener measure. Let $Z(t)$ be as defined in* (29) *and suppose that the assumptions of Theorem 2 hold. Let $(\Omega,\mathcal{F})$ be the σ-algebra associated to B_T. For any $F\in\mathcal{F}$, define the measure*

$$\mathbb{Q}(F) = \mathbb{E}_{\mathbb{P}}[Z(T)\mathbf{1}_F]\,.$$

$\mathbb{Q}$ is a probability measure because of (30)*. Under the probability measure $\mathbb{Q}$, the stochastic process $\{\tilde W(t)\}_{0\le t\le T}$ defined as*

$$\tilde W(t) = W(t) - \textstyle\int_0^t \theta_s\,\mathrm{d}s$$

is a standard Wiener process. That is, for any $n\ge 0$ and any $0 = t_0 < t_1 < \cdots < t_n$, the increments $\{\tilde W(t_{i+1}) - \tilde W(t_i)\}_{i=0}^{n-1}$ are independent and $\mathbb{Q}$-Gaussian distributed with mean zero and covariance $(t_{i+1}-t_i)\mathrm{I}$, which means that for any $\alpha\in\mathbb{R}^d$, the moment generating function of $\tilde W(t_{i+1}) - \tilde W(t_i)$ with respect to $\mathbb{Q}$ is as follows:

$$\begin{aligned}&\mathbb{E}_{\mathbb{Q}}[\exp(\langle\alpha,\tilde W(t_{i+1}) - \tilde W(t_i)\rangle)]\\ &:= \mathbb{E}_{\mathbb{P}}\big[\exp\big(\big\langle\alpha, W(t_{i+1}) - \textstyle\int_0^{t_{i+1}}\theta_s\,\mathrm{d}s - W(t_i) + \int_0^{t_i}\theta_s\,\mathrm{d}s\big\rangle\big)Z(T)\big] = \exp\big(\frac{(t_{i+1}-t_i)\|\alpha\|^2}{2}\big).\end{aligned}$$

Corollary 1 (Girsanov theorem for SDEs)**.** *If the two SDEs*

$$\begin{aligned}\mathrm{d}X_t &= b_1(X_t,t)\,\mathrm{d}t + \sigma(X_t,t)\,\mathrm{d}B_t, \qquad X_0 = x_{\mathrm{init}}\\ dY_t &= (b_1(Y_t,t) + b_2(Y_t,t))\,\mathrm{d}t + \sigma(Y_t,t)\,\mathrm{d}B_t, \qquad Y_0 = x_{\mathrm{init}}\end{aligned}$$

admit unique strong solutions on $[0,T]$, then for any bounded continuous functional Φ on $C([0,T])$, we have that

$$\begin{aligned}\mathbb{E}[\Phi(X)] &= \mathbb{E}\big[\Phi(Y)\exp\big(-\textstyle\int_0^T\sigma(Y_t,t)^{-1}b_2(Y_t,t)\,\mathrm{d}B_t - \frac{1}{2}\int_0^T\|\sigma(Y_t,t)^{-1}b_2(Y_t,t)\|^2\,\mathrm{d}t\big)\big]\\ &= \mathbb{E}\big[\Phi(Y)\exp\big(-\textstyle\int_0^T\sigma(Y_t,t)^{-1}b_2(Y_t,t)\,d\tilde B_t + \frac{1}{2}\int_0^T\|\sigma(Y_t,t)^{-1}b_2(Y_t,t)\|^2\,\mathrm{d}t\big)\big],\end{aligned}$$

where $\tilde B_t = B_t + \int_0^t\sigma(Y_s,s)^{-1}b_2(Y_s,s)\,\mathrm{d}s$. More generally, b_1 and b_2 can be random processes that are adapted to filtration of B.

Lemma 2. *For an arbitrary $v\in\mathcal{U}$, let $\mathbb{P}^v$ and $\mathbb{P}$ be respectively the laws of the SDEs*

$$\begin{aligned}\mathrm{d}X_t^v &= (b(X_t^v,t) + \sigma(t)v(X_t^v,t))\,\mathrm{d}t + \sqrt{\lambda}\sigma(t)\mathrm{d}B_t, \qquad X_0^v\sim p_0,\\ \mathrm{d}X_t &= b(X_t,t)\,\mathrm{d}t + \sqrt{\lambda}\sigma(t)\mathrm{d}B_t, \qquad X_0\sim p_0.\end{aligned}$$

We have that

$$\begin{aligned}\tfrac{d\mathbb{P}}{d\mathbb{P}^v}(X^v) &= \exp\big(-\lambda^{-1/2}\textstyle\int_0^T\langle v(X_t^v,t),\mathrm{d}B_t^v\rangle + \frac{\lambda^{-1}}{2}\int_0^T\|v(X_t^v,t)\|^2\,\mathrm{d}t\big) \qquad (31)\\ &= \exp\big(-\lambda^{-1/2}\textstyle\int_0^T\langle v(X_t^v,t),\mathrm{d}B_t\rangle - \frac{\lambda^{-1}}{2}\int_0^T\|v(X_t^v,t)\|^2\,\mathrm{d}t\big),\\ \tfrac{d\mathbb{P}^v}{d\mathbb{P}}(X) &= \exp\big(\lambda^{-1/2}\textstyle\int_0^T\langle v(X_t,t),\mathrm{d}B_t\rangle - \frac{\lambda^{-1}}{2}\int_0^T\|v(X_t,t)\|^2\,\mathrm{d}t\big). \qquad (32)\end{aligned}$$

where $B_t^v := B_t + \lambda^{-1/2}\int_0^t v(X_s^v,s)\,\mathrm{d}s$. For the optimal control u^*, we have that

$$\tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^{u^*}) = \exp\big(\lambda^{-1}\big(-V(X_0^{u^*},0) + \mathcal{W}(X^{u^*},0)\big)\big), \qquad (33)$$

$$\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) = \exp\big(\lambda^{-1}\big(V(X_0,0) - \mathcal{W}(X,0)\big)\big), \qquad (34)$$

where the functional $\mathcal{W}$ is defined in (10)*.*

Proof. The proof of (31)-(32) follows directly from Cor. 1. To prove (34), we use that by (28),

$$\mathcal{W}(X,0) = V(X_0,0) + \tfrac{1}{2}\textstyle\int_0^T\|u^*(X_s,s)\|^2\,\mathrm{d}s - \sqrt{\lambda}\int_0^T\langle u^*(X_s,s),\mathrm{d}B_s\rangle, \qquad (35)$$

which implies that

$$\begin{aligned}\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) &= \exp\big(\lambda^{-1/2}\textstyle\int_0^T\langle u^*(X_t,t),\mathrm{d}B_t\rangle - \frac{\lambda^{-1}}{2}\int_0^T\|u^*(X_t,t)\|^2\,\mathrm{d}t\big)\\ &= \exp\big(\lambda^{-1}\big(V(X_0,0) - \mathcal{W}(X,0)\big)\big).\end{aligned}$$

To prove (33), we use that since $\mathrm{d}X_t^{u^*} = b(X_t^{u^*}, t)\,\mathrm{d}t + \sqrt{\lambda}\sigma(t)\mathrm{d}B_t^{u^*}$, equation (35) holds if we replace X and B by X^{u^*} and B^{u^*}, which reads

$$\mathcal{W}(X^{u^*}, 0) = V(X_0^{u^*}, 0) + \tfrac{1}{2}\int_t^T \|u^*(X_s^{u^*}, s)\|^2\,\mathrm{d}s - \sqrt{\lambda}\int_t^T \langle u^*(X_s^{u^*}, s), \mathrm{d}B_s^v\rangle.$$

Hence,

$$\begin{aligned}\tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^{u^*}) &= \exp\big(- \lambda^{-1/2}\int_0^T \langle u^*(X_t^{u^*}, t), \mathrm{d}B_t^{u^*}\rangle + \tfrac{\lambda^{-1}}{2}\int_0^T \|u^*(X_t^{u^*}, t)\|^2\,\mathrm{d}t\big) \\ &= \exp\big(\lambda^{-1}\big(- V(X_0^{u^*}, 0) + \mathcal{W}(X^{u^*}, 0)\big)\big).\end{aligned}$$

□

Lemma 3. *The following expression holds:*

$$\mathbb{E}_{\mathbb{P}^u}\big[\log \tfrac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}\big] = \lambda^{-1}\mathbb{E}\big[\int_0^T \big(\tfrac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\big)\,\mathrm{d}t + g(X_T^u) - V(X_0^u, 0)\big], \quad (36)$$

Proof. To prove (36), we write

$$\begin{aligned}\log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^u) &= \log\big(\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^u)\tfrac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)\big) = \log\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^u) + \log\tfrac{d\mathbb{P}}{d\mathbb{P}^u}(X^u) \\ &= \lambda^{-1}\big(V(X_0^u, 0) - \int_0^T f(X_t^u, t)\,\mathrm{d}t - g(X_T^u)\big) \\ &\quad - \lambda^{-1/2}\int_0^T \langle u(X_t^u, t), \mathrm{d}B_t\rangle - \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^u, t)\|^2\,\mathrm{d}t\ .\end{aligned}$$

Since $\mathbb{E}_{\mathbb{P}^u}\big[\log\tfrac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}\big] = -\mathbb{E}_{\mathbb{P}^u}\big[\log\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}\big]$, and $\mathbb{E}_{\mathbb{P}^u}\big[\int_0^T \langle u(X_t^u, t), \mathrm{d}B_t\rangle\big] = 0$, the result follows.

□

Proposition 3. *(i) The following two expressions hold for arbitrary controls u, v in the class $\mathcal{U}$ of admissible controls:*

$$\begin{aligned}\tilde{\mathcal{L}}_{\mathrm{CE}}(u) = \mathbb{E}_{\mathbb{P}^{u^*}}\big[\log\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}\big] = \mathbb{E}\big[\big(&- \lambda^{-1/2}\int_0^T \langle u(X_t^v, t), \mathrm{d}B_t\rangle - \lambda^{-1}\int_0^T \langle u(X_t^v, t), v(X_t^v, t)\rangle\,\mathrm{d}t \\ &+ \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v, t)\|^2\,\mathrm{d}t + \lambda^{-1}\big(V(X_0^v, 0) - \mathcal{W}(X^v, 0)\big)\big) \\ &\times \exp\big(\lambda^{-1}\big(V(X_0^v, 0) - \mathcal{W}(X^v, 0)\big) \\ &\quad - \lambda^{-1/2}\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t\rangle - \tfrac{\lambda^{-1}}{2}\int_0^T \|v(X_t^v, t)\|^2\,\mathrm{d}t\big)\big],\end{aligned} \quad (37)$$

$$\tilde{\mathcal{L}}_{\mathrm{CE}}(u) = \tfrac{\lambda^{-1}}{2}\mathbb{E}\big[\int_0^T \|u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t)\|^2\,\mathrm{d}t\big]. \quad (38)$$

When p_0 is concentrated at a single point x_{init}, the terms $V(x_{\mathrm{init}}, 0)$ are constant and can be removed without modifying the landscape. In other words, $\tilde{\mathcal{L}}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{CE}}$ are equal up to constant terms and constant factors.

(ii) When p_0 is a generic probability measure, $\tilde{\mathcal{L}}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{CE}}$ have different landscapes, and $\mathcal{L}_{\mathrm{CE}}(u) = \mathbb{E}_{\mathbb{P}^{u^*}}\big[\log\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}\exp\big(- \lambda^{-1}V(X_0^{u^*}, 0)\big)\big]$. u^* is still the only minimizer of the loss $\mathcal{L}_{\mathrm{CE}}$, and for some constant K, we have that

$$\mathcal{L}_{\mathrm{CE}}(u, 0) = \tfrac{\lambda^{-1}}{2}\mathbb{E}\big[\int_0^T \|u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t)\|^2\,\mathrm{d}t\exp\big(- \lambda^{-1}V(X_0^{u^*}, 0)\big)\big] + K. \quad (39)$$

Proof. We begin with the proof of (i), and prove (37) first. Note that by the Girsanov theorem (Thm. 3),

$$\begin{aligned}\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^{u^*})\big] &= -\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log\tfrac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}(X^{u^*})\big] = -\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log\tfrac{d\mathbb{P}^u}{d\mathbb{P}}(X^{u^*}) + \log\tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^{u^*})\big] \\ &= -\mathbb{E}_{\mathbb{P}^v}\big[\big(\log\tfrac{d\mathbb{P}^u}{d\mathbb{P}}(X^v) + \log\tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^v)\big)\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^v)\tfrac{d\mathbb{P}}{d\mathbb{P}^v}(X^v)\big]\end{aligned} \quad (40)$$

Note that by equations (32) and (34),

$$\begin{aligned}\log\tfrac{d\mathbb{P}^u}{d\mathbb{P}}(X^v) &= \lambda^{-1/2}\int_0^T \langle u(X_t^v, t), \mathrm{d}B_t^v\rangle - \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v, t)\|^2\,\mathrm{d}t, \\ &= \lambda^{-1/2}\int_0^T \langle u(X_t^v, t), \mathrm{d}B_t\rangle + \lambda^{-1}\int_0^T \langle u(X_t^v, t), v(X_t^v, t)\rangle\,\mathrm{d}t - \tfrac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v, t)\|^2\,\mathrm{d}t, \\ \log\tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^v) &= \lambda^{-1}\big(- V(X_0^v, 0) + \mathcal{W}(X^v, 0)\big).\end{aligned} \quad (41)$$

where $B_t^v := B_t + \lambda^{-1/2} \int_0^t v(X_s^v, s) \, \mathrm{d}s$. Also,

$$\begin{aligned} &\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^v) = \exp\big(\lambda^{-1}\big(V(X_0^v, 0) - \mathcal{W}(X^v, 0)\big)\big), \\ &\tfrac{d\mathbb{P}}{d\mathbb{P}^v}(X^v) = \exp\big(- \lambda^{-1/2} \int_0^T \langle v(X_t^v, t), \mathrm{d}B_t \rangle - \tfrac{\lambda^{-1}}{2} \int_0^T \| v(X_t^v, t) \|^2 \, \mathrm{d}t \big). \end{aligned} \tag{42}$$

If we plug (41) and (42) into the right-hand side of (40), we obtain

$$\begin{aligned} &\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^{u^*})\big] = -\mathbb{E}_{\mathbb{P}^{u^*}}\big[(\log \tfrac{d\mathbb{P}^u}{d\mathbb{P}}(X^v) + \log \tfrac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^v)) \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^v) \tfrac{d\mathbb{P}}{d\mathbb{P}^u}(X^v)\big] \\ &= -\mathbb{E}\big[\big(\lambda^{-1/2} \int_0^T \langle u(X_t^v, t), \mathrm{d}B_t \rangle + \lambda^{-1} \int_0^T \langle u(X_t^v, t), v(X_t^v, t) \rangle \, \mathrm{d}t \\ &\qquad - \tfrac{\lambda^{-1}}{2} \int_0^T \| u(X_t^v, t) \|^2 \, \mathrm{d}t + \lambda^{-1}\big(- V(X_0^v, 0) + \mathcal{W}(X^v, 0)\big)\big) \\ &\qquad \times \exp\big(\lambda^{-1}\big(V(X_0^v, 0) - \mathcal{W}(X^v, 0)\big) - \lambda^{-1/2} \int_0^T \langle v(X_t^v, t), \mathrm{d}B_t \rangle - \tfrac{\lambda^{-1}}{2} \int_0^T \| v(X_t^v, t) \|^2 \, \mathrm{d}t\big)\big], \end{aligned}$$

which concludes the proof.

