Collective Variable Free Transition Path Sampling with Generative Flow Network

Kiyoung Seong [1] **Seonghyun Park** [1] **Seonghwan Kim** [2] **Woo Youn Kim** [2] **Sungsoo Ahn** [1]

Abstract

Understanding transition paths between meta-stable states in molecular systems is fundamental for material design and drug discovery. However, sampling these paths via unbiased molecular dynamics simulations is computationally prohibitive due to the high energy barriers between the meta-stable states. Recent machine learning approaches are often restricted to simple systems or rely on collective variables (CVs) extracted from expensive domain knowledge. In this work, we propose to leverage generative flow networks (GFlowNets) to sample transition paths without relying on CVs. We reformulate the problem as amortized energy-based sampling over transition paths and train a neural bias potential by minimizing the squared log-ratio between the target distribution and the generator, derived from the flow matching objective of GFlowNets. Our evaluation on three proteins (Alanine Dipeptide, Polyproline Helix, and Chignolin) demonstrates that our approach, called TPS-GFN, generates more realistic and diverse transition paths than the previous CV-free machine learning approach.

1. Introduction

In material design and drug discovery, understanding the mechanisms and kinetics of transitions between meta-stable states of molecular systems, such as protein folding and chemical reactions (Spotte-Smith et al., 2022; Ahn et al., 2019; Mulholland, 2005; Piana et al., 2012), is crucial. A comprehensive study of them requires sampling diverse transition paths (Lee et al., 2017; Elber, 2016), which provide insights into mechanisms and energy landscapes. However, sampling transition paths through unbiased molecular dynamics (MD) simulations is challenging due to the high energy barriers of intermediate states, which cause an exponential decay in transition probability (Pechukas, 1981). Thus, a brute-force search through unbiased MD simulations is impractical within a realistic computational budget.

To address this problem, researchers have developed bias potential enhanced sampling (BPES) methods such as 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 (Invernizzi & Parrinello, 2020, OPES), and adaptive biasing force (Comer et al., 2015, ABF) methods. BPES methods rely on *bias potentials (or forces)* to facilitate transitions across high energy barriers. These methods design bias potentials using collective variables (CVs), functions of atomic coordinates that capture the slow modes of the transition. While effective for some systems, the reliance on expensive domain knowledge limits the applicability of BPES methods to systems where CVs are less understood.

Recently, machine learning has emerged as a promising paradigm for both designing and training neural bias potentials—neural network predicting the bias potential—without relying on CVs (Das et al., 2021; Lelièvre et al., 2023; Petersen et al., 2023; Holdijk et al., 2024). For instance, Das et al. (2021) and Lelièvre et al. (2023) used reinforcement learning to sample transition paths from biased MD with a parametrized force that approximates the transition path distribution. However, these works are limited to two-dimensional systems. Holdijk et al. (2024) trained the neural bias potential by minimizing KL divergence on the paths collected by itself, i.e., on-policy training. Still, this approach suffers from mode-seeking behavior, and the on-policy training does not reuse the collected paths, leading to inefficiency in sample complexity.

Contribution. In this work, we propose TPS-GFN, generative flow networks (Bengio et al., 2021, GFlowNets or GFNs) for transition path sampling (TPS), which trains the neural bias potential without requiring collective variables

[1]Pohang University of Science and Technology (POSTECH), Pohang, South Korea [2]Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Correspondence to: Kiyoung Seong <kyseong@postech.ac.kr>, Seonghyun Park <shpark26@postech.ac.kr>, Sungsoo Ahn <sungsoo.ahn@postech.ac.kr>, Seonghwan Kim <dmdtka00@kaist.ac.kr>, Woo Youn Kim <wooyoun@kaist.ac.kr>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

(CVs). To this end, we formulate the problem as the energy-based sampling over transition paths between the given meta-stable states. GFlowNet allows off-policy training and avoids collapsing to a particular mode of distribution over transition paths.

In particular, our method trains the neural bias potential by minimizing the squared log-ratio between the target distribution and the generator, which we derive from the one-step trajectory balance (TB) objective (Bengio et al., 2023; Malkin et al., 2022a). We use a replay buffer to store the paths, which are later used for off-policy training of the generator. Furthermore, we also design a new reward function to provide dense training signals and evaluate paths with various lengths between the meta-stable states.

We extensively evaluate our method for three peptides, i.e., Alenine Dipeptide, Polyproline Helix, and Chignolin. We demonstrate that our method faithfully generates a realistic transition path between the given meta-stable states in Figure 1, outperforming the prior ML-based approach (Holdijk et al., 2024). We also show that our method generates diverse transition paths and even discovers transition states.

2. Related work

2.1. Collective variable-based transition path sampling

Bias potential-based methods. Bias potential (or force) was originally introduced in enhanced sampling to explore molecular conformations that are difficult to access by the unbiased molecular dynamics (MD) within limited simulation times (Hénin et al., 2022). We focus on adaptive bias simulations such as meta-dynamics (Branduardi et al., 2012), on-the-fly probability-enhanced sampling (Invernizzi & Parrinello, 2020, OPES), and adaptive biasing force (Comer et al., 2015, ABF). These methods bias potential (or force) in a time-dependent manner and enhance the sampling along a few well-designed collective variables (CVs). However, selecting collective variables of a molecule requires domain expertise, restricting the applications to systems where CVs are little known.

MCMC-based methods. Shooting is a well-established method using the Markov chain Monte Carlo (MCMC) procedure on path space (Dellago et al., 1998). This method perturbs the velocity of a shooting point—one of the points on a transition path—to initiate MD simulations. It then performs MD simulations from this perturbed point and proposes new transition paths. Borrero & Dellago (2016) moved the shooting point with meta-dynamics to avoid traps in path space. Jung et al. (2023) used a neural network to model commitor function, selecting a shooting point with a high acceptance rate. Plainer et al. (2023) used latent proposals to leverage information from the latent space of the Boltzmann generator (Noé et al., 2019). However, the MCMC-based method suffers from a long mixing time to sample independent transition paths, and there is a trade-off between the acceptance rate and diversity. To avoid long mixing time, Falkner et al. (2022) directly generated independent shooting points from the Boltzmann generator with the umbrella sampling, but this method requires predefined CVs and shooting regions.

Automated learning of CV. To avoid the requirement of expensive domain knowledge about CVs, researchers have explored automating the derivation of CV from data (Bonati et al., 2020; 2021; Trizio & Parrinello, 2021; Ray et al., 2023; Yang et al., 2024). Bonati et al. (2020; 2021) and Trizio & Parrinello (2021) used discriminant analysis and time-lagged independent component analysis, respectively, on the neural network outputs from the physical descriptors collected by MD simulations. Ray et al. (2023) and Yang et al. (2024) considered the transition state ensemble as data augmentation strategies for discriminant analysis.

2.2. Collective variable-free transition path sampling

Generation from a fixed dataset. Given a dataset of transition paths, one can train generative models using the dataset to sample a transition path. Petersen et al. (2023) and Lelièvre et al. (2023) applied diffusion probabilistic models (Ho et al., 2020) and variational auto-encoders (Kingma & Welling, 2013), respectively. However, these methods are limited to small systems, as collecting the transition paths is notoriously difficult due to high energy barriers.

Neural bias potential-based method. Without a previously collected dataset, Das et al. (2021) and Hua et al. (2024) considered transition path sampling as a reinforcement learning (RL) problem and trained the neural bias potentials (or forces) of the policy by minimizing KL divergence. However, these works are limited to two-dimensional systems. Notably, Holdijk et al. (2024) scaled up to larger systems such as Chignolin and applied the path integral cross-entropy (Kappen & Ruiz, 2016) used in stochastic optimal control framework to train the neural bias potential.

