TRANSITION PATH SAMPLING WITH IMPROVED OFF-POLICY TRAINING OF DIFFUSION PATH SAMPLERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding transition pathways between meta-stable states in molecular systems is crucial to advance material design and drug discovery. However, unbiased molecular dynamics (MD) simulations are computationally infeasible due to the high energy barriers separating these states. Although recent machine learning techniques offer potential solutions, they are often limited to simple systems or rely on collective variables (CVs) derived from costly domain expertise. In this paper, we introduce a novel approach that trains diffusion path samplers (DPS) for transition path sampling (TPS) without the need for CVs. We recast the problem as an amortized sampling of the path measure of transition paths, minimizing the log-variance divergence between the path measure induced by DPS and that of transition paths. Leveraging the log-variance divergence, we propose learnable control variates for reducing the variance of gradient estimators and off-policy training objective with replay buffers and simulated annealing to improve sample efficiency and diversity. We also propose a scale-based equivariant parameterization of the bias forces to ensure scalability for high-dimensional tasks. We evaluate our approach, coined TPS-DPS, on a synthetic double-well potential and three peptides: Alanine Dipeptide, Polyproline Helix, and Chignolin. Results show that our approach produces more realistic and diverse transition pathways compared to existing baselines. We also provide links to our project page and code.

1 INTRODUCTION

In material design and drug discovery, it is crucial to understand the mechanisms and kinetics of transitions between meta-stable states of molecular systems, such as protein folding and chemical reactions (Mulholland, 2005; Piana et al., 2012; Ahn et al., 2019; Spotte-Smith et al., 2022). Their comprehensive study requires sampling transition paths (Elber, 2016; Lee et al., 2017), which provides insight into mechanisms and energy landscapes. However, naïvely sampling transition paths by unbiased molecular dynamics (MD) simulations is often computationally costly due to high energy barriers, which cause an exponential decay in probability to make a transition (Pechukas, 1981).

To address this problem, researchers have developed enhanced sampling approaches such as steered MD (SMD; Schlitter et al., 1994; Izrailev et al., 1999), umbrella sampling (Torrie & Valleau, 1977; Kästner, 2011), meta-dynamics (Ensing et al., 2006; Branduardi et al., 2012; Bussi & Branduardi, 2015), on-the-fly probability-enhanced sampling (OPES; Invernizzi & Parrinello, 2020), and adaptive biasing force (ABF; Comer et al., 2015) methods. These methods rely on *bias forces* to facilitate transitions across high energy barriers. They are mainly designed based on collective variables (CVs), which are functions of atomic coordinates that capture the slow modes of the transition. Although effective for some systems, the reliance on expensive domain knowledge limits the applicability of the methods to systems where CVs are less understood.

Recently, machine learning has emerged as a promising paradigm for CV-free transition path sampling (TPS) (Das et al., 2021; Lelièvre et al., 2023; Holdijk et al., 2024). The key idea is to parameterize the bias force using a neural network and train it to sample transition paths directly with the corresponding biased MD simulation. In particular, Lelièvre et al. (2023) considered reinforcement learning to sample paths escaping meta-stable states. Das et al. (2021); Hua et al. (2024); Holdijk et al. (2024) considered TPS problem as minimizing the reverse Kullback-Leibler (KL) divergence between the path measures induced by the neural network and the target path measure. However, minimizing

the reverse KL divergence suffers from mode collapse, capturing only a subset of modes of the target distribution (Vargas et al., 2023; Richter & Berner, 2024). Furthermore, Das et al. (2021); Lelièvre et al. (2023); Hua et al. (2024) limited their evaluation to low-dimensional synthetic systems. Designing machine learning algorithms for CV-free TPS for real molecules remains an open problem.

Contribution. In this work, we propose the *diffusion path sampler* (DPS) to solve the transition path sampling problem.[1] Our approach, coined TPS-DPS, (1) trains the bias force by minimizing a recently proposed log-variance divergence (Nüsken & Richter, 2021) between the path measure induced by the biased MD and the target path measure, and (2) uses scale-based parameterization of the bias force to handle high-dimensional tasks, e.g., Chignolin folding. Specifically, to leverage desirable properties of the log-variance divergence, such as robustness of gradient estimator and degree of freedom in reference path measure, we propose to learn control variates for reducing the variance of gradient estimators and use *off-policy* training scheme with replay buffer and simulated annealing to improve sample efficiency and diversity and prevent the mode collapse.

We also introduce a new $SE(3)$ equivariant scale-based parameterization for the bias force to sample meaningful paths more frequently in training. Our key idea is to predict the atom-wise positive scaling factor of displacement from current molecular states to the target meta-stable state, which guarantees the bias force to decrease the distance between them for every MD step. We also use the Kabsch algorithm (Kabsch, 1976) to align the current molecular states with the target meta-stable state, guaranteeing $SE(3)$ equivariance of bias force for better generalization across the states.

We extensively evaluate our method on the synthetic double-well potential and three peptides: Alanine Dipeptide, Polyproline Helix, and Chignolin. We compare TPS-DPS with prior ML approach (PIPS; Holdijk et al., 2024), as well as classical non-ML methods, e.g., two-way shooting and steered MD (SMD; Schlitter et al., 1994; Izrailev et al., 1999). Our experiments demonstrate that TPS-DPS consistently generates realistic and diverse transition paths, similar to the ground truth ensemble. In addition, we do ablation studies of the proposed components to verify the effectiveness of our approach. In Appendix C, we further show the promise of our method on three fast folding proteins: Trpcage, BBA, and BBL (Lindorff-Larsen et al., 2011).

2 Related work

Transition path sampling (TPS) without ML. Metadynamics (Branduardi et al., 2012), on-the-fly probability-enhanced sampling (OPES; Invernizzi & Parrinello, 2020), adaptive biasing force (ABF; Comer et al., 2015), and steered molecular dynamics (SMD; Schlitter et al., 1994; Izrailev et al., 1999) were introduced to explore molecular conformations that are difficult to access by unbiased molecular dynamics (MD) within limited simulation times (Hénin et al., 2022). However, they mostly rely on collective variables (CVs) for high-dimensional problems and are inapplicable to systems with unknown CVs. To sample transition paths without CVs, Dellago et al. (1998) proposed shooting methods that use the Markov chain Monte Carlo (MCMC) procedure on path space. In this work, we compare our method with SMD and variable-length two-way shooting as non-ML baselines.

Data driven ML approaches. Recently, generative models have been trained to sample new transition paths given a dataset of transition paths. Petersen et al. (2023); Triplett & Lu (2023) and Lelièvre et al. (2023) applied diffusion probabilistic models (Ho et al., 2020) and variational auto-encoders (Kingma & Welling, 2013) for transition path sampling, respectively. However, these methods are limited to small systems. Klein et al. (2024); Schreiner et al. (2024); Jing et al. (2024) proposed to accelerate MD by generating time-coarsened dynamics, but the time-coarsed dynamics cannot capture the fine-grained details of the transition, e.g., the transition states. Duan et al. (2023); Kim et al. (2024) use neural networks to generate transition states of a given chemical reaction, but cannot generate transition paths.

Data free ML approaches. Without a previously collected dataset, Das et al. (2021); Lelièvre et al. (2023); Sipka et al. (2023); Hua et al. (2024); Holdijk et al. (2024) trained the bias forces to directly sample transition paths using the biased MD. Lelièvre et al. (2023) used reinforcement learning to train the bias forces but focused on escaping an initial meta-stable state rather than targeting a given meta-stable state. Sipka et al. (2023) used differentiable biased MD simulation to

[1] We coin our method diffusion path sampler since it samples paths using diffusion SDE, similar to diffusion samplers (Zhang & Chen, 2022; Vargas et al., 2023) that use diffusion SDEs for sampling the final state.

train bias potential and introduce partial back-propagation and graph mini-batching techniques to resolve computational issues in differentiable simulation. Das et al. (2021); Hua et al. (2024); Holdijk et al. (2024) considered the TPS problem as minimizing the reverse KL divergence between path distribution from biased MD and transition path distribution. Das et al. (2021); Hua et al. (2024) limited their evaluation to low-dimensional synthetic systems. In this work, we mainly compare our method with (PIPS; Holdijk et al., 2024). Recently, Du et al. (2024) considered the TPS problem as minimizing Doob's Lagrangian objective with boundary constraints. They parameterized marginal distribution as (mixture) Gaussian path distribution to satisfy the boundary constraints without relying on simulation in training time and sampled transition paths with the bias force derived from the Fokker-Planck equation in inference time.

3 Transition Path Sampling with Diffusion Path Samplers

In this section, we introduce our method, coined transition path sampling with diffusion path sampler (TPS-DPS). Our main idea is to formulate the transition path sampling (TPS) problem as a minimization of log-variance divergence (Nüsken & Richter, 2021) between two path measures: the path measure induced by DPS and that of transition paths. Our main methodological contribution is twofold: (1) a new off-policy training algorithm that minimizes the log-variance divergence with the learnable control variate, replay buffer, and simulated annealing (2) a $SE(3)$ equivariant scale-based parameterization of the bias force that provides inductive bias for dense training signals in high-dimensional problems.

3.1 Problem Setup

Our goal is to sample transition paths from one meta-stable state to another meta-stable state given a molecule system. We provide an example of the problem for Alanine Dipeptide in Figure 1. We view this as a task to sample paths from an unbiased molecular dynamics (MD) in Equation (1) conditioned on its starting and ending points of initial and target meta-stable states, respectively. To solve this task, we train the bias force parameterized by a neural network to amortize the sampling procedure.

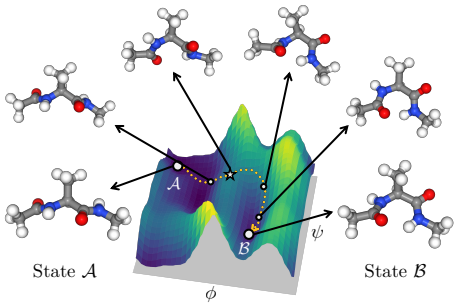


Figure 1: **Problem setup.** Potential energy landscape of Alanine Dipeptide, where the sampled path from state $\mathcal{A}$ to state $\mathcal{B}$ is highlighted in yellow dotted lines. We visualize snapshots in the transition path highlighted in white circles and the transition state highlighted in the white star.

Molecular dynamics. We consider a MD simulation on time interval $[0, T]$, i.e., the motion of a molecular state $\boldsymbol{X}_t = (\boldsymbol{R}_t, \boldsymbol{V}_t) \in \mathbb{R}^{6N}$ at time t where N is the number of atoms, $\boldsymbol{R}_t \in \mathbb{R}^{3N}$ is the atom-wise positions and $\boldsymbol{V}_t \in \mathbb{R}^{3N}$ is the atom-wise velocities. In particular, we adopt Langevin dynamics (Bussi & Parrinello, 2007) defined as the following SDE:

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{u}(\boldsymbol{X}_t)\mathrm{d}t + \Sigma \mathrm{d}\boldsymbol{W}_t, \;\; \boldsymbol{u}(\boldsymbol{X}_t) = \left(\boldsymbol{V}_t, -\frac{\nabla U(\boldsymbol{R}_t)}{\boldsymbol{m}} - \gamma \boldsymbol{V}_t\right), \;\; \Sigma = \mathrm{diag}\left(\boldsymbol{\zeta}, \sqrt{\frac{2\gamma k_B \lambda}{\boldsymbol{m}}}\right) \quad (1)$$

where U, $\boldsymbol{m}$, γ, k_B, λ, and $\boldsymbol{W}_t$ denote the potential energy function, the atom-wise masses, the friction term, the Boltzmann constant, the absolute temperature, and the Brownian motion, respectively, and $\boldsymbol{\zeta} \in \mathbb{R}^{3N}$ is a vector of positive infinitesimal values. MD in Equation (1) induces the path measure, denoted by $\mathbb{P}_0$, which refers to the positive measure defined on measurable subsets of the path space $\mathcal{C}([0, T]; \mathbb{R}^{6N})$ consisting of continuous functions $\boldsymbol{X} : [0, T] \to \mathbb{R}^{6N}$. The path (probability) measure $\mathbb{P}_0$ induced by MD assigns high probability to a set of the probable paths when solving MD.

