

# Estimating reciprocal partition functions to enable design space sampling

Cite as: J. Chem. Phys. 153, 204102 (2020); <https://doi.org/10.1063/5.0025358>

Submitted: 14 August 2020 . Accepted: 30 October 2020 . Published Online: 23 November 2020

 Alex Albaugh, and  Todd R. Gingrich



View Online



Export Citation



CrossMark



## Your Qubits. Measured.

Meet the next generation of quantum analyzers

- Readout for up to 64 qubits
- Operation at up to 8.5 GHz, mixer-calibration-free
- Signal optimization with minimal latency

Find out more



# Estimating reciprocal partition functions to enable design space sampling

Cite as: J. Chem. Phys. 153, 204102 (2020); doi: 10.1063/5.0025358

Submitted: 14 August 2020 • Accepted: 30 October 2020 •

Published Online: 23 November 2020



View Online



Export Citation



CrossMark

Alex Albaugh  and Todd R. Gingrich<sup>a)</sup> 

## AFFILIATIONS

Department of Chemistry, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208, USA

**Note:** This paper is part of the JCP Emerging Investigators Special Collection.

<sup>a)</sup> Author to whom correspondence should be addressed: [todd.gingrich@northwestern.edu](mailto:todd.gingrich@northwestern.edu)

## ABSTRACT

Reaction rates are a complicated function of molecular interactions, which can be selected from vast chemical design spaces. Seeking the design that optimizes a rate is a particularly challenging problem since the rate calculation for any one design is itself a difficult computation. Toward this end, we demonstrate a strategy based on transition path sampling to generate an ensemble of designs and reactive trajectories with a preference for fast reaction rates. Each step of the Monte Carlo procedure requires a measure of how a design constrains molecular configurations, expressed via the reciprocal of the partition function for the design. Although the reciprocal of the partition function would be prohibitively expensive to compute, we apply Booth's method for generating unbiased estimates of a reciprocal of an integral to sample designs without bias. A generalization with multiple trajectories introduces a stronger preference for fast rates, pushing the sampled designs closer to the optimal design. We illustrate the methodology on two toy models of increasing complexity: escape of a single particle from a Lennard-Jones potential well of tunable depth and escape from a metastable tetrahedral cluster with tunable pair potentials.

Published under license by AIP Publishing. <https://doi.org/10.1063/5.0025358>

## I. INTRODUCTION

One of the most challenging and important aims of theoretical and computational chemistry is the calculation of rates. The speed of chemical events can vary over many orders of magnitude, ranging from electron transfer on a femtosecond timescale to material aging over millennia. Direct simulation of quantum or classical dynamics can provide access to these fastest timescales, but numerically computing rates for activated processes is notoriously difficult due to the rare event problem.<sup>1–4</sup> To combat this problem, several related methodologies have been developed based on the connection between time correlation functions and rate constants.<sup>3,5–12</sup> Crucially, those correlation functions can often be calculated from dynamical trajectories of modest length using methods such as transition path sampling,<sup>3,13–22</sup> transition interface sampling,<sup>23–25</sup> and forward flux sampling.<sup>26–31</sup>

In principle, one should thus be able to attack chemical design problems—problems like determining what side chains of a peptide most effectively amplify a rate of catalysis. In practice, it becomes very expensive to perform a converged rate calculation for every candidate design. One approach to circumvent this expense is to

sample the design space with a random walker, statistically biased to spend most of its time visiting designs with fast rates. Let  $\lambda$  denote all parameters one seeks to design; these could be particle charges, amino acid identities, Lennard-Jones parameters, bond strengths, equilibrium bond lengths, etc. Each design has some rate constant  $k(\lambda)$ , and one might hope to sample possible designs from the probability distribution with probability density  $P(\lambda) \propto k(\lambda)$ , thereby giving extra statistical weight to those designs with faster rates. A simple, straightforward way to sample designs is to carry out a Markov Chain Monte Carlo (MCMC) simulation, consisting of three iterated steps: (1) attempt to transition from design  $\lambda$  to a new design  $\lambda'$  with probability  $P_{\text{gen}}(\lambda \rightarrow \lambda')$ ; (2) compute a converged rate calculation of both  $k(\lambda)$  and  $k(\lambda')$ ; and (3) accept the new design with probability

$$P_{\text{acc}}[\lambda \rightarrow \lambda'] = \min \left[ 1, \frac{P_{\text{gen}}(\lambda' \rightarrow \lambda)k(\lambda')}{P_{\text{gen}}(\lambda \rightarrow \lambda')k(\lambda)} \right]. \quad (1)$$

While this procedure would steer the sampled designs toward those with faster rates, it requires high-quality converged rate calculations for every proposed  $\lambda$ .

