Accelerated Sampling of Rare Events using a Neural Network Bias Potential

Xinru Hua
Department of Computer Science
Stanford University
huaxinru@stanford.edu

Rasool Ahmad
Department of Mechanical Engineering
Stanford University
rasool@stanford.edu

Jose Blanchet
Department of Management Science
and Engineering
Stanford University
jose.blanchet@stanford.edu

Wei Cai
Department of Mechanical Engineering
Stanford University
caiwei@stanford.edu

Abstract

In the field of computational physics and material science, the efficient sampling of rare events occurring at atomic scale is crucial. It aids in understanding mechanisms behind a wide range of important phenomena, including protein folding, conformal changes, chemical reactions and materials diffusion and deformation. Traditional simulation methods, such as Molecular Dynamics and Monte Carlo, often prove inefficient in capturing the timescale of these rare events by brute force. In this paper, we introduce a practical approach by combining the idea of importance sampling with deep neural networks (DNNs) that enhance the sampling of these rare events. In particular, we approximate the variance-free bias potential function with DNNs which is trained to maximize the probability of rare event transition under the importance potential function. This method is easily scalable to high-dimensional problems and provides robust statistical guarantees on the accuracy of the estimated probability of rare event transition. Furthermore, our algorithm can actively generate and learn from any successful samples, which is a novel improvement over existing methods. Using a 2D system as a test bed, we provide comparisons between results obtained from different training strategies, traditional Monte Carlo sampling and numerically solved optimal bias potential function under different temperatures. Our numerical results demonstrate the efficacy of the DNN-based importance sampling of rare events.

1 Introduction

Contemporary machine learning models suffer a substantial degradation in performance when confronted with long-tail events that are not represented well in collected data [1–3]. Several approaches, including the detection of out-of-distribution samples [4] and data resampling techniques [5, 6], can address this issue. In this paper, we propose a method that is specifically designed to efficiently sample these long-tail events, also referred to as rare events. Our method aims to efficiently collect rare events in simulation and increase their representation in datasets.

Specifically, in the domains of materials science and bio-chemistry, there is a pressing need to sample rare events that are associated with specific physical phenomena and estimate their probabilities. This is critical in advancing our understanding of various materials properties ranging from mechanical composites [7–9] to transport proteins [10], which are essential to aerospace and pharmaceutical

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

industries. The characterization of rare event transitions between two stationary metastable states has been a focus of a big body of research in the past [11–13], and continues to stimulate the present research undertakings. In the same vein, we seek to study rare event transitions occurring in atomic systems with probabilities as low as 10^{-6}. The molecules follow the Langevin dynamics governed by a potential energy function [14] which contains multiple minima corresponding to metastable states. During the dynamics, the molecular system spends most of the time performing random thermal motion near a metastable state before rarely escaping to the neighboring metastable state. Our goal is to sample such rare event transitions of the molecular system between two metastable states of the potential energy function. The traditional Monte Carlo sampling method is prohibitively expensive to capture such rare events and suffers from the curse of dimensionality in higher dimensions. Another line of research focuses on determining the committor function − the probability of making the rare event transition from a given molecular configuration − by solving a partial differential equation numerically with finite element method [15, 16]. However, such numerical determination of the committor function becomes exponentially difficult with the dimension of the problem. Thus, it is important to develop novel approaches to efficiently sample successful transitions and estimate the probability of rare events.

In this work, we present an approach that leverages deep neural networks (DNNs) to learn from and sample distributions of rare events in molecular dynamics. A holistic view of our method is presented in Fig. 1. The method of applying bias potential to enhance sampling rate is first introduced in [17, 18] and both works present promising results on 2D. The limitations of both methods lie in the construction of a trial importance function and the challenges with multiple degrees of freedom in higher dimensions. This work [19] first formulate the estimation of rare event probabilities as an optimization problem and employed importance sampling techniques to recover the unbiased probabilities. Our method adopts a similar optimization framework and provides a practical framework to train, test, and evaluate the sample quality. One novelty of our method is its ability to learn from past successful transitions, so that we can add humans in the loop to sketch possible transition paths, use collected transition paths in real-world experiments [20, 21] or we can remove partially the energy barrier first and then obtain some successful paths. The optimal DNN-based bias potential need to both minimize the KL divergence between the distribution of transition paths under the biased dynamics and unbiased dynamics and maximize the probability of rare events.

Our learning algorithm comprises three key steps: 1) The bias potential function is approximated by a DNN and the samples are generated by running the biased Langevin dynamics; 2) The DNN then learns to refine this sampled distribution and aligns it more closely to the unbiased distribution of rare event transitions; 3) After generating feasible rare events, we employ the importance sampling method to compute the actual unbiased probabilities of these rare events. Our paper makes several significant contributions:

C1 We introduce an efficient algorithm for training a DNN to approximate the variance-free importance/bias potential function. This trained DNN significantly improves the efficiency of our rare event sampling in molecular dynamics. Our method can effectively scale to higher-dimensional energy functions since it does not require data pre-collection or solving any equations.

C2 Our algorithm effectively learn from previously successfully sampled rare events which gives our algorithm the freedom and power to reuse and learn from any distribution of trajectories.

C3 Since we are sampling the molecule's trajectories, which exist in a space of more than 1000 dimensions, ensuring the statistical robustness of our estimator presents a significant challenge. To address this, we statistically assess the reliability of our estimator. This statistical measure enables us to compare the efficiency and accuracy of our estimator with other competing estimators in a fair and rigorous manner.