To show (38), we use that by Cor. 1,

$$\tfrac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}(X^{u^*}) = \exp\big(-\lambda^{-1/2} \int_0^T \langle u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t), \mathrm{d}B_t \rangle - \tfrac{\lambda^{-1}}{2} \int_0^T \| u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t) \|^2 \, \mathrm{d}t\big).$$

Hence,

$$\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}\big] = -\mathbb{E}_{\mathbb{P}^{u^*}}\big[\log \tfrac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}}\big] = \tfrac{\lambda^{-1}}{2} \mathbb{E}\big[\int_0^T \| u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t) \|^2 \, \mathrm{d}t\big].$$

Next, we prove (ii). The first instance of $V(X_0^v, 0)$ in (37) can be removed without modifying the landscape of the loss. Hence, we are left with

$$\begin{aligned} \bar{\mathcal{L}}_{\mathrm{CE}}(u) = \mathbb{E}\big[\big(&- \lambda^{-1/2} \int_0^T \langle u(X_t^v, t), \mathrm{d}B_t^v \rangle - \lambda^{-1} \int_0^T \langle u(X_t^v, t), v(X_t^v, t) \rangle \, \mathrm{d}t \\ &+ \tfrac{\lambda^{-1}}{2} \int_0^T \| u(X_t^v, t) \|^2 \, \mathrm{d}t - \lambda^{-1} \mathcal{W}(X^v, 0)\big) \\ &\times \exp\big(\lambda^{-1}\big(V(X_0^v, 0) - \mathcal{W}(X^v, 0)\big) - \lambda^{-1/2} \int_0^T \langle v(X_t^v, t), \mathrm{d}B_t \rangle - \tfrac{\lambda^{-1}}{2} \int_0^T \| v(X_t^v, t) \|^2 \, \mathrm{d}t\big)\big] \end{aligned} \tag{43}$$

And this can be expressed as

$$\bar{\mathcal{L}}_{\mathrm{CE}}(u) = \mathbb{E}\big[g(u; X_0^v) \exp\big(\lambda^{-1} V(X_0^v, 0)\big)\big],$$

where

$$\begin{aligned} g(u; x) = \mathbb{E}\big[\big(&- \lambda^{-1/2} \int_0^T \langle u(X_t^v, t), \mathrm{d}B_t^v \rangle - \lambda^{-1} \int_0^T \langle u(X_t^v, t), v(X_t^v, t) \rangle \, \mathrm{d}t \\ &+ \tfrac{\lambda^{-1}}{2} \int_0^T \| u(X_t^v, t) \|^2 \, \mathrm{d}t - \lambda^{-1} \mathcal{W}(X^v, 0)\big) \\ &\times \exp\big(- \lambda^{-1} \mathcal{W}(X^v, 0) - \lambda^{-1/2} \int_0^T \langle v(X_t^v, t), \mathrm{d}B_t \rangle - \tfrac{\lambda^{-1}}{2} \int_0^T \| v(X_t^v, t) \|^2 \, \mathrm{d}t\big) | X_0^v = x\big]. \end{aligned}$$

If we consider $g(u; x)$ as a loss function for u, note that it is equivalent to the loss $\bar{L}_{\mathrm{CE}}(u)$ equation in (43) for the choice $p_0 = \delta_x$, i.e., p_0 concentrated at x. Since the optimal control u^* is independent of the starting distribution p_0, we deduce that u^* is the unique minimizer of $g(u; x)$, for all $x \in \mathbb{R}^d$. In consequence, u^* is the unique minimizer of $\mathcal{L}_{\mathrm{CE}}(u) = \mathbb{E}[g(u; X_0^v)]$.

To prove (39), note that up to a constant term, the only difference between $\bar{\mathcal{L}}_{\mathrm{CE}}(u)$ and $\mathcal{L}_{\mathrm{CE}}(u)$ is the expectation is reweighted importance weight $\exp\big(- \lambda^{-1} V(X_0^v, 0)\big)$. □

Lemma 4. *(i) We can rewrite*

$$\begin{aligned} \tilde{\mathcal{L}}_{\mathrm{Var}_v}(u) &= \mathrm{Var}\big(\exp\big(\tilde{Y}_T^{u,v} - \lambda^{-1} g(X_T^v) + \lambda^{-1} V(X_0^v, 0)\big)\big), \\ \tilde{\mathcal{L}}_{\mathrm{Var}_v}^{\log}(u) &= \mathrm{Var}\big(\tilde{Y}_T^{u,v} - \lambda^{-1} g(X_T^v) + \lambda^{-1} V(X_0^v, 0)\big). \end{aligned}$$

When p_0 is concentrated at x_{init}, the terms $V(x_{\mathrm{init}}, 0)$ are constants and can be removed without modifying the landscape. In other words, $\tilde{\mathcal{L}}_{\mathrm{Var}_v}$ and $\tilde{\mathcal{L}}_{\mathrm{Var}_v}^{\log}$ are equal to $\mathcal{L}_{\mathrm{Var}_v}$ and $\mathcal{L}_{\mathrm{Var}_v}^{\log}$ up to a constant term and a constant factor, respectively.

(ii) When p_0 is general, $\tilde{\mathcal{L}}_{\mathrm{Var}_v}$ and $\mathcal{L}_{\mathrm{Var}_v}$ have a different landscape, and the optimum of $\mathcal{L}_{\mathrm{Var}_v}$ may be different from u^*. A related loss that does preserve the optimum is:

$$\begin{aligned}\bar{\mathcal{L}}_{\mathrm{Var}_v}(u) &= \mathbb{E}[\mathrm{Var}_{\mathbb{P}^v}(\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^v)|X_0^v)\exp(-\lambda^{-1}V(X_0^v,0))] \\ &= \mathbb{E}[\mathrm{Var}\big(\exp(\tilde{Y}_T^{u,v} - \lambda^{-1}g(X_T^v))|X_0^v\big)].\end{aligned}$$

In practice, this is implemented by sampling the m trajectories in one batch starting at the same point X_0^v.

(iii) Also, $\tilde{\mathcal{L}}_{\mathrm{Var}_v}^{\log}$ and $\mathcal{L}_{\mathrm{Var}_v}^{\log}$ have a different landscape, and the optimum of $\mathcal{L}_{\mathrm{Var}_v}^{\log}$ may be different from u^*. In particular, $\mathcal{L}_{\mathrm{Var}_v}^{\log}(u) = \mathrm{Var}_{\mathbb{P}^v}(\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^v)\exp(-\lambda^{-1}V(X_0^v,0)))$. A loss that does preserve the optimum u^* is

$$\begin{aligned}\bar{\mathcal{L}}_{\mathrm{Var}_v}^{\log}(u) &= \mathbb{E}[\mathrm{Var}_{\mathbb{P}^v}(\log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^v)|X_0^v)\exp(-\lambda^{-1}V(X_0^v,0))] \\ &= \mathbb{E}[\mathrm{Var}\big(\tilde{Y}_T^{u,v} - \lambda^{-1}g(X_T^v)|X_0^v\big)].\end{aligned}$$

Proof. Using (34) and (31), we have that

$$\begin{aligned}\tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^v) &= \exp\big(\lambda^{-1}\big(V(X_0^v,0) - \mathcal{W}(X^v,0)\big)\big), \\ \tfrac{d\mathbb{P}}{d\mathbb{P}^u}(X^v) &= \exp\big(-\lambda^{-1/2}\textstyle\int_0^T \langle u(X_t^v,t), \mathrm{d}B_t^v\rangle + \frac{\lambda^{-1}}{2}\int_0^T \|u(X_t^v,t)\|^2\,\mathrm{d}t\big) \\ &= \exp\big(-\lambda^{-1/2}\textstyle\int_0^T \langle u(X_t^v,t), \mathrm{d}B_t\rangle - \lambda^{-1}\int_0^T \langle u(X_t^v,t), v(X_t^v,t)\rangle\,\mathrm{d}t \\ &\qquad + \tfrac{\lambda^{-1}}{2}\textstyle\int_0^T \|u(X_t^v,t)\|^2\,\mathrm{d}t\big).\end{aligned}$$

Hence,

$$\log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^v) = \log \tfrac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^v) + \log \tfrac{d\mathbb{P}}{d\mathbb{P}^u}(X^v) = \tilde{Y}_T^{u,v} - \lambda^{-1}g(X_T^v) + \lambda^{-1}V(X_0^v,0).$$

Since $\tilde{\mathcal{L}}_{\mathrm{Var}_v}(u) = \mathrm{Var}_{\mathbb{P}^v}(\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u})$ and $\tilde{\mathcal{L}}_{\mathrm{Var}_v}^{\log}(u) = \mathrm{Var}_{\mathbb{P}^v}(\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u})$, this concludes the proof of (i).

To prove (ii), note that for general p_0, $V(X_0^v,0)$ is no longer a constant, but it is if we condition on X_0^v. The proof of (iii) is analogous. □

C Proofs of Sec. 3

C.1 Proof of Thm. 1 and Prop. 2

We prove Thm. 1 and Prop. 2 at the same time. Recall that by (9), the optimal control is of the form $u^*(x,t) = -\sigma(t)^\top \nabla V(x,t)$. Consider the loss

$$\tilde{\mathcal{L}}(u) = \mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\|u(X_t,t) + \sigma(t)^\top \nabla V(X_t,t)\big\|^2\,\mathrm{d}t\,\exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big].$$

Clearly, the unique optimum of $\tilde{\mathcal{L}}$ is $-\sigma(t)^\top \nabla V$. We can rewrite $\tilde{\mathcal{L}}$ as

$$\begin{aligned}\tilde{\mathcal{L}}(u) = \mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big(\big\|u(X_t,t)\big\|^2 + 2\langle u(X_t,t), \sigma(t)^\top \nabla V(X_t,t)\rangle + \|\sigma(t)^\top \nabla V(X_t,t)\big\|^2\big)\,\mathrm{d}t \\ \times \exp\big(-\lambda^{-1}\textstyle\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big]\ .\end{aligned} \tag{44}$$

Hence, we can express $\tilde{\mathcal{L}}$ as a sum of three terms: one involving $\|u(X_t,t)\|^2$, another involving $\langle u(X_t,t), \sigma(t)^\top V(X_t,t)\rangle$, and a third one, which is constant with respect to u, involving $\|\nabla V(X_t,t)\|^2$. The following lemma provides an alternative expression for the cross term:

Lemma 5. *The following equality holds:*

$$\begin{aligned}&\mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \langle u(X_t,t), \sigma(t)^\top \nabla V(X_t,t)\rangle\,\mathrm{d}t\,\exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big] \\ &= -\lambda\mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\langle u(X_t,t), \sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big]\big\rangle \\ &\qquad \times \exp\big(-\lambda^{-1}\textstyle\int_0^t f(X_s,s)\,\mathrm{d}s\big)\,\mathrm{d}t\big].\end{aligned} \tag{45}$$

Proof. Recall the definition of $\mathcal{W}(X,t)$ in (35), which means that

$$\mathcal{W}(X,0) = \mathcal{W}(X,t) + \int_0^t f(X_s,s)\,\mathrm{d}s. \tag{46}$$

Let $\{\mathcal{F}_t\}_{t\in[0,T]}$ be the filtration generated by the Brownian motion B. Then, equation (9) implies that

$$\sigma(t)^\top \nabla V(X_t,t) = -\frac{\lambda\sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big]}{\mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big]} \tag{47}$$

We proceed as follows:

$$\begin{aligned}
&\mathbb{E}\big[\tfrac{1}{T}\int_0^T \langle u(X_t,t), \sigma(t)^\top \nabla V(X_t,t)\rangle \,\mathrm{d}t \,\exp\big(-\lambda^{-1}\mathcal{W}(X,0)\big)\big] \\
&\overset{\text{(i)}}{=} -\lambda\mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle u(X_t,t), \tfrac{\sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big]}{\mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big]}\big\rangle \\
&\qquad\qquad \times \mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big] \exp\big(-\lambda^{-1}\int_0^t f(X_s,s)\,\mathrm{d}s\big)\,\mathrm{d}t\big] \\
&= -\lambda\mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle u(X_t,t), \sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|\mathcal{F}_t\big]\big\rangle \exp\big(-\lambda^{-1}\int_0^t f(X_s,s)\,\mathrm{d}s\big)\,\mathrm{d}t\big] \\
&\overset{\text{(ii)}}{=} -\lambda\mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle u(X_t,t), \sigma(t)^\top \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\mathcal{W}(X,t)\big)\big|X_t = x\big]\big\rangle \exp\big(-\lambda^{-1}\int_0^t f(X_s,s)\,\mathrm{d}s\big)\,\mathrm{d}t\big].
\end{aligned}$$

Here, (i) holds by equation (47), the law of total expectation and equation (46), and (ii) holds by the Markov property of the solution of an SDE. □

The following proposition, which we prove in Subsec. C.2, provides an alternative expression for $\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big]$. The technique, which is novel and we denote by *Girsanov reparamaterization trick*, is of independent interest and may be applied in other settings, as we discuss in Sec. 5.

Proposition 1 (Path-wise reparameterization trick for stochastic optimal control). *For each $t \in [0,T]$, let $M_t : [t,T] \to \mathbb{R}^{d\times d}$ be an arbitrary continuously differentiable function matrix-valued function such that $M_t(t) = \mathrm{Id}$. We have that*

$$\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big] \\
&= \mathbb{E}\big[\big(-\lambda^{-1}\int_t^T M_t(s)\nabla_x f(X_s,s)\,\mathrm{d}s - \lambda^{-1}M_t(T)\nabla g(X_T) \\
&\qquad + \lambda^{-1/2}\int_t^T (M_t(s)\nabla_x b(X_s,s) - \partial_s M_t(s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big) \\
&\qquad \times \exp\big(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big].
\end{aligned} \tag{23}$$

Plugging (23) into the right-hand side of (45), we obtain that

$$\begin{aligned}
&\mathbb{E}\big[\tfrac{1}{T}\int_0^T \langle u(X_t,t), \sigma(t)^\top \nabla V(X_t,t)\rangle \,\mathrm{d}t \,\exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big] \\
&= \mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle u(X_t,t), \sigma(t)^\top\big(\int_t^T M_t(s)\nabla_x f(X_s,s)\,\mathrm{d}s + M_t(T)\nabla g(X_T) \\
&\qquad - \lambda^{1/2}\int_t^T (M_t(s)\nabla_x b(X_s,s) - \partial_s M_t(s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big)\big\rangle \,\mathrm{d}t \\
&\qquad \times \exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big].
\end{aligned}$$

If we plug this into the right-hand side of (44) and complete the squared norm, we get that

$$\begin{aligned}
\tilde{\mathcal{L}}(u) = \mathbb{E}\big[\tfrac{1}{T}\int_0^T \big(\big\|u(X_t,t) - \tilde{w}(t,X,B,M_t)\big\|^2 - \big\|\tilde{w}(t,X,B,M_t)\big\|^2 \\
+ \big\|u^*(X_t,t)\big\|^2\big)\,\mathrm{d}t \exp\big(-\lambda^{-1}\mathcal{W}(X,0)\big)\big]
\end{aligned}$$

where $\tilde{w}$ is defined as:

$$\begin{aligned}
\tilde{w}(t,X,B,M_t) = \sigma(t)^\top\big(-\int_t^T M_t(s)\nabla_x f(X_s,s)\,\mathrm{d}s - M_t(T)\nabla g(X_T) \\
+ \lambda^{1/2}\int_t^T (M_t(s)\nabla_x b(X_s,s) - \partial_s M_t(s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big).
\end{aligned}$$

We also define $\Phi(u;X,B)$ as

$$\Phi(u;X,B) = \tfrac{1}{T}\int_0^T \big(\big\|u(X_t,t) - \tilde{w}(t,X,B,M_t)\big\|^2\big)\,\mathrm{d}t.$$