To radically reduce the computational expense, one might instead hope to carry out the MCMC dynamics with noisy estimates for  $k(\lambda)$ , akin to Ceperley and Dewing's penalty method for random walks with noisy energies.<sup>32</sup> The essential idea is to execute a random walk in the higher-dimensional space of designs *and* reactive trajectories, those that transition from the reactant to product in a fixed observation time  $t_{\text{obs}}$ . Every step of the Monte Carlo procedure outlined in Eq. (1) requires a converged rate calculation to decide whether to accept a newly proposed design, but each step of the joint-space random walker uses a noisy estimate of that rate. This noisy estimate characterizes how probable it is to generate a reactive trajectory given the design  $\lambda$ , assuming the trajectory was initialized in an equilibrium reactant configuration.

The requirement that reactive trajectories be initialized in an equilibrium ensemble presents a significant technical problem, the resolution of which is the focus of this manuscript. The challenge is that computing the acceptance probability for a Monte Carlo step requires the Boltzmann probability of the initial condition, which depends on a design-dependent canonical partition function. Were the design held fixed, a ratio of identical partition functions would cancel in the Monte Carlo acceptance formulas. Without that cancellation, the MCMC procedure requires unbiased estimates of the reciprocal of the partition function. In this manuscript, we show how those estimates can be obtained using Booth's method for generating unbiased estimates of integrals.<sup>33</sup> By applying that strategy to the rate design problem, we can sample  $P(\lambda) \propto k(\lambda)$  without converged rate estimates. A stronger preference for designs with large rates is applied by introducing  $L$  independent trajectories to sample  $P(\lambda) \propto k(\lambda)^L$ .

This manuscript is split into two parts. We review and develop the theoretical tools in Sec. II. We then illustrate the methodology on two model systems in Sec. III. The first simple system—sampling different strengths of attraction between two Lennard-Jones particles in proportion to their unbinding rate—serves to validate the mathematics by comparing the results from sampling with explicit numerical calculations. The second system—sampling a three-dimensional design space of interaction parameters between a particle metastably trapped in a cage—more clearly illustrates the potential benefits when studying more complex, higher-dimensional problems.

## II. THEORY

### A. Reaction rates and trajectory-space sampling

Consider the phase space of a classical system,  $\mathbf{x} = \{\mathbf{r}, \mathbf{p}\}$ , which consists of a set of positions of all  $N$  particles,  $\mathbf{r} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ , and their momenta,  $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ . The total energy of this system is given by the Hamiltonian  $H$ , the sum of the kinetic energy  $K$  and potential energy  $U$ :  $H(\mathbf{x}; \lambda) = U(\mathbf{r}; \lambda) + K(\mathbf{p}; \lambda)$ . The energy depends not only on  $\mathbf{r}$  but also on some parameters  $\lambda$ , which could include particle charges, Lennard-Jones parameters, bond strengths, etc. In this work, we imagine these “design parameters” to be time-independent and controllable. A system evolving under  $H(\mathbf{x}; \lambda)$  traces out trajectories in phase space that we denote  $\vec{\mathbf{x}}$ . We will focus on discrete time-evolution generated by numerical integration such that the trajectory is a sequence of  $M + 1$  points in phase space separated by increments of time  $\Delta t$ :  $\vec{\mathbf{x}} = \{\mathbf{x}(0), \mathbf{x}(\Delta t), \dots, \mathbf{x}(t_{\text{obs}})\}$ , with observation time  $t_{\text{obs}} = M\Delta t$ .

