Stochastic Optimal Control Matching

Carles Domingo-Enrich
(joint work with Jiequn Han, Brandon Amos, Joan Bruna, Ricky T.Q. Chen)

Microsoft Research New England (work done while at New York University & Meta AI)

September 5, 2024

Content of the talk

Introduction

- Stochastic optimal control: definition
- Examples: robotics, sampling unnormalized densities, importance sampling for diffusions
- Existing approaches: the adjoint method

[1] **Domingo-Enrich, C.**, Han, J., Amos, B., Bruna, J., Chen, R.T.Q. *Stochastic optimal control matching*, arXiv preprint, 2023.

Content of the talk

Introduction

- Stochastic optimal control: definition
- Examples: robotics, sampling unnormalized densities, importance sampling for diffusions
- Existing approaches: the adjoint method

Stochastic Optimal Control Matching

- Comparing stochastic optimal control with normalizing flows
- Our algorithm: SOCM [1]
- Main features of our algorithm
- Experiments

[1] **Domingo-Enrich, C.**, Han, J., Amos, B., Bruna, J., Chen, R.T.Q. *Stochastic optimal control matching*, arXiv preprint, 2023.

Content of the talk

Introduction

- Stochastic optimal control: definition
- Examples: robotics, sampling unnormalized densities, importance sampling for diffusions
- Existing approaches: the adjoint method

Stochastic Optimal Control Matching

- Comparing stochastic optimal control with normalizing flows
- Our algorithm: SOCM [1]
- Main features of our algorithm
- Experiments

Key ideas

- Derivation of the SOCM loss
- The path-wise reparameterization trick
- Conclusions and future directions

[1] **Domingo-Enrich, C.**, Han, J., Amos, B., Bruna, J., Chen, R.T.Q. *Stochastic optimal control matching*, arXiv preprint, 2023.

Outline

Theme of the talk

Good functional dependence + low gradient variance $\rightarrow$ good empirical performance.

Theme of the talk

Good functional dependence + low gradient variance $\rightarrow$ good empirical performance.

- Continuous normalizing flows were originally trained to maximize the likelihood of training samples using the adjoint method (**highly non-convex functional landscape**).

Theme of the talk

Good functional dependence + low gradient variance → good empirical performance.

- Continuous normalizing flows were originally trained to maximize the likelihood of training samples using the adjoint method (**highly non-convex functional landscape**).
- Diffusion models: least-squares loss to learn score function (**convex functional landscape**).

Theme of the talk

Good functional dependence + low gradient variance → good empirical performance.

- Continuous normalizing flows were originally trained to maximize the likelihood of training samples using the adjoint method (**highly non-convex functional landscape**).
- Diffusion models: least-squares loss to learn score function (**convex functional landscape**).
- Currently, stochastic optimal control is solved relying on the adjoint method (**highly non-convex functional landscape**).

Theme of the talk

Good functional dependence + low gradient variance $\rightarrow$ good empirical performance.

- Continuous normalizing flows were originally trained to maximize the likelihood of training samples using the adjoint method (**highly non-convex functional landscape**).
- Diffusion models: least-squares loss to learn score function (**convex functional landscape**).
- Currently, stochastic optimal control is solved relying on the adjoint method (**highly non-convex functional landscape**).
- Our work is about developing a **least-squares loss** for stochastic optimal control.

Stochastic optimal control

- **Control**: we want to 'control' a system (e.g. driving a car or managing investments) to achieve a desired outcome or behavior.

Stochastic optimal control

- **Control**: we want to 'control' a system (e.g. driving a car or managing investments) to achieve a desired outcome or behavior.
- **Optimal** Control: When driving from point A to B, the 'optimal' path would be the one that minimizes time and fuel.

Stochastic optimal control

- **Control**: we want to 'control' a system (e.g. driving a car or managing investments) to achieve a desired outcome or behavior.
- **Optimal** Control: When driving from point A to B, the 'optimal' path would be the one that minimizes time and fuel.
- **Stochastic** Optimal Control: The systems or processes are random and unpredictable; we need to be robust to noise.

Example I: Robotics

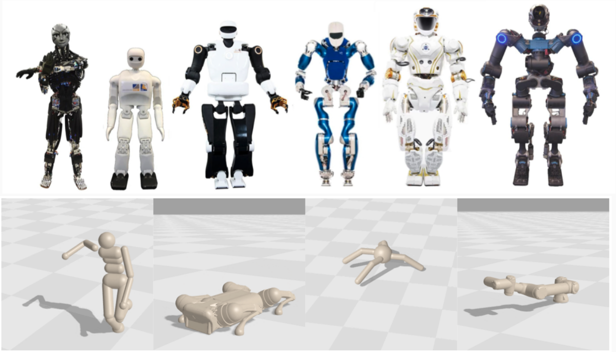


Figure 1: Sources: [FB21; Fre+21]

- Goal: move the robot from an initial position to a final one to accomplish a task

Example I: Robotics

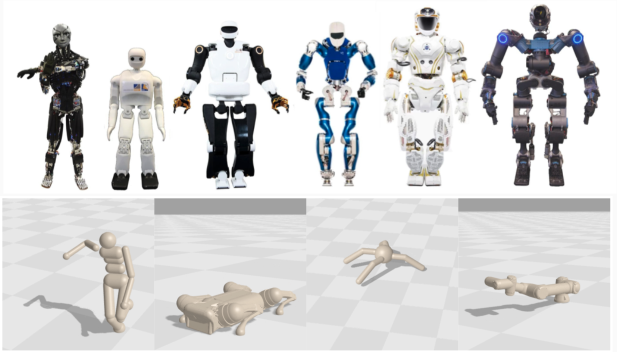


Figure 1: Sources: [FB21; Fre+21]

- Goal: move the robot from an initial position to a final one to accomplish a task
- Controls: torque applied by joints, force applied by linear actuators

Example I: Robotics

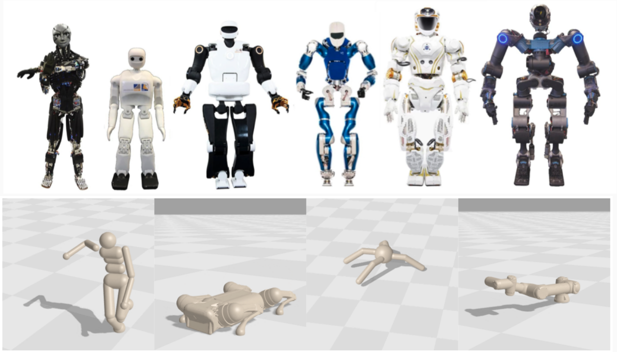


Figure 1: Sources: [FB21; Fre+21]

- Goal: move the robot from an initial position to a final one to accomplish a task
- Controls: torque applied by joints, force applied by linear actuators
- Optimality: accomplish the task using minimal energy

Example I: Robotics

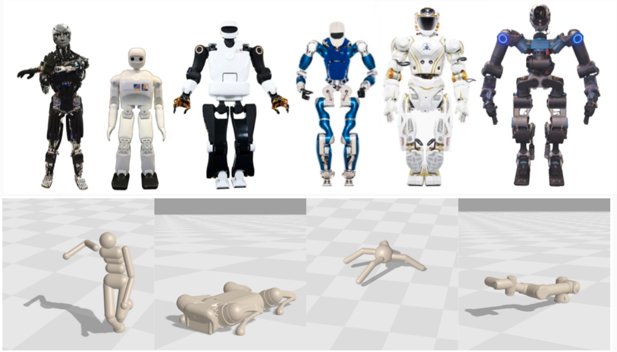


Figure 1: Sources: [FB21; Fre+21]

- Goal: move the robot from an initial position to a final one to accomplish a task
- Controls: torque applied by joints, force applied by linear actuators
- Optimality: accomplish the task using minimal energy
- Stochasticity: sensor noise, unexpected user behavior, variable conditions

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*

  Robot Example (RE): angles and angular velocities of each joint of the robot

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*
  Robot Example (RE): angles and angular velocities of each joint of the robot

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*
  Robot Example (RE): angles and angular velocities of each joint of the robot

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*
  Robot Example (RE): angles and angular velocities of each joint of the robot

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*

  Robot Example (RE): angles and angular velocities of each joint of the robot
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,

  RE: encodes Newtonian mechanics

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*

  Robot Example (RE): angles and angular velocities of each joint of the robot
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,

  RE: encodes Newtonian mechanics
- $\lambda \in \mathbb{R}$ is the *noise variance* and $\sigma : [0, T] \to \mathbb{R}^{d \times d}$ is the *covariance matrix*.

