# **Stochastic Optimal Control Matching**

Carles Domingo-Enrich<sup>1,2</sup>, Jiequn Han<sup>3</sup>, Brandon Amos<sup>2</sup>, Joan Bruna<sup>1,3</sup>, Ricky T. Q. Chen<sup>2</sup>

<sup>1</sup>Courant Institute of Mathematical Sciences, New York University, <sup>2</sup>Meta AI, <sup>3</sup>Flatiron Institute

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.

Date: April 19, 2024 Correspondence: Carles Domingo-Enrich at cd2754@nyu.edu Code: https://github.com/facebookresearch/SOC-matching

 $\infty$ 

## 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 (Hartmann et al., 2014; Hartmann and Schütte, 2012; Zhang et al., 2014; Holdijk et al., 2023), finance and economics (Pham, 2009; Fleming and Stein, 2004), stochastic filtering and data assimilation (Mitter, 1996; Reich, 2019), nonconvex optimization (Chaudhari et al., 2018), power systems and energy markets (Belloni et al., 2016; Powell and Meisel, 2016), and robotics (Theodorou et al., 2011; Gorodetsky et al., 2018). Stochastic optimal has also been very impactful in neighboring fields such as mean-field games (Carmona et al., 2018), optimal transport (Villani, 2003, 2008), backward stochastic differential equations (BSDEs) (Carmona, 2016) and large deviations (Feng and Kurtz, 2006).

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 (Nüsken and Richter, 2021) (see subsection 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 Chen et al. (2018) (building on top of Rezende and Mohamed (2015)), 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 (Chen et al., 2018, 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 (Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021) and flow-matching, a.k.a. stochastic interpolant, methods (Lipman et al., 2022; Albergo and Vanden-Eijnden, 2022; Pooladian et al., 2023; Albergo et al., 2023), which are currently the preferred algorithms to train CNFs. Aside from architectural improvements such as the UNet (Ronneberger et al., 2015), 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 2.1) as the training loss (see (7)). As in ML-CNFs, computing the gradient of this loss requires backpropagating through the computation of the trajectories of the SDE (2.1), 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 2). 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  $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \ge 0}, \mathcal{P})$  be a fixed filtered probability space on which is defined a Brownian motion  $B = (B_t)_{t \ge 0}$ . We consider the control-affine problem

$$\min_{u \in \mathcal{U}} \mathbb{E} \Big[ \int_0^T \left( \frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \right) dt + g(X_T^u) \Big],$$
  
where  $dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t)) dt + \sqrt{\lambda}\sigma(t) dB_t, \qquad X_0^u \sim p_0.$ 

and where  $X_t^u \in \mathbb{R}^d$  is the state,  $u : \mathbb{R}^d \times [0, T]$  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 Appendix 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 Appendix B and refer the reader to Oksendal (2013, Chap. 11) and Nüsken and Richter (2021, 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  $J(u; x, t) := \mathbb{E}\left[\int_t^T \left(\frac{1}{2} \|u_s(X_s^u)\|^2 + f_s(X_s^u)\right) dt + g(X_T^u) |X_t^u = x\right]$ . 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 a point x and time t is defined as the minimum value of the cost functional across all possible controls:

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

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:

$$\begin{aligned} &(\partial_t + L)V(x,t) - \frac{1}{2} \| (\sigma^\top \nabla V)(x,t) \|^2 + f(x,t) = 0, \\ &V(x,T) = g(x). \end{aligned}$$

The verification theorem (Pavliotis, 2014, 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 (2.1) of the problem (2.1)-(2.1). An implication of the verification theorem is that for every  $u \in \mathcal{U}$ ,

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

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 (2.1) can be deduced by integrating the HJB equation (2.1) over [t,T], and taking the conditional expectation with respect to  $X_t^u = x$ . We include the proof of (2.1) in Appendix B for completeness.

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

$$dX_t = b(X_t, t) dt + \sqrt{\lambda \sigma(t)} dB_t, \qquad X_0 \sim p_0,$$
  
$$dY_t = (-f(X_t, t) + \frac{1}{2} ||Z_t||^2) dt + \sqrt{\lambda} \langle Z_t, dB_t \rangle, \qquad Y_T = g(X_T).$$

where  $Y : \Omega \times [0,T] \to \mathbb{R}$  and  $Z : \Omega \times [0,T] \to \mathbb{R}^d$  are progressively measurable <sup>1</sup> 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 (2.1). We include the proof in Appendix B for completeness.

An analytic expression for the value function From the forward-backward equations (2.1)-(2.1), one can derive a closed-form expression for the value function V:

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

where  $X_t$  is the solution of the uncontrolled SDE (2.1). This is a classical result, but we still include its proof in Appendix B. Given that  $u^*(x,t) = -\sigma(t)^\top \nabla V(x,t)$ , an immediate, yet important, consequence of (2.1) is the following representation of the optimal control:

Lemma 1 (Path-integral representation of the optimal control (Kappen, 2005)).

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

Remark that the right-hand side of this equation involves the gradient of logarithm of a conditional expectation. This is reminiscent of the vector fields that are learned when training diffusion models or flow matching algorithms. For example, the target vector field for variance-exploding score-based diffusion loss (Song et al., 2021) can be expressed as  $\nabla_x \log p_t(x) = \nabla_x \log \mathbb{E}_{Y \sim p_{\text{data}}} \left[ \frac{\exp(-||x-Y||^2/(2\sigma_t^2))}{(2\pi\sigma_t^2)^{d/2}} \right]$ . Note, however, that in (1) the gradient is taken with respect to the initial condition of the process, which requires the development of novel techniques.

<sup>&</sup>lt;sup>1</sup>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 Karatzas and Shreve (1991)).

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 (2.1) 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 (2.1), and define the work functional

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

It turns out (Lemma 4 in Appendix B) that the Radon-Nikodym derivative  $\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}$  satisfies  $\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) = \exp(\lambda^{-1}(V(X_0,0) - \mathcal{W}(X,0)))$ . Also, a straight-forward application of the Girsanov theorem for SDEs (Cor. 1) shows that

$$\frac{d\mathbb{P}^{u}}{d\mathbb{P}^{u^{*}}}(X^{u^{*}}) = \exp\left(-\lambda^{-1/2} \int_{0}^{T} \langle u^{*}(X_{t}^{u^{*}}, t) - u(X_{t}^{u^{*}}, t), \mathrm{d}B_{t} \rangle - \frac{\lambda^{-1}}{2} \int_{0}^{T} \|u^{*}(X_{t}^{u^{*}}, t) - u(X_{t}^{u^{*}}, t)\|^{2} \, \mathrm{d}t \right),$$

which means that the only control  $u \in \mathcal{U}$  such that  $\mathbb{P}^u = \mathbb{P}^{u^*}$  is the optimal control itself. Such changes of process are the basic tools to design IDO losses, and we leverage them as well.

#### 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 (2.1). The main approaches include *finite difference methods* (Bonnans et al., 2004; Ma and Ma, 2020; Baňas et al., 2022), which approximate the derivatives and gradients of the value function using finite differences, *finite element methods* (Jensen and Smears, 2013), which involve restricting the solution to domain-dependent function spaces, and semi-Lagrangian schemes (Debrabant and Jakobsen, 2013; Carlini et al., 2020; Calzola et al., 2022), which trace back characteristics and have better stability than finite difference methods. See Greif (2017) for an overview on these techniques, and Baňas et al. (2022) for a comparison between them. Hutzenthaler et al. (2016) 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 (Beck et al., 2019; Hutzenthaler et al., 2019, 2018; Hutzenthaler and Kruse, 2020).

High dimensional methods leveraging FBSDEs The FBSDE formulation in equations (2.1)-(2.1) has given rise to multiple methods to learn controls. One such approach is *least-squares Monte Carlo* (see Pham (2009, Chapter 3) and Gobet (2016) for an introduction, and Gobet et al. (2005); Zhang et al. (2004) for an extensive analysis), where trajectories from the forward process (2.1) 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. (2017); Han et al. (2018). They parameterize the control using a neural network  $u_{\theta}$ , 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 (2.1) with initial condition  $y_0$  and control  $u_{\theta}$ . 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 (Beck et al., 2018; Chan-Wai-Nam et al., 2019; Zhou et al., 2021), control (Becker et al., 2019; Hartmann et al., 2019), multi-agent games (Han and Hu, 2020; Carmona and Laurière, 2021, 2022); see E et al. (2021) 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 (2.1), 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 (2021), 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 Appendix B.

#### 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  $\mathcal{L} \in \{\mathcal{L}_{\mathrm{Adj}}(7), \mathcal{L}_{\mathrm{CE}}(7), \mathcal{L}_{\mathrm{Var}_v}(7), \mathcal{L}_{\mathrm{Mom}_v}^{\log}(7), \mathcal{L}_{\mathrm{Mom}_v}(7)\}$ 

- 2 Simulate m trajectories of the process  $X^v$  controlled by  $v = u_{\theta_n}$ , e.g., using Euler-Maruyama updates
- 3 if  $\mathcal{L} \neq \mathcal{L}_{Adj}$  then detach the *m* trajectories from the computational graph, so that gradients do not backpropagate;
- 4 Using the *m* trajectories, compute an *m*-sample Monte Carlo approximation  $\hat{\mathcal{L}}(u_{\theta_n})$  of the loss  $\mathcal{L}(u_{\theta_n})$
- **5** Compute the gradients  $\nabla_{\theta} \hat{\mathcal{L}}(u_{\theta_n})$  of  $\hat{\mathcal{L}}(u_{\theta_n})$  w.r.t.  $\theta_n$
- **6** Obtain  $\theta_{n+1}$  with via an Adam update on  $\theta_n$  (or another stochastic algorithm)

7 end

**Output:** Learned control  $u_{\theta_N}$ 

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 5 in Appendix B, or Hartmann and Schütte (2012); Kappen et al. (2012)):

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

This is exactly the control objective in (2.1). This connection has been studied extensively (Bierkens and Kappen, 2014; Gómez et al., 2014; Hartmann and Schütte, 2012; Kappen et al., 2012; Rawlik et al., 2013). Hence, the relative entropy loss is a very natural one, and is widely used; see Onken et al. (2023); Zhang and Chen (2022) for some examples on multiagent systems and sampling.

Solving optimization problems of the form (7) has a long history that dates back to Pontryagin (1962). Note that  $\mathcal{L}_{Adj}(u)$  depends on u both explicitly, and implicitly through the process  $X^u$ . To compute the gradient  $\nabla_{\theta} \hat{\mathcal{L}}_{Adj}(u_{\theta_n})$  of a Monte Carlo approximation  $\hat{\mathcal{L}}_{Adj}(u_{\theta_n})$  of  $\mathcal{L}_{Adj}(u_{\theta_n})$  as required by Algorithm 1, we need to backpropagate through the simulation of the m trajectories, which is why we do *not* detach them from the computational graph. One can alternatively compute the gradient  $\nabla_{\theta} \hat{\mathcal{L}}_{Adj}(u_{\theta_n})$  by explicitly solving an ODE, a technique which is known as the *adjoint method*. The adjoint method was introduced by Pontryagin (1962), popularized in deep learning by Chen et al. (2018), and further developed for SDEs in Li et al. (2020).

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 Appendix B):

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

The cross-entropy loss has a rich literature (Hartmann et al., 2017; Kappen and Ruiz, 2016; Rubinstein and Kroese, 2013; Zhang et al., 2014) and has been recently used in applications such as molecular dynamics (Holdijk et al., 2023).

