
Transition Path Sampling Using Gradient Matching

Balakrishnan Nagaraj

Abstract

Transition path sampling (TPS) is a powerful molecular dynamics technique for efficiently sampling rare but essential events, such as chemical reactions and protein folding, by focusing on transition paths between metastable states, thereby overcoming the limitations of standard simulations. In this work, we introduce an expressive and accurate transition path sampling approach based on training a neural network potential specifically designed to realize a given transition. The potential is trained by minimizing the Kullback–Leibler divergence between the path probability distributions induced by Langevin dynamics under the original molecular force field and those induced by the neural network potential. The learned potential guides the system smoothly along the transition pathway, reducing the likelihood of becoming trapped in the initial or other metastable states.

1. Introduction

In molecular dynamics simulations, transitions between states of interest often occur on timescales that are prohibitively long. For example, in protein dynamics, transitions between open and closed conformations may take on the order of milliseconds, making direct simulation computationally infeasible. This challenge arises because the majority of computational effort is spent resolving thermal fluctuations within metastable states rather than the rare transition events themselves. Consequently, developing methods that accelerate the sampling of transitions between metastable states is of fundamental interest.

In this work, we propose a neural network potential explicitly designed using specified initial and final metastable states to efficiently sample transition paths between them. The potential is trained by minimizing the Kullback–Leibler (KL) divergence between the path probability distributions generated by Langevin dynamics under the original molecular force field—which represents the true physical system—and those generated by the neural network potential. Rather than approximating the entire potential energy surface, the neural network focuses specifically on the transition region, where rare events occur. This targeted modeling

enables efficient capture of the critical transition dynamics and leads to significant computational speedup. Although the integration time step remains the same as that used in standard molecular dynamics simulations, rare transition events are observed much more frequently due to the design of the neural network potential, which actively guides the system along the transition pathway. Our contributions include:

1. Derivation of a simplified expression for the Kullback–Leibler divergence between path probability measures induced by two stochastic differential equations under fixed endpoint constraints.
2. Design of end-points conditioned neural network potential via interpolants that enable learning all possible metastable state transitions.

When the initial state is fixed, the Kullback–Leibler (KL) divergence between path probability measures induced by two stochastic differential equations can be obtained using Girsanov’s theorem (Särkkä & Solin, 2019). We extend this setting to the case in which both the initial and final states are fixed. To construct paths satisfying these boundary conditions, we introduce a neural network potential explicitly designed to enforce the prescribed endpoints.

The advantages of our method are:

1. The proposed approach does not require computationally expensive molecular dynamics data of rare transition events for training. Requiring such data would defeat the purpose of accelerating the sampling of rare transitions.
2. The underlying molecular dynamics are learned directly from the molecular force field. As a result, the method is physics-informed and produces physically consistent and accurate transition paths.

2. Background

The Kullback–Leibler divergence between path probability measures has appeared in various contexts in the literature, including Schrödinger bridge problems. Here, we adapt this framework to the construction of transition paths between

two metastable states on the molecular potential energy surface.

We begin by considering the SDE of the form

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}_t, \quad (1)$$

a generalized version of the first-order Langevin dynamics equation used in MD simulations

$$\frac{d\mathbf{x}_t}{dt} = -\frac{D_0}{k_B T} \nabla U(\mathbf{x}_t) + \sqrt{2D_0} \mathbf{R}(t) \quad (2)$$

where $\mathbf{R}(t)$ is a Gaussian process. Given the initial and final states: $\mathbf{x}_0 = \mathcal{A}$ and $\mathbf{x}_T = \mathcal{B}$, we are interested in sampling all the paths between them.

Proposition 2.1. *The path-probability $P(\{\mathbf{x}_t\}_{t \in [0, T]})$ of paths with fixed \mathbf{x}_0 and \mathbf{x}_T can be represented in the following path integral form:*

$$P(\{\mathbf{x}_t\}_{t \in [0, T]}) = e^{-\mathcal{S}[D\mathbf{x}_t]}, \quad (3)$$

with $\mathcal{S} := \int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t)dt + J$, where $L(\dot{\mathbf{x}}_t, \mathbf{x}_t)$ is called *Onsager–Machlup function* (Onsager & Machlup, 1953) defined by

$$L(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \mathbf{f}(\mathbf{x}_t, t)\|^2}{2g(t)^2},$$

and J is the Jacobian associated with the chosen discretization scheme in stochastic process.

We will be following Ito’s discretization rule for all our computations. For Ito’s discretization, it can be shown that the Jacobian J is zero. The path measure $[D\mathbf{x}_t]$ contains the normalization factor.

