A dissertation submitted for the degree of Master of Science in

SCIENTIFIC AND DATA INTENSIVE COMPUTING

UNIVERSITY COLLEGE LONDON

Deep Reinforcement Learning for Rare-Event Sampling in Molecular Dynamics

By
XIAOYU WANG

First Supervisor:Prof.Edina Rosta
Second Supervisor:Dr.Wenhao Deng

Department of Physics and Astronomy
UNIVERSITY COLLEGE LONDON
Date of submission: August 19, 2025

DECLARATION

I, Xiaoyu Wang confirm that the work presented in this report is my own. Where information has been derived from other sources, I confirm that this has been indicated in the report.

_____

Xiaoyu Wang

ABSTRACT

Molecular dynamics (MD) simulation enables the tracking of particle trajectories at the atomic scale and serves as a powerful tool for investigating the structural and dynamical properties of molecular systems. The mean first-passage time (MFPT), as a key statistical descriptor of dynamical processes, quantitatively characterizes the expected time for a system to transition from an initial state to a target state. However, in complex systems, the presence of high energy barriers and rare events renders conventional sampling methods inefficient within feasible simulation timescales, thereby limiting their applicability in kinetic modeling and rate estimation.To address this challenge, we propose an adaptive bias optimization framework based on deep reinforcement learning (DRL), aiming to enhance the sampling efficiency of rare events by minimizing MFPT. The proposed method employs the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, which allows the agent to dynamically learn optimal biasing potentials from trajectory history and efficiently guide the system across energy barriers. Specifically, a Markov State Model (MSM) is first utilized to discretize the state space and estimate state-to-state transition probabilities, enabling accurate modeling of system kinetics. Subsequently, the Dynamic Histogram Analysis Method (DHAM) is applied to recover unbiased dynamical information from the biased trajectories. Numerical experiments demonstrate that the proposed method achieves stable policy convergence and significantly accelerates sampling across a range of representative model potential systems. The observed reduction in average relaxation time validates the potential of DRL-based adaptive sampling as a generalizable and efficient approach for kinetic modeling in complex molecular systems.

Keywords:TD3 algorithm, molecular dynamics simulations, MFPT, reinforcement learning, adaptive biasing

Code: `https://github.com/Allen6848/dissertation_code`

ACKNOWLEDGEMENTS

I extend my deepest gratitude to my supervisor, Professor.Edina Rosta, for her exceptional mentorship, unwavering support, and profound expertise throughout this journey. Her insightful guidance, patience, and intellectual rigor not only shaped this work but also inspired my growth as a scholar. I am truly fortunate to have learned from her.

To my beloved parents, thank you for your endless sacrifices, encouragement, and unconditional love. Your belief in me has been my anchor, giving me the strength and courage to pursue my dreams.

I am also deeply grateful to my classmates for their camaraderie, stimulating discussions, and shared dedication to learning. Additionally, I sincerely thank all faculty members whose passion for teaching and commitment to excellence enriched my academic experience and broadened my perspective.

This achievement would not have been possible without the collective support of these remarkable individuals.

ACRONYMS

DHAM Dynamic Histogram Analysis Method.

DQN Deep Q-Network.

DRL Deep Reinforcement Learning.

FES Free Energy Surface.

MD Molecular Dynamics.

MFPT Mean First Passage Time.

MSM Markov State Model.

PPO Proximal Policy Optimization.

RL Reinforcement Learning.

TD3 Twin Delayed Deep Deterministic Policy Gradient.

US Ultrasound.

CONTENTS

Declaration **1**

Abstract **2**

Acknowledgement **3**

List of Figures **6**

1 Introduction **7**
1.1 Background 7
1.2 Molecular Dynamics simulation 8
1.3 Mean First Passage Time 9
1.4 Deep Reinforcement Learning 9

2 Materials and Methods **13**
2.1 Problem Formulation 13
2.2 TD3 Algorithm 15
2.2.1 TD3 Algorithm Description 15
2.2.2 The Workflow of TD3 16
2.2.2.1 Initialization Phase 16
2.2.2.2 Training Procedure 16
2.2.2.3 Policy Update 17
2.3 Model MD as MDP 17

3 Results **20**
3.1 Experiments 20
3.1.1 Parameter Settings 20
3.1.2 Representative Episode Evaluation on TD3 algortihm 20
3.1.3 Comparative Experimental Analysis 23

4 Discussion **26**
4.1 Discussion 26

5 Conclusion and Future Outlook **27**
5.1 Conclusion 27
5.2 Future Outlook 27

References **28**

LIST OF FIGURES

1.1 Framework of Deep Reinforcement Learning. 10

2.1 A schematic representation of a particle undergoing rare-event transition along a one-dimensional potential energy surface. The particle starts in the initial state (left well) and attempts to reach the target region (shaded area) by crossing an energy barrier. A time-dependent bias potential $U_{\text{bias}}(x,t)$ (red dashed curve) is applied to reshape the energy landscape and accelerate the transition, thereby reducing the MFPT..... 14

2.2 Architecture of the TD3 algorithm (Twin Delayed DDPG). The online actor π_θ outputs an action $a = \pi_\theta(s) + \mathcal{N}$ for exploration, and the transition (s, a, r, s') from the environment (ENV) is stored in the replay buffer. Two critics Q_{ϕ_1} and Q_{ϕ_2} are trained using the clipped double-Q target $y = r + \gamma \min_{i\in\{1,2\}} Q_{\phi_i^-}\big(s',\, \pi_{\theta^-}(s') + \epsilon\big)$ with target policy smoothing noise $\epsilon \sim \text{clip}(\mathcal{N}(0,\sigma), -c, c)$. Target networks $(\pi_{\theta^-}, Q_{\phi_1^-}, Q_{\phi_2^-})$ are updated by Polyak averaging, while the actor is updated less frequently than the critics (delayed policy update). The dashed block indicates the optimization pipeline with the twin critics and their targets..... 15

3.1 Representative particle trajectories over multiple episodes. The plots show distinct dynamical transitions from the initial state toward the target metastable basin, guided by the learned reinforcement learning policy..... 21

3.2 Total potential energy profile of the system. The profile shows two prominent wells and a transition barrier, characterizing stable and metastable states. 22

3.3 Evolution of Gaussian bias parameters during training. (a) Amplitude a; (b) Bias center b; (c) Width c. The trends reflect the agent's adaptation toward stable and localized biasing strategies..... 24

3.4 Comparison of particle trajectories and total potential energy profiles for A2C and DQN algorithms. 25

1. INTRODUCTION

1.1. BACKGROUND

Molecular dynamics (MD) simulation is a significant computational method used to investigate the structural and dynamical evolution of matter at atomic and molecular scales. Currently, MD has been widely applied across various disciplines, including physics, chemistry, and materials science. By numerically solving the microscopic equations of motion under predefined potential energy functions, MD simulations are capable of uncovering key processes such as energy transfer, conformational changes, phase behavior, and reaction pathways.[1] In addition, for complex systems characterized by high energy barriers and rare events, the mean first-passage time (MFPT) serves as a critical statistical metric. It quantitatively describes the expected time required for a system to transition from an initial configuration to a target state, thereby providing a theoretical foundation for analyzing the kinetic behavior of such systems.

Traditional sampling approaches often rely on direct molecular simulations over extended timescales, which are highly inefficient in capturing rare events such as barrier-crossing transitions or low-frequency conformational changes—particularly under limited computational resources.[2] To address this challenge, various enhanced sampling techniques have been developed, including metadynamics and umbrella sampling. These methods accelerate exploration of the potential energy surface by introducing biasing potentials along predefined reaction coordinates, thereby improving sampling efficiency to a certain extent. However, as these methods are fundamentally thermodynamics-driven and primarily focus on reconstructing free energy surfaces, they tend to exhibit significant limitations in accurately characterizing the true kinetic pathways and associated timescales.[3] In recent years, the rapid advancement of artificial intelligence has led to the growing application of reinforcement learning (RL) and its deep learning extensions (Deep Reinforcement Learning, DRL) in molecular simulations.[4] RL agents learn optimal strategies through sequential decision-making by interacting with their environment, which enables them to adaptively identify efficient transition pathways in high-dimensional or unknown state spaces. This learning paradigm reduces the dependence on predefined reaction coordinates or prior knowledge of the potential energy surface, offering new possibilities for rare event sampling. In this study, we propose a DRL-based adaptive biasing framework aimed at enhancing the sampling efficiency of rare events in complex systems by minimizing the mean first-passage time (MFPT). Specifically, we employ the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm as the core learning component, enabling the agent to generate biasing potentials dynamically based on trajectory history and effectively guide the system across energy barriers. To improve the accuracy of kinetic modeling, we incorporate a Markov State Model (MSM) to discretize the state space and estimate state-to-state transition probabilities. Additionally, the Dynamic Histogram

Analysis Method (DHAM) is utilized to recover unbiased kinetic information from the biased trajectories. The overall strategy optimization is formulated within the RL framework as a control problem with the objective of minimizing MFPT. Experimental results demonstrate that the proposed method achieves rapid policy convergence and high sampling efficiency across several representative model potential systems, significantly reducing the average relaxation time. These findings highlight the potential of deep reinforcement learning as a general and intelligent paradigm for rare event sampling and kinetic analysis in complex molecular systems.

