The Journal of Chemical Physics

RESEARCH ARTICLE | AUGUST 26 2024

# BAD-NEUS: Rapidly converging trajectory stratification

John Strahan ; Chatipat Lorpaiboon ; Jonathan Weare ; Aaron R. Dinner

*J. Chem. Phys.* 161, 084109 (2024)
https://doi.org/10.1063/5.0215975

AIP Publishing

The Journal of Chemical Physics | ARTICLE | pubs.aip.org/aip/jcp

# BAD-NEUS: Rapidly converging trajectory stratification

Cite as: J. Chem. Phys. 161, 084109 (2024); doi: 10.1063/5.0215975
Submitted: 26 April 2024 • Accepted: 25 July 2024 •
Published Online: 26 August 2024

John Strahan,[1] Chatipat Lorpaiboon,[1] Jonathan Weare,[2] and Aaron R. Dinner[1,a)]

## AFFILIATIONS

[1] Department of Chemistry and James Franck Institute, University of Chicago, Chicago, Illinois 60637, USA
[2] Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA

[a)] Author to whom correspondence should be addressed: dinner@uchicago.edu

## ABSTRACT

An issue for molecular dynamics simulations is that events of interest often involve timescales that are much longer than the simulation time step, which is set by the fastest timescales of the model. Because of this timescale separation, direct simulation of many events is prohibitively computationally costly. This issue can be overcome by aggregating information from many relatively short simulations that sample segments of trajectories involving events of interest. This is the strategy of Markov state models (MSMs) and related approaches, but such methods suffer from approximation error because the variables defining the states generally do not capture the dynamics fully. By contrast, once converged, the weighted ensemble (WE) method aggregates information from trajectory segments so as to yield unbiased estimates of both thermodynamic and kinetic statistics. Unfortunately, errors decay no faster than unbiased simulation in WE as originally formulated and commonly deployed. Here, we introduce a theoretical framework for describing WE that shows that the introduction of an approximate stationary distribution on top of the stratification, as in nonequilibrium umbrella sampling (NEUS), accelerates convergence. Building on ideas from MSMs and related methods, we generalize the NEUS approach in such a way that the approximation error can be reduced systematically. We show that the improved algorithm can decrease the simulation time required to achieve the desired precision by orders of magnitude.

## I. INTRODUCTION

The sampling of a stochastic process can be controlled by evolving an ensemble and splitting and merging the trajectories of its members, and various algorithms based on this strategy have been introduced.[1–7] Because the trajectory segments between splitting events are unbiased, such algorithms can yield dynamical statistics, such as transition probabilities and mean first passage times (MFPTs) to selected states, in addition to steady-state probability distributions and averages. Furthermore, splitting algorithms generally require little communication between the members of the ensemble, making them relatively straightforward to implement regardless of the underlying dynamics, and community software is available.[8,9] As a result, splitting algorithms are widely used.

Recent mathematical analysis of one of the oldest splitting algorithms in the molecular simulation literature, weighted ensemble (WE),[1] shows that it is asymptotically unbiased and can dramatically reduce the variance of statistics when the splitting criteria, which are based on a partition of the state space, are chosen appropriately.[10] However, the method relies on convergence of the steady-state ensemble of trajectories, which is known to be slow. In fact, as we argue below and was observed previously,[10] it is as slow as running direct unbiased simulations. Prior to convergence, the method is systematically biased.

Some previous work to accelerate the convergence of WE focused on improving the initialization of walkers using an approximation of the stationary distribution. In one case, the approximation of the stationary distribution was obtained from a free-energy method that biased trajectories to escape metastable wells,[11] while in another it was obtained from a machine-learning method based on unbiased short trajectories.[12] The former approach requires the underlying dynamics to obey detailed balance to ensure the validity of the free energy method, while the latter approach relies on a Markov approximation (but does not make other restrictions on the dynamics). In both cases, however, the approximation to the stationary distribution is used only for the first iteration, and so if the approximation is deficient, significant bias can persist in the WE calculation for many iterations.

Others focused on improving the rate estimates from WE.[13,14] Of particular relevance to the present study, Copperman and Zuckerman exploited the fact that splitting algorithms sample unbiased trajectory segments to construct history-augmented Markov state models (haMSMs)[15] from their WE data.[13] In an haMSM, the system is divided into discrete states, and the probabilities for transitions between them are computed from data, just as in a standard MSM, except that the path ensemble is split based on the last metastable state visited; the rate constant is then computed as the flux from $A$ to $B$ conditioned on having last visited $A$, normalized by the steady-state probability of having last visited $A$. The idea is that finer discretization of the state space than the WE simulation yields improved estimates. In itself, this approach does not accelerate the convergence of the sampling. However, Refs. 8 and 13 suggest that iteratively restarting the WE simulation from the steady-state distribution estimated from the haMSM could accelerate convergence.

Here, we build on this idea to introduce a general framework for how to apply a short-trajectory based approximation to the stationary distribution between successive iterations. As we show below, the approximation must satisfy certain properties and be correctly integrated in the algorithm; if these criteria are met, the approach can accelerate the convergence by several orders of magnitude without introducing systematic bias. Our approach is able to compute the full stationary distribution with minimal assumptions about the dynamics, as well as kinetic statistics such as rates and committors (probabilities of visiting selected states before others).

Conceptually, one can view our approach as a form of stratification, in which the weights of not just individual trajectory segments but groups of them are manipulated. Stratification was first introduced for controlling the sampling of trajectory segments in nonequilibrium umbrella sampling (NEUS).[3,4,16–19] NEUS groups trajectories that are in the same regions of state space, as defined by collective variables (CVs), and estimates the steady-state probabilities of the regions by solving a global flux balance equation; this strategy was subsequently adopted in extensions of WE[20] and exact milestoning (EM).[6] A unified framework for trajectory stratification that incorporates elements of all of the above-mentioned algorithms and is unbiased in the limit that each region contains an infinite number of ensemble members is presented in Ref. 19, which also shows that the regions can be defined in terms of path-based quantities. Mathematical analyses of trajectory stratification algorithms can be found in Refs. 21 and 22.

Our paper is organized as follows. In Sec. II, we review the WE algorithm and recast it to show why an initialization bias is slow to disappear. This formulation also clarifies the relation of WE to the NEUS algorithm as described in Ref. 19, and we show how software for WE can be easily extended to enable trajectory stratification in Sec. III. In Sec. IV, we introduce a generalization of MSMs that represents the dynamics through a basis expansion[23–25] and, in turn, our strategy for accelerating sampling, which we term Basis-Accelerated NEUS (BAD-NEUS). We show that the NEUS algorithm is a special case of our new scheme. In Secs. V and VI, we demonstrate BAD-NEUS on a two-dimensional model with a known steady-state distribution and a molecular example.

## II. WEIGHTED ENSEMBLE (WE)

As described earlier, in WE, the state space is partitioned, and then trajectories are copied (cloned or split) or removed (killed or pruned) from the ensemble based on criteria. Mathematically, we denote the state of ensemble member (henceforth, walker) $i$ at time $t$ by $X_t^i$, and we associate with $i$ a weight $w_t^i$ such that $\sum_i w_0^i = 1$. We track the index of the region containing $X_t^i$ by an index process $J_t^i$. For example, $J_t^i$ might track the element of a partition of state space in which $X_t^i$ currently resides. However, much more general choices, such as ones based on path quantities[19,26] or distances between walkers,[27,28] are also possible. See Ref. 19 for further discussion.

Here, we use a relatively simple procedure and represent the splitting and pruning by resampling the ensemble within each region (or, more generally, value of the index process). Each iteration of the sampling thus consists of two steps: evolution according to the underlying dynamics and resampling the ensemble. In the evolution step, for each walker $i$, we numerically integrate for a time interval $\Delta$ to obtain $X_{t+\Delta}^i$ from $X_t^i$ and update $J_{t+\Delta}^i$ accordingly. In this step the weights are unchanged; $w_{t+\Delta}^i = w_t^i$. We note that $\Delta$ can be a random stopping time, not just a fixed time interval. An often useful choice is to take $t+\Delta$ to be the first time after time $t$ that the value of the index process $J^i$ changes.[3,4,16–19] In this case, each walker has a different value for $\Delta$. In the resampling step, for each value $k$ of the index process (e.g., for each region in a partition of state space), if there is at least one walker with $J_{t+\Delta}^i = k$, we select a number $N_k$. Then we sample from the set $\Phi = \{i : J_{t+\Delta}^i = k\}$ $N_k$ times with replacement according to the probabilities $p_i = w_{t+\Delta}^i / \sum_{i\in\Phi} w_{t+\Delta}^i$. For each index $r$ so chosen, we append $(X_{t+\Delta}^r, J_{t+\Delta}^r, \sum_{i\in\Phi} w_{t+\Delta}^i / N_k)$ to the new ensemble.

If the underlying dynamics are ergodic, in the case where $\Delta$ is a fixed time horizon, WE can be used to compute steady-state averages of functions as

$$\begin{aligned}&\mathbb{E}_{X_0\sim\pi}[g(X_0, X_1, \ldots, X_{\Delta-1})]\\ &\quad = \lim_{T\to\infty}\frac{1}{T}\sum_{n=0}^{T}\frac{1}{N}\sum_{i=0}^{N} g(X_{n\Delta}^i, X_{n\Delta+1}^i, \ldots, X_{n\Delta+\Delta-1}^i)\, w_{n\Delta}^i, \qquad (1)\end{aligned}$$

where the subscript on the expectation indicates that we draw the initial state $X_0$ from the steady-state distribution $\pi$, and $n$ indexes successive weighted ensemble iterations. We present necessary modifications for the case where $\Delta$ is a random variable later. Here and below, $X$ without a superscript indicates a realization of the underlying Markov process rather than a walker in an ensemble. An important example is the steady-state flux into a set $D$ of interest, which can be used to compute the MFPT to $D$ from another set in the right algorithmic setup.[10] This flux can be obtained by setting $g(X_0, \ldots, X_\Delta) = \mathbb{1}_{D^c}(X_0)\mathbb{1}_D(X_1)$, where $\mathbb{1}_D(x)$ is an indicator function that is 1 if $x \in D$ and 0 otherwise, and $D^c$ is the complement of the set $D$. This amounts to counting the number of walkers that enter $D$ in a single time step and summing the total weight of those walkers.

Having stated the basic WE algorithm, we now present WE in a new way. While this may initially appear to complicate the description, it reveals that WE is slow to converge and facilitates the introduction of approaches to accelerate convergence. In this description, which we call the distribution representation of WE, we directly evolve distributions rather than approximating them through individual walkers. To this end, we define the flux distributions

$$\overline{\pi}_\ell(dx|k) = \mathbb{P}[X_{S(\ell)} \in dx | J_{S(\ell)} = k], \tag{2}$$

where $dx$ is an infinitesimal volume in state space that is located at a specific value of $x$, and $S(\ell)$ is an increasing sequence of stopping times. For example, if we choose $S(\ell) = \ell\Delta$ for a fixed $\Delta$, the following description corresponds to the standard version of WE already sketched. Alternatively, we can let $S(\ell)$ be the time of the $\ell$th change in the value of the index process $J$, as in NEUS,[3,4,16–19]

$$\begin{aligned} S(0) &= 0, \\ S(1) &= \min\{t > 0 : J_t \neq J_{t-1}\}, \\ &\vdots \\ S(\ell) &= \min\{t > S(\ell-1) : J_t \neq J_{t-1}\}. \end{aligned} \tag{3}$$

This choice of stopping time has the advantage that walkers cannot go significantly beyond their regions, providing more control over the sampling.