2.3. Generative flow networks

Generative flow networks (GFlowNets) are the learning framework for amortized inference, generating objects through a series of decisions with a stochastic policy, or forward policy (Bengio et al., 2021; 2023). By learning the policy to sample an object proportionally to their rewards, GFlowNets can produce diverse samples (Bengio et al., 2021). Specifically, they train the forward policy by matching its trajectory flow with an auxiliary backward trajectory flow. Training objectives such as flow matching (Bengio et al., 2021), trajectory balance (Malkin et al., 2022a), and detailed balance (Bengio et al., 2023) have been proposed. Recent studies have successfully applied

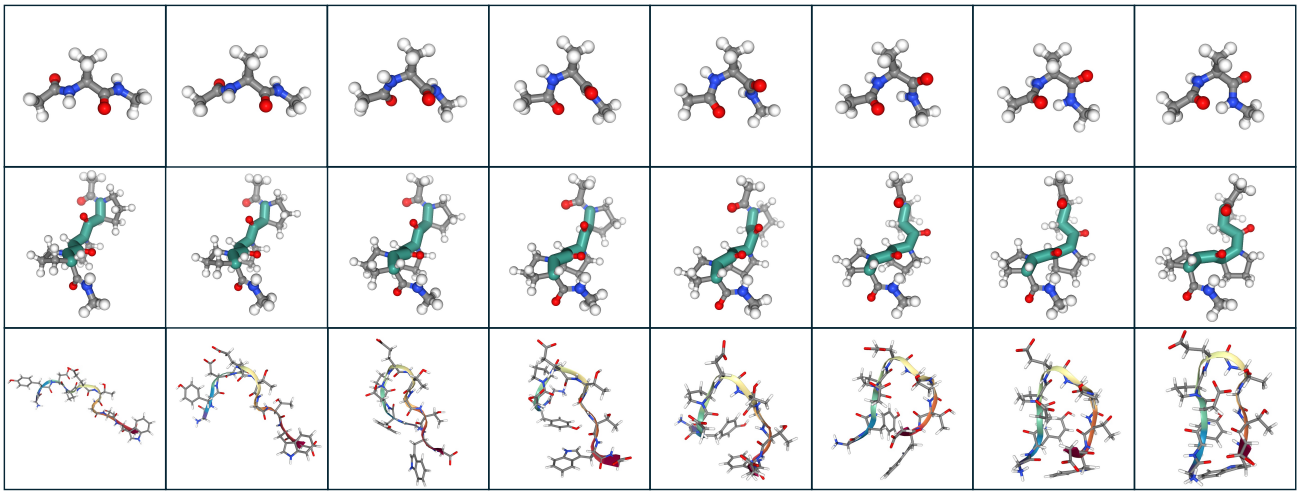


Figure 1. Transition paths generated by TPS-GFN, progressing from left to right over time. **(Top)** A conformation change of Alanine Dipeptide. **(Middle)** An isomerization of Polyproline Helix from left-handed to right-handed helix. **(Bottom)** A Chignolin folding process.

GFlowNets to tasks including molecular conformer generation (Volokhova et al., 2023), drug discovery (Shen et al., 2023), crystal structure design (Hernandez-Garcia et al., 2023), and biological sequence design (Jain et al., 2022).

Comparison with RL and variational inference. RL and variational inference are similar to GFlowNets in learning a target distribution specified by a reward function. GFlowNets and RL make sequential decisions to achieve high rewards, but GFLowNets differ by sampling proportionally to the reward rather than maximizing it. Moreover, while variational inference aim to learn the target (Boltzmann) distribution, GFlowNets reduce the variance of gradient estimator by estimating marginal quantities and allow off-policy learning without the need for importance sampling (Malkin et al., 2022b).

3. Method

3.1. Problem setup

At a high level, our goal is to sample transition paths from one meta-stable state to another meta-stable state through unbiased molecular dynamics (MD), e.g., transition paths from the meta-stable region $C5$ to $C7_{ax}$ of Alanine Dipeptide, as illustrated in Figure 2. Since such events rarely occur due to the high energy barrier between the two meta-stable regions, we train a neural network to search for such transition paths.

Molecular dynamics. We consider MD simulations that describe motion of a molecular state $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{v})$ with N atoms, given the atom-wise positions (or conformation) $\boldsymbol{r} \in \mathbb{R}^{N\times 3}$ and the atom-wise velocities $\boldsymbol{v} \in \mathbb{R}^{N\times 3}$. In particular, we consider Langevin dynamics (Bussi & Parrinello, 2007) to describe the molecular motion:

$$\mathrm{d}\boldsymbol{r} = \boldsymbol{v}\mathrm{d}t,$$
$$\mathrm{d}\boldsymbol{v} = \frac{-\nabla_{\boldsymbol{r}} U(\boldsymbol{r})}{\boldsymbol{m}}\mathrm{d}t - \gamma \boldsymbol{v}\mathrm{d}t + \sqrt{\frac{2\gamma k_B \lambda}{\boldsymbol{m}}}\mathrm{d}\boldsymbol{w},$$

where $U(\boldsymbol{r})$, $\boldsymbol{m}$, γ, k_B, λ, and $\mathrm{d}\boldsymbol{w}$ denote the potential energy, the atom-wise masses, the friction term, the Boltzmann constant, the temperature, and the Brownian motion, respectively.

Euler Maruyama discretization. Langevin dynamics can be discretized into a sequence of molecular states $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ using the Euler Maruyama method (Kloeden et al., 2012) as follows:

$$\boldsymbol{r}_t = \boldsymbol{r}_{t-1} + \boldsymbol{v}_{t-1}\Delta,$$
$$\boldsymbol{v}_t = (1-\gamma\Delta)\boldsymbol{v}_{t-1} + \frac{-\nabla_{\boldsymbol{r}} U(\boldsymbol{r}_t)}{\boldsymbol{m}}\Delta + \sqrt{\frac{2\gamma k_B \lambda}{\boldsymbol{m}}}\boldsymbol{\varepsilon}_t\Delta,$$

where Δ is the discretization step size and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{N\times 3}$ is a noise from the standard normal distribution. We denote the conditional probabilities described above as $p_{\mathrm{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

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 region $\mathcal{A} \subseteq \mathbb{R}^{N\times 3}$ before transitioning into a distinct meta-stable region $\mathcal{B} \subseteq \mathbb{R}^{N\times 3}$. Our goal is to sample a transition path $\boldsymbol{x}_{0:T} = (\boldsymbol{x}_0, \ldots \boldsymbol{x}_T)$ that links the distinct regions $\mathcal{A}$ and $\mathcal{B}$ where $\boldsymbol{r}_0 \in \mathcal{A}$, $\boldsymbol{r}_T \in \mathcal{B}$ and T is the length of the path.

Since the meta-stable region $\mathcal{A}$ and $\mathcal{B}$ are not well-specified for many molecular systems, we find unique local minima

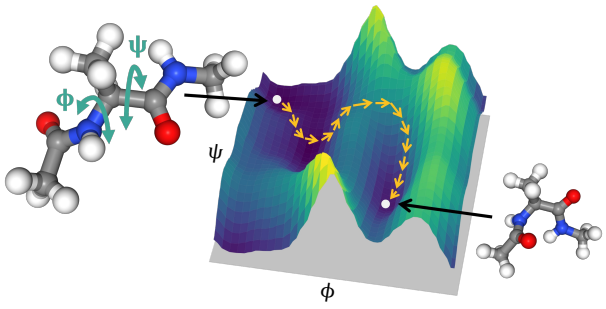


Figure 2. A transition path from the meta-stable state $C5$ to $C7_{ax}$ on the potential energy landscape of Alanine Dipeptide, represented by its well-known collective variables (CV), which are the two backbone dihedral angles (ϕ, ψ). The transition path is highlighted with yellow arrows.