1.2. MOLECULAR DYNAMICS SIMULATION

Molecular Dynamics (MD) simulation is a numerical computational method based on classical mechanics, designed to investigate the microscopic evolution of many-particle systems under given conditions. By numerically integrating Newton's equations of motion, MD simulates the trajectories of individual particles subjected to interatomic or intermolecular forces, thereby capturing the dynamical behavior of the system at atomic or molecular scales.

In MD simulations, particle interactions are typically described by force field functions, which account for both bonded interactions—such as bond stretching, angle bending, and dihedral torsions—and non-bonded interactions, including van der Waals forces and electrostatic interactions. The total force acting on each particle is computed from the force field, and the resulting acceleration is used to update particle positions and velocities over time, generating time-resolved trajectories.

MD simulations are not only capable of capturing the microscopic structural evolution of a system, but also enable the estimation of various macroscopic and thermodynamic properties through statistical analysis of trajectories. These include temperature, pressure, diffusion coefficients, radial distribution functions, and energy distributions. Owing to its ability to provide both structural and dynamical insights, MD has been widely applied in diverse scientific fields, including physics, chemistry, materials science, and biomolecular research. Specific applications involve elucidating mechanisms of energy transfer, conformational transitions, phase behavior, reaction pathways, and nanoscale transport phenomena, offering powerful computational support for understanding complex systems at the molecular level.

At present, MD simulations have been widely applied to study molecular behavior across different phases and environments, including liquids, solids, and biological systems, providing essential theoretical insights into the structure and function of complex molecular systems. In the field of civil engineering, MD techniques have been extensively used to investigate the mechanical, thermal, and chemical properties of cement-based materials, particularly in evaluating the influence of external factors such as temperature, pressure, and humidity.[5,6]

For instance, Valavi et al.[7] developed the ERICA force field specifically for cementitious systems and demonstrated its effectiveness in accurately capturing atomistic behaviors and material properties, highlighting the critical importance of force field selection in MD-based studies of cement materials. Similarly, Li et al. (2022)[?] integrated experimental characterization with MD simulations to explore the micro-mechanism of chloride ion erosion in cement mortars under coastal environmental conditions, further illustrating the potential of MD in elucidating degradation processes under corrosive exposure.

Beyond civil engineering, MD simulations have also found broad applications in chemistry and materials science. For example, both classical and two-phase MD methods have been employed to investigate the gas–liquid interfacial properties of 1,3-butadiene,[?] demonstrating the versatility of MD in handling diverse molecular systems. In the case of cyclohexane–carbon dioxide mixtures, researchers have combined MD simulations with experimental measurements, density gradient theory (DGT), and the PCP-SAFT equation of state to analyze thermodynamic and interfacial properties.[?] Additionally, MD has been used to study vapor–liquid equilibrium and interfacial behavior in various binary gas–liquid mixtures, including N2, C2H6, C3H8, C10H22, and C12H26.[8]

In metallurgical engineering, MD techniques have been applied to simulate microscopic interactions and dynamic processes within materials, thereby uncovering the mechanisms of structural evolution, ion transport behavior, and the relationship between microstructure and macroscopic performance.[?] For example, Reference[?] employed three potential models—BMH, Buckingham, and MiTra—to systematically investigate silicate polymerization and the effects of slag basicity, simulating the structural configurations of ternary slag systems with varying compositional ratios.

1.3. MEAN FIRST PASSAGE TIME

In complex systems involving high energy barriers and rare events, the mean first passage time (MFPT) is a statistical metric that quantifies the expected time for a system to reach a target state from an initial state serves as a theoretical foundation for analyzing the system's kinetic characteristics.[9]

$$\text{MFPT} = E[T_{\text{first passage}}] \tag{1.1}$$

where $T_{\text{first passage}}$ is a random variable representing the time elapsed from the initial state to the target state upon first arrival.

In complex physical systems characterized by metastable states and high energy barriers—such as protein folding, chemical reactions, or nucleation events—the Mean First Passage Time (MFPT) serves as a fundamental kinetic descriptor. It quantifies the average time required for the system to escape a local minimum and reach a predefined target state for the first time. Because transitions over energy barriers are typically rare and stochastic, MFPT captures the timescale of such events and is widely used to assess the dynamical behavior of systems undergoing infrequent state changes.

1.4. DEEP REINFORCEMENT LEARNING

Deep reinforcement learning(DRL) combines the representational power of deep neural networks with the decision-making framework of reinforcement learning.[10] It effectively models the nonlinear relationships among states, actions, and rewards, thereby enhancing learning efficiency and policy generalization in high-dimensional and nonlinear environments.

As shown in Figure 1.1, DRL is a framework for optimal control problems based on Markov Decision Processes (MDPs), where an agent learns to make sequential decisions through interactions with an environment. A standard MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, p, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p(s_{t+1} \mid s_t, a_t)$ denotes the transition probability, and $r(s_t, a_t)$ is the reward function. At each time step

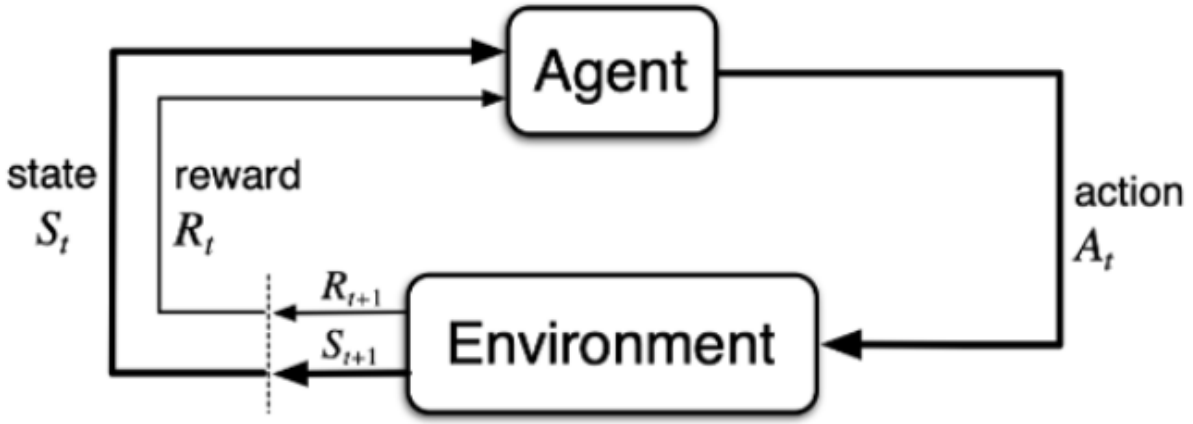


Figure 1.1: Framework of Deep Reinforcement Learning.

t, the agent observes a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to a policy $\pi(a_t \mid s_t)$, receives a scalar reward r_t, and transitions to a new state s_{t+1}. The objective of RL is to learn a policy π that maximizes the expected cumulative discounted reward:

$$E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \tag{1.2}$$

where $\gamma \in [0, 1)$ is a discount factor that balances immediate and future rewards.

The goal of reinforcement learning is to find an optimal policy π^* that maximizes the expected cumulative discounted reward. The total return at time t is defined as:

$$U_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{1.3}$$

where $\gamma \in [0, 1]$ is the discount factor that determines the relative importance of future rewards. Three core functions are commonly used to evaluate a policy.

First is the policy $\pi(a_t \mid s_t)$, which specifies the probability of selecting action a_t given state s_t. For deterministic policies, $\pi(a_t \mid s_t)$ is a delta function, while for stochastic policies, it represents a probability distribution over actions and is often parameterized for easier optimization (e.g., in policy gradient methods).