In order to sample from the path-probability, we will construct an alternative system which is easy to sample from and optimize for the parameters in the system by minimizing the KL divergence between the path-probabilities of the two systems.

Proposition 2.2. *Given two path probabilities $P(\{\mathbf{x}_t\}_{t \in [0, T]})$ and $Q(\{\mathbf{x}_t\}_{t \in [0, T]})$ between initial and final states $\mathbf{x}_0 = \mathcal{A}$ and $\mathbf{x}_T = \mathcal{B}$, the KL divergence of path-probabilities can be represented by the path integral form,*

$$D_{\text{KL}}(Q\|P) = \int e^{-\tilde{\mathcal{S}}} \log \frac{e^{-\tilde{\mathcal{S}}}}{e^{-\mathcal{S}}} [D\mathbf{x}_t],$$

and it can be computed to be

$$D_{\text{KL}}(Q\|P) = \int_0^T \frac{1}{2g(t)^2} \mathbb{E}_{q_t} \|\tilde{\mathbf{f}}(\mathbf{x}_t, t) - \mathbf{f}(\mathbf{x}_t, t)\|^2 dt. \quad (4)$$

For the molecular-dynamics Langevin equation, the KL divergence is

$$D_{\text{KL}}(Q\|P) = \frac{D_0}{4(k_B T)^2} \int_0^T \mathbb{E}_{q_t} \|\nabla U(\mathbf{x}_t, t) - \nabla \tilde{U}(\mathbf{x}_t, t)\|^2 dt. \quad (5)$$

In our work we take $U(\mathbf{x})$ to be the molecule’s force field and $\tilde{U}(\mathbf{x}_t, t)$ is parameterized by a neural network.

3. Method

We construct a Gaussian-like potential and minimize the KL divergence to optimize the constructed potential.

3.1. Approximating the potential

We approximate the potential along the most probable path as

$$\tilde{U}(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{c}_\theta(\tau))^\top \Sigma_\theta(\tau)^{-1} (\mathbf{x} - \mathbf{c}_\theta(\tau))$$

where

$$\mathbf{c}_\theta(t) = (1-t)\mathcal{A} + t\mathcal{B} + t(1-t)\mathbf{NN}_\theta(t, \mathcal{A}, \mathcal{B})$$

and

$$\Sigma_\theta(t) = t(1-t)\mathbf{NN}_\theta(t, \mathcal{A}, \mathcal{B}).$$

optimize it using the loss function

$$\mathcal{L}_\theta = \mathbb{E}_{t \sim U[0,1]} \mathbb{E}_{q_t} \|\nabla U(\mathbf{x}_t, t) - \nabla \tilde{U}(\mathbf{x}_t, t)\|^2. \quad (6)$$

In order to construct a positive semi-definite covariance matrix, we use Cholesky decomposition $\Sigma = \mathbf{L}\mathbf{L}^T$, where \mathbf{L} is a lower triangular matrix with positive diagonal elements. The points \mathcal{A} and \mathcal{B} can be chosen to be some representative point of each of the metastable states of the molecule.

3.2. Reparameterization

We will use the reparameterization $x_t = c_\theta(t) + \Sigma_\theta(t)^{1/2} \cdot \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, I)$. Then the KL divergence is given by

$$\mathcal{L}_\theta = \mathbb{E}_{t \sim U[0,1]} \mathbb{E}_\epsilon \left[\|\nabla U(x_t, t) - \Sigma_\theta(t)^{-1/2} \cdot \epsilon\|^2 \right].$$

This form of the KL divergence allows efficient gradient-based optimization. More details on the mathematical results can be found in the appendix.

References

- Onsager, L. and Machlup, S. Fluctuations and irreversible processes. *Phys. Rev.*, 91:1505–1512, Sep 1953. doi: 10.1103/PhysRev.91.1505. URL <https://link.aps.org/doi/10.1103/PhysRev.91.1505>.
- Särkkä, S. and Solin, A. Applied stochastic differential equations. *Cambridge University Press*, 2019.