$\boldsymbol{r}_{\mathcal{A}}$ and $\boldsymbol{r}_{\mathcal{B}}$ of the potential function inside the meta-stable regions $\mathcal{A}$ and $\mathcal{B}$, respectively. Afterwards, we sample a transition path $\boldsymbol{x}_{0:T}$ that starts from the state $\boldsymbol{x}_0 = \boldsymbol{x}_{\mathcal{A}}$ and ends nearby the conformation $\boldsymbol{r}_{\mathcal{B}}$, i.e., $\boldsymbol{r}_T \in \mathcal{B}$. To be specific, we fix $\boldsymbol{x}_0 = \boldsymbol{x}_{\mathcal{A}}$ and sample the path $\boldsymbol{x}_{1:T}$ from the following distribution:

$$p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \frac{1}{Z}\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)\prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad (1)$$

where Z is the normalizing constant and $\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)$ is the binary indicator function for the conformation $\boldsymbol{r}_T$ being in the meta-stable region $\mathcal{B}$. In other words, $\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T) = 1$ if $\boldsymbol{r}_T \in \mathcal{B}$, and $\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T) = 0$ otherwise.

To sample transition paths from the target distribution $p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, one can use rejection sampling. This involves sampling $\boldsymbol{x}_{1:T}$ through unbiased MD simulations and then checking if $\boldsymbol{r}_T \in \mathcal{B}$. However, this method is computationally expensive due to the low acceptance rate, which is caused by the high-energy barriers.

3.2. TPS-GFN: GFlowNets for transition path sampling

In this section, we explain our approach, called TPS-GFN, that amortizes the expensive cost of sampling transition paths from the distribution $p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$. Our key idea is threefold: (1) using the trajectory balance (TB) objective to train a generator $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ towards the true distribution $p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, (2) relaxation of the target distribution, and (3) parameterization of the generator with the neural bias potential. We provide a full description of the algorithm in Algorithm 1.

Training with trajectory balance (TB). We aim to make the generator $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ imitate the target distribution $p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$. From Equation (1), we can achieve this by

Algorithm 1 GFlowNet training for transition path sampling

Initialize an empty replay buffer $\mathcal{B}$, a neural biased potential b_θ, a scalar parameter Z_θ, and a temperature schedule $\lambda_{\text{start}} = \lambda_1 > \cdots > \lambda_I = \lambda_{\text{end}}$.
for $i = 1, \ldots, I$ **do**
  Generate M_1 paths $\{\boldsymbol{x}_{0:T}^{(m)}\}_{m=1}^{M_1}$ from the biased MD simulations at the temperature λ_i.
  Update the replay buffer $\mathcal{B} \leftarrow \mathcal{B} \cup \{\boldsymbol{x}_{0:T}^{(m)}\}_{m=1}^{M_1}$.
  for $j = 1, \ldots, J$ **do**
    Sample M_2 paths $\{\boldsymbol{x}_{0:T}^{(m)}\}_{m=1}^{M_2}$ from the buffer $\mathcal{B}$.
    Update parameters of b_θ and Z_θ to minimize the loss $\mathcal{L}_{\text{TB}}$ (Equation (2)) using these samples.
  end for
end for

satisfying the following identity:

$$\left(\log \frac{Z\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)\prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right)^2 = 0,$$

for all paths $\boldsymbol{x}_{1:T}$. To this end, we propose to use the one-step TB objective (Malkin et al., 2022a), the squared log-ratio between the unnormalized generator $Z_\theta p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ and the unnormalized target distribution $\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)\prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, as follows:

$$\mathcal{L}_{\text{TB}}(\boldsymbol{x}_{1:T}) = \left(\log \frac{Z_\theta\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)\prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right)^2, \quad (2)$$

where $Z_\theta \in \mathbb{R}$ is a learnable scalar parameter to estimate the normalization constant Z. We match the generator with the target distribution by minimizing the loss function $\mathcal{L}_{\text{TB}}(\boldsymbol{x}_{1:T})$ over paths $\boldsymbol{x}_{1:T}$ sampled from an *arbitrary* training distribution, i.e., off-policy training. We provide an explicit derivation of the one-step TB objective (Equation (2)) in Appendix A.

To leverage the ability of the TB objective, we use a replay buffer that collects the samples from the tempered version of the generator and reuses them for training. This improves sample efficiency and training stability, both for GFlowNets (Bengio et al., 2021) and reinforcement learning (Mnih et al., 2013). In particular, we collect paths from biased MD simulations with a predefined temperature schedule, starting from a high temperature λ_{start}, decreasing the temperature with each rollout, and ending at a temperature λ_{end}. This annealing strategy rapidly identifies target metastable states and explores diverse transition paths.

As described in Algorithm 1, the training algorithm of TPS-GFN iterates through four steps: (1) generating paths from the tempered version of the generator, (2) storing generated paths in the replay buffer, (3) sampling paths from the replay

Table 1. Benchmark scores on Alanine Dipeptide which has well-known CVs. All metrics are averaged over 64 trajectories. ETP and EFP are reported only if the final state is in the target meta-stable state. The best values are highlighted in **bold** except unbiased MD (300K). Values from the paper Holdijk et al. (2024) and the model parameter of its official code are indicated by † and ‡, respectively. Methods using bias potential and force are denoted by (P) and (F), respectively.

Method	EPD (↓) $\text{nm} \times 10^{-3}$	THP (↑) %	ETP (↓) kJmol^{-1}	LL (↑) $\times 10^2$	EPCD (↓) nm	EFP (↓) kJmol^{-1}
MD (300K)	7.52 ± 0.96	0.00	–	5.72 ± 0.00	84.73 ± 4.97	–
MD (9000K)	7.96 ± 2.36	4.68	870.11 ± 164.52	-13.14 ± 2.83	115.06 ± 31.37	789.54 ± 134.34
PIPS (P)†	1.21 ± 0.31	63.50	-8.35 ± 8.04	–	–	–
PIPS (F)‡	2.08 ± 0.29	46.87	3.49 ± 12.19	5.56 ± 0.01	12.06 ± 3.41	-24.85 ± 13.56
TPS-GFN (P)	$\mathbf{1.15 \pm 0.07}$	**100.00**	$\mathbf{-9.94 \pm 9.89}$	$\mathbf{5.58 \pm 0.01}$	$\mathbf{3.93 \pm 0.95}$	$\mathbf{-50.63 \pm 8.02}$
TPS-GFN (F)	1.48 ± 0.21	81.25	11.66 ± 18.44	5.57 ± 0.02	8.77 ± 4.12	-30.14 ± 11.46

Table 2. Benchmark scores on Polyproline Helix and Chignolin. All metrics are averaged over 64 trajectories. The best values are highlighted in **bold** except unbiased MD. Reproduced results from Holdijk et al. (2024) are indicated by †. Methods using bias potential and force are denoted by (P) and (F), respectively.