Chemical systems tend to be high-dimensional with potential energy surfaces that possess multiple metastable basins. Trajectories occupy a metastable region of phase space for relatively long periods of time before making rare transitions to another metastable region. In the simplest scenario, there are two principal, non-overlapping basins,  $A$  and  $B$ , which correspond to reactants and products, respectively. Provided that the transitions are rare, there exists a first-order rate constant  $k_{AB}(\lambda)$  that depends on the particular design. If each  $A \rightarrow B$  transition is independent of previous transitions, e.g., if memory is lost, then the process is Poissonian with the time between reactions,  $\tau$ , coming from the distribution

$$P(\tau|\lambda) = k_{AB}(\lambda)e^{-k_{AB}(\lambda)\tau}. \quad (2)$$

The probability that a trajectory, starting in  $A$ , will exhibit at least one reaction in time  $t_{\text{obs}}$  is thus given by

$$\int_0^{t_{\text{obs}}} d\tau k_{AB}(\lambda)e^{-k_{AB}(\lambda)\tau} = 1 - e^{-k_{AB}(\lambda)t_{\text{obs}}} \approx k_{AB}(\lambda)t_{\text{obs}}. \quad (3)$$

The final approximation is justified when  $t_{\text{obs}} \ll 1/k_{AB}(\lambda)$ , in which case trajectories only have time for either zero or one reaction event.

Following Ref. 3, the probability of a reaction can also be computed as

$$P_{\text{reaction}}(\lambda, t_{\text{obs}}) = \frac{\int \mathcal{D}\vec{\mathbf{x}} h_A(\mathbf{x}(0))h_B(\mathbf{x}(t_{\text{obs}}))e^{-\beta H(\mathbf{x}(0);\lambda)} P(\vec{\mathbf{x}}|\lambda, \mathbf{x}(0))}{Z_A(\lambda)}, \quad (4)$$

where  $\beta = (k_B T)^{-1}$  is the inverse temperature,  $k_B$  is Boltzmann's constant, and  $h_A$  and  $h_B$  are indicator functions that evaluates to zero or one so as to constrain trajectories to end as products. Here, the probability of trajectory  $\vec{\mathbf{x}}$  has been decomposed into a product of the equilibrium Boltzmann probability for the initial reactant configuration  $\mathbf{x}(0)$  times the (normalized) probability of subsequent dynamics, given that initialization,  $P(\vec{\mathbf{x}}|\lambda, \mathbf{x}(0))$ . The canonical partition function of the reactant state,

$$Z_A(\lambda) = \int d\mathbf{x}(0) h_A(\mathbf{x}(0))e^{-\beta H(\mathbf{x}(0);\lambda)}, \quad (5)$$

measures how  $\lambda$  impacts the Boltzmann probability for being initialized in a reactant configuration.

Provided  $t_{\text{obs}} \ll 1/k_{AB}(\lambda)$ , the Poisson description of Eq. (3) can be equated with the trajectory-space average of Eq. (4). This equality expresses the rate constant in terms of a trajectory-space average,

$$k_{AB}(\lambda) = \frac{1}{t_{\text{obs}}} \int \mathcal{D}\vec{\mathbf{x}} \rho(\vec{\mathbf{x}}, \lambda), \quad (6)$$

where

$$\rho(\vec{\mathbf{x}}, \lambda) = h_A(\mathbf{x}(0))h_B(\mathbf{x}(t_{\text{obs}})) \frac{e^{-\beta H(\mathbf{x}(0);\lambda)}}{Z_A(\lambda)} P(\vec{\mathbf{x}}|\lambda, \mathbf{x}(0)). \quad (7)$$

Upon normalizing,  $P(\vec{\mathbf{x}}, \lambda) = \rho(\vec{\mathbf{x}}, \lambda)/\mathcal{N}$  has the interpretation of a joint distribution over trajectories and designs, with normalization

$$\mathcal{N} = \int d\lambda \int \mathcal{D}\vec{\mathbf{x}} \rho(\vec{\mathbf{x}}, \lambda). \quad (8)$$

If one samples trajectories and designs from that distribution, the marginal distribution over designs would therefore be

$$P(\lambda) = \int \mathcal{D}\vec{x} P(\vec{x}, \lambda) = \frac{t_{\text{obs}}}{\mathcal{N}} k_{AB}(\lambda). \quad (9)$$

The ratio of rates for designs  $\lambda$  and  $\lambda'$  is consequently the relative likelihood of observing  $\lambda$  and  $\lambda'$  in the sampling of  $P(\vec{x}, \lambda)$ ,

$$\frac{P(\lambda)}{P(\lambda')} = \frac{k_{AB}(\lambda)}{k_{AB}(\lambda')}. \quad (10)$$

