Conditioning Diffusions Using Malliavin Calculus

Jakiw Pidstrigach, Libby Baker, Carles Domingo-Enrich, George Deligiannidis, and Nik Nüsken

Department of Statistics
University of Oxford

UNIVERSITY OF OXFORD
DEPARTMENT OF STATISTICS

Math4DL Conference,
University of Bath, 6th of July 2025

Reference System

Assume we have a system

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + \mathrm{d}B_t, \qquad X_0 = x_0. \tag{1}$$

Examples:

Reference System

Assume we have a system

$$dX_t = b_t(X_t)\, dt + dB_t, \qquad X_0 = x_0. \tag{1}$$

Examples:

- Stochastic dynamical system (Weather, Finance)

Reference System

Assume we have a system

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + \mathrm{d}B_t, \qquad X_0 = x_0. \tag{1}$$

Examples:

- Stochastic dynamical system (Weather, Finance)
- Diffusion Model / Flow Matching

Reference System

Assume we have a system

$$dX_t = b_t(X_t)\,dt + dB_t, \qquad X_0 = x_0. \tag{1}$$

Examples:

- Stochastic dynamical system (Weather, Finance)
- Diffusion Model / Flow Matching

More general diffusion coefficients possible!

Conditioning the Reference System

We now want to *modify* the reference dynamics.

Conditioning the Reference System

We now want to *modify* the reference dynamics.

We do this *to sample conditional events* of $X|Y = y$.

Conditioning the Reference System

We now want to *modify* the reference dynamics.

We do this *to sample conditional events* of $X|Y = y$.

- Probability of no rain in Bath, conditioned on it raining in Oxford?

Conditioning the Reference System

We now want to *modify* the reference dynamics.

We do this *to sample conditional events* of $X|Y = y$.

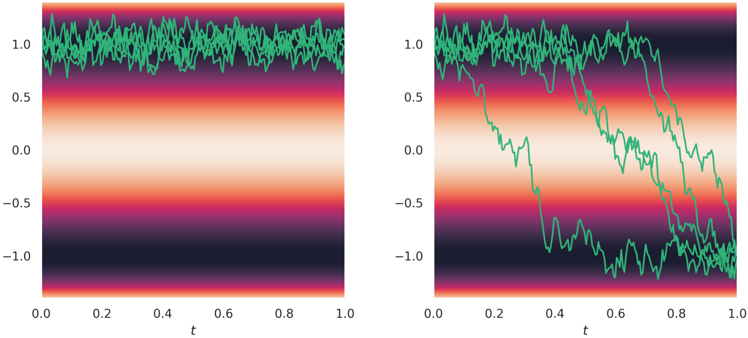
- Probability of no rain in Bath, conditioned on it raining in Oxford?

This can also be seen as *weighting the reference path measure* with a "reward" given by the *likelihood*

$$G(X_T; y).$$

.

Double Well Example



(a) Unconditioned Paths

(b) Conditioned Paths

Figure: Double Well Transitions

Singular Rewards

We concentrate on the case of Bridge simulation:

Bridge simulation: The task of conditioning X on

$$X_T = x_T.$$

Singular Rewards

We concentrate on the case of Bridge simulation:

Bridge simulation: The task of conditioning X on

$$X_T = x_T.$$

This can be seen as

Singular Rewards

We concentrate on the case of Bridge simulation:

Bridge simulation: The task of conditioning X on

$$X_T = x_T.$$

This can be seen as

1. Singular likelihood $G \approx \delta_{x_T}(X_T)$.

Singular Rewards

We concentrate on the case of Bridge simulation:

Bridge simulation: The task of conditioning X on

$$X_T = x_T.$$

This can be seen as

1. Singular likelihood $G \approx \delta_{x_T}(X_T)$.
2. Full-information observation $Y = X_T$.

Singular Rewards

We concentrate on the case of Bridge simulation:

Bridge simulation: The task of conditioning X on

$$X_T = x_T.$$

This can be seen as

1. Singular likelihood $G \approx \delta_{x_T}(X_T)$.
2. Full-information observation $Y = X_T$.

All algorithms can be generalised to other Y.

Doob's H-Transform

Goal: Condition

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + \mathrm{d}B_t.$$

on $X_T = x_t$.