Method	EPD (↓) $\text{nm} \times 10^{-3}$	LL (↑) $\times 10^2$	EPCD (↓) nm
Polyproline Helix			
MD (300K)	24.14 ± 3.10	13.89 ± 0.01	87.96 ± 3.93
PIPS (P)†	14.81 ± 5.32	13.77 ± 0.02	73.83 ± 17.94
PIPS (F)†	30.25 ± 3.29	13.75 ± 0.02	104.89 ± 4.60
TPS-GFN (P)	$\mathbf{6.95 \pm 0.67}$	13.51 ± 0.03	$\mathbf{58.48 \pm 4.25}$
TPS-GFN (F)	17.63 ± 1.32	$\mathbf{13.87 \pm 0.02}$	82.08 ± 4.02
Chignolin			
MD (300K)	367.84 ± 101.43	36.01 ± 0.04	258.49 ± 21.15
PIPS (P)†	59.73 ± 11.19	34.03 ± 0.44	184.97 ± 18.57
PIPS (F)†	90.60 ± 43.86	33.83 ± 0.33	214.42 ± 39.63
TPS-GFN (P)	$\mathbf{16.46 \pm 5.94}$	33.88 ± 0.21	$\mathbf{62.76 \pm 14.93}$
TPS-GFN (F)	55.21 ± 10.15	$\mathbf{34.54 \pm 0.14}$	148.34 ± 18.14

buffer, and (4) training the generator by minimizing the TB objective. After training, TPS-GFN directly generates transition paths with biased MD simulation by amortizing inference in the target distribution.

As shown by Malkin et al. (2022b), the TB objective automatically performs gradient variance reduction for variational inference by estimating the normalizing constant Z that acts as a baseline and allows off-policy learning without the need for importance sampling. This is also a slight modification to the VarGrad objective (Richter et al., 2020; Deleu et al., 2024) that was proposed to reduce the variance of the gradient estimator of the evidence lower bound.

We note the similarity of TPS-GFN with the prior work (Holdijk et al., 2024), which minimizes the Kullback-Leibler (KL) divergence between the generator and the target distribution. In comparison, the TB objective yields better training dynamics due to reduced variance in the gradients, and better sample efficiency due to the ability to reuse the collected sample with the replay buffer. Furthermore, TPS-GFN does not suffer from the mode-seeking behavior of KL divergence.

Target distribution relaxation. Learning to imitate the target distribution $p_{\mathcal{A},\mathcal{B}}$ with the TB objective is challenging since its probability is rarely non-zero due to the binary indicator function $\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)$ of the target distribution. To alleviate this issue, we (1) replace the binary indicator function with a piecewise smooth function having positive values for all input paths and (2) allow paths to reach the meta-stable region $\mathcal{B}$ before termination.

First, we relax the target distribution as follows:

$$\tilde{p}_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \frac{1}{\tilde{Z}} \tilde{\mathbf{1}}_{\mathcal{B}}(\boldsymbol{r}_{1:T}) \prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}),$$

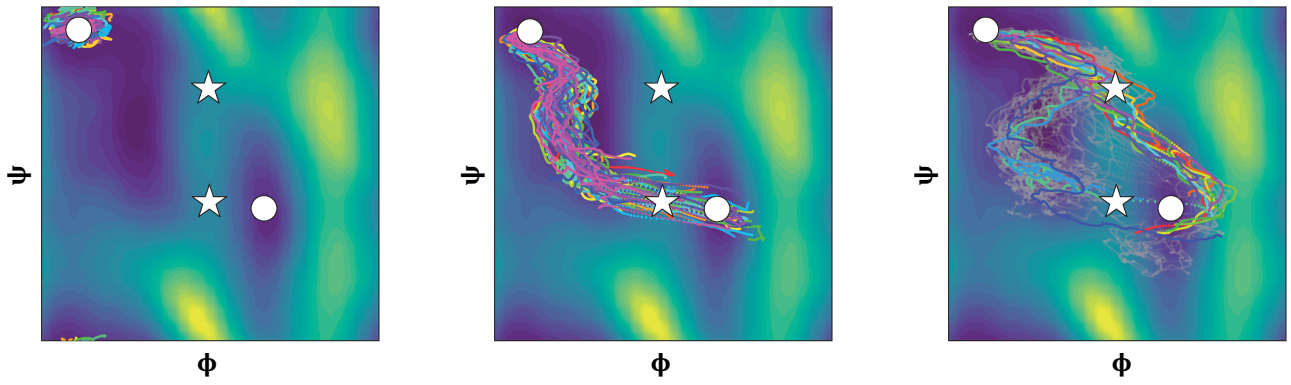
where $\tilde{Z}$ is the normalizing constant and $\tilde{\mathbf{1}}_{\mathcal{B}}(\boldsymbol{r}_{1:T})$ is the relaxed binary indicator function defined as follows:

$$\tilde{\mathbf{1}}_{\mathcal{B}}(\boldsymbol{r}_{1:T}) = \max_{t \in \{1,\dots,T\}} \exp\left(-\frac{\|D(\boldsymbol{r}_t) - D(\boldsymbol{r}_{\mathcal{B}})\|_F^2}{2\sigma^2}\right),$$

where $\|\cdot\|_F$ is the Frobenius norm, $\sigma > 0$ controls the degree of relaxation. $D(\boldsymbol{r}) \in \mathbb{R}^{N \times N}$ is a matrix whose (i,j)-th element is the Euclidean distance $\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2$ between i-th and the j-th atoms. Second, we allow the paths to have flexible lengths, by cutting off the tail of the generated paths at $t_* = \text{argmin}_{t \in \{1,\dots,T\}} \|D(\boldsymbol{r}_t) - D(\boldsymbol{r}_{\mathcal{B}})\|_F$. This enables finding transition paths of various lengths.

Parameterizing generator with neural bias potential. Similar to Holdijk et al. (2024), we model the generator $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ as the discretization of the following biased Langevin dynamics:

$$\mathrm{d}\boldsymbol{r} = \boldsymbol{v}\mathrm{d}t,$$
$$\mathrm{d}\boldsymbol{v} = \frac{-\nabla_{\boldsymbol{r}} U_b(\boldsymbol{r})}{\boldsymbol{m}}\mathrm{d}t - \gamma \boldsymbol{v}\mathrm{d}t + \sqrt{\frac{2\gamma k_B \lambda}{\boldsymbol{m}}}\mathrm{d}\boldsymbol{w},$$



(a) Unbiased MD (b) PIPS (c) **TPS-GFN (Ours)**

Figure 3. 64 sampled paths for each method on the Ramachandran plot of Alanine Dipeptide. White circles indicate meta-stable states, and stars indicate transition states. (a) The paths from unbiased MD simulations that fail to escape the initial meta-stable region. (b) The paths generated by PIPS pass through only one transition state. (c) The paths generated by TPS-GFN pass through both transition states. For clarity, 10 paths are highlighted.

where $U_b(r) = U(r) + b_\theta(\boldsymbol{r})$ and b_θ is the neural bias potential that guides the Langevin dynamics towards the meta-stable region $\mathcal{B}$. We discretize the biased Langevin dynamics as follows:

$$\boldsymbol{r}_{t+1} = \boldsymbol{r}_t + \boldsymbol{v}_t\Delta,$$
$$\boldsymbol{v}_{t+1} = (1-\gamma\Delta)\boldsymbol{v}_t + \frac{-\nabla_{\boldsymbol{r}}U_b(\boldsymbol{r}_t)}{\boldsymbol{m}}\Delta + \sqrt{\frac{2\gamma k_B\lambda}{\boldsymbol{m}}}\varepsilon_t\Delta,$$

We model the neural bias potential $b_\theta(\boldsymbol{r})$ or directly force $\mathbf{b}_\theta(\boldsymbol{r}) = -\nabla_{\boldsymbol{r}}b_\theta(\boldsymbol{r})$ for the generator. We use a multi-layered perception (MLP) architecture to parameterize the neural bias potential or force.