Transition path sampling. One of the challenges in sampling transition paths through unbiased MD simulations is the meta-stability: a state remains trapped for a long time in the initial meta-stable state $\mathcal{A} \subseteq \mathbb{R}^{3N}$ before transitioning into a distinct meta-stable state $\mathcal{B} \subseteq \mathbb{R}^{3N}$. To capture the rare event where transition from $\mathcal{A}$ to $\mathcal{B}$ occurs, we constrain paths $\boldsymbol{X} = (\boldsymbol{X}_t)_{0 \leq t \leq T}$ sampled from unbiased MD to satisfy $\boldsymbol{R}_0 \in \mathcal{A}$, $\boldsymbol{R}_T \in \mathcal{B}$ for a fixed time T. Since the meta-stable state $\mathcal{A}$ and $\mathcal{B}$ are not well-specified for many molecular systems, we simplify this task by (1) fixing a local minima

$\boldsymbol{R}_{\mathcal{A}}, \boldsymbol{R}_{\mathcal{B}}$ of the potential energy function in the meta-stable states $\mathcal{A}, \mathcal{B}$ and (2) sampling a transition path $\boldsymbol{X}$ that starts from the state $\boldsymbol{R}_0 = \boldsymbol{R}_{\mathcal{A}}$ and ends at the vicinity of $\boldsymbol{R}_{\mathcal{B}}$.

To be specific, we aim to sample from the target path measure $\mathbb{Q}$, which is obtained by reweighting the path measure $\mathbb{P}_0$ with the (normalized) indicator function. The indicator function assigns zero weight to paths that do not reach the vicinity of the target position $\boldsymbol{R}_{\mathcal{B}}$. Formally, the reweighting function is called the Radon–Nikodym derivative defined as follows:

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_0}(\boldsymbol{X}) = \frac{1_{\mathcal{B}}(\boldsymbol{X})}{Z}, \quad 1_{\mathcal{B}}(\boldsymbol{X}) = \begin{cases} 1 & \text{if } \|\rho_T \cdot \boldsymbol{R}_{\mathcal{B}} - \boldsymbol{R}_T\| \leq \delta, \\ 0 & \text{otherwise,} \end{cases} \quad Z = \mathbb{E}_{\mathbb{P}_0}\left[1_{\mathcal{B}}(\boldsymbol{X})\right], \tag{2}$$

where $\cdot$ denotes group action associated with the $SE(3)$ space and $\rho_T \cdot \boldsymbol{R}_{\mathcal{B}}$ is the aligned target position by the optimal roto-translation $\rho_T \in SE(3)$ to minimize its Euclidean distance to $\boldsymbol{R}_T$, i.e., $\rho_T = \mathrm{argmin}_{\rho \in SE(3)} \|\boldsymbol{R}_T - \rho \cdot \boldsymbol{R}_{\mathcal{B}}\|$. Such a transformation can be obtained from the Kabsch algorithm in $O(N)$ complexity (Kabsch, 1976).

Note that one may consider naïve rejection sampling to sample transition paths, based on running unbiased MD to sample a path $\boldsymbol{X}$ from the path measure $\mathbb{P}_0$ and accepting if the path $\boldsymbol{X}$ arrives at the neighborhood of the position $\boldsymbol{R}_{\mathcal{B}}$ with the radius δ. However, this method does not scale to high-dimensional or low-temperature problems, since the sampled path by unbiased MD rarely reaches the target states due to the high energy barriers, i.e., the rejection ratio is too high.

3.2 Log-variance minimization

In this section, we propose our algorithm to amortize transition path sampling. Our key idea is to train a neural network to induce a path measure that matches the target path measure $\mathbb{Q}$, using the log-variance divergence (Nüsken & Richter, 2021) between the path measures. We propose a new training scheme to minimize the log-variance divergence based on learning the control variate of its gradient and a replay buffer to improve sample efficiency and diversity.

Amortizing transition path sampling with log-variance divergence. To match the target path measure $\mathbb{Q}$, we consider a biased MD defined by a policy $\boldsymbol{v}$ (or bias force $\boldsymbol{b}$) as the following SDE:

$$\mathrm{d}\boldsymbol{X}_t = (\boldsymbol{u}(\boldsymbol{X}_t) + \Sigma \boldsymbol{v}(\boldsymbol{X}_t))\mathrm{d}t + \Sigma \mathrm{d}\boldsymbol{W}_t, \quad \boldsymbol{v}(\boldsymbol{X}_t) = \Sigma^{-1}\left(\boldsymbol{0}, \frac{\boldsymbol{b}(\boldsymbol{X}_t)}{\boldsymbol{m}}\right). \tag{3}$$

We also let $\mathbb{P}_{\boldsymbol{v}}$ denote the path measure induced by the SDE. To amortize transition path sampling, we match the path measure $\mathbb{P}_{\boldsymbol{v}_\theta}$ of a parameterized policy $\boldsymbol{v}_\theta$ with the target path measure $\mathbb{Q}$ by minimizing the log-variance divergence:

$$D_{\mathrm{LV}}^{\mathbb{P}}(\mathbb{P}_{\boldsymbol{v}_\theta} \| \mathbb{Q}) = \mathbb{V}_{\mathbb{P}}\left[\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}\right] = \mathbb{E}_{\mathbb{P}}\left[\left(\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}} - \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}\right]\right)^2\right], \tag{4}$$

where $\mathbb{P}$ is an arbitrary reference path measure with $\mathbb{E}_{\mathbb{P}}[\log(\mathrm{d}\mathbb{Q}/\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta})] < \infty$. To express the log-variance divergence in detail, we let $\mathbb{P} = \mathbb{P}_{\tilde{\boldsymbol{v}}}$ for some policy $\tilde{\boldsymbol{v}}$ and apply the Girsanov's theorem to Equation (4), deriving the following formulation:

$$D_{\mathrm{LV}}^{\mathbb{P}_{\tilde{\boldsymbol{v}}}}(\mathbb{P}_{\boldsymbol{v}_\theta} \| \mathbb{Q}) = \mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[(F_{\boldsymbol{v}_\theta, \tilde{\boldsymbol{v}}} - \mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}[F_{\boldsymbol{v}_\theta, \tilde{\boldsymbol{v}}}])^2\right], \tag{5}$$

$$F_{\boldsymbol{v}_\theta, \tilde{\boldsymbol{v}}}(\boldsymbol{X}) = \frac{1}{2}\int_0^T \|\boldsymbol{v}_\theta(\boldsymbol{X}_t)\|^2 \mathrm{d}t - \int_0^T (\boldsymbol{v}_\theta \cdot \tilde{\boldsymbol{v}})(\boldsymbol{X}_t)\mathrm{d}t - \int_0^T \boldsymbol{v}_\theta(\boldsymbol{X}_t) \cdot \mathrm{d}\boldsymbol{W}_t + \log 1_{\mathcal{B}}(\boldsymbol{X}). \tag{6}$$

The first three terms in Equation (6) correspond to the deviation of the biased MD from the unbiased MD integrated over the path sampled from $\mathbb{P}_{\tilde{\boldsymbol{v}}}$. The last term reweights the unbiased MD to the target path measure $\mathbb{Q}$. As a result, minimizing Equation (5) could be thought as minimizing the variation between $\mathbb{P}_{\boldsymbol{v}_\theta}$ and $\mathbb{Q}$. We provide the full derivation in Appendix A.1. Compared to KL divergence, the log-variance divergence provides a robust gradient estimator and avoids differentiating through the SDE solver. (Richter et al., 2020; Nüsken & Richter, 2021).

Minimizing with learnable control variate. To minimize the log-variance divergence, we consider the following loss function that replaces the estimation of $\mathbb{E}_{\mathbb{P}_{\boldsymbol{v}_\theta}}[F_{\boldsymbol{v}_\theta, \boldsymbol{v}_\theta}]$ by learning a parameter w:

$$\mathcal{L}(\theta, w) = \mathbb{E}_{\mathbb{P}_{\boldsymbol{v}_\theta}}\left[(F_{\boldsymbol{v}_\theta, \boldsymbol{v}_\theta} - w)^2\right], \tag{7}$$

Algorithm 1 Training

1: Initialize an empty replay buffer $\hat{\mathcal{D}}$, an policy $\boldsymbol{v}_\theta$, a scalar parameter w, the number of rollout I and training per rollout J, and an annealing schedule $\lambda_{\text{start}} = \lambda_1 > \cdots > \lambda_I = \lambda_{\text{end}}$.
2: **for** $i = 1, \ldots, I$ **do**
3: Generate M paths $\{\boldsymbol{x}_{0:L}^{(m)}\}_{m=1}^M$ from the biased MD simulations with $\boldsymbol{v}_\theta$ at temperature λ_i.
4: Update the replay buffer $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{D}} \cup \{\boldsymbol{x}_{0:L}^{(m)}\}_{m=1}^M$.
5: **for** $j = 1, \ldots, J$ **do**
6: Sample K data $\{\boldsymbol{x}_{0:L}^{(k)}\}_{k=1}^K$ from $\hat{\mathcal{D}}$.
7: Update θ and w with the gradient of $\frac{1}{K}\sum_{k=1}^K \left(\log \frac{p_0(\boldsymbol{x}_{0:L}^{(k)}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L}^{(k)})}{p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L}^{(k)})} - w\right)^2$.
8: **end for**
9: **end for**

where w is a *control variate* that controls the variance of the gradient estimator of $\nabla_\theta \mathcal{L}(\theta, w)$ without changing the gradient. Note that we set $\tilde{\boldsymbol{v}} = \boldsymbol{v}_\theta$ in Equation (6), which implies that the gradient of Equation (7) coincides with the KL divergence (Richter et al., 2020; Nüsken & Richter, 2021). When optimized, the control variate w estimates the expectation $\mathbb{E}_{\mathbb{P}_{\boldsymbol{v}_\theta}}[F_{\boldsymbol{v}_\theta, \boldsymbol{v}_\theta}]$ since $\operatorname{argmin}_w \mathcal{L}(\theta, w) = \mathbb{E}_{\mathbb{P}_{\boldsymbol{v}_\theta}}[F_{\boldsymbol{v}_\theta, \boldsymbol{v}_\theta}]$. Thus, jointly optimizing (θ, w) with the gradient step can be interpreted as jointly minimizing log-variance divergence and estimating $\mathbb{E}_{\mathbb{P}_{\boldsymbol{v}_\theta}}[F_{\boldsymbol{v}_\theta, \boldsymbol{v}_\theta}]$ using w.

Off-policy training with replay buffer and simulated annealing. To leverage the degree of freedom in reference path measure for the log-variance divergence, we allow discrepancy between reference path measure and current path measure, called off-policy training, which is widely used in discrete-time reinforcement learning (Mnih et al., 2013; Bengio et al., 2021). For the sample efficiency, we reuse the samples with a replay buffer $\mathcal{D}$ which stores path samples from the path measure $\mathbb{P}_{\boldsymbol{v}_{\bar{\theta}}}$ associated with previous policies $\boldsymbol{v}_{\bar{\theta}}$. Our modified loss function $\mathcal{L}^{\mathcal{D}}$ with $\mathcal{D}$ is defined as follows:

$$\mathcal{L}^{\mathcal{D}}(\theta, w) = \mathbb{E}_{(\boldsymbol{v}_{\bar{\theta}}, \boldsymbol{X}) \sim \mathcal{D}}[(F_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{X}) - w)^2]. \tag{8}$$

Using the replay buffer also prevents mode collapse, using diverse paths from different path measures. Similar to other off-policy training algorithms (Malkin et al., 2022; Kim et al., 2023), we use simulated annealing to collect diverse paths that cross high energy barriers.