Doob's H-Transform

Goal: Condition

$$dX_t = b_t(X_t)\,dt + u_t(X_t)dt + \,dB_t.$$

on $X_T = x_t$.

The solution is given by adding

$$u_t(x_t) = \nabla_{x_t} \log p(X_T = x_T | X_t = x_t).$$

Doob's H-Transform

Goal: Condition

$$dX_t = b_t(X_t)\,dt + u_t(X_t)dt + \,dB_t.$$

on $X_T = x_t$.

The solution is given by adding

$$u_t(x_t) = \nabla_{x_t} \log p(X_T = x_T | X_t = x_t).$$

However, $p(X_T = x_t | X_t = x_t)$ is unknown.

Doob's H-Transform

Goal: Condition

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + u_t(X_t)\mathrm{d}t + \mathrm{d}B_t.$$

on $X_T = x_t$.

The solution is given by adding

$$u_t(x_t) = \nabla_{x_t} \log p(X_T = x_T | X_t = x_t).$$

However, $p(X_T = x_t | X_t = x_t)$ is unknown.

How do we approximate it?

Inspirations from Optimal Control

We can rewrite

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{p(X_T = x_T | X_t = x_t)} \nabla_{x_t} p(X_T = x_T | X_t = x_t)$$

Inspirations from Optimal Control

We can rewrite

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{p(X_T = x_T | X_t = x_t)} \nabla_{x_t} p(X_T = x_T | X_t = x_t)$$

Now, formally, we can write:

$$\begin{aligned} \nabla p(X_T = x_T | X_t = x_t) =& \nabla_{x_t} \mathbb{E}[\delta_{x_T}(X_T^{x_t})] \\ =& \mathbb{E}[\nabla \delta_{x_T}(X_T^{x_t}) J_{T|t}], \end{aligned}$$

where $J_{T|t} = \nabla_{X_t} X_T$.

Inspirations from Optimal Control

We can rewrite

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{p(X_T = x_T | X_t = x_t)} \nabla_{x_t} p(X_T = x_T | X_t = x_t)$$

Now, formally, we can write:

$$\begin{aligned} \nabla p(X_T = x_T | X_t = x_t) =& \nabla_{x_t} \mathbb{E}[\delta_{x_T}(X_T^{x_t})] \\ =& \mathbb{E}[\nabla \delta_{x_T}(X_T^{x_t}) J_{T|t}], \end{aligned}$$

where $J_{T|t} = \nabla_{X_t} X_T$.

- This looks like we are optimizing the expected likelihood $G = \delta_{x_T}(X_T)$.

Inspirations from Optimal Control

We can rewrite

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{p(X_T = x_T | X_t = x_t)} \nabla_{x_t} p(X_T = x_T | X_t = x_t)$$

Now, formally, we can write:

$$\begin{aligned} \nabla p(X_T = x_T | X_t = x_t) =& \nabla_{x_t} \mathbb{E}[\delta_{x_T}(X_T^{x_t})] \\ =& \mathbb{E}[\nabla \delta_{x_T}(X_T^{x_t}) J_{T|t}], \end{aligned}$$

where $J_{T|t} = \nabla_{X_t} X_T$.

- This looks like we are optimizing the expected likelihood $G = \delta_{x_T}(X_T)$.
- The problem is that G is very singular, and has no derivatives.

What to do?

What do we do with $\mathbb{E}[\nabla \delta_{x_T}(X_T^{x_t}) J_{T|t}]$?

What to do?

What do we do with $\mathbb{E}[\nabla\delta_{x_T}(X_T^{x_t})J_{T|t}]$?

What do we do if we have the derivative of a non-smooth function?

What to do?

What do we do with $\mathbb{E}[\nabla \delta_{x_T}(X_T^{x_t}) J_{T|t}]$?

What do we do if we have the derivative of a non-smooth function?