2 Related Work

Optimal Control and Importance Sampling: Several papers have delved into computing an optimal bias potential and utilizing importance sampling to estimate probability of rare events. Among them, a number of works focus on theoretical results on the optimal control and on transition paths [22, 19, 23], and the others provide numerical results [24–26]. The main advantage of our work is the statistical guarantees of our estimator and the ability to learn from successful transitions.

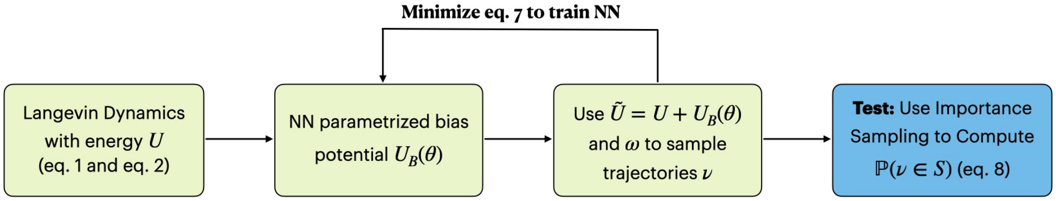
Minimize eq. 7 to train NN
Langevin Dynamics with energy U (eq. 1 and eq. 2)
NN parametrized bias potential $U_B(\theta)$
Use $\tilde{U} = U + U_B(\theta)$ and ω to sample trajectories ν
Test: Use Importance Sampling to Compute $\mathbb{P}(\nu \in S)$ (eq. 8)

Figure 1: Full details of training are described in Algorithm 1 in Sec. 4. Our method adopts a Reinforcement Learning approach: it samples trajectories and subsequently learns from them.

Additionally, our algorithm do not require data collection like in [25] and is scalable to higher dimension. In addition, [27] introduces the method to compute the bias potential to greatly enhance the rate of transitions, so that they can obtain the probability with much higher efficiency. However, their approach of computing the ground truth bias potential is restricted to dynamics on a grid. We have expanded upon this paper, adapting it to free dynamics in both 2D and higher dimensions.

Learning Committor Function: Another popular line of research involves using neural networks to approximate the committor function, which requires solving a partial differential equation with neural networks [28–31]. One notable limitation of this method is the significant growth in the size of the datasets and computational costs as the dimension grows. Specifically, it mandates the preparation of a dataset prior to the training. In contrast, our method can actively generate samples and learn from the samples simultaneously.

Traditional Sampling Method: There have been various sampling method to increase the efficiency of sampling low-probability events in molecular dynamics [32], for example, the army ants algorithm [33] and adaptive importance sampling method [17]. One shortcoming mentioned in [17] is difficulty of finding a suitable trial function for importance sampling that is successful in higher dimension and involving multiple degrees of freedom. Many books [34] and review papers give a comprehensive summary to all the methods.

3 Molecular Rare Event

We study a class of rare events associated with transition paths in molecular dynamics of chemical reaction networks. The dynamics follow overdamped Langevin dynamics [35, 14]:

$$d\mathbf{x} = -\nabla U(\mathbf{x})/(m\gamma) \cdot dt + \sqrt{2k_B T/(m\gamma)} \cdot dB(t).$$

Here $\mathbf{x} \in \mathbb{R}^d$, $k_B = 8.617 \times 10^{-5}$ is Boltzmann's constant, T is the absolute temperature, m is the mass of the particle, γ is the damping ratio, and $B(t)$ is a Brownian motion. For further simplification, we define $\epsilon = 2k_B T/(m\gamma)$. To introduce the discretized dynamics with timestep of Δt, we first let $\omega = (\delta_0, \delta_1, ..., \delta_N)$ to be a sequence of i.i.d noises, where $\delta_i \sim \mathcal{N}(0, \Delta t), \Delta t > 0$. As a result, the discretized dynamics follows the equation:

$$\Delta \mathbf{x}_t = -\nabla U(\mathbf{x}_{t-1})/(m\gamma) \cdot \Delta t + \sqrt{\epsilon} \cdot \delta_t. \tag{1}$$

In this work, we focus on the 2D domain, but our method can be extended to general higher dimensions. We let $m = 1, \gamma = 1$, and an energy function U on position (x, y), defined as

$$U(\mathbf{x}) = 0.05y + \frac{1}{6}\left(4(1 - x^2 - y^2)^2 + 2(x^2 - 2)^2 + [(x + y)^2 - 1]^2 + [(x - y)^2 - 1]^2 - 2\right). \tag{2}$$

Figure 1 illustrates the energy function that has two potential wells separated by a potential barrier. In this figure, A and B represent the two minima of the energy function [1]. Our objective is to calculate the probability of a particle starting near A and reaching a region close to B before a set deadline. When the temperature is low, escaping the energy well around A becomes highly challenging, resulting in a very low success rate.

We aim to calculate the probability that a particle starts near A, escapes the energy basin, and moves to point B within a predetermined number of steps, denoted as $N \in \mathbb{N}^+$. We consider the particle

[1] In the scenario where energy minima are unknown, we can use linear search and gradient descent method to determine energy minima.