Now, by the Girsanov theorem (Thm. 3), we have that for an arbitrary control $v \in \mathcal{U}$,

$$\begin{aligned}
&\mathbb{E}[\Phi(u; X, B) \exp\big(- \lambda^{-1} \mathcal{W}(X, 0)\big)] \\
&= \mathbb{E}\big[\Phi(u; X^v, B^v) \exp\big(- \lambda^{-1} \mathcal{W}(X^v, 0) - \lambda^{-1/2} \textstyle\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t^v \rangle + \frac{\lambda^{-1}}{2} \int_0^T \|v(X_t^v, t)\|^2 \, \mathrm{d}t\big)\big] \\
&= \mathbb{E}\big[\Phi(u; X^v, B^v) \exp\big(- \lambda^{-1} \mathcal{W}(X^v, 0) - \lambda^{-1/2} \textstyle\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t \rangle - \frac{\lambda^{-1}}{2} \int_0^T \|v(X_t^v, t)\|^2 \, \mathrm{d}t\big)\big],
\end{aligned}$$

where $B_t^v := B_t + \lambda^{-1/2} \int_0^t v(X_s^v, s) \, \mathrm{d}s$. Reexpressing B^v in terms of B, we can rewrite $\Phi(u; X^v, B^v)$ and $\tilde{w}(t, X^v, B^v, M_t)$ as follows:

$$\begin{aligned}
\Phi(u; X^v, B^v) &= \tfrac{1}{T} \textstyle\int_0^T \big\|u(X_t^v, t) - \tilde{w}(t, X^v, B^v, M_t)\big\|^2 \, \mathrm{d}t, \\
\tilde{w}(t, X^v, B^v, M_t) &= \sigma(t)^\top \big(- \textstyle\int_t^T M_t(s) \nabla_x f(X_s^v, s) \, \mathrm{d}s - M_t(T) \nabla g(X_T^v) \\
&\qquad + \lambda^{1/2} \textstyle\int_t^T (M_t(s) \nabla_x b(X_s^v, s) - \partial_s M_t(s))(\sigma^{-1})^\top (X_s^v, s) \mathrm{d}B_s \\
&\qquad + \textstyle\int_t^T (M_t(s) \nabla_x b(X_s^v, s) - \partial_s M_t(s))(\sigma^{-1})^\top (X_s^v, s) v(X_s^v, s) \mathrm{d}s\big).
\end{aligned}$$

Putting everything together, we obtain that

$$\tilde{\mathcal{L}}(u) = \mathcal{L}_{\mathrm{SOCM}}(u, M) - K,$$

where $\mathcal{L}(u, M)$ is the loss defined in (19) (note that $w(t, v, X^v, B, M_t) := \tilde{w}(t, X^v, B^v, M_t)$), and

$$K = \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T (\big\|\tilde{w}(t, X, B, M_t)\big\|^2 - \big\|u^*(X_t, t)\big\|^2) \, \mathrm{d}t \, \exp\big(- \lambda^{-1} \mathcal{W}(X, 0)\big)\big]$$

To complete the proof of equation (24), remark that $\tilde{\mathcal{L}}(u)$ can be rewritten as

$$\begin{aligned}
\tilde{\mathcal{L}}(u) &= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|u(X_t, t) - u^*(X_t, t)\big\|^2 \, \mathrm{d}t \, \exp\big(- \lambda^{-1} \mathcal{W}(X, 0)\big)\big] \\
&= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|u(X_t, t) - u^*(X_t, t)\big\|^2 \, \mathrm{d}t \, \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) \exp(-\lambda^{-1} V(X_0, 0))\big] \\
&= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|u(X_t^{u^*}, t) - u^*(X_t^{u^*}, t)\big\|^2 \, \mathrm{d}t \, \exp(-\lambda^{-1} V(X_0^{u^*}, 0))\big].
\end{aligned}$$

It only remains to reexpress K. Note that by Prop. 1, we have that

$$\begin{aligned}
u^*(X_t, t) &= \frac{\mathbb{E}\Big[\tilde{w}(t, X, B, M_t) \exp\Big(-\lambda^{-1} \mathcal{W}(X, 0)\Big) | \mathcal{F}_t\Big]}{\mathbb{E}\Big[\exp\Big(-\lambda^{-1} \mathcal{W}(X, 0)\Big) | \mathcal{F}_t\Big]} \\
&= \frac{\mathbb{E}\Big[\tilde{w}(t, X, B, M_t) \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) | \mathcal{F}_t\Big] \exp(-\lambda^{-1} V(X_0, 0))}{\mathbb{E}\Big[\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) | \mathcal{F}_t\Big] \exp(-\lambda^{-1} V(X_0, 0))} = \frac{\mathbb{E}\Big[\tilde{w}(t, X, B, M_t) \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) | \mathcal{F}_t\Big]}{\mathbb{E}\Big[\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) | \mathcal{F}_t\Big]} \\
&= \mathbb{E}\big[\tilde{w}(t, X^{u^*}, B^{u^*}, M_t) | X_t^{u^*} = X_t\big]
\end{aligned}$$

Hence, using the Girsanov theorem (Thm. 3) several times, we have that

$$K = \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \|\tilde{w}(t, X^{u^*}, B^{u^*}, M_t)\|^2 - \|\mathbb{E}\big[\tilde{w}(t, X^{u^*}, B^{u^*}, M_t) | X_t^{u^*}\big]\big\|^2 \, \mathrm{d}t \, \exp(-\lambda^{-1} V(X_0^{u^*}, 0))\big] \tag{48}$$

$$\begin{aligned}
&= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|\tilde{w}(t, X^{u^*}, B^{u^*}, M_t) - \mathbb{E}\big[\tilde{w}(t, X^{u^*}, B^{u^*}, M_t) | X_t^{u^*}\big]\big\|^2 \, \mathrm{d}t \, \exp(-\lambda^{-1} V(X_0^{u^*}, 0))\big] \\
&= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|\tilde{w}(t, X, B, M_t) - \frac{\mathbb{E}[\tilde{w}(t, X, B, M_t) \exp(-\lambda^{-1} \mathcal{W}(X, 0)) | X_t]}{\mathbb{E}[\exp(-\lambda^{-1} \mathcal{W}(X, 0)) | X_t]}\big\|^2 \, \mathrm{d}t \, \exp(-\lambda^{-1} \mathcal{W}(X, 0))\big] \\
&= \mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\|w(t, v, X^v, B, M_t) - \frac{\mathbb{E}[w(t, v, X^v, B, M_t) \alpha(v, X^v, B) | X_t^v]}{\mathbb{E}[\alpha(v, X^v, B) | X_t^v]}\big\|^2 \, \mathrm{d}t \, \alpha(v, X^v, B)\big],
\end{aligned}$$

which concludes the proof, noticing that $K = \mathrm{Var}(w; M)$.

Remark 1 (The optimal M_t is the same for all v)**.** *Looking at equation* (48)*, we observe that* $\mathrm{Var}(w; M)$ *does not depend on the base control v. Since minimizing $\mathcal{L}_{\mathrm{SOCM}}(u, M)$ with respect to M is equivalent to minimizing* $\mathrm{Var}(w; M)$*, we deduce that the optimal M does not depend on the vector field v.*

C.2 Proof of the path-wise reparameterization trick (Prop. 1)

We prove a more general statement (Prop. 4), and show that Prop. 1 is a particular case of it.

Proposition 4 (Path-wise reparameterization trick)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and $B : \Omega \times [0,T] \to \mathbb{R}^d$ be a Brownian motion. Let $X : \Omega \times [0,T] \to \mathbb{R}^d$ be the uncontrolled process given by* (6)*, and let $\psi : \Omega \times \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ be an arbitrary random process such that:*

- *For all $z \in \mathbb{R}^d$, the process $\psi(\cdot, z, \cdot) : \Omega \times [0,T] \to \mathbb{R}^d$ is adapted to the filtration $(\mathcal{F}_s)_{s\in[0,T]}$ of the Brownian motion B.*
- *For all $\omega \in \Omega$, $\psi(\omega, \cdot, \cdot) : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is a twice-continuously differentiable function such that $\psi(\omega, z, 0) = z$ for all $z \in \mathbb{R}^d$, and $\psi(\omega, 0, s) = 0$ for all $s \in [0,T]$.*

Let $F : C([0,T]; \mathbb{R}^d) \to \mathbb{R}$ be a Fréchet-differentiable functional. We use the notation $X + \psi(z, \cdot) = (X_s(\omega) + \psi(\omega, z, s))_{s\in[0,T]}$ to denote the shifted process, and we will omit the dependency of ψ on ω in the proof. Then,

$$\begin{aligned}&\nabla_x \mathbb{E}\big[\exp\big(-F(X)\big)\big|X_0 = x\big] \qquad (49)\\ &= \mathbb{E}\big[\big(-\nabla_z F(X+\psi(z,\cdot))|_{z=0} + \lambda^{-1/2} \textstyle\int_0^T (\nabla_z \psi(0,s) \nabla_x b(X_s, s) - \nabla_z \partial_s \psi(0,s))(\sigma^{-1})^\top(s) \mathrm{d}B_s\big) \\ &\qquad \times \exp\big(-F(X)\big)\big|X_0 = x\big]\end{aligned}$$

Proof of Prop. 1. Given a family of functions $(M_t)_{t\in[0,T]}$ satisfying the conditions in Prop. 1, we can define a family $(\psi_t)_{t\in[0,T]}$ of functions $\psi_t : \mathbb{R}^d \times [t,T] \to \mathbb{R}^d$ as $\psi_t(z,s) = M_t(s)^\top z$. Note that $\psi_t(z,t) = z$ for all $z \in \mathbb{R}^d$ and $\psi_t(0,s) = 0$ for all $s \in [t,T]$, and that $\nabla_z \psi_t(z,s) = M_t(s)$. Hence, ψ_t can be seen as a random process which is constant with respect to $\omega \in \Omega$, and which fulfills the conditions in Prop. 4 up to a trivial time change of variable from $[t,T]$ to $[0,T]$.

We also define the family $(F_t)_{t\in[0,T]}$ of functionals $F_t : C([t,T]; \mathbb{R}^d) \to \mathbb{R}$ as $F_t(X) = \lambda^{-1} \int_t^T f(X_s, s) \, \mathrm{d}s + \lambda^{-1} g(X_T)$. We have that

$$\begin{aligned}&\nabla_z F_t(X+\psi_t(z,\cdot)) \\ &= \nabla_z\big(\lambda^{-1} \textstyle\int_t^T f(X_s + \psi_t(z,s), s) \, \mathrm{d}s + \lambda^{-1} g(X_T + \psi_t(z,T))\big) \\ &\overset{\text{(i)}}{=} \lambda^{-1} \textstyle\int_t^T \nabla_z \psi_t(z,s) \nabla f(X_s+\psi_t(z,s), s) \, \mathrm{d}s + \lambda^{-1} \nabla_z \psi_t(z,T) \nabla g(X_T+\psi_t(z,T)) \\ &= \lambda^{-1} \textstyle\int_t^T M_t(s) \nabla f(X_s+\psi_t(z,s), s) \, \mathrm{d}s + \lambda^{-1} M_t(T) \nabla g(X_T+\psi_t(z,T)),\end{aligned}$$

where equality (i) holds by the Leibniz rule. Using that $\psi_t(0,s) = 0$, we obtain that:

$$\nabla_z F_t(X + \psi_t(z,\cdot))\big|_{z=0} = \lambda^{-1} \textstyle\int_t^T \nabla_z \psi_t(0,s) \nabla f(X_s, s) \, \mathrm{d}s + \lambda^{-1} \nabla_z \psi_t(T,0) \nabla g(X_T),$$

Up to a trivial time change of variable from $[t,T]$ to $[0,T]$, Prop. 1 follows from plugging these choices into equation (49).

Remark 2. *We can use matrices $M_t(s)$ that depend on the process X up to time s, since the resulting processes $\psi_t(\cdot, z, \cdot)$ are adapted to the filtration of the Brownian motion B. More specifically, if we let $M_t : \mathbb{R}^d \times [t,T] \to \mathbb{R}^{d\times d}$ be an arbitrary continuously differentiable function matrix-valued function such that $M_t(x,t) = \mathrm{Id}$ for all $x \in \mathbb{R}^d$, and we define the exponential moving average of X as the process $X^{(\upsilon)}$ given by*

$$X_t^{(\upsilon)} = \upsilon \textstyle\int_0^t e^{-\upsilon(t-s)} X_s \, \mathrm{d}s,$$

we have that

$$\tfrac{\mathrm{d}}{\mathrm{d}s} M_t(X_s^{(\upsilon)}, s) = \langle \nabla M_t(X_s^{(\upsilon)}, s), \tfrac{\mathrm{d}X_s^{(\upsilon)}}{\mathrm{d}s} \rangle + \partial_s M_t(X_s^{(\upsilon)}, s) = \lambda \langle \nabla_x M_t(X_s^{(\upsilon)}, s), X_s - X_s^{(\upsilon)} \rangle + \partial_s M_t(X_s^{(\upsilon)}, s),$$

and we can write

$$\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big] \\
&= \mathbb{E}\big[\big(-\lambda^{-1}\textstyle\int_t^T M_t(X_s^{(v)},s)\nabla_x f(X_s,s)\,\mathrm{d}s - \lambda^{-1} M_t(X_T^{(v)},T)\nabla g(X_T) \\
&\qquad + \lambda^{-1/2}\textstyle\int_t^T (M_t(X_s^{(v)},s)\nabla_x b(X_s,s) - \frac{\mathrm{d}}{\mathrm{d}s}M_t(X_s^{(v)},s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big) \\
&\qquad \times \exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big].
\end{aligned}$$

Plugging this into the proof of Thm. 1, we would obtain a variant of SOCM (Alg. 2) where the matrix-valued neural network M_ω takes inputs (t,s,x) instead of (t,s). Since the optimization class is larger, from the bias-variance in Prop. 2 we deduce that this variant would yield a lower variance of the vector field w, and likely an algorithm with lower error. This is at the expense of an increased number of function evaluations (NFE) of M_ω; one would need $\frac{K(K+1)m}{2}$ NFE per batch instead of only $\frac{K(K+1)}{2}$, which may be too expensive if the architecture of M_ω is large. A way to speed up the computation per batch is to parameterize M_t using cubic splines.

□

Proof of Prop. 4. Recall that

$$\mathrm{d}X_s = b(X_s,s)\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s, \qquad X_0 \sim p_0,$$

is the SDE for the uncontrolled process. For arbitrary $x, z \in \mathbb{R}^d$, we consider the following SDEs conditioned on the initial points:

$$\mathrm{d}X_s^{(x+z)} = b(X_s^{(x+z)},s)\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s, \qquad X_0^{(x+z)} = x+z, \tag{50}$$

$$\mathrm{d}X_s^{(x)} = b(X_s^{(x)},s)\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s, \qquad X_0^{(x)} = x. \tag{51}$$

Suppose that $\psi : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ satisfies the properties in the statement of Prop. 4. If $\tilde{X}^{(x)}$ is a solution of

$$\mathrm{d}\tilde{X}_s^{(x)} = (b(\tilde{X}_s^{(x)} + \psi(z,s),s) - \partial_s\psi(z,s))\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s, \qquad \tilde{X}_0^{(x)} = x,$$

then $X^{(x+z)} = \tilde{X}^{(x)} + \psi(z,\cdot)$ is a solution of (50). This is because $X_0^{(x+z)} = \tilde{X}_0^{(x)} + \psi(z,0) = \tilde{X}_0^{(x)} + z = x + z$, and

$$\begin{aligned}
\mathrm{d}X_s^{(x+z)} &= \mathrm{d}\tilde{X}_s^{(x)} + \partial_s\psi(z,s)\,\mathrm{d}s \\
&= (b(\tilde{X}_s^{(x)} + \psi(z,s),s) - \partial_s\psi(z,s))\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s + \partial_s\psi(z,s)\,\mathrm{d}s \\
&= b(X_s^{(x+z)},s)\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s,
\end{aligned}$$

Note that we may rewrite (51) as

$$\begin{aligned}
\mathrm{d}X_s^{(x)} &= (b(X_s^{(x)} + \psi(z,s),s) - \partial_s\psi(z,s))\,\mathrm{d}s \\
&\qquad + (b(X_s^{(x)},s) - b(X_s^{(x)} + \psi(z,s),s) + \partial_s\psi(z,s))\,\mathrm{d}s + \sqrt{\lambda}\sigma(s)\,\mathrm{d}B_s, \qquad X_t^{(x)} \sim p_0.
\end{aligned}$$

Hence, since $\psi(z,s)$ is a random process adapted to the filtration of B, we can apply the Girsanov theorem for SDEs (Corollary 1) on $\tilde{X}^{(x)}$ and $X^{(x)}$, and we have that for any bounded continuous functional Φ,

$$\begin{aligned}
&\mathbb{E}[\Phi(\tilde{X}^{(x)})] \\
&= \mathbb{E}\big[\Phi(X^{(x)})\exp\big(\textstyle\int_0^T \lambda^{-1/2}\sigma(s)^{-1}(b(X_s^{(x)} + \psi(z,s),s) - b(X_s^{(x)},s) - \partial_s\psi(z,s))\,\mathrm{d}B_s \\
&\qquad - \frac{1}{2}\textstyle\int_0^T \|\lambda^{-1/2}\sigma(s)^{-1}(b(X_s^{(x)} + \psi(z,s),s) - b(X_s^{(x)},s) - \partial_s\psi(z,s))\|^2\,\mathrm{d}s\big)\big].
\end{aligned} \tag{52}$$