In all practical implementations of the sampling algorithms that we discuss, the conditional flux distributions take the form of empirical distributions of $N$ walkers,

$$\overline{\pi}_\ell(dx|k) \approx \frac{1}{\overline{z}_\ell^k} \sum_{i=1}^{N} w^i_{S^i(\ell)} \delta_{X^i_{S^i(\ell)}}(dx) \mathbb{1}_{\{k\}}\Big(J^i_{S^i(\ell)}\Big), \tag{4}$$

where $\delta_x$ is the Dirac delta function centered at position $x$, and the normalization constant $\overline{z}_\ell^k$ is the total weight in region $k$ at time $S(\ell)$,

$$\overline{z}_\ell^k = \mathbb{P}[J_{S(\ell)} = k] \approx \sum_{i=1}^{N} w^i_{S^i(\ell)} \mathbb{1}_{\{k\}}\Big(J^i_{S^i(\ell)}\Big). \tag{5}$$

The outputs of the weighted ensemble iteration and the accelerated variants that we consider here are approximations of the steady-state conditional flux distributions, as well as estimates of the corresponding normalization constants (region weights). The conditional flux distributions are related to joint flux distributions through the normalization constants,

$$\overline{\pi}_\ell(dx, k) = \mathbb{P}[X_{S(\ell)} \in dx, J_{S(\ell)} = k], \tag{6}$$

$$= \overline{z}_\ell^k \overline{\pi}_\ell(dx|k), \tag{7}$$

and we work with $\overline{\pi}_\ell(dx,k)$ below.

In particular, we now write the evolution and resampling steps in terms of operators. For the former, we define the operator $\mathcal{U}$, which propagates the joint flux distribution of $(X_{S(\ell)}, J_{S(\ell)})$ for a duration of $S(\ell+1) - S(\ell)$,

$$\mathcal{U}\overline{\pi}_\ell(dx,k) = \overline{\pi}_{\ell+1}(dx,k). \tag{8}$$

In the long-time limit, this yields the eigenequation

$$\overline{\pi}(dx,k) = \mathcal{U}\overline{\pi}(dx,k), \tag{9}$$

where $\overline{\pi}$ is the steady-state joint flux distribution. In practice, one approximates the distributions through walkers, and we denote the corresponding evolution by $\tilde{\mathcal{U}}$. Mathematically,

$$\begin{aligned} &\tilde{\mathcal{U}}\left[\frac{1}{N}\sum_{i=1}^{N} w^i_{S^i(\ell)} \delta_{X^i_{S^i(\ell)}}(dx)\mathbb{1}_{\{k\}}\Big(J^i_{S^i(\ell)}\Big)\right] \\ &\quad = \frac{1}{N}\sum_{i=1}^{N} w^i_{S^i(\ell)} \delta_{X^i_{S^i(\ell+1)}}(dx)\mathbb{1}_{\{k\}}\Big(J^i_{S^i(\ell+1)}\Big). \end{aligned} \tag{10}$$

The subscript on the weight does not change because the evolution step only affects the state and index of a walker, not its weight. Operationally, this equation corresponds to propagating an ensemble's empirical distribution by running the dynamics until the stopping criterion and then assigning the initial weight of each walker to its final state and index. $\tilde{\mathcal{U}}$ as defined in (10) is an unbiased stochastic approximation of $\mathcal{U}$ in the sense that, for any function $g$ and any distribution $\rho$,

$$\sum_k \int g(x,k)[\mathcal{U}\rho](dx,k) = \mathbb{E}\left[\sum_k \int g(x,k)[\tilde{\mathcal{U}}\rho](dx,k)\right]. \tag{11}$$

If distributions are represented with some ansatz such as a neural network, one can also apply an approximate propagator based on a variational method as outlined in Refs. 29 and 30. Such a scheme would not require new trajectories to be run at each iteration, nor would it require a resampling step.

As noted earlier, we represent resampling through an operator, $\mathcal{R}_k$. We define it such that, for any distribution $\rho$ and any function $g$

$$\int g(x,k)\left(\frac{1}{\zeta_k}\rho(dx,k)\right) = \mathbb{E}\left[\int g(x,k)\,\mathcal{R}_k\rho\right], \tag{12}$$

and

$$\zeta_k = \int \rho(dx,k). \tag{13}$$

Here, the LHS of (12) selects the portion of the distribution $\rho$ in region $k$ and then renormalizes it, while the RHS corresponds to resampling from the distribution. This equation expresses the condition that the resampling operator must preserve the weighted distribution in the sense that the empirical mean of any function is the same in expectation after resampling. A simple choice that is commonly employed for the operator $\mathcal{R}_k$ is to sample the set of walkers in region $k$ at the end of the evolution (denoted $K$) from a multinomial distribution with probability proportional to $\{w^i_{S^i(\ell)}\mathbb{1}_{\{k\}}(J^i_{S^i(\ell)})\}_{i\in K}$ with $N_k$ trials and then return the distribution

$$\mathcal{R}_k\left[\frac{1}{N}\sum_{i=1}^{N} w^i_{S^i(\ell)} \delta_{X^i_{S^i(\ell)}}(dx)\mathbb{1}_{\{k\}}(J^i_{S^i(\ell)})\right] = \frac{1}{N_k}\sum_{i\in K} \delta_{X^i_{S^i(\ell)}}(dx), \tag{14}$$

and normalization

$$\overline{z}_\ell^k \approx \sum_i \mathbb{1}_{\{k\}}(J^i_{S^i(\ell)}) w^i_{S^i(\ell)}. \tag{15}$$

In other words, (14) shows that $\mathcal{R}_k$ constructs the state distribution from a sum over the resampled walkers in each region. There are other possible ways of resampling in the walker representation,[27,28,31] and these may in practice be better. However, any choice that satisfies (12) suffices for our discussion.

Finally, we connect the flux distributions $\overline{\pi}(dx|k)$ to the steady-state distribution $\pi(x,k)$ for the process $(X_t, J_t)$, where the absence of superscripts again indicates the underlying Markov process rather than a walker in an ensemble. We do this with a key identity, which we now state. For a function $g$ of a length $\tau$ trajectory,

$$\mathbb{E}_{(X_0,J_0)\sim\pi}[g(X_0,J_0,\ldots,X_\tau,J_\tau)] = \frac{\sum_k \overline{z}^k \int \overline{\pi}(dx|k)\mathbb{E}_{X_0=x,J_0=k}\left[\sum_{t=0}^{S(1)-1} g(X_t,J_t,\ldots,X_{t+\tau},J_{t+\tau})\right]}{\sum_k \overline{z}^k \int \overline{\pi}(dx|k)\mathbb{E}_{X_0=x,J_0=k}[S(1)]}, \tag{16}$$

where now $\pi$ is the steady-state joint distribution of $(X_t, J_t)$. This identity says that, for each region $k$, we draw initial walker states $X_0$ from $\overline{\pi}(dx|k)$ and set $J_0 = k$; we evolve those walkers until they leave region $k$; then we compute averages over them and weight the contribution from the walkers that started in region $k$ by $\overline{z}^k$, which is the steady-state limit of $\overline{z}^k_\ell$. We derive (16) in the Appendix. The distribution representation of WE is summarized in Algorithm 1.

We see that each step of WE applies the operator $\mathcal{U}$ to the previous distribution $\overline{\pi}_\ell(dx,k)$. Therefore, WE can be seen as performing a power iteration in $\mathcal{U}$, starting from some initial distribution. The resampling step in WE plays a role analogous to the normalization step in standard power iteration in the following sense. In standard power iteration, the normalization step serves to prevent the iterate from becoming too small or too large and introducing numerical instability. In WE, the resampling step serves to prevent any of the region distributions from becoming poorly sampled, which would increase variance. However, the normalization step in power iteration and, in turn, the resampling step in WE do not accelerate decay of initialization bias, nor is the WE autocorrelation time reduced relative to direct sampling.

Algorithm 1 suggests that, if $\tilde{\overline{\pi}} = \overline{\pi}$ in step 3, convergence would be achieved. Therefore, our strategy for accelerating WE is to replace the power iteration step with a more accurate approximation. Specifically, we compute changes of measure (i.e., reweighting factors) $v_\ell$ such that

$$\overline{\pi}(dx,k) \approx v_\ell(x,k)\overline{\pi}_\ell(dx,k). \tag{17}$$

If we set up our approximation such that whenever $\pi_\ell = \pi$, we get $v_\ell = 1$, then the resulting modification of WE will have a fixed point at the true steady-state distribution, and there will be no approximation error resulting from deficiencies in the approximation algorithm's specific ansatz. Our scheme has the advantage that it can potentially approach steady state much faster than WE. This is the idea that we develop in Secs. III and IV.

**ALGORITHM 1.** WE (distribution representation).

```
Require: Approximate Markov propagator $\tilde{\mathcal{U}}$, resampling operator
    $\mathcal{R}_k$, starting flux distributions $\{\overline{\pi}_0(dx|k)\}_{k=1}^n$, initial
    region weights $\{\overline{z}_0^k\}_{k=1}^n$ with $\sum_k \overline{z}_0^k = 1$, total number
    of iterations $L$, number of strata $n$.
1:  for $\ell = 0, \ldots, L$ do
2:      $\overline{\pi}_\ell(dx,k) \leftarrow \overline{z}_\ell^k \overline{\pi}_\ell(dx|k)$
3:      $\tilde{\overline{\pi}} \leftarrow [\tilde{\mathcal{U}}\overline{\pi}_\ell](dx,k)$        ▷ one power iteration step
4:      for $k = 1, \ldots, n$ do
5:          $\overline{\pi}_{\ell+1}(dx|k), \overline{z}_{\ell+1}^k \leftarrow \mathcal{R}_k[\tilde{\overline{\pi}}]$
6:      end for
7:  end for
8:  return $\{\overline{\pi}_L(dx|k)\}_{k=1}^n$ and $\{\overline{z}_L^k\}_{k=1}^n$
```

## III. NONEQUILIBRIUM UMBRELLLA SAMPLING (NEUS)

We now present NEUS as a simple extension to WE that accelerates convergence in the way suggested in (17); EM[6] can be formulated similarly. Our development is based on the algorithm in Ref. 19, which corrects a small systematic bias in earlier NEUS papers.[3,4,16–18] We begin by making the observation that, at steady state,

$$\mathbb{E}_{X_0\sim\overline{\pi}}[\mathbb{1}_{\{k\}}(J_0)] = \mathbb{E}_{X_0\sim\overline{\pi}}[\mathbb{1}_{\{k\}}(J_{S(1)})], \tag{18}$$

or, in terms of the density we wish to compute,

$$\int \overline{\pi}(dx,k) = \sum_j \int \overline{\pi}(dx,j)\mathbb{E}_{X_0=x,J_0=j}[\mathbb{1}_{\{k\}}(J_{S(1)})]. \tag{19}$$

This equation says that the steady-state probability of index process $k$ is the same as the total probability of walkers that were initialized from the steady-state distribution and flowed into (or remained in) $k$ at the stopping time. This observation introduces one equation per possible value of the index $k$. Therefore, we can parameterize the change of measure with as many free parameters,

$$\overline{\pi}(dx,k) \approx c_\ell^k \overline{z}_\ell^k \overline{\pi}_\ell(dx|k). \tag{20}$$

Substituting this ansatz into (19) and simplifying yields the matrix equation

$$c_\ell^k \overline{z}_\ell^k = \sum_j c_\ell^j \overline{z}_\ell^j \overline{G}_{jk}, \tag{21}$$

where

$$\overline{G}_{jk} = \int \overline{\pi}(dx|j)\mathbb{E}_{X_0=x,J_0=j}[\mathbb{1}_{\{k\}}(J_{S(1)})], \tag{22}$$

tracks the total flux from region $j$ to region $k$. We summarize NEUS in Algorithm 2.