Second is the state-value function $v_\pi(s)$, which estimates the expected return when starting from state s and following policy π thereafter:

$$v_\pi(s) = E_\pi \left[R_t \mid S_t = s \right] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s \right]. \tag{1.4}$$

Third is the action-value function $q_\pi(s, a)$, which further evaluates the expected return when taking action a in state s, and then following policy π:

$$q_\pi(s, a) = E_\pi \left[U_t \mid S_t = s, A_t = a \right] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s, A_t = a \right]. \tag{1.5}$$

The optimal policy π^* corresponds to the maximum expected return. The corresponding optimal value functions are defined as:

$$v^*(s) = \max_{\pi} v_\pi(s), \tag{1.6}$$

$$q^*(s,a) = \max_{\pi} q_\pi(s,a). \tag{1.7}$$

To find the optimal policy π^*, the value functions must satisfy the Bellman optimality equations. The optimal state-value function $v^*(s_t)$ and the optimal action-value function $q^*(s_t, a_t)$ are respectively defined as:

$$\begin{aligned} v^*(s_t) &= \max_{a_t} E\left[R_{t+1} + \gamma v^*(S_{t+1}) \mid S_t = s_t, A_t = a_t\right] \\ &= \max_{a_t \in \mathcal{A}(s_t)} \sum_{s_{t+1}, r_{t+1}} p(s_{t+1}, r_{t+1} \mid s_t, a_t)\left[r_{t+1} + \gamma v^*(s_{t+1})\right] \end{aligned} \tag{1.8}$$

$$\begin{aligned} q^*(s_t, a_t) &= E\left[R_{t+1} + \gamma \max_{a_{t+1}} q^*(S_{t+1}, a_{t+1}) \mid S_t = s_t, A_t = a_t\right] \\ &= \sum_{s_{t+1}, r_{t+1}} p(s_{t+1}, r_{t+1} \mid s_t, a_t)\left[r_{t+1} + \gamma \max_{a_{t+1}} q^*(s_{t+1}, a_{t+1})\right] \end{aligned} \tag{1.9}$$

To discover the optimal policy, the agent must explore a variety of actions. However, always choosing the current best-known action may lead to suboptimal long-term performance. This trade-off is referred to as the exploration–exploitation dilemma: exploration involves trying new actions to discover potentially better policies, while exploitation relies on the best-known actions to maximize immediate rewards. Effective reinforcement learning requires a careful balance between exploration and exploitation to ensure both policy improvement and sufficient environmental coverage.

Policy improvement constitutes a fundamental step in the policy iteration framework, whose objective is to iteratively enhance a policy π based on its estimated performance. This process involves two alternating phases: policy evaluation and policy improvement. In the policy evaluation phase, the value function associated with the current policy π—denoted as the state-value function v_π—is estimated to assess the expected long-term return from each state under that policy.

Formally, the value of a state under policy π at iteration $t+1$ can be computed using the Bellman expectation equation as follows:

$$\begin{aligned} v_{t+1}(s_t) &= E_\pi\left[R_t + \gamma v_t(S_{t+1}) \mid S_t = s_t\right] \\ &= \sum_{a_t} \pi(a_t \mid s_t) \sum_{s_{t+1}, r_t} p(s_{t+1}, r_t \mid s_t, a_t)\left[r_t + \gamma v_t(s_{t+1})\right]. \end{aligned} \tag{1.10}$$

Updating the value of all states using the above relation is referred to as a *Backup* operation. Once the state-value function v_t is obtained for the current policy, a policy improvement step can be performed. This involves updating the policy to select actions that yield higher action-value estimates. The action-value function $q_t(s_t, a_t)$ under policy π is computed as:

$$
\begin{aligned}
q_{t+1}(s_t, a_t) &= E_\pi \left[R_t + \gamma v_t(S_{t+1}) \mid S_t = s_t, A_t = a_t \right] \\
&= \sum_{s_{t+1}, r_t} p(s_{t+1}, r_t \mid s_t, a_t) \left[r_t + \gamma v_t(s_{t+1}) \right]. \qquad (1.11)
\end{aligned}
$$

Based on the updated q-function, the new policy π_{t+1} is obtained by choosing the action that maximizes the expected return:

$$
\begin{aligned}
\pi_{t+1}(s_t) &= \arg\max_{a_t} q_t(s_t, a_t) \\
&= \arg\max_{a_t} E_\pi \left[R_t + \gamma v_t(S_{t+1}) \mid S_t = s_t, A_t = a_t \right] \\
&= \arg\max_{a_t} \sum_{s_{t+1}, r_t} p(s_{t+1}, r_t \mid s_t, a_t) \left[r_t + \gamma v_t(s_{t+1}) \right]. \qquad (1.12)
\end{aligned}
$$

Through this iterative process, the agent alternates between policy evaluation and policy improvement. Over time, the policy converges toward an optimal strategy. However, depending on the approximation method used, slight differences may arise between the estimated values and the improved policy.

2. MATERIALS AND METHODS

2.1. PROBLEM FORMULATION

Rare events, such as barrier crossing in conformational transitions or chemical reactions, pose significant challenges for conventional MD simulations due to the prohibitively long timescales required to observe such transitions. In this work, we consider a simplified one-dimensional model system to emulate such dynamics, with the aim of efficiently estimating and minimizing the MFPT required for the system to transition from an initial metastable state to a designated target state.

In the MD framework, the motion of a particle subjected to thermal fluctuations and a predefined potential energy surface $U(x)$ can be described by the overdamped Langevin equation:

$$\gamma \frac{dx_t}{dt} = -\frac{dU(x_t)}{dx} + \sqrt{2\gamma k_B T}\,\eta(t), \tag{2.1}$$

where: x_t is the particle's position along the reaction coordinate at time t; γ is the friction coefficient; $U(x)$ is the one-dimensional potential energy surface (e.g., multi-well or funnel-shaped); k_B is the Boltzmann constant, T is temperature; $\eta(t)$ is Gaussian white noise with zero mean and unit variance.

Furthermore, we define the simulation domain as $x \in [0, 2\pi]$, with the particle initialized at $x_0 = x_{\text{start}}$ and expected to reach a target region $\mathcal{X}_{\text{goal}}$ centered at x_{goal}.

The First Passage Time (FPT) is the stochastic time it takes for the particle to first reach the target region:

$$T_{\text{FPT}} = \inf\{t > 0 \mid x_t \in \mathcal{X}_{\text{goal}}\}, \tag{2.2}$$

and the MFPT is defined as its expected value over many trajectories:

$$\text{MFPT} = E[T_{\text{FPT}}]. \tag{2.3}$$

In practical applications, additional biasing forces (e.g., time-dependent Gaussian potentials) are introduced to accelerate the transition. These biases reshape the potential energy surface to increase sampling efficiency without compromising physical relevance. Let $U_{\text{bias}}(x, t)$ denote such a bias, then the effective potential becomes:

$$U_{\text{eff}}(x, t) = U(x) + U_{\text{bias}}(x, t), \tag{2.4}$$

and the Langevin dynamics evolve accordingly.

To explicitly formulate the optimization objective, we consider a parametrized family of time-dependent bias potentials $U_{\text{bias}}(x, t; \theta)$, where θ denotes the set of tunable parameters that define the bias function, which are the amplitude, width, and center of

Gaussian kernels. These parameters directly influence the effective potential $U_{\text{eff}}(x,t)$ and thus the transition kinetics of the system.

The core objective of the optimization task is to identify the parameter set θ^* that minimizes the MFPT under the biased dynamics. Mathematically, the problem can be written as:

$$\theta^* = \arg\min_{\theta} \; E\left[T_{\text{FPT}} \mid U_{\text{eff}}(x,t;\theta) = U(x) + U_{\text{bias}}(x,t;\theta)\right]. \tag{2.5}$$

Alternatively, this objective can be reframed as maximizing a reward functional that penalizes long trajectories and rewards fast, direct transitions. This formulation serves as the theoretical foundation for the learning-based modeling approaches discussed in the subsequent sections.

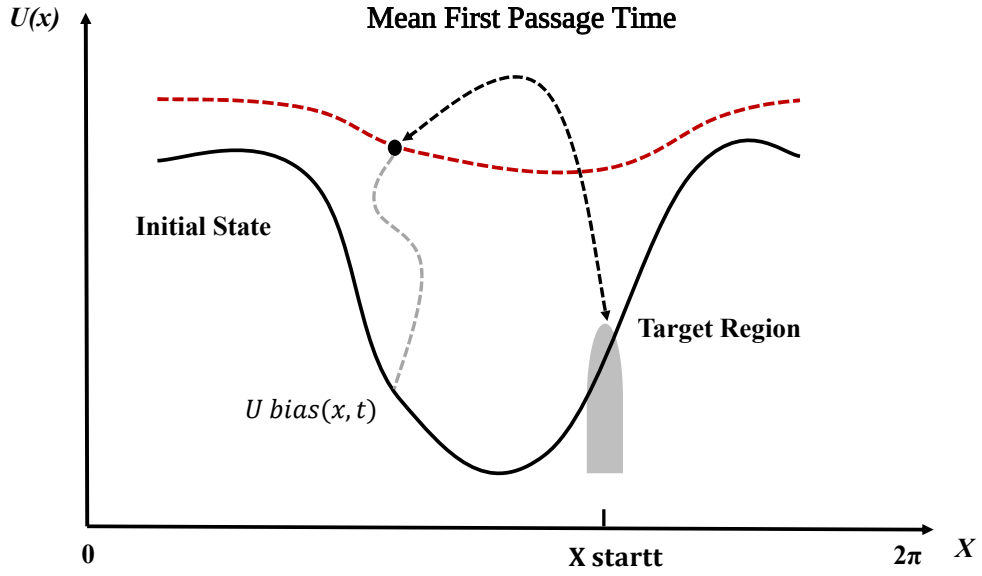


Figure 2.1: A schematic representation of a particle undergoing rare-event transition along a one-dimensional potential energy surface. The particle starts in the initial state (left well) and attempts to reach the target region (shaded area) by crossing an energy barrier. A time-dependent bias potential $U_{\text{bias}}(x,t)$ (red dashed curve) is applied to reshape the energy landscape and accelerate the transition, thereby reducing the MFPT.