  RE: models stochastic behavior

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*

  Robot Example (RE): angles and angular velocities of each joint of the robot
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,

  RE: encodes Newtonian mechanics
- $\lambda \in \mathbb{R}$ is the *noise variance* and $\sigma : [0, T] \to \mathbb{R}^{d\times d}$ is the *covariance matrix*.

  RE: models stochastic behavior

Controlled process

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

where:

Uncontrolled process vs. controlled process

Uncontrolled process

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

where:

- $X : [0, T] \to \mathbb{R}^d$ is the (random) *uncontrolled process*

  Robot Example (RE): angles and angular velocities of each joint of the robot
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,

  RE: encodes Newtonian mechanics
- $\lambda \in \mathbb{R}$ is the *noise variance* and $\sigma : [0, T] \to \mathbb{R}^{d \times d}$ is the *covariance matrix*.

  RE: models stochastic behavior

Controlled process

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

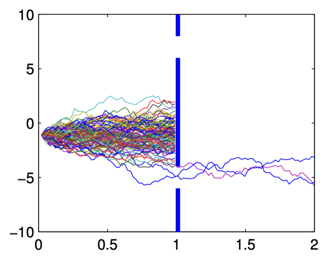
where:

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*

  Robot Example (RE): torque (force) applied to each joint of the robot

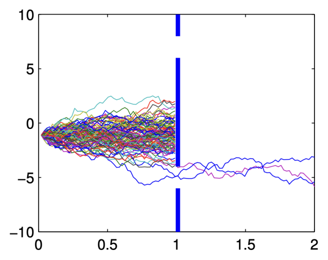
Uncontrolled process vs. controlled process

Uncontrolled process: $dX_t = b(X_t, t)\, dt + \sqrt{\lambda}\sigma(t) dB_t$

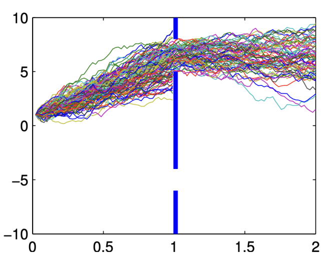


Uncontrolled process vs. controlled process

Uncontrolled process: $dX_t = b(X_t, t)\, dt + \sqrt{\lambda}\sigma(t)dB_t$



Controlled process: $dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t$



Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big],$$

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

where

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*
  Robot Example (RE): torque (force) applied to each joint of the robot

Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big],$$

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

where

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*

  Robot Example (RE): torque (force) applied to each joint of the robot
- $X^u : [0, T] \to \mathbb{R}^d$ is the (random) *controlled process*

  RE: angles and angular velocities of each joint

Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\Big)\,dt + g(X_T^u)\Big],$$

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

where

- $u : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is the *control*
  Robot Example (RE): torque (force) applied to each joint of the robot
- $X^u : [0,T] \to \mathbb{R}^d$ is the (random) *controlled process*
  RE: angles and angular velocities of each joint
- $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ is the *state cost*,
  RE: small for physically possible configurations, very large for impossible ones

Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big],$$

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

where

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*
  Robot Example (RE): torque (force) applied to each joint of the robot
- $X^u : [0, T] \to \mathbb{R}^d$ is the (random) *controlled process*
  RE: angles and angular velocities of each joint
- $f : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *state cost*,
  RE: small for physically possible configurations, very large for impossible ones
- $g : \mathbb{R}^d \to \mathbb{R}$ is the *terminal cost*,
  RE: small for desired final configuration, very large otherwise

Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t) \right) dt + g(X_T^u) \right],$$

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

where

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*
  Robot Example (RE): torque (force) applied to each joint of the robot
- $X^u : [0, T] \to \mathbb{R}^d$ is the (random) *controlled process*
  RE: angles and angular velocities of each joint
- $f : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *state cost*,
  RE: small for physically possible configurations, very large for impossible ones
- $g : \mathbb{R}^d \to \mathbb{R}$ is the *terminal cost*,
  RE: small for desired final configuration, very large otherwise
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,
  RE: encodes Newtonian mechanics

Stochastic optimal control: the problem

Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big],$$

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

where

- $u : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is the *control*
  Robot Example (RE): torque (force) applied to each joint of the robot
- $X^u : [0, T] \to \mathbb{R}^d$ is the (random) *controlled process*
  RE: angles and angular velocities of each joint
- $f : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *state cost*,
  RE: small for physically possible configurations, very large for impossible ones
- $g : \mathbb{R}^d \to \mathbb{R}$ is the *terminal cost*,
  RE: small for desired final configuration, very large otherwise
- $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ is the *base drift*,
  RE: encodes Newtonian mechanics
- $\lambda \in \mathbb{R}$ is the *noise variance* and $\sigma : [0, T] \to \mathbb{R}^{d\times d}$ is the *covariance matrix*.

Example II: Sampling from unnormalized densities

Reminder: Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t) \Big)\, dt + g(X_T^u) \Big],$$

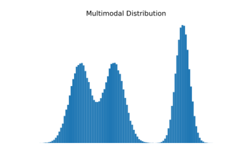
$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t) dB_t, \qquad X_0^u \sim p_0.$$

Example II: Sampling from unnormalized densities

Reminder: Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\Big)\,dt + g(X_T^u)\Big],$$

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



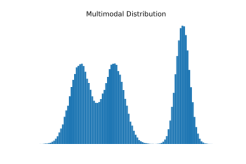
- Challenge: Given a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} e^{-g(x)}\,dx < +\infty$, generate samples from the distribution $\pi(x) \propto e^{-g(x)}$.

Example II: Sampling from unnormalized densities

Reminder: Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\right) dt + g(X_T^u)\right],$$

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



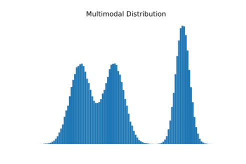
- Challenge: Given a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} e^{-g(x)}\,dx < +\infty$, generate samples from the distribution $\pi(x) \propto e^{-g(x)}$.
- Applications in Bayesian statistics (sampling from posterior) and computational physics (free energy computations).

Example II: Sampling from unnormalized densities

Reminder: Stochastic Optimal Control problem

$$\min_{u} \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t) \right) dt + g(X_T^u) \right],$$

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



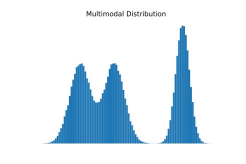
- Challenge: Given a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} e^{-g(x)}\, dx < +\infty$, generate samples from the distribution $\pi(x) \propto e^{-g(x)}$.
- Applications in Bayesian statistics (sampling from posterior) and computational physics (free energy computations).
- Common approach: MCMC algorithms. Issue: They struggle with multimodality.

Example II: Sampling from unnormalized densities

Stochastic Optimal Control approach to sampling [BRU23]

$$\min_{u} \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_t^u, t)\|^2 - (\nabla \cdot b)(X_t^u, t)\right) dt + g(X_T^u)\right],$$

subject to $dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sigma(t)dB_t, \qquad X_0^u \sim N(0, \mathrm{I}),$



- Challenge: Given a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} e^{-g(x)}\, dx < +\infty$, generate samples from the distribution $\pi(x) \propto e^{-g(x)}$.
- Applications in Bayesian statistics (sampling from posterior) and computational physics (free energy computations).
- Common approach: MCMC algorithms. Issue: They struggle with multimodality.
- Stochastic optimal control approach:
  - Let $b(x, t) = x$ be an arbitrary base drift and $T \gg 1$.
  - Set g as the terminal cost.
  - Set $-\nabla \cdot b$ as the state cost.

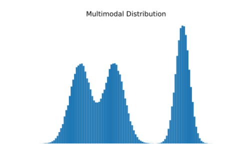
Then, $X_T^u \sim \pi \propto e^{-g(x)}$.

Example II: Sampling from unnormalized densities

Stochastic Optimal Control approach to sampling [BRU23]

$$\min_{u} \mathbb{E}\left[\int_0^T \left(\frac{1}{2}\|u(X_t^u, t)\|^2 - (\nabla \cdot b)(X_t^u, t) \right) dt + g(X_T^u) \right],$$

subject to $dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sigma(t) dB_t, \qquad X_0^u \sim N(0, \mathrm{I}),$

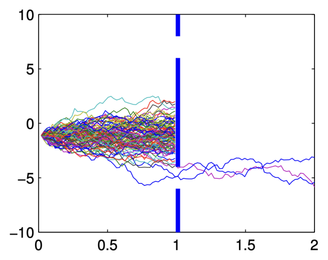


- Challenge: Given a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} e^{-g(x)}\, dx < +\infty$, generate samples from the distribution $\pi(x) \propto e^{-g(x)}$.
- Applications in Bayesian statistics (sampling from posterior) and computational physics (free energy computations).
- Common approach: MCMC algorithms. Issue: They struggle with multimodality.
- Stochastic optimal control approach:
  - Let $b(x, t) = x$ be an arbitrary base drift and $T \gg 1$.
  - Set g as the terminal cost.
  - Set $-\nabla \cdot b$ as the state cost.