The goal of sampling  $P(\lambda) \propto k_{AB}(\lambda)$  has been reduced to the problem of sampling the joint distribution  $P(\vec{x}, \lambda)$ , provided  $t_{\text{obs}} \ll 1/k_{AB}$ . Equation (10) would break down if  $t_{\text{obs}}$  were too large, so one may imagine using arbitrarily small  $t_{\text{obs}}$ . That choice results in a different issue. Rare  $A \rightarrow B$  transitions occupy  $A$  for a comparatively long time before carrying out a rapid passage over the barrier. By pushing to smaller  $t_{\text{obs}}$ , one can excise some of that waiting time without impacting the mechanism of the barrier crossing, but there is a minimum amount of time,  $t_{\text{cross}}$ , needed to cross. Using an observation time that is less than this crossing time also causes Eq. (10) to break down. We therefore require that a suitable  $t_{\text{obs}}$  is chosen such that both timescale restrictions are met ( $t_{\text{cross}} < t_{\text{obs}} \ll 1/k_{AB}$ ) for all sampled designs. The necessary timescale separation is illustrated explicitly for our Lennard-Jones unbinding problem in Fig. 1.

## B. The reciprocal partition function problem

The strategy of Sec. II A allowed us to convert the problem of sampling  $P(\lambda) \propto k_{AB}(\lambda)$  into the higher-dimensional joint problem  $P(\vec{x}, \lambda) \propto \rho(\vec{x}, \lambda)$ , with  $\rho$  given by Eq. (7). That higher-dimensional space can be sampled with a Metropolis–Hastings MCMC procedure by proposing a change from  $\vec{x}, \lambda$  to some new  $\vec{x}', \lambda'$  according to a generation probability  $P_{\text{gen}}(\vec{x}, \lambda \rightarrow \vec{x}', \lambda')$ . That proposal move is then conditionally accepted with probability

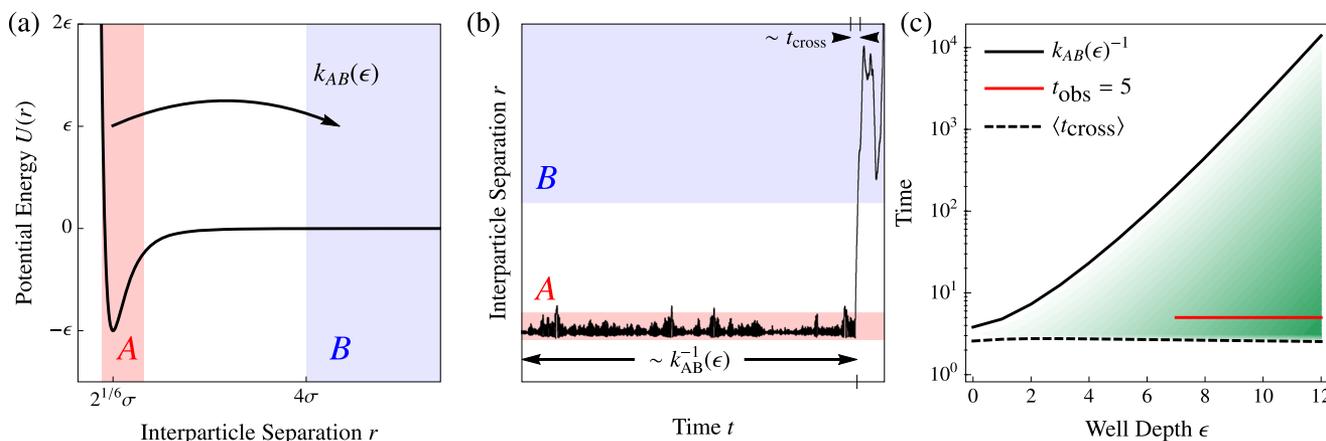
$$P_{\text{acc}} = \min \left[ 1, \frac{\rho(\vec{x}', \lambda') P_{\text{gen}}(\vec{x}', \lambda' \rightarrow \vec{x}, \lambda)}{\rho(\vec{x}, \lambda) P_{\text{gen}}(\vec{x}, \lambda \rightarrow \vec{x}', \lambda')} \right]. \quad (11)$$