Furthermore, we note that the cross-entropy loss can be significantly simplified and written in terms of the  $L^2$  error of the control u with respect to the optimal control  $u^*$ :

**Lemma 2** (Cross-entropy loss in terms of control  $L^2$  error).

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

This characterization, which is proven in Prop. 3(ii) in Appendix 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  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}(u) = \operatorname{Var}_{\mathbb{P}^{v}}(\frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}})$  and  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}^{\log}(u) = \operatorname{Var}_{\mathbb{P}^{v}}(\log \frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}})$  whenever  $\mathbb{E}_{\mathbb{P}^{v}}|\frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}}| < +\infty$  and  $\mathbb{E}_{\mathbb{P}^{v}}|\log \frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}}| < +\infty$ ,

<sup>1</sup> for  $n \in \{0, \dots, N-1\}$  do

respectively. Define

$$\begin{split} \tilde{Y}_{T}^{u,v} &= -\lambda^{-1} \int_{0}^{T} \langle u(X_{t}^{v},t), v(X_{t}^{v},t) \rangle \,\mathrm{d}t - \lambda^{-1} \int_{0}^{T} f(X_{t}^{v},t) \,\mathrm{d}t - \lambda^{-1/2} \int_{0}^{T} \langle u(X_{t}^{v},t), \mathrm{d}B_{t} \rangle \\ &+ \frac{\lambda^{-1}}{2} \int_{0}^{T} \| u(X_{t}^{v},t) \|^{2} \,\mathrm{d}t. \end{split}$$

Then,  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}$  and  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}^{\log}$  are equivalent, respectively, to the following losses (see Lemma 6):

$$\mathcal{L}_{\operatorname{Var}_{v}}(u) := \operatorname{Var}\left(\exp\left(\tilde{Y}_{T}^{u,v} - \lambda^{-1}g(X_{T}^{v})\right)\right)$$
$$\mathcal{L}_{\operatorname{Var}_{v}}^{\log}(u) := \operatorname{Var}\left(\tilde{Y}_{T}^{u,v} - \lambda^{-1}g(X_{T}^{v})\right),$$

The variance and log-variance losses were introduced by Nüsken and Richter (2021). Unlike for the crossentropy loss, the choice of the control v does lead to different losses. When using  $\mathcal{L}_{\operatorname{Var}_v}$  or  $\mathcal{L}_{\operatorname{Var}_v}^{\log}$  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

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

where  $\tilde{Y}_T^{u,v}$  is defined in (7). Note the similarity with the log-variance loss (7); 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 (7) yields exactly the log-variance loss. The moment loss was introduced by Hartmann et al. (2019, Section III.B), and it is a generalization of the FBSDE method pioneered by E et al. (2017); Han et al. (2018) and referenced earlier in this subsection. In fact, the original method corresponds to setting v = 0.

### 3 Stochastic Optimal Control Matching

w

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 subsection 2.2. The general idea is to leverage the analytic expression of  $u^*$  in Lemma 1 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  $M_t(t) = \text{Id.}$  Let  $v \in \mathcal{U}$  be an arbitrary control. Let  $\mathcal{L}_{\text{SOCM}} : L^2(\mathbb{R}^d \times [0,T];\mathbb{R}^d) \times L^2([0,T]^2;\mathbb{R}^{d \times d}) \to \mathbb{R}$  be the loss function defined as

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

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

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

 $\mathcal{L}_{\text{SOCM}}$  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  $\alpha$  as the *importance weight*. We present a proof sketch of Theorem 1; the full proofs for all the results in this section are in Appendix C.

*Proof sketch of Theorem 1* Let X be the uncontrolled process (2.1). Consider the loss

$$\begin{split} \tilde{\mathcal{L}}(u) &= \mathbb{E} \left[ \frac{1}{T} \int_0^T \left\| u(X_t, t) - u^*(X_t, t) \right\|^2 \mathrm{d}t \, \exp\left( -\lambda^{-1} \int_0^T f(X_t, t) \, \mathrm{d}t - \lambda^{-1} g(X_T) \right) \right] \\ &= \mathbb{E} \left[ \frac{1}{T} \int_0^T \left( \left\| u(X_t, t) \right\|^2 - 2\langle u(X_t, t), u^*(X_t, t) \rangle + \left\| u^*(X_t, t) \right\|^2 \right) \mathrm{d}t \\ &\quad \times \exp\left( -\lambda^{-1} \int_0^T f(X_t, t) \, \mathrm{d}t - \lambda^{-1} g(X_T) \right) \right]. \end{split}$$

Clearly, the only optimum of this loss is the optimal control  $u^*$ . Using the analytic expression of  $u^*$  in Lemma 1, the cross-term can be rewritten as (see Lemma 7 in Appendix C):

$$\mathbb{E}\left[\frac{1}{T}\int_{0}^{T} \langle u(X_{t},t), u^{*}(X_{t},t)\rangle \, \mathrm{d}t \, \exp\left(-\lambda^{-1}\int_{0}^{T} f(X_{t},t) \, \mathrm{d}t - \lambda^{-1}g(X_{T})\right)\right]$$
  
=  $\lambda \mathbb{E}\left[\frac{1}{T}\int_{0}^{T} \langle u(X_{t},t), \sigma(t)^{\top} \nabla_{x} \mathbb{E}\left[\exp\left(-\lambda^{-1}\int_{t}^{T} f(X_{s},s) \, \mathrm{d}s - \lambda^{-1}g(X_{T})\right)\right| X_{t} = x\right] \rangle \, \mathrm{d}t$   
 $\times \exp\left(-\lambda^{-1}\int_{0}^{t} f(X_{s},s) \, \mathrm{d}s\right)\right].$ 

It remains to evaluate the conditional expectation  $\nabla_x \mathbb{E}\left[\exp\left(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\right)|X_t = x\right]$ , 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  $M_t(t) = \text{Id}$ . We have that

$$\nabla_{x} \mathbb{E} \Big[ \exp \left( -\lambda^{-1} \int_{t}^{T} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} g(X_{T}) \right) \Big| X_{t} = x \Big]$$
  
=  $\mathbb{E} \Big[ \Big( -\lambda^{-1} \int_{t}^{T} M_{t}(s) \nabla_{x} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} M_{t}(T) \nabla g(X_{T}) + \lambda^{-1/2} \int_{t}^{T} (M_{t}(s) \nabla_{x} b(X_{s}, s) - \partial_{s} M_{t}(s)) (\sigma^{-1})^{\top}(s) \mathrm{d}B_{s} \Big) \times \exp \Big( -\lambda^{-1} \int_{t}^{T} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} g(X_{T}) \Big) \Big| X_{t} = x \Big].$ 

We prove a more general form of this result (Prop. 4) in subsection C.2 and also provide an intuitive derivation in subsection 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 (1) into the right-hand side of (3), and then this back into (3), and we complete the square, we obtain that for some constant K independent of u,

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

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

The following proposition sheds some light onto the role of reparameterization matrices and connects the SOCM loss to the cross-entropy loss.

**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:

$$\mathcal{L}_{\text{SOCM}}(u, M) = \underbrace{\mathbb{E}\left[\frac{1}{T} \int_{0}^{T} \left\| u(X_{t}^{u^{*}}, t) - u^{*}(X_{t}^{u^{*}}, t) \right\|^{2} dt \exp(-\lambda^{-1}V(X_{0}^{u^{*}}, 0))\right]}_{Bias \ of \ u} + \underbrace{\underbrace{\text{CondVar}(w; M)}_{Variance \ of \ w}}_{Variance \ of \ w}$$

where

$$\begin{aligned} \operatorname{Var}(w; M) \\ &= \mathbb{E} \Big[ \frac{1}{T} \int_0^T \left\| \tilde{w}(t, X, B, M_t) - \frac{\mathbb{E}[\tilde{w}(t, X, B, M_t) \exp(-\lambda^{-1} \mathcal{W}(X, 0)) | X_t]}{\mathbb{E}[\exp(-\lambda^{-1} \mathcal{W}(X, 0)) | X_t]} \right\|^2 \mathrm{d}t \, \exp(-\lambda^{-1} \mathcal{W}(X, 0)) \Big] \\ &= \mathbb{E} \Big[ \frac{1}{T} \int_0^T \left\| w(t, v, X^v, B, M_t) - \frac{\mathbb{E}[w(t, v, X^v, B, M_t) \alpha(v, X^v, B) | X_t^v]}{\mathbb{E}[\alpha(v, X^v, B) | X_t^v]} \right\|^2 \mathrm{d}t \, \alpha(v, X^v, B) \Big], \end{aligned}$$

and

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

Remark that the bias term in equation (2) is equal to the characterization of the cross-entropy loss in Lemma 2. In other words, the landscape of  $\mathcal{L}_{SOCM}(u, M)$  with respect to u is the landscape of the cross-entropy loss  $\mathcal{L}_{CE}(u)$ . Thus, the SOCM loss can be seen as some form of variance reduction method for the cross-entropy loss, and performs substantially better experimentally (section 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  $\mathcal{L}_{\text{SOCM}}$  in (5)

1 for  $n \in \{0, \ldots, N-1\}$  do

- 2 Simulate m trajectories of the process  $X^v$  controlled by  $v = u_{\theta_n}$ , e.g., using Euler-Maruyama updates
- $\mathbf{z}$  Detach the *m* trajectories from the computational graph, so that gradients do not backpropagate
- 4 Using the *m* trajectories, compute an *m*-sample Monte-Carlo approximation  $\hat{\mathcal{L}}_{SOCM}(u_{\theta_n}, M_{\omega_n})$  of the loss  $\mathcal{L}_{SOCM}(u_{\theta_n}, M_{\omega_n})$  in (5)
- **5** Compute the gradients  $\nabla_{(\theta,\omega)} \hat{\mathcal{L}}_{SOCM}(u_{\theta_n}, M_{\omega_n})$  of  $\hat{\mathcal{L}}_{SOCM}(u_{\theta_n}, M_{\omega_n})$  at  $(\theta_n, \omega_n)$
- **6** Obtain  $\theta_{n+1}$ ,  $\omega_{n+1}$  with via an Adam update on  $\theta_n$ ,  $\omega_n$ , resp.
- 7 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  $\frac{1}{T} \int_0^T ||u(X_t^v, t) - w(t, v, X^v, B, M_t)||^2 dt \times \alpha(v, X^v, B)$ , 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  $\alpha$  We want to use a vector field v such that  $\operatorname{Var}[\alpha(v, X^v, B)]$  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. (2017), but we include it in subsection C.5 for completeness.

**Lemma 3.** When we set  $v = u^*$ , the conditional variance  $\operatorname{Var}[\alpha(v, X^v, B)|X_0^v = x_{\operatorname{init}}]$  is zero for any  $x_{\operatorname{init}} \in \mathbb{R}^d$ .

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  $u_{\rm WS}(x,t) + u_{\theta}(x,t)$ , where the warm-start  $u_{\rm WS}$  is a reasonably good control obtained via a different strategy (see Appendix D).

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 CondVar(w; M) in the right-hand side of equation (2). Since CondVar(w; M) does not depend on the specific v, the optimal M does not depend on v either. And since the first term in the right-hand side of equation (2) does not depend on  $M = (M_t)_{t \in [0,T]}$ , minimizing CondVar(w; M) is equivalent to minimizing  $\mathcal{L}(u)$  with respect to M. In practice, we parameterize M using a neural network with a two-dimensional input (t, s) and a  $d^2$ -dimensional output.

Furthermore, the following theorem 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. The proof is in subsection C.6.



**Figure 1** Plots of the  $L^2$  error incurred by the learned control (top), and the norm squared of the gradient with respect to the parameters  $\theta$  of the control (*bottom*), for the QUADRATIC ORNSTEIN UHLENBECK (EASY) setting and for each IDO loss. Both plots show exponential moving averages computed from the trajectories used during training.