  Then, $X_T^u \sim \pi \propto e^{-g(x)}$.

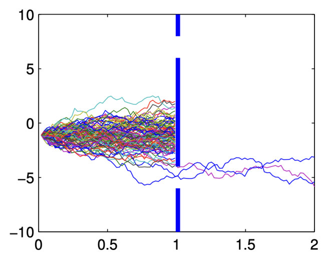
Example III: Importance sampling for diffusions [Zha+14]

We want to estimate the probability that the process X satisfying $dX_t = b(X_t, t)\,dt + \sigma(t)\,dB_t$, $X_0 = x_0$ will go through the top hole, i.e. $P(X_T \in \mathcal{O})$.



Example III: Importance sampling for diffusions [Zha+14]

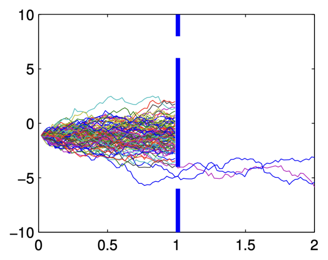
We want to estimate the probability that the process X satisfying $dX_t = b(X_t, t)\,dt + \sigma(t)\,dB_t$, $X_0 = x_0$ will go through the top hole, i.e. $P(X_T \in \mathcal{O})$.



Very unlikely event! Monte Carlo estimation is high-variance.

Example III: Importance sampling for diffusions [Zha+14]

We want to estimate the probability that the process X satisfying $dX_t = b(X_t, t)\,dt + \sigma(t)\,dB_t$, $X_0 = x_0$ will go through the top hole, i.e. $P(X_T \in \mathcal{O})$.



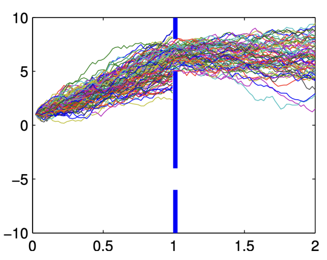
Very unlikely event! Monte Carlo estimation is high-variance. In general, we want to estimate:

$$\mathbb{E}\Big[\exp\Big(-\int_0^T f(X_t, t)\,dt - g(X_T)\Big)\Big]. \tag{1}$$

We recover $P(X_T \in \mathcal{O})$ by setting $f(x,t) = 0$, $g(x) = -\log \mathbb{1}_{\mathcal{O}}(x)$.

Example III: Importance sampling for diffusions [Zha+14]

We want to estimate the probability that the process X satisfying $dX_t = b(X_t, t)\,dt + \sigma(t)\,dB_t$, $X_0 = x_0$ will go through the top hole, i.e. $P(X_T \in \mathcal{O})$.



Very unlikely event! Monte Carlo estimation is high-variance. In general, we want to estimate:

$$\mathbb{E}\left[\exp\left(-\int_0^T f(X_t, t)\,dt - g(X_T)\right)\right]. \tag{1}$$

We recover $P(X_T \in \mathcal{O})$ by setting $f(x, t) = 0$, $g(x) = -\log \mathbb{1}_{\mathcal{O}}(x)$. We need to perform importance sampling using a process that goes through the top hole often!

Example III: Importance sampling for diffusions [Zha+14]

- Let

$$F(X) = \exp\left(- \int_0^T f(X_t, t)\, dt - g(X_{\tau \wedge T}) \right).$$

Example III: Importance sampling for diffusions [Zha+14]

- Let
$$F(X) = \exp\left(- \int_0^T f(X_t, t)\, dt - g(X_{\tau \wedge T}) \right).$$
- Importance sampling: Estimate $\mathbb{E}[F(X)]$ using a Monte Carlo estimate of $\mathbb{E}[F(X^u) \frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$, where
  - u is arbitrary,
  - X^u is a solution of $dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sigma(t)\, dB_t,\ X_0^u = x_0$,
  - $\mathbb{P}$, $\mathbb{P}^u$ are the laws of X and X^u,
  - $\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)$ is computed using the Girsanov theorem.

Example III: Importance sampling for diffusions [Zha+14]

- Let
$$F(X) = \exp\Big(- \int_0^T f(X_t, t)\, dt - g(X_{\tau \wedge T}) \Big).$$
- Importance sampling: Estimate $\mathbb{E}[F(X)]$ using a Monte Carlo estimate of $\mathbb{E}[F(X^u) \frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$, where
  - u is arbitrary,
  - X^u is a solution of $dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sigma(t)\, dB_t, \; X_0^u = x_0$,
  - $\mathbb{P}$, $\mathbb{P}^u$ are the laws of X and X^u,
  - $\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)$ is computed using the Girsanov theorem.
- The control u that minimizes the variance $\mathrm{Var}[F(X^u) \frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$ achieves zero variance, and is the solution of:

Example III: Importance sampling for diffusions [Zha+14]

- Let
$$F(X) = \exp\left(- \int_0^T f(X_t, t)\, dt - g(X_{\tau \wedge T}) \right).$$
- Importance sampling: Estimate $\mathbb{E}[F(X)]$ using a Monte Carlo estimate of $\mathbb{E}[F(X^u)\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$, where
  - u is arbitrary,
  - X^u is a solution of $dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sigma(t)\, dB_t,\ X_0^u = x_0$,
  - $\mathbb{P}$, $\mathbb{P}^u$ are the laws of X and X^u,
  - $\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)$ is computed using the Girsanov theorem.
- The control u that minimizes the variance $\mathrm{Var}[F(X^u)\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$ achieves zero variance, and is the solution of:

Stochastic Optimal Control problem

$$\min_u \mathbb{E}\left[\int_0^T \left(\frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \right) dt + g(X_T^u) \right],$$
$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sqrt{\lambda} \sigma(t) dB_t, \qquad X_0^u = x_0.$$

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \Big)\, dt + g(X_T^u) \Big], \tag{2}$$

$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t))\, dt + \sqrt{\lambda} \sigma(t) dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \|u(X_t^u, t)\|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big], \tag{2}$$

$$\text{subject to } dX_t^u = \left(b(X_t^u, t) + \sigma(t) u(X_t^u, t)\right) dt + \sqrt{\lambda} \sigma(t) dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].
- Dimension d higher: the state-of-the-art approach, which is also classical [Pon62; Onk+23; NR23], is the *adjoint method*:

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big], \tag{2}$$

$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].
- Dimension d higher: the state-of-the-art approach, which is also classical [Pon62; Onk+23; NR23], is the *adjoint method*:
  - Parameterize the control with a neural network $u \equiv u_\theta$

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big], \tag{2}$$

$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].
- Dimension d higher: the state-of-the-art approach, which is also classical [Pon62; Onk+23; NR23], is the *adjoint method*:
  - Parameterize the control with a neural network $u \equiv u_\theta$
  - Simulate a batch of trajectories of the SDE (3) to approximate the control objective $\mathcal{L}(u)$

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big], \tag{2}$$

$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t)) \, dt + \sqrt{\lambda} \sigma(t) dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].
- Dimension d higher: the state-of-the-art approach, which is also classical [Pon62; Onk+23; NR23], is the *adjoint method*:
  - Parameterize the control with a neural network $u \equiv u_\theta$
  - Simulate a batch of trajectories of the SDE (3) to approximate the control objective $\mathcal{L}(u)$
  - Compute the gradient of the approximate control objective w.r.t. θ

Existing approaches for Stochastic Optimal Control

Reminder: Stochastic Optimal Control problem

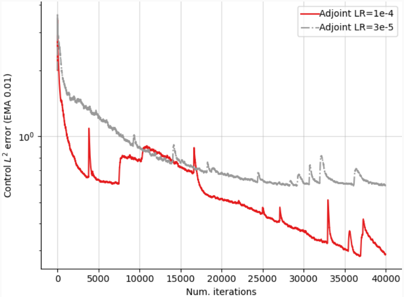
$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \| u(X_t^u, t) \|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big], \tag{2}$$

$$\text{subject to } dX_t^u = (b(X_t^u, t) + \sigma(t) u(X_t^u, t)) \, dt + \sqrt{\lambda} \sigma(t) dB_t, \qquad X_0^u \sim p_0. \tag{3}$$