(Everyone in unison) **We integrate by parts!**

(Excitement grips the room — chaos erupts, chairs are launched, people start yelling formulas)

Proof

Integration by parts is done via **Malliavin calculus**.

$$D_s\varphi(X_T) = \nabla\varphi(X_T)D_sX_T = \nabla\varphi(X_T)J_{T|s} \tag{2}$$

$$\nabla\varphi(X_T) = D_s\varphi(X_T)J_{T|s}^{-1}, \tag{3}$$

$$\nabla\varphi(X_T^x)\, J_{T|0} = \int_0^T D_s\varphi(X_T)J_{T|s}^{-1}\beta_s \mathrm{d}s\, J_{T|0} = \int_0^T D_s\varphi(X_T)J_{s|0}\beta_s \mathrm{d}s. \tag{4}$$

$$\mathbb{E}[\nabla_x\varphi(X_T^x)] = \mathbb{E}\left[\int_0^T D_s\varphi(X_T)J_{s|0}\beta_s \mathrm{d}s\right] \tag{5}$$

$$= \mathbb{E}\left[\varphi(X_T)\int_0^T J_{s|0}\beta_s)^\top \mathrm{d}B_s\right]. \tag{6}$$

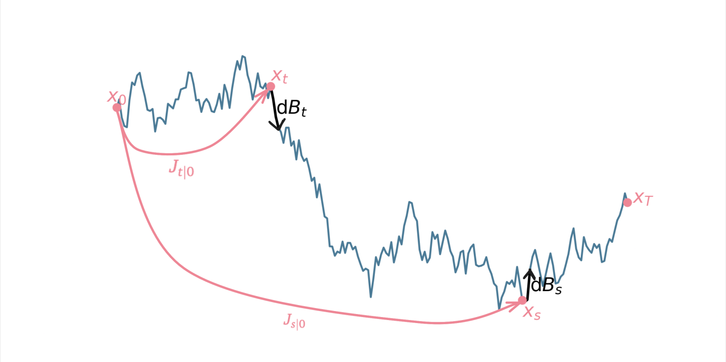
End Result

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{T-t}\mathbb{E}[\int_t^T J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

End Result

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{T-t}\mathbb{E}[\int_t^T J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

- We pick only the paths that land at x_T and predict the drift of the Brownian motion (It has a drift because of conditioning, think of Girsanov).



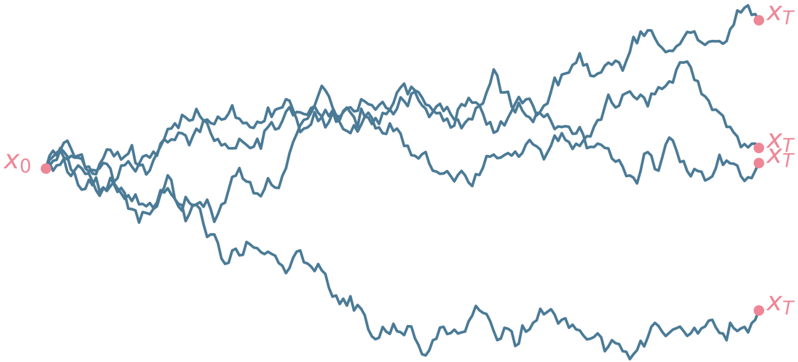
Simplified Special Case $X_t = B_t$

Special case: $X_t = B_t$ and therefore $J_{t|s} = \mathrm{Id}$. Then:

$$\nabla \log p(X_T = x_T | X_t = x_t) = \frac{1}{T-t} \mathbb{E}[B_T - B_t | X_t = x_t, X_T = x_T].$$

is predicting the **average drift** of the **driving Brownian motion**.

Amortization



Main Result

Theorem

For any β_s, define the score process

$$\mathcal{S}_t := \int_t^T \beta_s \, J_{s|t}^\top \mathrm{d}B_s.$$

Then the minimizer

$$u^* = \operatorname{argmin}_u \mathbb{E}\left[\int_0^T \left\|u_s(X_s; X_T) - \mathcal{S}_s\right\|^2 \, \mathrm{d}s\right],$$

is the diffusion bridge drift, i.e. the law of

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + u_t^*(X_t; x_T)\,\mathrm{d}t + \mathrm{d}B_t$$

coincides with the conditional law of the reference process X, given $X_T = x_T$.

Algorithm 1 BEL - Training Step

Require: $\alpha : [0,1] \to \mathbb{R}^{n\times n}$, initial condition x_0, batch size N, current drift approximation u^θ, time grid $\{t_0, t_1, \dots t_M\}$. Initialize

for $i = 1$ to N **do**

  Sample a sample path X with corresponding Brownian motion path B from the SDE X_t.