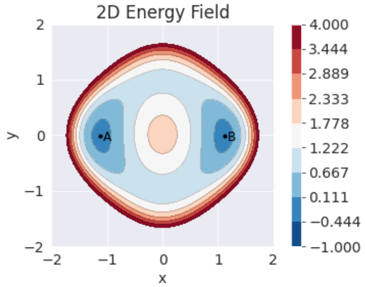

Figure 2: Plot of the 2D energy function in eq. 2. A and B represent the two minima of the energy.

to be close to B if its distance to B is within δ. The particle forms an N-step trajectory $\nu = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_N)$ as it follows the Langevin dynamics. In addition, we denote the step that the particle is closest to B as τ_B, that is

$$\tau_B = \min\{n : \|\mathbf{x}_n - B\| < \delta\}.$$

The simulation stops when the particle reaches B or the deadline arrives. In this way, the rare event is defined as:

$$S = \{\nu : \tau_B \leq \tau, \|\mathbf{x}_0 - A\| < \delta\}. \tag{3}$$

From the original Langevin dynamics in eq. 1, we aim to learn a neural network with parameters θ to parameterize the bias potential function so the modified dynamics equation becomes

$$\Delta \mathbf{x}(t) = -\nabla \left(U_B(\theta)(\mathbf{x}) + U(\mathbf{x})\right) \Delta t + \sqrt{\epsilon} \cdot \mathcal{N}(0, \Delta t). \tag{4}$$

The optimal modified dynamics significantly increases the probability of rare event S and the successful trajectories are similar to the trajectories sampled with the original potential. Lastly, we employ importance sampling to recover the probability of the rare event with the original energy function U.

Dimension of the problem: Although the molecule's movement is restricted to a 2D plane, we aim to sample its trajectories. For each step in these trajectories, its position (x and y) is recorded, resulting in a distribution with a dimensionality of $2N$, where N is the number of steps in a trajectory. In our experiments, $N = 500$, so the distribution is 1000-dimensional.

Choice on τ: Generally, the time taken for a transition to occur is of the order $\mathcal{O}\left(\exp\left(\frac{1}{\epsilon}\right)\right)$, but the particle spends a substantial proportion of this time around A without crossing the energy barrier. Thus, if we want the particle to start from A and move to B without going back to A, we can select a deadline $\tau = \mathcal{O}\left(\frac{1}{\epsilon^r}\right)$ for some suitably chosen r, so that the particle has sufficient time to travel to B. More precisely, we choose r such that $\tau > \mathbb{E}[\tau_B \mid \|\mathbf{x}_0 - A\| < \delta, \tau_B < \tau_A]$.

3.1 Variational Formulation of the Probability

In this section, we explain how we write the rare event probability as a variational formula and obtain an optimization problem to train the neural network. After we sample ω, we get our trajectory ν, so we write trajectory as $\nu(\omega)$. Our goal is to express the probability $P(\nu(\omega) \in S)$ as the solution to an optimization problem. To do this, we introduce a function:

$$F(\nu(\omega)) = \begin{cases} 0, & \text{when } \nu \in S \\ \infty, & \text{otherwise} \end{cases}$$

so

$$\exp(-F(\nu(\omega))) = \begin{cases} 1, & \text{when } \nu \in S \\ 0, & \text{otherwise} \end{cases}$$

With this, we can calculate $P(\nu(\omega) \in S) = \mathbb{E}[\exp(-F(\nu(\omega)))]$. Suppose Q is the new probability of trajectories that we sample ν with the modified potential function. Then, using the Jensen's inequality same as in [19], we can write the probability into a variational form:

$$\log(\mathbb{P}(\nu \in S)) = -\inf\Big\{\mathbb{E}_Q\left[\log\left(\frac{dQ(\nu)}{dP(\nu)}\right)\right] + \mathbb{E}_Q(F(\nu))\Big\}. \tag{5}$$

We need to ensure that $dQ(x) = 0$ if and only if $dP(x) = 0$, and it is satisfied as they are both Gaussians. Here, $dP(\nu)$ denotes the probability distribution of trajectories controlled by the original energy function, and $dQ(\nu)$ represents the distribution with the modified energy function. Then, [19, Thm. 1] proves that the optimal solution to (5) leads to a variation-free estimator for $\mathbb{P}(\nu \in S)$.

3.2 Smooth Indicator Function

From the definition of F, to minimize (5), we aim for every sampled ν to belong to S, so that $\mathbb{E}_Q(F) = 0$. To make the optimization approachable in practice, we replace F by a smooth function: $F_{smooth}(\nu) = s \cdot \tanh\left(\|\nu(N) - B\|^2 - (r + 0.02)^2\right)$, where $\nu(N) = \mathbf{x}_N$, and s and r are both parameters. This function reaches its minimum at point B, increases smoothly as x moves away from B, and remains constant when $|x - B| > r + 0.02$. In our training scheme, r decreases from 1.0 to 0.05 as we train the neural network, so trajectories need to move closer and closer to B.

3.3 Likelihood Ratio Computation

We sample molecular trajectories by sampling ω and use the discretized dynamics model $\Delta \mathbf{x}_t = -\nabla \tilde{U}(\mathbf{x}_{t-1}) \cdot \Delta t + \sqrt{\epsilon} \cdot \delta_t$ with the modified potential function $\tilde{U} = U + U_B(\theta)$ parameterized by a neural network θ. At each step of the dynamics, we sample the random noise δ_t from a Gaussian distribution. In this way, the probability to model δ_i is the probability density functions (PDF) of the normal distribution, and we can write the likelihood of ω with respect to the Lebesgue measure as $p(\omega)$, which is a product of N Gaussian PDFs. Since the randomness of ν is fully described by that of ω, the likelihood of the trajectory ν equals to that of the corresponding ω.