A. Appendix

A.1. Discretization schemes

Suppose we have $\{\mathbf{x}_t\}_{t \in [0, T]}$ obeying SDE (1). We discretize the time interval $[0, T]$ with width Δt and $M := T/\Delta t$, which is taken to be an integer. Then the Itô's discretization scheme is given by

$$\int \mathbf{f} \cdot d\mathbf{x} := \int \mathbf{f}_t \cdot (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) = \lim_{\Delta t \rightarrow 0} \sum_{n=0}^{M-1} \mathbf{f}_{t_n} \cdot (\mathbf{x}_{t_n+\Delta t} - \mathbf{x}_{t_n}), \quad (7)$$

where $t_n := n\Delta t$ and $\mathbf{f}_t = \mathbf{f}(\mathbf{x}_t, t)$. The Stratonovich scheme is to take the midpoint,

$$\int \mathbf{f} \circ d\mathbf{x} := \int \frac{\mathbf{f}_t + \mathbf{f}_{t+\Delta t}}{2} \cdot (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) = \lim_{\Delta t \rightarrow 0} \sum_{n=0}^{M-1} \frac{\mathbf{f}_{t_n} + \mathbf{f}_{t_n+\Delta t}}{2} \cdot (\mathbf{x}_{t_n+\Delta t} - \mathbf{x}_{t_n}). \quad (8)$$

A.2. Path-probability

A.3. Rewriting Onsager-Machlup function

For the SDE (1), we have the Onsager-Machlup function

$$L(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \mathbf{f}(\mathbf{x}_t, t)\|^2}{2g(t)^2}. \quad (9)$$

Let $\mathbf{f}(\mathbf{x}_t, t) = \nabla \phi(\mathbf{x}_t, t)$. Then the total differential

$$d\phi(\mathbf{x}_t, t) = \frac{\partial \phi}{\partial t} dt + \nabla \phi \cdot d\mathbf{x} + \frac{1}{2} \nabla^2 \phi d\mathbf{x} \cdot d\mathbf{x} + O(dt^2). \quad (10)$$

Using the SDE (1) and Ito's lemma, we get

$$d\phi = \frac{\partial \phi}{\partial t} dt + \nabla \phi \cdot d\mathbf{x} + \frac{1}{2} g(t)^2 \nabla^2 \phi dt. \quad (11)$$

Now the integral

$$\begin{aligned} \int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t) dt &= \int_0^T \frac{\|\dot{\mathbf{x}}_t - \mathbf{f}(\mathbf{x}_t, t)\|^2}{2g(t)^2} dt \\ &= \int_0^T \frac{1}{2g(t)^2} (\dot{\mathbf{x}}_t^2 + \nabla \phi(\mathbf{x}_t, t)^2 - 2\dot{\mathbf{x}}_t \cdot \nabla \phi(\mathbf{x}_t, t)) dt \\ &= \int_0^T \frac{1}{2g(t)^2} (\dot{\mathbf{x}}_t^2 + \nabla \phi(\mathbf{x}_t, t)^2) dt - \int_0^T \frac{1}{g(t)^2} \nabla \phi(\mathbf{x}_t, t) \cdot \frac{d\mathbf{x}}{dt} dt \\ &= \int_0^T \frac{1}{2g(t)^2} (\dot{\mathbf{x}}_t^2 + \nabla \phi(\mathbf{x}_t, t)^2) dt - \int_0^T \frac{1}{g(t)^2} \left(\frac{d\phi}{dt} - \frac{\partial \phi}{\partial t} - \frac{1}{2} g(t)^2 \nabla^2 \phi \right) dt \\ &= \int_0^T \frac{1}{2g(t)^2} (\dot{\mathbf{x}}_t^2 + \nabla \phi(\mathbf{x}_t, t)^2 + g(t)^2 \nabla^2 \phi) dt - \int_0^T \frac{1}{g(t)^2} \left(\frac{d\phi}{dt} - \frac{\partial \phi}{\partial t} \right) dt \end{aligned} \quad (12)$$

We use the above form of the Onsager-Machlup function for the molecular dynamics Langevin equation (2). We have the mapping

$$\phi(\mathbf{x}_t) \longrightarrow -\frac{D_0}{k_B T} U(\mathbf{x}_t), \quad g(t) \longrightarrow \sqrt{2D_0}. \quad (13)$$

Therefore we get

$$\int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t) dt = \frac{U(\mathbf{x}_T) - U(\mathbf{x}_0)}{2k_B T} + \int_0^T \left(\frac{\dot{\mathbf{x}}_t^2}{4D_0} + V_{eff}(\mathbf{x}_t) \right) dt \quad (14)$$

where

$$V_{eff}(\mathbf{x}_t) = \frac{D_0}{4(k_B T)^2} ((\nabla U(\mathbf{x}_t))^2 - 2k_B T \nabla^2 U(\mathbf{x}_t)) \quad (15)$$