4. Experiment

In this section, we evaluate TPS-GFN on three benchmark systems, (1) Alanine Dipeptide, (2) Polyproline Helix, and (3) Chignolin based on realisticity and diversity of generated paths. We compare our method with unbiased MD and path integral stochastic optimal control for path sampling (Holdijk et al., 2024, PIPS) both quantitatively and qualitatively. We run molecular dynamics (MD) simulations, using OpenMM library (Eastman et al., 2023). We report OpenMM and model configurations in Appendix B.1 and Appendix B.2, respectively.

4.1. Molecular systems

Alanine Dipeptide. Alanine Dipeptide is a small peptide of two alanine residues connected by a peptide bond. Alanine Dipeptide has the well-known collective variable (CV), the two backbone dihedral angles (ϕ, ψ). In our experiments, we sample conformation changes from the meta-stable region $C5$ to $C7_{ax}$ as shown in Figure 2.

Polyproline Helix. Polyproline helix is a protein secondary structure formed by a sequence of proline amino acids (Adzhubei et al., 2013). Polyproline has two meta-stable states: Polyproline I (PP-I) and Polyproline II (PP-II). PP-I is a right-handed helix with all peptide bonds in the cis configuration, while PP-II is a left-handed helix with all peptide bonds in the trans configuration (Moradi et al., 2009). In our experiments, we use Polyproline consisting of 3 proline residues and sample isomerizations from PP-II to PP-I.

Chignolin. Chignolin is a synthetic protein consisting of 10 amino acids and folds into a stable β-hairpin structure (Honda et al., 2004; Seibert et al., 2005). A network of hydrogen bonds and hydrophobic interactions stabilizes the β-hairpin structure of chignolin. In our experiments, we sample folding processes from unfolded to folded states.

4.2. Quantitative results

We consider three metrics (1) Expected pairwise distance (EPD) (2) target hit percentage (THP) (3) energy transition point (ETP) introduced by Holdijk et al. (2024). EPD measures the similarity between the final conformation and the target conformation by comparing their interatomic distances. THP evaluates how many paths achieve the target meta-stable region in a binary manner. Unlike EPD and THP which only evaluate the final conformations of transition paths, ETP measures the maximum potential energy of a transition path, referring to the ability of a model to find the transition states when crossing the energy barrier.

However, ETP only evaluates the potential energy of the transition state, it can not reflect the realisticity of the overall transition path. To resolve this issue, we propose log-likelihood (LL) which refers to the log-likelihood of conditional density induced by the unbiased MD simulation.

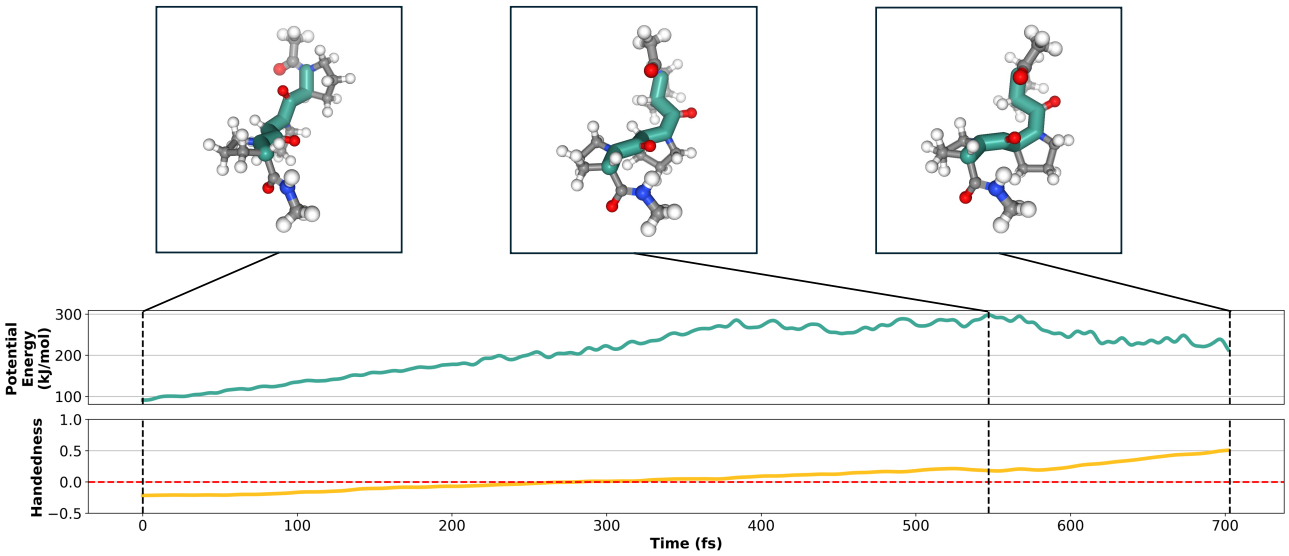


Figure 4. An isomerization from the meta-stable region PP-II to PP-I of Polyproline generated by TPS-GFN. **(Top)** 3d views of three states: initial, transition, and final state. The backbone of the Polyproline Helix is highlighted in green. **(Middle)** The potential energy of states over time. **(Bottom)** The handedness of states over time. The red line at $y = 0$ differentiates between PP-II and PP-I.

Similarly, since EPD and THP only consider the distance between the final conformation of sampled paths and the target conformation, they can not reflect the potential energy of the final conformation which explains its stability and realisticity. To evaluate it, we propose the potential energy of the final point (EFP), and the expected pairwise Coulomb distance (EPCD). EPCD measures both geometric and electrostatic similarity by comparing the Coulomb matrices of the final conformation and the target conformation. The Coulomb matrix captures atomic mass and energy level around the final state better than the distance matrix. More details of metrics are in Appendix B.3.

Alanine Dipeptide. Since Alanine Dipeptide has the well-known CV, i.e., two backbone dihedral angles (ϕ, ψ), we can evaluate THP, ETP, and EFP which requires defining the meta-stable region with the CV. In Table 1, TPS-GFN outperforms the baselines for both previous metrics (EPD, THP, ETP) and proposed metrics (LL, EPCD, EFP) regardless of predicting bias potential or force. The paths generated by our model reach the target meta-stable state better than unbiased MD simulations at 9000K while maintaining the realisticity at 300K. This shows that the TB objective used in our method leads to more realistic and high-quality transition paths than the KL divergence used in PIPS. Thus, our method faithfully amortizes the rejection sampling with unbiased MD simulations at 300K.

Polyproline Helix and Chignolin. Unlike Alanine Dipeptide, the CVs of Polyproline Helix, and Chignolin are less understood. Therefore, we exclude metrics THP, ETP, and EFP which require CVs to define the meta-stable state. In Table 2, our method outperforms baselines in both Polyproline Helix and Chignolin, regardless of predicting bias potential or force. This shows that our model samples more realistic transition paths. We note that predicting bias potential produces better transition paths than predicting bias force since the external force from the bias potential is conservative.

4.3. Qualitative results

In this section, we evaluate our model based on visualizations of paths. For Alanine Dipeptide, we compare transition paths from our method with baselines for diversity. For Polyproline Helix and Chignolin, we visualize the 3D conformations of a transition path with a potential energy plot. For more visualizations, please refer to project page.