In eq. 5, P, Q represent the distribution of molecular trajectories under two energy functions: $U, \tilde{U}$. To compute $\frac{dQ}{dP}$, we first need to compute the corresponding ω_P and then compute the likelihood ratio $\frac{dQ}{dP} = \frac{p(\omega)}{p(\omega_P)}$. For a trajectory $\nu = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_N)$, we first compute the Gaussian noises at each step as if it is generated by the original potential function U:

$$\delta'_t = \frac{1}{\sqrt{\epsilon}}(\mathbf{x}_t - \mathbf{x}_{t-1} + \nabla U(\mathbf{x}_{t-1})\Delta t)$$

Then, the likelihood ratio of the trajectory is

$$\log\left(\frac{dQ(\nu)}{dP(\nu)}\right) = \log\left(\frac{p(\omega)}{p(\omega_P)}\right) = \log\left(\frac{p(\delta_N)p(\delta_{N-1})...}{p(\delta'_N)p(\delta'_{N-1})...}\right) = \frac{1}{2\epsilon\, dt}\sum_{t=1}^{N}\left(\delta'^2_t - \delta^2_t\right) \tag{6}$$

In practice, due to the time deadline and the fact that we smooth the indicator function, the bias potential may not always ensure that every trajectory successfully reaches point B. The final optimization problem then becomes:

$$\inf_{\theta} \mathbb{E}_Q\left[\log\left(\frac{dQ(\nu_\theta(\omega)}{dP(\nu_\theta(\omega))}\right)\mathbb{1}(\nu \in S) + F_{\text{smooth}}\left(\nu_\theta(\omega)\right)\right], \tag{7}$$

where Q is the probability distribution of ν that is driven by the modified energy function $U + U_B(\theta)$.

4 Algorithm

Using the bias potential, we want to see the likelihood of rare events significantly increases, and the distribution of successful trajectories remains unchanged. In our experiments, we use the energy function 1 as the loss function and batch stochastic gradient descent method [36] to train our DNN. After many simulation runs, the bias potential U_B allow us to sample from a distribution of trajectories Q that minimizes the objective function (7). Algorithm 1 describes the training process. In line 12, we minimize two components in the loss function: the first part is the KL divergence between Q and P, which tries to match the distribution of successful trajectories under the modified and the original energy, and the second part penalizes unsuccessful trajectories.

After training the neural network, we employ importance sampling to compute the rare event's probability under the original distribution P. We run simulations using the biased dynamics as described

Algorithm 1 Train a DNN as a bias potential function to sample from $P\left(\nu | \nu \in S\right)$

Input: Two minima A and B, learning rate α, starting position x_0, original potential U, threshold δ, time step Δt, number of simulations M, number of timesteps N.

Output: A bias potential function U_B parameterized by neural network $\theta : \mathbb{R}^2 \to \mathbb{R}$

```
for $i = 1, 2, \ldots, M$ do
    for $t = 1, \ldots, N$ do
        $x_t = x_{t-1} - \nabla(U + U_B(\theta))(x_{t-1})\Delta t + \sqrt{\epsilon}\mathcal{N}(0, \Delta t)$.    ▷ Parallelized over a batch of samples.
        for each unfinished trajectory in the batch do
            if $\|x_t - B\| < \delta$ then
                Compute the KL loss in eq. 7: $L_{\text{KL}} = \mathbb{E}_Q\left[\log\left(\frac{dQ(\nu)}{dP(\nu)}\right)\right]$.
                Mark this trajectory as finished.
            end if
        end for
    end for
    Compute the $F$_smooth in eq. 7 as $L_{\text{smooth}}$ and $L_{\text{total}} = L_{\text{KL}} + L_{\text{smooth}}$.
    Run gradient descent: $\theta = \theta - \alpha\nabla_\theta L_{\text{total}}$.
end for
return $v_\theta$
```

in eq. 3. As in eq. 7, we define Q to only model the successful trajectories. After computing ω_p, the probability of $\nu \in S$ under P is given by:

$$\mathbb{P}(\nu \in S) = \mathbb{E}_Q\left[\frac{dP(\nu)}{dQ(\nu)}\right] = \frac{1}{M}\sum_i \frac{p(\omega_{p,i})}{p(\omega_i)}\mathbb{1}(\nu_i(\omega_i) \in S). \tag{8}$$

Here P represents the distribution of ν under the original potential and Q is the distribution of ν under the modified potential. The density ratio $\frac{p(\omega_p)}{p(\omega)}$ can be computed analogously to eq. 6.

To achieve an efficient estimation, we want the density ratio of successful trajectories $W_i = \frac{dP(\nu)}{dQ(\nu)}$ to be at the same scale with every sample, so that the estimation is not described by a limited subset of samples. Therefore, we use a metric called Effective Sample Size (ESS) [37] and coefficient of variance (CV) to measure the efficiency of importance sampling:

$$\text{ESS} = \frac{(\sum_{i:\nu_i \in S} W_i)^2}{\sum_{i:\nu_i \in S} W_i^2}, \quad CV = \frac{\sigma}{\mu}, \text{where } \sigma \text{ is the sample standard deviation.}$$

Additionally, we utilize the confidence interval and standard deviation of the estimation to measure the reliability of our estimators.