Discretization. To implement the algorithm, we discretize Equation (8). Given a discretization step size Δt, we consider the discretized paths $\boldsymbol{x}_{0:L} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$ of $\boldsymbol{X}$ from MD simulations where $L = T/\Delta t$ and $\boldsymbol{x}_\ell = \boldsymbol{X}(\ell \Delta t)$. In discrete cases, the discretized paths $\boldsymbol{x}_{0:L}$ from previous policies $\boldsymbol{v}_{\bar{\theta}}$ and their (gradient-detached) policy values $(\boldsymbol{v}_{\bar{\theta}}(\boldsymbol{x}_0), ..., \boldsymbol{v}_{\bar{\theta}}(\boldsymbol{x}_L))$ are used to approximate the value $F_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{X})$ in Equation (6) as follows:

$$\hat{F}_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{x}_{0:L}) = \frac{1}{2}\sum_{\ell=0}^{L-1} \|\boldsymbol{v}_\theta(\boldsymbol{x}_\ell)\|^2 \Delta t - \sum_{\ell=0}^{L-1} (\boldsymbol{v}_\theta \cdot \boldsymbol{v}_{\bar{\theta}})(\boldsymbol{x}_\ell) \Delta t - \sum_{\ell=0}^{L-1} \boldsymbol{v}_\theta(\boldsymbol{x}_\ell) \cdot \boldsymbol{\epsilon}_\ell + \log 1_{\mathcal{B}}(\boldsymbol{x}_{0:L}), \tag{9}$$

where the noise $\boldsymbol{\epsilon}_\ell = \Sigma^{-1}(\boldsymbol{x}_{\ell+1} - \boldsymbol{x}_\ell - (\boldsymbol{u}(\boldsymbol{x}_\ell) + \Sigma \boldsymbol{v}_{\bar{\theta}}(\boldsymbol{x}_\ell))\Delta t)$ is the discretized Brownian motions of the Langevin dynamics with policy $\boldsymbol{v}_{\bar{\theta}}$. For implementation, we further derive a simple discretized loss of Equation (8) from Equation (9) as follows:

$$\mathbb{E}_{\boldsymbol{x}_{0:L} \sim \hat{\mathcal{D}}}\left[\left(\log \frac{p_0(\boldsymbol{x}_{0:L}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L})}{p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L})} - w\right)^2\right], \tag{10}$$

where the buffer $\hat{\mathcal{D}}$ stores paths $\boldsymbol{x}_{0:L}$ sampled from the previous policies, and p_0 and $p_{\boldsymbol{v}_\theta}$ denote discrete time transition probability induced by Equations (1) and (3), respectively. We provide a formal derivation of the discretized loss in Appendix A.2. Note that the same objective was derived in the name of relative trajectory balance by Venkatraman et al. (2024).

We describe our training algorithm in Algorithm 1. Overall, our off-policy training algorithm iterates through four steps: (1) sampling paths from the biased MD simulation with current policy $\boldsymbol{v}_\theta$ at high temperature, (2) storing sampled paths in the replay buffer $\hat{\mathcal{D}}$, (3) sampling a batch of the paths from

the replay buffer, and (4) training current policy $\boldsymbol{v}_\theta$ by minimizing the loss in Equation (10). After minimization, biased MD simulation can directly sample transition paths by amortizing inference in the target path measure.

3.3 Parameterization for high dimensional tasks

In this section, we introduce new parameterizations of the bias force and the indicator function for high-dimensional tasks. Our parameterization is designed around alleviating the problem of sparse training signal, where the model struggle to collect meaningful paths that end near the target meta-stable state in training. This problem is especially severe in large molecules.

Bias force parameterization. To frequently sample the meaningful paths, we aim to parameterize the bias force which guarantees to reduce the distance between the current molecular state and the target meta-stable state for every MD step. This is achieved by predicting the atom-wise positive scaling factor of the direction to the aligned target meta-stable state rather than predicting force or potential directly. Moreover, we design the bias force to satisfy roto-translational equivariance to the current molecular state input $\boldsymbol{X}_t$, aligning with the symmetry of the transition path sampling problem for better generalization.

To be specific, we use a roto-translation $\rho_t \in SE(3)$ to align $\boldsymbol{R}_\mathcal{B}$ with $\boldsymbol{R}_t$, as we do for indicator function in Equation (2). To achieve $SE(3)$ equivariant, we parameterize the bias force as follows:

$$\boldsymbol{b}(\boldsymbol{X}_t) = \mathrm{diag}(\boldsymbol{s}_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t))(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t), \tag{11}$$

where $\boldsymbol{s}_\theta(\cdot) \in \mathbb{R}_+^{3N}$ is a neural network constrained to have positive output elements and predicts atom-wise scaling factors. We note that the bias force (divided by atom-wise masses) is positively correlated with the direction to target state, i.e., $(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) > 0$.

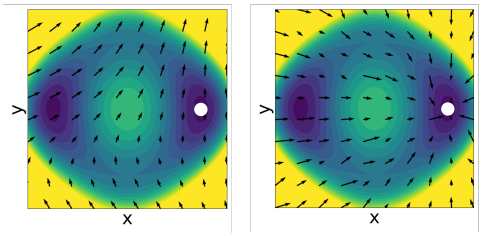
To formalize the benefit of positive correlation between the bias force and the direction to the target state, one can prove that there always exists a small enough step size Δt that decrease the distance between the current state $\boldsymbol{R}_t$ and the aligned target state $\rho'_{t+\Delta t} \cdot \boldsymbol{R}_\mathcal{B}$, i.e.,

$$\|\rho'_{t+\Delta t} \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\| < \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|, \tag{12}$$

where $\boldsymbol{R}'_{t+\Delta t} = \boldsymbol{R}_t + \boldsymbol{b}(\boldsymbol{X}_t)\Delta t/\boldsymbol{m}$ is the position updated by the bias force with step size Δt and $\rho'_{t+\Delta t} = \mathrm{argmin}_{\rho \in SE(3)} \|\boldsymbol{R}'_{t+\Delta t} - \rho \cdot \boldsymbol{R}_\mathcal{B}\|$. We formalize this statement and provide the proof of Equation (12) in Appendix A.3.

In the experiments, we also consider other equivariant parameterizations that are less constrained: (1) directly predicting the equivariant bias force by $\rho_t \cdot \boldsymbol{b}_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t) \in \mathbb{R}^{3N}$ and (2) predicting the invariant bias potential $b_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t) \in \mathbb{R}$ and taking gradient of it $\nabla b_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t)$. We observe these two parameterizations to be useful for low-dimensional tasks but struggle to produce meaningful paths in large molecules during training. As shown in Figure 2, bias forces with the positive scaling parameterization are positively correlated with the direction to the target position (white circle) regardless of network parameters, unlike force parameterizations.

Indicator function parameterization. We propose to relax the indicator function $1_\mathcal{B}$ as a radial basis function (RBF) kernel $\tilde{1}_\mathcal{B}(\boldsymbol{X}) = k(\boldsymbol{R}_T, \rho^{-1} \cdot \boldsymbol{R}_\mathcal{B}; \sigma^2)$ which measures the similarity between two positions where $\sigma > 0$ controls the degree of relaxation. The range of RBF kernel k is bounded by the interval $(0, 1]$ so that $\log \tilde{1}_\mathcal{B}(\boldsymbol{X})$ is well-defined and $\tilde{1}_\mathcal{B}(\boldsymbol{X})$ represents the binary indicator function smoothly. To capture a high training signal from subtrajectories of sampled paths, we propose to take maximum over RBF kernel values of all intermediate states by $\tilde{1}_\mathcal{B}^{\max}(\boldsymbol{X}) = \max_{t \in [0,T]} k(\boldsymbol{R}_t, \tilde{\boldsymbol{R}}_\mathcal{B}; \sigma^2)$. To extract the subtrajectory with a high training signal, we can truncate the paths at the time that maximizes RBF kernel values, allowing variable path lengths. Notably, the relaxed indicator function is $SE(3)$ invariant to $\boldsymbol{R}_t$ because of the Kabsch algorithm.



(a) Direct prediction (b) Positive scaling

Figure 2: **Visualization of the bias force fields from untrained neural networks of two different bias force parameterizations**. **(a)** directly predicting the bias force and **(b)** predicting the positive scaling factors of direction to the target position (white circle).

Table 1: **Benchmark scores on the double-well system and three real-world molecular systems**: Alanine Dipeptide, Polyproline Helix, and Chignolin. All metrics are averaged over 1024 paths for the double-well system, and 64 paths for real-world molecular systems. ETS is computed for paths that hit the target meta-stable state, and the best results are highlighted in bold. TPS-DPS predicting the bias force, potential, and atom-wise scaling are denoted by (F), (P), and (S), respectively. UMD (λ) denotes unbiased MD with temperature λ and SMD (k) denotes steered MD with the force constant k. Unless otherwise specified, paths are generated by MD simulation at 1200K for double-well and 300K for real-world molecules. $*$ denotes results reported by Du et al. (2024).

Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
Double-well			
UMD (1200K)	2.21 ± 0.10	0.00	-
UMD (2400K)	2.11 ± 0.38	3.03	1.69 ± 0.31
UMD (3600K)	1.85 ± 0.68	12.60	2.12 ± 0.41
UMD (4800K)	1.54 ± 0.81	21.58	2.77 ± 0.69
Two-way shooting	0.26 ± 0.05	**100.00**	1.41 ± 0.16
SMD (0.5)	0.98 ± 0.90	52.15	1.54 ± 0.21
SMD (1)	0.14 ± 0.08	99.80	1.85 ± 0.16
TPS-DPS (F, Ours)	$\mathbf{0.01 \pm 0.02}$	**99.90**	1.38 ± 0.16
TPS-DPS (P, Ours)	$\mathbf{0.01 \pm 0.03}$	99.71	$\mathbf{1.36 \pm 0.15}$
TPS-DPS (S, Ours)	$\mathbf{0.01 \pm 0.03}$	99.80	1.73 ± 0.20
Polyproline Helix			
UMD (300K)	2.22 ± 0.11	0.00	-
UMD (1200K)	1.38 ± 0.45	10.94	1010.28 ± 38.44
SMD (5k)	1.68 ± 0.17	54.69	350.58 ± 14.36
SMD (10k)	1.26 ± 0.06	**100.00**	355.62 ± 14.83
PIPS (F)	2.64 ± 0.15	0.00	-
PIPS (P)	1.85 ± 0.13	93.75	574.66 ± 20.49
TPS-DPS (F, Ours)	1.53 ± 0.12	98.44	418.17 ± 45.54
TPS-DPS (P, Ours)	1.35 ± 0.12	**100.00**	345.00 ± 32.58
TPS-DPS (S, Ours)	$\mathbf{1.17 \pm 0.02}$	**100.00**	$\mathbf{342.00 \pm 20.28}$