Alanine Dipeptide. In Figure 3, we sample transition paths of three methods from $C5$ to $C7_{ax}$ on Ramachandran plot of Alanine Dipeptide. We sampled 64 transition paths for all cases. The paths from unbiased MD simulations in Figure 3a remain trapped in the initial meta-stable region, and fail to produce transition paths. Both PIPS (Holdijk et al., 2024) and TPS-GFN generate valid transition paths between two meta-stable regions as shown in Figure 3b and Figure 3c, respectively. PIPS generates transition paths that pass near one of the transition states while TPS-GFN generates transition paths that pass near both transition states as shown in Figure 3c. This shows that TPS-GFN captures diverse transition paths and discovers transition states.

Polyproline Helix. We plot potential energy over time and visualize 3D structures of three states: initial, transition, and

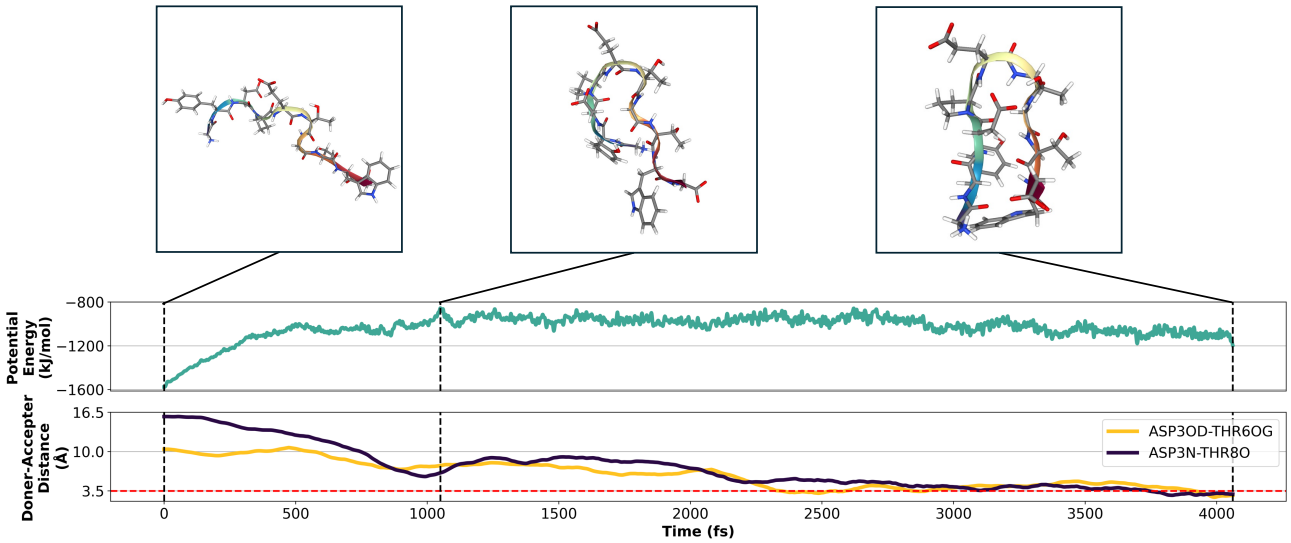


Figure 5. A folding process of Chignolin generated by TPS-GFN. **(Top)** 3d views of three states: initial, transition, and final state. **(Middle)** The potential energy over time. **(Bottom)** The doner-accepter distance of the two key hydrogen bonds, ASP3OD-THR6OG and ASP3N-THR8O over time. To form the hydrogen bonds, the donor-acceptor distance must be lower than the red line at $y = 3.5$Å.

final state of a transition path. We also consider handedness, H, which differentiates between PP-II and PP-I (Moradi et al., 2009). Specifically, $H < 0$ indicates a left-handed helix, while $H > 0$ indicates a right-handed helix. In Figure 4, our method captures the isomerization of Polyproline Helix from left-handed to right-handed Helix.

Chignolin. Similarly, we plot potential energy and visualize 3D structures. We consider the donor-acceptance distance of the two key hydrogen bonds, Asp3OD–Thr6OG and Asp3N–Thr8O. In Figure 5, our method captures the folding process, by forming the key hydrogen bonds at the final state. Notably, our method generates folding processes about 4 ps, which is shorter than the average folding time of 0.6 μs reported by Lindorff-Larsen et al. (2011).

5. Conclusion

In this work, we introduced a novel CV-free approach to sampling transition paths using generative flow networks (GFlowNets) to train neural bias potentials for efficient and diverse sampling. By reformulating the problem as amortized energy-based sampling and employing the one-step trajectory balance (TB) objective, our approach significantly improves the discovery of realistic transition paths compared to existing methods. Evaluations on proteins such as Alanine Dipeptide, Polyproline Helix, and Chignolin demonstrate superior accuracy and diversity in identifying transition states. We believe that our works expand to more complex systems, with significant implications for drug discovery and material design.

Limitation. While our algorithm successfully samples transition paths for small peptides, we did not evaluate our algorithm for larger peptides or proteins due to the lack of computational budget. In addition, our work does not generalize across different pairs of meta-stable states or different molecular systems. This points to an interesting venue for future research, which would be more appealing for practical applications in drug discovery or material design.

References

Adzhubei, A. A., Sternberg, M. J., and Makarov, A. A. Polyproline-ii helix in proteins: structure and function. *Journal of molecular biology*, 425(12):2100–2132, 2013.

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

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Bonati, L., Rizzi, V., and Parrinello, M. Data-driven collective variables for enhanced sampling. *The journal of physical chemistry letters*, 11(8):2998–3004, 2020.

Bonati, L., Piccini, G., and Parrinello, M. Deep learning the slow modes for rare events sampling. *Proceedings of the National Academy of Sciences*, 118(44):e2113533118, 2021.

Borrero, E. E. and Dellago, C. Avoiding traps in trajectory space: Metadynamics enhanced transition path sampling. *The European Physical Journal Special Topics*, 225:1609–1620, 2016.

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

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

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

Comer, J., Gumbart, J. C., Hénin, J., Lelièvre, T., Pohorille, A., and Chipot, C. 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.

Das, A., Rose, D. C., Garrahan, J. P., and Limmer, D. T. Reinforcement learning of rare diffusive dynamics. *The Journal of Chemical Physics*, 155(13), 2021.

Deleu, T., Nouri, P., Malkin, N., Precup, D., and Bengio, Y. Discrete probabilistic inference as control in multi-path environments, 2024.

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

Eastman, P., Galvelis, R., Peláez, R. P., Abreu, C. R., Farr, S. E., Gallicchio, E., Gorenko, A., Henry, M. M., Hu, F., Huang, J., et al. Openmm 8: Molecular dynamics simulation with machine learning potentials. *The Journal of Physical Chemistry B*, 128(1):109–116, 2023.

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

Ensing, B., De Vivo, M., Liu, Z., Moore, P., and Klein, M. L. Metadynamics as a tool for exploring free energy landscapes of chemical reactions. *Accounts of chemical research*, 39(2):73–81, 2006.

Falkner, S., Coretti, A., Romano, S., Geissler, P., and Dellago, C. Conditioning normalizing flows for rare event sampling. *arXiv preprint arXiv:2207.14530*, 2022.

Hénin, J., Lelièvre, T., Shirts, M. R., Valsson, O., and Delemotte, L. Enhanced sampling methods for molecular dynamics simulations [article v1. 0]. *Living Journal of Computational Molecular Science*, 4(1):1583–1583, 2022.

Hernandez-Garcia, A., Duval, A., Volokhova, A., Bengio, Y., Sharma, D., Carrier, P. L., Koziarski, M., and Schmidt, V. Crystal-GFN: sampling crystals with desirable properties and constraints. In *37th Conference on Neural Information Processing Systems (NeurIPS 2023)-AI4MAt workshop*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Holdijk, L., Du, Y., Hooft, F., Jaini, P., Ensing, B., and Welling, M. Stochastic optimal control for collective variable free sampling of molecular transition paths. *Advances in Neural Information Processing Systems*, 36, 2024.