**Theorem 2** (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

$$\mathcal{T}_{t}(\dot{M}_{t})](s) = \int_{t}^{T} \dot{M}_{t}(s') \mathbb{E} \left[ \chi(s', X^{v}, B) \chi(s, X^{v}, B)^{\top} \times \alpha(v, X^{v}, B) \right] \mathrm{d}s'$$

where

$$\begin{split} \chi(t, X^{v}, B) &:= \int_{t}^{T} \nabla_{x} f(X^{v}_{s}, s) \, \mathrm{d}s + \nabla g(X^{v}_{T}) + (\sigma^{-1}_{t})^{\top}(t) v(X^{v}_{t}, t) \\ &- \int_{t}^{T} \nabla_{x} b(X^{v}_{s}, s) (\sigma^{-1}_{s})^{\top}(s) v(X^{v}_{t}, t) \, \mathrm{d}s - \int_{t}^{T} \nabla_{x} b(X^{v}_{s}, s) (\sigma^{-1}_{s})^{\top}(s) \, \mathrm{d}B_{s}. \end{split}$$

If we define  $N_t(s) = -\mathbb{E}\left[\left(\nabla g(X_T^v) + \int_t^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s'\right) \chi(t, X^v, B)^\top \times \alpha(v, X^v, B)\right]$ , 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:

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

Solving the Fredholm equation (2) numerically is expensive, as 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, this is a computation that must be done only once and that may be affordable in some settings.

Parameterizing the matrices  $M_t$  In practice, we parameterize the matrices  $(M_t)_{t \in [0,T]}$  using a common function  $M_{\omega}$  with two arguments (t,s). A simple way to enforce that  $M_{\omega}(t,t) = \text{Id}$  is to set  $M_{\omega}(t,s) = e^{-\gamma(s-t)}\text{Id} + (1 - e^{-\gamma(s-t)})\tilde{M}_{\tilde{\omega}}(t,s)$ , 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.

### 4 Experiments

We consider four experimental settings that we adapt from Nüsken and Richter (2021): QUADRATIC ORNSTEIN UHLENBECK (EASY), QUADRATIC ORNSTEIN UHLENBECK (HARD), LINEAR ORNSTEIN UHLENBECK and DOUBLE WELL. We describe them in detail in Appendix E. 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 1 (top) we plot the control  $L^2$  error for each IDO algorithm described in subsection 2.2, and for the SOCM algorithm (Algorithm 2), for the QUADRATIC OU (EASY) setting. 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)



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



**Figure 3** Plots of the  $L^2$  error incurred by the learned control (top), and the norm squared of the gradient with respect to the parameters  $\theta$  of the control (*bottom*), for the QUADRATIC ORNSTEIN UHLENBECK (HARD) setting and for each IDO loss. All the algorithms use a warm-started control (see Appendix D).

SOCM	SOCM $M_t = I$	SOCM adjoint	Adjoint
0.222	0.090	0.099	0.169
Cross entropy	Log-variance	Moment	Variance
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 1).

SOCM-Adjoint, where we estimate the conditional expectation in equation (1) using the adjoint method for SDEs instead of the path-wise reparameterization trick (see subsection C.4).

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 1 (*bottom*) we show the squared norm of the gradient of each loss with respect to the parameters  $\theta$  of the control: algorithms with small noise variance tend to have low error values. Table 1 shows the average times per iteration for each algorithm.

In Figure 2, 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 second smallest error, slightly behind the adjoint method, but the latter shows instabilities. As we show in Figure 7 in Appendix E, these instabilities are inherent to the adjoint method and they do not disappear for small learning rates. Both in Figure 1 and Figure 7, we observe that learning the reparameterization matrices is critical to obtain gradient estimates with high signal-to-noise ratio.

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,  $u_{\theta_n}$  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 Appendix D and Figure 3, we showcase a control warm-start strategy that can help and speed up convergence.

# 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. 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 four different control 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 control  $L^2$  error for the SOCM and cross-entropy losses remains high and fluctuates heavily due to the large variance of  $\alpha$ . The large variance of  $\alpha$  is due to the mismatch between the probability measures induced by the learned control and the optimal control. Similar problems are encountered in out-of-distribution generalization for reinforcement learning, and some approaches may be carried over from that area (Munos et al., 2016).

## References

- Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2022.
- 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.
- 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.
- 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.
- 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.
- Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. Journal of Machine Learning Research, 20, 2019.
- 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.
- Joris Bierkens and Hilbert J Kappen. Explicit solution of relative entropy weighted control. Systems & Control Letters, 72:36–43, 2014.
- 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.
- 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.
- 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.
- René Carmona. Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications, volume 1. SIAM, 2016.
- 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.
- 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.
- René Carmona, François Delarue, et al. Probabilistic Theory of Mean Field Games with Applications I-II. Springer, 2018.
- Quentin Chan-Wai-Nam, Joseph Mikael, and Xavier Warin. Machine learning for semilinear PDEs. Journal of Scientific Computing, 79(3):1667–1712, 2019.
- 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.
- 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.
- 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.
- 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.
- 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.

- Jin Feng and Thomas G Kurtz. Large deviations for stochastic processes. Number 131. American Mathematical Soc., 2006.
- Wendell H Fleming and Jerome L Stein. Stochastic optimal control, international finance and debt. Journal of Banking & Finance, 28(5):979–996, 2004.
- Emmanuel Gobet. Monte-Carlo methods and stochastic processes: from linear to non-linear. CRC Press, 2016.
- 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.
- 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.
- 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.
- Constantin Greif. Numerical methods for hamilton-jacobi-bellman equations. 2017.
- 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.
- 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.
- Carsten Hartmann and Christof Schütte. Efficient rare event simulation by optimal nonequilibrium forcing. Journal of Statistical Mechanics: Theory and Experiment, 2012(11):P11004, 2012.
- Carsten Hartmann, Ralf Banisch, Marco Sarich, Tomasz Badowski, and Christof Schütte. Characterization of rare events in molecular dynamics. *Entropy*, 16(1):350–376, 2014.
- Carsten Hartmann, Lorenz Richter, Christof Schütte, and Wei Zhang. Variational characterization of free energy: Theory and algorithms. *Entropy*, 19(11), 2017.
- 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.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Advances in Neural Information Processing Systems, volume 33. Curran Associates, Inc., 2020.
- 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.
- 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.
- 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.
- Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, et al. Multilevel picard iterations for solving smooth semilinear parabolic heat equations. arXiv preprint arXiv:1607.03295, 2016.
- 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.
- 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.
- 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.
- H J Kappen. Path integrals and symmetry breaking for optimal control theory. Journal of Statistical Mechanics: Theory and Experiment, 2005(11), nov 2005.
- Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. Machine learning, 87(2):159–182, 2012.

- Hilbert Johan Kappen and Hans Christian Ruiz. Adaptive importance sampling for control and inference. Journal of Statistical Physics, 162(5):1244–1266, 2016.
- I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Graduate Texts in Mathematics (113) (Book 113). Springer New York, 1991.
- Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning*, 2021.
- 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.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2022.
- Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos A. Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching, 2023.
- 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.
- Sanjoy K Mitter. Filtering and stochastic control: A historical perspective. *IEEE Control Systems Magazine*, 16(3): 67–76, 1996.
- Remi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016.
- 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.
- Bernt Oksendal. Stochastic differential equations: an introduction with applications. Springer Science & Business Media, 2013.
- 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.
- Grigorios A Pavliotis. Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations, volume 60. Springer, 2014.
- Huyên Pham. Continuous-time stochastic control and optimization with financial applications, volume 61. Springer Science & Business Media, 2009.
- L.S. Pontryagin. The Mathematical Theory of Optimal Processes. Interscience Publishers, 1962.
- 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.
- 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.
- 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.
- Sebastian Reich. Data assimilation: The Schrödinger perspective. Acta Numerica, 28:635–711, 2019.
- Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.
- 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.
- 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.

- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. arXiv preprint arXiv:1907.05600, 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Scorebased generative modeling through stochastic differential equations. In International Conference on Learning Representations (ICLR 2021), 2021.
- 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.
- Ramon Van Handel. Stochastic calculus, filtering, and stochastic control. Course notes, URL http://www. prince- ton. edu/rvan/acm217/ACM217, 2007.
- C. Villani. Topics in Optimal Transportation. Graduate studies in mathematics. American Mathematical Society, 2003.
- C. Villani. Optimal Transport: Old and New. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.
- Jianfeng Zhang et al. A numerical scheme for BSDEs. The annals of applied probability, 14(1):459–488, 2004.
- Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In International Conference on Learning Representations, 2022.
- 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.
- 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.

# Appendix

# Contents

Α	Technical assumptions	16
В	Proofs of section 2	16
С	Proofs of section 3C.1Proof of Theorem 1 and Prop. 2C.2Proof of the path-wise reparameterization trick (Prop. 1)C.3Informal derivation of the path-wise reparameterization trickC.4SOCM with the adjoint method for SDEsC.5Proof of Lemma 3C.6Proof of Theorem 2	<b>21</b> 21 23 26 28 29 29
D	Control warm-starting	31
Е	Experimental details and additional plots         E.1       Experimental details         E.2       Model architectures         E.3       Additional plots	<b>33</b> 33 34 34
		04

# A Technical assumptions

Throughout our work, we make the same assumptions as Nüsken and Richter (2021), 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  $\sigma$  are continuously differentiable,  $\sigma$  has bounded first-order spatial derivatives, and  $(\sigma\sigma^{\top})(x,s)$  is positive definite for all  $(x,s) \in \mathbb{R}^d[0,T]$ . Furthermore, there exist constants  $C, c_1, c_2 > 0$  such that

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

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

# B Proofs of section 2

*Proof of* (2.1) By Itô's lemma, we have that

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

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 (2.1),

$$\begin{split} \partial_{s}V(X^{u}_{s},s) &+ \langle b(X^{u}_{s},s) + \sigma(X^{u}_{s},s)u(X^{u}_{s},s), \nabla V(X^{u}_{s},s) \rangle \\ &+ \frac{\lambda}{2} \sum_{i,j=1}^{d} (\sigma\sigma^{\top})_{ij}(X^{u}_{s},s) \partial_{x_{i}} \partial_{x_{j}} V(X^{u}_{s},s) \\ &= \frac{1}{2} \| (\sigma^{\top} \nabla V)(X^{u}_{s},s) \|^{2} - f(X^{u}_{s},s) + \langle \sigma(X^{u}_{s},s)u(X^{u}_{s},s), \nabla V(X^{u}_{s},s) \rangle \\ &= \frac{1}{2} \| (\sigma^{\top} \nabla V)(X^{u}_{s},s) + u(X^{u}_{s},s) \|^{2} - \frac{1}{2} \| u(X^{u}_{s},s) \|^{2} - f(X^{u}_{s},s), \end{split}$$

and this implies that

$$g(X_T^u) - V(X_t^u, t) = \int_t^T \left(\frac{1}{2} \| (\sigma^\top \nabla V)(X_s^u, s) + u(X_s^u, s) \|^2 - \frac{1}{2} \| u(X_s^u, s) \|^2 - f(X_s^u, s) \right) \mathrm{d}s + S_t^u$$

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

*Proof of* (2.1)-(2.1) By Itô's lemma, we have that

$$dV(X_s,s) = \left(\partial_s V(X_s,s) + \langle b(X_s,s), \nabla V(X_s,s) \rangle + \frac{\lambda}{2} \sum_{i,j=1}^d (\sigma \sigma^\top)_{ij} (X_s,s) \partial_{x_i} \partial_{x_j} V(X_s,s) \right) \mathrm{d}s + \sqrt{\lambda} \nabla V(X_s^u,s)^\top \sigma(X_s^u,s) \,\mathrm{d}B_s,$$

Note that by (2.1),

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

Plugging this into (B) concludes the proof.

Proof of (2.1) Since 
$$Y_s = V(X_s, s)$$
 and  $Z_s = \sigma^{\top}(s)\nabla V(X_s, s) = -u^*(X_s, s)$  satisfy (2.1), we have that

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

Hence, recalling the definition of the work functional in (2.1), we have that

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

By Novikov's theorem (Thm. 3), we have that

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

which concludes the proof of (2.1).