We thus see that NEUS is distinguished from WE in its original formulation by steps 2–4, in which the total fluxes between pairs of regions are estimated and the region weights are adjusted to satisfy a global balance condition. In practice, walkers are drawn and their dynamics are simulated to determine $S(\ell+1)$ in the loop over

**ALGORITHM 2.** NEUS.

```
Require: Approximate Markov propagator $\tilde{\mathcal{U}}$, resampling operator
    $\mathcal{R}_k$, starting flux distributions $\{\overline{\pi}_0(dx|k)\}_{k=1}^{n}$, initial
    region weights $\{\overline{z}_0^k\}_{k=1}^{n}$ with $\sum_k \overline{z}_0^k = 1$, total number
    of iterations $L$, number of strata $n$.
1:  for $\ell = 0, \ldots, L$ do
2:      $\overline{G}_{jk} \leftarrow \int \overline{\pi}_\ell(dx|j)\mathbb{E}_{X_0=x,J_0=j}[\mathbb{1}_{\{k\}}(J_{S(1)})]$
3:      Solve $c_\ell \overline{z}_\ell \overline{G} = c_\ell \overline{z}_\ell$ for $c_\ell \overline{z}_\ell$
4:      $\overline{\pi}_\ell(dx,k) \leftarrow c_\ell^k \mathbb{1}_{\{k\}} \overline{z}_\ell^k \overline{\pi}_\ell(dx|k)$
5:      $\tilde{\overline{\pi}} \leftarrow \tilde{\mathcal{U}}\overline{\pi}_\ell(dx,k)$
6:      for $k = 1, \ldots, n$ do
7:          $\overline{\pi}_{\ell+1}(dx|k), \overline{z}_{\ell+1}^k \leftarrow \mathcal{R}_k[\tilde{\overline{\pi}}]$
8:      end for
9:  end for
10: return $\{\overline{\pi}_L(dx|k)\}_{k=1}^{n}$ and $\{\overline{z}_L^k\}_{k=1}^{n}$
```

$\ell$ prior to step 2 (the resulting state distribution is used later in step 5); then, $\overline{G}_{jk}$ is computed in step 2 from walkers at iteration $\ell$ as

$$\overline{G}_{jk} \approx \frac{\sum_i \mathbb{1}_{\{j\}}\left(J^i_{S^i(\ell)}\right)\mathbb{1}_{\{k\}}\left(J^i_{S^i(\ell+1)}\right)}{\sum_i \mathbb{1}_{\{j\}}\left(J^i_{S^i(\ell)}\right)}. \tag{23}$$

We solve for product $c_\ell \overline{z}_\ell$ directly and update the approximation for $\overline{\pi}$ accordingly. We then proceed as in WE.

If the flux distributions and region weights assume their steady-state values such that $\overline{\pi}_\ell(dx|k) = \overline{\pi}(dx|k)$ and $\overline{z}_\ell^k = \overline{z}^k$, a global balance condition is satisfied. That is,

$$\overline{z}^k = \sum_j \overline{z}^j \overline{G}_{jk}. \tag{24}$$

In this limit, (21) is solved by $c^k = 1$, and NEUS has the correct fixed point (i.e., the same fixed point as $\mathcal{U}$, which encodes the unbiased dynamics).

## IV. BAD-NEUS

To improve on NEUS, it is necessary to improve the approximation of the change of measure in (17). Here, we do so by introducing a basis expansion. This allows us to vary the expressivity of the approximation through the number of basis functions, which can exceed the number of regions, in contrast to the steady-state condition in (18).

To this end, we note that, for any lag time $\tau$ and any function $f(x,k)$,

$$\mathbb{E}_{(X_0,J_0)\sim\pi}[f(X_0,J_0) - f(X_\tau,J_\tau)] = 0. \tag{25}$$

We stress that this relation holds for any fixed time $\tau$. Expanding this expectation using (16), with $g(X_0,J_0,\ldots,X_\tau,J_\tau) = f(X_0,J_0) - f(X_\tau,J_\tau)$ and multiplying through by the normalization in the denominator, we obtain

$$0 = \sum_k \overline{z}^k \int \overline{\pi}(dx|k)\mathbb{E}_{X_0=x,J_0=k}\left[\sum_{t=0}^{S(1)-1}(f(X_t,J_t) - f(X_{t+\tau},J_{t+\tau}))\right]. \tag{26}$$

We then introduce the basis set $\{\phi_p(x,k)\}_p$ for the change of measure in (17) and write

$$\pi(dx,k) = \overline{z}^k\overline{\pi}(dx|k) \approx \sum_p c_\ell^p \phi_p(x,k)\overline{z}_\ell^k\overline{\pi}_\ell(dx|k). \tag{27}$$

Making the choice $f = \phi_r$ and inserting the basis expansion into (26) gives

$$0 = \sum_k \sum_p c_\ell^p \overline{z}_\ell^k \int \phi_p(x,k)\overline{\pi}_\ell(dx|k) \times \mathbb{E}_{X_0=x,J_0=k}\left[\sum_{t=0}^{S(1)-1}(\phi_r(X_t,J_t) - \phi_r(X_{t+\tau},J_{t+\tau}))\right]. \tag{28}$$

This is a linear system that can be solved for the expansion coefficients,

$$0 = \sum_p c_\ell^p M_{pr}, \tag{29}$$

with the matrix entries given by

$$M_{pr} = \sum_k \overline{z}_\ell^k \int \phi_p(x,k)\overline{\pi}_\ell(dx|k) \times \mathbb{E}_{X_0=x,J_0=k}\left[\sum_{t=0}^{S(1)-1}(\phi_r(X_t,J_t) - \phi_r(X_{t+\tau},J_{t+\tau}))\right]. \tag{30}$$

The basis set must include the constant function in its span so that the system has a unique nontrivial solution.

We summarize the BAD-NEUS algorithm in Algorithm 3. If the flux distributions and region weights assume their steady-state values such that $\overline{\pi}_\ell(dx|k) = \overline{\pi}(dx|k)$ and $\overline{z}_\ell^k = \overline{z}^k$, then (26) is satisfied for all $f$. In that case, just as (24) and (21) lead to $c^k = 1$ in NEUS, (26) and (28) lead to $\sum_p c_p\phi(x,k) = 1$ in BAD-NEUS. In this

**ALGORITHM 3.** BAD-NEUS.

```
Require: Approximate Markov propagator $\tilde{\mathcal{U}}$, resampling operator
    $\mathcal{R}_k$, starting flux distributions $\{\overline{\pi}_0(dx|k)\}_{k=1}^{n}$, initial region
    weights $\{\overline{z}_0^k\}_{k=1}^{n}$ with $\sum_k \overline{z}_0^k = 1$, basis set $\{\phi_p(x,k)\}_p$,
    lag time $\tau$, number of past iterations to retain $h$,
    total number of iterations $L$, number of strata $n$.
1:  for $\ell = 0, \ldots, L$ do
2:      $\overline{\pi}_\ell \leftarrow \overline{\pi}_{\ell,h}$
3:      $\Delta\phi \leftarrow \sum_{t=0}^{S(1)-1} \phi_r(X_t,J_t) - \phi_r(X_{t+\tau},J_{t+\tau})$
4:      $M_{pr} \leftarrow \sum_k \int \overline{z}_\ell^k \phi_p(x,k)\overline{\pi}_\ell(dx|k)\mathbb{E}_{X_0=x,J_0=k}[\Delta\phi]$
5:      Solve $cM = 0$
6:      $\overline{\pi}_\ell(dx,k) \leftarrow \sum c_\ell^p \phi_p(x,k)\overline{z}_\ell^k\overline{\pi}_\ell(dx|k)$
7:      $\tilde{\overline{\pi}} \leftarrow \tilde{\mathcal{U}}\overline{\pi}_\ell(dx,j)$
8:      for $k = 1, \ldots, n$ do
9:          $\overline{\pi}_{\ell+1}(dx|k), \overline{z}_{\ell+1}^k \leftarrow \mathcal{R}_k[\tilde{\overline{\pi}}]$
10:     end for
11: end for
12: return $\{\overline{\pi}_L(dx|k)\}_{k=1}^{n}$ and $\{\overline{z}_L^k\}_{k=1}^{n}$
```

case, step 6 in Algorithm 3 reduces to $\overline{\pi}_{\ell+1}(dx,k) \leftarrow \overline{z}^{k}\overline{\pi}(dx|k)$, and BAD-NEUS has the correct fixed point.

Specific choices for the basis functions, lag time, and stopping time correspond to existing methods. NEUS as described in Sec. III and Algorithm 2 corresponds to the choice

$$\phi_p(x,k) = \mathbb{1}_{\{p\}}(k), \tag{31}$$

$\tau = 1$, and $\Delta$ as defined in (3). We can obtain a version of iteratively restarted WE as suggested in Refs. 8 and 13 by choosing the basis set to be indicator functions that are one on sets that are finer than the stratification ("microbins"), $\tau = 1$, and a fixed stopping time $\Delta$. For these choices, the sum over $t$ in (30) telescopes, and $M$ can be row-normalized and written as $T - I$, where $T$ is the transition matrix for the MSM defined by the basis set.