- Dimension d small ($d \leq 3$): solve the *Hamilton-Jacobi-Bellman (HJB)* partial differential equation using *dynamic programming* [Bel57].
- Dimension d higher: the state-of-the-art approach, which is also classical [Pon62; Onk+23; NR23], is the *adjoint method*:
  - Parameterize the control with a neural network $u \equiv u_\theta$
  - Simulate a batch of trajectories of the SDE (3) to approximate the control objective $\mathcal{L}(u)$
  - Compute the gradient of the approximate control objective w.r.t. θ
  - Update θ using a stochastic optimization algorithm (e.g. Adam)

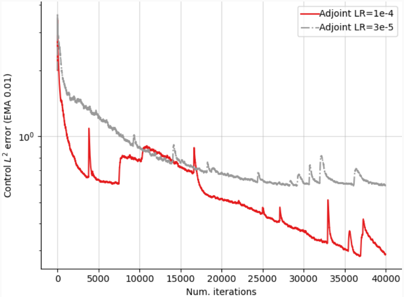
The adjoint method has unstable training dynamics

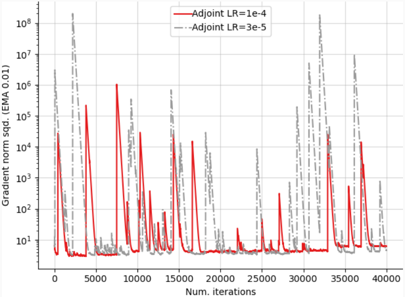
L^2 error for control u: $\mathbb{E}_{t,X^{u^*}} \|u(X_t^{u^*}, t) - u^*(X_t^{u^*}, t)\|^2$, where u^* is the optimal control



The adjoint method has unstable training dynamics

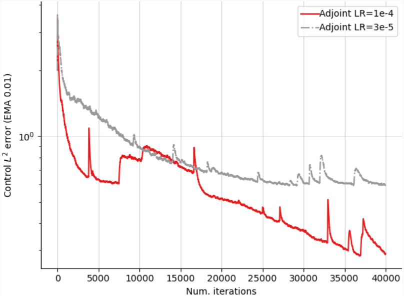
L^2 error for control u: $\mathbb{E}_{t,X^{u^*}} \|u(X_t^{u^*}, t) - u^*(X_t^{u^*}, t)\|^2$, where u^* is the optimal control

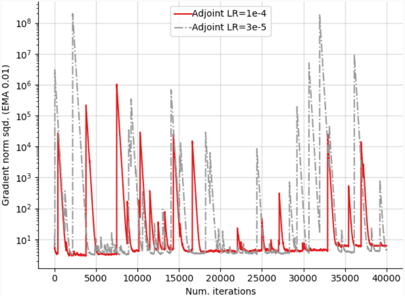




The adjoint method has unstable training dynamics

L^2 error for control u: $\mathbb{E}_{t,X^{u^*}} \|u(X_t^{u^*}, t) - u^*(X_t^{u^*}, t)\|^2$, where u^* is the optimal control



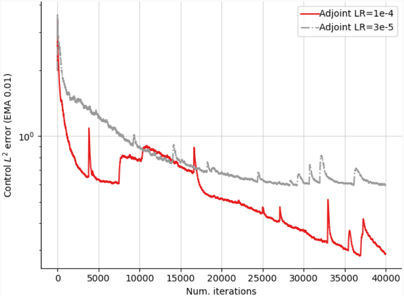


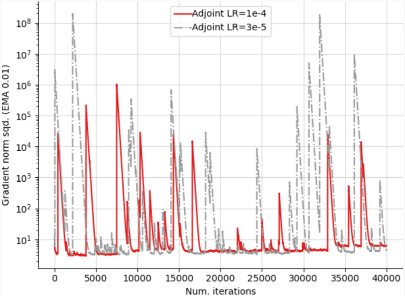
Why? Because the adjoint loss is **highly non-convex** w.r.t. the control u!

$$\min_u \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\,dt + g(X_T^u)\Big], \quad \text{s.t.} \quad \begin{cases} dX_t^u = (b(X_t^u, t) + u(X_t^u, t))dt + \sqrt{\lambda}dB_t, \\ X_0^u \sim p_0. \end{cases}$$

The adjoint method has unstable training dynamics

L^2 error for control u: $\mathbb{E}_{t,X^{u^*}} \|u(X_t^{u^*},t) - u^*(X_t^{u^*},t)\|^2$, where u^* is the optimal control





Why? Because the adjoint loss is **highly non-convex** w.r.t. the control u!

$$\min_u \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u,t)\|^2 + f(X_t^u,t)\Big)\,dt + g(X_T^u)\Big], \quad \text{s.t.} \begin{cases} dX_t^u = (b(X_t^u,t) + u(X_t^u,t))dt + \sqrt{\lambda}dB_t, \\ X_0^u \sim p_0. \end{cases}$$

Can we design algorithms with stable training?

Outline

Introduction

Stochastic Optimal Control Matching

Key ideas

Detour: Maximum Likelihood Normalizing Flows vs. Diffusion Models

We've seen a similar story in generative modeling...

- *Continuous Normalizing Flows* (CNFs) [Che+18]: originally trained to maximize the log-likelihood of the generated samples, using the *adjoint method*.

$$\min_{v} \mathbb{E}\left[-\log p_0(X_0^v) + \int_0^1 \nabla \cdot v_t(X_t^v)\, dt \right], \text{ s.t. } \begin{cases} dX_t^v = v(X_t^v, t) dt, \\ X_0^v \sim p_0. \end{cases}$$

Detour: Maximum Likelihood Normalizing Flows vs. Diffusion Models

We've seen a similar story in generative modeling...

- *Continuous Normalizing Flows* (CNFs) [Che+18]: originally trained to maximize the log-likelihood of the generated samples, using the *adjoint method*.

$$\min_{v} \mathbb{E}\left[-\log p_0(X_0^v) + \int_0^1 \nabla \cdot v_t(X_t^v)\, dt \right], \text{ s.t. } \begin{cases} dX_t^v = v(X_t^v, t)dt, \\ X_0^v \sim p_0. \end{cases}$$

- This loss is **highly non-convex** w.r.t v!

Detour: Maximum Likelihood Normalizing Flows vs. Diffusion Models

We've seen a similar story in generative modeling...

- *Continuous Normalizing Flows* (CNFs) [Che+18]: originally trained to maximize the log-likelihood of the generated samples, using the *adjoint method*.

$$\min_{v} \mathbb{E}\left[-\log p_0(X_0^v) + \int_0^1 \nabla \cdot v_t(X_t^v)\, dt \right], \text{ s.t. } \begin{cases} dX_t^v = v(X_t^v, t)dt, \\ X_0^v \sim p_0. \end{cases}$$

- This loss is **highly non-convex** w.r.t v!
- Diffusion models superseded maximum likelihood CNFs, and require solving least squares problems:

$$\text{DDPM [HJA20]:} \qquad \min_v \mathbb{E}_{t, X_0, X_1} \| v_t(e^{-t} X_1 + \sqrt{1 - e^{-2t}} X_0) - X_0 \|^2 \qquad (4)$$

Detour: Maximum Likelihood Normalizing Flows vs. Diffusion Models

We've seen a similar story in generative modeling...

- *Continuous Normalizing Flows* (CNFs) [Che+18]: originally trained to maximize the log-likelihood of the generated samples, using the *adjoint method*.

$$\min_{v} \mathbb{E}\left[-\log p_0(X_0^v) + \int_0^1 \nabla \cdot v_t(X_t^v)\, dt \right], \text{ s.t. } \begin{cases} dX_t^v = v(X_t^v, t)dt, \\ X_0^v \sim p_0. \end{cases}$$

- This loss is **highly non-convex** w.r.t v!
- Diffusion models superseded maximum likelihood CNFs, and require solving least squares problems:

$$\text{DDPM [HJA20]:} \qquad \min_v \mathbb{E}_{t,X_0,X_1} \| v_t(e^{-t}X_1 + \sqrt{1-e^{-2t}}X_0) - X_0 \|^2 \tag{4}$$

- This loss is **convex** with respect to v!

From non-convex to convex landscapes

Task	**Non-convex functional landscape**	**Least squares functional landscape**
Generative modeling	Maximum Likelihood CNFs	Diffusion models, Flow Matching
Stochastic optimal control	Adjoint method	?