**Theorem 3** (Novikov's theorem). Let  $\theta_s$  be a locally- $\mathcal{H}_2$  process which is adapted to the natural filtration of the Brownian motion  $(B_t)_{t\geq 0}$ . Define

$$Z(t) = \exp\left(\int_0^t \theta_s \,\mathrm{d}B_s - \frac{1}{2}\int_0^t \|\theta_s\|^2 \,\mathrm{d}s\right).$$

If for each  $t \geq 0$ ,

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

then for each  $t \geq 0$ ,

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

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 4** (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 (3) and suppose that the assumptions of Theorem 3 hold. Let  $(\Omega, \mathcal{F})$  be the  $\sigma$ -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 (3). Under the probability measure  $\mathbb{Q}$ , the stochastic process  $\{\tilde{W}(t)\}_{0 \leq t \leq T}$  defined as

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

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  $(t_{i+1} - t_i)I$ , 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:

$$\begin{aligned} & \mathbb{E}_{\mathbb{Q}}[\exp(\langle \alpha, \tilde{W}(t_{i+1}) - \tilde{W}(t_i) \rangle)] \\ & := \mathbb{E}_{\mathbb{P}}\Big[\exp\left(\langle \alpha, W(t_{i+1}) - \int_{0}^{t_{i+1}} \theta_s \, \mathrm{d}s - W(t_i) + \int_{0}^{t_i} \theta_s \, \mathrm{d}s \rangle\Big) Z(T)\Big] = \exp\left(\frac{(t_{i+1} - t_i) \|\alpha\|^2}{2}\right). \end{aligned}$$

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

$$dX_t = b_1(X_t, t) dt + \sigma(X_t, t) dB_t, \qquad X_0 = x_{\text{init}} dY_t = (b_1(Y_t, t) + b_2(Y_t, t)) dt + \sigma(Y_t, t) dB_t, \qquad Y_0 = x_{\text{init}}$$

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

$$\mathbb{E}[\Phi(X)] = \mathbb{E}\left[\Phi(Y)\exp\left(-\int_0^T \sigma(Y_t, t)^{-1}b_2(Y_t, t)\,\mathrm{d}B_t - \frac{1}{2}\int_0^T \|\sigma(Y_t, t)^{-1}b_2(Y_t, t)\|^2\,\mathrm{d}t\right)\right] \\ = \mathbb{E}\left[\Phi(Y)\exp\left(-\int_0^T \sigma(Y_t, t)^{-1}b_2(Y_t, t)\,\mathrm{d}\tilde{B}_t + \frac{1}{2}\int_0^T \|\sigma(Y_t, t)^{-1}b_2(Y_t, t)\|^2\,\mathrm{d}t\right)\right],$$

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

**Lemma 4.** For an arbitrary  $v \in U$ , let  $\mathbb{P}^v$  and  $\mathbb{P}$  be respectively the laws of the SDEs

$$dX_t^v = (b(X_t^v, t) + \sigma(t)v(X_t^v, t)) dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^v \sim p_0,$$
  
$$dX_t = b(X_t, t) dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

We have that

$$\begin{split} \frac{d\mathbb{P}}{d\mathbb{P}^v}(X^v) &= \exp\left(-\lambda^{-1/2}\int_0^T \langle v(X_t^v,t), \mathrm{d}B_t^v \rangle + \frac{\lambda^{-1}}{2}\int_0^T \|v(X_t^v,t)\|^2 \,\mathrm{d}t\right) \\ &= \exp\left(-\lambda^{-1/2}\int_0^T \langle v(X_t^v,t), \mathrm{d}B_t \rangle - \frac{\lambda^{-1}}{2}\int_0^T \|v(X_t^v,t)\|^2 \,\mathrm{d}t\right), \\ \frac{d\mathbb{P}^v}{d\mathbb{P}^v}(X) &= \exp\left(\lambda^{-1/2}\int_0^T \langle v(X_t,t), \mathrm{d}B_t \rangle - \frac{\lambda^{-1}}{2}\int_0^T \|v(X_t,t)\|^2 \,\mathrm{d}t\right). \end{split}$$

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

$$\frac{d\mathbb{P}}{d\mathbb{P}^{u^*}}(X^{u^*}) = \exp\left(\lambda^{-1}\left(-V(X_0^{u^*}, 0) + \mathcal{W}(X^{u^*}, 0)\right)\right),\\ \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X) = \exp\left(\lambda^{-1}\left(V(X_0, 0) - \mathcal{W}(X, 0)\right)\right),$$

where the functional W is defined in (2.1).

*Proof.* The proof of (4)-(4) follows directly from Cor. 1. To prove (4), we use that by (B),

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

which implies that

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

To prove (4), we use that since  $dX_t^{u^*} = b(X_t^{u^*}, t) dt + \sqrt{\lambda}\sigma(t)dB_t^{u^*}$ , equation (B) holds if we replace X and B by  $X^{u^*}$  and  $B^{u^*}$ , which reads

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

Hence,

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

Lemma 5. The following expression holds:

$$\mathbb{E}_{\mathbb{P}^{u}} \Big[ \log \frac{d\mathbb{P}^{u}}{d\mathbb{P}^{u^{*}}} \Big] = \lambda^{-1} \mathbb{E} \Big[ \int_{0}^{T} \left( \frac{1}{2} \| u(X_{t}^{u}, t) \|^{2} + f(X_{t}^{u}, t) \right) dt + g(X_{T}^{u}) - V(X_{0}^{u}, 0) \Big],$$

*Proof.* To prove (5), we write

$$\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^u) = \log \left(\frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^u)\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)\right) = \log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}}(X^u) + \log \frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)$$
$$= \lambda^{-1} \left(V(X_0^u, 0) - \int_0^T f(X_t^u, t) \, \mathrm{d}t - g(X_T^u)\right)$$
$$- \lambda^{-1/2} \int_0^T \langle u(X_t^u, t), \mathrm{d}B_t \rangle - \frac{\lambda^{-1}}{2} \int_0^T \|u(X_t^u, t)\|^2 \, \mathrm{d}t \; .$$

Since  $\mathbb{E}_{\mathbb{P}^{u}}\left[\log \frac{d\mathbb{P}^{u}}{d\mathbb{P}^{u^{*}}}\right] = -\mathbb{E}_{\mathbb{P}^{u}}\left[\log \frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}}\right]$ , and  $\mathbb{E}_{\mathbb{P}^{u}}\left[\int_{0}^{T} \langle u(X_{t}^{u}, t), \mathrm{d}B_{t} \rangle\right] = 0$ , the result follows.

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

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

When  $p_0$  is concentrated at a single point  $x_{init}$ , the terms  $V(x_{init}, 0)$  are constant and can be removed without modifying the landscape. In other words,  $\tilde{\mathcal{L}}_{CE}$  and  $\mathcal{L}_{CE}$  are equal up to constant terms and constant factors. (ii) When  $p_0$  is a generic probability measure,  $\tilde{\mathcal{L}}_{CE}$  and  $\mathcal{L}_{CE}$  have different landscapes, and  $\mathcal{L}_{CE}(u) = \mathbb{E}_{\mathbb{P}^{u^*}} \left[ \log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u} \exp\left(-\lambda^{-1}V(X_0^{u^*}, 0)\right) \right]$ .  $u^*$  is still the only minimizer of the loss  $\mathcal{L}_{CE}$ , and for some constant K, we have that

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

*Proof.* We begin with the proof of (i), and prove (3) first. Note that by the Girsanov theorem (Theorem 4),

$$\begin{split} \mathbb{E}_{\mathbb{P}^{u^*}} \left[ \log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u} (X^{u^*}) \right] &= -\mathbb{E}_{\mathbb{P}^{u^*}} \left[ \log \frac{d\mathbb{P}^u}{d\mathbb{P}^{u^*}} (X^{u^*}) \right] = -\mathbb{E}_{\mathbb{P}^{u^*}} \left[ \log \frac{d\mathbb{P}^u}{d\mathbb{P}} (X^{u^*}) + \log \frac{d\mathbb{P}}{d\mathbb{P}^{u^*}} (X^{v}) \right] \\ &= -\mathbb{E}_{\mathbb{P}^v} \left[ \left( \log \frac{d\mathbb{P}^u}{d\mathbb{P}} (X^v) + \log \frac{d\mathbb{P}}{d\mathbb{P}^{u^*}} (X^v) \right) \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}} (X^v) \frac{d\mathbb{P}}{d\mathbb{P}^v} (X^v) \right] \end{split}$$

Note that by equations (4) and (4),

$$\begin{split} \log \frac{d\mathbb{P}^{u}}{d\mathbb{P}}(X^{v}) &= \lambda^{-1/2} \int_{0}^{T} \langle u(X_{t}^{v}, t), \mathrm{d}B_{t}^{v} \rangle - \frac{\lambda^{-1}}{2} \int_{0}^{T} \|u(X_{t}^{v}, t)\|^{2} \, \mathrm{d}t, \\ &= \lambda^{-1/2} \int_{0}^{T} \langle u(X_{t}^{v}, t), \mathrm{d}B_{t} \rangle + \lambda^{-1} \int_{0}^{T} \langle u(X_{t}^{v}, t), v(X_{t}^{v}, t) \rangle \, \mathrm{d}t - \frac{\lambda^{-1}}{2} \int_{0}^{T} \|u(X_{t}^{v}, t)\|^{2} \, \mathrm{d}t, \\ &\log \frac{d\mathbb{P}}{d\mathbb{P}^{u^{*}}}(X^{v}) = \lambda^{-1} \big( - V(X_{0}^{v}, 0) + \mathcal{W}(X^{v}, 0) \big). \end{split}$$

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

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

If we plug (B) and (B) into the right-hand side of (B), we obtain

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

which concludes the proof.

To show (3), we use that by Cor. 1,

$$\frac{d\mathbb{P}^{u}}{d\mathbb{P}^{u^{*}}}(X^{u^{*}}) = \exp\left(-\lambda^{-1/2} \int_{0}^{T} \langle u^{*}(X_{t}^{u^{*}}, t) - u(X_{t}^{u^{*}}, t), \mathrm{d}B_{t} \rangle - \frac{\lambda^{-1}}{2} \int_{0}^{T} \|u^{*}(X_{t}^{u^{*}}, t) - u(X_{t}^{u^{*}}, t)\|^{2} \, \mathrm{d}t\right).$$

Hence,

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

Next, we prove (ii). The first instance of  $V(X_0^v, 0)$  in (3) can be removed without modifying the landscape of the loss. Hence, we are left with

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

And this can be expressed as

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

where

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

If we consider g(u; x) as a loss function for u, note that it is equivalent to the loss  $\overline{L}_{CE}(u)$  equation in (B) 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  $\mathcal{L}_{CE}(u) = \mathbb{E}[g(u; X_0^v)]$ .

To prove (3), note that up to a constant term, the only difference between  $\bar{\mathcal{L}}_{CE}(u)$  and  $\mathcal{L}_{CE}(u)$  is the expectation is reweighted importance weight exp $(-\lambda^{-1}V(X_0^v, 0))$ .