We can write

$$\begin{aligned}
&\mathbb{E}\big[\exp\big(-F(X)\big)\big|X_0 = x+z\big] \overset{\text{(i)}}{=} \mathbb{E}\big[\exp\big(-F(X^{(x+z)})\big)\big] \overset{\text{(ii)}}{=} \mathbb{E}\big[\exp\big(-F(\tilde{X}^{(x)}+\psi(z,\cdot))\big)\big] \\
&\overset{\text{(iii)}}{=} \mathbb{E}\big[\exp\big(-F(X^{(x)}+\psi(z,\cdot))\big) \\
&\qquad \times \exp\big(\textstyle\int_0^T \lambda^{-1/2}\sigma(s)^{-1}(b(X_s^{(x)}+\psi(z,s),s) - b(X_s^{(x)},s) - \partial_s\psi(z,s))\,\mathrm{d}B_s \\
&\qquad\qquad - \tfrac{1}{2}\textstyle\int_0^T \|\lambda^{-1/2}\sigma(s)^{-1}(b(X_s^{(x)}+\psi(z,s),s) - b(X_s^{(x)},s) - \partial_s\psi(z,s))\|^2\,\mathrm{d}s\big)\big] \\
&\overset{\text{(iv)}}{=} \mathbb{E}\big[\exp\big(-F(X+\psi(z,\cdot)) + \textstyle\int_0^T \lambda^{-1/2}\sigma(s)^{-1}(b(X_s+\psi(z,s),s) - b(X_s,s) - \partial_s\psi(z,s))\,\mathrm{d}B_s \\
&\qquad\qquad - \tfrac{1}{2}\textstyle\int_0^T \|\lambda^{-1/2}\sigma(s)^{-1}(b(X_s+\psi(z,s),s) - b(X_s,s) - \partial_s\psi(z,s))\|^2\,\mathrm{d}s\big)\big|X_0 = x\big]
\end{aligned} \tag{53}$$

Equality (i) holds by the definition of $X^{(x+z)}$, equality (ii) holds by the fact $X_s^{(x+z)} = \tilde{X}_s^{(x)} + \psi(z,s)$, equality (iii) holds by equation (52), and equality (iv) holds by the definition of $X_s^{(x)}$. We conclude the proof by differentiating the right-hand side of (53) with respect to z. Namely,

$$\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-F(X)\big)\big|X_0 = x\big] = \nabla_z \mathbb{E}\big[\exp\big(-F(X)\big)\big|X_0 = x+z\big]\big|_{z=0} \\
&\overset{\text{(i)}}{=} \mathbb{E}\big[\big(-\nabla_z F(X+\psi(z,\cdot)) + \lambda^{-1/2}\textstyle\int_0^T (\nabla_z\psi(0,s)\nabla_x b(X_s,s) - \nabla_z\partial_s\psi(0,s))(\sigma^{-1})^\top(s)\mathrm{d}B_s\big) \\
&\qquad\quad \times \exp\big(-F(X)\big)\big|X_0 = x\big]
\end{aligned}$$

In equality (i) we used (53), and that:

- by the Leibniz rule,
$$\begin{aligned}
&\nabla_z \textstyle\int_0^T \|\sigma(s)^{-1}(b(X_s+\psi(z,s),s) - b(X_s,s) - \partial_s\psi(z,s))\|^2\,\mathrm{d}s\big|_{z=0} \\
&= \textstyle\int_0^T \nabla_z\|\sigma(s)^{-1}(b(X_s+\psi(z,s),s) - b(X_s,s) - \partial_s\psi(z,s))\|^2\big|_{z=0}\,\mathrm{d}s = 0.
\end{aligned}$$
- and by the Leibniz rule for stochastic integrals (see [42]),
$$\begin{aligned}
&\nabla_z\big(\textstyle\int_0^T \sigma(s)^{-1}(b(X_s+\psi(z,s),s) - b(X_s,s) - \partial_s\psi(z,s))\,\mathrm{d}B_s\big)\big|_{z=0} \\
&= \textstyle\int_0^T (\nabla_z\psi(0,s)\nabla_x b(X_s,s) - \nabla_z\partial_s\psi(0,s))(\sigma^{-1})^\top(s)\,\mathrm{d}B_s.
\end{aligned}$$

□

C.3 Informal derivation of the path-wise reparameterization trick

In this subsection, we provide an informal, intuitive derivation of the path-wise reparameterization trick as stated in Prop. 4. For simplicity, we particularize the functional F to $F(X) = \lambda^{-1}\int_0^T f(X_s,s)\,\mathrm{d}s + \lambda^{-1}g(X_T)$. Consider the Euler-Maruyama discretization of the uncontrolled process X defined in (6), with $K+1$ time steps (let $\delta = T/K$ be the step size). This is a family of random variables $\hat{X} = (\hat{X}_k)_{k=0:K}$ defined as

$$\hat{X}_0 \sim p_0, \qquad \hat{X}_{k+1} = \hat{X}_k + \delta b(\hat{X}_k, k\delta) + \sqrt{\delta\lambda}\sigma(k\delta)\varepsilon_k, \qquad \varepsilon_k \sim N(0,I).$$

Note that we can approximate

$$\begin{aligned}
&\mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_0^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_0 = x\big] \\
&\approx \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta\textstyle\sum_{k=0}^{K-1} f(\hat{X}_k,s) - \lambda^{-1}g(\hat{X}_K)\big)\big|\hat{X}_0 = x\big],
\end{aligned}$$

and that this is an equality in the limit $K \to \infty$, as the interpolation of the Euler-Maruyama discretization $\hat{X}^{(x)}$ converges to the process $X^{(x)}$. Now, remark that for $k \in \{0,\dots,K-1\}$, $\hat{X}_{k+1}|\hat{X}_k \sim N(\hat{X}_k + \delta b(\hat{X}_k,k\delta), \delta\lambda(\sigma\sigma^\top)(k\delta))$. Hence,

$$\begin{aligned}
&\mathbb{E}\big[\exp\big(-\lambda^{-1}\delta\textstyle\sum_{k=0}^{K-1} f(\hat{X}_k,s) - \lambda^{-1}g(\hat{X}_K)\big)\big|\hat{X}_0 = x\big] \\
&= C^{-1}\textstyle\iint_{(\mathbb{R}^d)^K} \exp\big(-\lambda^{-1}\delta\sum_{k=0}^{K-1} f(\hat{x}_k,s) - \lambda^{-1}g(\hat{x}_K) \\
&\qquad\qquad\qquad - \tfrac{1}{2\delta\lambda}\textstyle\sum_{k=1}^{K-1}\|\sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k,k\delta))\|^2 \\
&\qquad\qquad\qquad - \tfrac{1}{2\delta\lambda}\|\sigma^{-1}(0)(\hat{x}_1 - x - \delta b(x,0))\|^2\big)\,\mathrm{d}\hat{x}_1\cdots\mathrm{d}\hat{x}_K,
\end{aligned}$$

where $C = \sqrt{(2\pi\delta\lambda)^K \prod_{k=0}^{K-1} \det((\sigma\sigma^\top)(k\delta))}$. Now, let $\psi : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ be an arbitrary twice differentiable function such that $\psi(z,0) = z$ for all $z \in \mathbb{R}^d$, and $\psi(0,s) = 0$ for all $s \in [0,T]$. We can write

$$\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big) | \hat{X}_0 = x\big] \\
&= \nabla_z \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big) | \hat{X}_0 = x + z\big]|_{z=0} \\
&= C^{-1} \nabla_z \big(\textstyle\iint_{(\mathbb{R}^d)^K} \exp\big(-\lambda^{-1}\delta \sum_{k=0}^{K-1} f(\hat{x}_k, s) - \lambda^{-1} g(\hat{x}_K) \\
&\qquad\qquad - \tfrac{1}{2\delta\lambda} \textstyle\sum_{k=1}^{K-1} \|\sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k, k\delta))\|^2 \\
&\qquad\qquad - \tfrac{1}{2\delta\lambda} \|\sigma^{-1}(0)(\hat{x}_1 - (x+z) - \delta b(x+z, 0))\|^2\big) \,\mathrm{d}\hat{x}_1 \cdots \mathrm{d}\hat{x}_K\big)|_{z=0} \\
&= C^{-1} \nabla_z \big(\textstyle\iint_{(\mathbb{R}^d)^K} \exp\big(-\lambda^{-1}\delta \sum_{k=0}^{K-1} f(\hat{x}_k + \psi(z, k\delta), s) - \lambda^{-1} g(\hat{x}_K + \psi(z, K\delta)) \\
&\qquad - \tfrac{1}{2\delta\lambda} \textstyle\sum_{k=1}^{K-1} \|\sigma^{-1}(k\delta)(\hat{x}_{k+1} + \psi(z, (k+1)\delta) - \hat{x}_k - \psi(z, k\delta) - \delta b(\hat{x}_k + \psi(z, k\delta), k\delta))\|^2 \\
&\qquad - \tfrac{1}{2\delta\lambda} \|\sigma^{-1}(0)(\hat{x}_1 + \psi(z, \delta) - (x + \psi(z,0)) - \delta b(x + \psi(z,0), 0))\|^2\big) \,\mathrm{d}\hat{x}_1 \cdots \mathrm{d}\hat{x}_K\big)|_{z=0},
\end{aligned} \tag{54}$$

In the last equality, we used that for $k \in \{1, \dots, K\}$, the variables $\hat{x}_k$ are integrated over $\mathbb{R}^d$, which means that adding an offset $\psi(z, k\delta)$ does not change the value of the integral. We also used that $\psi(z,0) = z$. Now, for fixed values of $\hat{x} = (\hat{x}_1, \dots, \hat{x}_K)$, and letting $\hat{x}_0 = x$, we define

$$\begin{aligned}
G_{\hat{x}}(z) &= \lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{x}_k + \psi(z, k\delta), s) + \lambda^{-1} g(\hat{x}_K + \psi(z, K\delta)) \\
&\quad + \tfrac{1}{2\delta\lambda} \textstyle\sum_{k=0}^{K-1} \|\sigma^{-1}(k\delta)(\hat{x}_{k+1} + \psi(z, (k+1)\delta) - \hat{x}_k - \psi(z, k\delta) - \delta b(\hat{x}_k + \psi(z, k\delta), k\delta))\|^2.
\end{aligned}$$

Using that $\psi(0,s) = 0$ for all $s \in [0,T]$, we have that:

$$G_{\hat{x}}(0) = \lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{x}_k, s) + \lambda^{-1} g(\hat{x}_K) + \tfrac{1}{2\delta\lambda} \textstyle\sum_{k=0}^{K-1} \|\sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k, k\delta))\|^2.$$

$$\begin{aligned}
\nabla G_{\hat{x}}(z)|_{z=0} &= \lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} \nabla\psi(0, k\delta) \nabla f(\hat{x}_k, s) + \lambda^{-1} \nabla\psi(0, K\delta) \nabla g(\hat{x}_K) \\
&\quad + \tfrac{1}{\delta\lambda} \textstyle\sum_{k=0}^{K-1} (\nabla_z \psi(0, (k+1)\delta) - \nabla_z \psi(0, k\delta) - \delta \nabla\psi(0, k\delta) \nabla b(\hat{x}_k, k\delta)) \\
&\qquad \times ((\sigma^{-1})^\top \sigma^{-1})(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k, k\delta))
\end{aligned}$$

And we can express the right-hand side of (54) in terms of $G_{\hat{x}}(0)$ and $\nabla G_{\hat{x}}(z)|_{z=0}$:

$$\begin{aligned}
&\nabla_z \big(C^{-1} \textstyle\iint_{(\mathbb{R}^d)^K} \exp\big(-G_{\hat{x}}(z)\big) \,\mathrm{d}y_1 \cdots \mathrm{d}y_K\big) \\
&= -C^{-1} \textstyle\iint_{(\mathbb{R}^d)^K} \nabla G_{\hat{x}}(z)|_{z=0} \exp\big(-G_{\hat{x}}(0)\big) \,\mathrm{d}y_1 \cdots \mathrm{d}y_K
\end{aligned}$$

We define $\epsilon_k = \frac{1}{\sqrt{\delta\lambda}} \sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k, k\delta))$, and then, we are able to write

$$\hat{x}_{k+1} = \hat{x}_k + \delta b(\hat{x}_k, k\delta) + \sqrt{\delta\lambda}\sigma(k\delta)\epsilon_k, \qquad \hat{x}_0 = x \tag{55}$$

$$\begin{aligned}
G_{\hat{x}}(0) &= \lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{x}_k, s) + \lambda^{-1} g(\hat{x}_K) + \tfrac{1}{2} \textstyle\sum_{k=0}^{K-1} \|\epsilon_k\|^2, \\
\nabla G_{\hat{x}}(z)|_{z=0} &= \lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} \nabla\psi(0, k\delta) \nabla f(\hat{x}_k, s) + \lambda^{-1} \nabla\psi(0, K\delta) \nabla g(\hat{x}_K) \\
&\quad + \sqrt{\delta\lambda^{-1}} \textstyle\sum_{k=0}^{K-1} (\partial_s \nabla_z \psi(0, k\delta) + O(\delta) - \nabla\psi(0, k\delta) \nabla b(\hat{x}_k, k\delta))(\sigma^{-1})^\top(k\delta)\epsilon_k.
\end{aligned} \tag{56}$$

Then, taking the limit $K \to \infty$ (i.e. $\delta \to 0$), we recognize (55) as Euler-Maruyama discretization of the uncontrolled process X in equation (6) conditioned on $X_0 = x$, and the last term in (56) as the Euler-Maruyama discretization of the stochastic integral $\lambda^{-1/2} \int_0^T (\partial_s \nabla_z \psi(0,s) - \nabla\psi(0,s) \nabla b(X_s^{(x)}, s))(\sigma^{-1})^\top(s) \,\mathrm{d}B_s$. Thus,

$$\begin{aligned}
&\lim_{K\to\infty} \nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big)\big] \\
&= \mathbb{E}\big[\big(-\lambda^{-1} \textstyle\int_0^T \nabla\psi(0,s) \nabla_x f(X_s, s) \,\mathrm{d}s - \lambda^{-1} \nabla\psi(0,T) \nabla g(X_T) \\
&\qquad + \lambda^{-1/2} \textstyle\int_0^T (\nabla\psi(0,s) \nabla_x b(X_s, s) - \partial_s \nabla\psi(0,s))(\sigma^{-1})^\top(s) \,\mathrm{d}B_s\big) \\
&\qquad \times \exp\big(-\lambda^{-1} \textstyle\int_0^T f(X_s, s) \,\mathrm{d}s - \lambda^{-1} g(X_T)\big) \big| X_0 = x\big],
\end{aligned}$$

which concludes the derivation.

C.4 SOCM-Adjoint: replacing the path-wise reparameterization trick with the adjoint method

Proposition 5. *Let $\mathcal{L}_{\mathrm{SOCM-Adj}} : L^2(\mathbb{R}^d \times [0,T]; \mathbb{R}^d) \to \mathbb{R}$ be the loss function defined as*

$$\mathcal{L}_{\mathrm{SOCM-Adj}}(u) := \mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\|u(X_t^v,t) + \sigma(t)^\top a(t,X^v)\big\|^2 \, \mathrm{d}t \times \alpha(v,X^v,B)\big] \, ,$$

where X^v is the process controlled by v (i.e., $dX_t = (b(X_t,t) + \sigma(t)v(X_t,t))\,\mathrm{d}t + \sqrt{\lambda}\sigma(X_t,t)\,\mathrm{d}B_t$ and $X_0 \sim p_0$), $\alpha(v,X^v,B)$ is the importance weight defined in (20)*, and $a(t,X^v)$ is the solution of the ODE*

$$\begin{aligned} \tfrac{da(t)}{dt} &= -\nabla_x b(X_t^v,t)a(t) - \nabla_x f(X_t^v,t), \\ a(T) &= \nabla g(X_T^v), \end{aligned}$$

$\mathcal{L}_{\mathrm{SOCM-Adj}}$ has a unique optimum, which is the optimal control u^*.