Consistent with WE and NEUS, in practice, we estimate the integral $M_{pr}$ through a sum over walkers,

$$\begin{aligned} M_{pr} \approx \sum_{k}\sum_{i} \overline{z}_{\ell}^{k}\phi_p(X^i_{S^i(\ell)},k)\mathbb{1}_{\{k\}}(J^i_0) \\ \times\left[\sum_{t=S^i(\ell)}^{S^i(\ell+1)-1}(\phi_r(X^i_t,J^i_t) - \phi_r(X^i_{t+\tau},J^i_{t+\tau}))\right]. \end{aligned} \tag{32}$$

Given $M$ and, in turn, the estimated expansion coefficients, we update the weights as

$$w^i_{S^i(\ell)} = \frac{\overline{z}_{\ell}^{J^i_{S^i(\ell)}}}{Z}\sum_{p} c^p_{\ell}\phi_p\left(X^i_{S^i(\ell)},J^i_{S^i(\ell)}\right), \tag{33}$$

where $Z$ normalizes the total of the walker weights to one. Operationally, we loop over each walker before the resampling step and evaluate (33) to adjust its weight. Physically, this corresponds to apportioning the weight of the region containing the walker $[\overline{z}_{\ell}^{J^i_{S^i(\ell)}}/Z$ in (33)] according to the basis expansion representing the change of measure to the steady-state. We then proceed with the resampling step as in the usual WE algorithm.

To reduce variance in our estimate of the matrix $M$ to ensure a stable solve, it is often useful to work with the mixture distribution for the last $h$ iterations,

$$\overline{\pi}_{\ell,h}(dx|k) = \frac{1}{h}\sum_{t=\ell-h+1}^{\ell}\overline{\pi}_t(dx|k). \tag{34}$$

When working with walkers, this mixture distribution corresponds to concatenating the walkers from the last $h$ iterations and, therefore, contains more data than using a single iteration. We simply substitute $\overline{\pi}_{\ell,h}$ for $\overline{\pi}_{\ell}$ in all of our algorithms. A unified walker-based algorithm that we use in practice is given in Algorithm 4, with the additional steps needed for NEUS and BAD-NEUS in Algorithm 5.

Finally, we note that the general strategy of improving the approximation of the change of measure is not limited to using a basis expansion. Just as one can use various means to solve the equations of the operator that encodes the statistics of the dynamics,[23,24,30,32] one can represent the change of measure here by either a basis expansion or a nonlinear representation. In particular, our tests based on the neural-network approach in Ref. 30 show significant promise (unpublished results).

**ALGORITHM 4.** Trajectory stratification (walker representation).

```
Require: Starting ensemble of walkers $E_0 = \{(X^i_0, J^i_0, w^i_0)\}_{i=1}^{N}$, with $\sum_{i=1}^{N} w^i_0 = 1$, lag time $\tau$, number of past
  iterations to retain $h$, total number of iterations $L$, number of strata $n$.
for $\ell = 0,\dots,L$ do
    $N \leftarrow \mathrm{length}(E_\ell)$
    Generate a list of trajectories
    $T_\ell \leftarrow \left\{\left[(X^i_{S^i(\ell)}, J^i_{S^i(\ell)}, w^i_{S^i(\ell)}), (X^i_{S^i(\ell)+1}, J^i_{S^i(\ell)+1}, w^i_{S^i(\ell)}), \dots, (X^i_{S^i(\ell+1)+\tau}, J^i_{S^i(\ell+1)+\tau}, w^i_{S^i(\ell)})\right]\right\}_{i=1}^{N}$, where $(X^i_{S^i(\ell)}, J^i_{S^i(\ell)}, w^i_{S^i(\ell)}) = E_\ell[i]$
    $T_\ell \leftarrow \mathrm{concatenate}(\{T_\ell, T_{\ell-1},\dots,T_{\ell-h+1}\})$
    $N \leftarrow \mathrm{length}(T_\ell)$
    Renormalize the weights in $T_\ell$ by dividing each by $h$.
    Update weights $\{w^i_{S^i(\ell)}\}_{i=1}^{N}$ using Algorithm 5. (Omit for weighted ensemble)
    Initialize an empty list $E_{\ell+1} = \{\}$
    for $k = 1,\dots,n$ do
        $\overline{z}^k_{\ell+1} \leftarrow \sum_{i=1}^{N} w^i_{S^i(\ell)}\mathbb{1}_{\{k\}}(J^i_{S^i(\ell+1)})$
        for $r = 1,\dots,N_k$ do
            Sample an index $b$ with probability proportional to $w^b_{S^b(\ell)}\mathbb{1}_{\{k\}}(J^b_{S^b(\ell+1)})$
            Append to $E_\ell$ the configuration $(X^b_{S^b(\ell+1)}, J^b_{S^b(\ell+1)}, \overline{z}^k_{\ell+1}/N_k)$
        end for
    end for
end for
return $T_L$ and $\{\overline{z}^k_L\}_{k=1}^{n}$
```

**ALGORITHM 5.** Approximating the steady-state flux distribution (walker representation).

```
Require: List of trajectories $T_\ell$, lag time $\tau$, basis functions $\{\phi_p\}_p$, region weights $\{\bar{z}_\ell^k\}_{k=1}^n$
$M_{pr} \leftarrow \sum_{ki} \bar{z}_\ell^k \phi_p(X^i_{S^i(\ell)}, k)\mathbb{1}_{\{k\}}(J_0^i)\Big[\sum_{t=S^i(\ell)}^{S^i(\ell+1)-1}(\phi_r(X_t^i, J_t^i) - \phi_r(X_{t+\tau}^i, J_{t+\tau}^i))\Big]$
Solve $c_\ell M = 0$
$w^i_{S^i(\ell)} \leftarrow (\bar{z}_\ell^{J^i_{S^i(\ell)}}/Z)\sum_p c_\ell^p \phi_p\Big(X^i_{S^i(\ell)}, J^i_{S^i(\ell)}\Big)$
return $\{w^i_{S^i(\ell)}\}_{i=1}^N$
```

## V. TWO-DIMENSIONAL TEST SYSTEM

We first illustrate our approach by sampling a two-dimensional system, which enables us to compare estimates of the steady-state distribution to the Boltzmann probability to ensure that simulations are run until convergence. Specifically, the system $x = (u, v)$ is defined by the Müller–Brown potential,[33] which is a sum of four Gaussian functions,

$$V_{\mathrm{MB}}(u,v) = \frac{1}{20}\sum_{i=1}^{4} C_i \exp\Big[a_i(u-v_i)^2 + b_i(u-u_i)(v-v_i) + c_i(v-v_i)^2\Big]. \tag{35}$$

For all results shown, we use $C_i = \{-200, -100, -170, 15\}$, $a_i = \{-1, -1, -6.5, 0.7\}$, $b_i = \{0, 0, 11, 0.6\}$, $c_i = \{-10, -10, -6.5, 0.7\}$, $u_i = \{1, -0.27, -0.5, -1\}$, and $v_i = \{0, 0.5, 1.5, 1\}$. The potential with these parameter choices is shown in Fig. 1. The presence of multiple metastable states and a minimum energy pathway that does not parallel the axes of the variables used for the numerical integration make this system representative of difficulties commonly encountered in molecular simulations. Because the model is two-dimensional, we can readily visualize results and compare them with statistics independently computed using the grid-based scheme in Ref. 30.

We evolve the system with overdamped Langevin dynamics, discretized with the BAOAB algorithm,[34]

$$X_{t+dt} = X_t - \nabla V_{\mathrm{MB}}(X_t)dt + \sqrt{\frac{dt}{2\beta}}(Z_t + Z_{t-dt}), \tag{36}$$

where $dt$ is the time step, $\beta$ is the inverse temperature, and $Z_t \sim N(0, I)$ is a random vector with components drawn from the normal distribution with zero mean and unit standard deviation ($I$ is the two-dimensional identity matrix). For all results shown, we use $dt = 0.001$ and $\beta = 2$.

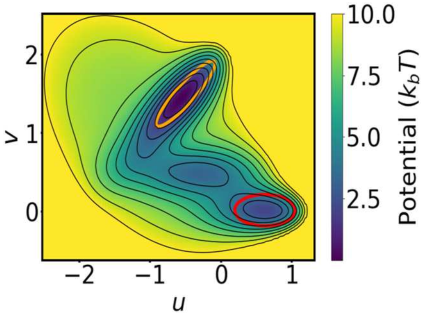

**FIG. 1.** Müller–Brown potential. Orange and red ovals indicate states $A$ and $B$, respectively. Contours are spaced every $k_BT$.

### A. Comparison of algorithms

A key point of our theoretical development is that NEUS and BAD-NEUS are simple elaborations of WE. This enables us to define a unified walker-based algorithm that we use in practice (Algorithm 4, with the added operations needed for NEUS and BAD-NEUS in Algorithm 5). Splitting and stratification are defined through the rule for switching the index process. The sequence of stopping times $S(\ell)$ is then determined by the index process through (3). The update rule for $J_t$ that we employ for all three algorithms is as follows. Let $\{\psi_k(x)\}_{k=1,\ldots,n}$ be a set of nonnegative functions. If $\psi_{J_t}(X_{t+dt}) > 0$, then $J_{t+dt} = J_t$; otherwise, $\mathbb{P}[J_{t+dt} = k] = \psi_k(X_{t+dt})/\sum_k \psi_k(X_{t+dt})$. That is, the value of the index process remains the same until the walker leaves the support of $\psi_{J_t}$, and then a new index value $k$ is drawn with likelihood proportional to $\psi_k(X_{t+dt})$. For the Müller–Brown model, we use

$$\psi_k(u,v) = \begin{cases} 1, & \text{if } |v - v_k| < \varepsilon_k \text{ and } 1 < k < n, \\ 1, & \text{if } v - v_k < \varepsilon_k \text{ and } k = 1, \\ 1, & \text{if } v_k - v > \varepsilon_k \text{ and } k = n, \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

Unless otherwise indicated, we set $n = 10$, space the $v_k$ uniformly in the interval $[-0.2, 1.8]$, and set $\varepsilon_k = 0.6(v_{k+1} - v_k)$, so that the regions of support (strata) overlap. This choice prevents walkers in barrier regions from rapidly switching back and forth between index values, limiting the overhead of the algorithms. Unless otherwise indicated, we use 2000 walkers per region and run them until their index processes switch values. Initial configurations for $J_0 = k$ are generated by uniformly sampling the support of $\psi_k$.

For BAD-NEUS, we use a basis set consisting of ten indicator functions per stratum. The indicator functions are determined by clustering all the samples in a stratum with $k$-means clustering and partitioning it into Voronoi polyhedra based on the cluster means. Unless otherwise indicated, we compute statistics using data from the last $h = 3$ iterations and use a lag time of $\tau = 10$ time steps.