Honda, S., Yamasaki, K., Sawada, Y., and Morii, H. 10 residue folded peptide designed by segment statistics. *Structure*, 12(8):1507–1518, 2004.

Hua, X., Ahmad, R., Blanchet, J., and Cai, W. Accelerated sampling of rare events using a neural network bias potential. *arXiv preprint arXiv:2401.06936*, 2024.

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

Jain, M., Bengio, E., Hernandez-Garcia, A., Rector-Brooks, J., Dossou, B. F., Ekbote, C. A., Fu, J., Zhang, T., Kilgour, M., Zhang, D., et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pp. 9786–9801. PMLR, 2022.

Jung, H., Covino, R., Arjun, A., Leitold, C., Dellago, C., Bolhuis, P. G., and Hummer, G. Machine-guided path sampling to discover mechanisms of molecular self-organization. *Nature Computational Science*, 3(4):334–345, 2023.

Kappen, H. J. and Ruiz, H. C. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162:1244–1266, 2016.

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

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. 2013.

Kloeden, P. E., Platen, E., and Schurz, H. *Numerical solution of SDE through computer experiments*. Springer Science & Business Media, 2012.

Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-Garcıa, A., Ezzine, L. N., Bengio, Y., and Malkin, N. A theory of continuous generative flow networks. In *International Conference on Machine Learning*, pp. 18269–18300. PMLR, 2023.

Lee, J., Lee, I.-H., Joung, I., Lee, J., and Brooks, B. R. Finding multiple reaction pathways via global optimization of action. *Nature Communications*, 8(1):15443, 2017.

Lelièvre, T., Robin, G., Sekkat, I., Stoltz, G., and Cardoso, G. V. Generative methods for sampling transition paths in molecular dynamics. *ESAIM: Proceedings and Surveys*, 73:238–256, 2023.

Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

Malkin, N., Jain, M., Bengio, E., Sun, C., and Bengio, Y. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022a.

Malkin, N., Lahlou, S., Deleu, T., Ji, X., Hu, E. J., Everett, K. E., Zhang, D., and Bengio, Y. Gflownets and variational inference. In *The Eleventh International Conference on Learning Representations*, 2022b.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning, 2013.

Moradi, M., Babin, V., Roland, C., Darden, T. A., and Sagui, C. Conformations and free energy landscapes of polyproline peptides. *Proceedings of the National Academy of Sciences*, 106(49):20746–20751, 2009.

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

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

Petersen, M., Roig, G., and Covino, R. Dynamicsdiffusion: Generating and rare event sampling of molecular dynamic trajectories using diffusion models. 2023.

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

Plainer, M., Stark, H., Bunne, C., and Günnemann, S. Transition path sampling with boltzmann generator-based mcmc moves. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Ray, D., Trizio, E., and Parrinello, M. Deep learning collective variables from transition path ensemble. *The Journal of Chemical Physics*, 158(20), 2023.

Richter, L., Boustati, A., Nüsken, N., Ruiz, F., and Akyildiz, O. D. Vargrad: a low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33:13481–13492, 2020.

Rupp, M., Tkatchenko, A., Müller, K.-R., and Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

Seibert, M. M., Patriksson, A., Hess, B., and Van Der Spoel, D. Reproducible polypeptide folding and structure prediction using molecular dynamics simulations. *Journal of molecular biology*, 354(1):173–183, 2005.

Shen, T., Pandey, M., and Ester, M. TacoGFN: Target conditioned gflownet for structure-based drug design. 2023.

Spotte-Smith, E. W. C., Kam, R. L., Barter, D., Xie, X., Hou, T., Dwaraknath, S., Blau, S. M., and Persson, K. A. Toward a mechanistic model of solid–electrolyte interphase formation and evolution in lithium-ion batteries. *ACS Energy Letters*, 7(4):1446–1453, 2022.

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

Trizio, E. and Parrinello, M. From enhanced sampling to reaction profiles. *The Journal of Physical Chemistry Letters*, 12(35):8621–8626, 2021.

Volokhova, A., Koziarski, M., Hernández-García, A., Liu, C.-H., Miret, S., Lemos, P., Thiede, L., Yan, Z., Aspuru-Guzik, A., and Bengio, Y. Towards equilibrium molecular conformation generation with gflownets. *arXiv preprint arXiv:2310.14782*, 2023.

Yang, S., Nam, J., Dietschreit, J. C., and Gómez-Bombarelli, R. Learning collective variables for protein folding with labeled data augmentation through geodesic interpolation. *arXiv preprint arXiv:2402.01542*, 2024.

A. Generative Flow Networks

In this section, we briefly introduce the concept of GFlowNets and explain how we derive Equation (2). We refer to Lahlou et al. (Lahlou et al., 2023) for a comprehensive study on formalizing GFlowNets on continuous state spaces.

Generative flow networks. GFlowNet (Bengio et al., 2021; 2023) is a framework to sequentially construct samples $s \in \mathcal{S}$ proportional to a reward function $R(s)$, where $\mathcal{S}$ is a set of data. In other words, GFlowNet aims to learn a target distribution $p^*(s)$ defined as:

$$p^*(s) = \frac{R(s)}{Z}, \qquad Z = \int_{\mathcal{S}} R(s)ds,$$

where Z is the normalizing constant. To this end, GFlowNet samples a trajectory $\tau = (s_0, ..., s_L) \in \{s_0\} \times \mathcal{S}^L$ of length L through a series of state transitions made by a forward policy $P_F(s_l|s_{l-1};\theta)$ for $l = 1, \dots, L$, where s_0 is the fixed initial state of all trajectories and $s = s_L$ is the terminal state we want to sample.

Flow matching. To train the forward policy, GFlowNet introduces an auxiliary backward policy $P_B(s_{l-1}|s_l;\theta)$ and matches the *forward flow* $Z_\theta P_F(\tau;\theta)$ with *backward flow* $R(s)P_B(\tau|s;\theta)$ using the *flow matching* objective over the trajectories $\tau = (s_0, ..., s_L)$:

$$Z_\theta \prod_{l=1}^{L} P_F(s_l|s_{l-1};\theta) \approx R(s) \prod_{l=1}^{L} P_B(s_{l-1}|s_l;\theta) \tag{3}$$

where $Z_\theta \in \mathbb{R}$ is a learnable scalar parameter to estimate the normalization constant Z. Equation (3) induces $Z_\theta P_F(s) \approx R(s)$, by marginalizing forward and backward flows over set of trajectories $\mathcal{T}(s)$ ending at the terminal state s, i.e., $\int_{\tau\in\mathcal{T}(s)} Z_\theta P_F(\tau;\theta)d\tau = Z_\theta P_F(s)$ and $\int_{\tau\in\mathcal{T}(s)} R(s)P_B(\tau|s;\theta)d\tau = R(s)$. In other words, we can sequentially construct a sample s proportional to a reward function $R(s)$ through a series of state transitions by the forward policy after minimizing the flow matching objective.