The acceptance probability depends on the manner that new designs and trajectories are generated, that is, on  $P_{\text{gen}}$ . Specific choices of proposal moves are discussed in Appendix A, but a typical feature of those strategies is that the ratio of generation probabilities can be explicitly computed for any  $\vec{x}, \lambda$  and  $\vec{x}', \lambda'$ . In contrast, it is *not* typically possible to compute the ratio of  $\rho$  factors in Eq. (11),

$$\frac{\rho(\vec{x}', \lambda')}{\rho(\vec{x}, \lambda)} = \frac{Z_A(\lambda')^{-1}}{Z_A(\lambda)^{-1}} \times \frac{e^{-\beta H(\mathbf{x}'(0); \lambda')}}{e^{-\beta H(\mathbf{x}(0); \lambda)}} \times \frac{h_A(\mathbf{x}'(0)) h_B(\mathbf{x}'(t_{\text{obs}}))}{h_A(\mathbf{x}(0)) h_B(\mathbf{x}(t_{\text{obs}}))} \times \frac{P(\vec{x}' | \lambda', \mathbf{x}'(0))}{P(\vec{x} | \lambda, \mathbf{x}(0))}. \quad (12)$$

The only problematic term is the ratio of the reciprocal of the partition functions. Upon proposing a new  $\lambda'$ , Eqs. (11) and (12) require that one can compute  $Z_A(\lambda')^{-1}$ , but computing a partition function is computationally expensive. Even if one were to exhaustively compute  $Z_A(\lambda')$  by sampling phase space, the partition function would only be known up to some sampling error, and an unbiased estimate for  $Z_A$  would give a biased estimate for  $Z_A^{-1}$ . Inserting that biased noise into the acceptance probability would bias the Markov chain's stationary distribution.

The problem is quite similar to Ceperley and Dewing's consideration of Monte Carlo with noisy energies,<sup>32</sup> except now the noise comes from imperfect computations of the  $Z_A(\lambda)^{-1}$  terms. The resolution is to replace  $Z_A(\lambda)^{-1}$  by an unbiased estimate  $\widehat{Z_A(\lambda)^{-1}}(\eta)$ ,



**FIG. 1.** (a) Rate constants can be calculated by computing the probability of rare trajectories that execute transitions between states. Those rate constants depend on the molecular design. In the test system of Sec. III, the molecular design is the choice of well depth of a Lennard-Jones potential. The rare reactive trajectories start in region A and end in region B, given by the shaded red and blue areas, respectively. (b) An example of an unbinding trajectory that escapes from A to B. The timescale of the overall rate is labeled as  $k_{AB}^{-1}(\epsilon)$ , and the much shorter timescale of just the escape event is labeled as  $t_{\text{cross}}$ . (c) The transition path sampling framework for computing rates requires the separation of timescales:  $t_{\text{cross}} < t_{\text{obs}} \ll k_{AB}^{-1}(\epsilon)$ . The possible Lennard-Jones well depth  $\epsilon$  and observation time  $t_{\text{obs}}$  must be selected so the sampling is confined to the green shaded region, where that timescale separation is valid. The red line represents the  $t_{\text{obs}}$  and  $\epsilon$  range that we numerically sampled in Secs. III B and III C.

where the variables  $\boldsymbol{\eta}$  are all of the random numbers drawn from a distribution  $P(\boldsymbol{\eta})$  and used to estimate the reciprocal partition function. For example, if the estimate requires one to compute energies of representatively sampled configurations,  $\boldsymbol{\eta}$  would be the random numbers necessary to construct such samples and  $P(\boldsymbol{\eta})$  would be built up from the Gaussian or uniform distributions that the computer's random number generator used to select those random numbers. The unbiased estimate will appear naturally in the acceptance probability when one samples  $\vec{\mathbf{x}}, \boldsymbol{\lambda}$ , and  $\boldsymbol{\eta}$  in proportion to

$$\tilde{\rho}(\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta}) = P(\boldsymbol{\eta}) \widehat{Z_A(\boldsymbol{\lambda})}^{-1}(\boldsymbol{\eta}) h_A(\mathbf{x}(0)) h_B(\mathbf{x}(t_{\text{obs}})) \times e^{-\beta H(\mathbf{x}(0); \boldsymbol{\lambda})} P(\vec{\mathbf{x}} | \boldsymbol{\lambda}, \mathbf{x}(0)). \quad (13)$$