We measure convergence by computing the root mean square (RMS) difference in $\ln(\tilde{\pi}(x))$ from $-\beta V_{\mathrm{MB}}(x)$, where $\tilde{\pi}(x)$ is an estimate of the steady-state distribution from the normalized histogram of samples. For the histogram, we use a uniform $50 \times 50$ grid on the rectangle defined by the minimum and maximum values in the NEUS dataset. Since some grid regions are empty because they

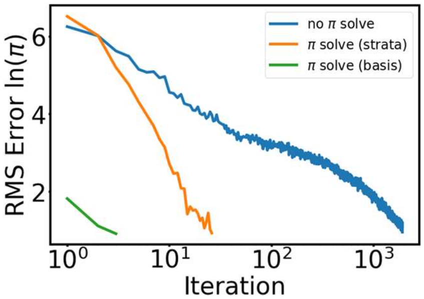

**FIG. 2.** Convergence of the steady-state distribution of the Müller–Brown system for WE as defined in Algorithm 1 (blue, no $\pi$ solve), NEUS as defined in Algorithm 2 [orange, $\pi$ solve using the strata in (37) as a basis], and BAD-NEUS as defined in Algorithm 3 (green, $\pi$ solve using a basis defined by $k$-means clustering, as described in the text).

correspond to energies too high for NEUS to sample, we only compute errors over bins with $V_{\mathrm{MB}} < 7$. We stop each simulation when the RMS difference drops below one. Results are shown in Fig. 2. We see that WE (without any acceleration strategy) requires about 73 times more iterations than NEUS, which in turn requires about 13 times more iterations than BAD-NEUS to reach the convergence criterion.

## B. Choice of hyperparameters

Having demonstrated that trajectory stratification outperforms WE (without any acceleration strategy) for this system, we now examine the effect of key hyperparameters in NEUS and BAD-NEUS.

First, we investigate the effect of the number of strata. To compare simulations that require essentially the same resources, we hold the total number of walkers fixed at 40 000, and we divide the walkers uniformly across the strata. For these simulations, we use a lag time of $\tau = 1$ and retain $h = 3$ past iterations. In Fig. 3, we plot both the number of iterations to reach convergence and the total number of time steps needed for convergence as we vary the number of strata. We see that the number of BAD-NEUS iterations required for convergence increases linearly with the number of strata. However, the lengths of trajectories decrease because the strata are spaced more closely. Due to this interplay, the total effort to achieve convergence decreases until about 12 strata and then levels off. While actual performance will depend on the overhead incurred by stopping and starting the dynamics engine and computing the BAD-NEUS weights, these results indicate that finer stratification is better.

Next, we examine the dependence of convergence on the number of iterations used to compute statistics, $h$ (other hyperparameters are set to the default values given earlier). Using data from a larger number of iterations allows for more averaging, but it can also contaminate the statistics with samples obtained before the steady-state distribution has converged. While the trends are clearer for NEUS than BAD-NEUS because the former converges more slowly, Fig. 4 indicates that retaining fewer iterations is better for both NEUS and BAD-NEUS. It is, therefore, important to use enough walkers per iteration that data from past iterations need not be retained. If one has access to a large number of processing elements and many walkers can be simulated in parallel, this is not a severe restriction. Once convergence is achieved, one can begin accumulating data to reduce variance.

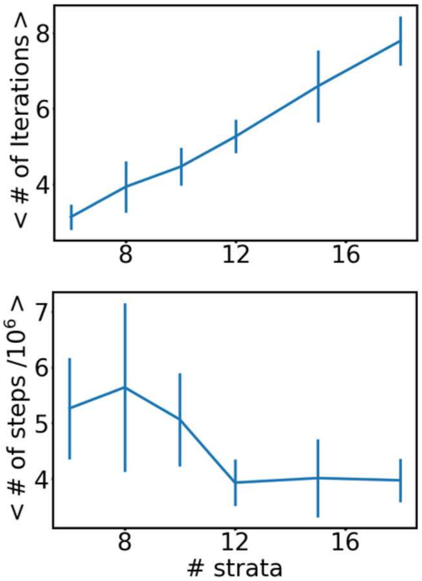

**FIG. 3.** Effect of the number of strata on convergence of BAD-NEUS applied to the Müller–Brown system. The number of iterations required (top) and the total number of time steps (bottom)—a proxy for computational effort—until convergence. The total number of walkers is held fixed. Error bars are the standard deviation over ten replicate simulations. For these simulations, there are 40 000 walkers total, and the lag time is $\tau = 1$ time step.

Finally, we investigate the impact of the lag time in Fig. 5 (other hyperparameters are set to the default values given earlier). We find

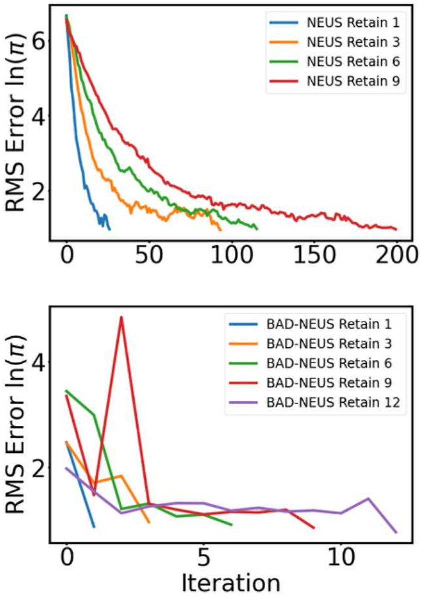

**FIG. 4.** Effect of retaining past iterations on convergence of NEUS (top) and BAD-NEUS (bottom) applied to the Müller–Brown system.

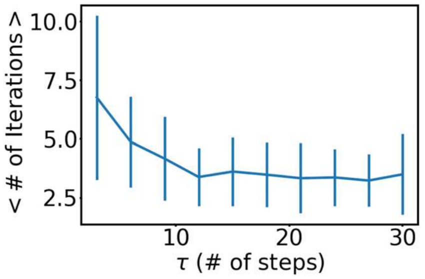

**FIG. 5.** Dependence of convergence on the lag time for BAD-NEUS applied to the Müller–Brown system. Error bars are the standard deviation over ten replicate simulations.

that using longer lag times weakly decreases the number of iterations required for convergence.

### C. Kinetic statistics

A common problem in molecular dynamics is that while equilibrium averages are relatively straightforward to calculate from biased simulations (via, for example, the various methods reviewed in Ref. 35), dynamical averages are much harder. In this section, we use BAD-NEUS to compute dynamical averages efficiently. To do so, we split the ensemble of transition paths based on the last metastable state visited as previously for NEUS.[17,26,36] We specifically compute the backward committor, $q_-$, which is the probability that the system last visited a reactant state $A$ rather than a product state $B$, and the transition path theory (TPT) rate, defined as the mean number of $A$ to $B$ transitions per unit time divided by the fraction of time spent last in $A$. The former can be obtained without saving additional information once we split the path ensemble. The forward committor, $q_+$, which is the probability that the system next visits state $B$ rather than state $A$, can also be obtained from NEUS but requires saving additional information.[26] Here, both of the systems that we consider are microscopically reversible, so we can obtain $q_+$ from $q_+ = 1 - q_-$.

For the Müller–Brown system, we consider the following history dependent stratification. Let $\{\psi_k^A(x)\}_{k=1,\ldots,n}$ and $\{\psi_k^B(x)\}_{k=1,\ldots,m}$ be non-negative functions; let $\psi_k = \psi_k^A$ if $k \le n$; otherwise, let $\psi_k = \psi_{k-n}^B$, and let $\mathbb{1}_A$ and $\mathbb{1}_B$ be indicator functions on sets $A$ and $B$, respectively. Let $T_{A\cup B}^-(t)$ be the time the system sampled at time $t$ was last in $A$ or $B$ so that $\mathbb{1}_A(X_{T_{A\cup B}^-(t)}) = 1 - \mathbb{1}_B(X_{T_{A\cup B}^-(t)})$ reports whether $A$ rather than $B$ was last visited. Define the index process by the following update rule:

$$\mathbb{P}[J_{t+dt} = k] = \begin{cases} \psi_k(X_{t+dt})\mathbb{1}_A(X_{T_{A\cup B}^-(t+dt)})/\sum_i \psi_i^A(X_{t+dt}), & k \le n, \psi_{J_t}(X_{t+dt}) = 0 \text{ or} \\ & \quad \mathbb{1}_A(X_{T_{A\cup B}^-(t+dt)}) \ne \mathbb{1}_A(X_{T_{A\cup B}^-(t)}), \\ \psi_k(X_{t+dt})\mathbb{1}_B(X_{T_{A\cup B}^-(t+dt)})/\sum_i \psi_i^B(X_{t+dt}), & k > n, \psi_{J_t}(X_{t+dt}) = 0 \text{ or} \\ & \quad \mathbb{1}_A(X_{T_{A\cup B}^-(t+dt)}) \ne \mathbb{1}_A(X_{T_{A\cup B}^-(t)}), \\ \mathbb{1}_{\{k\}}(J_t), & \text{otherwise.} \end{cases} \tag{38}$$

Therefore, there are $n + m$ total strata, the first $n$ of which contain walkers that last visited $A$, and the remainder of which contain walkers that last visited $B$. The steady-state distribution of walkers that last visited $A$ is proportional to $\pi(x)q_-(x)$, and that which last visited $B$ is proportional to $\pi(x)(1 - q_-(x))$. This ensemble lets us compute dynamical averages. The backward committor projected onto a space of CVs can be computed using

$$q_-^\theta(s) = \frac{\mathbb{E}[\mathbb{1}_{\{0,\ldots,n\}}(J_t)\mathbb{1}_{\{ds\}}(\theta(X_t))]}{\mathbb{E}[\mathbb{1}_{\{ds\}}(\theta(X_t))]}, \tag{39}$$

where $\theta$ is a vector with components that are the CVs and $ds$ is a bin in the space. The TPT rate constant is defined as

$$k_{AB} = \frac{\mathbb{E}_\pi[q_-(x)\mathbb{1}_{B^c}(x)\mathcal{T}_1\mathbb{1}_B(x)]}{\mathbb{E}_\pi[q_-(x)]}, \tag{40}$$

where $B^c$ is the complement of $B$, and $\mathcal{T}_t$ is the operator that describes the evolution of expectations of functions,

$$\mathcal{T}_t g(x,j) = E_{X_0=x,J_0=j}[g(X_t,J_t)]. \tag{41}$$

In practice, we compute (40) from (16) by choosing

$$g(X_0, J_0 \ldots, X_t, J_t) = \mathbb{1}_{B^c}(X_0)\mathbb{1}_B(X_1)\mathbb{1}_{\{0,\ldots,n\}}(J_0), \tag{42}$$

for the numerator and

$$g(X_0, J_0 \ldots, X_t, J_t) = \mathbb{1}_{\{0,\ldots,n\}}(J_0), \tag{43}$$

for the denominator. This corresponds to the steady-state flux into $B$ from trajectories that originated in $A$.