4.1 Learn from Existing Successful Trajectories

In the case of higher dimensions or more transition channels, it will be extremely difficult to successfully sample trajectories in all the transition channels in one simulation run. One great advantage of our algorithm is that it is adapted to learn a bias potential from a dataset of successful paths. With this functionality, we can obtain some successful paths with any method first and then train a possible bias potential to speed up further sampling: (1) experts in material science can sketch possible transition paths based on their understanding, (2) we can remove the energy barrier partially and then obtain some successful paths, (3) we can also obtain successful paths from mechanical experiments. Curating a balanced dataset of successful paths helps the neural network learn to generate all modes of trajectories and be less biased. It also greatly speeds up the process of training a neural network.

In the 2D problem, there are two channels that the molecule can transit through: going up and going down, like plotted in Fig. 3. Since our energy function is asymmetric on y, it is more likely to escape the energy basin through the bottom channel. During training, sometimes the neural networks converge to local minima where they generate trajectories only going through one channel, so combining the two modes and learning from all trajectories is useful.

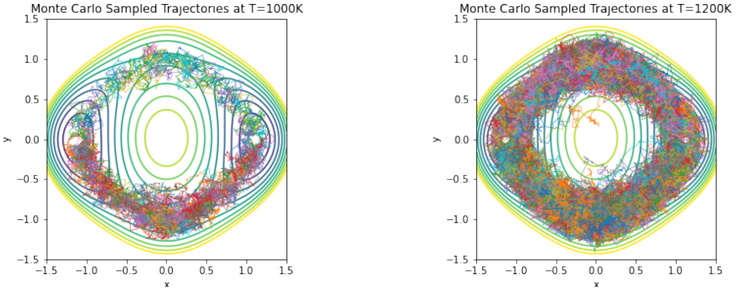

Figure 3: Trajectories at temperature 1000K and 1200K obtained by traditional Monte Carlo sampling method. The molecules escape at a higher rate at the elevated temperature of 1200K. However, the two channels of transition are more precise and refined at the lower temperature.

We minimize the KL divergence between the probability of the existing paths under the original potential (p) and the probability of these existing paths under the bias potential that is being learned (q). We continue using the existing paths and the neural network does not generate new paths. If we have a collection of successful paths: $\{\nu_i, i = 1...n\}$, we can solve:

$$\min_{\theta} \sum_{i} \log\left(\frac{q_\theta(\nu_i)}{p(\nu_i)} q_\theta(\nu_i)\right) \tag{9}$$

5 Results

The neural network architecture consists of 4 hidden layers and $\tanh$ as the activation function (except the output layer). It maps a molecular position to the bias energy: $\mathbb{R}^2 \to \mathbb{R}$. We find that adding two control variables $\exp(\|x - A\|_2)$ and $\exp(\|x - B\|_2)$ allows the DNN to better utilize the information of the distance between x and A or B. The project is implemented using the PyTorch library. The gradient of bias energy $\nabla U_B(\mathbf{x})$ is computed by the PyTorch autograd module. All the experiments are performed on one AWS c5.4xlarge instance.

Our method is a generative approach, so it does not require a standard train/val/test split. Every trajectory is sampled with random Gaussian noises. During the test, we freeze the DNN-based bias potential and use it in eq. (3) to sample trajectories. We train our method with a batch size of 512 and train 300 steps. The comparisons of the confidence interval, success rate, ESS, and time consumption under two temperature settings are demonstrated in Table 1 and Table 2. We use a statistical significance at a P-value of 0.05 throughout the tests. We showcase two functionalities of our method:

A: Exploration We train a DNN from scratch following Alg. 1 with the loss function (7). We obtain the ground truth bias potential $U_B^{gt}(\mathbf{x}) = -2K_BT\log(q(\mathbf{x}))$ by numerically solving the committor function with finite element method (FEM) [15] from the following partial differential equation (PDE) [16, 38]:

$$\nabla \cdot \left(e^{-\beta U(x)} q(x)\right) = 0, \text{in } D \setminus (A \cup B), \quad q(x)|_{\partial A} = 0, \quad q(x)|_{\partial B} = 1. \tag{10}$$

In higher dimensions, solving the committor function from this PDE becomes infeasible [28]. Thus, it is critical to develop other methods that does not require solving the PDE with FEM or other numerical methods in higher dimensions. A comparison between our method's bias potential and the ground truth bias potential is in Fig. 4.

B: Combine Given successfully sampled trajectories, we combine all the past knowledge into one DNN with the loss function in eq. 9. The trajectories can be from other DNNs, the Monte Carlo method, or human experts. We only need the positions of the trajectories, so that we can combine the knowledge into one bias potential. The pipeline and the results are shown in Fig. 5.