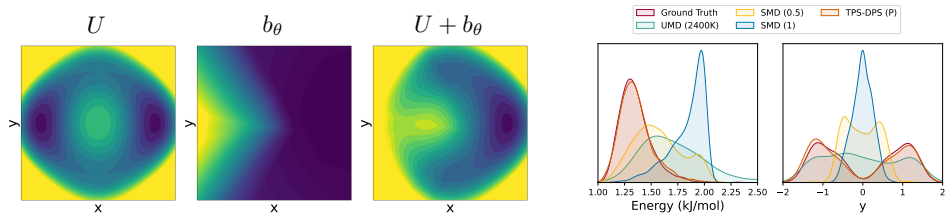
Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
Alanine Dipeptide			
UMD (300K)	1.59 ± 0.15	0.00	-
UMD (3600K)	1.19 ± 0.32	6.25	812.47 ± 148.80
Doob's Lagrangian*	-	-	69.26 ± 0.21
Two-way shooting	0.38 ± 0.24	**100.00**	527.66 ± 450.51
SMD (20)	0.56 ± 0.27	54.69	78.40 ± 12.76
PIPS (F)	0.66 ± 0.15	43.75	28.17 ± 10.86
PIPS (P)	1.66 ± 0.03	0.00	-
TPS-DPS (F, Ours)	$\mathbf{0.16 \pm 0.06}$	**92.19**	19.82 ± 15.88
TPS-DPS (P, Ours)	$\mathbf{0.16 \pm 0.10}$	87.50	$\mathbf{18.37 \pm 10.86}$
TPS-DPS (S, Ours)	0.25 ± 0.20	76.00	22.79 ± 13.57
Chignolin			
UMD (300K)	7.98 ± 0.41	0.00	-
UMD (1200K)	7.23 ± 0.93	1.56	388.17
SMD (10k)	1.26 ± 0.31	6.25	-527.95 ± 93.58
SMD (15k)	$\mathbf{1.17 \pm 0.31}$	23.44	-237.15 ± 122.29
PIPS (F)	4.66 ± 0.17	0.00	-
PIPS (P)	4.67 ± 0.32	0.00	-
TPS-DPS (F, Ours)	4.41 ± 0.49	0.00	-
TPS-DPS (P, Ours)	3.87 ± 0.42	0.00	-
TPS-DPS (S, Ours)	$\mathbf{1.17 \pm 0.66}$	**59.38**	$\mathbf{-780.18 \pm 216.93}$

4 EXPERIMENT

In this section, we compare our method, called TPS-DPS, with both classical and ML approaches, assessing the accuracy and diversity of sampled transition paths. We begin with a synthetic double-well system at 1200K, followed by three real-world molecular systems with various numbers of amino acids: Alanine Dipeptide, Polyproline Helix, and Chignolin. Additionally, we conduct ablation studies to validate the effectiveness of each component in our method. All real-world molecular systems are simulated using the OpenMM library (Eastman et al., 2023). Details on OpenMM simulation and model configurations are provided in Appendices B.1 and B.2, respectively. In Appendix C, we further evaluate our method on three fast folding proteins: Trpcage, BBA, and BBL (Lindorff-Larsen et al., 2011). In Appendix D, we analyze the time complexity of TPS-DPS and evaluate the number of energy evaluations and runtime in training and inference time.

Evaluation Metrics. We consider three metrics to evaluate models: RMSD, THP, and ETS. The root mean square distance (RMSD) measures the ability to produce final positions of paths close to the target position $\boldsymbol{R}_{\mathcal{B}}$, with the final positions aligned to the target. The target hit percentage (THP) measures the ability to produce final positions of paths that successfully arrive at the target meta-stable state $\mathcal{B}$. Finally, the energy of the transition state (ETS) measures the ability to identify probable transition states. For further details, refer to Appendix B.3.

Baselines. We compare TPS-DPS with both non-ML and ML baselines. For non-ML baselines, we consider unbiased MD (UMD) with various temperatures, variable length two-way shooting with uniform shooting point selection, and steered MD (SMD; Schlitter et al., 1994; Izrailev et al., 1999) with various force constants k and collective variables (CVs). For ML baselines, we consider a CV-free transition path sampling method, path integral path sampling (PIPS; Holdijk et al., 2024) which also trains a bias force by minimizing the KL divergence between path measures induced by the biased MD and the target path measure. For simplicity, we denote parameterizations for predicting force, potential, and atom-wise scaling factors as (F), (P) and (S), respectively.



(a) Double-well energy landscapes (b) Distributions of the energy and y

Figure 3: **Visualization of potential energy landscapes and distributions of the double-well system.** (a) Visualization of the neural bias potential b_θ of TPS-DPS (P). (b) Distributions of the potential energy and y coordinate of transition states from 1024 transition paths sampled by methods.

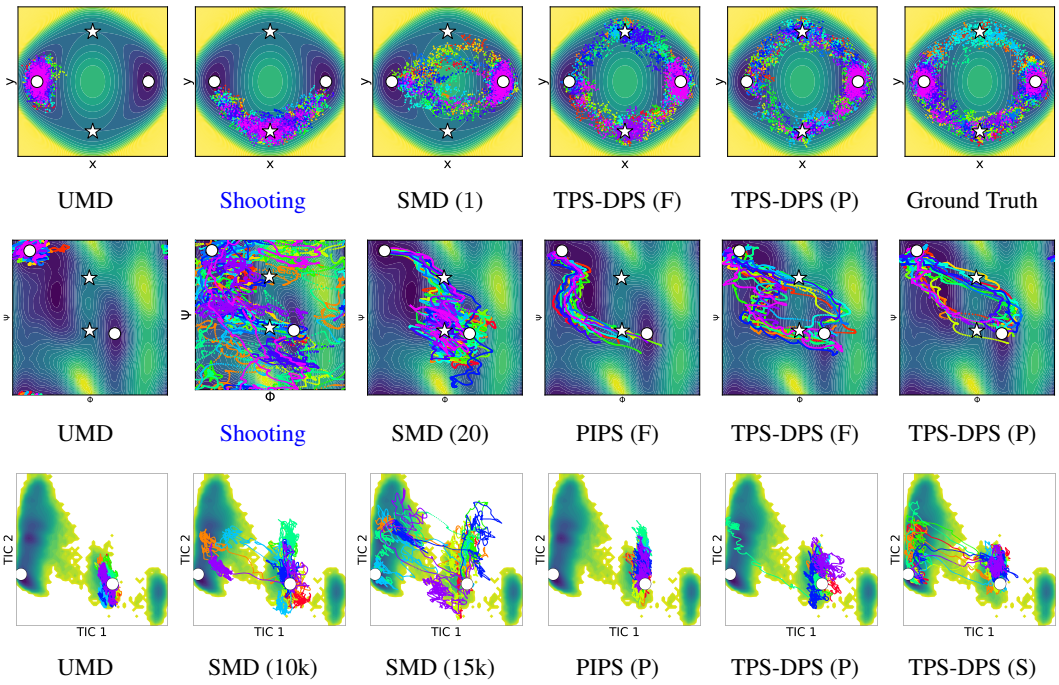
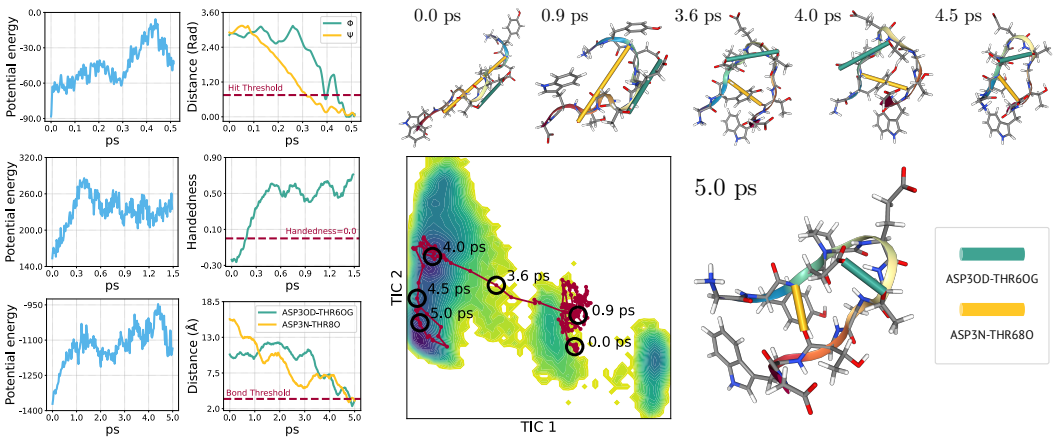


Figure 4: **16 sampled transition paths of three systems.** White circles indicate meta-stable states and white stars indicate saddle points. Double-well on potential energy landscape (top). Alanine Dipeptide on Ramachandran plot (middle). Chignolin on top two TICA components (bottom).

4.1 Double-well system

We begin by evaluating our method on a two-dimensional synthetic system, i.e., a double-well potential at 1200K. This system has two global minima representing the meta-stable states, and two reaction pathways via saddle points. We sample transition paths from the left state $\boldsymbol{R}_\mathcal{A}$ to the right meta-stable states $\mathcal{B} = \{\boldsymbol{R} \mid \|\boldsymbol{R} - \boldsymbol{R}_\mathcal{B}\| < 0.5\}$. We collect ground truth path ensembles by rejection sampling which proposes paths sampled from the unbiased MD simulations and accepts if the final states are in the target meta-stable states $\mathcal{B}$. We provide more details on the system in Appendix B.4.

In Table 1, Figure 3, and Figure 4, TPS-DPS outperforms baselines regardless of the bias force design, generates transition paths more similar to the ground truth than baselines. In Figure 3, the neural bias potential accelerates the transition by increasing the potential energy near the initial meta-stable state while decreasing the potential energy near the two energy barriers. Moreover, the distribution of energy and y coordinate of the transition states from TPS-DPS is closest to the ground truth compared with other baselines, successfully capturing two reaction channels. In Figure 4, UMD at 1200K fails to escape the initial state and SMD struggles to pass the saddle points.



(a) Potential energy and CVs over time.

(b) Visualization of hydrogen bond formation in Chignolin.

Figure 5: **Qualitative evaluation on transition path sampled from TPS-DPS. (a)** Potential energies and collective variables (CVs) of Alanine dipeptide, Polyporline Helix, and Chignolin from top to bottom. For CVs, we plot the distance of the backbone dihedral angles between the current state and the target state for Alanine Dipeptide, the handedness of the backbone for Polyproline Helix, and the hydrogen bond distance for Chignolin. **(b)** Visualization of hydrogen bond formation in Chignolin. We highlighted each hydrogen bond in green and yellow.

4.2 Alanine Dipeptide Conformation Change

We first consider Alanine Dipeptide for a real-world molecule consisting of two alanine residues, sampling transition paths from the $C5$ (upper left) to the $C7ax$ (lower right) as seen in Figure 4. The target meta-stable states are defined as $\mathcal{B} = \{\boldsymbol{R} \mid \|\xi(\boldsymbol{R}) - \xi(\boldsymbol{R}_{\mathcal{B}})\| < 0.75\}$, where $\xi(\boldsymbol{R}) = (\phi, \psi)$ is a well-known collective variable which consists of two backbone dihedral angles. Alanine Dipeptide has two reaction channels between the $C5$ and $C7ax$ state passing through the saddle points.

In Table 1 and Figure 4, our method shows superior performance regardless of the bias force design and successfully generates diverse transition paths that pass two reaction channels. UMD at 300K fails to escape the initial state, SMD with the two backbone torsion CV generates transition paths with less probable transition states, and two-way shooting struggles to find plausible transition states. PIPS generates transition paths of only one reaction channel, suffering from mode collapse.

4.3 Polyproline Helix Isomerization

Next, we consider a more complex molecule Polyproline Helix than Alanine Dipeptide which consists of three proline residues. We sample transition paths from the left-handed state (PPII) in the *cis*-configuration to the right-handed state (PPI) in the *trans*-configuration. We define the target meta-stable state based on handedness H as $\mathcal{B} = \{\boldsymbol{R} \mid H(\boldsymbol{R}) > 0\}$. For the formal definition of the handedness, refer to Appendix B.4. In Table 1, TPS-DPS consistently outperforms baselines regardless of bias force parameterization.