For the Müller–Brown system, we define states $A$ and $B$ as

$$\begin{gathered} A = \{u, v : 6.5(u+0.5)^2 - 11(u+0.5)(v-1.5) + 6.5(u-1.5)^2 < 0.3\}, \\ B = \{u, v : (u-0.6)^2 + 0.5(v-0.02)^2 < 0.2\}. \end{gathered} \tag{44}$$

We use a similar index process construction as for the equilibrium calculations, except there are 10 $v_k$ that are evenly spaced in the interval (0, 1.6) for $\psi_k^A$ and 10 $v_k$ that are evenly spaced in the interval (−0.3, 0.8) for $\psi_k^B$, for a total of 20 strata. We use different definitions for $\psi_k^A$ and $\psi_k^B$ because, for walkers originating from state $A$ ($B$), a

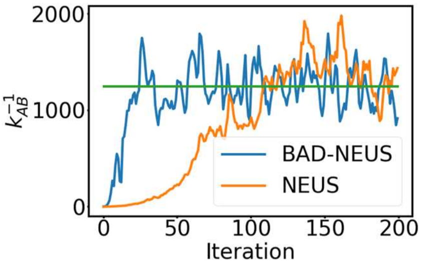

**FIG. 6.** Convergence of the TPT inverse rate estimate for the Müller–Brown system. The green line is the grid-based reference obtained from 45. For these simulations, there are 2000 walkers per stratum, $h = 4$ iterations are retained, and the lag time is $\tau = 10$ time steps.

stratum located below (above) state $B$ $(A)$ is unlikely to be populated. Both NEUS and BAD-NEUS use 2000 walkers per stratum, and we compute statistics using data from the last $h = 3$ iterations. For BAD-NEUS, we use a basis set consisting of ten indicator functions per stratum, again based on $k$-means clustering; we use a lag time of $\tau = 1$.

To compute reference values for these kinetic statistics, we use the grid-based approximation to the generator in Ref. 30, with the same grid parameters. Since the dynamics are microscopically reversible, we obtain the backward committor by solving for the forward committor using the approach in Ref. 30, then setting $q_- = 1 - q_+$. We solve for the reaction rate using

$$k_{AB} = \frac{2(\vec{\pi q_-})^\top (P\vec{\mathbb{1}}_B)}{(\vec{\pi}^\top \vec{q}_-)(\beta\epsilon_x)}, \tag{45}$$

where $P$ is the discretized transition matrix defined on the grid, $\epsilon_x$ is the grid spacing, and arrows indicate vectors of function values on the grid points ($\vec{\pi q_-}$ is a single vector of product values).

Figure 6 shows estimates for the TPT rate. We see that BAD-NEUS converges several fold faster than NEUS. Each BAD-NEUS iteration requires an average of 407 time units of sampling (arising from 2000 walkers in each of 20 strata, with an average time before leaving the stratum of 10.2 time steps). Each iteration is, therefore, significantly less total computational cost than generating a single $A$ to $B$ transition on average (1200 time units, as evidenced by $k_{AB}^{-1}$) and is amenable to parallelization.

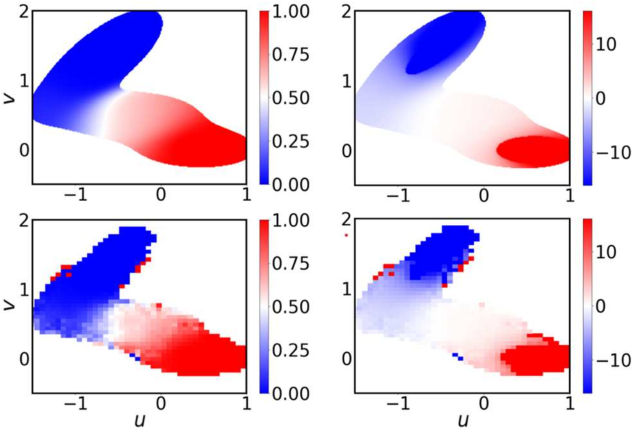

**FIG. 7.** Illustration of the backward committor computed from BAD-NEUS. (top) Reference. (bottom) BAD-NEUS results. (left) Backward committor. (right) $\ln(q_-/(1 - q_-))$, which emphasizes states near $A$ and $B$. These results are obtained from the same simulations as Fig. 6.

Figure 7 shows the backward committor computed at BAD-NEUS iteration 20 using (39) with the variables of numerical integration as the CVs. We represent the results in two ways: the committor itself and its logit function. The former emphasizes the transition region ($q_- \approx 0.5$), while the latter emphasizes values close to states $A$ and $B$ ($q_- \approx 0$ and $q_- \approx 1$). Both show excellent agreement with the reference.

## VI. MOLECULAR TEST SYSTEM

As a more challenging test of the method, we estimate the folding rate of the first 39 amino acids of the N-terminal domain of ribosomal protein L9 (NTL9). The native secondary structure of NTL9 is $\beta\beta\alpha\beta\alpha$ sequentially. In the native state, the $\beta$-strands form an anti-parallel $\beta$-sheet, and the $\alpha$-helices pack on either side of it. NTL9(1-39) lacks the C-terminal helix (Fig. 8). We choose this system because it enables direct comparison with an earlier computational study[37] (described further below), which estimated the folding to be on the millisecond timescale, consistent with experimental studies of (full) NTL9[38,39] and earlier simulations of NTL9(1-39).[40]

All molecular dynamics simulations were performed with OpenMM.[41] We compute CVs including backbone RMS deviations (RMSDs) and fractions of native contacts using MDtraj.[42] To enable direct comparison with Ref. 37, we model NTL9(1-39) using the AMBER FF14SB force field[43] and the Hawkins, Cramer, and Truhlar generalized Born implicit solvent model.[44,45] We use a Langevin thermostat with a time step of 2 fs, a temperature of 300 K, and a friction constant of $\gamma = 80$ ps$^{-1}$, which corresponds to the high-viscosity case considered in Ref. 37.

We run the molecular dynamics for each walker in intervals of 20 ps and compute the CVs to update the index process at the end of each interval. For BAD-NEUS, after an index process changes values [($S(1)$ in (3)], we run the walker $\tau$ additional molecular dynamics steps to ensure that we have a total of $\tau$ steps beyond $S(1)$. While this procedure allows walkers to run beyond the time that they exit their stratum, it does not bias the results. Since our algorithm only requires us to stop and start the molecular dynamics engine while

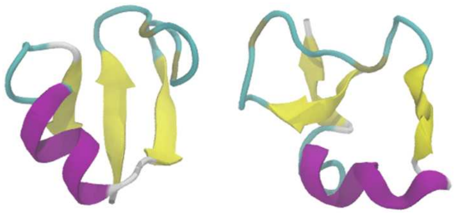

**FIG. 8.** Representative NTL9(1-39) structures drawn from the (left) native and (right) denatured states in the BAD-NEUS simulation. There is residual helical structure in the denatured state in both our simulations and those to which we compare.[37]

**TABLE I.** Hyperparameter choices used for NTL9(1-39) simulations.

| Hyperparameter | Kinetic statistics |
|---|---|
| Stratification CV | $Q$ and $\mathbb{1}_A(T^-_{A\cup B}(t))$ |
| Stopping rule | Stratum exit |
| Walkers per stratum | 4000 |
| Number of strata | 40 |
| Number of iterations retained, $h$ | 4 |
| Type of basis set | $k$-means indicator |
| Number of basis functions per stratum | 10 |
| Lag time, $\tau$ | 10 ps |

running unbiased dynamics, we do not need to modify OpenMM in any way.

The stratification is similar to that for the Müller–Brown system. Namely, we set

$$\psi_k^A(X) = \psi_k^B(X) = \begin{cases} 1, & \text{if } |Q(X) - Q_k| < \varepsilon_k, \\ 0, & \text{otherwise,} \end{cases} \tag{46}$$

where $Q(X)$ is the fraction of native contacts using the definition of Ref. 46. We determine the native contacts from the crystal structure in the Protein Data Bank (PDB), PDB ID 2HBB.[47] We space 20 $Q_k$ values uniformly in the interval (0.35, 0.85) and set $\varepsilon_k = 0.6(Q_{k+1} - Q_k)$, so that the regions overlap. Therefore, there are 40 total strata. We define the folded state as full backbone RMSD <0.28 nm and $Q > 0.85$; the denatured state has full backbone RMSD >0.80 nm and $Q < 0.35$.

To initialize the simulation, we minimize the energy starting from the PDB structure, draw velocities from a Maxwell–Boltzmann distribution for 380 K, and simulate for 80 ns at that temperature; this denatures the protein (Fig. 8). We sample 4000 frames in each of the 40 strata from this trajectory and weight them uniformly; this set forms our starting walkers, $\{X_0^i, J_0^i, 1/N\}_{i=1}^N$, where $N = 40$. We use a lag time of 10 ps, and we retain a history of $h = 4$ iterations. The initialization pipeline, integrator settings, and stratification choice are the same between NEUS and BAD-NEUS. The hyperparameter choices are summarized in Table I.

To construct the basis set for BAD-NEUS, we define the following seven CVs: (1) the fraction of native contacts, (2) the full backbone RMSD, (3–5) the backbone RMSD for each of the three $\beta$-strands taken individually (residues 1–4, 17–20, and 36–38), (6) the backbone RMSD for the $\alpha$-helix (residues 22–29), and (7) the backbone RMSD for the three $\beta$-strands together. Our basis set consists of ten indicator functions on each stratum constructed by $k$-means clustering on the seven-dimensional CV space. At each iteration (i.e., one pass through the outer loop in Algorithm 4), we determine the basis functions for stratum $j$ by taking the associated cluster centers from the prior iteration and refining them with ten iterations of Lloyd's algorithm[48] using all the samples with index process $J_t^i = j$ from the last $h$ iterations.

Here, we focus on the convergence of the folding rate; a detailed analysis of the NTL9 folding mechanism (with and without the C-terminal helix) based on potentials of mean force and committors will be presented elsewhere (see also Ref. 49). As mentioned earlier, we compare our results to those of Ref. 37, which estimates rates using haMSMs applied to data from WE (haMSM-WE). As discussed in Sec. I, BAD-NEUS goes beyond haMSM-WE by using the basis set (Markov model) to accelerate the convergence of the sampling rather than just the rate estimates.

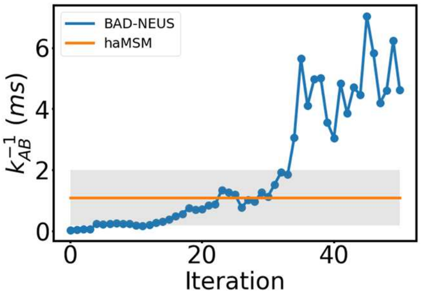

**FIG. 9.** Comparison of inverse rates for NTL9(1-39) folding from BAD-NEUS and haMSM-WE. The haMSM-WE results are from Ref. 37. The shaded area shows the Bayesian 95% credible interval reported in that publication.