Figure 2.1 illustrates a typical scenario in which a particle evolves along a one-dimensional potential energy surface (PES), serving as a simplified model for rare-event transitions in complex molecular systems. The energy landscape exhibits a characteristic multi-well structure, where the particle is initially confined within a stable well on the left, and the target state is located in a separate well on the right. Due to the presence of a high energy barrier between the two regions, the particle rarely crosses the barrier under thermal fluctuations alone, resulting in a First Passage Time (FPT) that is both highly variable and statistically sparse.

To accelerate such rare transitions, a time-dependent bias potential (depicted as a red dashed curve) is introduced in the figure. This bias modifies the effective energy landscape while preserving the system's physical consistency, thereby enhancing the likelihood of barrier crossing and reducing the Mean First Passage Time (MFPT). The dashed arrow indicates a representative transition path, highlighting the core objective of the MFPT optimization task: to identify biasing strategies that enable fast and efficient transitions from the initial to the target state by appropriately tuning the

parameters of the bias potential.

2.2. TD3 ALGORITHM

2.2.1. TD3 Algorithm Description

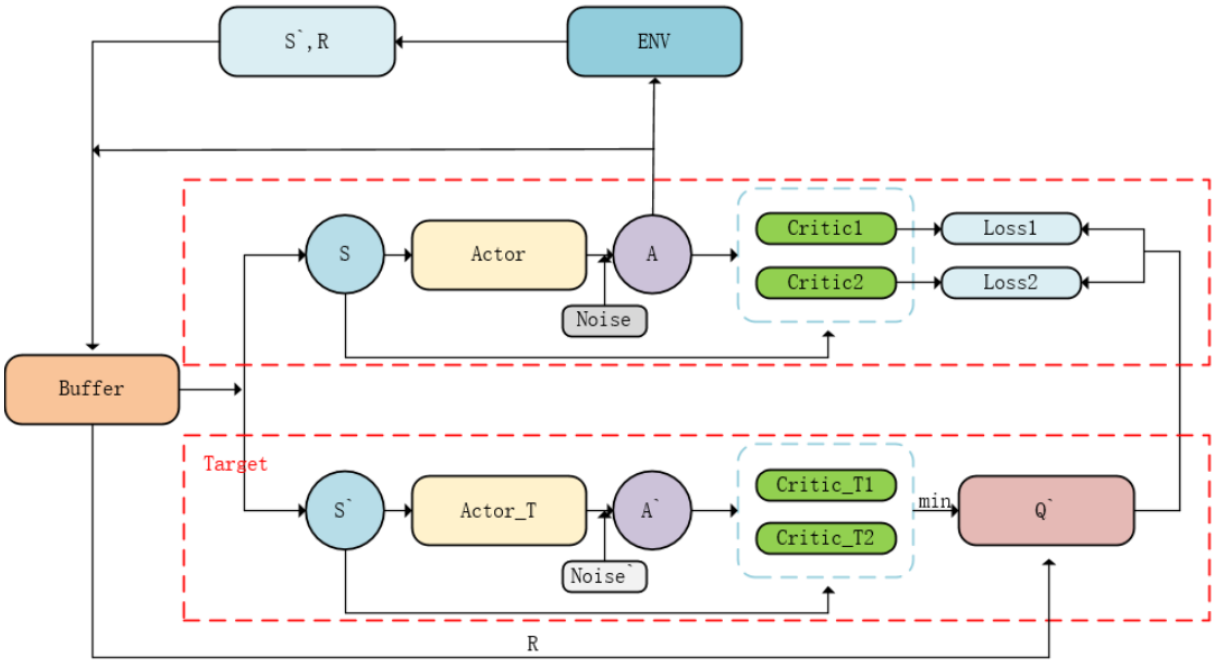


Figure 2.2: Architecture of the TD3 algorithm (Twin Delayed DDPG). The online actor π_θ outputs an action $a = \pi_\theta(s) + \mathcal{N}$ for exploration, and the transition (s, a, r, s') from the environment (ENV) is stored in the replay buffer. Two critics Q_{ϕ_1} and Q_{ϕ_2} are trained using the clipped double-Q target $y = r + \gamma \min_{i\in\{1,2\}} Q_{\phi_i^-}\big(s',\, \pi_{\theta^-}(s') + \epsilon\big)$ with target policy smoothing noise $\epsilon \sim \mathrm{clip}(\mathcal{N}(0,\sigma), -c, c)$. Target networks $(\pi_{\theta^-}, Q_{\phi_1^-}, Q_{\phi_2^-})$ are updated by Polyak averaging, while the actor is updated less frequently than the critics (delayed policy update). The dashed block indicates the optimization pipeline with the twin critics and their targets.

The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is a prominent method in deep reinforcement learning, specifically designed to address the known limitations of its predecessor, Deep Deterministic Policy Gradient (DDPG). In particular, DDPG often suffers from overestimation bias in Q-value estimation and instability during training, which can significantly hinder performance in continuous control tasks. TD3 builds upon the actor–critic architecture and follows an off-policy learning paradigm. It aims to enhance both the stability and precision of policy optimization, especially in environments with high-dimensional state and action spaces.

To mitigate Q-value overestimation, TD3 employs two independent Q-networks, Q_{θ_1} and Q_{θ_2}, and computes the target value using the minimum of the two estimates. This conservative target selection reduces the likelihood of propagating overestimated Q-values during training. The target update is computed as:

$$y = r + \gamma \min\left(Q_{\theta_1'}(s', \pi_{\phi'}(s') + \epsilon),\ Q_{\theta_2'}(s', \pi_{\phi'}(s') + \epsilon)\right), \tag{2.6}$$

where $\epsilon \sim \mathcal{N}(0, \sigma)$ is a clipped noise term added to the target action for smoothing, γ is the discount factor, and $\pi_{\phi'}$ is the target actor network.

Furthermore, TD3 introduces a delayed update strategy that reduces the frequency of policy updates, thereby mitigating instability caused by frequent updates. In TD3, the delayed update mechanism has two aspects: (1) the policy network (i.e., the actor) is updated only after a fixed number of steps; (2) the target Q-network is updated periodically using soft updates. The update rule is:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{2.7}$$

Here, τ is the soft update coefficient, usually set to a small value to avoid large deviations between the target and current networks. In DDPG, the target value is directly calculated using the target policy, which may still lead to overestimation. To address this, TD3 introduces target policy smoothing by adding noise to the target actions, making the target value smoother and more stable, and thus reducing the risk of estimation bias during action selection.

In TD3, the target value y is computed by adding noise to the target action, and the final target is generated as follows:

$$y = r + \gamma \cdot \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi'}(s') + \epsilon), \quad \epsilon \sim \text{clip}(\mathcal{N}(0,\sigma), -c, c) \tag{2.8}$$

Here, ϵ is a noise term, typically sampled from a Gaussian or uniform distribution, and c and $-c$ represent the upper and lower bounds of the action space, used to clip the noise within a reasonable range.

2.2.2. The Workflow of TD3

2.2.2.1. Initialization Phase

In the initialization phase of the TD3 algorithm, all neural network components required for policy learning and value estimation are instantiated. Specifically, two independent Q-networks, denoted as Q_{θ_1} and Q_{θ_2}, are initialized to approximate the state-action value function from different perspectives, thereby enabling the application of Clipped Double Q-learning to mitigate overestimation bias. In parallel, a deterministic policy network π_ϕ is initialized, which maps states to continuous actions and serves as the agent's behavioral policy during interaction with the environment.