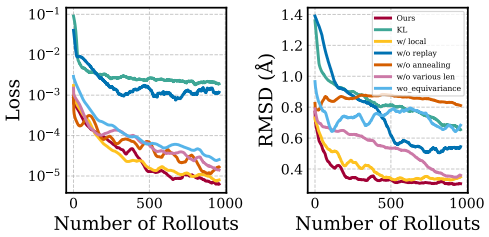
4.4 Chignolin Folding

Finally, we consider a challenging molecule, Chignolin, an artificial protein consisting of 10 amino acids (Honda et al., 2004), which folds into a β-hairpin structure by hydrogen bonds. We sample transition paths from the unfolded state (right) to the folded state (left) as seen in Figure 4. We define the target meta-stable state $\mathcal{B} = \{\boldsymbol{R} \mid \|\xi(\boldsymbol{R}) - \xi(\boldsymbol{R}_{\mathcal{B}})\| < 0.75\}$ where ξ consists of the top two components of time-lagged independent component analysis (TICA; Pérez-Hernández et al., 2013). We further describe TICA in Appendix B.4.

In Table 1 and Figure 4, Only TPS-DPS (S) successfully samples transition paths that pass probable transition states. While SMD hits the target meta-stable, its transition paths pass less probable transition states. UMD, PIPS, TPS-DPS (F), and TPS-DPS (P) fail to hit the target meta-stable state.

Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
Ours	**0.16 ± 0.06**	**92.19**	**19.82 ± 15.88**
w/ KL	0.43 ± 0.34	53.12	27.88 ± 14.38
w/ local	0.24 ± 0.15	73.44	22.53 ± 14.45
w/o replay	0.33 ± 0.27	64.06	24.38 ± 12.31
w/o annealing	0.67 ± 0.21	9.38	69.86 ± 30.15
w/o various len	0.23 ± 0.11	75.00	29.49 ± 14.13
w/o equivariance	0.34 ± 0.17	56.25	22.12 ± 16.96

(a) Component-wise performance on Alanine Dipeptide.



(b) Loss and RMSD curves over rollouts.

Figure 6: **Ablation studies on the components of TPS-DPS in the Alanine Dipeptide.** **(a)** Benchmark scores on Alanine Dipeptide. **(b)** Loss and RMSD curves averaged over 8 different seeds.

In Figure 5, we validate the sampled paths using the potential energy and donor-acceptance distance of the two key hydrogen bonds. The sampled path forms two key hydrogen bonds, ASP3OD-THR6OG and ASP3N-THR8O, reducing the donor-acceptance distance below the threshold 3.5Å.

4.5 Ablation Study

Effectiveness of algorithmic components. We conduct ablation studies to verify the effectiveness of the five proposed components: log-variance loss, learnable control variate, replay buffer, simulated annealing, maximum over RBF values for various path lengths, and $SE(3)$ equivariance. To be specific, we (1) replace our loss with the KL divergence, (2) replace the learnable control variate with the local control variate used in Nüsken & Richter (2021), (3) remove the replay buffer and use data only from the current policy, (4) use only one temperature λ, (5) remove maximum operation over RBF kernel values using only the final state, (6) remove the Kabsch algorithm.

As seen in Figure 6, all the proposed components improve performance. Our loss is smaller than the KL divergence by more than two orders of magnitude and significantly improves performance. Learning the control variate slightly improves performance, showing that utilizing data from previous policies is effective. The replay buffer significantly improves training efficiency, and shows that the large performance gap between our loss and KL divergence comes from the replay buffer. Simulated annealing for biased MD simulation is critical to finding transition paths. RMSD does not decrease without simulated annealing while loss decreases significantly. For the relaxed indicator function, maximum operation accelerates convergence and improves performance with frequent training signals from the subtrajectories. Leveraging the symmetry of the bias force with Kabsch algorithm improves performance. We further compare with reverse KL divergence in Appendix E.

5 Conclusion

In this work, we introduced a novel CV-free diffusion path sampler, called TPS-DPS, to amortize cost of sampling transition paths, using log-variance divergence with the learnable control variate and off-policy training with the replay buffer and simulated annealing. We also propose new $SE(3)$ equivariant scale-based parameterization of bias force and relaxed indicator function for frequent training signals. Evaluations on synthetic double-well and real-molecule systems such as Alanine Dipeptide, Polyproline Helix, and Chignolin demonstrate superior accuracy and diversity of our model compared to both classical and ML approaches.

Limitation. While our experiments show promise, they are limited to relatively small (up to 50 amino acids) and fast-folding proteins. The application of our method remains challenging for real-world proteins with up to 500 amino acids since the MD simulations required for training our model take significantly longer. For example, our algorithm can be trained for BBL protein with 47 amino acids under 170 GPU hours on a single A5000 GPU. However, training our algorithm on Glutamine Synthetase (Yamashita et al., 1989) with 469 amino acids would take at least 1700 GPU hours. Furthermore, our method does not generalize across unseen pairs of meta-stable states or different molecular systems. These points to an interesting venue for future research, which would be more appealing for practical applications in drug discovery or material design.

REFERENCES

Seihwan Ahn, Mannkyu Hong, Mahesh Sundararajan, Daniel H Ess, and Mu-Hyun Baik. Design and optimization of catalysts based on mechanistic insights derived from quantum chemical reaction modeling. *Chemical reviews*, 119(11):6509–6560, 2019.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Davide Branduardi, Giovanni Bussi, and Michele Parrinello. Metadynamics with adaptive gaussians. *Journal of chemical theory and computation*, 8(7):2247–2254, 2012.

Giovanni Bussi and Davide Branduardi. Free-energy calculations with metadynamics: Theory and practice. *Reviews in Computational Chemistry Volume 28*, pp. 1–49, 2015.

Giovanni Bussi and Michele Parrinello. Accurate sampling using langevin dynamics. *Physical Review E*, 75(5):056707, 2007.

Jeffrey Comer, James C Gumbart, Jérôme Hénin, Tony Lelièvre, Andrew Pohorille, and Christophe Chipot. The adaptive biasing force method: Everything you always wanted to know but were afraid to ask. *The Journal of Physical Chemistry B*, 119(3):1129–1151, 2015.

Avishek Das, Dominic C Rose, Juan P Garrahan, and David T Limmer. Reinforcement learning of rare diffusive dynamics. *The Journal of Chemical Physics*, 155(13), 2021.

Christoph Dellago, Peter G Bolhuis, and David Chandler. Efficient transition path sampling: Application to lennard-jones cluster rearrangements. *The Journal of chemical physics*, 108(22):9236–9245, 1998.

Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noe, Carla P Gomes, Alan Aspuru-Guzik, and Kirill Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. In *ICML 2024 AI for Science Workshop*, 2024.

Chenru Duan, Yuanqi Du, Haojun Jia, and Heather J Kulik. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *Nature Computational Science*, 3(12):1045–1055, 2023.

Peter Eastman, Raimondas Galvelis, Raúl P Peláez, Charlles RA Abreu, Stephen E Farr, Emilio Gallicchio, Anton Gorenko, Michael M Henry, Frank Hu, Jing Huang, et al. Openmm 8: Molecular dynamics simulation with machine learning potentials. *The Journal of Physical Chemistry B*, 128 (1):109–116, 2023.

Ron Elber. Perspective: Computer simulations of long time dynamics. *The Journal of chemical physics*, 144(6), 2016.

Bernd Ensing, Marco De Vivo, Zhiwei Liu, Preston Moore, and Michael L Klein. Metadynamics as a tool for exploring free energy landscapes of chemical reactions. *Accounts of chemical research*, 39 (2):73–81, 2006.

Jérôme Hénin, Tony Lelièvre, Michael R Shirts, Omar Valsson, and Lucie Delemotte. Enhanced sampling methods for molecular dynamics simulations [article v1. 0]. *Living Journal of Computational Molecular Science*, 4(1):1583–1583, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Berend Ensing, and Max Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths. *Advances in Neural Information Processing Systems*, 36, 2024.

Shinya Honda, Kazuhiko Yamasaki, Yoshito Sawada, and Hisayuki Morii. 10 residue folded peptide designed by segment statistics. *Structure*, 12(8):1507–1518, 2004.

Xinru Hua, Rasool Ahmad, Jose Blanchet, and Wei Cai. Accelerated sampling of rare events using a neural network bias potential. *arXiv preprint arXiv:2401.06936*, 2024.

Michele Invernizzi and Michele Parrinello. Rethinking metadynamics: from bias potentials to probability distributions. *The journal of physical chemistry letters*, 11(7):2731–2736, 2020.

Sergei Izrailev, Sergey Stepaniants, Barry Isralewitz, Dorina Kosztin, Hui Lu, Ferenc Molnar, Willy Wriggers, and Klaus Schulten. Steered molecular dynamics. In *Computational Molecular Dynamics: Challenges, Methods, Ideas: Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, pp. 39–65. Springer, 1999.

Bowen Jing, Hannes Stark, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*, 2024.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Johannes Kästner. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(6):932–942, 2011.

Minsu Kim, Joohwan Ko, Dinghuai Zhang, Ling Pan, Taeyoung Yun, Woo Chang Kim, Jinkyoo Park, and Yoshua Bengio. Learning to scale logits for temperature-conditional gflownets. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Seonghwan Kim, Jeheon Woo, and Woo Youn Kim. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. *Nature Communications*, 15(1):341, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *The Second International Conference on Learning Representations*, 2013.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Juyong Lee, In-Ho Lee, InSuk Joung, Jooyoung Lee, and Bernard R Brooks. Finding multiple reaction pathways via global optimization of action. *Nature Communications*, 8(1):15443, 2017.

Tony Lelièvre, Geneviève Robin, Innas Sekkat, Gabriel Stoltz, and Gabriel Victorino Cardoso. Generative methods for sampling transition paths in molecular dynamics. *ESAIM: Proceedings and Surveys*, 73:238–256, 2023.

Kresten Lindorff-Larsen, Stefano Piana, Kim Palmo, Paul Maragakis, John L Klepeis, Ron O Dror, and David E Shaw. Improved side-chain torsion potentials for the amber ff99sb protein force field. *Proteins*, 78:1950–1958, 2010.

Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

JA Maier, C Martinez, K Kasavajhala, L Wickstrom, KE Hauser, and C Simmerling ff14SB. Improving the accuracy of protein side chain and backbone parameters from ff99sb., 2015, 11. *DOI: https://doi. org/10.1021/acs. jctc. 5b00255*, pp. 3696–3713, 2015.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35: 5955–5967, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

Mahmoud Moradi, Volodymyr Babin, Christopher Roland, Thomas A Darden, and Celeste Sagui. Conformations and free energy landscapes of polyproline peptides. *Proceedings of the National Academy of Sciences*, 106(49):20746–20751, 2009.

Adrian J Mulholland. Modelling enzyme reaction mechanisms, specificity and catalysis. *Drug discovery today*, 10(20):1393–1402, 2005.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021.

Philip Pechukas. Transition state theory. *Annual Review of Physical Chemistry*, 32(1):159–177, 1981.

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1), 2013.

Magnus Petersen, Gemma Roig, and Roberto Covino. Dynamicsdiffusion: Generating and rare event sampling of molecular dynamic trajectories using diffusion models. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Stefano Piana, Kresten Lindorff-Larsen, and David E Shaw. Protein folding kinetics and thermodynamics from atomistic simulation. *Proceedings of the National Academy of Sciences*, 109(44): 17845–17850, 2012.

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h4pNROsO06.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco Ruiz, and Omer Deniz Akyildiz. Vargrad: a low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33:13481–13492, 2020.

Martin K Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé. Pyemma 2: A software package for estimation, validation, and analysis of markov models. *Journal of chemical theory and computation*, 11(11):5525–5542, 2015.

Jürgen Schlitter, Michael Engels, and Peter Krüger. Targeted molecular dynamics: a new approach for searching pathways of conformational transitions. *Journal of molecular graphics*, 12(2):84–89, 1994.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Martin Sipka, Johannes CB Dietschreit, Lukáš Grajciar, and Rafael Gómez-Bombarelli. Differentiable simulations for enhanced sampling of rare events. In *International Conference on Machine Learning*, pp. 31990–32007. PMLR, 2023.