Figure 9 shows the inverse rate constant obtained from BAD-NEUS and haMSM-WE.[37] The average inverse rate over the final 15 iterations of BAD NEUS is 4.74 ms, with a standard deviation of 0.99 ms, while the authors of Ref. 37 used a Bayesian bootstrapping approach to estimate a confidence interval of 0.17–1.9 ms, which they state is likely an underestimate of the true 95% confidence interval owing to the limited number of independent samples used in the analysis. Given that we have only a single BAD NEUS run, the standard deviation of the final 15 (highly correlated) inverse rate

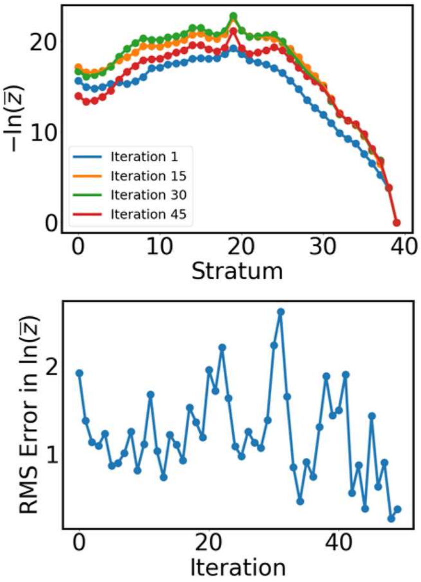

**FIG. 10.** (Top) BAD NEUS bin weights for the NTL9(1-39) system. Strata with indices less than 20 correspond to the folding direction (trajectories last in the denatured state), and strata with indices greater than or equal to 20 correspond to the unfolding direction (trajectories last in the native state). (Bottom) Deviation of weights ($\ln(\bar{z})$) from those in the last iteration.

estimates is also likely a significant underestimate of our statistical error. We thus view the results in Fig. 9 as in agreement.

We examine the weights in more detail in Fig. 10. The increase in the inverse rate around iteration 35 in Fig. 9 [and concomitant decrease below 5 in Fig. 10 (bottom)] corresponds to a shift in the weights of strata with indices close to 25 [compare iterations 30 and 45 in Fig. 10 (top)]. These strata contain walkers that are nearly denatured from trajectories that last visited the native state, and the shift in weights in that region stabilizes the denatured state relative to the native state.

We run each walker until its index process changes and then an additional 10 ps, so the trajectory length and time per iteration are random variables. For the NTL9 simulations, the average trajectory length was 50 ps, and the average time per iteration was 4 $\mu$s. The results in Fig. 9 thus required about 204 $\mu$s in total, comparable to the 252 $\mu$s for the haMSM-WE calculation. The time to compute the rate constant by BAD-NEUS and haMSM-WE is much less than the millisecond timescale of a single folding event, which we also expect to be the time to converge WE without any acceleration strategy. BAD-NEUS actually provides some speedup relative to haMSM-WE in the sense that it provides potentials of mean force and committors in its present form, whereas a calculation in the unfolding direction (performed either separately or simultaneously, as here) would be required to obtain these statistics from haMSM-WE. By the same token, we likely could reduce the computation time for BAD-NEUS by "teleporting" walkers from $B$ to $A$ and computing only the forward rate.[50] However, we expect most investigators would seek the mechanistic information provided by the potentials of mean force and committor (which can be used to also obtain reactive currents[24]) if they are investing the computational resources to compute the rate, and free energies computed from ratios of forward and backward rates can serve as checks on the rates.[51]

## VII. CONCLUSIONS

Trajectory stratification methods enable enhanced sampling for estimating both thermodynamic (equilibrium) and kinetic (nonequilibrium) statistics. Here, we introduced a new trajectory stratification method, BAD-NEUS, which converges faster than existing ones. We show that our method is a natural generalization of NEUS and EM, which in turn can be formulated as extensions of WE. In the process, we show that WE as originally formulated[1] converges no faster than unbiased dynamics. The key modification introduced in NEUS and elaborated here is the insertion of an approximation to the steady-state distribution before the resampling step. Importantly, we design this approximation algorithm so that it preserves the steady-state distribution of the dynamics and, therefore, does not introduce systematic errors into the algorithm in the large-data limit.

Our approach defines precise ingredients needed for iterative restart strategies to converge to the correct fixed point (i.e., a fixed point consistent with unbiased dynamics). In this way, it both restricts and generalizes iterative restart strategies previously proposed,[8,13] and we provide numerical demonstrations that such strategies can accelerate the convergence of both equilibrium and kinetic statistics. In the future, it would be interesting to compare BAD-NEUS with the WE milestoning approach, which instead embeds WE simulations within coarse regions (i.e., between milestones).[52,53] While this approach also combines fine and coarse information, it is fundamentally different from BAD-NEUS and haMSM-WE in that it introduces an additional sampling rather than analysis algorithm at the fine level. Indeed, adding an element of stratification to the WE simulations within WE milestoning may result in further speedups. The similarities and differences between various approaches highlight the importance of a theoretical framework for relating algorithms like the one that we introduce here; we believe it can serve as the basis for a thorough, systematic cataloging of methods for computing dynamical statistics, analogous to recent attempts for enhanced sampling methods for equilibrium statistics.[35]

In the present study, we use a simple basis expansion to model the steady-state distribution, but our strategy is general, and alternatives are also possible. One could use neural networks to learn the steady-state distribution and/or the basis functions.[29,30] One could also incorporate memory.[54] These alternatives, which could be used separately or together, alleviate the need to identify basis sets and/or CVs that describe the dynamics well and, in the case of memory, may reduce the sampling required. We thus expect our method to be a significant step toward accurate estimates of rates for protein folding and similarly complex molecular conformational changes.

## ACKNOWLEDGMENTS

We thank Daniel Zuckerman for discussions concerning the manuscript. This work was supported by National Institutes of Health Award No. R35 GM136381 and National Science Foundation Award No. DMS-2054306. This work was completed in part with resources provided by the University of Chicago Research Computing Center. "Beagle-3: A Shared GPU Cluster for Biomolecular Sciences" is supported by the National Institutes of Health (NIH) under the High-End Instrumentation (HEI) grant program Award No. 1S10OD028655-0.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

J.S. and C.L. authors contributed equally to this work.

**John Strahan**: Conceptualization (lead); Investigation (lead); Writing – original draft (equal); Writing – review & editing (equal). **Chatipat Lorpaiboon**: Conceptualization (lead); Investigation (equal). **Jonathan Weare**: Conceptualization (equal); Formal analysis (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Aaron R. Dinner**: Conceptualization (equal); Funding acquisition (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

## DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## APPENDIX: DERIVATION OF (16)

Here, we derive (16) by modifying the proof from Ref. 55. Let $Z_t$ be a Markov process, $\mathcal{T}_t$ be the operator defined in (41), and $S_n = \sum_{t=0}^{n-1} g(Z_t)$ be the partial sum of a function $g$. Let $\omega(z)$ satisfy the linear problem

$$(\mathcal{T}_1 - \mathcal{I})\omega(z) = -(g(z) - \mathbb{E}_\pi[g]), \tag{A1}$$

subject to the constraint $\mathbb{E}_\pi[\omega] = 0$.

Consider the process,

$$U_n = S_n - \mathbb{E}_\pi[g]n + \omega(Z_n). \tag{A2}$$

Then $U_t$ is a martingale with respect to $Z_0, \ldots, Z_t$. To see this, we compute

$$\begin{aligned}\mathbb{E}[U_{t+1} - U_t|Z_0, \ldots Z_t] &= \mathbb{E}[S_{t+1} - \mathbb{E}_\pi[g](t+1) + \omega(Z_{t+1}) \\ &\quad - (S_t - \mathbb{E}_\pi[g]t + \omega(Z_t))|Z_0, \ldots Z_t] \\ &= \mathbb{E}[g(Z_t) - \mathbb{E}_\pi[g] + \omega(Z_{t+1}) - \omega(Z_t)|Z_0, \ldots Z_t] \\ &= g(Z_t) - \mathbb{E}_\pi[g] + \mathbb{E}[\omega(Z_{t+1}) - \omega(Z_t)|Z_0, \ldots Z_t] \\ &= g(Z_t) - \mathbb{E}_\pi[g] + (\mathcal{T}_1 - \mathcal{I})\omega(Z_t) \\ &= g(Z_t) - \mathbb{E}_\pi[g] - g(Z_t) + \mathbb{E}_\pi[g] \\ &= 0.\end{aligned} \tag{A3}$$

We refer the reader to Ref. 55 for a more technical discussion and conditions under which the relevant expectations are bounded such that we may apply the optional stopping theorem. Then, for any stopping time $T$ with $\mathbb{E}[T] < \infty$,

$$0 = \mathbb{E}[U_T - U_0] = \mathbb{E}[S_T] - \mathbb{E}_\pi[g]\mathbb{E}[T] + \mathbb{E}[\omega(Z_T)] - \mathbb{E}[\omega(Z_0)], \tag{A4}$$

which yields the result,

$$\frac{\mathbb{E}\left[\sum_{t=0}^{T-1} g(Z_t)\right]}{\mathbb{E}[T]} = \mathbb{E}_\pi[g] + \frac{\mathbb{E}[\omega(Z_0)] - \mathbb{E}[\omega(Z_T)]}{\mathbb{E}[T]}. \tag{A5}$$

Now take $Z_t = (X_t, J_t, \ldots, X_{t+\tau}, J_{t+\tau})$ and take the stopping time to be $S(1)$. In the case where $X_0, J_0$ is distributed according to the steady state flux distribution $\overline{\pi}(dx, k)$, (9) implies that $(X_T, J_T) \sim \overline{\pi}(dx, k)$, and hence the distribution of $Z_0$ and $Z_T$ are the same, and so the residual term in (A5) is zero. In this case, we note that the distribution of $(X_0, J_0)$ is given by

$$\begin{aligned}\pi(dx) &= \mathbb{P}[X_0 \in dx], &\text{(A6)}\\ &= \sum_k \mathbb{P}[X_0 \in dx \mid J_0 = k]\mathbb{P}[J_0 = k], &\text{(A7)}\\ &= \sum_k \overline{z}^k \overline{\pi}(dx|k). &\text{(A8)}\end{aligned}$$

The joint distribution for $(X, J)$ is $\overline{\pi}(dx, k) = \overline{z}^k \overline{\pi}(dx|k)$. Therefore,

$$\mathbb{E}\left[\sum_{t=0}^{S(1)-1} g(X_t, J_t, \ldots, X_{t+\tau}, J_{t+\tau})\right] = \sum_k \overline{z}^k \int \overline{\pi}(dx|k)\mathbb{E}_{X_0=x, J_0=k}\left[\sum_{t=0}^{S(1)-1} g(X_t, J_t, \ldots, X_{t+\tau}, J_{t+\tau})\right], \tag{A9}$$

and

$$\mathbb{E}[S(1)] = \sum_k \overline{z}^k \int \overline{\pi}(dx|k)\mathbb{E}_{X_0=x, J_0=k}[S(1)]. \tag{A10}$$

Substituting these into (A5) and remembering that the residual term is zero yields (16).

## REFERENCES

[1] G. A. Huber and S. Kim, "Weighted-ensemble Brownian dynamics simulations for protein association reactions," Biophys. J. **70**(1), 97–110 (1996).

[2] R. J. Allen, C. Valeriani, and P. Rein ten Wolde, "Forward flux sampling for rare event simulations," J. Phys.: Condens. Matter **21**(46), 463102 (2009).

[3] A. Warmflash, P. Bhimalapuram, and A. R. Dinner, "Umbrella sampling for nonequilibrium processes," J. Chem. Phys. **127**(15), 154112 (2007).