Proof. The proof follows the same structure as that of Thm. 1. Instead of plugging the path-wise reparameterization trick (Prop. 1) in the right-hand side of (22), we make use of Lemma 6 to evaluate $\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,dt - \lambda^{-1}g(X_T)\big)|X_0 = x\big]$. Particular cases of the result in Lemma 6 have been used in previous works such as [54, 51]. We present a more general form that covers state costs f, as well as stochastic integrals. We also present a simpler proof of the result based on Lagrange multipliers. □

Lemma 6 (Adjoint method for SDEs)**.** *[54, 51] Let $X : \Omega \times [0,T] \to \mathbb{R}^d$ be the uncontrolled process defined in* (6)*, with initial condition $X_0 = x$. We define the random process $a : \Omega \times [0,T] \to \mathbb{R}^d$ such that for all $\omega \in \Omega$, using the short-hand $a(t) := a(\omega,t)$,*

$$\begin{aligned} \mathrm{d}a_t(\omega) &= \big(-\nabla_x b(X_t(\omega),t)a_t(\omega) - \nabla_x f(X_t(\omega),t)\big)\,\mathrm{d}t - \nabla_x h(X_t(\omega),t)\,\mathrm{d}B_t, \\ a_T(\omega) &= \nabla_x g(X_T(\omega)), \end{aligned}$$

we have that

$$\begin{aligned} &\nabla_x \mathbb{E}\big[\int_0^T f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle + g(X_T(\omega))|X_0(\omega) = x\big] = \mathbb{E}\big[a_0(\omega)\big], \\ &\nabla_x \mathbb{E}\big[\exp\big(-\int_0^T f(X_t(\omega),t)\,\mathrm{d}t - \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle - g(X_T(\omega))\big)|X_0(\omega) = x\big] \\ &= -\mathbb{E}\big[a_0(\omega)\exp\big(-\int_0^T f(X_t(\omega),t)\,\mathrm{d}t - \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle - g(X_T(\omega))\big)|X_0(\omega) = x\big]. \end{aligned}$$

Proof. We will use an approach based on Lagrange multipliers. Define a process $a : \Omega \times [0,T] \to \mathbb{R}^d$ such that for any $\omega \in \Omega$, $a(\omega,\cdot)$ is differentiable. For a given $\omega \in \Omega$, we can write

$$\begin{aligned} &\int_0^T f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle + g(X_T(\omega)) \\ &= \int_0^T f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle + g(X_T(\omega)) \\ &\quad - \int_0^T \langle a_t(\omega), (dX_t(\omega) - b(X_t(\omega),t)\,\mathrm{d}t - \sigma(t)\,\mathrm{d}B_t)\rangle. \end{aligned}$$

By Lemma 7, we have that

$$\int_0^T \langle a_t(\omega), dX_t(\omega)\rangle = \langle a_T(\omega), X_T(\omega)\rangle - \langle a_0(\omega), X_0(\omega)\rangle - \int_0^T \langle X_t(\omega), \tfrac{da_t}{dt}(\omega)\rangle\,dt.$$

Hence,

$$\begin{aligned} &\nabla_x\big(\int_0^T f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle + g(X_T(\omega))\big) \\ &= \nabla_x\big(\int_0^T f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \langle h(X_t(\omega),t),\, \mathrm{d}B_t\rangle + g(X_T(\omega)) \\ &\qquad - \langle a_T(\omega), X_T(\omega)\rangle + \langle a_0(\omega), X_0(\omega)\rangle + \int_0^T \big(\langle a_t(\omega), b(X_t(\omega),t)\rangle + \langle \tfrac{da_t}{dt}(\omega), X_t(\omega)\rangle\big)\,\mathrm{d}t \\ &\qquad + \int_0^T \langle a_t(\omega), \sigma(t)\,\mathrm{d}B_t\rangle\big) \\ &= \int_0^T \nabla_x X_t(\omega)\nabla_x f(X_t(\omega),t)\,\mathrm{d}t + \int_0^T \nabla_x X_t(\omega)\nabla_x h(X_t(\omega),t)\,\mathrm{d}B_t + \nabla_x X_T(\omega)\nabla_x g(X_T(\omega)) \\ &\qquad - \nabla_x X_T(\omega)a_T(\omega) + \nabla_x X_0(\omega)a_0(\omega) \\ &\qquad + \int_0^T \big(\nabla_x X_t(\omega)\nabla_x b(X_t(\omega),t)a_t(\omega) + \nabla_x X_t(\omega)\tfrac{da_t}{dt}(\omega)\big)\,\mathrm{d}t \\ &= \int_0^T \nabla_x X_t(\omega)\big(\nabla_x f(X_t(\omega),t) + \nabla_x b(X_t(\omega),t)a_t(\omega) + \tfrac{da_t}{dt}(\omega)\big)\,\mathrm{d}t \\ &\qquad + \nabla_x X_T(\omega)\big(\nabla_x g(X_T(\omega)) - a_T(\omega)\big) + a_0(\omega) + \int_0^T \nabla_x X_t(\omega)\nabla_x h(X_t(\omega),t)\,\mathrm{d}B_t. \end{aligned}$$

In the last line we used that $\nabla_x X_0(\omega) = \nabla_x x = \mathrm{I}$. If choose a such that

$$\begin{aligned} da_t(\omega) &= \big(- \nabla_x b(X_t(\omega), t) a_t(\omega) - \nabla_x f(X_t(\omega), t)\big) \, \mathrm{d}t - \nabla_x h(X_t(\omega), t) \, \mathrm{d}B_t, \\ a_T(\omega) &= \nabla_x g(X_T(\omega)), \end{aligned}$$

then we obtain that

$$\nabla_x \big(\int_0^T f(X_t(\omega), t) \, \mathrm{d}t + \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle + g(X_T(\omega)) \big) = a_0(\omega),$$

and by the Leibniz rule,

$$\begin{aligned} &\nabla_x \mathbb{E}\big[\int_0^T f(X_t(\omega), t) \, \mathrm{d}t + \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle + g(X_T(\omega)) \big] \\ &= \mathbb{E}\big[\nabla_x \big(\int_0^T f(X_t(\omega), t) \, \mathrm{d}t + \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle + g(X_T(\omega)) \big) \big] = \mathbb{E}\big[a_0(\omega)\big], \end{aligned}$$

and

$$\begin{aligned} &\nabla_x \mathbb{E}\big[\exp \big(- \int_0^T f(X_t(\omega), t) \, \mathrm{d}t - \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle - g(X_T(\omega)) \big) \big] \\ &= -\mathbb{E}\big[\nabla_x \big(\int_0^T f(X_t(\omega), t) \, \mathrm{d}t + \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle + g(X_T(\omega)) \big) \\ &\qquad \times \exp \big(- \int_0^T f(X_t(\omega), t) \, \mathrm{d}t - \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle - g(X_T(\omega)) \big) \big] \\ &= -\mathbb{E}\big[a_0(\omega) \exp \big(- \int_0^T f(X_t(\omega), t) \, \mathrm{d}t - \int_0^T \langle h(X_t(\omega), t), \, \mathrm{d}B_t \rangle - g(X_T(\omega)) \big) \big]. \end{aligned}$$

□

Lemma 7 (Stochastic integration by parts, [60])**.** *Let*

$$\begin{aligned} \mathrm{d}X_t &= a_t \, \mathrm{d}t + b_t \, \mathrm{d}W_t^1, \\ \mathrm{d}Y_t &= f_t \, \mathrm{d}t + g_t \, \mathrm{d}W_t^2. \end{aligned}$$

where a_t, b_t, f_t, g_t are continuous square integrable processes adapted to a filtration $(\mathcal{F}_t)_{t \in [0,T]}$, and W^1, W^2 are Brownian motions adapted to the same filtration. Then,

$$\begin{aligned} X_t Y_t - X_0 Y_0 &= \int_0^t X_s \, \mathrm{d}Y_s + \int_0^t Y_s \, \mathrm{d}X_s + \int_0^t \langle \mathrm{d}X_s, \mathrm{d}Y_s \rangle \\ &= \int_0^t X_s \, \mathrm{d}Y_s + \int_0^t Y_s \, \mathrm{d}X_s + \int_0^t \langle b_s \, \mathrm{d}W_s^1, g_s \, \mathrm{d}W_s^2 \rangle. \end{aligned}$$

Remark 3 (Related work to the path-wise reparameterization trick: sensitivity analysis)**.** *As shown above, the adjoint method for SDEs is an alternative to the path-wise reparameterization trick. Prior to [54], an array of works developed methods to compute derivatives of functionals of stochastic processes with respect to generic parameters α that appear either in the drift or diffusion coefficients [52]. This area is known as sensitivity analysis, and has been developed largely with financial applications in mind (more specifically, to compute the "Greeks"). In low dimensions, dynamic programming [3] or finite differences [26, 53] work well, but they scale poorly to high dimensions. In high dimensions, several approaches have been proposed (see the section 1 of [29] for a comprehensive although dated overview):*

- *The* path-wise method *(which we refer to as adjoint method) involves taking the gradient $\nabla_\alpha \mathbb{E}[f(X_t)]$ inside of the expectation as $\mathbb{E}[\nabla_\alpha f(X_t)]$ and was first described by [82].*
- *The* likelihood method *or* score method *[27, 69] consists in rewriting $\nabla_\alpha \mathbb{E}[f(X_T)]$ as $\mathbb{E}[f(X_T)H]$, where H is a random variable which is equal to $\nabla_\alpha \log p(\alpha, X_T)$, $p(\alpha, \cdot)$ being the density of the law of X_T with respect to the Lebesgue measure. [82] provide explicit weights H, under the restrictions that α appears only in the drift of the SDE (and not in the diffusion coefficient) and that the diffusion coefficient is elliptic, using the Girsanov theorem. [29] provide an expression for H in the case where H also appears in the diffusion coefficient, using Malliavin calculus.*

The estimator of the path-wise reparameterization trick is formally similar to the likelihood method estimator, but it is different in that α is the initial condition of the process, and does not appear either in the drift nor the diffusion coefficient.

C.5 Proof of Lemma 1

Proof. Since the equality (28) holds almost surely for the pair (X, B), it must also hold almost surely for (X^v, B^v), which satisfy the same SDE. That is

$$\mathcal{W}(X^v, 0) = V(X_0^v, 0) + \tfrac{1}{2} \int_0^T \|u^*(X_s^v, s)\|^2 \, \mathrm{d}s - \sqrt{\lambda} \int_0^T \langle u^*(X_s^v, s), \mathrm{d}B_s^v \rangle,$$

Thus, we obtain that

$$\begin{aligned} \alpha(v, X^v, B) &= \exp\big(- \lambda^{-1} \mathcal{W}(X^v, 0) - \lambda^{-1/2} \textstyle\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t^v \rangle + \frac{\lambda^{-1}}{2} \int_0^T \|v(X_t^v, t)\|^2 \, \mathrm{d}t \big) \\ &= \exp\big(- \lambda^{-1} V(X_0^v, 0) - \tfrac{\lambda^{-1}}{2} \textstyle\int_0^T \|u^*(X_s^v, s)\|^2 \, \mathrm{d}s + \lambda^{-1/2} \int_0^T \langle u^*(X_s^v, s), \mathrm{d}B_s^v \rangle \\ &\qquad - \lambda^{-1/2} \textstyle\int_0^T \langle v(X_t^v, t), \mathrm{d}B_t^v \rangle + \frac{\lambda^{-1}}{2} \int_0^T \|v(X_t^v, t)\|^2 \, \mathrm{d}t \big), \end{aligned}$$

and this is equal to $\exp\big(- V(X_0^v, 0) \big)$ when $v = u^*$. Since we condition on $X_0^v = x_{\text{init}}$, we have obtained that the random variable takes constant value $\exp\big(- V(x_{\text{init}}, 0) \big)$ almost surely, which means that its variance is zero. □

D Optimal reparameterization matrices

Theorem 4 (Optimal reparameterization matrices)**.** *Let v be an arbitrary control in $\mathcal{U}$. Define the integral operator $\mathcal{T}_t : L^2([t,T]; \mathbb{R}^{d \times d}) \to L^2([t,T]; \mathbb{R}^{d \times d})$ as*

$$[\mathcal{T}_t(\dot{M}_t)](s) = \int_t^T \dot{M}_t(s') \mathbb{E}\big[\chi(s', X^v, B) \chi(s, X^v, B)^\top \times \alpha(v, X^v, B)\big] \, \mathrm{d}s',$$

where

$$\begin{aligned} \chi(t, X^v, B) := \textstyle\int_t^T \nabla_x f(X_s^v, s) \, \mathrm{d}s + \nabla g(X_T^v) + (\sigma_t^{-1})^\top(t) v(X_t^v, t) \\ - \textstyle\int_t^T \nabla_x b(X_s^v, s) (\sigma_s^{-1})^\top(s) v(X_t^v, t) \, \mathrm{d}s - \int_t^T \nabla_x b(X_s^v, s) (\sigma_s^{-1})^\top(s) \, \mathrm{d}B_s. \end{aligned}$$

If we define $N_t(s) = -\mathbb{E}\big[\big(\nabla g(X_T^v) + \int_t^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s'\big) \chi(t, X^v, B)^\top \times \alpha(v, X^v, B)\big]$, the optimal $M^* = (M_t^*)_{t \in [0,T]}$ is of the form $M_t^*(s) = I + \int_t^s \dot{M}_t^*(s') \, \mathrm{d}s'$, where $\dot{M}_t^*$ is the unique solution of the following Fredholm equation of the first kind:

$$\mathcal{T}_t(\dot{M}_t) = N_t.$$

The proof of (25) shows that minimizing $\mathrm{Var}(w; M)$ is equivalent to minimizing

$$\mathbb{E}\big[\tfrac{1}{T} \textstyle\int_0^T \big\| w(t, v, X^v, B, M_t) \big\|^2 \, \mathrm{d}t \, \alpha(v, X^v, B)\big]. \tag{57}$$

To optimize with respect to M, it is convenient to reexpress it in terms of $\dot{M} = (\dot{M}_t)_{t \in [0,T]}$ as $M_t(s) = I + \int_t^s \dot{M}_t(s') \, \mathrm{d}s'$. By Fubini's theorem, we have that

$$\begin{aligned} \textstyle\int_t^T M_t(s) \nabla_x f(X_s^v, s) \, \mathrm{d}s &= \textstyle\int_t^T \big(I + \int_t^s \dot{M}_t(s') \, \mathrm{d}s' \big) \nabla_x f(X_s^v, s) \, \mathrm{d}s \\ &= \textstyle\int_t^T \nabla_x f(X_s^v, s) \, \mathrm{d}s + \int_t^T \dot{M}_t(s) \int_s^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s' \, \mathrm{d}s, \end{aligned}$$

$$\begin{aligned} &- \textstyle\int_t^T (M_t(s) \nabla_x b(X_s^v, s) - \dot{M}_t(s)) (\sigma^{-1})^\top(s) v(X_s^v, s) \, \mathrm{d}s \\ &= \textstyle\int_t^T \dot{M}_t(s) (\sigma^{-1})^\top(s) v(X_s^v, s) \, \mathrm{d}s - \int_t^T \dot{M}_t(s) \int_s^T \nabla_x b(X_{s'}^v, s') (\sigma_{s'}^{-1})^\top(s') v(X_s^v, s) \, \mathrm{d}s' \, \mathrm{d}s, \end{aligned}$$

$$\begin{aligned} &- \lambda^{1/2} \textstyle\int_t^T (M_t(s) \nabla_x b(X_s^v, s) - \dot{M}_t(s)) (\sigma^{-1})^\top(s) \, \mathrm{d}B_s \\ &= \lambda^{1/2} \big(\textstyle\int_t^T \dot{M}_t(s) (\sigma^{-1})^\top(s) v(X_s^v, s) \, \mathrm{d}s - \int_t^T \dot{M}_t(s) \int_s^T \nabla_x b(X_{s'}^v, s') (\sigma_{s'}^{-1})^\top(s') \, \mathrm{d}B_{s'} \, \mathrm{d}s \big). \end{aligned}$$