To be explicit, proposed changes  $\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta} \rightarrow \vec{\mathbf{x}}', \boldsymbol{\lambda}', \boldsymbol{\eta}'$  are accepted with probability

$$P_{\text{acc}} = \min \left[ 1, \frac{\tilde{\rho}(\vec{\mathbf{x}}', \boldsymbol{\lambda}', \boldsymbol{\eta}') P_{\text{gen}}(\vec{\mathbf{x}}', \boldsymbol{\lambda}' \rightarrow \vec{\mathbf{x}}, \boldsymbol{\lambda}) P_{\text{gen}}(\boldsymbol{\eta}')}{\tilde{\rho}(\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta}) P_{\text{gen}}(\vec{\mathbf{x}}, \boldsymbol{\lambda} \rightarrow \vec{\mathbf{x}}', \boldsymbol{\lambda}') P_{\text{gen}}(\boldsymbol{\eta})} \right]. \quad (14)$$

We assume that the new estimate of the reciprocal partition function is generated by drawing new random numbers from  $P_{\text{gen}}(\boldsymbol{\eta}')$  =  $P(\boldsymbol{\eta}')$ . As before, the ratio of  $P_{\text{gen}}$  terms for  $\vec{\mathbf{x}}$  and  $\boldsymbol{\lambda}$  in Eq. (11) can be explicitly computed. The remaining ratio takes the same form as Eq. (12) except that  $Z_A(\boldsymbol{\lambda})^{-1}$  has been replaced by the estimate  $\widehat{Z_A(\boldsymbol{\lambda})}^{-1}$ ,

$$\frac{\tilde{\rho}(\vec{\mathbf{x}}', \boldsymbol{\lambda}', \boldsymbol{\eta}') P_{\text{gen}}(\boldsymbol{\eta}')}{\tilde{\rho}(\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta}) P_{\text{gen}}(\boldsymbol{\eta})} = \frac{\widehat{Z_A(\boldsymbol{\lambda}')^{-1}}(\boldsymbol{\eta}')}{\widehat{Z_A(\boldsymbol{\lambda})^{-1}}(\boldsymbol{\eta})} \times \frac{e^{-\beta H(\mathbf{x}'(0); \boldsymbol{\lambda}')}}{e^{-\beta H(\mathbf{x}(0); \boldsymbol{\lambda})}} \times \frac{h_A(\mathbf{x}'(0)) h_B(\mathbf{x}'(t_{\text{obs}}))}{h_A(\mathbf{x}(0)) h_B(\mathbf{x}(t_{\text{obs}}))} \times \frac{P(\vec{\mathbf{x}}' | \boldsymbol{\lambda}', \mathbf{x}'(0))}{P(\vec{\mathbf{x}} | \boldsymbol{\lambda}, \mathbf{x}(0))}. \quad (15)$$

The resulting MCMC procedure in  $\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta}$  space can therefore accept and reject proposal moves based on the noisy estimate in lieu of the intractable reciprocal partition function. It is important to note that the reciprocal partition function estimates are computed from  $\boldsymbol{\eta}$ , which, upon rejection, does not get updated to the trial value  $\boldsymbol{\eta}'$ . As such, the old estimate of the reciprocal partition function with  $\boldsymbol{\eta}$  must be “reused” until the next move is accepted.

By choosing a noisy estimate that is unbiased, we ensure that we will recover the original  $\rho$  after marginalizing over the  $\boldsymbol{\eta}$  variables,

$$\rho(\vec{\mathbf{x}}, \boldsymbol{\lambda}) = \int d\boldsymbol{\eta} \tilde{\rho}(\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta}) \quad (16)$$

because

$$\left\langle \widehat{Z_A(\boldsymbol{\lambda})}^{-1} \right\rangle = \int d\boldsymbol{\eta} P(\boldsymbol{\eta}) \widehat{Z_A(\boldsymbol{\lambda})}^{-1}(\boldsymbol{\eta}) = Z_A(\boldsymbol{\lambda})^{-1}. \quad (17)$$

Indeed, this marginalization step required that we expressed Eq. (13) in terms of an estimate of the reciprocal of the partition function, not a reciprocal of an estimate of the partition function. The latter would have been simpler to estimate, but it would not have marginalized to give Eq. (7). Notably, we have not required that the acceptance

probabilities computed in Eq. (14) are unbiased estimates for those of Eq. (11); they generally will not be.

The sampling scheme does not assume a low-variance estimator, only one which is unbiased, but that variance can affect computational performance. A large-variance estimate typically results in more Monte Carlo rejections than one with low variance,<sup>32,34</sup> but it could nevertheless be advantageous to use the large-variance estimate if it is particularly cheap to compute.