  Sample an observation $Y = G(X_T)$.

  Compute the Monte Carlo estimator $\mathcal{S}_s$.

  Calculate the single-path loss

$$l_i(\theta) = \sum_{j=1}^{M-1} \|u^\theta_{t_j}(X_{t_j}; Y) - \mathcal{S}_{t_j}(X, B)\|^2,$$

end for

Summ for the full-batch loss $\mathcal{L}^M(\theta) = \sum_{i=1}^N l_i(\theta)$.

Take a gradient step on $\mathcal{L}^B(\theta)$.

A few remarks

Remark 1: General Score Representation

Assume we have an SDE

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + \sigma_t(X_t)\,\mathrm{d}B_t, \tag{7}$$

Lemma

The score can be represented as

$$\nabla \log p_t(x) = \mathbb{E}\left[\int_0^t (\sigma(X_s)^{-1} J_{t|s}^{-1})^\top \alpha_s' \,\mathrm{d}B_s | X_t = x\right].$$

In particular, for any such α, the loss

$$\mathcal{L}(u) = \mathbb{E}\left[\|u_t(X_t) - \int_0^t (\sigma_s(X_s)^{-1} J_{t|s}^{-1})^\top \alpha_s' \,\mathrm{d}B_s\|^2\right] \tag{8}$$

has a unique minimiser given by $u_t(x) = \nabla \log p_t(x)$.

Remark 2: Infinite dimensions and manifolds

Since the formula only relies on conditional expectations, it can be naturally extended to **infinite dimensional** or **manifold-valued** settings.

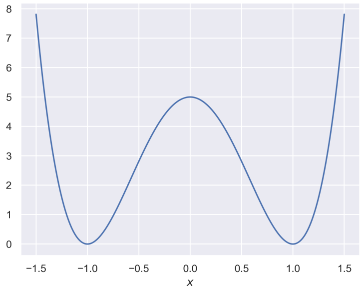
Remark 2: Infinite dimensions and manifolds

Since the formula only relies on conditional expectations, it can be naturally extended to **infinite dimensional** or **manifold-valued** settings.

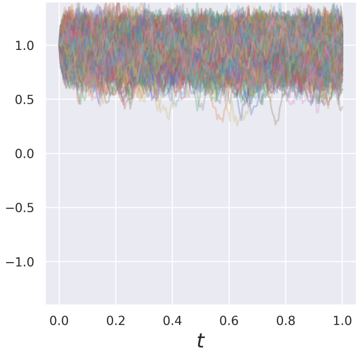
This is not the case for formulas involving the Lebesgue-density p_t, as in $\nabla \log p_t$.

Now some nice pictures!

Double Well

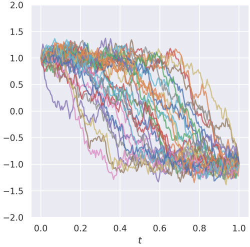


(a)

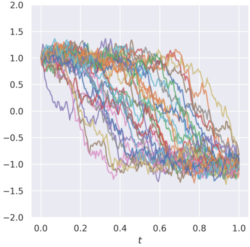


(b)

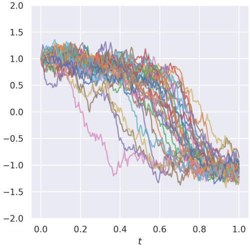
Figure: In (a), we plot a double-well potential. In (b), we sample 1,000 paths from the unconditioned SDE and observe that they remain confined to a single potential minimum, failing to transition between them. This highlights the inherent difficulty of the problem.



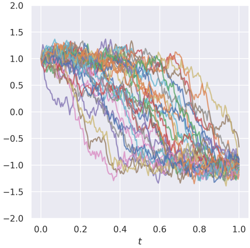
(a) Ground Truth



(b) BEL-first



(c) BEL-last

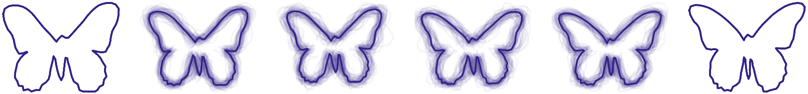


(d) BEL-optimal

Figure: Double Well 1D

Shape Spaces

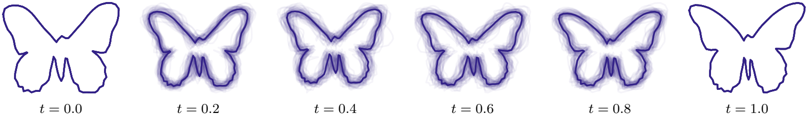
Infinite dimensional problem in theory, non-constant diffusion coefficient.