To stabilize training and avoid divergence during value propagation, TD3 maintains delayed target networks for both the critic and the actor. These include two target Q-networks $Q_{\theta_1'}$ and $Q_{\theta_2'}$, as well as a target policy network $\pi_{\phi'}$. All target networks are initialized by copying the weights of their respective primary networks. During training, they are updated using soft target updates, which ensures a slow and stable transfer of learning dynamics and reduces the risk of oscillations during policy improvement.

2.2.2.2. Training Procedure

The training procedure forms the core of the TD3 algorithm. During each training iteration, the agent selects an action according to the current policy network and interacts with the environment. The environment returns a reward r and the next state s'. The current Q-networks Q_{θ_1} and Q_{θ_2} are then used to estimate the Q-value of the next action.

To avoid Q-value overestimation, TD3 introduces a double Q-network architecture. Specifically, TD3 maintains two separate Q-networks and computes their Q-values. The target value is calculated as follows:

$$y = r + \gamma \cdot \min\left(Q_{\theta_1'}(s', a'), Q_{\theta_2'}(s', a')\right) \tag{2.9}$$

Here, θ_1' and θ_2' are the parameters of the two target Q-networks, and y denotes the target Q-value. Taking the minimum reduces the risk of overestimation. Additionally, TD3 introduces noise to the target action to further reduce estimation bias, defined as:

$$a' = \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c) \tag{2.10}$$

In this equation, ϵ is the noise term, typically sampled from a Gaussian distribution, and σ is the standard deviation. The noise is clipped within a bounded range $[-c, c]$. Every few steps, TD3 updates the policy and target Q-networks. The update process adopts soft updates to ensure stability in parameter changes, as shown below:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i' \tag{2.11}$$

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi' \tag{2.12}$$

This approach allows the target networks to gradually track the current networks, avoiding sudden changes that may destabilize training.

2.2.2.3. Policy Update

Policy update is the final step in the TD3 training loop. The goal is to improve the policy by using estimates from the Q-networks and the target networks. This enables the policy to choose higher-valued actions and maximize the expected return.

TD3 updates the policy network by maximizing the Q-value of the action suggested by the current policy. The policy gradient is computed as:

$$\nabla_\phi J = E_{s\sim\mathcal{D}}\left[\nabla_a Q_\theta(s,a)\big|_{a=\pi_\phi(s)} \cdot \nabla_\phi \pi_\phi(s)\right] \tag{2.13}$$

Here, $\nabla_\phi J$ is the gradient of the policy objective with respect to the policy parameters ϕ, indicating how the policy should be adjusted to improve its performance.

2.3. MODEL MD AS MDP

In this work, we formulate the DRL task as a Markov Decision Process (MDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the state transition probability, $\mathcal{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. The environment is based on a 1D biased molecular dynamics simulation, where a reinforcement learning agent applies Gaussian bias potentials to drive the system towards a predefined target state in the configuration space. The TD3 algorithm is employed to learn an optimal policy in this continuous control setting.

State Space $\mathcal{S}$: Each state $s \in \mathcal{S}$ is represented by a padded trajectory of particle positions along the x-axis obtained from molecular dynamics propagation. The state is a fixed-length vector derived from the last segment of the simulation trajectory, padded with zeros if necessary. Formally, the state $s_t \in R^d$, where d is the maximum trajectory length determined by the simulation time and trajectory recording frequency:

$$s_t = \text{pad\_trajectory}(\text{coor}_x, d)$$

Action Space $\mathcal{A}$: Each action $a \in \mathcal{A} \subset R^3$ represents the parameters of a Gaussian bias potential applied to the system. Specifically, an action is a 3-dimensional continuous vector $a = (a, b, c)$, where a is the amplitude of the Gaussian (typically fixed as 1), b is the center of the Gaussian along the reaction coordinate and is scaled to the interval $[0, 2\pi]$, and c is the width of the Gaussian (also fixed in practice). In the current implementation, the agent primarily learns to control the parameter b, which determines the position at which the bias is applied in order to guide the system toward the target region of the configuration space.

Reward Function $\mathcal{R}$: The reward is designed to encourage the system to reach a specific target state x_{target}. It consists of two components: a distance-based penalty and a large bonus for reaching the target. Let traj denote the propagated trajectory and $d_t = \text{EWA}(|x_i - x_{\text{target}}|^2)$ be the exponentially weighted average squared distance to the target over the trajectory. Then the reward is computed as:

$$r_t = -d_t + 1000 \cdot I_{\{\text{target reached}\}}$$

where $I_{\{\cdot\}}$ is the indicator function. This formulation incentivizes the agent to guide the system efficiently toward the target configuration.

State Transition Probability $\mathcal{P}$: The transition dynamics $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ are governed by the underlying molecular dynamics simulation, which is deterministic given the physical laws but exhibits stochasticity due to thermal noise introduced by the Langevin integrator. Therefore, the environment can be viewed as a partially stochastic simulator, where transitions are influenced both by the applied bias (action) and by stochastic thermal fluctuations. This hybrid nature makes learning robust policies challenging, particularly in high-dimensional systems.

Discount Factor γ: The discount factor $\gamma \in [0, 1]$ determines the importance of future rewards relative to immediate ones. In this work, we set $\gamma = 0.99$, which encourages the agent to learn long-term strategies that accumulate high cumulative reward over time. A value close to 1 biases the policy toward maximizing long-horizon objectives, which is appropriate in molecular systems where reaching a metastable or target state may require a sequence of intermediate actions.

Based on the aforementioned Markov Decision Process formulation, the TD3 algorithm is employed to guide the learning and application of bias potentials, thereby enhancing the sampling efficiency of rare events and accelerating the system's progression toward the target configuration. The following pseudocode outlines the core training procedure of TD3 within this reinforcement learning-driven MD framework. The workflow includes state acquisition, action selection, environment interaction, experience storage, updates of the Q-networks and the policy network, as well as soft updates of the target networks.

Algorithm 1 TD3 for Biased Molecular Dynamics Sampling

Initialize: Q-networks $Q_{\theta_1}, Q_{\theta_2}$, policy network π_ϕ
Initialize target networks $Q_{\theta'_1}, Q_{\theta'_2}, \pi_{\phi'}$ with $\theta'_i \leftarrow \theta_i,\ \phi' \leftarrow \phi$
Initialize empty replay buffer $\mathcal{D}$
for episode $= 1$ to M **do**
    Initialize MD system and obtain initial state s_0
    for timestep $t = 1$ to T **do**
        Select action $a_t = \pi_\phi(s_t) + \mathcal{N}(0, \sigma)$
        Apply Gaussian bias with parameters a_t in MD simulation
        Propagate biased MD for fixed steps and obtain s_{t+1}, r_t
        Store transition (s_t, a_t, r_t, s_{t+1}) in buffer $\mathcal{D}$
        Sample minibatch of transitions from $\mathcal{D}$
        Compute target action:

$$a' = \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$$

        Compute target Q-value:

$$y = r + \gamma \cdot \min(Q_{\theta'_1}(s', a'), Q_{\theta'_2}(s', a'))$$

        Update Q-networks by minimizing:

$$\mathcal{L}(\theta_i) = E_{(s,a,r,s')}\left[(Q_{\theta_i}(s, a) - y)^2\right]$$

        if t mod policy_delay == 0 **then**
            Update policy by maximizing $Q_{\theta_1}(s, \pi_\phi(s))$
            Soft update target networks:

$$\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$$

$$\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$$

        end if
        $s_t \leftarrow s_{t+1}$
    end for
end for

3. RESULTS

3.1. EXPERIMENTS

In this section, we evaluate the performance of the proposed reinforcement learning framework based on the TD3 algorithm applied to a one-dimensional molecular system. The agent learns to apply adaptive Gaussian bias potentials in order to drive the system efficiently toward a predefined target configuration along the reaction coordinate. The environment is implemented using OpenMM for molecular dynamics simulation, and the interaction between the agent and the environment is managed via a custom wrapper that supports bias injection, trajectory tracking, and reward computation.

Our experiments are conducted using a Python-based simulation pipeline, which integrates the TD3 agent, a 1D reaction-coordinate-based molecular dynamics environment, and customized utilities for simulation, reward shaping, and data recording. The simulation employs Langevin dynamics at a temperature of 300 K, and uses a harmonic potential to constrain motion in non-target dimensions. During training, the agent interacts with the environment by proposing Gaussian bias parameters, which are applied dynamically to the system and evaluated based on the resulting state trajectory.

We assess the learning performance in terms of accumulated reward, trajectory convergence, and the system's ability to reach the target state across episodes. In the following subsections, we detail the simulation setup, hyperparameter configurations, evaluation metrics, and comparative results.

3.1.1. Parameter Settings

The key hyperparameters used in the TD3 training and simulation environment are summarized in Table I. These include learning rates, noise parameters, update frequencies, and environment-specific configurations such as temperature and trajectory window size. All parameters are selected based on standard practices in continuous control tasks and validated through preliminary experiments to ensure training stability and convergence.