From one perspective, the strategies employed are an exercise in the usefulness of lifts to Monte Carlo methods.<sup>35–39</sup> We ultimately are interested in sampling  $P(\boldsymbol{\lambda})$ , a distribution over possible designs, but we access that distribution by targeting higher-dimensional distributions. A lift from  $\boldsymbol{\lambda}$  to  $(\vec{\mathbf{x}}, \boldsymbol{\lambda})$  left us with the problematic reciprocal partition function, which we subsequently replaced by an estimate via a second lift to  $(\vec{\mathbf{x}}, \boldsymbol{\lambda}, \boldsymbol{\eta})$ . To utilize this final lift, we require a method for generating unbiased estimates of the reciprocal partition function, a problem addressed in a more general setting by Booth.<sup>33</sup>

### C. Estimating reciprocals of partition functions

The partition function  $Z_A(\boldsymbol{\lambda})$  involves an integral over all of phase space, both  $\mathbf{r}$  and  $\mathbf{p}$ . Because the Hamiltonian decouples into a potential energy depending on positions and a kinetic energy depending on momenta, the (classical) partition function can be decomposed as

$$Z_A(\boldsymbol{\lambda}) = \frac{1}{C} \tilde{Z}(\boldsymbol{\lambda}) \tilde{Z}_A(\boldsymbol{\lambda}), \quad (18)$$

where

$$\tilde{Z}_A(\boldsymbol{\lambda}) = \int d\mathbf{r} h_A(\mathbf{r}) e^{-\beta U(\mathbf{r}; \boldsymbol{\lambda})} \quad (19)$$

is the configurational partition function,

$$\tilde{Z}(\boldsymbol{\lambda}) = \int d\mathbf{p} e^{-\beta K(\mathbf{p}; \boldsymbol{\lambda})} \quad (20)$$

is the partition function for the momenta, and  $C$  is a constant that handles the exchange symmetry for identical particles and the discretization of phase space. For example, the case of identical classical particles in three dimensions gives  $C = h^{3N} N!$ , where  $h$  is Planck's constant. The integral over momenta  $\tilde{Z}(\boldsymbol{\lambda})$  does not need to be estimated because the quadratic form of kinetic energy allows it to be computed explicitly as a Gaussian integral. In contrast, for all but the simplest potential energies, we must estimate the configurational contribution to get estimates of the reciprocal partition functions,

$$\widehat{Z_A(\boldsymbol{\lambda})}^{-1} = C \tilde{Z}(\boldsymbol{\lambda})^{-1} \widehat{\tilde{Z}_A(\boldsymbol{\lambda})}^{-1}. \quad (21)$$

In this work, we limit ourselves to changes of design that alter neither  $C$  nor  $\tilde{Z}$ , in which case the contribution to a Monte Carlo acceptance ratio comes from the  $\widehat{\tilde{Z}_A(\boldsymbol{\lambda})}^{-1}$  term.

That term is the reciprocal of an integral over  $\mathbf{r}$ , precisely the situation where Booth's method provides an unbiased estimate.<sup>33</sup> The core insight behind Booth's approach is to replace the reciprocal of  $\tilde{Z}_A(\boldsymbol{\lambda})$  by a series expansion and to generate unbiased estimates for each term in that series. A similar idea, extended to partition functions of general probability distributions, is outlined in Ref. 40. Using a geometric series, the expansion can be written in terms of a

fixed reference design  $\lambda_{\text{ref}}$  as

$$\begin{aligned} \frac{1}{\tilde{Z}_A(\lambda)} &= \frac{1}{\int d\mathbf{r} h_A(\mathbf{r}) e^{-\beta U(\mathbf{r};\lambda)}} \\ &= \frac{(1 + a(1 + a(1 + a(1 + \dots))))}{\int d\mathbf{r} h_A(\mathbf{r}) e^{-\beta U(\mathbf{r};\lambda_{\text{ref}})}}, \end{aligned} \quad (22)$$

where

$$a = \frac{\int d\mathbf{r} h_A(\mathbf{r}) (e^{-\beta U(\mathbf{r};\lambda_{\text{ref}})} - e^{-\beta U(\mathbf{r};\lambda)})}{\int d\mathbf{r} h_A(\mathbf{r}) e^{-\beta U(\mathbf{r};\lambda_{\text{ref}})}} = 1 - \frac{\tilde{Z}_A(\lambda)}{\tilde{Z}_A(\lambda_{\text{ref}})} \quad (23)$$