Hence, we can rewrite (57) as

$$\begin{aligned} \mathcal{G}(\dot{M}) = \mathbb{E}\big[& \tfrac{1}{T} \textstyle\int_0^T \big\| \sigma(t)^\top \big(\int_t^T \nabla_x f(X_s^v, s) \, \mathrm{d}s + \nabla g(X_T^v) \\ & + \textstyle\int_t^T \dot{M}_t(s) \big(\int_s^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s' + \nabla g(X_T^v) + (\sigma^{-1})^\top(s) v(X_s^v, s) \\ & - \textstyle\int_s^T \nabla_x b(X_{s'}^v, s') (\sigma_{s'}^{-1})^\top(s') v(X_s^v, s) \, \mathrm{d}s' - \int_s^T \nabla_x b(X_{s'}^v, s') (\sigma_{s'}^{-1})^\top(s') \, \mathrm{d}B_{s'} \big) \, \mathrm{d}s \big) \big\|^2 \, \mathrm{d}t \\ & \times \alpha(v, X^v, B) \big] \end{aligned}$$

The first variation $\frac{\delta \mathcal{G}}{\delta \dot{M}}(\dot{M})$ of $\mathcal{G}$ at $\dot{M}$ is defined as the family $Q = (Q_t)_{t\in[0,T]}$ of matrix-valued functions such that for any collection of matrix-valued functions $P = (P_t)_{t\in[0,T]}$,

$$\partial_\epsilon \mathcal{V}(\dot{M} + \epsilon P)|_{\epsilon=0} = \lim_{\epsilon\to 0} \frac{\mathcal{V}(\dot{M}+\epsilon P)-\mathcal{V}(M)}{\epsilon} = \langle P, Q\rangle := \int_0^T \int_t^T \langle P_t(s), Q_t(s)\rangle_F \,\mathrm{d}s\,\mathrm{d}t,$$

where $\dot{M} + \epsilon P := (\dot{M}_t + \epsilon P_t)_{t\in[0,T]}$. Now, note that

$$\begin{aligned}
\partial_\epsilon \mathcal{V}(\dot{M} + \epsilon P)|_{\epsilon=0} &= \partial_\epsilon \mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\|\sigma(t)^\top\big(\int_t^T \nabla_x f(X_s^v, s)\,\mathrm{d}s + \nabla g(X_T^v) \\
&\quad + \int_t^T (\dot{M}_t(s) + \epsilon P_t(s))\big(\int_s^T \nabla_x f(X_{s'}^v, s')\,\mathrm{d}s' + \nabla g(X_T^v) + (\sigma^{-1})^\top(s) v(X_s^v, s) \\
&\qquad - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s') v(X_s^v, s)\,\mathrm{d}s' - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s')\,\mathrm{d}B_{s'}\big)\,\mathrm{d}s\big)\big\|^2 \,\mathrm{d}t \\
&\quad \times \alpha(v, X^v, B)\big]\big|_{\epsilon=0} \\
&= \mathbb{E}\big[\tfrac{2}{T}\int_0^T \big\langle \sigma(t)\sigma(t)^\top\big(\int_t^T \nabla_x f(X_s^v, s)\,\mathrm{d}s + \nabla g(X_T^v) \\
&\quad + \int_t^T \dot{M}_t(s)\big(\int_s^T \nabla_x f(X_{s'}^v, s')\,\mathrm{d}s' + \nabla g(X_T^v) + (\sigma^{-1})^\top(s) v(X_s^v, s) \\
&\qquad - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s') v(X_s^v, s)\,\mathrm{d}s' - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s')\,\mathrm{d}B_{s'}\big)\,\mathrm{d}s\big), \\
&\qquad \int_t^T P_t(s)\big(\int_s^T \nabla_x f(X_{s'}^v, s')\,\mathrm{d}s' + \nabla g(X_T^v) + (\sigma^{-1})^\top(s) v(X_s^v, s) \\
&\qquad - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s') v(X_s^v, s)\,\mathrm{d}s' - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s')\,\mathrm{d}B_{s'}\big)\,\mathrm{d}s\big\rangle \,\mathrm{d}t \\
&\quad \times \alpha(v, X^v, B)\big].
\end{aligned} \tag{58}$$

If we define

$$\begin{aligned}
\chi(s, X^v, B) := \int_s^T \nabla_x f(X_{s'}^v, s')\,\mathrm{d}s' + \nabla g(X_T^v) + (\sigma^{-1})^\top(s) v(X_s^v, s) \\
- \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s') v(X_s^v, s)\,\mathrm{d}s' - \int_s^T \nabla_x b(X_{s'}^v, s')(\sigma_{s'}^{-1})^\top(s')\,\mathrm{d}B_{s'},
\end{aligned}$$

we can rewrite (58) as

$$\begin{aligned}
\partial_\epsilon \mathcal{V}(\dot{M}+\epsilon P)|_{\epsilon=0} = \mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle \sigma(t)\sigma(t)^\top\big(\int_t^T \nabla_x f(X_s^v, s)\mathrm{d}s + \nabla g(X_T^v) + \int_t^T M_t(s)\chi(s, X^v, B)\,\mathrm{d}s\big), \\
\int_t^T P_t(s)\chi(s, X^v, B)\mathrm{d}s\big\rangle \,\mathrm{d}s \times \alpha(v, X^v, B)\big]
\end{aligned} \tag{59}$$

Now let us reexpress equation (59) as:

$$\begin{aligned}
&\mathbb{E}\big[\tfrac{1}{T}\int_0^T \big\langle \sigma\sigma^\top(t)\big(\nabla g(X_T^v) + \int_t^T \big(\nabla_x f(X_s^v, s) + \dot{M}_t(s)\chi(s, X^v, B)\big)\,\mathrm{d}s\big), \\
&\qquad \int_t^T P_t(s)\chi(s, X^v, B)\,\mathrm{d}s\big\rangle \,\mathrm{d}t \times \alpha(v, X^v, B)\big] \\
&\overset{\text{(i)}}{=} \mathbb{E}\big[\tfrac{1}{T}\int_0^T \int_0^s \big\langle P_t(s)\chi(s, X^v, B), \\
&\qquad \sigma\sigma^\top(t)\big(\nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s')\chi(s', X^v, B))\,\mathrm{d}s'\big)\big\rangle \,\mathrm{d}t\,\mathrm{d}s \times \alpha(v, X^v, B)\big] \\
&\overset{\text{(ii)}}{=} \mathbb{E}\big[\tfrac{1}{T}\int_0^T \int_0^s \big\langle \sigma\sigma^\top(t)\big(\nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s')\chi(s', X^v, B))\,\mathrm{d}s'\big)\chi(X^v, s, B)^\top, \\
&\qquad P_t(s)\big\rangle_F \,\mathrm{d}t\,\mathrm{d}s \times \alpha(v, X^v, B)\big] \\
&= \int_0^T \int_0^s \big\langle \tfrac{1}{T}\sigma\sigma^\top(t)\mathbb{E}\big[\big(\nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s')\chi(X^v, s', B))\,\mathrm{d}s'\big)\chi(X^v, s, B)^\top \alpha(v, X^v, B)\big], \\
&\qquad P_t(s)\big\rangle_F \,\mathrm{d}t\,\mathrm{d}s.
\end{aligned} \tag{60}$$

Here, equality (i) holds by Lemma 8 with the choices $\alpha(t, s) = P_t(s)\chi(X^v, s, B)$, $\gamma(t) = \sigma\sigma^\top(t)\big(\nabla g(X_T^v) + \int_t^T \big(\nabla_x f(X_s^v, s) + \dot{M}_t(s)\chi(X^v, s, B)\big)\,\mathrm{d}s$. Equality (ii) follows from the fact that for any matrix A and vectors b, c, $\langle Ab, c\rangle = c^\top Ab = \mathrm{Tr}(c^\top Ab) = \mathrm{Tr}(Abc^\top) = \langle B, cb^\top\rangle_F$, where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product.

The first-order necessary condition for optimality states that at the optimal $\dot{M}^*$, the first variation $\frac{\delta \mathcal{G}}{\delta \dot{M}}(\dot{M}^*)$ is zero. In other words, $\partial_\epsilon \mathcal{V}(\dot{M} + \epsilon P)|_{\epsilon=0}$ is zero for any P. Hence, the right-hand side of (60) must be zero for any P, which implies that almost everywhere with respect to $t \in [0,T]$, $s \in [s,T]$,

$$\mathbb{E}\big[\big(\nabla g(X_T^v)+\int_t^T (\nabla_x f(X_{s'}^v, s')+\dot{M}_t(s')\chi(X^v, s', B)) \, \mathrm{d}s'\big)\chi(X^v, s, B)^\top \alpha(v, X^v, B)\big] = 0.$$

To derive this, we also used that $\sigma(t)$ is invertible by assumption.

Define the integral operator $\mathcal{T}_t : L^2([t,T];\mathbb{R}^{d\times d}) \to L^2([t,T];\mathbb{R}^{d\times d})$ as

$$[\mathcal{T}_t(\dot{M}_t)](s) = \int_t^T \dot{M}_t(s')\mathbb{E}\big[\chi(X^v, s', B)\chi(X^v, s, B)^\top \times \alpha(v, X^v, B)\big] \, \mathrm{d}s'$$

If we define $N_t(s) = -\mathbb{E}\big[\big(\nabla g(X_T^v) + \int_t^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s'\big)\chi(X^v, s, B)^\top \times \alpha(v, X^v, B)\big]$, the problem that we need to solve to find the optimal $\dot{M}_t$ is

$$\mathcal{T}_t(\dot{M}_t) = N_t.$$

This is a Fredholm equation of the first kind.

Lemma 8. *If $\alpha, \beta : [0,T] \times [0,T] \to \mathbb{R}^d$, $\gamma : [0,T] \to \mathbb{R}^d$, $\delta : [0,T] \to \mathbb{R}^{d\times d}$ are arbitrary integrable functions, we have that*

$$\int_0^T \big\langle \int_t^T \alpha(t,s) \, \mathrm{d}s, \gamma(t)\big\rangle \, \mathrm{d}t = \int_0^T \int_0^s \big\langle \alpha(t,s), \gamma(t)\big\rangle \, \mathrm{d}t \, \mathrm{d}s,$$

Proof. We have that:

$$\begin{aligned}
&\int_0^T \int_t^T \big\langle \alpha(t,s), \gamma(t)\big\rangle \, \mathrm{d}s \, \mathrm{d}t \overset{\text{(i)}}{=} \int_0^T \int_0^{T-t} \big\langle \alpha(t, T-s), \gamma(t)\big\rangle \, \mathrm{d}s \, \mathrm{d}t \\
&\overset{\text{(ii)}}{=} \int_0^T \int_0^t \big\langle \alpha(T-t, T-s), \gamma(T-t)\big\rangle \, \mathrm{d}s \, \mathrm{d}t \overset{\text{(iii)}}{=} \int_0^T \int_s^T \big\langle \alpha(T-t, T-s), \gamma(T-t)\big\rangle \, \mathrm{d}t \, \mathrm{d}s \\
&\overset{\text{(iv)}}{=} \int_0^T \int_{T-s}^T \big\langle \alpha(T-t, s), \gamma(T-t)\big\rangle \, \mathrm{d}t \, \mathrm{d}s \overset{\text{(v)}}{=} \int_0^T \int_0^s \big\langle \alpha(t,s), \gamma(t)\big\rangle \, \mathrm{d}t \, \mathrm{d}s
\end{aligned}$$

Here, in equalities (i), (ii), (iv) and (v) we make changes of variables of the form $t \mapsto T-t$, $s \mapsto T-s$, $s' \mapsto T-s'$. In equality (iii) we use Fubini's theorem. □

E Control warm-starting

We introduce the *Gaussian warm-start*, a control warm-start strategy that we adapt from [56], and that we use in our experiments in Figure 7. Their work tackles generalized Schrödinger bridge problems, which are different from the control setting in that the final distribution is known and there is no terminal cost. The following proposition, that provides an analytic expression of the control needed for the density of the process to be Gaussian at all times, is the foundation of our method.

Proposition 6. *Given $Z \sim N(0, I)$ define the random process Y as*

$$Y_t = \mu(t) + \tilde{\Gamma}(t)Z, \qquad \textit{where } \mu(t) \in \mathbb{R}^d, \ \tilde{\Gamma}(t) = \sqrt{t}\Gamma(t) \in \mathbb{R}^{d\times d}. \tag{61}$$

Define the control $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ as

$$u(x,t) = \sigma(t)^{-1}\big(\partial_t\mu(t) + \big(\big(\partial_t\Gamma(t)\big)\Gamma(t)^{-1} + \tfrac{I-(\sigma\sigma^\top)(t)(\Sigma\Sigma^\top)^{-1}(t)}{2t}\big)(x-\mu(t)) - b(x,t)\big) \tag{62}$$

Then, if $\Gamma_0 = \sigma(0)$, the controlled process X^u defined in equation (2) *has the same marginals as Y. That is, for all $t \in [0,T]$,* $\mathrm{Law}(Y_t) = \mathrm{Law}(X_t^u)$.

Proof. Following [56], we have that

$$\begin{aligned}
\partial_t X_t = \partial_t\mu_t + \partial_t\tilde{\Gamma}(t)Z = \partial_t\mu(t) + (\partial_t\tilde{\Gamma}(t))\tilde{\Gamma}(t)^{-1}(X_t - \mu(t)), \\
\nabla \log p_t(x) = -\tilde{\Sigma}(t)^{-1}(x-\mu(t)), \qquad \tilde{\Sigma}(t) = \tilde{\Gamma}(t)\tilde{\Gamma}(t)^\top.
\end{aligned}$$

Now, p_t satisfies the continuity equation equation

$$\partial_t p_t = -\nabla \cdot ((\partial_t\mu(t) + (\partial_t\tilde{\Gamma}(t))\tilde{\Gamma}(t)^{-1}(x-\mu(t)))p_t) \tag{63}$$

Let $D(t) = \frac{1}{2}\sigma(t)\sigma(t)^\top$. We want to reexpress (63) as a Fokker-Planck equation of the form

$$\partial_t p_t = -\nabla \cdot (v(x,t)p_t) + \sum_{i=1}^d \sum_{j=1}^d \partial_i \partial_j (D_{ij}(t)p_t) = -\nabla \cdot (v(x,t)p_t) + \sum_{i=1}^d \partial_i \sum_{j=1}^d (D_{ij}(t)\partial_j p_t)$$

$$\begin{aligned} &= -\nabla \cdot (v(x,t)p_t) + \nabla \cdot (D(t)\nabla p_t) = -\nabla \cdot (v(x,t)p_t) + \nabla \cdot (D(t)\nabla \log p_t(x) p_t) \\ &= -\nabla \cdot ((v(x,t) - D(t)\nabla \log p_t(x))p_t). \end{aligned}$$

Hence, we need that

$$\begin{aligned} v(x,t) - D(t)\nabla \log p_t &= \partial_t \mu(t) + (\partial_t \tilde{\Gamma}(t))\tilde{\Gamma}(t)^{-1}(x - \mu(t)), \\ \implies v_t(x) &= \partial_t \mu(t) + ((\partial_t \tilde{\Gamma}(t))\tilde{\Gamma}(t)^{-1}(x - \mu(t)) + \tfrac{(\sigma\sigma^\top)(t)}{2} \nabla \log p_t(x) \\ &= \partial_t \mu(t) + (\partial_t \tilde{\Gamma}(t))\tilde{\Gamma}(t)^{-1}(x - \mu(t)) - \tfrac{(\sigma\sigma^\top)(t)}{2}\Sigma(t)^{-1}(x - \mu(t)). \end{aligned}$$

If we let $\tilde{\Gamma}(t) = \Gamma(t)\sqrt{t}$, then $\tilde{\Sigma}(t) = t\Gamma(t)\Gamma(t)^\top = t\Sigma(t)$ and $\partial_t \tilde{\Gamma}(t) = \partial_t \Gamma(t)\sqrt{t} + \frac{\Gamma(t)}{2\sqrt{t}}$. That is,

$$\begin{aligned} v(x,t) &= \partial_t \mu(t) + \big(\partial_t \Gamma(t)\sqrt{t} + \tfrac{\Gamma(t)}{2\sqrt{t}}\big)\tfrac{\Gamma(t)^{-1}}{\sqrt{t}}(x - \mu(t)) - \tfrac{(\sigma\sigma^\top)(t)}{2}\tfrac{\Sigma(t)^{-1}}{t}(x - \mu(t)) \\ &= \partial_t \mu(t) + \big(\partial_t \Gamma(t)\big)\Gamma(t)^{-1}(x - \mu(t)) + \tfrac{1}{2t}(x - \mu(t)) - \tfrac{(\sigma\sigma^\top)(t)\Sigma(t)^{-1}}{2t}(x - \mu(t)) \end{aligned}$$