3.1.2. Representative Episode Evaluation on TD3 algortihm

As shown in Fig.3.1, the particle guided by the agent exhibits distinct dynamical transitions across multiple episodes. It can be observed that the particle starts at an initial position of approximately 2.2 nm. In Prop 0, the particle primarily undergoes rapid, small-amplitude oscillations near its initial position without any significant transitions,

TABLE I

Hyperparameter Settings for TD3 and Simulation Environment

Parameter Description	Value
Actor/Critic hidden layer size	128
Discount factor γ	0.99
Soft update rate τ	0.005
Actor/Critic learning rate	$1e{-4}$
Policy noise / noise clip	0.2 / 0.5
Exploration noise (action)	0.1
Policy update delay	Every 2 steps
Replay buffer size	10000
Batch size	256
Max actions per episode	20
Simulation temperature	300 K
Simulation time per step	2000 fs
Trajectory window size	100
Number of bins (state discretization N)	40

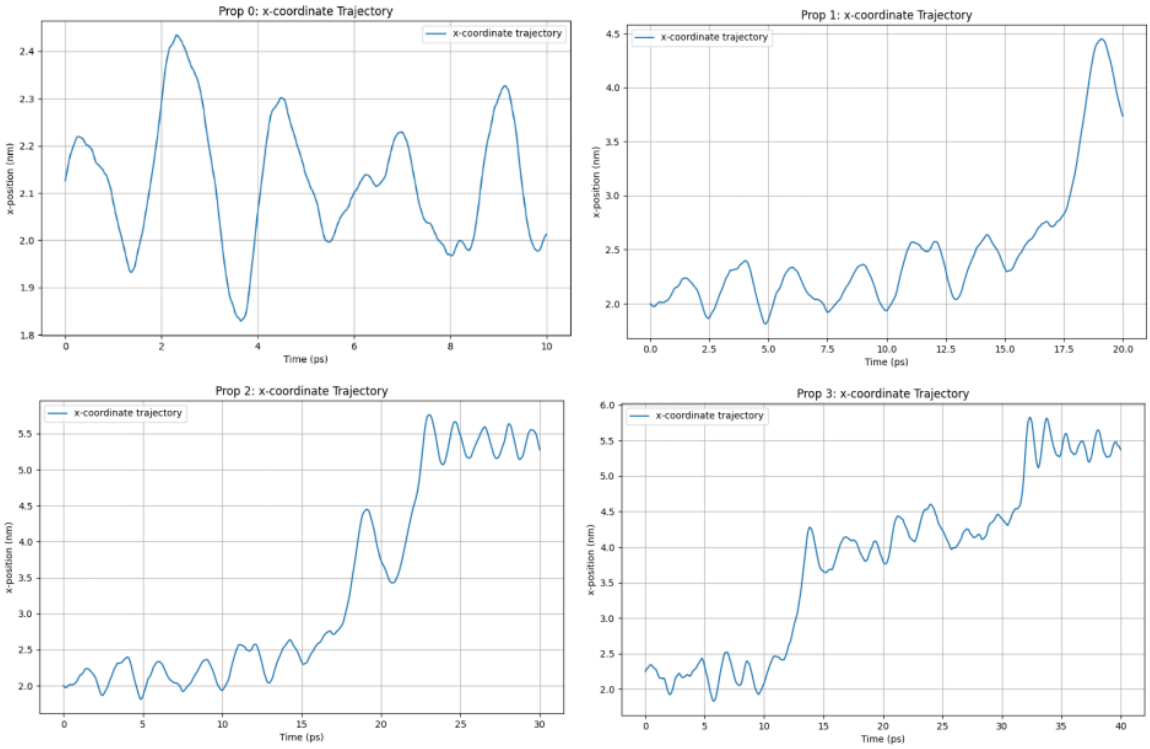


Figure 3.1: Representative particle trajectories over multiple episodes. The plots show distinct dynamical transitions from the initial state toward the target metastable basin, guided by the learned reinforcement learning policy.

indicating that the particle is still in the early stage of exploration at this point.

Subsequently, the trajectory begins to show an upward trend around 8 ps, as illustrated in Prop 1. By the end of this episode, the particle reaches approximately 4.2 nm, indicating a slow yet continuous transition behavior. Furthermore, the trajectory in Prop 2 exhibits two distinct transition phases: the first occurs around 10 ps, during which the particle jumps from approximately 2.2 nm to 3.7 nm; the second transition takes place at around 30 ps, ultimately stabilizing near 5.5 nm. The absence of significant backtracking suggests that the particle successfully overcomes the free energy barrier and reaches the target metastable state.

The final complete trajectory is shown in Prop 3. Specifically, Prop 3 presents a more pronounced multi-stage transition process: initial fluctuations of moderate amplitude are followed by several abrupt transition phases, ultimately stabilizing near 5.6 nm. The MFPT is approximately 35 ps. The irreversibility of the transitions observed in Prop 2 and Prop 3 effectively demonstrates the strong convergence and guidance capability of the applied reinforcement learning strategy. Overall, the experimental trajectories indicate that the reinforcement learning policy can effectively identify optimal pathways, facilitate the occurrence of rare events, and robustly guide the particle into the target free energy basin.

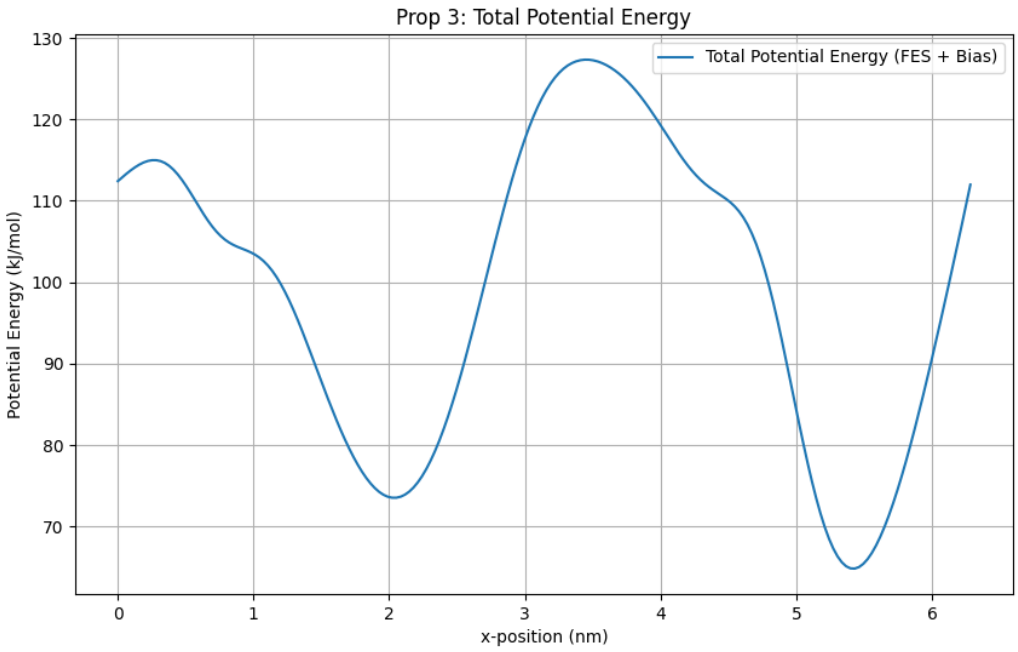


Figure 3.2: Total potential energy profile of the system. The profile shows two prominent wells and a transition barrier, characterizing stable and metastable states.

As shown in the total potential energy profile in Fig. 3.2, the system exhibits multiple stable and metastable states. The lowest potential well is located at approximately 5.6 nm, which closely matches the final stable region of the particle observed in Fig. 3.1, indicating that the reinforcement learning agent effectively guided the particle toward a globally or locally optimal configuration. Another relatively deep potential well appears at around 1.8 nm, representing the particle's initial state. A distinct energy barrier exists between the two wells, peaking near 3.4 nm, which corresponds to a typical transition-state barrier in the energy landscape.

TABLE II
Representative Actions and Particle Positions

Step	Particle Position (nm)	Action Parameters (a, b, c)
1	≈ 1.93	(1.64, 0.83, 0.84)
2	≈ 3.75	(1.17, 4.52, 0.58)
3	≈ 5.28	(1.00, 0.001, 0.50)

As shown in Table II, the three action tuples illustrate the evolving decision-making process of the reinforcement learning agent in guiding the particle through metastable states. Initially, the bias potential is placed relatively far from the particle, providing limited driving force. As learning progresses, the agent applies a sharper and forward-positioned bias, effectively enabling the particle to overcome the free energy barrier. Finally, once the particle reaches the global minimum near 5.28 nm, the bias becomes negligible, suggesting that the optimal state has been reached and maintained. This behavior highlights the agent's ability to apply spatially and temporally adaptive bias potentials to accelerate rare event transitions.