Lemma 6. (i) We can rewrite

$$\begin{split} \tilde{\mathcal{L}}_{\operatorname{Var}_{v}}(u) &= \operatorname{Var}\left(\exp\left(\tilde{Y}_{T}^{u,v} - \lambda^{-1}g(X_{T}^{v}) + \lambda^{-1}V(X_{0}^{v},0)\right)\right),\\ \tilde{\mathcal{L}}_{\operatorname{Var}_{v}}^{\log}(u) &= \operatorname{Var}\left(\tilde{Y}_{T}^{u,v} - \lambda^{-1}g(X_{T}^{v}) + \lambda^{-1}V(X_{0}^{v},0)\right). \end{split}$$

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

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

$$\begin{split} \bar{\mathcal{L}}_{\operatorname{Var}_{v}}(u) &= \mathbb{E}[\operatorname{Var}_{\mathbb{P}^{v}}(\frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}}(X^{v})|X_{0}^{v})\exp(-\lambda^{-1}V(X_{0}^{v},0))]\\ &= \mathbb{E}[\operatorname{Var}(\exp(\tilde{Y}_{T}^{u,v}-\lambda^{-1}g(X_{T}^{v}))|X_{0}^{v})]. \end{split}$$

In practice, this is implemented by sampling the *m* trajectories in one batch starting at the same point  $X_0^v$ . (iii) Also,  $\tilde{\mathcal{L}}_{\operatorname{Var}_v}^{\log}$  and  $\mathcal{L}_{\operatorname{Var}_v}^{\log}$  have a different landscape, and the optimum of  $\mathcal{L}_{\operatorname{Var}_v}^{\log}$  may be different from  $u^*$ . In particular,  $\mathcal{L}_{\operatorname{Var}_v}^{\log}(u) = \operatorname{Var}_{\mathbb{P}^v}(\log \frac{d\mathbb{P}^{u^*}}{d\mathbb{P}^u}(X^v)\exp(-\lambda^{-1}V(X_0^v,0)))$ . A loss that does preserve the optimum  $u^*$  is

$$\begin{split} \bar{\mathcal{L}}_{\operatorname{Var}_{v}}^{\log}(u) &= \mathbb{E}[\operatorname{Var}_{\mathbb{P}^{v}}(\log \frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}}(X^{v})|X_{0}^{v})\exp(-\lambda^{-1}V(X_{0}^{v},0))]\\ &= \mathbb{E}[\operatorname{Var}\big(\tilde{Y}_{T}^{u,v} - \lambda^{-1}g(X_{T}^{v})|X_{0}^{v}\big)]. \end{split}$$

*Proof.* Using (4) and (4), we have that

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

Hence,

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

Since  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}(u) = \operatorname{Var}_{\mathbb{P}^{v}}(\frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}})$  and  $\tilde{\mathcal{L}}_{\operatorname{Var}_{v}}^{\log}(u) = \operatorname{Var}_{\mathbb{P}^{v}}(\log \frac{d\mathbb{P}^{u^{*}}}{d\mathbb{P}^{u}})$ , 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 section 3

### C.1 Proof of Theorem 1 and Prop. 2

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

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

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

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

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 7. The following equality holds:

$$\mathbb{E}\left[\frac{1}{T}\int_{0}^{T} \langle u(X_{t},t),\sigma(t)^{\top}\nabla V(X_{t},t)\rangle \,\mathrm{d}t \,\exp\left(-\lambda^{-1}\int_{0}^{T} f(X_{t},t)\,\mathrm{d}t-\lambda^{-1}g(X_{T})\right)\right] \\ = -\lambda \mathbb{E}\left[\frac{1}{T}\int_{0}^{T} \langle u(X_{t},t),\sigma(t)^{\top}\nabla_{x}\mathbb{E}\left[\exp\left(-\lambda^{-1}\int_{t}^{T} f(X_{s},s)\,\mathrm{d}s-\lambda^{-1}g(X_{T})\right)\big|X_{t}=x\right]\right\rangle \mathrm{d}t \\ \times \exp\left(-\lambda^{-1}\int_{0}^{t} f(X_{s},s)\,\mathrm{d}s\right)\right].$$

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

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

Let  $\{\mathcal{F}_t\}_{t\in[0,T]}$  be the filtration generated by the Brownian motion B. Then, equation (1) implies that

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

We proceed as follows:

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

Here, (i) holds by equation (C.1), the law of total expectation and equation (C.1), and (ii) holds by the Markov property of the solution of an SDE.  $\Box$ 

The following proposition, which we prove in subsection C.2, provides an alternative expression for  $\nabla_x \mathbb{E}\left[\exp\left(-\lambda^{-1}\int_t^T f(X_s,s) \,\mathrm{d}s - \lambda^{-1}g(X_T)\right) | X_t = x\right]$ . The technique, which is novel and we denote by *path-wise* reparamaterization trick, is of independent interest and may be applied in other settings, as we discuss in section 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  $M_t(t) = \text{Id}$ . We have that

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

Plugging (1) into the right-hand side of (7), we obtain that

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

If we plug this into the right-hand side of (C.1) and complete the squared norm, we get that

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

where  $\tilde{w}$  is defined in equation (2). We also define  $\Phi(u; X, B)$  as

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

Now, by the Girsanov theorem (Theorem 4), we have that for an arbitrary control  $v \in \mathcal{U}$ ,

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

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:

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

Putting everything together, we obtain that

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

where  $\mathcal{L}(u, M)$  is the loss defined in (5) (note that  $w(t, v, X^v, B, M_t) := \tilde{w}(t, X^v, B^v, M_t)$ ), and

$$K = \mathbb{E}\left[\frac{1}{T}\int_{0}^{T} (\left\|\tilde{w}(t, X, B, M_{t})\right\|^{2} - \left\|u^{*}(X_{t}, t)\right\|^{2}) dt \exp\left(-\lambda^{-1}\mathcal{W}(X, 0)\right)\right]$$

To complete the proof of equation (2), remark that  $\tilde{\mathcal{L}}(u)$  can be rewritten as

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

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

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

Hence, using the Girsanov theorem (Theorem 4) several times, we have that

$$\begin{split} K &= \mathbb{E}\left[\frac{1}{T}\int_{0}^{T}\left\|\tilde{w}(t, X^{u^{*}}, B^{u^{*}}, M_{t})\right\|^{2} - \left\|\mathbb{E}\left[\tilde{w}(t, X^{u^{*}}, B^{u^{*}}, M_{t})|X_{t}^{u^{*}}\right]\right\|^{2} \mathrm{d}t\,\exp(-\lambda^{-1}V(X_{0}^{u^{*}}, 0))\right] \\ &= \mathbb{E}\left[\frac{1}{T}\int_{0}^{T}\left\|\tilde{w}(t, X^{u^{*}}, B^{u^{*}}, M_{t}) - \mathbb{E}\left[\tilde{w}(t, X^{u^{*}}, B^{u^{*}}, M_{t})|X_{t}^{u^{*}}\right]\right\|^{2} \mathrm{d}t\,\exp(-\lambda^{-1}V(X_{0}^{u^{*}}, 0))\right] \\ &= \mathbb{E}\left[\frac{1}{T}\int_{0}^{T}\left\|\tilde{w}(t, X, B, M_{t}) - \frac{\mathbb{E}\left[\tilde{w}(t, X, B, M_{t})\exp(-\lambda^{-1}W(X, 0))|X_{t}\right]}{\mathbb{E}\left[\exp(-\lambda^{-1}W(X, 0))|X_{t}\right]}\right\|^{2} \mathrm{d}t\,\exp(-\lambda^{-1}\mathcal{W}(X, 0))\right] \\ &= \mathbb{E}\left[\frac{1}{T}\int_{0}^{T}\left\|w(t, v, X^{v}, B, M_{t}) - \frac{\mathbb{E}\left[w(t, v, X^{v}, B, M_{t})\alpha(v, X^{v}, B)|X_{t}^{v}\right]}{\mathbb{E}\left[\alpha(v, X^{v}, B)|X_{t}^{v}\right]}\right\|^{2} \mathrm{d}t\,\alpha(v, X^{v}, B)\right], \end{split}$$

which concludes the proof, noticing that K = Var(w; M).

#### 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 (2.1), 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  $\psi$  on  $\omega$  in the proof. Then,

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

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,  $\psi_t$  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  $F_t(X) = \lambda^{-1} \int_t^T f(X_s, s) \, ds + \lambda^{-1}g(X_T)$ . We have that

$$\begin{split} \nabla_z F_t(X+\psi_t(z,\cdot)) \\ &= \nabla_z \left(\lambda^{-1} \int_t^T f(X_s+\psi_t(z,s),s) \,\mathrm{d}s + \lambda^{-1} g(X_T+\psi_t(z,T))\right) \\ \stackrel{(\mathrm{i})}{=} \lambda^{-1} \int_t^T \nabla_z \psi_t(z,s) \nabla f(X_s+\psi_t(z,s),s) \,\mathrm{d}s + \lambda^{-1} \nabla_z \psi_t(z,T) \nabla g(X_T+\psi_t(z,T)) \\ &= \lambda^{-1} \int_t^T M_t(s) \nabla f(X_s+\psi_t(z,s),s) \,\mathrm{d}s + \lambda^{-1} M_t(T) \nabla g(X_T+\psi_t(z,T)), \end{split}$$

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

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

Up to a trivial time change of variable from [t, T] to [0, T], Prop. 1 follows from plugging these choices into equation (4).

**Remark 1.** 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  $M_t(x,t) = \text{Id for all } x \in \mathbb{R}^d$ , and we define the exponential moving average of X as the process  $X^{(v)}$  given by

$$X_t^{(v)} = v \int_0^t e^{-v(t-s)} X_s \, ds,$$

we have that

$$\begin{split} \frac{d}{ds}M_t(X_s^{(\upsilon)},s) &= \langle \nabla M_t(X_s^{(\upsilon)},s), \frac{dX_s^{(\upsilon)}}{ds} \rangle + \partial_s M_t(X_s^{(\upsilon)},s) \\ &= \lambda \langle \nabla_x M_t(X_s^{(\upsilon)},s), X_s - X_s^{(\upsilon)} \rangle + \partial_s M_t(X_s^{(\upsilon)},s), \end{split}$$

and we can write

$$\begin{aligned} \nabla_{x} \mathbb{E} \Big[ \exp \left( -\lambda^{-1} \int_{t}^{T} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} g(X_{T}) \right) \Big| X_{t} &= x \Big] \\ &= \mathbb{E} \Big[ \Big( -\lambda^{-1} \int_{t}^{T} M_{t}(X_{s}^{(\upsilon)}, s) \nabla_{x} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} M_{t}(X_{T}^{(\upsilon)}, T) \nabla g(X_{T}) \\ &+ \lambda^{-1/2} \int_{t}^{T} (M_{t}(X_{s}^{(\upsilon)}, s) \nabla_{x} b(X_{s}, s) - \frac{d}{ds} M_{t}(X_{s}^{(\upsilon)}, s)) (\sigma^{-1})^{\top}(s) \mathrm{d}B_{s} \Big) \\ &\times \exp \Big( -\lambda^{-1} \int_{t}^{T} f(X_{s}, s) \, \mathrm{d}s - \lambda^{-1} g(X_{T}) \Big) \Big| X_{t} &= x \Big]. \end{aligned}$$

Plugging this into the proof of Theorem 1, we would obtain a variant of SOCM (Alg. 2) where the matrix-valued neural network  $M_{\omega}$  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  $M_{\omega}$ ; 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  $M_{\omega}$  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

 $dX_s = b(X_s, s) ds + \sqrt{\lambda}\sigma(s) dB_s, \qquad X_0 \sim p_0,$ 

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:

$$\begin{split} \mathrm{d} X^{(x+z)}_s &= b(X^{(x+z)}_s,s)\,\mathrm{d} s + \sqrt{\lambda}\sigma(s)\,\mathrm{d} B_s, \qquad X^{(x+z)}_0 = x+z, \\ \mathrm{d} X^{(x)}_s &= b(X^{(x)}_s,s)\,\mathrm{d} s + \sqrt{\lambda}\sigma(s)\,\mathrm{d} B_s, \qquad X^{(x)}_0 = x. \end{split}$$

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

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

then  $X^{(x+z)} = \tilde{X}^{(x)} + \psi(z, \cdot)$  is a solution of (C.2). This is because  $X_0^{(x+z)} = \tilde{X}_0^{(x)} + \psi(z, 0) = \tilde{X}_0^{(x)} + z = x + z$ , and