5.1 Robustness of Estimator

In this section, we plot the scaling behavior of our estimator, focusing on variance, ESS, and computational time as functions of the number of test samples in Fig. 6. Our findings reveal that our

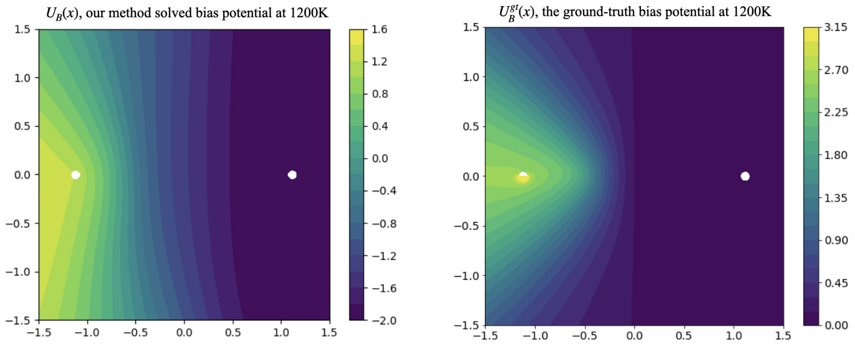

Figure 4: **Left:** Bias potential functions generated by our method in mode A. **Right:** Bias potential generated by the PDE's numerical solution. The shapes of two functions are very similar, although our optimization process does not involve solving the PDE.

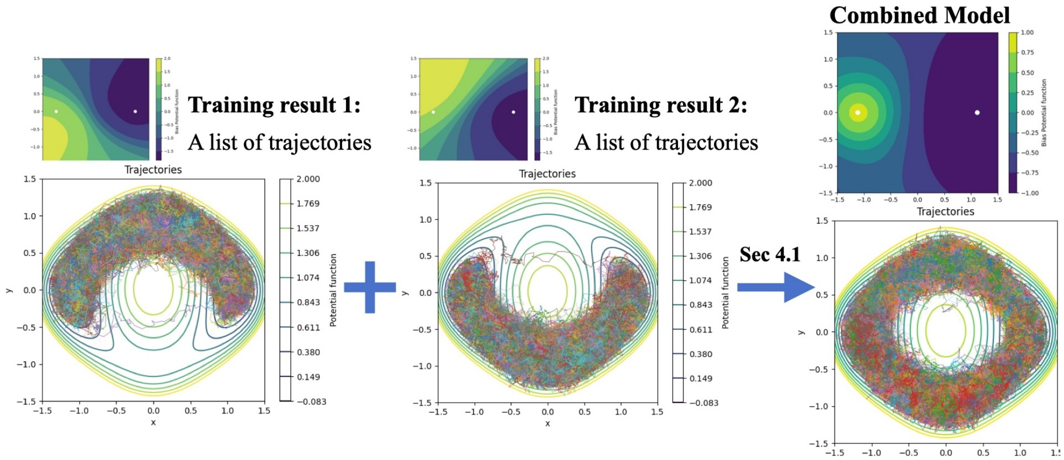

Figure 5: Mode B of our method at temperature 1200K. Training in mode A sometimes suffers from mode collapse as other generative methods. In this case, we only need the positions of the successful paths, and we minimize (9) to train a new bias potential that can sample paths that are similar to all the past training examples. The sampled trajectories are also similar to the Monte Carlo's in Fig. 3.

Table 1: Comparison between our method A, B and Monte Carlo method under 1200K. We test our method with 5120 examples and the Monte Carlo method with 10^8 samples. Even though our method does not directly minimize the variance of the estimator, the variance from our method A is smaller than Monte Carlo. ESS is not applied to Monte Carlo. All computational times recorded are measured in minutes. Combining training and testing, our method A achieves a 5.13x speedup over naive Monte Carlo and our method B achieves 2.5x speedup.
Note: Our training time is an upfront cost and does not grow as the number of samples grows. If only comparing the test time, our method offers a 44x speedup.

Temperature 1200K	Confidence Interval	CV	Success Rate	ESS ratio	t_{train}	t_{test}
Our method A	$4.037 \pm 0.342\text{e}{-6}$	3.0933	0.770	0.095	122	16
Our method B	$3.232 \pm 0.743\text{e}{-5}$	8.402	0.396	0.014	14	16
Monte Carlo	$4.410 \pm 0.412\text{e}{-6}$	1505.846	4.410e−6	-	-	708

estimator mirrors the trend observed in Monte Carlo methods. Specifically, as the number of samples escalates, there is a reduction in variance coupled with an enhancement in ESS. Notably, the testing time increases slower than linearly, due to the considerable overhead associated with initializing and loading the neural network bias potential. This underscores the robustness and reliability of our approach in comparison to traditional Monte Carlo.

Table 2: Similar to Table 1, we compare the results between different methods. Our method A also has smaller variance and 4.4x speed up compared to naive Monte Carlo.

Temperature 1000K	Confidence Interval	CV	Success Rate	ESS ratio	t_{train}	t_{test}
Our method A	$3.433 \pm 0.285\text{e}{-7}$	3.032	0.819	0.098	132	15
Our method B	$4.993 \pm 0.759\text{e}{-6}$	8.093	0.396	0.015	10	16
Monte Carlo	$3.600 \pm 1.176\text{e}{-7}$	5270.463	$3.600\text{e}{-7}$	-	-	646