Fig. 3.3 presents the final values of parameters a, b, and c at the end of each training episode, offering insights into how the reinforcement learning agent adaptively adjusts the biasing scheme over time.

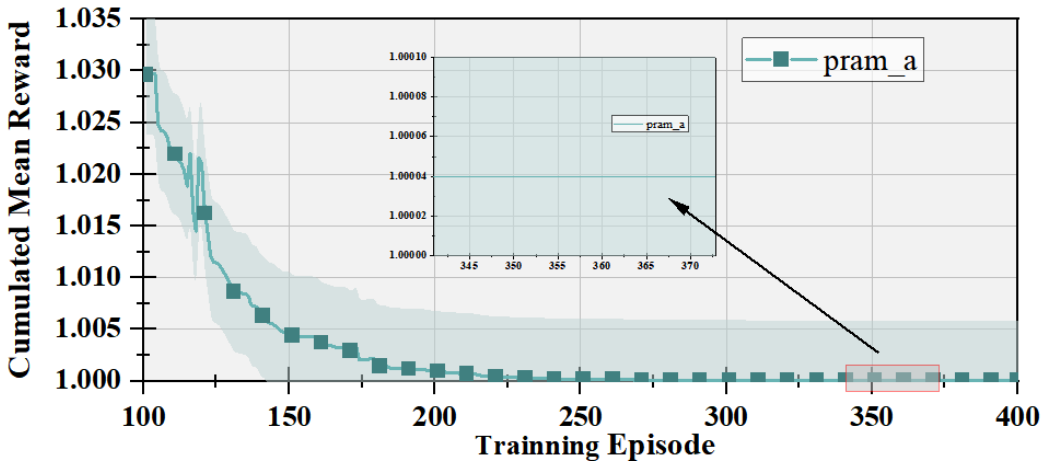
As shown in Fig. 3.3(a), parameter a, which controls the amplitude of the Gaussian bias potential, exhibits a clear convergence trend. Initially fluctuating due to exploration, it stabilizes rapidly around 1.0 after approximately 150 episodes. This behavior indicates that the agent learns to apply a bias of consistent strength at each step, effectively balancing local exploration and global guidance. From a physical standpoint, a stable $a \approx 1$ implies the adoption of a uniform yet effective energetic strategy that consistently drives the particle out of metastable regions.

Fig. 3.3(b) illustrates the trajectory of parameter b, which defines the center position of the Gaussian bias relative to the particle's location. After about 300 episodes, b gradually converges toward zero. This indicates that the agent learns to apply the bias directly at the particle's current position—a behavior reminiscent of adaptive metadynamics. Such placement ensures localized reinforcement without exerting force on distant configurations, thereby maintaining dynamical fidelity while enhancing stability in the final state.

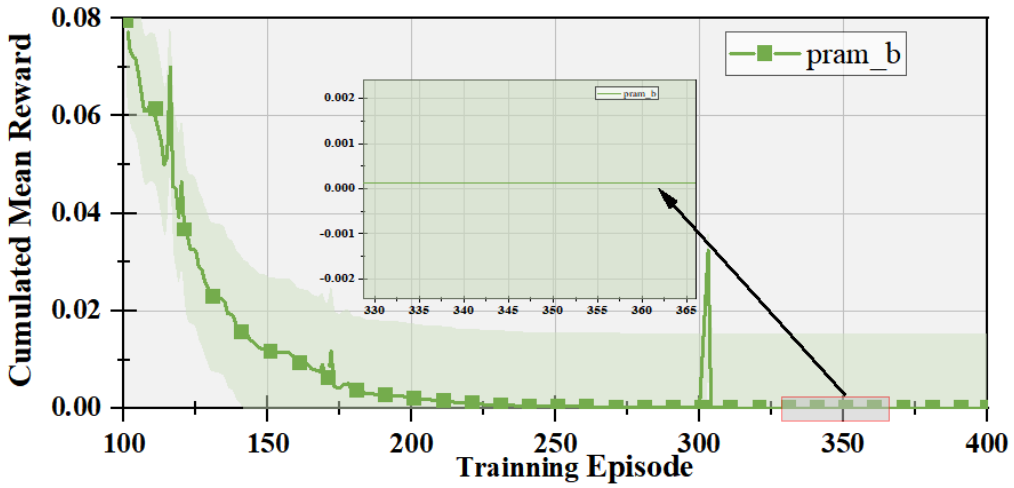
In Fig. 3.3(c), parameter c, corresponding to the standard deviation (i.e., width) of the Gaussian bias, quickly converges to approximately 0.5 and remains stable throughout training. This suggests that the agent has identified an optimal spatial scale for applying the bias: not too narrow to be ineffective, nor too broad to diffuse the effect. Physically, this reflects a bias with sufficient spatial extent to facilitate crossing of energy barriers while preserving sensitivity to local structural features.

3.1.3. Comparative Experimental Analysis

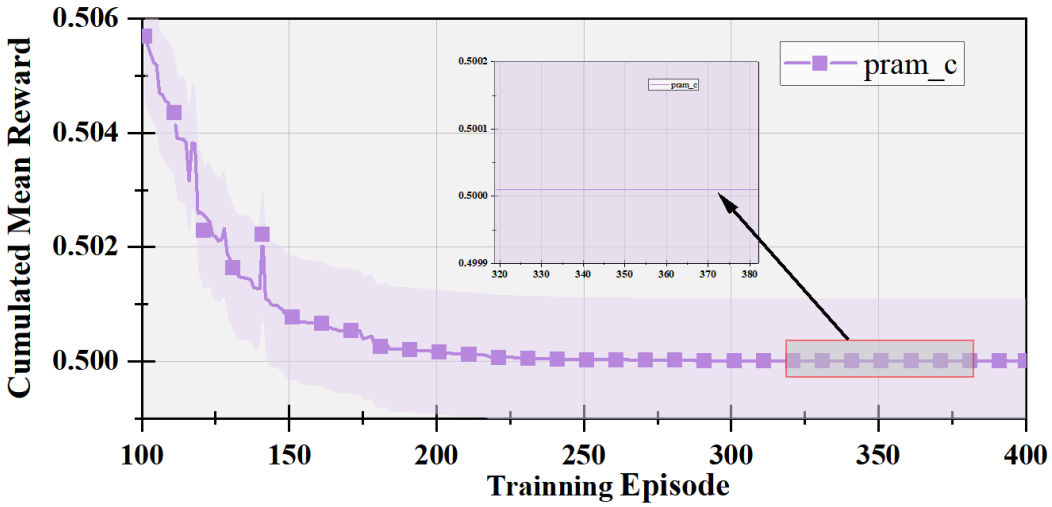
To assess the performance of different reinforcement learning strategies in computing the mean first-passage time (MFPT) for molecular dynamics, this study employed the Deep Q-Network (DQN), Advantage Actor-Critic (A2C), and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithms under identical system conditions. The comparison was conducted based on the reaction coordinate trajectories and the reconstructed total potential energy surfaces.



(a) Parameter a: Amplitude

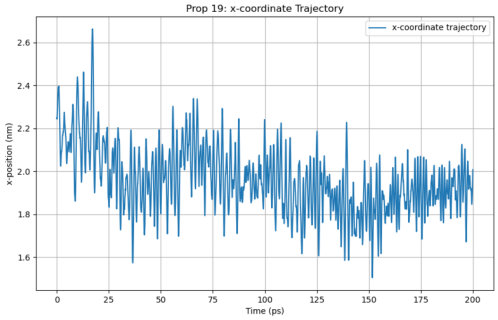


(b) Parameter b: Bias center

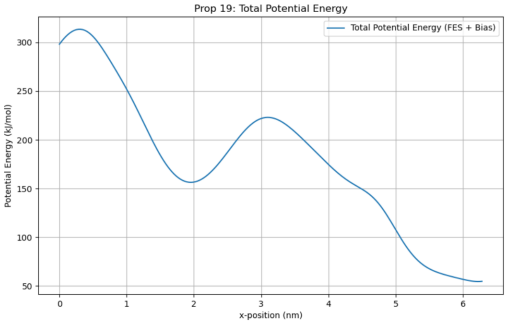


(c) Parameter c: Width

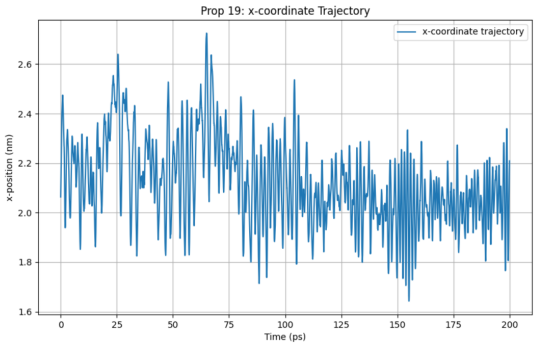
Figure 3.3: Evolution of Gaussian bias parameters during training. (a) Amplitude a; (b) Bias center b; (c) Width c. The trends reflect the agent's adaptation toward stable and localized biasing strategies.