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

Note that we may rewrite (C.2) as

$$dX_{s}^{(x)} = (b(X_{s}^{(x)} + \psi(z, s), s) - \partial_{s}\psi(z, s)) ds + (b(X_{s}^{(x)}, s) - b(X_{s}^{(x)} + \psi(z, s), s) + \partial_{s}\psi(z, s)) ds + \sqrt{\lambda}\sigma(s) dB_{s}, \qquad X_{t}^{(x)} \sim p_{0}.$$

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  $\Phi$ ,

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

We can write

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

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 (C.2), and equality (iv) holds by the definition of  $X_s^{(x)}$ . We conclude the proof by

differentiating the right-hand side of (C.2) with respect to z. Namely,

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

In equality (i) we used (C.2), and that:

• by the Leibniz rule,

$$\begin{aligned} \nabla_z \int_0^T \|\sigma(s)^{-1} (b(X_s + \psi(z, s), s) - b(X_s, s) - \partial_s \psi(z, s))\|^2 \, \mathrm{d}s \Big|_{z=0} \\ &= \int_0^T \nabla_z \|\sigma(s)^{-1} (b(X_s + \psi(z, s), s) - b(X_s, s) - \partial_s \psi(z, s))\|^2 \Big|_{z=0} \, \mathrm{d}s = 0. \end{aligned}$$

• and by the Leibniz rule for stochastic integrals (see Hutton and Nelson (1984)),

$$\begin{aligned} \nabla_z \Big( \int_0^T \sigma(s)^{-1} (b(X_s + \psi(z, s), s) - b(X_s, s) - \partial_s \psi(z, s)) \, \mathrm{d}B_s \Big) \Big|_{z=0} \\ &= \int_0^T (\nabla_z \psi(0, s) \nabla_x b(X_s, s) - \nabla_z \partial_s \psi(0, s)) (\sigma^{-1})^\top(s) \, \mathrm{d}B_s. \end{aligned}$$

г		

### 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  $F(X) = \lambda^{-1} \int_0^T f(X_s, s) \, ds + \lambda^{-1} g(X_T)$ . Consider the Euler-Maruyama discretization of the uncontrolled process X defined in (2.1), 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

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

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  $k \in \{0, \ldots, K-1\}$ ,  $\hat{X}_{k+1} | \hat{X}_k \sim N(\hat{X}_k + \delta b(\hat{X}_k, k\delta), \delta\lambda(\sigma\sigma^{\top})(k\delta))$ . Hence,

$$\begin{split} \mathbb{E}\Big[\exp\left(-\lambda^{-1}\delta\sum_{k=0}^{K-1}f(\hat{X}_{k},s)-\lambda^{-1}g(\hat{X}_{K})\right)\Big|\hat{X}_{0}&=x\Big]\\ &=C^{-1}\iint_{(\mathbb{R}^{d})^{K}}\exp\left(-\lambda^{-1}\delta\sum_{k=0}^{K-1}f(\hat{x}_{k},s)-\lambda^{-1}g(\hat{x}_{K})\right.\\ &\left.-\frac{1}{2\delta\lambda}\sum_{k=1}^{K-1}\|\sigma^{-1}(k\delta)(\hat{x}_{k+1}-\hat{x}_{k}-\delta b(\hat{x}_{k},k\delta))\|^{2}\right.\\ &\left.-\frac{1}{2\delta\lambda}\|\sigma^{-1}(0)(\hat{x}_{1}-x-\delta b(x,0))\|^{2}\right)\mathrm{d}\hat{x}_{1}\cdots\mathrm{d}\hat{x}_{K}, \end{split}$$

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

$$\begin{split} \nabla_{x} \mathbb{E} \Big[ \exp \left( -\lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{X}_{k}, s) - \lambda^{-1} g(\hat{X}_{K}) \right) | \hat{X}_{0} = x \Big] \\ &= \nabla_{z} \mathbb{E} \Big[ \exp \left( -\lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{X}_{k}, s) - \lambda^{-1} g(\hat{X}_{K}) \right) | \hat{X}_{0} = x + z \Big] |_{z=0} \\ &= C^{-1} \nabla_{z} \Big( \iint_{(\mathbb{R}^{d})^{K}} \exp \left( -\lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{x}_{k}, s) - \lambda^{-1} g(\hat{x}_{K}) \right) \\ &\quad - \frac{1}{2\delta\lambda} \sum_{k=1}^{K-1} \| \sigma^{-1} (k\delta) (\hat{x}_{k+1} - \hat{x}_{k} - \delta b(\hat{x}_{k}, k\delta)) \|^{2} \\ &\quad - \frac{1}{2\delta\lambda} \| \sigma^{-1} (0) (\hat{x}_{1} - (x+z) - \delta b(x+z, 0)) \|^{2} \Big) \, d\hat{x}_{1} \cdots d\hat{x}_{K} \Big) |_{z=0} \\ &= C^{-1} \nabla_{z} \Big( \iint_{(\mathbb{R}^{d})^{K}} \exp \left( -\lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{x}_{k} + \psi(z, k\delta), s) - \lambda^{-1} g(\hat{x}_{K} + \psi(z, K\delta)) \right) \\ &\quad - \frac{1}{2\delta\lambda} \sum_{k=1}^{K-1} \| \sigma^{-1} (k\delta) (\hat{x}_{k+1} + \psi(z, (k+1)\delta) - \hat{x}_{k} - \psi(z, k\delta) - \delta b(\hat{x}_{k} + \psi(z, k\delta), k\delta)) \|^{2} \\ &\quad - \frac{1}{2\delta\lambda} \| \sigma^{-1} (0) (\hat{x}_{1} + \psi(z, \delta) - (x + \psi(z, 0)) - \delta b(x + \psi(z, 0), 0)) \|^{2} \Big) \, d\hat{x}_{1} \cdots d\hat{x}_{K} \Big) |_{z=0}, \end{split}$$

In the last equality, we used that for  $k \in \{1, \ldots, K\}$ , 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  $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_K)$ , and letting  $\hat{x}_0 = x$ , we define

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

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

$$\begin{split} G_{\hat{x}}(0) &= \lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{x}_{k}, s) + \lambda^{-1} g(\hat{x}_{K}) + \frac{1}{2\delta\lambda} \sum_{k=0}^{K-1} \|\sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_{k} - \delta b(\hat{x}_{k}, k\delta))\|^{2}.\\ \nabla G_{\hat{x}}(z)|_{z=0} &= \lambda^{-1} \delta \sum_{k=0}^{K-1} \nabla \psi(0, k\delta) \nabla f(\hat{x}_{k}, s) + \lambda^{-1} \nabla \psi(0, K\delta) \nabla g(\hat{x}_{K}) \\ &+ \frac{1}{\delta\lambda} \sum_{k=0}^{K-1} (\nabla_{z} \psi(0, (k+1)\delta) - \nabla_{z} \psi(0, k\delta) - \delta \nabla \psi(0, k\delta) \nabla b(\hat{x}_{k}, k\delta)) \\ &\times ((\sigma^{-1})^{\top} \sigma^{-1})(k\delta)(\hat{x}_{k+1} - \hat{x}_{k} - \delta b(\hat{x}_{k}, k\delta)). \end{split}$$

And we can express the right-hand side of (C.3) in terms of  $G_{\hat{x}}(0)$  and  $\nabla G_{\hat{x}}(z)|_{z=0}$ :

$$\nabla_z \left( C^{-1} \iint_{(\mathbb{R}^d)^K} \exp\left( -G_{\hat{x}}(z) \right) \mathrm{d}y_1 \cdots \mathrm{d}y_K \right) = -C^{-1} \iint_{(\mathbb{R}^d)^K} \nabla G_{\hat{x}}(z)|_{z=0} \exp\left( -G_{\hat{x}}(0) \right) \mathrm{d}y_1 \cdots \mathrm{d}y_K.$$

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

$$\begin{aligned} \hat{x}_{k+1} &= \hat{x}_k + \delta b(\hat{x}_k, k\delta) + \sqrt{\delta} \lambda \sigma(k\delta) \epsilon_k, \qquad \hat{x}_0 = x \\ G_{\hat{x}}(0) &= \lambda^{-1} \delta \sum_{k=0}^{K-1} f(\hat{x}_k, s) + \lambda^{-1} g(\hat{x}_K) + \frac{1}{2} \sum_{k=0}^{K-1} \|\epsilon_k\|^2, \\ \nabla G_{\hat{x}}(z)|_{z=0} &= \lambda^{-1} \delta \sum_{k=0}^{K-1} \nabla \psi(0, k\delta) \nabla f(\hat{x}_k, s) + \lambda^{-1} \nabla \psi(0, K\delta) \nabla g(\hat{x}_K) \\ &+ \sqrt{\delta} \lambda^{-1} \sum_{k=0}^{K-1} (\partial_s \nabla_z \psi(0, k\delta) + O(\delta) - \nabla \psi(0, k\delta) \nabla b(\hat{x}_k, k\delta)) (\sigma^{-1})^\top (k\delta) \epsilon_k. \end{aligned}$$

Then, taking the limit  $K \to \infty$  (i.e.  $\delta \to 0$ ), we recognize (C.3) as Euler-Maruyama discretization of the uncontrolled process X in equation (2.1) conditioned on  $X_0 = x$ , and the last term in (C.3) as the Euler-Maruyama discretization of the stochastic integral  $\lambda^{-1/2} \int_0^T (\partial_s \nabla_z \psi(0,s) - \nabla \psi(0,s) \nabla b(X_s^{(x)},s)) (\sigma^{-1})^\top(s) dB_s$ . Thus,

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

which concludes the derivation.

### C.4 SOCM with the adjoint method for SDEs

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

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

where  $X^v$  is the process controlled by v (i.e.,  $dX_t = (b(X_t, t) + \sigma(t)v(X_t, t)) dt + \sqrt{\lambda}\sigma(X_t, t) dB_t$  and  $X_0 \sim p_0$ ),  $\alpha(v, X^v, B)$  is the importance weight defined in Theorem 1, and  $\alpha(t, X^v)$  is the solution of the ODE

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

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

*Proof.* The proof follows the same structure as that of Theorem 1. Instead of plugging the path-wise reparameterization trick (Prop. 1) in the right-hand side of (3), we make use of (8) to evaluate  $\nabla_x \mathbb{E}[\exp(-\lambda^{-1} \int_0^T f(X_t, t) dt - \lambda^{-1} g(X_T))|X_0 = x]$ .