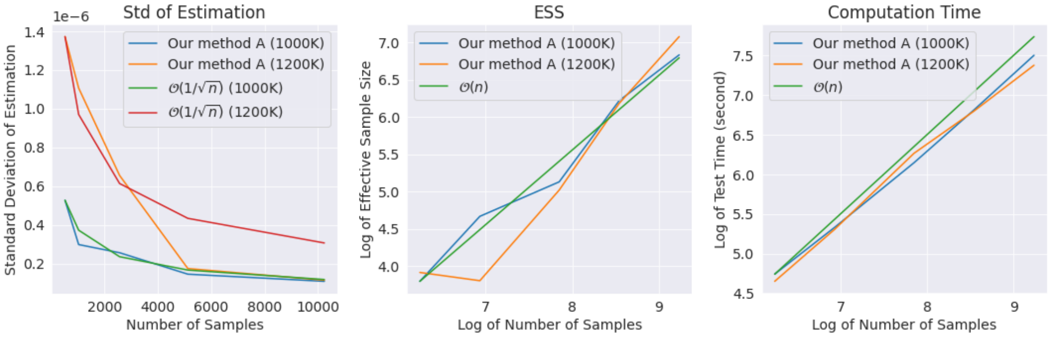

Figure 6: **Left:** Our method achieves $\mathcal{O}(1/\sqrt{n})$ reduction in the standard deviation of population mean, the same rate as naive Monte Carlo. Here the numbers for 1000K are multiplied with 10 to be at the same scale as the numbers of 1200K. **Middle:** The ESS increases in the order of $\mathcal{O}(n)$, which means the effective sample size is proportional to the sample size, with the proportionality constant being stable. **Right:** Our method's computation time exhibits a growth rate of $\mathcal{O}(n)$.

6 Conclusion

In this work, we focus on challenges of efficiently sampling rare events in molecular dynamics, which underpins the technologically important properties of materials and molecules. We propose to utilize deep neural networks as a bias/importnace potential function. We successfully formulate the probability as an optimization problem which is used to train the neural network. Our proposed approach can be scaled to high-dimensional cases and also provides robust statistical guarantees on the accuracy of the estimated probabilities. The ability of our algorithm to learn from successful samples makes our method versatile and marks an improvement over existing methodologies. We compare our method with traditional Monte Carlo sampling and numerical FEM method under different temperatures to measure the efficacy of our approach. Our estimator exhibits a smaller variance than the traditional Monte Carlo and achieves about 5x speedup comparing the training and test time combined, and more than 44x speedup if only comparing the test time. We also test the robustness and scalability of our bias potential. With more test samples increasing, the variance decreases in the order of $\mathcal{O}(1/\sqrt{n})$, and effective sample size and time increase in the order of $\mathcal{O}(n)$.

The immediate future direction is applying our method to higher dimensional models, similar to [39]. In higher dimensions, we anticipate our method to offer even greater speed advantages over the naive Monte Carlo approach, which suffers from the curse of dimensionality. Another intriguing avenue to explore is the potential for training neural networks using existing molecular dynamics datasets, as described in [40, 21]. One limitation of our method lies in the variance of likelihood ratio of importance sampling. We plan to test the stability and sensitivity of the importance sampling method by adding artificial noises to the bias potential and measuring how much the results change. We will also compare against another line of research [28, 31], which solves the committor function and tries to use it to enhance the sampling rate.

7 Acknowledgement

We would like to express our gratitude to the anonymous reviewers for their insightful comments. Our appreciation extends to Professor Masha (Maria) K. Cameron for sharing her FEM code with us, and to Jiaxin (Margot) Yuan for valuable discussions. We are also thankful to Dr. Tianyi Shi for meticulously proofreading this manuscript and for meaningful discussions. The research is supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0397. Additional support is gratefully acknowledged from NSF 1915967, 2118199, 2229012, 2312204.

References

[1] Y. Fu, L. Xiang, Y. Zahid, G. Ding, T. Mei, Q. Shen, and J. Han, "Long-tailed visual recognition with deep models: A methodological survey and evaluation," *Neurocomputing*, vol. 509, pp. 290–309, 2022.

[2] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10795–10816, 2023.

[3] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 2881–2891, Curran Associates, Inc., 2020.

[4] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[6] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249–259, 2018.

[7] R. Harada, R. Morita, and Y. Shigeta, "Free-energy profiles for membrane permeation of compounds calculated using rare-event sampling methods," *Journal of Chemical Information and Modeling*, vol. 63, pp. 259–269, 01 2023.

[8] C. W. Jang, J. W. Mullinax, and J. W. Lawson, "Mechanical properties and failure of aerospace-grade epoxy resins from reactive molecular dynamics simulations with nanoscale defects," *ACS Applied Polymer Materials*, vol. 4, no. 8, pp. 5269–5274, 2022.

[9] M. F. C. N. A. Marks and C. Kocer, "The importance of rare events in thin film deposition: a molecular dynamics study of tetrahedral amorphous carbon," *Molecular Simulation*, vol. 32, no. 15, pp. 1271–1277, 2006.

[10] R. Elber and M. Karplus, "Multiple conformational states of proteins: A molecular dynamics analysis of myoglobin," *Science*, vol. 235, no. 4786, pp. 318–321, 1987.

[11] E. Vanden-Eijnden *et al.*, "Towards a theory of transition paths," *Journal of statistical physics*, vol. 123, no. 3, pp. 503–523, 2006.

[12] E. Vanden-Eijnden *et al.*, "Transition-path theory and path-finding algorithms for the study of rare events.," *Annual review of physical chemistry*, vol. 61, pp. 391–420, 2010.

[13] R. E. Gillilan and K. R. Wilson, "Shadowing, rare events, and rubber bands. a variational verlet algorithm for molecular dynamics," *The Journal of chemical physics*, vol. 97, no. 3, pp. 1757–1772, 1992.