From non-convex to convex landscapes

Task	Non-convex functional landscape	Least squares functional landscape
Generative modeling	Maximum Likelihood CNFs	Diffusion models, Flow Matching
Stochastic optimal control	Adjoint method	Stochastic Optimal Control Matching (ours)

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \|u(X_t^u, t)\|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v} \big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v) \big],$$

where

- $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is the control

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\,dt + g(X_T^u)\Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v}\big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\big],$$

where

- $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is the control
- v is a fixed arbitrary control, X^v is the solution of the SDE with control v

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \|u(X_t^u, t)\|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v} \big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v) \big],$$

where

- $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is the control
- v is a fixed arbitrary control, X^v is the solution of the SDE with control v
- $M : [0, 1]^2 \to \mathbb{R}^{d \times d}$ is the *reparameterization matrix*

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2} \|u(X_t^u, t)\|^2 + f(X_t^u, t) \Big) \, dt + g(X_T^u) \Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v} \big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v) \big],$$

where

- $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is the control
- v is a fixed arbitrary control, X^v is the solution of the SDE with control v
- $M : [0, 1]^2 \to \mathbb{R}^{d \times d}$ is the *reparameterization matrix*
- w is the *matching vector field*

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v}\big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\big],$$

where

- $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is the control
- v is a fixed arbitrary control, X^v is the solution of the SDE with control v
- $M : [0,1]^2 \to \mathbb{R}^{d \times d}$ is the *reparameterization matrix*
- w is the *matching vector field*
- α is the *importance weight*

SOCM: a least squares loss for stochastic control

Stochastic Optimal Control problem

$$\min_{u \in \mathcal{U}} \mathcal{L}(u) \triangleq \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\|u(X_t^u, t)\|^2 + f(X_t^u, t)\Big)\, dt + g(X_T^u)\Big],$$

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

$$\downarrow$$

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u, M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t, X^v}\big[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\big],$$

where

- $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is the control
- v is a fixed arbitrary control, X^v is the solution of the SDE with control v
- $M : [0,1]^2 \to \mathbb{R}^{d \times d}$ is the *reparameterization matrix*
- w is the *matching vector field*
- α is the *importance weight*

w and α depend on f, g, λ, σ (full expressions later on).

Some details on SOCM

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

Some details on SOCM

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

- **How to derive SOCM?** The optimal control admits an analytic expression as the score (gradient of logarithm) of an unnormalized density! Similar to score-based diffusion.

Some details on SOCM

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

- **How to derive SOCM?** The optimal control admits an analytic expression as the score (gradient of logarithm) of an unnormalized density! Similar to score-based diffusion.
- **Key idea:** *path-wise reparameterization trick*, a novel technique to reexpress the gradient of a conditional expectation.

Some details on SOCM

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

- **How to derive SOCM?** The optimal control admits an analytic expression as the score (gradient of logarithm) of an unnormalized density! Similar to score-based diffusion.
- **Key idea:** *path-wise reparameterization trick*, a novel technique to reexpress the gradient of a conditional expectation.
- **What is the role of the reparameterization matrix M?**

$$\mathcal{L}(u, M) = \underbrace{\mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - u^*(X_t^v, t)\|^2 \alpha(v, X^v, B)\right]}_{L^2 \text{ error of } u}$$
$$+ \underbrace{\mathbb{E}_{t,X^v}\left[\left\| w(t, v, X^v, B, M_t) - \frac{\mathbb{E}[w(t, v, X^v, B, M_t)\alpha(v, X^v, B)|t, X_t^v]}{\mathbb{E}[\alpha(v, X^v, B)|t, X_t^v]}\right\|^2 \alpha(v, X^v, B)\right]}_{\text{Conditional variance of } w}$$

We train M to minimize the conditional variance of the matching vector field w.

Some details on SOCM

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

- **How to derive SOCM?** The optimal control admits an analytic expression as the score (gradient of logarithm) of an unnormalized density! Similar to score-based diffusion.
- **Key idea:** *path-wise reparameterization trick*, a novel technique to reexpress the gradient of a conditional expectation.
- **What is the role of the reparameterization matrix M?**

$$\mathcal{L}(u, M) = \underbrace{\mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - u^*(X_t^v, t)\|^2 \alpha(v, X^v, B)\right]}_{L^2 \text{ error of } u}$$

$$+ \underbrace{\mathbb{E}_{t,X^v}\left[\left\|w(t, v, X^v, B, M_t) - \frac{\mathbb{E}[w(t, v, X^v, B, M_t)\alpha(v, X^v, B)|t, X_t^v]}{\mathbb{E}[\alpha(v, X^v, B)|t, X_t^v]}\right\|^2 \alpha(v, X^v, B)\right]}_{\text{Conditional variance of } w}$$

We train M to minimize the conditional variance of the matching vector field w.

- **How do we choose v?** We want v such that $\alpha(v, X^v)$ has low variance. In general, we take v to be the current learned control u.

Experiments: settings, baselines and ablations

Settings:

- Quadratic Ornstein Uhlenbeck / Linear Quadratic Regulator: Linear base drift b, quadratic state cost f, quadratic terminal cost g
- Linear Ornstein Uhlenbeck: Linear base drift b, quadratic state cost f, linear terminal cost g
- Double Well: terminal cost g is minus log-density of high-dimensional double well (2^d modes).

Experiments: settings, baselines and ablations

Settings:

- Quadratic Ornstein Uhlenbeck / Linear Quadratic Regulator: Linear base drift b, quadratic state cost f, quadratic terminal cost g
- Linear Ornstein Uhlenbeck: Linear base drift b, quadratic state cost f, linear terminal cost g
- Double Well: terminal cost g is minus log-density of high-dimensional double well (2^d modes).

Baselines:

- Adjoint method [Pon62]
- Cross-entropy loss [Zha+14]
- Log-variance loss [NR23]
- Variance loss [NR23]
- Moment loss [WHJ17; HJE18]

Experiments: settings, baselines and ablations

Settings:

- Quadratic Ornstein Uhlenbeck / Linear Quadratic Regulator: Linear base drift b, quadratic state cost f, quadratic terminal cost g
- Linear Ornstein Uhlenbeck: Linear base drift b, quadratic state cost f, linear terminal cost g
- Double Well: terminal cost g is minus log-density of high-dimensional double well (2^d modes).

Baselines:

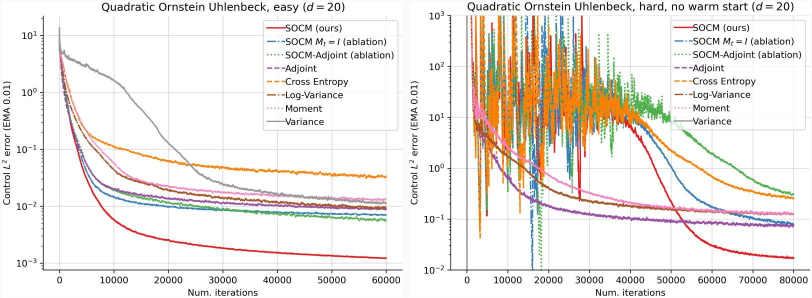
- Adjoint method [Pon62]
- Cross-entropy loss [Zha+14]
- Log-variance loss [NR23]
- Variance loss [NR23]
- Moment loss [WHJ17; HJE18]

Ablations:

- SOCM with constant $M_t = \mathrm{Id}$
- SOCM-Adjoint: modification of SOCM where the adjoint method is used instead of the path-wise reparameterization trick

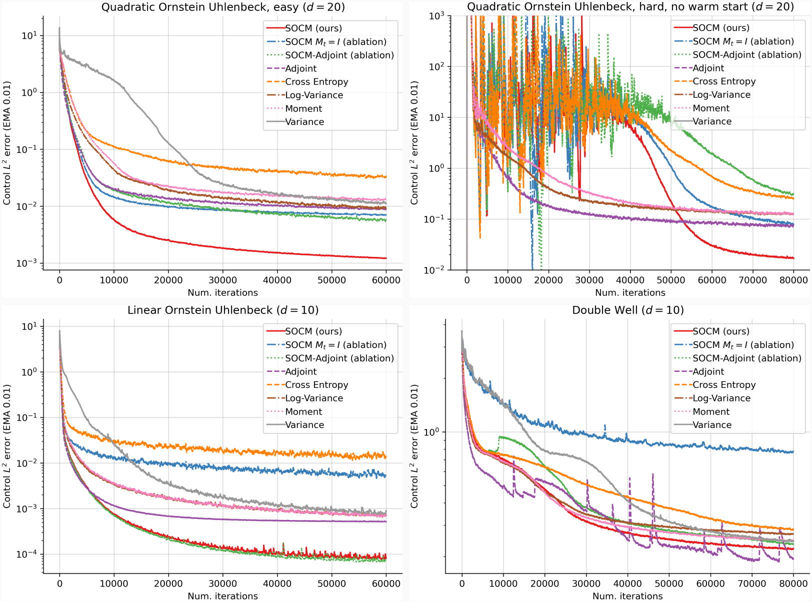
Experiments: Control L^2 error

Control L^2 error: $\mathbb{E}_{t,X^v}\left[\|u(X_t^v,t) - u^*(X_t^v,t)\|^2 \alpha(v,X^v,B)\right] / \mathbb{E}_{t,X^v}\left[\alpha(v,X^v,B)\right]$