**Lemma 8** (Adjoint method for SDEs, Li et al. (2020); Kidger et al. (2021)). Let  $X : \Omega \times [0,T] \to \mathbb{R}^d$ be the uncontrolled process defined in (2.1), 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  $a(t) := a(\omega, t)$ ,

$$\frac{da(t)}{dt} = -\nabla_x b(X_t, t) a(t) - \nabla_x f(X_t, t) a(t) - \nabla_x f(X_t, t) a(t) = \nabla g(X_T),$$

Then, we have that

$$\nabla_{x} \mathbb{E}[\int_{0}^{T} f(X_{t}, t) dt + g(X_{T}) | X_{0} = x] = \mathbb{E}[a(0) | X_{0} = x],$$
  
$$\nabla_{x} \mathbb{E}\left[\exp\left(-\lambda^{-1} \int_{0}^{T} f(X_{t}, t) dt - \lambda^{-1} g(X_{T})\right) | X_{0} = x\right]$$
  
$$= -\lambda^{-1} \mathbb{E}[a(0) \exp\left(-\lambda^{-1} \int_{0}^{T} f(X_{t}, t) dt - \lambda^{-1} g(X_{T})\right) | X_{0} = x]$$

*Proof.* We will use an approach based on Lagrangian 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

$$\int_0^T f(X_t(\omega), t) dt + g(X_T(\omega))$$
  
=  $\int_0^T f(X_t(\omega), t) dt + g(X_T(\omega)) - \int_0^T \langle a_t(\omega), (dX_t(\omega) - b(X_t(\omega), t) dt - \sqrt{\lambda}\sigma(t) dB_t) \rangle.$ 

By Lemma 9, we have that

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

Hence,

$$\begin{aligned} \nabla_x \Big( \int_0^T f(X_t(\omega), t) \, dt + g(X_T(\omega)) \Big) \\ &= \nabla_x \Big( \int_0^T f(X_t(\omega), t) \, dt + g(X_T(\omega)) - \langle a_T(\omega), X_T(\omega) \rangle + \langle a_0(\omega), X_0(\omega) \rangle \\ &+ \int_0^T \big( \langle a_t(\omega), b(X_t(\omega), t) \rangle + \langle \frac{da_t}{dt}(\omega), X_t(\omega) \rangle \big) \, dt + \sqrt{\lambda} \int_0^T \langle a_t(\omega), \sigma(t) \, dB_t \rangle \Big) \\ &= \int_0^T \nabla_x X_t(\omega) \nabla_x f(X_t(\omega), t) \, dt + \nabla_x X_T(\omega) \nabla_x g(X_T(\omega)) - \nabla_x X_T(\omega) a_T(\omega) + \nabla_x X_0(\omega) a_0(\omega) \\ &+ \int_0^T \big( \nabla_x X_t(\omega) \nabla_x b(X_t(\omega), t) a_t(\omega) + \nabla_x X_t(\omega) \frac{da_t}{dt}(\omega) \big) \, dt \\ &= \int_0^T \nabla_x X_t(\omega) \big( \nabla_x f(X_t(\omega), t) + \nabla_x b(X_t(\omega), t) a_t(\omega) + \frac{da_t}{dt}(\omega) \big) \, dt \\ &+ \nabla_x X_T(\omega) \big( \nabla_x g(X_T(\omega)) - a_T(\omega) \big) + a_0(\omega). \end{aligned}$$

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

$$da_t(\omega) = \left(-\nabla_x b(X_t(\omega), t)a_t(\omega) - \nabla_x f(X_t(\omega), t)\right) dt - \nabla_x h(X_t(\omega), t) dB_t,$$
  
$$a_T(\omega) = \nabla_x g(X_T(\omega)),$$

then we obtain that

$$\nabla_x \left( \int_0^T f(X_t(\omega), t) \, dt + g(X_T(\omega)) \right) = a_0(\omega),$$

and by the Leibniz rule,

$$\nabla_x \mathbb{E}\left[\int_0^T f(X_t(\omega), t) \, dt + g(X_T(\omega))\right] = \mathbb{E}\left[\nabla_x \left(\int_0^T f(X_t(\omega), t) \, dt + g(X_T(\omega))\right)\right] = \mathbb{E}\left[a_0(\omega)\right].$$

and

$$\begin{aligned} \nabla_x \mathbb{E}\Big[\exp\left(-\lambda^{-1}\int_0^T f(X_t(\omega),t)\,dt - \lambda^{-1}g(X_T(\omega))\right)\Big] \\ &= -\lambda^{-1} \mathbb{E}\Big[\nabla_x\Big(\int_0^T f(X_t(\omega),t)\,dt + g(X_T(\omega))\Big)\exp\left(-\lambda^{-1}\int_0^T f(X_t(\omega),t)\,dt - \lambda^{-1}g(X_T(\omega))\Big)\Big] \\ &= -\lambda^{-1} \mathbb{E}\Big[a_0(\omega)\exp\left(-\lambda^{-1}\int_0^T f(X_t(\omega),t)\,dt - \lambda^{-1}g(X_T(\omega))\right)\Big].\end{aligned}$$

_	_
_	_

Lemma 9 (Stochastic integration by parts, Oksendal (2013)). Let

$$dX_t = a_t dt + b_t dW_t^1,$$
  
$$dY_t = f_t dt + g_t dW_t^2.$$

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,

$$X_t Y_t - X_0 Y_0 = \int_0^t X_s \, dY_s + \int_0^t Y_s \, dX_s + \int_0^t b_s g_s \langle dX_s, dY_s \rangle.$$

### C.5 Proof of Lemma 3

*Proof.* Since the equality (B) 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

$$\mathcal{W}(X^{v},0) = V(X_{0}^{v},0) + \frac{1}{2} \int_{0}^{T} \|u^{*}(X_{s}^{v},s)\|^{2} \,\mathrm{d}s - \sqrt{\lambda} \int_{0}^{T} \langle u^{*}(X_{s}^{v},s), \mathrm{d}B_{s}^{v} \rangle,$$

Thus, we obtain that

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

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

### C.6 Proof of Theorem 2

The proof of (2) shows that minimizing Var(w; M) is equivalent to minimizing

$$\mathbb{E}\left[\frac{1}{T}\int_0^T \left\|w(t, v, X^v, B, M_t)\right\|^2 \mathrm{d}t \,\alpha(v, X^v, B)\right].$$

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  $M_t(s) = I + \int_t^s \dot{M}_t(s') \, ds'$ . By Fubini's theorem, we have that

$$\int_t^T M_t(s) \nabla_x f(X_s^v, s) \, \mathrm{d}s = \int_t^T \left( I + \int_t^s \dot{M}_t(s') \, \mathrm{d}s' \right) \nabla_x f(X_s^v, s) \, \mathrm{d}s$$
$$= \int_t^T \nabla_x f(X_s^v, s) \, \mathrm{d}s + \int_t^T \dot{M}_t(s) \int_s^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s' \, \mathrm{d}s,$$

$$-\int_{t}^{T} (M_{t}(s)\nabla_{x}b(X_{s}^{v},s) - \dot{M}_{t}(s))(\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \,\mathrm{d}s \\ =\int_{t}^{T} \dot{M}_{t}(s)(\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \,\mathrm{d}s - \int_{t}^{T} \dot{M}_{t}(s)\int_{s}^{T} \nabla_{x}b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s')v(X_{s}^{v},s) \,\mathrm{d}s' \,\mathrm{d}s,$$

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

Hence, we can rewrite (C.6) as

-

$$\begin{aligned} \mathcal{G}(\dot{M}) &= \mathbb{E} \Big[ \frac{1}{T} \int_{0}^{T} \left\| \sigma(t)^{\top} \Big( \int_{t}^{T} \nabla_{x} f(X_{s}^{v}, s) \, \mathrm{d}s + \nabla g(X_{T}^{v}) \\ &+ \int_{t}^{T} \dot{M}_{t}(s) \Big( \int_{s}^{T} \nabla_{x} f(X_{s'}^{v}, s') \, \mathrm{d}s' + \nabla g(X_{T}^{v}) + (\sigma^{-1})^{\top}(s) v(X_{s}^{v}, s) \\ &- \int_{s}^{T} \nabla_{x} b(X_{s'}^{v}, s') (\sigma_{s'}^{-1})^{\top}(s') v(X_{s}^{v}, s) \, \mathrm{d}s' - \int_{s}^{T} \nabla_{x} b(X_{s'}^{v}, s') (\sigma_{s'}^{-1})^{\top}(s') \, \mathrm{d}B_{s'} \Big) \, \mathrm{d}s \Big) \Big\|^{2} \, \mathrm{d}t \\ &\times \alpha(v, X^{v}, B) \Big] \end{aligned}$$

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

$$\begin{split} \partial_{\epsilon} \mathcal{V}(\dot{M}+\epsilon P)|_{\epsilon=0} &= \partial_{\epsilon} \mathbb{E}\left[\frac{1}{T} \int_{0}^{T} \left\|\sigma(t)^{\top} \left(\int_{t}^{T} \nabla_{x} f(X_{s}^{v},s) \, \mathrm{d}s + \nabla g(X_{T}^{v}) \right. \\ &+ \int_{t}^{T} (\dot{M}_{t}(s) + \epsilon P_{t}(s)) \left(\int_{s}^{T} \nabla_{x} f(X_{s'}^{v},s') \, \mathrm{d}s' + \nabla g(X_{T}^{v}) + (\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \right. \\ &- \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s')v(X_{s}^{v},s) \, \mathrm{d}s' - \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s') \, \mathrm{d}B_{s'}\right) \, \mathrm{d}s\right) \Big\|^{2} \, \mathrm{d}t \\ &\times \alpha(v, X^{v}, B) \Big]\Big|_{\epsilon=0} \\ &= \mathbb{E} \Big[\frac{2}{T} \int_{0}^{T} \left\langle \sigma(t)\sigma(t)^{\top} \left(\int_{t}^{T} \nabla_{x} f(X_{s}^{v},s) \, \mathrm{d}s + \nabla g(X_{T}^{v}) + (\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \right. \\ &+ \int_{t}^{T} \dot{M}_{t}(s) \left(\int_{s}^{T} \nabla_{x} f(X_{s'}^{v},s') \, \mathrm{d}s' + \nabla g(X_{T}^{v}) + (\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \right. \\ &- \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s')v(X_{s}^{v},s) \, \mathrm{d}s' - \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s') \, \mathrm{d}B_{s'}\right) \, \mathrm{d}s \Big), \\ &\int_{t}^{T} P_{t}(s) \left(\int_{s}^{T} \nabla_{x} f(X_{s'}^{v},s') \, \mathrm{d}s' + \nabla g(X_{T}^{v}) + (\sigma^{-1})^{\top}(s)v(X_{s}^{v},s) \right. \\ &- \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s')v(X_{s}^{v},s) \, \mathrm{d}s' - \int_{s}^{T} \nabla_{x} b(X_{s'}^{v},s')(\sigma_{s'}^{-1})^{\top}(s') \, \mathrm{d}B_{s'}\right) \, \mathrm{d}s \Big\rangle \, \mathrm{d}t \\ &\times \alpha(v, X^{v}, B)\Big]. \end{split}$$

If we define

$$\begin{aligned} \chi(s, X^{v}, B) &:= \int_{s}^{T} \nabla_{x} f(X^{v}_{s'}, s') \, \mathrm{d}s' + \nabla g(X^{v}_{T}) + (\sigma^{-1})^{\top}(s) v(X^{v}_{s}, s) \\ &- \int_{s}^{T} \nabla_{x} b(X^{v}_{s'}, s') (\sigma^{-1}_{s'})^{\top}(s') v(X^{v}_{s}, s) \, \mathrm{d}s' - \int_{s}^{T} \nabla_{x} b(X^{v}_{s'}, s') (\sigma^{-1}_{s'})^{\top}(s') \, \mathrm{d}B_{s'}, \end{aligned}$$

we can rewrite (C.6) as

$$\begin{split} \partial_{\epsilon} \mathcal{V}(\dot{M} + \epsilon P)|_{\epsilon=0} = & \mathbb{E} \Big[ \frac{1}{T} \int_{0}^{T} \left\langle \sigma(t) \sigma(t)^{\top} \left( \int_{t}^{T} \nabla_{x} f(X_{s}^{v}, s) \mathrm{d}s + \nabla g(X_{T}^{v}) + \int_{t}^{T} M_{t}(s) \chi(s, X^{v}, B) \mathrm{d}s \right\rangle, \\ & \int_{t}^{T} P_{t}(s) \chi(s, X^{v}, B) \mathrm{d}s \right\rangle \mathrm{d}s \times \alpha(v, X^{v}, B) \Big] \end{split}$$