Trajectory balance. For flow matching, Malkin et al. (Malkin et al., 2022a) proposed the trajectory balance objective:

$$\mathcal{L}_{\text{TB}}(s_1, \dots, s_L;\theta) = \left(\log \frac{Z_\theta \prod_{l=1}^{L} P_F(s_l|s_{l-1};\theta)}{R(s_L)\prod_{l=1}^{L} P_B(s_{l-1}|s_l;\theta)}\right)^2,$$

which corresponds to minimizing the squared difference of logarithms on the left and right-hand side of Equation (3). In particular, one-step GFlowNets, i.e., $L = 1$, induces the following training objective:

$$\mathcal{L}_{\text{TB}}(s_1;\theta) = \left(\log \frac{Z_\theta P_F(s_1|s_0;\theta)}{R(s_1)}\right)^2, \tag{4}$$

where $P_B(s_0|s_1;\theta)$ is ignored since s_0 is the constant initial state. Note that the detailed balance (Bengio et al., 2023) objective also reduces to the same formula as in Equation (4) with one-step GFlowNets.

Transition path sampling via GFlowNets. In this work, we formulate the transition path sampling problem (Equation (1)) as one-step GFlowNets. We set $\mathcal{S}$ as a set of paths $\boldsymbol{x}_{1:T}$ of length T starting at the initial state $s_0 = \boldsymbol{x}_0$ and the reward function as

$$R(\boldsymbol{x}_{1:T}) = \mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T) \prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \propto p_{\mathcal{A},\mathcal{B}}(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0). \tag{5}$$

Since $P_F(s_1|s_0;\theta) = p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, one can derive Equation (2) from Equation (4):

$$\mathcal{L}(\boldsymbol{x}_{1:T};\theta) = \left(\log \frac{Z_\theta \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{\mathbf{1}_{\mathcal{B}}(\boldsymbol{r}_T)\prod_{t=1}^{T} p_{\text{MD}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right)^2.$$

B. Experiment details

We implement codes based on OpenMM (Eastman et al., 2023) and Pytorch (Paszke et al., 2019) libraries and run codes on a single GPU (NVIDIA RTX 3090 or RTX A5000).

B.1. OpenMM configurations

We use the Langevin integrator (Bussi & Parrinello, 2007) with the step size $\Delta = 1$ fs and the friction term $\gamma = 1$ ps^{-1}. In training TPS-GFN, we start simulations from a temperature $\lambda_{\text{start}} = 600$K, linearly decrease temperature, and end at a temperature $\lambda_{\text{end}} = 300$K. Other OpenMM configurations are shown in Table 3. For PIPS, the number of steps T of Alanine Dipeptide is 500 as given by Holdijk et al. (2024).

Table 3. OpenMM configurations.

Molecule	# steps (T)	Force field	Solvent
Alanine Dipeptide	1000	`amber99sbildn`	`tip3p`
Polyproline Helix	10000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`
Chignolin	10000	`amber/protein.ff14SBonlysc`	`implicit/gbn2`

B.2. Model configurations

For PIPS, we use the model configurations reported by Holdijk et al. (2024). For TPS-GFN, we roll out $I = 1000$ times, the learning rate to 0.0001 for the neural bias potential, and 0.01 for a scalar parameter $w = \log Z_\theta$. We concatenate the distance feature $d_i = \|\boldsymbol{r}_i - (\boldsymbol{r}_{\mathcal{B}})_i\|_2$ and the position $\boldsymbol{r}_i$ of the i-th atom, using this as input for the neural bias potential. We report other model configurations in Table 4. We use the same configurations for both the neural bias potential and force, except for the output dimension of neural networks.

Table 4. Model configurations of TPS-GFN.

Molecule	Trains per rollout (J)	# Samples (M_1)	Batch Size (M_2)	Buffer Size	Relaxation (σ)
Alanine Dipeptide	2000	16	16	2048	0.05
Polyproline Helix	100	2	2	256	0.1
Chignolin	100	2	2	256	0.2

B.3. Evaluation metrics

Following Holdijk et al. (2024), we adopt the following three metrics for evaluation.

Expected pairwise distance (EPD). The interatomic pairwise distance matrix $D(\cdot)$ expresses molecular conformation in an E(3)-invariant manner. EPD measures the similarity between the final conformation $\boldsymbol{r}_{t_*}$ and the target conformation $\boldsymbol{r}_{\mathcal{B}}$ by comparing interatomic pairwise distance matrices as follows:

$$\text{EPD}(\boldsymbol{r}_{t_*}, \boldsymbol{r}_{\mathcal{B}}) = \frac{1}{N^2}\|D(\boldsymbol{r}_{t_*}) - D(\boldsymbol{r}_{\mathcal{B}})\|_F^2$$

Where t_* is the index of the final conformation of transition paths.

Target hit percentage (THP). THP measures the success rate of paths arriving at the target meta-stable state $\mathcal{B}$ in a binary manner. For the final conformations $\{r^{(i)}\}_{i=1}^M$ of M paths, THP is defined as follows:

$$\text{THP} = \frac{|\{i : r^{(i)} \in \mathcal{B}\}|}{M}$$

For the alanine dipeptide, we use $\mathcal{B} = \{r : |\phi(r) - \phi(r_{\mathcal{B}})| + |\psi(r) - \psi(r_{\mathcal{B}})| < 0.75\}$ where (ϕ, ψ) is the collective variable consisting of two backbone dihedral angles.

Energy transition point (ETP). ETP measures the ability of the method to find the transition states when crossing the energy barrier. ETP refers to the maximum potential energy among states in a transition path. Formally, given a transition path $\boldsymbol{x}_{0:t_*}$ that reaches the target meta-stable region i.e., $\boldsymbol{r}_{t_*} \in \mathcal{B}$, ETP is defined as follows:

$$\mathrm{ETP}(\boldsymbol{x}_{0:t_*}) = \max_{i \in [0, t_*]} U(\boldsymbol{r}_i)$$

However, we point out that these metrics can not measure the realisticity of the overall paths with respect to unbiased molecular dynamics (MD) and final conformations with respect to their potential energy.

Log Likelihood (LL). Log-likelihood (LL) evaluates the realisticity of paths with respect to unbiased MD simulation at 300K, referring to the log-likelihood of conditional density of unbiased MD simulation as follows:

$$\mathrm{LL}(\boldsymbol{x}_{0:T}) = \frac{1}{T} \sum_{t=1}^{T} \log p_{\mathrm{MD}}(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$$

We divide by T to evaluate paths regardless of length.

For the realisticity of final conformations, we focus on the potential energy of them. We propose two metrics, potential Energy of the final point (EFP) and Expected pairwise coulomb distance (EPCD).

Potential energy of final point (EFP). The final conformations of a transition path from unbiased MD simulation have low potential energy due to meta-stability. EFP measures the potential energy $U(r_{t_*})$ of the final conformations of transition paths sampled from the model.

Expected pairwise Coulomb distance (EPCD). Coulomb matrix (Rupp et al., 2012) $C(r)$ of the conformation r describes electrostatic interaction between nuclei and is formulated as follows:

$$C(\boldsymbol{r})_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{for } i = j \\ Z_i Z_j / \|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2 & \text{for } i \neq j \end{cases}$$

where Z_i and Z_j are the atomic numbers and r_i, r_j are the positions of atoms i and j, respectively. The atomic numbers Z_i and Z_j of the Coulomb matrix reflect mass and energy level around the final conformation better than the distance matrix. We propose EPCD which replaces the distance metrics used in EPD with the Coulomb matrix and is defined as follows:

$$\mathrm{EPCD}(\boldsymbol{r}_t, \boldsymbol{r}_{\mathcal{B}}) = \frac{1}{N^2} \|C(\boldsymbol{r}_{t_*}) - C(\boldsymbol{r}_{\mathcal{B}})\|_F^2$$

EPCD measures geometric and electrostatic similarities between the final conformation $\boldsymbol{r}_{t_*}$ and the target conformation $\boldsymbol{r}_{\mathcal{B}}$.