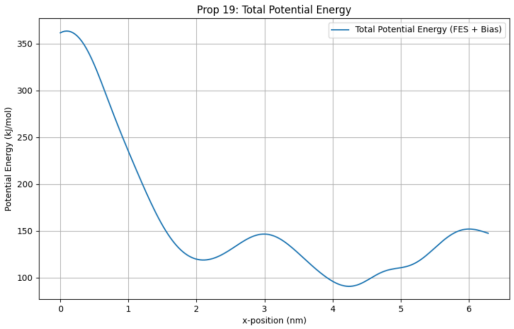
(a) Representative particle trajectories on A2C algorithm.



(b) Total potential energy profile of the system on A2C algorithm.



(c) Representative particle trajectories on DQN algorithm.



(d) Total potential energy profile of the system on DQN algorithm.

Figure 3.4: Comparison of particle trajectories and total potential energy profiles for A2C and DQN algorithms.

From the trajectory plots, the A2C method demonstrates a high frequency of transitions across different potential wells, allowing the particle to traverse multiple metastable states and achieve relatively uniform sampling over the reaction coordinate space. This indicates that its biasing strategy effectively reduces the energy barrier constraints, facilitating exploration of high-energy regions; however, it may cause oversampling in certain areas. In contrast, the DQN method shows more pronounced fluctuations in the early stages but exhibits prolonged residence times in later stages, particularly within deep potential wells. This suggests reduced efficiency in crossing high barriers, leading to insufficient sampling in the high x-position region of the potential energy curve.

In summary, TD3 outperforms the other two methods in barrier-crossing efficiency, potential energy surface reconstruction accuracy, and overall sampling effectiveness, resulting in the smallest expected MFPT. A2C ranks second, with barrier-crossing performance close to that of TD3 but slightly inferior in reconstructing the fine details of the potential surface. DQN shows stability in low-energy sampling but lacks efficiency in exploring high-barrier regions, leading to a larger expected MFPT. These results indicate that policy optimization in continuous action spaces (as in TD3) offers significant advantages for rare-event sampling in molecular dynamics.

4. DISCUSSION

4.1. DISCUSSION

The results presented in this work demonstrate the effectiveness of reinforcement learning (RL), specifically the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, in accelerating rare-event transitions within a one-dimensional molecular system by minimizing the Mean First Passage Time (MFPT). Through the dynamic placement of Gaussian bias potentials, the RL agent successfully guides the particle across energy barriers and stabilizes it in the target metastable basin, as confirmed by the trajectory profiles and potential energy landscape.

A particularly noteworthy observation is the convergence behavior of the bias parameters (Fig. 3.3). The amplitude parameter a stabilizes near 1.0, indicating that the agent learns to apply consistent yet sufficient energetic encouragement throughout the episode. Meanwhile, the bias center parameter b converges toward zero, suggesting that the learned policy favors placing the bias near the particle's current location—a hallmark of localized and adaptive sampling strategies. The width parameter c, stabilized at approximately 0.5, further reflects the agent's ability to balance local specificity with global guidance.These findings highlight the adaptability and robustness of TD3 in learning physically meaningful control policies, capable of mimicking principles from traditional enhanced sampling methods such as metadynamics.

Despite these promising results, several limitations should be acknowledged. First, the model system is simplified to one dimension, which restricts the generalizability of the current policy to high-dimensional, realistic molecular systems. Second, although the reward function is designed to encourage rapid transitions, it does not explicitly penalize physically unfeasible trajectories, which could affect the transferability of the learned bias in real-world applications. Third, while convergence was observed in most training runs, further statistical validation across multiple random seeds is necessary to assess policy robustness.

Nevertheless, this study provides a solid foundation for integrating deep reinforcement learning into kinetic modeling and rare-event sampling. Future work can focus on extending this framework to multidimensional reaction coordinates, incorporating physical constraints during learning, and benchmarking against state-of-the-art enhanced sampling algorithms in realistic biomolecular or material systems.

5. CONCLUSION AND FUTURE OUTLOOK

5.1. CONCLUSION

In this study, we proposed a reinforcement learning framework based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm to accelerate rare-event transitions in molecular systems. By dynamically applying Gaussian bias potentials, the agent effectively learns to guide a particle across energy barriers toward a target metastable state, significantly reducing the mean first passage time (MFPT).

Comprehensive trajectory analyses and potential energy profiling confirmed that the learned policy is capable of producing physically interpretable transition pathways. The RL agent not only promotes rare transitions but also exhibits convergent behavior in the bias parameters. The amplitude, center, and width of the applied Gaussian bias potentials evolve during training toward values consistent with efficient and localized sampling strategies, resembling principles observed in traditional enhanced sampling methods such as metadynamics.

Overall, our results demonstrate that deep reinforcement learning offers a flexible and adaptive approach for designing system-specific biasing schemes that can accelerate rare-event sampling in stochastic dynamical systems.

5.2. FUTURE OUTLOOK

Building on the current success in one-dimensional systems, several promising directions for future research can be identified:

Extension to higher dimensions: Realistic molecular systems involve complex, high-dimensional configuration spaces. Extending the RL framework to handle multiple collective variables or high-dimensional reaction coordinates will be crucial.

Integration of physical constraints: Introducing domain-specific physical priors—such as energy conservation, symmetry, or force continuity—into the learning process can enhance the physical plausibility and transferability of the learned policies.

Comparative benchmarking: Evaluating the RL-based approach against state-of-the-art enhanced sampling techniques (e.g., metadynamics, umbrella sampling, adaptive biasing force) on benchmark molecular systems would provide more insight into its practical value.

Application to real molecular processes:Applying this methodology to simulate transitions in protein folding, ligand binding, or phase changes in materials could demonstrate its utility in solving real-world scientific problems.

These extensions will further validate the capacity of reinforcement learning to serve as a general-purpose tool for rare-event acceleration in molecular simulation and beyond.

BIBLIOGRAPHY

[1] S. AlRawashdeh, K. H. Barakat, Applications of molecular dynamics simulations in drug discovery, Computational drug discovery and design (2023) 127–141.

[2] H. A. Filipe, L. M. Loura, Molecular dynamics simulations: advances and applications, Molecules 27 (7) (2022) 2105.

[3] I. Srivastava, A. Kotia, S. K. Ghosh, M. K. A. Ali, Recent advances of molecular dynamics simulations in nanotribology, Journal of Molecular Liquids 335 (2021) 116154.

[4] Y. Li, Deep reinforcement learning: An overview, arXiv preprint arXiv:1701.07274 (2017).

[5] Z. Chen, J. Pei, R. Li, F. Xiao, Performance characteristics of asphalt materials based on molecular dynamics simulation – a review, Construction and Building Materials 189 (2018) 695–710.

[6] M. J. Abdolhosseini Qomi, L. Brochard, T. Honorio, I. Maruyama, M. Vandamme, Advances in atomistic modeling and understanding of drying shrinkage in cementitious materials, Cement and Concrete Research 148 (2021) 106536.

[7] M. Valavi, Z. Casar, A. Kunhi Mohamed, P. Bowen, S. Galmarini, Molecular dynamic simulations of cementitious systems using a newly developed force field suite erica ff, Cement and Concrete Research 154 (2022) 106712.

[8] B. Sankar, P. Ramadoss, Mechanical and durability properties of high strength concrete incorporating different combinations of supplementary cementitious materials: a review, in: Proceedings of Fourth International Conference on Inventive Material Science Applications: ICIMA 2021, Springer, 2021, pp. 543–557.

[9] D. E. Kleiman, H. Nadeem, D. Shukla, Adaptive sampling methods for molecular dynamics in the era of machine learning, The Journal of Physical Chemistry B 127 (50) (2023) 10669–10681.

[10] Q. Bai, S. Liu, Y. Tian, T. Xu, A. J. Banegas-Luna, H. Pérez-Sánchez, J. Huang, H. Liu, X. Yao, Application advances of deep learning methods for de novo drug design and molecular dynamics simulation, Wiley Interdisciplinary Reviews: Computational Molecular Science 12 (3) (2022) e1581.