Now let us reexpress equation (C.6) as:

$$\begin{split} \mathbb{E} \begin{bmatrix} \frac{1}{T} \int_0^T \left\langle \sigma \sigma^\top(t) \left( \nabla g(X_T^v) + \int_t^T \left( \nabla_x f(X_s^v, s) + \dot{M}_t(s) \chi(s, X^v, B) \right) \mathrm{d}s \right), \\ \int_t^T P_t(s) \chi(s, X^v, B) \mathrm{d}s \right\rangle \mathrm{d}t &\times \alpha(v, X^v, B) \end{bmatrix} \\ \stackrel{(\mathrm{i})}{=} \mathbb{E} \begin{bmatrix} \frac{1}{T} \int_0^T \int_0^s \left\langle P_t(s) \chi(s, X^v, B), \\ \sigma \sigma^\top(t) \left( \nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s') \chi(s', X^v, B)) \mathrm{d}s' \right) \right\rangle \mathrm{d}t \, \mathrm{d}s \times \alpha(v, X^v, B) ] \\ \stackrel{(\mathrm{ii})}{=} \mathbb{E} \begin{bmatrix} \frac{1}{T} \int_0^T \int_0^s \left\langle \sigma \sigma^\top(t) \left( \nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s') \chi(s', X^v, B)) \mathrm{d}s' \right) \chi(X^v, s, B)^\top, \\ P_t(s) \right\rangle_F \mathrm{d}t \, \mathrm{d}s \times \alpha(v, X^v, B) \end{bmatrix} \\ = \int_0^T \int_0^s \left\langle \frac{1}{T} \sigma \sigma^\top(t) \mathbb{E} \left[ \left( \nabla g(X_T^v) + \int_t^T (\nabla_x f(X_{s'}^v, s') + \dot{M}_t(s') \chi(X^v, s', B)) \mathrm{d}s' \right) \chi(X^v, s, B)^\top \alpha(v, X^v, B) \right], \\ P_t(s) \right\rangle_F \mathrm{d}t \, \mathrm{d}s. \end{split}$$

Here, equality (i) holds by Lemma 10 with the choices  $\alpha(t,s) = P_t(s)\chi(X^v,s,B)$ ,  $\gamma(t) = \sigma\sigma^{\top}(t)(\nabla g(X_T^v) + \int_t^T (\nabla_x f(X_s^v,s) + \dot{M}_t(s)\chi(X^v,s,B)) \, ds$ . Equality (ii) follows from the fact that for any matrix A and vectors  $b, c, \langle Ab, c \rangle = c^{\top}Ab = \operatorname{Tr}(c^{\top}Ab) = \operatorname{Tr}(Abc^{\top}) = \langle B, cb^{\top} \rangle_F$ , 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 (C.6) must be zero for any P, which implies that almost everywhere with respect to  $t \in [0, T], s \in [s, T]$ ,

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

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

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

If we define  $N_t(s) = -\mathbb{E}\left[\left(\nabla g(X_T^v) + \int_t^T \nabla_x f(X_{s'}^v, s') \, \mathrm{d}s'\right) \chi(X^v, s, B)^\top \times \alpha(v, X^v, B)\right]$ , 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 10.** 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

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

*Proof.* We have that:

$$\begin{split} \int_0^T \int_t^T \left\langle \alpha(t,s), \gamma(t) \right\rangle \mathrm{d}s \, \mathrm{d}t &\stackrel{(\mathrm{i})}{=} \int_0^T \int_0^{T-t} \left\langle \alpha(t,T-s), \gamma(t) \right\rangle \mathrm{d}s \, \mathrm{d}t \\ &\stackrel{(\mathrm{ii})}{=} \int_0^T \int_0^t \left\langle \alpha(T-t,T-s), \gamma(T-t) \right\rangle \mathrm{d}s \, \mathrm{d}t \stackrel{(\mathrm{iii})}{=} \int_0^T \int_s^T \left\langle \alpha(T-t,T-s), \gamma(T-t) \right\rangle \mathrm{d}t \, \mathrm{d}s \\ &\stackrel{(\mathrm{iv})}{=} \int_0^T \int_{T-s}^T \left\langle \alpha(T-t,s), \gamma(T-t) \right\rangle \mathrm{d}t \, \mathrm{d}s \stackrel{(\mathrm{v})}{=} \int_0^T \int_0^s \left\langle \alpha(t,s), \gamma(t) \right\rangle \mathrm{d}t \, \mathrm{d}s \end{split}$$

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.

### D Control warm-starting

We introduce the *Gaussian warm-start*, a control warm-start strategy that we adapt from Liu et al. (2023), and that we use in our experiments in Figure 3. 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

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

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

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

Then, if  $\Gamma_0 = \sigma(0)$ , the controlled process  $X^u$  defined in equation (2.1) has the same marginals as Y. That is, for all  $t \in [0,T]$ ,  $\text{Law}(Y_t) = \text{Law}(X_t^u)$ .

*Proof.* Following Liu et al. (2023), we have that

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

Now,  $p_t$  satisfies the continuity equation equation

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

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

$$\begin{aligned} \partial_t p_t &= -\nabla \cdot (v(x,t)p_t) + \sum_{i=1}^d \sum_{j=1}^d \partial_i \partial_j (D_{ij}(t)p_t) = -\nabla \cdot (v(x,t)p_t) + \sum_{i=1}^d \partial_i \sum_{j=1}^d (D_{ij}(t)\partial_j p_t) \\ &= -\nabla \cdot (v(x,t)p_t) + \nabla \cdot (D(t)\nabla p_t) = -\nabla \cdot (v(x,t)p_t) + \nabla \cdot (D(t)\nabla \log p_t(x)p_t) \\ &= -\nabla \cdot ((v(x,t)-D(t)\nabla \log p_t(x))p_t). \end{aligned}$$

Hence, we need that

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

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,

$$v(x,t) = \partial_t \mu(t) + \left(\partial_t \Gamma(t)\sqrt{t} + \frac{\Gamma(t)}{2\sqrt{t}}\right) \frac{\Gamma(t)^{-1}}{\sqrt{t}} (x - \mu(t)) - \frac{(\sigma\sigma^{\top})(t)}{2} \frac{\Sigma(t)^{-1}}{t} (x - \mu(t)) \\ = \partial_t \mu(t) + \left(\partial_t \Gamma(t)\right) \Gamma(t)^{-1} (x - \mu(t)) + \frac{1}{2t} (x - \mu(t)) - \frac{(\sigma\sigma^{\top})(t)\Sigma(t)^{-1}}{2t} (x - \mu(t))$$

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 (2.1), we need that

$$v(x,t) = b(x,t) + \sigma(t)u(x,t),$$
  

$$\implies u(x,t) = \sigma(t)^{-1} \left(\partial_t \mu_t + \left( \left( \partial_t \Gamma(t) \right) \Gamma(t)^{-1} + \frac{I - (\sigma \sigma^\top)(t) \Sigma(t)^{-1}}{2t} \right) (x - \mu_t) - b(x,t) \right).$$

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 (6). Given splines  $\mu$  and  $\Gamma$ , we obtain the warm-start control using (6); 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

$$\begin{split} \widehat{\mu}(t) &= \frac{(t-b_{-}\Delta)\mu^{(b_{+})} + (b_{+}\Delta - t)\mu^{(b_{-})}}{\Delta}, \qquad \widehat{\partial_{t}\mu}(t) = \frac{\mu^{(b_{+})} - \mu^{(b_{-})}}{\Delta}, \\ \widehat{\Gamma}(t) &= \frac{(t-b_{-}\Delta)\Gamma^{(b_{+})} + (b_{+}\Delta - t)\Gamma^{(b_{-})}}{\Delta}, \qquad \widehat{\partial_{t}\Gamma}(t) = \frac{\Gamma^{(b_{+})} - \Gamma^{(b_{-})}}{\Delta}, \\ \widehat{u}(x,t) &= \sigma(t)^{-1} \big(\widehat{\partial_{t}\mu}(t) + \big(\widehat{\partial_{t}\Gamma}(t)\widehat{\Gamma}(t)^{-1} + \frac{I - (\sigma\sigma^{\top})(t)(\widehat{\Sigma}\widehat{\Sigma}^{\top})^{-1}(t)}{2t}\big)(x - \widehat{\mu}(t)) - b(x,t)\big). \end{split}$$

Algorithm 3 provides a method to learn the splines  $\mu$ ,  $\Gamma$ . It is a stochastic optimization algorithms in which the spline parameters are updated by sampling  $Y_t$  in (6) at different times, computing the control cost relying on (D), 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  $\Gamma_0 = (\Gamma_0^{(b)})_{b=0}^{\mathcal{B}}$ . 1 for  $n=0:(N-1)\}$  do Sample *m* i.i.d. variables  $(Z_i)_{i=1}^n \sim N(0, I)$  and *m* times  $(t_i)_{i=1}^n \sim \text{Unif}([0, T])$ . 2 3 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 (D), (D) using  $\mu_n$ ,  $\Gamma_n$ 4 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 (D); 5 end 6 Compute  $\hat{\mathcal{L}}_{\text{RGSOC}}(\mu_n, \Gamma_n) = \frac{1}{m} \sum_{i=1}^m \left( \frac{T}{K} \sum_{j=0}^{K-1} \left( \frac{1}{2} \| \hat{u}(Y_{ij}, t_j) \|^2 + f(Y_{ij}, t_j) \right) + g(Y_{iK}) \right)$ 7 Compute the gradient of  $\hat{\mathcal{L}}_{RGSOC}(\mu_n, \Gamma_n)$  with respect to the spline parameters  $(\mu_n, \Gamma_n)$ . 8 Obtain  $\mu_{n+1}$ ,  $\Gamma_{n+1}$  with via an Adam update on  $\mu_n$ ,  $\Gamma_n$  resp. (or another stochastic algorithm) 9 10 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  $u_\theta = \hat{u}_N + \tilde{u}_\theta$ .

### E Experimental details and additional plots

#### E.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.

For all losses and all settings, we train the control using Adam with learning rate  $1 \times 10^{-4}$ . For SOCM, we train the reparametrization matrices using Adam with learning rate  $1 \times 10^{-2}$ . 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  $3 \times 10^{-4}$  for N = 60000 iterations.

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

$$b(x,t) = Ax, \quad f(x,t) = x^{\top}Px, \quad g(x) = x^{\top}Qx, \quad \sigma(t) = \sigma_0.$$

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 (Van Handel, 2007, 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

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

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,  $x_{init} = 0.5N(0, I)$ . 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,  $x_{init} = 0.5N(0, I)$ . We use the Gaussian warm-start (Appendix D). 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 (Nüsken and Richter, 2021, Sec. A.4):

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

We use exactly the same functions as Nüsken and Richter (2021): we sample  $(\xi_{ij})_{1 \le i,j \le d}$  once at the beginning of the simulation, and set:

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

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

DOUBLE WELL We also use exactly the same functions as Nüsken and Richter (2021), which are the following:

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

where d = 10, and  $\kappa_i = 5$ ,  $\nu_i = 3$  for  $i \in \{1, 2, 3\}$  and  $\kappa_i = 1$ ,  $\nu_i = 1$  for  $i \in \{4, \ldots, 10\}$ . We set T = 1,  $\lambda = 1$  and  $x_{\text{init}} = 0$ . We take K = 200 time discretization steps, and we use random seed 0.

### E.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.

### E.3 Additional plots

Figure 4 shows the control objective (2.1) for the four settings. The error bars for the control objective plots show the confidence intervals for  $\pm$  one standard deviation. 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:  $\sqrt{\operatorname{Var}[\alpha(u, X^u, B)]}/\mathbb{E}[\alpha(u, X^u, B)]$ . By Lemma 3, when  $X_0^u = x_{\text{init}}$  for an arbitrary  $x_{\text{init}}$  (which is the case for all our experiments), this quantity is zero for the optimal control  $u^*$ . Hence, the normalized standard deviation of  $\alpha$  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 2. Understanding this phenomenon is outside of the scope of this paper.

Figure 7 shows that the instabilities of the adjoint method are inherent to the loss, because they also appear at small learning rates:  $3 \times 10^{-5}$  is smaller than the learning rates typically used for Adam, which hover from  $1 \times 10^{-4}$  to  $1 \times 10^{-3}$ .

Figure 8 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 Appendix D. Figure 3 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 9 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 2 does not present them.



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



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



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 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.



Figure 8 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 9 Plots of the training loss for SOCM and its two ablations: SOCM with constant  $M_t = I$ , and SOCM-Adjoint.