David A Sivak, John D Chodera, and Gavin E Crooks. Time step rescaling recovers continuous-time dynamical properties for discrete-time langevin integration of nonequilibrium systems. *The Journal of Physical Chemistry B*, 118(24):6466–6474, 2014.

Evan Walter Clark Spotte-Smith, Ronald L Kam, Daniel Barter, Xiaowei Xie, Tingzheng Hou, Shyam Dwaraknath, Samuel M Blau, and Kristin A Persson. Toward a mechanistic model of solid–electrolyte interphase formation and evolution in lithium-ion batteries. *ACS Energy Letters*, 7(4):1446–1453, 2022.

Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of computational physics*, 23(2):187–199, 1977.

Luke Triplett and Jianfeng Lu. Diffusion methods for generating transition paths. *arXiv preprint arXiv:2309.10276*, 2023.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

MM Yamashita, RJ Almassy, CA Janson, D Cascio, and D Eisenberg. Refined atomic model of glutamine synthetase at 3.5 å resolution. *Journal of Biological Chemistry*, 264(30):17681–17690, 1989.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y.

Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *2015 International joint conference on neural networks (IJCNN)*, pp. 1–4. IEEE, 2015.

A METHOD DETAILS

A.1 LOG VARIANCE FORMULATION

In this section, we derive Equation (5) from Equation (4) to get the explicit expression for log-variance divergence in terms of SDE in Equation (1) and Equation (3). We refer to Nüsken & Richter (2021, Appendix A.1) for the derivation in more general settings.

Our goal is to derive that

$$\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[\left(\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}-\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}\right]\right)^2\right]=\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[(F_{\boldsymbol{v}_\theta,\tilde{\boldsymbol{v}}}-\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}[F_{\boldsymbol{v}_\theta,\tilde{\boldsymbol{v}}}])^2\right], \tag{13}$$

To this end, we focus on calculating $\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}(\boldsymbol{X})$ when $\boldsymbol{X}\sim\mathbb{P}_{\tilde{\boldsymbol{v}}}$. Following (Nüsken & Richter, 2021, Lemma A.1), we apply Girsanov's Theorem to calculate the Radon-Nikodym derivative $\frac{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}{\mathrm{d}\mathbb{P}_0}$ as follows:

$$\frac{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}{\mathrm{d}\mathbb{P}_0}(\boldsymbol{X})=\exp\left(\int_0^T(\boldsymbol{v}_\theta^T\Sigma^{-1})(\boldsymbol{X}_t)\cdot\mathrm{d}\boldsymbol{X}_t-\int_0^T(\Sigma^{-1}\boldsymbol{u}\cdot\boldsymbol{v}_\theta)(\boldsymbol{X}_t)\mathrm{d}t-\frac{1}{2}\int_0^T\|\boldsymbol{v}_\theta(\boldsymbol{X}_t)\|^2\mathrm{d}t\right). \tag{14}$$

Since the state $\boldsymbol{X}_t$ follows the SDE $\mathrm{d}\boldsymbol{X}_t=(\boldsymbol{u}(\boldsymbol{X}_t)+\Sigma\tilde{\boldsymbol{v}}(\boldsymbol{X}_t))\mathrm{d}t+\Sigma\mathrm{d}\boldsymbol{W}_t$. We plug it into Equation (14) and utilize the definition of the target path measure $\mathbb{Q}$ in Equation (2) to compute $\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_0}$ as follows:

$$\begin{aligned}
\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}(\boldsymbol{X})&=\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_0}\frac{\mathrm{d}\mathbb{P}_0}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}(\boldsymbol{X}) && (15)\\
&=\log 1_{\mathcal{B}}(\boldsymbol{X})-\log Z-\int_0^T(\boldsymbol{v}_\theta^T\Sigma^{-1})(\boldsymbol{X}_t)\cdot\mathrm{d}\boldsymbol{X}_t && (16)\\
&\quad+\int_0^T(\Sigma^{-1}\boldsymbol{u}\cdot\boldsymbol{v}_\theta)(\boldsymbol{X}_t)\mathrm{d}t+\frac{1}{2}\int_0^T\|\boldsymbol{v}_\theta(\boldsymbol{X}_t)\|^2\mathrm{d}t && (17)\\
&=\log 1_{\mathcal{B}}(\boldsymbol{X})-\log Z-\int_0^T(\boldsymbol{v}_\theta\cdot\tilde{\boldsymbol{v}})(\boldsymbol{X}_t)\mathrm{d}t && (18)\\
&\quad-\int_0^T\boldsymbol{v}_\theta(\boldsymbol{X}_t)\cdot\mathrm{d}\boldsymbol{W}_t+\frac{1}{2}\int_0^T\|\boldsymbol{v}_\theta(\boldsymbol{X}_t)\|^2\mathrm{d}t && (19)\\
&=F_{\boldsymbol{v}_\theta,\tilde{\boldsymbol{v}}}(\boldsymbol{X})-\log Z && (20)
\end{aligned}$$

Since $\log Z$ is the constant, it is canceled out in the log-variance divergence as follows:

$$\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[\left(\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}-\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[\log\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}_{\boldsymbol{v}_\theta}}\right]\right)^2\right]=\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}\left[(F_{\boldsymbol{v}_\theta,\tilde{\boldsymbol{v}}}-\mathbb{E}_{\mathbb{P}_{\tilde{\boldsymbol{v}}}}[F_{\boldsymbol{v}_\theta,\tilde{\boldsymbol{v}}}])^2\right], \tag{21}$$

A.2 CONNECTION TO EXISTING LOSS FUNCTIONS ON DISCRETE-TIME DOMAIN

In this section, we connect our discretized loss of Equation (8) to the loss function, called relative trajectory balance (Venkatraman et al., 2024, RTB). Like our methods, RTB also amortized inference in target path distribution by training forward distribution on discrete-time domains such as vision, language, and control tasks. When discretized, our loss function is equivalent to the RTB objective.

Our goal is to show that for every paths $\boldsymbol{x}_{0:L}$ sampled from the path measure $\mathbb{P}_{\boldsymbol{v}_{\bar{\theta}}}$,

$$(\hat{F}_{\boldsymbol{v}_\theta,\boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{x}_{0:L})-w)^2=\left(\log\frac{p_0(\boldsymbol{x}_{0:L})1_{\mathcal{B}}(\boldsymbol{x}_{0:L})}{Z_\theta p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L})}\right)^2, \tag{22}$$

where $w=\log Z_\theta$ is a learnable scalar parameter, and path distribution $p_{\boldsymbol{v}}(\boldsymbol{x}_{0:L})=\prod_{\ell=0}^{L-1}p_{\boldsymbol{v}}(\boldsymbol{x}_{l+1}|\boldsymbol{x}_l)$ is Markovian, and its transition kernel $p_{\boldsymbol{v}}(\boldsymbol{x}_{l+1}|\boldsymbol{x}_l)$ are derived from Euler-Maruyama discretization of the SDE in Equation (3) as follows:

$$\boldsymbol{x}_{l+1}=\boldsymbol{x}_l+\boldsymbol{u}(\boldsymbol{x}_l)\Delta t+\Sigma\boldsymbol{v}(\boldsymbol{x}_l)\Delta t+\Sigma\boldsymbol{\epsilon}_l, \tag{23}$$

where $\boldsymbol{\epsilon}_l \sim \mathcal{N}(\mathbf{0}, \Delta t)$. To this end, we can calculate as follows:

$$\log p_0(\boldsymbol{x}_{0:L}) - \log p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L}) \tag{24}$$

$$= \sum_{\ell=0}^{L-1} \log p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{l+1}|\boldsymbol{x}_l) - \sum_{\ell=0}^{L-1} \log p_0(\boldsymbol{x}_{l+1}|\boldsymbol{x}_l) \tag{25}$$

$$= \frac{1}{2}\sum_{\ell=0}^{L-1} (\Sigma \boldsymbol{v}_{\bar{\theta}} \Delta t + \Sigma \boldsymbol{\epsilon}_l - \Sigma \boldsymbol{v}_\theta \Delta t)^T (\Sigma^T \Sigma \Delta t)^{-1} (\Sigma \boldsymbol{v}_{\bar{\theta}} \Delta t + \Sigma \boldsymbol{\epsilon}_l - \Sigma \boldsymbol{v}_\theta \Delta t) \tag{26}$$

$$- \frac{1}{2}\sum_{\ell=0}^{L-1} (\Sigma \boldsymbol{v}_{\bar{\theta}} \Delta t + \Sigma \boldsymbol{\epsilon}_l)^T (\Sigma^T \Sigma \Delta t)^{-1} (\Sigma \boldsymbol{v}_{\bar{\theta}} \Delta t + \Sigma \boldsymbol{\epsilon}_l) \tag{27}$$

$$= \frac{1}{2\Delta t}\sum_{\ell=0}^{L-1} (\|\boldsymbol{v}_{\bar{\theta}} \Delta t + \boldsymbol{\epsilon}_l - \boldsymbol{v}_\theta \Delta t\|^2 - \|\boldsymbol{v}_{\bar{\theta}} \Delta t + \boldsymbol{\epsilon}_l\|^2) \tag{28}$$

$$= \frac{1}{2}\sum_{\ell=0}^{L-1} \|\boldsymbol{v}_\theta(\boldsymbol{x}_\ell)\|^2 \Delta t - \sum_{\ell=0}^{L-1} (\boldsymbol{v}_\theta \cdot \boldsymbol{v}_{\bar{\theta}})(\boldsymbol{x}_\ell)\Delta t - \sum_{\ell=0}^{L-1} \boldsymbol{v}_\theta(\boldsymbol{x}_\ell) \cdot \boldsymbol{\epsilon}_\ell \tag{29}$$

$$= \hat{F}_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{x}_{0:L}) - \log 1_{\mathcal{B}}(\boldsymbol{x}_{0:L}) \tag{30}$$

which implies

$$\hat{F}_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{x}_{0:L}) = \log \frac{p_0(\boldsymbol{x}_{0:L}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L})}{p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L})}, \tag{31}$$

by subtracting w and squaring both sides, we have

$$(\hat{F}_{\boldsymbol{v}_\theta, \boldsymbol{v}_{\bar{\theta}}}(\boldsymbol{x}_{0:L}) - w)^2 = \left(\log \frac{p_0(\boldsymbol{x}_{0:L}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L})}{Z_\theta p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L})}\right)^2 \tag{32}$$

We can view $p_0(\boldsymbol{x}_{0:L}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L})$ as the unnormalized target distribution discretized from the target path measure $\mathbb{Q}$, and Z_θ as the estimator for normalizing constant $Z = \int p_0(\boldsymbol{x}_{0:L}) 1_{\mathcal{B}}(\boldsymbol{x}_{0:L}) d\boldsymbol{x}_{0:L}$, and $p_{\boldsymbol{v}_\theta}(\boldsymbol{x}_{0:L})$ as forward probability distribution to amortize inference in the target distribution. Based on these results, we provide our training algorithm in Algorithm 1.

A.3 Proof of scale-based parameterization

In this section, we prove that our scale-based parameterization of bias force strictly decreases the distance to the (aligned) target position for small step sizes, improving the ability to find informative paths in large molecules.

Proposition 1. *Consider the molecular state $\boldsymbol{R}_t$ at the t-th time step and the next state $\boldsymbol{R}'_{t+\Delta t} = \boldsymbol{R}_t + \boldsymbol{b}(\boldsymbol{X}_t)\Delta t/\boldsymbol{m}$ updated by step size Δt and the bias force $\boldsymbol{b}(\boldsymbol{X}_t) = diag(\boldsymbol{s}_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t))(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t)$. Then there always exists a small enough Δt that strictly decreases the distance towards the target state $\boldsymbol{R}_\mathcal{B}$:*

$$\|\rho'_{t+\Delta t} \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\| < \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|, \tag{33}$$

where $\rho'_{t+\Delta t} = \operatorname{argmin}_{\rho \in SE(3)} \|\rho \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\|$ and we assume that there does not exist a rotation that exactly aligns the current molecular state to the target state, i.e., $\|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\| > 0$.