For v to be finite at $t = 0$, we need that $(\sigma\sigma^\top)(0)\Sigma(0)^{-1} = I$, which holds, for example, if $\Gamma(0) = \sigma(0)$. Also, to match the form of (2), we need that

$$\begin{aligned} &v(x,t) = b(x,t) + \sigma(t)u(x,t), \\ \implies &u(x,t) = \sigma(t)^{-1}\big(\partial_t \mu_t + \big(\big(\partial_t \Gamma(t)\big)\Gamma(t)^{-1} + \tfrac{I - (\sigma\sigma^\top)(t)\Sigma(t)^{-1}}{2t}\big)(x - \mu_t) - b(x,t)\big). \end{aligned}$$

□

The warm-start control is computed as the solution of a *Restricted Gaussian Stochastic Optimal Control* problem, where we constrain the space of controls to those that induce Gaussian paths as described in Prop. 6. In practice, we learn a linear spline $\mu = (\mu^{(b)})_{b=0}^{\mathcal{B}}$, where $\mu^{(b)} \in \mathbb{R}^d$, and a linear spline $\Gamma = (\Gamma^{(b)})_{b=0}^{\mathcal{B}}$, where $\Gamma^{(b)} \in \mathbb{R}^{d \times d}$. These linear splines take the role of $\mu(t)$ and $\Sigma(t)$ in (61). Given splines μ and Γ, we obtain the warm-start control using (62); for a given $t \in [0,T)$, if we let $b_- = \lfloor \mathcal{B}t/T \rfloor$, $b_+ = b_- + 1$, $\Delta = T/\mathcal{B}$, we have that

$$\widehat{\mu}(t) = \frac{(t - b_-\Delta)\mu^{(b_+)} + (b_+\Delta - t)\mu^{(b_-)}}{\Delta}, \qquad \widehat{\partial_t \mu}(t) = \frac{\mu^{(b_+)} - \mu^{(b_-)}}{\Delta}, \tag{64}$$

$$\widehat{\Gamma}(t) = \frac{(t - b_-\Delta)\Gamma^{(b_+)} + (b_+\Delta - t)\Gamma^{(b_-)}}{\Delta}, \qquad \widehat{\partial_t \Gamma}(t) = \frac{\Gamma^{(b_+)} - \Gamma^{(b_-)}}{\Delta}, \tag{65}$$

$$\hat{u}(x,t) = \sigma(t)^{-1}\big(\widehat{\partial_t \mu}(t) + \big(\widehat{\partial_t \Gamma}(t)\widehat{\Gamma}(t)^{-1} + \tfrac{I - (\sigma\sigma^\top)(t)(\widehat{\Sigma}\widehat{\Sigma}^\top)^{-1}(t)}{2t}\big)(x - \widehat{\mu}(t)) - b(x,t)\big). \tag{66}$$

Algorithm 3 provides a method to learn the splines μ, Γ. It is a stochastic optimization algorithms in which the spline parameters are updated by sampling Y_t in (61) at different times, computing the control cost relying on (66), and taking its gradient.

Algorithm 3 Restricted Gaussian Stochastic Optimal Control

```
Input: State cost $f(x,t)$, terminal cost $g(x)$, diffusion coeff. $\sigma(t)$, base drift $b(x,t)$, noise level $\lambda$, number of
    iterations $N$, batch size $m$, number of time steps $K$, number of spline knots $\mathcal{B}$, initial mean spline knots
    $\mu_0 = (\mu_0^{(b)})_{b=0}^{\mathcal{B}}$, initial noise spline knots $\Gamma_0 = (\Gamma_0^{(b)})_{b=0}^{\mathcal{B}}$.
for $n = 0 : (N-1)\}$ do
    Sample $m$ i.i.d. variables $(Z_i)_{i=1}^n \sim N(0,I)$ and $m$ times $(t_i)_{i=1}^n \sim \mathrm{Unif}([0,T])$.
    for $j = 0 : K$ do
        Set $t_j = jT/K$, and compute $\widehat{\mu}_n(t_j)$, $\widehat{\partial_t \mu}_n(t_j)$, $\widehat{\Gamma}_n(t_j)$, $\widehat{\partial_t \Gamma}_n(t_j)$ according to (64), (65) using $\mu_n$,
          $\Gamma_n$
        for $i = 1 : m$ do compute $Y_{ij} = \hat{\mu}(t_j) + \sqrt{t_j}\widehat{\Gamma}(t_j)Z_i$ and $\hat{u}_n(Y_{ij}, t_j)$ using (66);
    end
    Compute $\hat{\mathcal{L}}_{\mathrm{RGSOC}}(\mu_n, \Gamma_n) = \frac{1}{m}\sum_{i=1}^m \big(\frac{T}{K}\sum_{j=0}^{K-1}\big(\frac{1}{2}\|\hat{u}(Y_{ij}, t_j)\|^2 + f(Y_{ij}, t_j)\big) + g(Y_{iK})\big)$
    Compute the gradient of $\hat{\mathcal{L}}_{\mathrm{RGSOC}}(\mu_n, \Gamma_n)$ with respect to the spline parameters $(\mu_n, \Gamma_n)$.
    Obtain $\mu_{n+1}, \Gamma_{n+1}$ with via an Adam update on $\mu_n, \Gamma_n$ resp. (or another stochastic algorithm)
end
Output: Learned splines $\mu_N, \Gamma_N$, control $\hat{u}_N$
```

Once we have access to the restricted control $\hat{u}_N$, we can warm-start the control in Algorithms 1 and 2 by introducing $\hat{u}_N$ as an offset. That is, we parameterize the control as $u_\theta = \hat{u}_N + \tilde{u}_\theta$.

F Experimental details and additional plots

F.1 Experimental details

The control L^2 error curves show the following quantity:

$$\mathbb{E}_{t,\mathbb{P}^{u^*}}[\|u^*(X_t^{u^*}, t) - u(X_t^{u^*}, t)\|^2 e^{-\lambda^{-1} V(X_0^{u^*}, 0)}]/\mathbb{E}_{t,\mathbb{P}^{u^*}}[e^{-\lambda^{-1} V(X_0^{u^*}, 0)}]$$

That is, we sample trajectories using the optimal control, and compute the error using a Monte Carlo estimate. In all our experiments, the distribution $X_0^{u^*}$ is a delta, which means that we do not need to compute $V(X_0^{u^*}, 0)$. We keep an exponential moving average (EMA) estimate of the control L^2 error, which we show in the plots. To compute it, we sample ten batches of optimally controlled trajectories every 10 training iterations, and we update the quantity with the average of the ten batches, using EMA coefficient 0.02. All other quantities shown in the plots are also smoothed out using EMA with coefficient 0.01, except for control objective values, which are computed as the average of 65536 samples, every 5000 training steps.

For all losses and all settings, we train the control using Adam with learning rate 1×10^{-4}. For SOCM, we train the reparametrization matrices using Adam with learning rate 1×10^{-2}. We use batch size $m = 128$ unless otherwise specified. When used, we run the warm-start algorithm (Algorithm 3) with $\mathcal{B} = 20$ knots, $K = 200$ time steps, and batch size $m = 512$, and we use Adam with learning rate 3×10^{-4} for $N = 60000$ iterations.

QUADRATIC ORNSTEIN-UHLENBECK The choices for the functions of the control problem are:

$$b(x,t) = Ax, \quad f(x,t) = x^\top P x, \quad g(x) = x^\top Q x, \quad \sigma(t) = \sigma_0.$$

where Q is a positive definite matrix. Control problems of this form are better known as linear quadratic regulator (LQR) and they admit a closed form solution [78, Thm. 6.5.1]. The optimal control is given by:

$$u_t^*(x) = -2\sigma_0^\top F_t x,$$

where F_t is the solution of the Ricatti equation

$$\frac{dF_t}{dt} + A^\top F_t + F_t A - 2F_t \sigma_0 \sigma_0^\top F_t + P = 0$$

with the final condition $F_T = Q$. Within the QUADRATIC OU class, we consider two settings:

- Easy: We set $d = 20$, $A = 0.2I$, $P = 0.2I$, $Q = 0.1I$, $\sigma_0 = I$, $\lambda = 1$, $T = 1$, $x_{\text{init}} = 0.5N(0, I)$. We do not use warm-start for any algorithm. We take $K = 50$ time discretization steps, and we use random seed 0.
- Hard: We set $d = 20$, $A = I$, $P = I$, $Q = 0.5I$, $\sigma_0 = I$, $\lambda = 1$, $T = 1$, $x_{\text{init}} = 0.5N(0, I)$. We use the *Gaussian warm-start* (App. E). We take batch size $m = 64$ and $K = 150$ time discretization steps, and we use random seed 0.

LINEAR ORNSTEIN-UHLENBECK The functions of the control problem are chosen as follows:

$$b(x,t) = Ax, \quad f(x,t) = 0, \quad g(x) = \langle \gamma, x \rangle, \quad \sigma(t) = \sigma_0.$$

The optimal control for this class of problems is given by [59, Sec. A.4]:

$$u_t^*(x) = -\sigma_0^\top e^{A^\top (T-t)} \gamma.$$

We use exactly the same functions as [59]: we sample $(\xi_{ij})_{1 \leq i,j \leq d}$ once at the beginning of the simulation, and set:

$$d = 10, \quad A = -I + (\xi_{ij})_{1 \leq i,j \leq d}, \quad \gamma = \mathbb{1}, \quad \sigma_0 = I + (\xi_{ij})_{1 \leq i,j \leq d},$$
$$T = 1, \quad \lambda = 1, \quad x_{\text{init}} = 0.5N(0, I).$$

We take $K = 100$ time discretization steps, and we use random seed 0.

DOUBLE WELL We also use exactly the same functions as [59], which are the following:

$$b(x,t) = -\nabla\Psi(x), \quad \Psi(x) = \sum_{i=1}^{d} \kappa_i (x_i^2 - 1)^2, \quad f(x) = 0, \quad g(x) = \sum_{i=1}^{d} \nu_i (x_i^2 - 1)^2, \quad \sigma_0 = \mathrm{I},$$

where $d = 10$, and $\kappa_i = 5$, $\nu_i = 3$ for $i \in \{1,2,3\}$ and $\kappa_i = 1$, $\nu_i = 1$ for $i \in \{4,\dots,10\}$. We set $T = 1$, $\lambda = 1$ and $x_{\mathrm{init}} = 0$. We take $K = 200$ time discretization steps, and we use random seed 0. The Double Well problem is actually highly non-trivial, and is multimodal. The only reason we can produce a "ground truth" control to compare to in this setting is that we use significant knowledge of the problem; we analytically reduce it to 1D problems by decoupling each dimension and apply numerical methods to solve the Hamilton-Jacobi-Bellman equation for these 1D problems. It is not a problem where we actually have the ground truth control in closed form.

PATH INTEGRAL SAMPLER ON MIXTURE OF GAUSSIANS We set

$$b(x,t) = 0, \qquad f(x,t) = 0, \qquad g(x) = \log(\mu^0(x)/\mu(x)) = -\tfrac{\|x\|^2}{2} - \tfrac{d}{2}\log(2\pi) - \log\mu(x),$$

where $T = 1$, and μ is the density of a mixture of two Gaussians with means $\pm e_1$, where $e_1 = (1,0,\dots,0)$, and variance Id. Note that we take μ to be normalized, i.e. $\int \mu(x)\,\mathrm{d}x = 1$, or equivalently, $\log Z = \log\big(\int \mu(x)\,\mathrm{d}x\big) = 0$. In Figure 1, we use the following Monte Carlo estimator of the control objective at the control u:

$$\hat{S}^u(X) = \textstyle\int_0^T \big(\tfrac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\big)\,\mathrm{d}t + g(X_T^u) + \int\langle u(X_t^u,t), \mathrm{d}B_t\rangle.$$

Note that this estimator is unbiased because $\mathbb{E}[\int\langle u(X_t^u,t),\mathrm{d}B_t\rangle] = 0$. This is known as the Sticking the Landing estimator, as it has zero variance when u is the optimal control [72]. The fact that $\mathbb{E}[-\hat{S}^u(X)] \leq \log Z = 0$ with equality when $u = u^*$ is stated as [84, Thm. 4].

F.2 Model architectures

As a general guideline, the control function can be thought of as the analog of the score function in diffusion models; hence, a natural choice for the architecture can be U-Nets or diffusion transformers if the control task is on images, audio or video. Other domains may require different architectures. In the experiments we report, we used the architecture implemented in the class `FullyConnectedUNet` within the file `SOC_matching/models.py`. It is a simplified version of the U-Net architecture where both the down-sampling and up-sampling layers are fully connected with ReLU activations, and the horizontal layers are linear transformations. We use three down-sampling and up-sampling steps, with widths 256, 128 and 64 (hence, the first down-sampling step is actually an up-sampling, because the data dimensions in our experiments range from 10 to 20).

The reparameterization matrices have an unusual trait, which is that their input dimension is small (two) while their output dimension is large (d^2). Hence, the kind of functions that they need to learn are low dimensional and hence easy. In our case, we used the architecture implemented in the class `SigmoidMLP` within the file `SOC_matching/models.py`, which is essentially a three layer multilayer perceptron with ReLU activations and output dimension d^2, whose output is averaged with the identity matrix using sigmoid weights, in order to enforce that $M_t(t)$ be the identity matrix.

F.3 Additional tables and plots

Table 1 shows the average times per iteration for each algorithm. Each algorithm was run using a 16GB V100 GPU.

SOCM	SOCM $M_t = I$	SOCM adj.	Adj.	Cross entropy	Log-variance	Moment	Variance
0.222	0.090	0.099	0.169	0.086	0.117	0.087	0.086

Table 1: Time per iteration (exponential moving average) for various algorithms in seconds per iteration, for the QUADRATIC OU (EASY) experiments (Figure 2).

Figure 4 shows the control objective (1) for the four settings. The error bars for the control objective plots show the confidence intervals for ± one standard deviation, computed via a Monte Carlo estimate using 65536 trajectories per data point. They show the standard error of the mean. As expected, SOCM also obtains the lowest values for the control objective, up to the estimation error.

Figure 5 shows the normalized standard deviation of the importance weight for the learned control u: $\sqrt{\mathrm{Var}[\alpha(u, X^u, B)]}/\mathbb{E}[\alpha(u, X^u, B)]$. By Lemma 1, when $X_0^u = x_{\mathrm{init}}$ for an arbitrary x_{init} (which is the case for all our experiments), this quantity is zero for the optimal control u^*. Hence, the normalized standard deviation of α is an alternative metric to measure the optimality of the learned control.

Figure 6 shows an exponential moving average of the norm squared of the gradient for LINEAR OU and DOUBLE WELL. For LINEAR OU, the minimum gradient norm is achieved by the adjoint method, while for DOUBLE WELL it is achieved by the cross entropy loss. The training instabilities of the adjoint method become apparent as well. Interestingly, in both settings the algorithms with smallest gradients are not SOCM, which is the algorithm with smallest error as shown in Figure 3. Understanding this phenomenon is outside of the scope of this paper.

Figure 7 shows plots of the control L^2 error, the norm squared of the gradient, and the control objective for the QUADRATIC OU (HARD) setting, using a warm-start strategy detailed in App. E. Figure 7 shows that SOCM is once again the algorithm that achieves the lowest error and the smallest gradients. Remark that the warm-start control is a reasonable approximation of the optimal control, as the initial control L^2 error is much lower than in the other figures.

Figure 8 shows the value of the training loss for SOCM and its two ablations: SOCM with constant $M_t = I$, and SOCM-Adjoint. For all such algorithms, the training loss is the sum of the L^2 error of the learned control u, and the expected conditional variance of the matching vector field w. Thus, the difference between the training loss plots and the L^2 error plots is the expected conditional variance of w. We observe that the expected conditional variance in the QUADRATIC OU setting is orders of magnitude smaller for SOCM than for its two ablations. For LINEAR OU, SOCM and SOCM-adjoint have similar expected conditional variance, and a possible explanation is that the LINEAR OU setting is very simple. In the DOUBLE WELL setting, the SOCM-adjoint training loss curve has spikes that are probably caused by instabilities of the adjoint method. These spikes can be attributed mostly to the expected conditional variance term, since the corresponding L^2 error curve in Figure 3 does not present them.