[4] A. Dickson and A. R. Dinner, "Enhanced sampling of nonequilibrium steady states," Annu. Rev. Phys. Chem. **61**, 441–459 (2010).

[5] N. Guttenberg, A. R. Dinner, and J. Weare, "Steered transition path sampling," J. Chem. Phys. **136**(23), 234103 (2012).

[6] J. M. Bello-Rivas and R. Elber, "Exact milestoning," J. Chem. Phys. **142**(9), 094102 (2015).

[7] D. M. Zuckerman and L. T. Chong, "Weighted ensemble simulation: Review of methodology, applications, and software," Annu. Rev. Biophys. **46**, 43–57 (2017).

[8] J. D. Russo, S. Zhang, J. M. G. Leung, A. T. Bogetti, J. P. Thompson, A. J. DeGrave, P. A. Torrillo, A. J. Pratt, K. F. Wong, J. Xia, J. Copperman, J. L. Adelman, M. C. Zwier, D. N. LeBard, D. M. Zuckerman, and L. T. Chong, "WESTPA 2.0: High-performance upgrades for weighted ensemble simulations and analysis of longer-timescale applications," J. Chem. Theory Comput. **18**(2), 638–649 (2022).

[9] S. D. Lotz and A. Dickson, "Wepy: A flexible software framework for simulating rare events with weighted ensemble resampling," ACS Omega **5**(49), 31608–31623 (2020).

[10] D. Aristoff, J. Copperman, G. Simpson, R. J. Webber, and D. M. Zuckerman, "Weighted ensemble: Recent mathematical developments," J. Chem. Phys. **158**(1), 014108 (2023).

[11] S.-H. Ahn, A. A. Ojha, R. E. Amaro, and J. A. McCammon, "Gaussian-accelerated molecular dynamics with the weighted ensemble method: A hybrid method improves thermodynamic and kinetic sampling," J. Chem. Theory Comput. **17**(12), 7938–7951 (2021).

[12] A. A. Ojha, S. Thakur, S.-H. Ahn, and R. E. Amaro, "DeepWEST: Deep learning of kinetic models with the weighted ensemble simulation toolkit for enhanced sampling," J. Chem. Theory Comput. **19**(4), 1342–1359 (2023).

[13] J. Copperman and D. M. Zuckerman, "Accelerated estimation of long-timescale kinetics from weighted ensemble simulation via non-Markovian 'microbin' analysis," J. Chem. Theory Comput. **16**(11), 6763–6775 (2020).

[14] A. J. DeGrave, A. T. Bogetti, and L. T. Chong, "The RED scheme: Rate-constant estimation from pre-steady state weighted ensemble simulations," J. Chem. Phys. **154**(11), 114111 (2021).

[15] E. Suárez, S. Lettieri, M. C. Zwier, S. R. Subramanian, L. T. Chong, and D. M. Zuckerman, "Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories," Biophys. J. **106**(2), 406a (2014).

[16] A. Dickson, A. Warmflash, and A. R. Dinner, "Nonequilibrium umbrella sampling in spaces of many order parameters," J. Chem. Phys. **130**(7), 074104 (2009).

[17] A. Dickson, A. Warmflash, and A. R. Dinner, "Separating forward and backward pathways in nonequilibrium umbrella sampling," J. Chem. Phys. **131**(15), 154104 (2009).

[18] A. Dickson, M. Maienschein-Cline, A. Tovo-Dwyer, J. R. Hammond, and A. R. Dinner, "Flow-dependent unfolding and refolding of an RNA by nonequilibrium umbrella sampling," J. Chem. Theory Comput. **7**(9), 2710–2720 (2011).

[19] A. R. Dinner, J. C. Mattingly, J. O. B. Tempkin, B. V. Koten, and J. Weare, "Trajectory stratification of stochastic dynamics," SIAM Rev. **60**(4), 909–938 (2018).

[20] D. Bhatt, B. W. Zhang, and D. M. Zuckerman, "Steady-state simulations using weighted ensemble path sampling," J. Chem. Phys. **133**(1), 014110 (2010).

[21] G. Earle and J. C. Mattingly, "Convergence of stratified MCMC sampling of non-reversible dynamics," in *Stochastics and Partial Differential Equations: Analysis and Computations* (Springer, 2024), pp. 1–41.

[22] G. Earle and B. Van Koten, "Aggregation methods for computing steady states in statistical physics," Multiscale Model. Simul. **21**(3), 1170–1209 (2023).

[23] E. H. Thiede, D. Giannakis, A. R. Dinner, and J. Weare, "Galerkin approximation of dynamical quantities using trajectory data," J. Chem. Phys. **150**(24), 244111 (2019).

[24] J. Strahan, A. Antoszewski, C. Lorpaiboon, B. P. Vani, J. Weare, and A. R. Dinner, "Long-time-scale predictions from short-trajectory data: A benchmark analysis of the trp-cage miniprotein," J. Chem. Theory Comput. **17**(5), 2948–2963 (2021).

[25] J. Finkel, R. J. Webber, E. P. Gerber, D. S. Abbot, and J. Weare, "Learning forecasts of rare stratospheric transitions from short simulations," Mon. Weather Rev. **149**(11), 3647–3669 (2021).

[26] B. P. Vani, J. Weare, and A. R. Dinner, "Computing transition path theory quantities with trajectory stratification," J. Chem. Phys. **157**(3), 034106 (2022).

[27] A. Dickson and C. L. Brooks III, "WExplore: Hierarchical exploration of high-dimensional spaces using the weighted ensemble algorithm," J. Phys. Chem. B **118**(13), 3532–3542 (2014).

[28] N. Donyapour, N. M. Roussey, and A. Dickson, "REVO: Resampling of ensembles by variation optimization," J. Chem. Phys. **150**(24), 244112 (2019).

[29] J. Wen, B. Dai, L. Li, and D. Schuurmans, "Batch stationary distribution estimation," in Proceedings of the 37th International Conference on Machine Learning, PMLR **119**, 10203–10213 (2020), https://proceedings.mlr.press/v119/wen20a.html.

[30] J. Strahan, S. C. Guo, C. Lorpaiboon, A. R. Dinner, and J. Weare, "Inexact iterative numerical linear algebra for neural network-based spectral estimation and rare-event prediction," J. Chem. Phys. **159**, 014110 (2023).

[31] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis* (IEEE, 2005), pp. 64–69.

[32] J. Strahan, J. Finkel, A. R. Dinner, and J. Weare, "Predicting rare events using neural networks and short-trajectory data," J. Comput. Phys. **488**, 112152 (2023).

[33] K. Müller and L. D. Brown, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure," Theor. Chim. Acta **53**(1), 75–93 (1979).

[34] B. Leimkuhler and C. Matthews, "Rational construction of stochastic numerical methods for molecular sampling," Appl. Math. Res. Express **2013**(1), 34–56.

[35] J. Hénin, T. Lelièvre, M. Shirts, O. Valsson, and L. Delemotte, "Enhanced sampling methods for molecular dynamics simulations [Article v1.0]," Living J. Comput. Mol. Sci. **4**(1), 1583 (2022).

[36] E. Vanden-Eijnden and M. Venturoli, "Exact rate calculations by trajectory parallelization and tilting," J. Chem. Phys. **131**(4), 044120 (2009).

[37] U. Adhikari, B. Mostofian, J. Copperman, S. R. Subramanian, A. A. Petersen, and D. M. Zuckerman, "Computational estimation of microsecond to second atomistic folding times," J. Am. Chem. Soc. **141**(16), 6519–6526 (2019).

[38] D. L. Luisi, B. Kuhlman, K. Sideras, P. A. Evans, and D. P. Raleigh, "Effects of varying the local propensity to form secondary structure on the stability and folding kinetics of a rapid folding mixed $\alpha/\beta$ protein: Characterization of a truncation mutant of the N-terminal domain of the ribosomal protein L9," J. Mol. Biol. **289**(1), 167–174 (1999).

[39] S. Sato, J.-H. Cho, I. Peran, R. G. Soydaner-Azeloglu, and D. P. Raleigh, "The N-terminal domain of ribosomal protein L9 folds via a diffuse and delocalized transition state," Biophys. J. **112**(9), 1797–1806 (2017).

[40] V. A. Voelz, G. R. Bowman, K. Beauchamp, and V. S. Pande, "Molecular simulation of *ab initio* protein folding for a millisecond folder NTL9(1–39)," J. Am. Chem. Soc. **132**(5), 1526–1528 (2010).

[41] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," PLoS Comput. Biol. **13**(7), e1005659 (2017).

[42] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. Pande, "MDTraj: A modern open library for the analysis of molecular dynamics trajectories," Biophys. J. **109**(8), 1528–1532 (2015).

[43] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, "ff14SB: Improving the accuracy of protein side chain and backbone parameters from ff99SB," J. Chem. Theory Comput. **11**(8), 3696–3713 (2015).

[44] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, "Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium," J. Phys. Chem. **100**(51), 19824–19839 (1996).

[45] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, "Pairwise solute descreening of solute charges from a dielectric medium," Chem. Phys. Lett. **246**(1–2), 122–129 (1995).

[46] R. B. Best, G. Hummer, and W. A. Eaton, "Native contacts determine protein folding mechanisms in atomistic simulations," Proc. Natl. Acad. Sci. U. S. A. **110**(44), 17874–17879 (2013).

[47] J.-H. Cho, W. Meng, S. Sato, E. Y. Kim, H. Schindelin, and D. P. Raleigh, "Energetically significant networks of coupled interactions within an unfolded protein," Proc. Natl. Acad. Sci. U. S. A. **111**(33), 12079–12084 (2014).

[48] S. Lloyd, "Least squares quantization in PCM," IEEE Trans. Inf. Theory **28**(2), 129–137 (1982).

[49] J. C. Strahan, "Short trajectory methods for rare event analysis and sampling," Ph.D. thesis, The University of Chicago, 2024.

[50] M. Baudel, A. Guyader, and T. Lelièvre, "On the Hill relation and the mean reaction time for metastable processes," Stochastic Processes Appl. **155**, 393–436 (2023).

[51] R. Hall, T. Dixon, and A. Dickson, "On calculating free energy differences using ensembles of transition paths," Front. Mol. Biosci. **7**, 106 (2020).

[52] D. Ray and I. Andricioaei, "Weighted ensemble milestoning (WEM): A combined approach for rare event simulations," J. Chem. Phys. **152**(23), 234114 (2020).

[53] D. Ray, S. E. Stone, and I. Andricioaei, "Markovian weighted ensemble milestoning (M-WEM): Long-time kinetics from short trajectories," J. Chem. Theory Comput. **18**(1), 79–95 (2022).

[54] C. Lorpaiboon, S. C. Guo, J. Strahan, J. Weare, and A. R. Dinner, "Accurate estimates of dynamical statistics using memory," J. Chem. Phys. **160**(8), 084108 (2024).

[55] G. V. Moustakides, "Extension of Wald's first lemma to Markov processes," J. Appl. Probab. **36**(1), 48–59 (1999).