Proof. The proof consists of two steps. We first show the (strictly) positive correlation between the bias force and the direction from the t-th state $\boldsymbol{R}_t$ to the target state $\boldsymbol{R}_\mathcal{B}$. Next, we show that the positive correlation gaurantees a strict decrease in distance between the states, i.e., $\|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|$, given that the distance was not already zero.

Step 1: First, we show that the bias force (divided by atom-wise masses) is positively correlated with the direction to the target position, i.e., $(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top (\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) > 0$. This follows from:

$$(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top (\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) = (\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t)^T \frac{\operatorname{diag}(\boldsymbol{s}_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t))}{\boldsymbol{m}} (\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) \tag{34}$$

$$= \sum_{i=1}^{3N} \left(\frac{\boldsymbol{s}_i}{\boldsymbol{m}_i}\right) (\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t)_i^2 > 0, \tag{35}$$

where $\boldsymbol{s}_i > 0$ is the i-th element of $\boldsymbol{s}_\theta(\rho_t^{-1} \cdot \boldsymbol{X}_t)$ and $(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t)_i$ is the i-th element of the direction to the target position.

Step 2: Next, we show that the positive correlation ensures distance reduction for a small enough step size. Consider the squared distance between the target position $\rho_t \cdot \boldsymbol{R}_\mathcal{B}$ and updated position $\boldsymbol{R}'$ by bias force

$$\begin{aligned}
&\|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\|^2 \\
&= \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - (\boldsymbol{R}_t + \boldsymbol{b}(\boldsymbol{X}_t)\Delta t/\boldsymbol{m})\|^2 && (36)\\
&= \|(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) - \boldsymbol{b}(\boldsymbol{X}_t)\Delta t/\boldsymbol{m}\|^2 && (37)\\
&= \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|^2 - 2\Delta t(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) + (\Delta t)^2\|\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m}\|^2. && (38)
\end{aligned}$$

Due to step 1, i.e., $(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t) > 0$, there exists a step size Δt satisfying:

$$0 < \Delta t < \frac{2(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t)}{\|\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m}\|^2}. \tag{39}$$

With this choice of Δt, multiplying $\Delta t\|\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m}\|^2$ leads to the following inequaliity:

$$(\Delta t)^2\|\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m}\|^2 < 2\Delta t(\boldsymbol{b}(\boldsymbol{X}_t)/\boldsymbol{m})^\top(\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t). \tag{40}$$

By subtracting the right-hand side from both sides and adding $\|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|^2$ to both sides, we have the following inequality:

$$\|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\|^2 < \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|^2. \tag{41}$$

Taking the square root of both sides, we have the following inequality:

$$\|\rho'_{t+\Delta t} \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\| \leq \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\| < \|\rho_t \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}_t\|, \tag{42}$$

where the first inequality follows from the definition of $\rho'_{t+\Delta t} = \operatorname{argmin}_{\rho \in SE(3)}\|\rho \cdot \boldsymbol{R}_\mathcal{B} - \boldsymbol{R}'_{t+\Delta t}\|$. This completes the proof. □

B EXPERIMENT DETAILS

B.1 OPENMM CONFIGURATIONS

For real-world molecules, we use the VVVR integrator (Sivak et al., 2014) with the step size $\Delta t = 1$ fs and the friction term $\gamma = 1$ ps^{-1}. In the TPS-DPS training algorithm, we start simulations at a temperature $\lambda_{\text{start}} = 600$K, and end at a temperature $\lambda_{\text{end}} = 300$K for Alanine Dipeptide, Polyproline Helix, and Chingolin and $\lambda_{\text{end}} = 400$K for Trpcage, BBA, and BBL. Other OpenMM configurations are shown in Table 2.

Table 2: **OpenMM configurations for real-world molecular systems.**

Molecule	Simulation time (T)	Force field	Solvent
Alanine Dipeptide	1000	`amber99sbildn` (Lindorff-Larsen et al., 2010)	`vaccum`
Polyproline Helix	10000	`amber/protein.ff14SBonlysc` (Maier et al., 2015)	`implicit/gbn2`
Chignolin	5000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`
Trpcage	5000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`
BBA	5000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`
BBL	5000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`

B.2 MODEL CONFIGURATIONS

We use a 3-layer MLP for the double-well system, and a 6-layer MLP for real-world molecules with ReLU activation functions for neural bias force, potential, and scale. To constrain the output of the neural bias scale parameterization to a positive value, we apply Softplus (Zheng et al., 2015) to the MLP output. As an input to the neural network, we concatenate the current position $(\boldsymbol{R}_t)_i$ of the i-th atom with its distance to the target position $d_i = \|(\tilde{\boldsymbol{R}}_{\mathcal{B}})_i - (\boldsymbol{R}_t)_i\|_2$. For real-world molecules, we apply the Kabsch algorithm (Kabsch, 1976) for heavy atoms to align $\boldsymbol{R}_{\mathcal{B}}$ with $\boldsymbol{R}_t$. We update the parameters of the neural network with a learning rate of 0.0001, while the scalar parameter w is updated with a learning rate of 0.001. We clip the gradient norm with 1 to prevent loss from exploding. we train $J = 1000$ times per rollout. We report other model configurations in Table 3. For PIPS, we use the model configurations reported by Holdijk et al. (2024). For CVs of SMD, we use backbone dihedral angles (ϕ, ψ) for Alanine Dipeptide and RMSD for Polyproline Helix and Chignolin.

Table 3: **Model configurations of TPS-DPS.**

System	# of rollouts (I)	# of samples (M)	Batch size (K)	Buffer size	Relaxation (σ)
Double-well	20	512	512	10000	3
Alanine Dipeptide	1000	16	16	1000	0.1
Polyproline Helix	100	16	4	200	0.2
Chignolin	100	16	4	200	0.5
Trpcage	100	16	4	100	0.5
BBA	100	16	4	100	0.5
BBL	100	16	2	100	0.5

B.3 EVALUATION METRICS

Root mean square distance (RMSD). We use the Kabsch algorithm (Kabsch, 1976) for heavy atoms to align the final position with the target position $\boldsymbol{R}_{\mathcal{B}}$, using the optimal (proper) rotation and translation to superimpose two heavy atom positions. We calculate RMSD between heavy atoms of the final position and the target position $\boldsymbol{R}_{\mathcal{B}}$.

Target hit percentage (THP). THP measures the success rate of paths arriving at the target metastable state $\mathcal{B}$ in a binary manner. Formally, given the final positions $\{R^{(i)}\}_{i=1}^{M}$ of M paths, THP is defined as follows:

$$\text{THP} = \frac{|\{i : \boldsymbol{R}^{(i)} \in \mathcal{B}\}|}{M} \tag{43}$$

Energy of transition state (ETS). ETS measures the ability of the method to find probable transition states when crossing the energy barrier. ETS refers to the maximum potential energy among states in a transition path. Formally, given a transition path $\boldsymbol{x}_{0:L}$ of length L that reaches the target meta-stable state i.e., $\boldsymbol{R}_L \in \mathcal{B}$, ETS is defined as follows:

$$\mathrm{ETS}(\boldsymbol{x}_{0:L}) = \max_{\ell \in [0,L]} U(\boldsymbol{R}_\ell) \tag{44}$$

B.4 System details

Double-well potential and dynamics. Double-well system follows the overdamped Langevin dynamics defined as follows:

$$\mathrm{d}\boldsymbol{R}_t = \frac{-\nabla U(\boldsymbol{R}_t)}{\boldsymbol{m}}\mathrm{d}t + \sqrt{\frac{2\gamma k_B \lambda}{\boldsymbol{m}}}\mathrm{d}\boldsymbol{W}_t\,. \tag{45}$$

For simplicity, we let $\boldsymbol{R} = (x, y) \in \mathbb{R}^2$, $\boldsymbol{m} = I, \gamma = 1, \Delta = 0.01, T = 10$, and $\lambda = 1200$K. To evaluate the ability to find diverse transition paths, we consider the following double-well potential (Hua et al., 2024):

$$U(x, y) = \frac{1}{6}\left(4(1 - x^2 - y^2)^2 + 2(x^2 - 2)^2 + [(x + y)^2 - 1]^2 + [(x - y)^2 - 1]^2 - 2\right)\,. \tag{46}$$

This potential has global minima and two saddle points, having two meta-stable states and two reaction channels.

Handedness. For four points A, B, C, D, we define a handedness (Moradi et al., 2009) as follows:

$$H_{ABCD} = \frac{\overrightarrow{EF} \cdot (\overrightarrow{CD} \times \overrightarrow{AB})}{|\overrightarrow{EF}| \cdot |\overrightarrow{CD}| \cdot |\overrightarrow{AB}|}. \tag{47}$$

Here, E and F are the midpoints of the vectors $\overrightarrow{AB}$ and $\overrightarrow{CD}$, respectively. With the backbone atoms $X_1, X_2, ..., X_N$ of a Polyproline Helix, we can define the handedness for N atoms as $H = \sum_{i=1}^{N-3} H_{X_i X_{i+1} X_{i+2} X_{i+3}}$. In our experiments, we take the alpha carbon X_1, the carbonyl carbon X_2 of the first proline residue, the alpha carbon X_3 of the second residue and alpha carbon X_4 of third residues to compute the handedness.

Time-lagged independent components (TICA). To extract the collective variable (CV) for fast folding proteins, we consider components of time-lagged independent component analysis (TICA; Pérez-Hernández et al., 2013). We run 1μs unbiased MD simulations with 2fs step size and record states per 2ps to collect MD trajectories, using the OpenMM library with the same configuration as in Appendix B.1. For the top two TICA components, we use PyEMMA library (Scherer et al., 2015) with a time lag $\tau = 500$ps for Chignolin and $\tau = 200$ for Trpcage, BBA, and BBL.

Reproducibility. We describe experiment details in Appendix B, including detailed simulation configuration and hyper-parameters. In the anonymous link, we provide the code for TPS-DPS.

C EXPERIMENTS ON FAST FOLDING PROTEINS

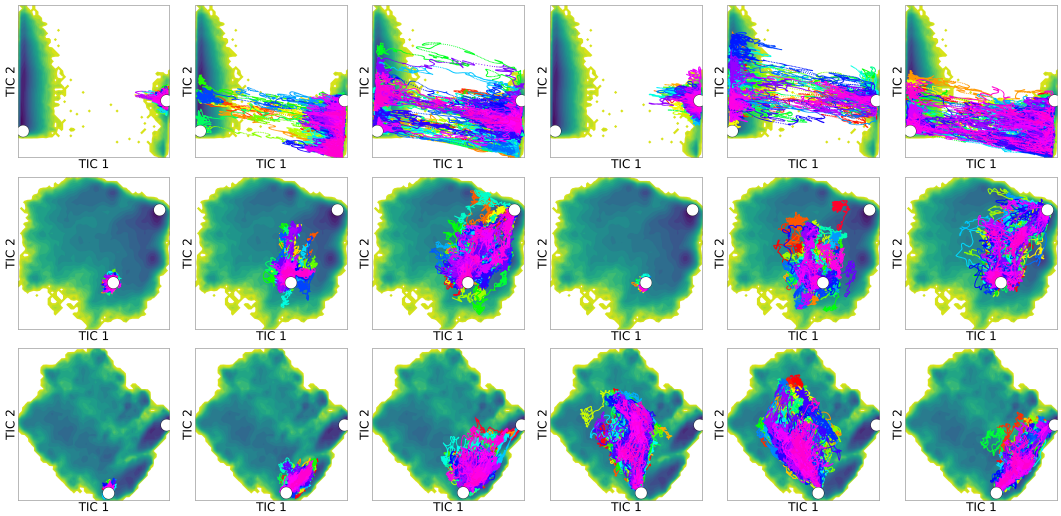
In this section, we evaluate our method, called TPS-DPS, on three fast-folding proteins (Lindorff-Larsen et al., 2011): Trpcage, BBA, and BBL at 400K.