A.4. Simplifying KL divergence between path-probabilities

We want to evaluate

$$D_{\text{KL}}(Q\|P) = \int e^{-\tilde{S}} \log \frac{e^{-\tilde{S}}}{e^{-S}} [D\mathbf{x}_t] \quad (16)$$

where S is the action corresponding to the path-probability P and \tilde{S} is the action corresponding to the path-probability Q

$$S := \int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t) dt = \int_0^T \left[\frac{1}{2g(t)^2} \|\dot{\mathbf{x}}_t - \mathbf{f}(\mathbf{x}_t, t)\|^2 \right] dt, \quad (17)$$

and

$$\tilde{S} := \int_0^T \tilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t) dt = \int_0^T \left[\frac{1}{2g(t)^2} \|\dot{\mathbf{x}}_t - \tilde{\mathbf{f}}(\mathbf{x}_t, t)\|^2 \right] dt. \quad (18)$$

The KL divergence of the path-probability $P(\{\mathbf{x}_t\}_{t \in [0, T]})$ from $Q(\{\mathbf{x}_t\}_{t \in [0, T]})$ is written as

$$\begin{aligned} D_{\text{KL}}(Q(\{\mathbf{x}_t\}_{t \in [0, T]})\|P(\{\mathbf{x}_t\}_{t \in [0, T]})) &= \mathbb{E}_{Q(\{\mathbf{x}_t\})} \left[\ln \frac{Q(\{\mathbf{x}_t\}_{t \in [0, T]})}{P(\{\mathbf{x}_t\}_{t \in [0, T]})} \right] \\ &= \mathbb{E}_{Q(\{\mathbf{x}_t\})} [S - \tilde{S}]. \end{aligned} \quad (19)$$

This can be further simplified as

$$\begin{aligned} \mathbb{E}_{Q(\{\mathbf{x}_t\})} [S - \tilde{S}] &= \mathbb{E}_{Q(\{\mathbf{x}_t\})} \int_0^T \frac{1}{2g(t)^2} \left[-2\dot{\mathbf{x}}_t \cdot (\mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)) + (\mathbf{f}(\mathbf{x}_t, t))^2 - (\tilde{\mathbf{f}}(\mathbf{x}_t, t))^2 \right] dt \\ &= \mathbb{E}_{Q(\{\mathbf{x}_t\})} \int_0^T \frac{1}{2g(t)^2} [\mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)] \cdot [-2\dot{\mathbf{x}}_t + \mathbf{f}(\mathbf{x}_t, t) + \tilde{\mathbf{f}}(\mathbf{x}_t, t)] dt. \end{aligned} \quad (20)$$

We discretize this using Ito's scheme and look at the contribution from the neighboring part $(t, t + \Delta t)$,

$$\begin{aligned} &\mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t+\Delta t})} \left[\Delta t \frac{1}{2g(t)^2} (\mathbf{f}_t - \tilde{\mathbf{f}}_t) \cdot \left[-2 \frac{(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t)}{\Delta t} + \mathbf{f}_t + \tilde{\mathbf{f}}_t \right] \right] \\ &= \mathbb{E}_{q_t(\mathbf{x}_t)} \left[\Delta t \frac{1}{2g(t)^2} (\mathbf{f}_t - \tilde{\mathbf{f}}_t) \cdot (-2\tilde{\mathbf{f}}_t + \mathbf{f}_t + \tilde{\mathbf{f}}_t) \right] \\ &= \mathbb{E}_{q_t(\mathbf{x}_t)} \left[\Delta t \frac{1}{2g(t)^2} \|\mathbf{f}_t - \tilde{\mathbf{f}}_t\|^2 \right], \end{aligned} \quad (21)$$

Summing up these contributions for $[0, T]$, we have

$$\mathbb{E}_{Q(\{\mathbf{x}_t\})} [S - \tilde{S}] = \mathbb{E}_{Q(\{\mathbf{x}_t\})} \left[\int_0^T \frac{1}{2g(t)^2} \|\mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)\|^2 dt \right]. \quad (22)$$

Thus, we have obtained the equation,

$$\begin{aligned} D_{\text{KL}}(Q(\{\mathbf{x}_t\}_{t \in [0, T]})\|P(\{\mathbf{x}_t\}_{t \in [0, T]})) &= \mathbb{E}_{Q(\{\mathbf{x}_t\})} \left[\int_0^T \frac{1}{2g(t)^2} \|\mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)\|^2 dt \right] \\ &= \int_0^T \frac{1}{2g(t)^2} \mathbb{E}_{q_t} \|\mathbf{f}(\mathbf{x}_t, t) - \tilde{\mathbf{f}}(\mathbf{x}_t, t)\|^2 dt. \end{aligned} \quad (23)$$