Figure 9 shows that the instabilities of the adjoint method are inherent to the loss, because they also appear at small learning rates: 3×10^{-5} is smaller than the learning rates typically used for Adam, which hover from 1×10^{-4} to 1×10^{-3}.

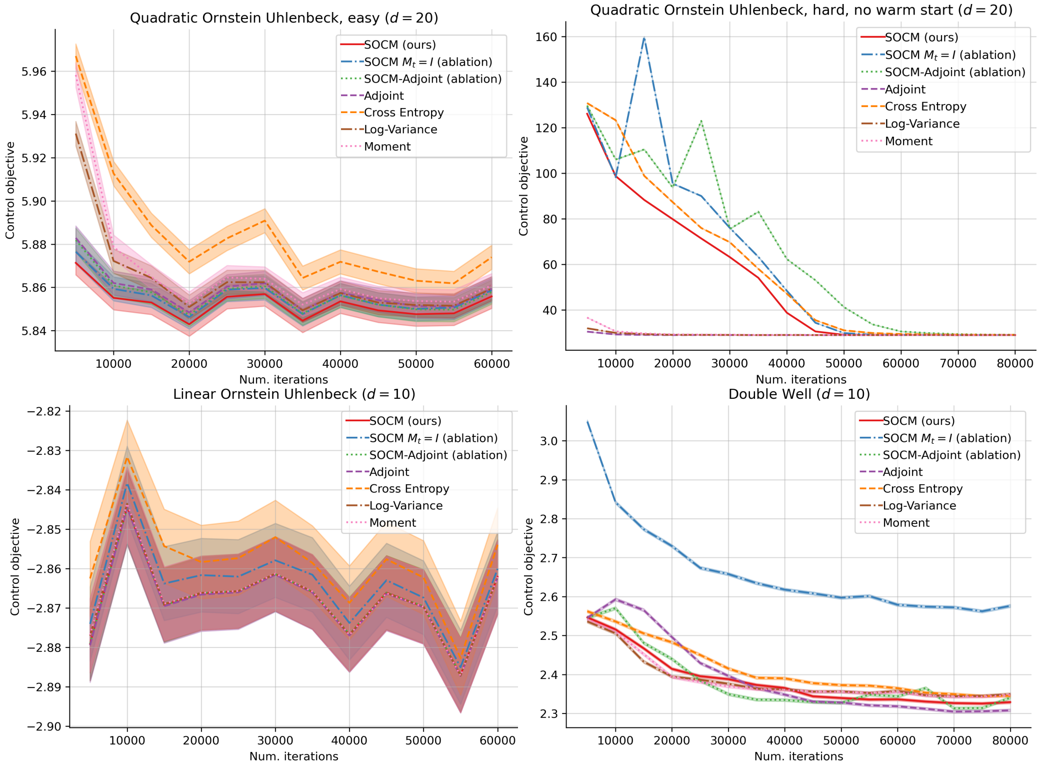

Figure 4: Plots of the control objective for the four settings.

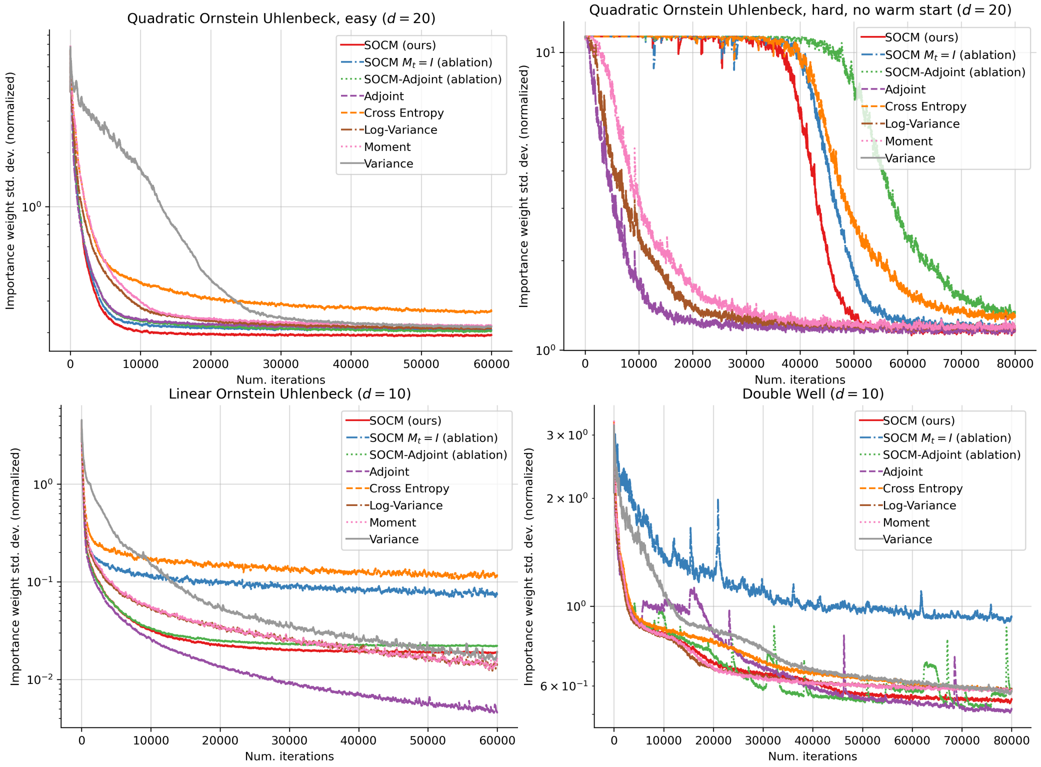

Figure 5: Plots of the normalized standard deviation of the importance weights: $\sqrt{\mathrm{Var}[\alpha(u, X^u, B)]}/\mathbb{E}[\alpha(u, X^u, B)]$.

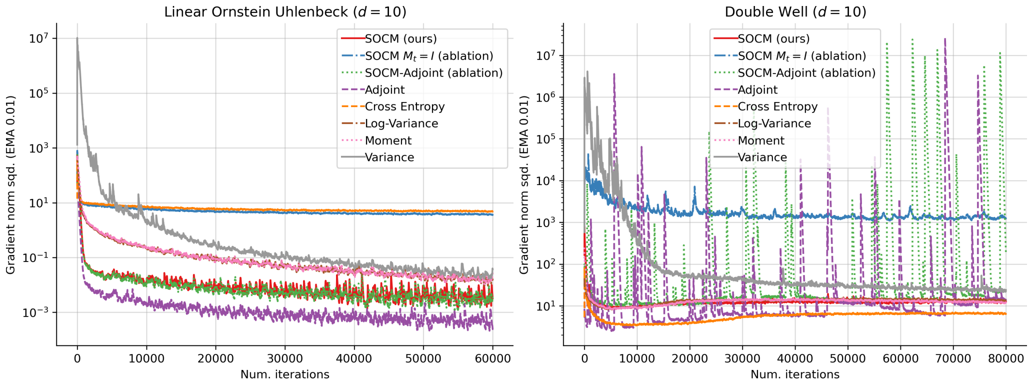

Figure 6: Plots of the norm squared of the gradient for the LINEAR ORNSTEIN UHLENBECK and DOUBLE WELL settings.

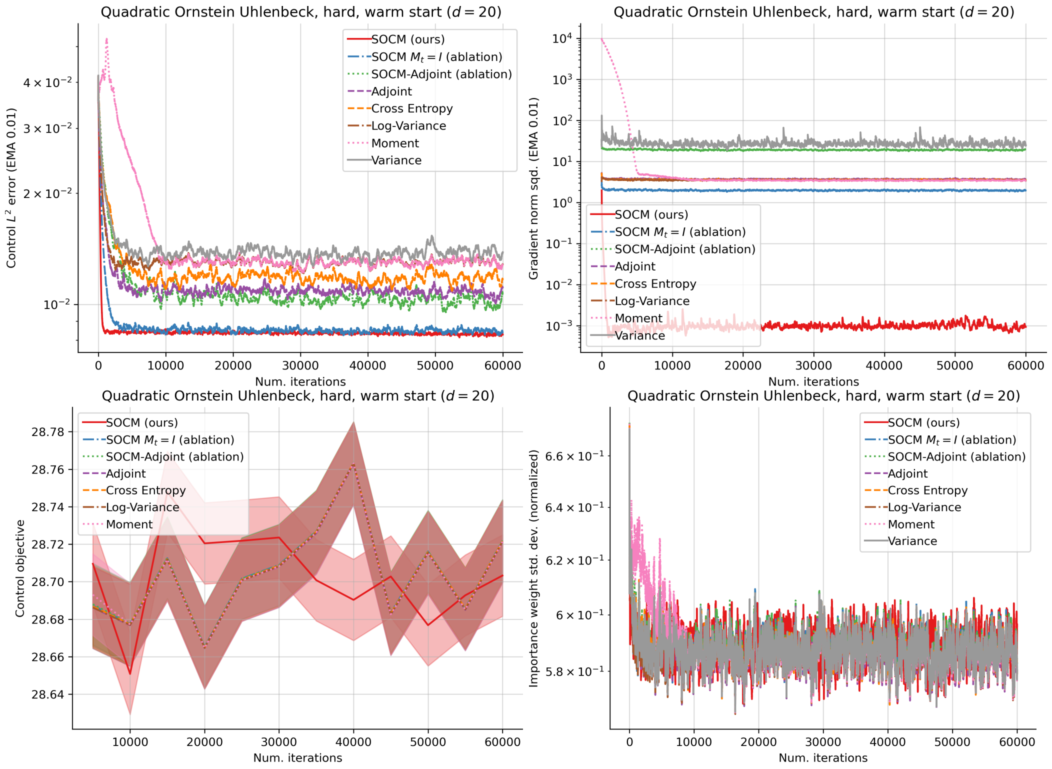

Figure 7: Plots of the control L^2 error, the norm squared of the gradient, and the control objective for the QUADRATIC ORNSTEIN-UHLENBECK (HARD) setting, without using warm-start.

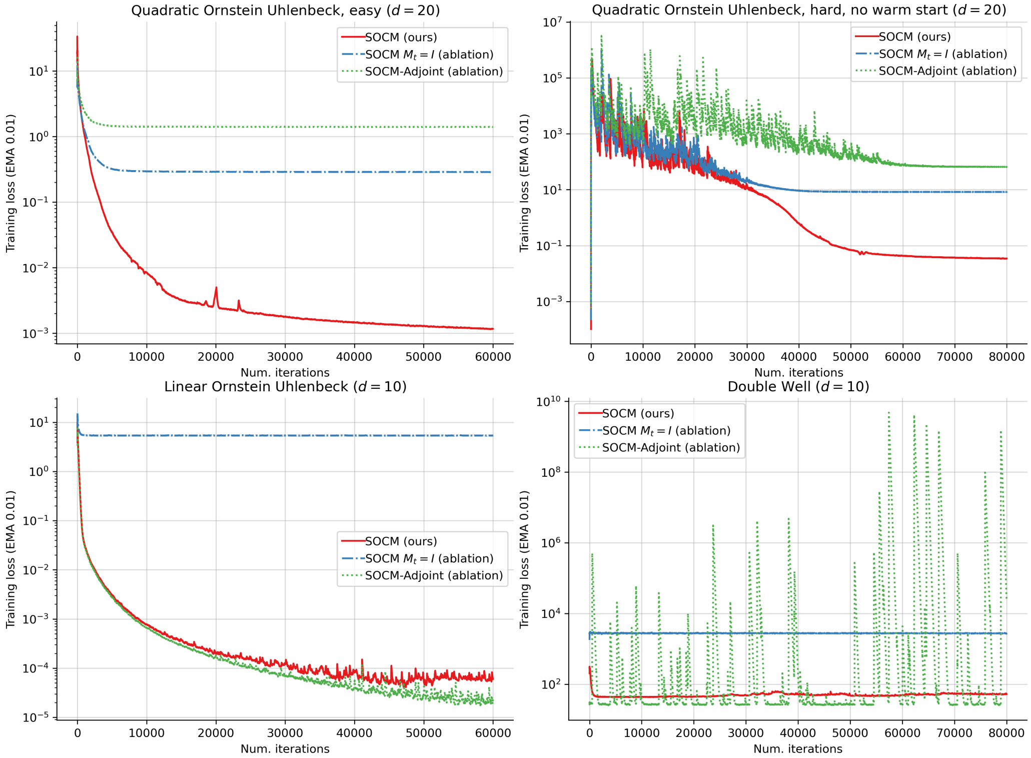

Figure 8: Plots of the training loss for SOCM and its two ablations: SOCM with constant $M_t = I$, and SOCM-Adjoint.

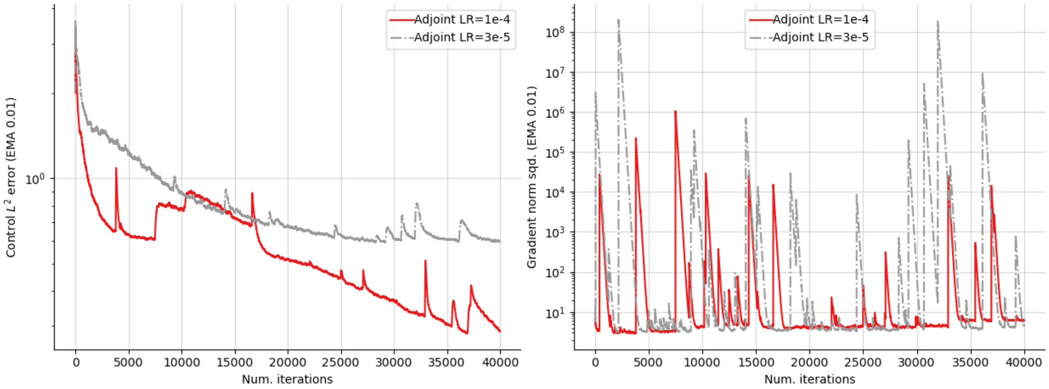

Figure 9: Plots of the control L^2 error and the norm squared of the gradient for the adjoint method on DOUBLE WELL, for two different values of the Adam learning rate. The instabilities of the adjoint method persist for small learning rates, signaling an inherent issue with the loss.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In the abstract and introduction we state that we introduce SOCM, a novel algorithm to solve stochastic optimal control problems, and we claim that it outperforms all existing algorithms in three out of four settings. All of these claims are supported by our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, in the second-to-last paragraph of Sec. 5 we discuss the main limitation of SOCM (our algorithm), which is that the variance of the importance weight α is too large in certain settings. Regarding the computational efficiency of our algorithm, Table 1 shows a comparison between our algorithm and existing ones.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

   Justification: All the theoretical results stated in the paper are numbered and have a corresponding detailed proof in the appendices. In the main text, we indicate the precise location of the proofs. We provide a proof sketch for Theorem 1 in the main paper, and a full proof in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We provide a description of experimental details in Subsec. F.1, and of model architectures in Subsec. F.2. We also provide a link to the GitHub repository that contains the code for this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We also provide a link to the GitHub repository that contains the code for this paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We provide a description of experimental details in Subsec. F.1, and of model architectures in Subsec. F.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: Most of the plots in the paper do not contain error bars, because the lines show exponential moving averages, which means that the error is already smoothed out and very small. The error bars for the control objective plots do show the confidence intervals for ±

one standard deviation, computed via a Monte Carlo estimate using 65536 trajectories per data point. They show the standard error of the mean. We state this in Subsec. F.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We explain the type and amount of GPUs we use in Subsec. F.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We reviewed the NeurIPS Code of Ethics and our research conforms to it. We made sure to preserve our anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

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

Answer: [No]

Justification: The research presented is foundational, and the code released does not have direct societal impact. Yet, it may serve as the basis to develop algorithms that improve the quality of generative models. We state this in Sec. 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [No]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We did not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [Yes]

    Justification: We present code implementing our algorithm. The code folder contains a README which explains how to reproduce the experiments. We describe our algorithm throughout the paper, and provide details in App. F.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [No]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [No]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.