Trpcage, BBA, and BBL are more challenging proteins than Chignolin since they have 20, 28, and 47 amino acids, respectively. As in previous experiments, we adopt three metrics: RMSD, THP, and ETS. We compare our methods with unbiased MD (UMD) steered MD (SMD). We define the target meta-stable state $\mathcal{B} = \{\boldsymbol{R} \mid \|\xi(\boldsymbol{R}) - \xi(\boldsymbol{R}_{\mathcal{B}})\| < 0.75\}$ where ξ consists of the top two components of time-lagged independent component analysis.

As shown in Table 4 and Figure 7, UMD and TPS-DPS (F) fail to sample transition paths. TPS-DPS (P) only succeeds in sampling transition paths of Trpcage and outperforms baselines in finding plausible transition states. TPS-DPS (S) outperforms baselines in RMSD and THP and finds more plausible transition states than SMD.

Table 4: **Benchmark scores on fast folding proteins**, all metrics are averaged over 64 paths. ETS is computed for paths that hit the target meta-stable state, and the best results are highlighted in bold. TPS-DPS predicting the bias force, potential, and atom-wise scaling are denoted by (F), (P), and (S), respectively. UMD (λ) denotes unbiased MD with temperature λ and SMD (k) denotes steered MD with the force constant k. Unless otherwise specified, paths are generated by MD simulation at 400K.

Molecule system	Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
Trpcage	UMD	7.94 ± 0.65	0.00	-
	UMD (1200K)	8.27 ± 1.13	0.00	-
	SMD (10K)	1.68 ± 0.23	3.12	-312.54 ± 20.67
	SMD (20K)	1.20 ± 0.20	42.19	-226.40 ± 85.59
	TPS-DPS (F, Ours)	6.35 ± 0.31	0.00	-
	TPS-DPS (P, Ours)	3.15 ± 0.52	12.50	$\mathbf{-512.97 \pm 56.89}$
	TPS-DPS (S, Ours)	$\mathbf{0.76 \pm 0.12}$	**81.25**	-317.61 ± 140.89
BBA	UMD	10.03 ± 0.39	0.00	-
	UMD (1200K)	10.81 ± 1.05	0.00	-
	SMD (10K)	2.89 ± 0.32	0.00	-
	SMD (20K)	1.66 ± 0.30	26.56	-3104.95 ± 97.57
	TPS-DPS (F, Ours)	9.48 ± 0.18	0.00	-
	TPS-DPS (P, Ours)	3.89 ± 0.35	0.00	-
	TPS-DPS (S, Ours)	$\mathbf{1.21 \pm 0.09}$	**84.38**	$\mathbf{-3801.68 \pm 139.38}$
BBL	UMD	18.48 ± 0.63	0.00	-
	UMD (1200K)	18.90 ± 1.16	0.00	-
	SMD (10K)	3.67 ± 0.22	0.00	-
	SMD (20K)	2.97 ± 0.33	7.81	-1738.57 ± 386.81
	TPS-DPS (F, Ours)	10.15 ± 0.54	0.00	-
	TPS-DPS (P, Ours)	6.45 ± 0.26	0.00	-
	TPS-DPS (S, Ours)	$\mathbf{1.60 \pm 0.19}$	**43.75**	$\mathbf{-3616.32 \pm 213.66}$



(a) UMD (b) SMD (10k) (c) SMD (20k) (d) TPS-DPS (F) (e) TPS-DPS (P) (f) TPS-DPS (S)

Figure 7: **64 sampled paths from each method for Trpcage, BBA, and BBL (from top to bottom rows) projected to the top two TICA components.** White circles indicate meta-stable states. Transition paths are from the unfolded to the folded state for each protein. SMD (k) denotes steered MD with the force constant k. All paths are generated by MD simulation at 400K. (F), (P), and (S) refer to predicting the bias force, the bias potential, and atom-wise positive scaling, respectively.

D COMPUTATIONAL COST

In this section, we analyze the time complexity of TPS-DPS and provide the number of energy evaluations and runtime in training and inference time for real molecules.

The training and inference time complexity of TPS-DPS is $O(NMLJ)$ and $O(NML)$, respectively, where N is the number of atoms, M is the number of samples, L is the number of MD steps, and J is the number of rollouts. To be specific, training consists of biased MD simulations with $O(NML)$ time complexity. Given the number of samples M, the total complexity of one biased MD step of TPS-DPS is $O(N)$.

To justify it, we note that the biased MD step consists of three stages: (1) calculating bias force, (2) calculating OpenMM force field, and (3) integrating the biased MD. Given the number of layers, and hidden units, MLP for bias force requires $O(N)$ and the Kabsch algorithm for equivariance requires $O(N)$. Calculating force field with cut-off and integrating MD with VVVR integrator (Sivak et al., 2014) also requires $O(N)$.

To measure computational cost, we consider the number of energy evaluations and runtime per rollout in training and inference time. As shown in Table 5, the inference cost of TPS-DPS is proportional to UMD and SMD which have time complexity $O(NML)$. TPS-DPS requires less energy evaluations than PIPS in training since TPS-DPS finds transition paths faster than PIPS by utilizing the replay buffer and simulated annealing.

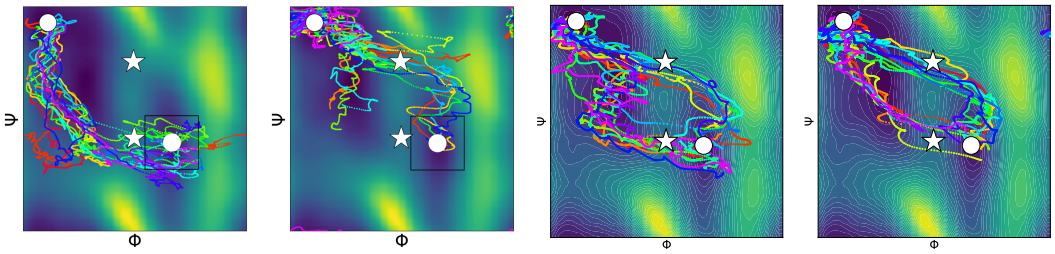
Table 5: **Cost comparison across molecular systems.** EET and EEI refer to the total number of energy evaluations in training and inference, respectively. RT and RI denote runtime (second) per rollout in training and inference on a single RTX A5000 GPU. MD simulations are conducted with $T = 1$ps for Alanine Dipeptide, $T = 5$ps for other systems, and $\Delta t = 1$fs.

Molecule	Method	EET (↓)	EEI (↓)	RT (↓)	RI (↓)
Alanine Dipeptide	UMD	-	64K	-	29.49
	SMD	-	64K	-	47.45
	PIPS (F)	240M	64K	44.22	71.05
	PIPS (P)	240M	64K	50.54	75.67
	TPS-DPS (F, Ours)	16M	64K	24.93	70.50
	TPS-DPS (P, Ours)	16M	64K	27.25	78.83
	TPS-DPS (S, Ours)	16M	64K	25.11	73.04
Chignolin	UMD	-	320K	-	224.23
	SMD	-	320K	-	283.45
	PIPS (F)	40M	320K	553.82	565.58
	PIPS (P)	40M	320K	632.89	622.87
	TPS-DPS (F, Ours)	8M	320K	209.29	562.90
	TPS-DPS (P, Ours)	8M	320K	224.36	623.63
	TPS-DPS (S, Ours)	8M	320K	215.18	581.26
Trp-Cage	UMD	-	320K	-	258.29
	SMD	-	320K	-	323.52
	TPS-DPS (F, Ours)	8M	320K	289.10	655.22
	TPS-DPS (P, Ours)	8M	320K	301.76	699.44
	TPS-DPS (S, Ours)	8M	320K	293.51	673.00
BBA	UMD	-	320K	-	395.12
	SMD	-	320K	-	542.35
	TPS-DPS (F, Ours)	8M	320K	422.23	1042.81
	TPS-DPS (P, Ours)	8M	320K	430.24	1091.97
	TPS-DPS (S, Ours)	8M	320K	426.48	1068.68
BBL	UMD	-	320K	-	673.55
	SMD	-	320K	-	853.77
	TPS-DPS (F, Ours)	8M	320K	560.95	1520.05
	TPS-DPS (P, Ours)	8M	320K	572.77	1607.62
	TPS-DPS (S, Ours)	8M	320K	563.45	1553.89

E COMPARISION WITH REVERSE KL DIVERGENCE

Table 6: **Benchmark scores of reverse KL divergence and TPS-DPS on Alanine Dipeptide system**. Metrics are averaged over 64 paths, and ETS is computed for paths that hit the target meta-stable state. The best results are highlighted in bold. TPS-DPS consistently outperforms reverse KL divergence on all metrics regardless of predicting bias force or potential.

Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
Reverse KL (F)	0.43 ± 0.34	53.12	27.88 ± 14.38
Reverse KL (P)	0.58 ± 0.34	48.43	21.61 ± 11.76
TPS-DPS (F, Ours)	$\mathbf{0.16 \pm 0.06}$	**92.19**	19.82 ± 15.88
TPS-DPS (P, Ours)	$\mathbf{0.16 \pm 0.10}$	87.50	$\mathbf{18.37 \pm 10.86}$



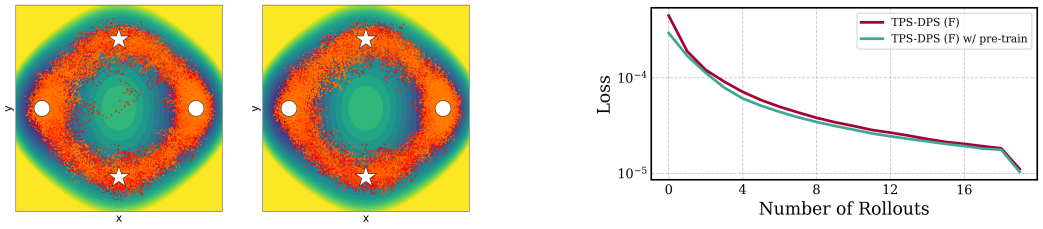
(a) Reverse KL (F) (b) Reverse KL (P) (c) TPS-DPS (F) (d) TPS-DPS (P)

Figure 8: **16 sampled paths from each method on the Ramachandran plot of Alanine Dipeptide.** White circles indicate meta-stable states and white stars indicate saddle points. We sample transition paths from the meta-stable state $C5$ (upper left) to $C7ax$ (lower right). Paths are generated by MD simulation at 300K. The reverse KL divergence struggles to find diverse reaction channels, suffering from mode collapse issues while the log-variance divergence of our method can capture two reaction channels and reach the target states better.

F PRE-TRAINING WITH GROUND TRUTH TRANSITION PATHS

In this section, we investigate the effectiveness of the pre-training model with a small amount of ground truth transition paths in the double-well system.

We pre-train the model with 1024 ground truth transition paths during 1000 epochs, using the same loss function described in Equation (10). We compare with the pre-trained model. As shown in Figure 9 and Table 7, TPS-DPS (F) with pre-training starts with lower loss but does not improve the final performance quantitatively and qualitatively since our approach nearly achieves maximum performance.



(a) Not pretrained (b) Pretrained (c) **Loss curves over rollouts.**

Figure 9: Visualizations of sampled transition paths (left) and loss curves over rollouts (right).

Table 7: **Performance of pre-training with ground truth transition paths**.

Method	RMSD (↓) Å	THP (↑) %	ETS (↓) kJmol^{-1}
TPS-DPS (F)	0.01 ± 0.02	99.90	1.38 ± 0.16
TPS-DPS (F) w/ pre-training	0.01 ± 0.02	100.00	1.37 ± 0.15