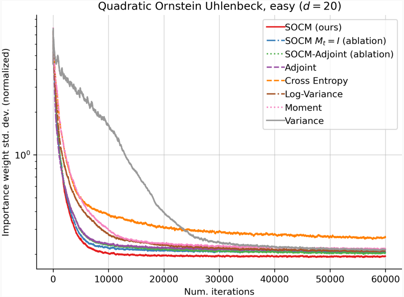
Experiments: Control L^2 error

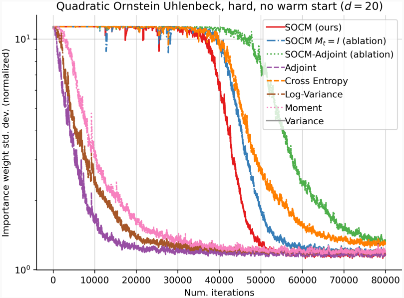
Control L^2 error: $\mathbb{E}_{t,X^v}\left[\|u(X_t^v,t) - u^*(X_t^v,t)\|^2 \alpha(v, X^v, B)\right] / \mathbb{E}_{t,X^v}\left[\alpha(v, X^v, B)\right]$



Experiments: Importance weight variance

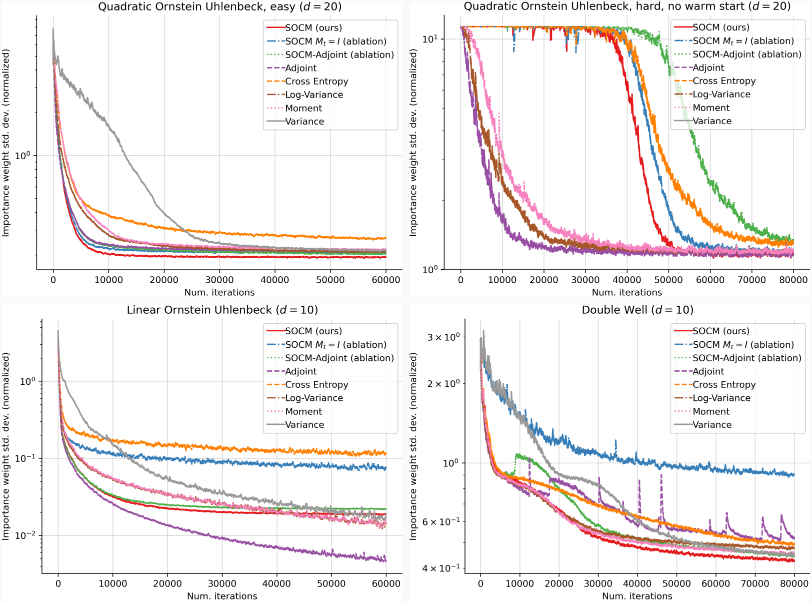
Importance weight variance: $\mathrm{Var}[\alpha(u, X^u, B)]/\mathbb{E}[\alpha(u, X^u, B)] = \mathrm{Var}[F(X^u)\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$





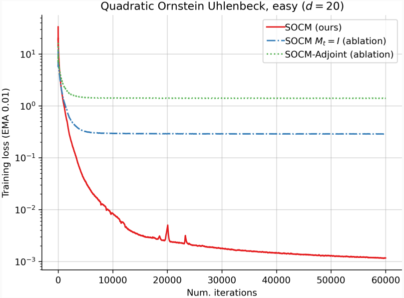
Experiments: Importance weight variance

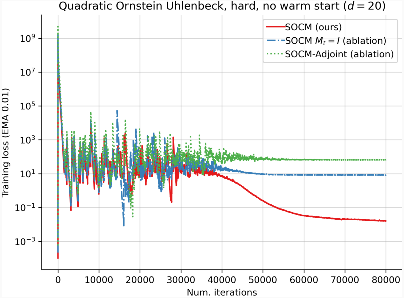
Importance weight variance: $\mathrm{Var}[\alpha(u, X^u, B)]/\mathbb{E}[\alpha(u, X^u, B)] = \mathrm{Var}[F(X^u)\frac{d\mathbb{P}}{d\mathbb{P}^u}(X^u)]$



Experiments: Training loss for SOCM and ablations

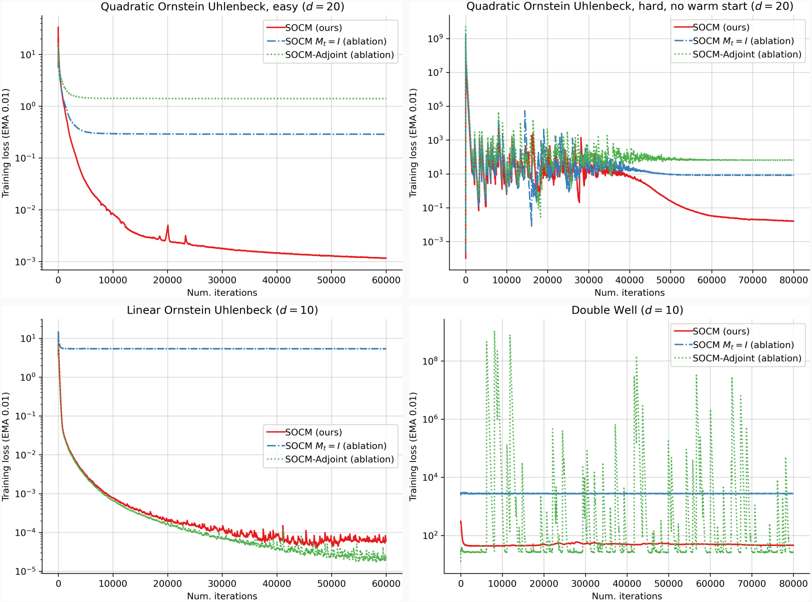
$\mathcal{L}(u, M) = L^2$ error of u + Conditional variance of w





Experiments: Training loss for SOCM and ablations

$\mathcal{L}(u, M) = L^2$ error of u + Conditional variance of w



Outline

Derivation of the SOCM loss (1/3)

Reminders:

- X^u is the process controlled by u; it is the solution of
$$dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0.$$
- X is the uncontrolled process; it is the solution of
$$dX_t = b(X_t, t)\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

Derivation of the SOCM loss (1/3)

Reminders:

- X^u is the process controlled by u; it is the solution of
$$dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0.$$
- X is the uncontrolled process; it is the solution of
$$dX_t = b(X_t, t)\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

Lemma (Path-integral representation of the optimal control)

The optimal control u^* satisfies
$$u^*(x,t) = \lambda\sigma(t)^\top \nabla_x \log \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s, s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big]. \quad (5)$$

Derivation of the SOCM loss (1/3)

Reminders:

- X^u is the process controlled by u; it is the solution of
$$dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\,dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0.$$
- X is the uncontrolled process; it is the solution of
$$dX_t = b(X_t, t)\,dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

Lemma (Path-integral representation of the optimal control)

The optimal control u^* satisfies
$$u^*(x,t) = \lambda\sigma(t)^\top \nabla_x \log \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s, s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big]\,. \quad (5)$$

Derivation of the SOCM loss (1/3)

Reminders:

- X^u is the process controlled by u; it is the solution of
$$dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\,dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0.$$
- X is the uncontrolled process; it is the solution of
$$dX_t = b(X_t, t)\,dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

Lemma (Path-integral representation of the optimal control)

The optimal control u^* satisfies
$$u^*(x,t) = \lambda\sigma(t)^\top \nabla_x \log \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s, s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big]\,. \tag{5}$$

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

Reminders:

- X^u is the process controlled by u; it is the solution of
$$dX_t^u = (b(X_t^u, t) + \sigma(t)u(X_t^u, t))\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0^u \sim p_0.$$
- X is the uncontrolled process; it is the solution of
$$dX_t = b(X_t, t)\, dt + \sqrt{\lambda}\sigma(t)dB_t, \qquad X_0 \sim p_0.$$

Lemma (Path-integral representation of the optimal control)

The optimal control u^* satisfies
$$u^*(x,t) = \lambda\sigma(t)^\top \nabla_x \log \mathbb{E}\big[\exp\big(-\lambda^{-1}\textstyle\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big]\ . \quad (5)$$

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

The only optimum of this loss is u^*. Using equation (5), the cross-term can be rewritten as:
$$\begin{aligned}&\mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \langle u(X_t,t), u^*(X_t,t)\rangle\,\mathrm{d}t \exp\big(-\lambda^{-1}\int_0^T f(X_t,t)\,\mathrm{d}t - \lambda^{-1}g(X_T)\big)\big] \\ &= -\lambda\mathbb{E}\big[\tfrac{1}{T}\textstyle\int_0^T \big\langle u(X_t,t), \sigma(t)^\top\nabla_x\mathbb{E}\big[\exp\big(-\lambda^{-1}\int_t^T f(X_s,s)\,\mathrm{d}s - \lambda^{-1}g(X_T)\big)\big|X_t = x\big]\big\rangle\,\mathrm{d}t \\ &\qquad \times \exp\big(-\lambda^{-1}\textstyle\int_0^t f(X_s,s)\,\mathrm{d}s\big)\big].\end{aligned}$$

Derivation of the SOCM loss (2/3)

To evaluate the derivative of the conditional expectation, we use:

Proposition (Path-wise reparameterization trick for stochastic optimal control)

For each $t \in [0, T]$, let $M_t : [t, T] \to \mathbb{R}^{d \times d}$ be an arbitrary continuously differentiable function matrix-valued function such that $M_t(t) = \mathrm{Id}$. We have that

$$
\begin{aligned}
&\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1} \textstyle\int_t^T f(X_s, s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big] \\
&= \mathbb{E}\big[\big(-\lambda^{-1} \textstyle\int_t^T M_t(s)\nabla_x f(X_s, s)\,\mathrm{d}s - \lambda^{-1} M_t(T)\nabla g(X_T) \\
&\qquad + \lambda^{-1/2} \textstyle\int_t^T (M_t(s)\nabla_x b(X_s, s) - \partial_s M_t(s))(\sigma^{-1})^\top (X_s, s)\mathrm{d}B_s\big) \\
&\qquad \times \exp\big(-\lambda^{-1} \textstyle\int_t^T f(X_s, s)\,\mathrm{d}s - \lambda^{-1} g(X_T)\big)\big|X_t = x\big].
\end{aligned}
\tag{6}
$$

Using (6) and completing the square, we obtain that for some constant K independent of u,

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

If we perform a change of process from X to X^v by applying the Girsanov theorem, where v is arbitrary, we obtain the loss $\mathcal{L}_{\mathrm{SOCM}}(u, M)$.

Derivation of the SOCM loss (3/3)

Stochastic Optimal Control Matching (SOCM) [Dom+23]

$$\min_{u,M} \mathcal{L}(u, M) \triangleq \mathbb{E}_{t,X^v}\left[\|u(X_t^v, t) - w(t, v, X^v, M)\|^2 \alpha(v, X^v)\right],$$

$$\begin{aligned}
w(t, v, X^v, M) = &- \int_t^T M(t, s)\nabla_x f(X_s^v, s)\, ds - M(t, T)\nabla g(X_T^v) \\
&- \int_t^T (M(t, s)\nabla_x b(X_s^v, s) - \partial_s M(t, s)) v(X_s^v, s)\, ds \\
&- \sqrt{\lambda} \int_t^T (M(t, s)\nabla_x b(X_s^v, s) - \partial_s M(t, s))\, dB_s, \\
\alpha(v, X^v) = \exp\Bigg(&- \frac{1}{\lambda}\int_0^T f(X_t^v, t)\, ds - \frac{1}{\lambda} g(X_T^v) \\
&- \frac{1}{\sqrt{\lambda}} \int_0^T \langle v(X_t^v, t), dB_t\rangle - \frac{1}{2\lambda}\int_0^T \|v(X_t^v, t)\|^2\, dt \Bigg)
\end{aligned}$$

Future directions within stochastic optimal control

Applications of SOCM

- Test and benchmark SOCM as an algorithm to sample from unnormalized densities

Applications of SOCM

- Test and benchmark SOCM as an algorithm to sample from unnormalized densities
- Test and benchmark SOCM as an importance sampling algorithm for stopped diffusions (more in backup slides)

Future directions within stochastic optimal control

Applications of SOCM

- Test and benchmark SOCM as an algorithm to sample from unnormalized densities
- Test and benchmark SOCM as an importance sampling algorithm for stopped diffusions (more in backup slides)

Technical improvements of SOCM

- Make reparameterization matrix M depend to the controlled process X^v.

Future directions within stochastic optimal control

Applications of SOCM

- Test and benchmark SOCM as an algorithm to sample from unnormalized densities
- Test and benchmark SOCM as an importance sampling algorithm for stopped diffusions (more in backup slides)

Technical improvements of SOCM

- Make reparameterization matrix M depend to the controlled process X^v.
- Test alternative way to use Girsanov theorem to lower the gradient variance when learning controls: explicit perturbations ψ. It can be combined with the path-wise reparameterization trick and also applied to other methods like the adjoint, cross-entropy, log-variance...

Future applications of the path-wise reparameterization trick

PWRT for diffusion processes

- The PWRT can be used to *generalize conditional score matching* to diffusion processes with generic drifts.

Future applications of the path-wise reparameterization trick

PWRT for diffusion processes

- The PWRT can be used to *generalize conditional score matching* to diffusion processes with generic drifts.
- I.e., we can learn $\nabla \log p_t$ for X s.t. $dX_t = b(X_t, t)\, dt + \sigma(t)\, dB_t$ using a least-squares loss (previously only *implicit score matching* was available).

Future applications of the path-wise reparameterization trick

PWRT for diffusion processes

- The PWRT can be used to *generalize conditional score matching* to diffusion processes with generic drifts.
- I.e., we can learn $\nabla \log p_t$ for X s.t. $dX_t = b(X_t, t)\, dt + \sigma(t)\, dB_t$ using a least-squares loss (previously only *implicit score matching* was available).
- When we set a linear drift $b(x, t) = -A(t)x$, we recover standard *conditional score matching*.

Future applications of the path-wise reparameterization trick

PWRT for diffusion processes

- The PWRT can be used to *generalize conditional score matching* to diffusion processes with generic drifts.
- I.e., we can learn $\nabla \log p_t$ for X s.t. $dX_t = b(X_t, t)\, dt + \sigma(t)\, dB_t$ using a least-squares loss (previously only *implicit score matching* was available).
- When we set a linear drift $b(x, t) = -A(t)x$, we recover standard *conditional score matching*.

PWRT for Neural SDEs

- Reminder: Neural SDEs are the SDE analog of Neural ODEs, they use the adjoint method.

Future applications of the path-wise reparameterization trick

PWRT for diffusion processes

- The PWRT can be used to *generalize conditional score matching* to diffusion processes with generic drifts.
- I.e., we can learn $\nabla \log p_t$ for X s.t. $dX_t = b(X_t, t)\, dt + \sigma(t)\, dB_t$ using a least-squares loss (previously only *implicit score matching* was available).
- When we set a linear drift $b(x, t) = -A(t)x$, we recover standard *conditional score matching*.

PWRT for Neural SDEs

- Reminder: Neural SDEs are the SDE analog of Neural ODEs, they use the adjoint method.
- We can replace the adjoint method by PWRT.