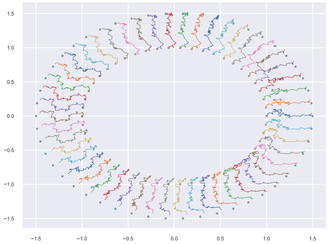


$t = 0.0$ $t = 0.2$ $t = 0.4$ $t = 0.6$ $t = 0.8$ $t = 1.0$

Shape Spaces

Infinite dimensional problem in theory, non-constant diffusion coefficient.

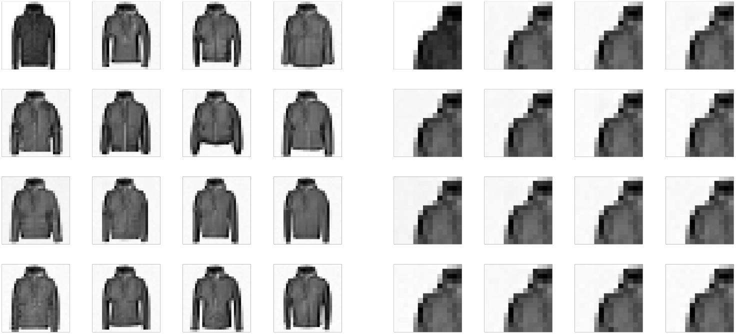




Method	Dist
BEL average	0.085
Time reversal	0.090
Adjoint paths	0.498
Untrained	1.396

Figure: Performance Metrics. We outperform competing methods.

Diffusion Models



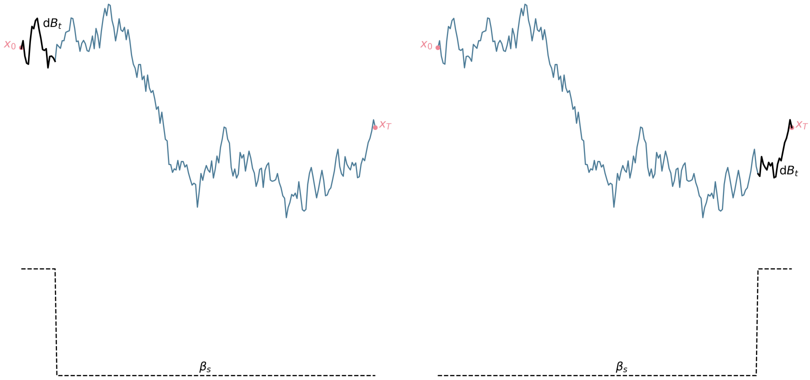
(a) (b)

Figure: In panel (a), the top-left image shows the ground truth used for conditioning. The remaining 15 images are samples generated by the diffusion model conditioned on the upper-left quarter of the ground truth image. Panel (b) displays only the conditioning inputs: again, the top-left image is the ground truth, while the others show the corresponding conditioned quarters used for generation.

Thank you!

More general formula: We can weigh which part of the Brownian motion we want to predict:

$$\nabla \log p(X_T = x_T | X_t = x_t) = \mathbb{E}[\int_t^T \beta_s \, J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$



Deriving the Loss

$$\nabla \log p(X_T = x_T | X_t = x_t) = \mathbb{E}[\int_t^T \beta_s \, J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

This tells us that if we can get Monte Carlo estimates of

$$\mathcal{S}_t := \int_t^T \beta_s J_{s|t}^\top \mathrm{d}B_s,$$

conditioned on X_T, then we can regress against them.

Deriving the Loss

$$\nabla \log p(X_T = x_T | X_t = x_t) = \mathbb{E}[\int_t^T \beta_s \, J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

This tells us that if we can get Monte Carlo estimates of

$$\mathcal{S}_t := \int_t^T \beta_s J_{s|t}^\top \mathrm{d}B_s,$$

conditioned on X_T, then we can regress against them.