[14] T. Schlick, *Molecular modeling and simulation: an interdisciplinary guide*, vol. 2. Springer, 2010.

[15] J. N. Reddy, *Introduction to the finite element method*. McGraw-Hill Education, 2019.

[16] J. Alberty, C. Carstensen, and S. A. Funken, "Remarks around 50 lines of matlab: short finite element implementation," *Numerical algorithms*, vol. 20, no. 2-3, pp. 117–137, 1999.

[17] M. de Koning, W. Cai, B. Sadigh, T. Oppelstrup, M. H. Kalos, and V. V. Bulatov, "Adaptive importance sampling monte carlo simulation of rare transition events," *The Journal of chemical physics*, vol. 122, no. 7, 2005.

[18] W. Cai, M. H. Kalos, M. de Koning, and V. V. Bulatov, "Importance sampling of rare transition events in markov processes," *Physical Review E*, vol. 66, no. 4, p. 046703, 2002.

[19] C. Hartmann, L. Richter, C. Schütte, and W. Zhang, "Variational characterization of free energy: theory and algorithms," *Entropy*, vol. 19, no. 11, p. 626, 2017.

[20] M. Karplus and G. A. Petsko, "Molecular dynamics simulations in biology," *Nature*, vol. 347, no. 6294, pp. 631–639, 1990.

[21] D. B. Korlepara, C. S. Vasavi, S. Jeurkar, P. K. Pal, S. Roy, S. Mehta, S. Sharma, V. Kumar, C. Muvva, B. Sridharan, A. Garg, R. Modee, A. P. Bhati, D. Nayar, and U. D. Priyakumar, "Plas-5k: Dataset of protein-ligand affinities from molecular dynamics for machine learning applications," *Scientific Data*, vol. 9, no. 1, p. 548, 2022.

[22] Y. Gao, T. Li, X. Li, and J.-G. Liu, "Transition path theory for langevin dynamics on manifolds: Optimal control and data-driven solver," *Multiscale Modeling & Simulation*, vol. 21, no. 1, pp. 1–33, 2023.

[23] C. Hartmann and C. Schütte, "Efficient rare event simulation by optimal nonequilibrium forcing," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 11, p. P11004, 2012.

[24] M. Chak, T. Lelièvre, G. Stoltz, and U. Vaes, "Optimal importance sampling for overdamped langevin dynamics," 2023.

[25] L. Zhang, H. Wang, *et al.*, "Reinforced dynamics for enhanced sampling in large atomic and molecular systems," *The Journal of chemical physics*, vol. 148, no. 12, 2018.

[26] D. Passerone and M. Parrinello, "Action-derived molecular dynamics in the study of rare events," *Physical Review Letters*, vol. 87, no. 10, p. 108302, 2001.

[27] W. Cai, M. H. Kalos, M. de Koning, and V. V. Bulatov, "Importance sampling of rare transition events in markov processes," *Phys. Rev. E*, vol. 66, p. 046703, Oct 2002.

[28] Y. Khoo, J. Lu, and L. Ying, "Solving for high-dimensional committor functions using artificial neural networks," *Research in the Mathematical Sciences*, vol. 6, pp. 1–13, 2019.

[29] Q. Li, B. Lin, and W. Ren, "Computing committor functions for the study of rare events using deep learning," *The Journal of Chemical Physics*, vol. 151, no. 5, 2019.

[30] J. Yuan, A. Shah, C. Bentz, and M. Cameron, "Optimal control for sampling the transition path process and estimating rates," 2023.

[31] H. Li, Y. Khoo, Y. Ren, and L. Ying, "A semigroup method for high dimensional committor functions based on neural network," in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference* (J. Bruna, J. Hesthaven, and L. Zdeborova, eds.), vol. 145 of *Proceedings of Machine Learning Research*, pp. 598–618, PMLR, 16–19 Aug 2022.

[32] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

[33] S. Nangia, A. W. Jasper, T. F. Miller III, and D. G. Truhlar, "Army ants algorithm for rare event sampling of delocalized nonadiabatic transitions by trajectory surface hopping and the estimation of sampling errors by the bootstrap method," *The Journal of chemical physics*, vol. 120, no. 8, pp. 3586–3597, 2004.

[34] I. Prigogine and S. A. Rice, *Advances in chemical physics*, vol. 250. John Wiley & Sons, 2009.

[35] R. Pastor, "Techniques and applications of langevin dynamics simulations," in *The Molecular Dynamics of Liquid Crystals*, pp. 85–138, Springer, 1994.

[36] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[37] L. Martino, V. Elvira, and F. Louzada, "Effective sample size for importance sampling based on discrepancy measures," *Signal Processing*, vol. 131, pp. 386–401, 2017.

[38] P.-O. Persson and G. Strang, "A simple mesh generator in matlab," *SIAM review*, vol. 46, no. 2, pp. 329–345, 2004.

[39] A. F. Voter, "Hyperdynamics: Accelerated molecular dynamics of infrequent events," *Physical Review Letters*, vol. 78, no. 20, p. 3908, 1997.

[40] M. Frassek, A. Arjun, and P. Bolhuis, "An extended autoencoder model for reaction coordinate discovery in rare event molecular dynamics datasets," *The Journal of Chemical Physics*, vol. 155, no. 6, 2021.