Informal derivation of the path-wise reparameterization trick (1/2)

Consider the Euler-Maruyama discretization $\hat{X} = (\hat{X}_k)_{k=0:K}$ of the uncontrolled process X with $K+1$ time steps (let $\delta = T/K$ be the step size):

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

Informal derivation of the path-wise reparameterization trick (1/2)

Consider the Euler-Maruyama discretization $\hat{X} = (\hat{X}_k)_{k=0:K}$ of the uncontrolled process X with $K+1$ time steps (let $\delta = T/K$ be the step size):

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

We can approximate

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

Informal derivation of the path-wise reparameterization trick (1/2)

Consider the Euler-Maruyama discretization $\hat{X} = (\hat{X}_k)_{k=0:K}$ of the uncontrolled process X with $K+1$ time steps (let $\delta = T/K$ be the step size):

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

We can approximate

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

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

$$\begin{aligned}&\mathbb{E}\big[\exp\big(-\lambda^{-1}\delta\textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1}g(\hat{X}_K)\big)\big|\hat{X}_0 = x\big] \\ &= C^{-1}\textstyle\iint_{(\mathbb{R}^d)^K} \exp\big(-\lambda^{-1}\delta\sum_{k=0}^{K-1} f(\hat{x}_k, s) - \lambda^{-1}g(\hat{x}_K) \\ &\qquad -\frac{1}{2\delta\lambda}\textstyle\sum_{k=1}^{K-1}\|\sigma^{-1}(k\delta)(\hat{x}_{k+1} - \hat{x}_k - \delta b(\hat{x}_k, k\delta))\|^2 \\ &\qquad -\frac{1}{2\delta\lambda}\|\sigma^{-1}(0)(\hat{x}_1 - x - \delta b(x, 0))\|^2\big)\,\mathrm{d}\hat{x}_1\cdots\mathrm{d}\hat{x}_K,\end{aligned} \tag{7}$$

where $C = \sqrt{(2\pi\delta\lambda)^K\prod_{k=0}^{K-1}\det((\sigma\sigma^\top)(k\delta))}$.

Informal derivation of the path-wise reparameterization trick (2/2)

We can write
$\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big)|\hat{X}_0 = x\big]$

Informal derivation of the path-wise reparameterization trick (2/2)

We can write

$$\nabla_x \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big) | \hat{X}_0 = x\big]$$
$$= \nabla_{z} \mathbb{E}\big[\exp\big(-\lambda^{-1}\delta \textstyle\sum_{k=0}^{K-1} f(\hat{X}_k, s) - \lambda^{-1} g(\hat{X}_K)\big) | \hat{X}_0 = x + z\big]|_{z=0}$$

Informal derivation of the path-wise reparameterization trick (2/2)

We can write

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

Informal derivation of the path-wise reparameterization trick (2/2)

We can write

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

- In the last equality, $\psi : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is 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 $s \in [0, T]$.

Informal derivation of the path-wise reparameterization trick (2/2)

We can write

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

- In the last equality, $\psi : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is 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 used that for $k \in \{1, \dots, K\}$, the variables $\hat{x}_k$ are integrated over $\mathbb{R}^d$, which means that adding an offset $\psi(z, k\delta)$ does not change the value of the integral. We also used that $\psi(z, 0) = z$.

Informal derivation of the path-wise reparameterization trick (2/2)

We can write

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

- In the last equality, $\psi : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is 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 used that for $k \in \{1, \dots, K\}$, the variables $\hat{x}_k$ are integrated over $\mathbb{R}^d$, which means that adding an offset $\psi(z, k\delta)$ does not change the value of the integral. We also used that $\psi(z, 0) = z$.
- To conclude the proof, we differentiate with respect to z under the integral sign, and define $M(s) = \nabla\psi(z, s)|_{z=0}$.

Content of the talk

Introduction

- Stochastic optimal control: definition
- Examples: robotics, sampling unnormalized densities, importance sampling for diffusions
- Existing approaches: the adjoint method

Stochastic Optimal Control Matching

- Comparing stochastic optimal control with normalizing flows
- Our algorithm: SOCM [2]
- Main features of our algorithm
- Experiments

Key ideas

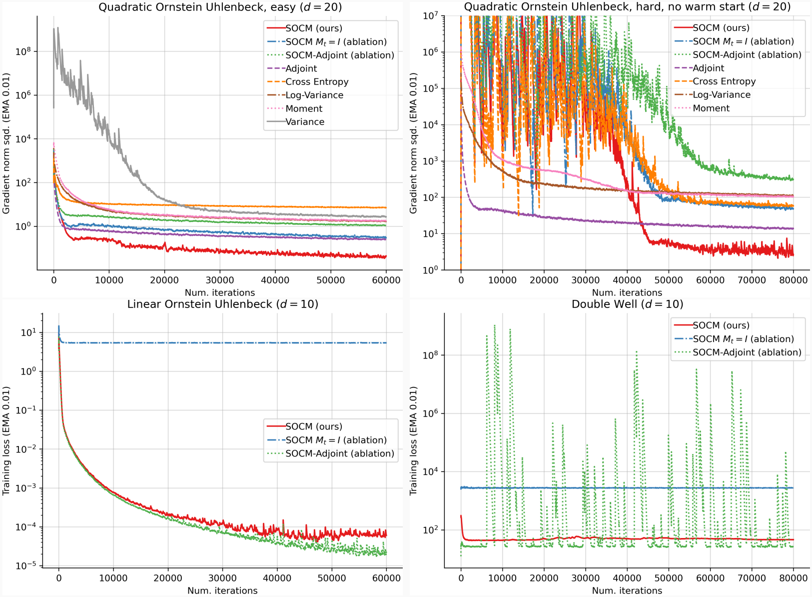
- Derivation of the SOCM loss
- The path-wise reparameterization trick
- Conclusions and future directions

[2] **Domingo-Enrich, C.**, Han, J., Amos, B., Bruna, J., Chen, R.T.Q. *Stochastic optimal control matching*, arXiv preprint, 2023.

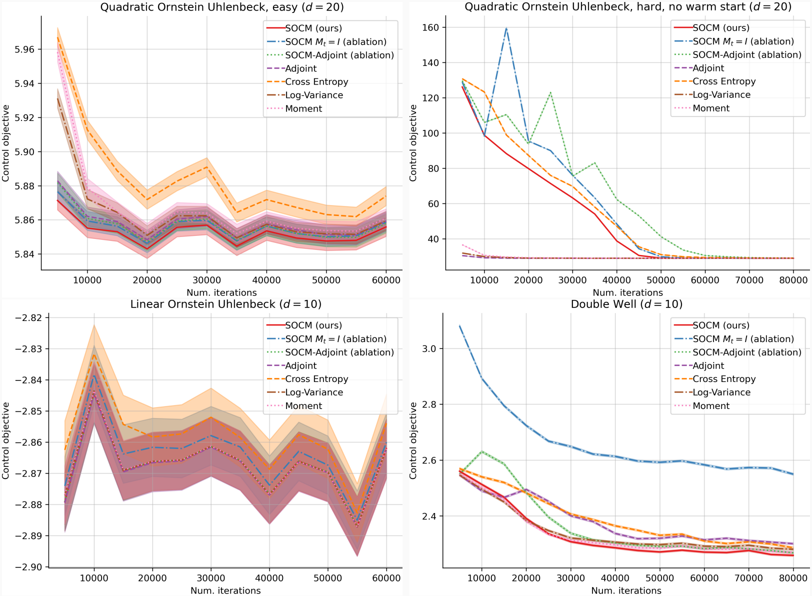
Thank you!

Contact: *cd2754@nyu.edu*

Experiments: Gradient norm



Experiments: Control objective



References i

[Bel57] Richard Bellman. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[BRU23] Julius Berner, Lorenz Richter, and Karen Ullrich. *An optimal control perspective on diffusion-based generative modeling*. 2023. arXiv: `2211.01364 [cs.LG]`.

[Che+18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. 2018.

[Dom+23] Carles Domingo-Enrich, Jiequn Han, Joan Bruna, Brandon Amos, and Ricky T. Q. Chen. *Stochastic Optimal Control Matching*. To appear. 2023.

[FB21] Grzegorz Ficht and Sven Behnke. "Bipedal Humanoid Hardware Design: a Technology Review". In: *Current Robotics Reports* 2 (2021), pp. 201–210. URL: `https://api.semanticscholar.org/CorpusID:232147157`.

[Fre+21] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. *Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation*. Version 0.9.3. 2021. URL: `http://github.com/google/brax`.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[HJE18] Jiequn Han, Arnulf Jentzen, and Weinan E. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.

[NR23] 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*. 2023.

[Onk+23] 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". In: *IEEE Transactions on Control Systems Technology* 31.1 (Jan. 2023), pp. 235–251.

[Pon62] L.S. Pontryagin. *The Mathematical Theory of Optimal Processes.* Interscience publishers. Interscience Publishers, 1962.

[WHJ17] E Weinan, Jiequn Han, and Arnulf Jentzen. "Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations". In: *Communications in Mathematics and Statistics* 5.4 (2017), pp. 349–380.

[Zha+14] 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". In: *SIAM Journal on Scientific Computing* 36.6 (2014), A2654–A2672.