Problem:

Deriving the Loss

$$\nabla \log p(X_T = x_T | X_t = x_t) = \mathbb{E}[\int_t^T \beta_s\, J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

This tells us that if we can get Monte Carlo estimates of

$$\mathcal{S}_t := \int_t^T \beta_s J_{s|t}^\top \mathrm{d}B_s,$$

conditioned on X_T, then we can regress against them.

Problem: We need samples from X conditioned on x_T.

Deriving the Loss

$$\nabla \log p(X_T = x_T | X_t = x_t) = \mathbb{E}[\int_t^T \beta_s \, J_{s|t}^\top \mathrm{d}B_s | X_t = x_t, X_T = x_T].$$

This tells us that if we can get Monte Carlo estimates of

$$\mathcal{S}_t := \int_t^T \beta_s J_{s|t}^\top \mathrm{d}B_s,$$

conditioned on X_T, then we can regress against them.

Problem: We need samples from X conditioned on x_T.

Solution: Amortization!

How to do general Observations?

- We now know how to control a diffusion towards a single endpoint x_T.

How to do general Observations?

- We now know how to control a diffusion towards a single endpoint x_T.
- If we have a general observation $Y = G(X_T)$, how can we steer it towards some X_T which generates that expectation? (Bayesian Posterior).

How to do general Observations?

- We now know how to control a diffusion towards a single endpoint x_T.
- If we have a general observation $Y = G(X_T)$, how can we steer it towards some X_T which generates that expectation? (Bayesian Posterior).

How to do general Observations?

- We now know how to control a diffusion towards a single endpoint x_T.
- If we have a general observation $Y = G(X_T)$, how can we steer it towards some X_T which generates that expectation? (Bayesian Posterior).

Proposition

We have that

$$\begin{aligned}\nabla_{X_t} \log p(Y|X_s) &= \mathbb{E}[\nabla_{X_t} \log p_t(X_T|X_s)|Y, X_s] \\ &= \mathbb{E}[\mathcal{S}_s|Y, X_s]\end{aligned}$$

Remark 2: Choice of β_s

Different choices of β lead to different algorithms:

1. Predict only the near future.
2. Predict the last movements of the Brownian motion before it hits the target.
3. Predict average movement of the Brownian motion over the full path.
4. Pick β_s to minimize the variance of the loss.
5. Learn β_s.

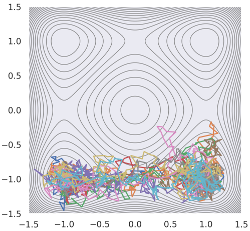
Remark 3: Calculation of $J_{s|t}$

- We never need to calculate the full matrix $J_{s|t}$.
- Can be done via an adjoint SDE method.

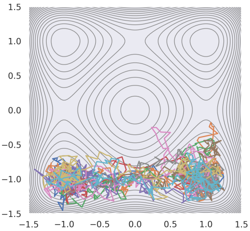
Remark 4: We rediscover other methods as special cases

- BEL-First is similar to a method used in the paper [1].
- The “Reparameterization Trick” from [2] is also a special case.

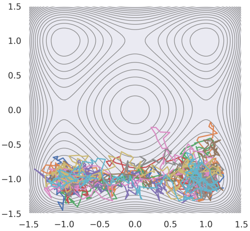
[1] *Simulating Diffusion Bridges with Score Matching*; Heng, De Bortoli, Doucet, Thornton
[2] *Stochastic Optimal Control Matching*; Domingo-Enrich, Han, Amos, Bruna, Chen



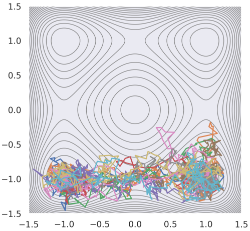
(a) Ground Truth



(b) BEL-first



(c) BEL-average



(d) BEL-optimal

Figure: Double Well 2D

Metrics

Loss	MV	Dist
BEL average	3.4×10^-1	3.0×10^-1
BEL first	2.1×10^-1	3.3×10^-1
BEL last	5.9×10^-1	1.1×10^0
BEL optimal	3.1×10^-1	3.0×10^-1
Reparametrization Trick	5.5×10^-1	5.2×10^-1