A.5. Reparametrization of KL Divergence

The KL divergence is expressed as:

$$D_{\text{KL}}(Q \| P) = \frac{D_0}{4(k_B T)^2} \int_0^T \mathbb{E}_{q^{(\theta)}(t)} \left[\|\nabla U(x_t, t) - \nabla \tilde{U}(x_t, t)\|^2 \right] dt$$

We want to compute the gradient of this expression with respect to the parameters θ , which affect both $c_\theta(t)$ and $\Sigma_\theta(t)$. Using the reparameterization trick, we sample from $q^{(\theta)}(t) = \mathcal{N}(c_\theta(t), \Sigma_\theta(t))$ by expressing x_t as:

$$x_t = c_\theta(t) + \Sigma_\theta(t)^{1/2} \cdot \epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$. The potential $\tilde{U}(x)$ is given by:

$$\tilde{U}(x) = \frac{1}{2}(x - c_\theta(t))^\top \Sigma_\theta(t)^{-1}(x - c_\theta(t))$$

The gradient of $\tilde{U}(x)$ with respect to x is:

$$\nabla \tilde{U}(x) = \Sigma_\theta(t)^{-1}(x - c_\theta(t))$$

Substituting $x_t = c_\theta(t) + \Sigma_\theta(t)^{1/2} \cdot \epsilon$, we get:

$$\nabla \tilde{U}(x_t, t) = \Sigma_\theta(t)^{-1/2} \cdot \epsilon$$

The KL divergence expression becomes:

$$D_{KL}(Q \parallel P) = \frac{D_0}{4(k_B T)^2} \int_0^T \mathbb{E}_\epsilon \left[\|\nabla U(x_t, t) - \Sigma_\theta(t)^{-1/2} \cdot \epsilon\|^2 \right] dt$$

Algorithm 1 Training Neural Networks with Reparameterized KL Divergence and Cholesky Factor for Covariance Matrix

Input: Number of epochs N_{epochs} , learning rate η , batch size B , time interval $[0, T]$, initial point A , final point B

Output: Neural network parameters θ_c (for $c_\theta(t, A, B)$) and θ_L (for lower triangular matrix $L_\theta(t, A, B)$)

not converged **Sample** a batch of time points $\{t_1, t_2, \dots, t_B\}$, where $t_i \sim \mathcal{U}(0, T)$, and noise samples $\epsilon_{t_i}^{(j)} \sim \mathcal{N}(0, I)$ for $j = 1, \dots, N_\epsilon$. each time point t_i **Compute** the mean $c_\theta(t_i, A, B)$ and lower triangular matrix $L_\theta(t_i, A, B)$ from the neural networks:

$$c_\theta(t_i, A, B) = \text{MeanNet}(t_i, A, B)$$

$$L_\theta(t_i, A, B) = \text{CovNet}(t_i, A, B)$$

Apply Softplus to the diagonal entries of $L_\theta(t_i, A, B)$ to ensure they are positive:

$$L_{\theta,ii}(t_i, A, B) = \text{softplus}(L_{\theta,ii}(t_i, A, B)), \quad \forall i = 1, \dots, d$$

Construct the covariance matrix:

$$\Sigma_\theta(t_i, A, B) = L_\theta(t_i, A, B)L_\theta(t_i, A, B)^\top$$

Compute the inverse square root of the covariance matrix:

$$\Sigma_\theta^{-1/2}(t_i, A, B) = L_\theta(t_i, A, B)^{-\top}$$

each noise sample $\epsilon_{t_i}^{(j)}$ **Reparameterization trick:**

$$x_{t_i}^{(j)} = c_\theta(t_i, A, B) + L_\theta(t_i, A, B) \cdot \epsilon_{t_i}^{(j)}$$

Compute the true gradient $\nabla U(x_{t_i}^{(j)}, t_i)$ of the potential function. **Scale noise** by the inverse square root of the covariance matrix:

$$\Sigma_\theta^{-1/2}(t_i, A, B) \cdot \epsilon_{t_i}^{(j)}$$

Compute the loss for time point t_i as the average over noise samples:

$$\mathcal{L}_{t_i} = \frac{1}{N_\epsilon} \sum_{j=1}^{N_\epsilon} \left\| \nabla U(x_{t_i}^{(j)}, t_i) - \Sigma_\theta^{-1/2}(t_i, A, B) \cdot \epsilon_{t_i}^{(j)} \right\|^2$$

Compute the total loss as the average over time points:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^B \mathcal{L}_{t_i}$$

Update neural network parameters θ_c and θ_L using the optimizer:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$$

Return: Optimized parameters θ_c, θ_L