must have a modulus less than one for the series to converge. Although  $a$  depends on  $\lambda$ , we have suppressed that dependence in the notation. It is generally intractable to compute  $a$  exactly, but we can get unbiased estimates by sampling independent configurations  $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots$  from the reference Boltzmann distribution  $P(\mathbf{r}; \lambda_{\text{ref}}) = h_A(\mathbf{r}) e^{-\beta U(\mathbf{r};\lambda_{\text{ref}})} / \tilde{Z}_A(\lambda_{\text{ref}})$ . The  $i$ th sampled configuration corresponds to the  $i$ th estimate,

$$a^{(i)} = 1 - e^{-\beta(U(\mathbf{r}^{(i)};\lambda) - U(\mathbf{r}^{(i)};\lambda_{\text{ref}}))}. \quad (24)$$

It is straightforward to confirm that each estimate  $a^{(1)}, a^{(2)}, \dots$  is unbiased,

$$\begin{aligned} \langle a^{(i)} \rangle &= \int d\mathbf{r}^{(i)} a^{(i)} P(\mathbf{r}^{(i)}; \lambda_{\text{ref}}) \\ &= 1 - \frac{\int d\mathbf{r}^{(i)} h_A(\mathbf{r}) e^{-\beta U(\mathbf{r}^{(i)};\lambda)}}{\tilde{Z}_A(\lambda_{\text{ref}})} = a. \end{aligned} \quad (25)$$

It follows that an unbiased estimate for  $\tilde{Z}(\lambda)^{-1}$  can be constructed by replacing each instance of  $a$  in Eq. (22) by  $a^{(1)}, a^{(2)}, \dots$ , as follows:

$$\frac{1}{\tilde{Z}_A(\lambda)} = \frac{1}{\tilde{Z}_A(\lambda_{\text{ref}})} (1 + a^{(1)} (1 + a^{(2)} (1 + \dots))). \quad (26)$$

As written, the series would require  $a$  to be estimated an infinite number of times. To be practically useful, we must convert the infinite series into a finite sum, ideally one with few terms. Truncation of the series at finite order, however, would introduce a bias to the estimate. Like Bhanot and Kennedy's unbiased estimates of  $e^x$ ,<sup>36,41</sup> Booth constructed an unbiased stochastic truncation from a "roulette procedure" that randomly chooses when to truncate the series in Eq. (26).<sup>33</sup> We use a similar roulette procedure where samples  $a^{(1)}, a^{(2)}, \dots$  are generated one by one. After sample  $a^{(n)}$  is generated, it is either incorporated into the nested product or it triggers the termination. Whether to incorporate  $a^{(n)}$  is determined by two factors: a tunable parameter  $0 < R < 1$  and the running product

$$\Pi^{(n)} \equiv \left| a^{(n)} \prod_{i=1}^{n-1} a^{(i)} \right| \max \left[ 1, \frac{R}{\Pi^{(i)}} \right]. \quad (27)$$

If  $\Pi^{(n)} < R$ , then the series is truncated with probability  $1 - (\Pi^{(n)}/R)$ . In the event of truncation, the estimate is constructed from the first  $n - 1$  terms as

$$\frac{1}{\tilde{Z}_A(\lambda)} = \frac{1}{\tilde{Z}_A(\lambda_{\text{ref}})} \left( 1 + \sum_{i=1}^{n-1} \prod_{j=1}^i a^{(j)} \max \left[ 1, \frac{R}{\Pi^{(j)}} \right] \right). \quad (28)$$

Appendix B explicitly shows that this stochastic truncation yields an unbiased estimate for the reciprocal of the partition function.

#### D. Sampling configurations from the reference distribution

In practice, our unbiased estimates  $a^{(i)}$  come from a library of pre-computed configurations  $\mathbf{r}^{(i)}$ , drawn from samples of the Boltzmann distribution  $P(\mathbf{r}; \lambda_{\text{ref}}) \propto e^{-\beta U(\mathbf{r};\lambda_{\text{ref}})}$ . Using a standard canonical sampling procedure with a fixed reference  $\lambda_{\text{ref}}$ , we store  $K$  independent configurations  $\mathbf{r}^{(i)}$  along with their respective reference potential energies  $U(\mathbf{r}^{(i)}; \lambda_{\text{ref}})$ . These  $K$  samples comprise a library that is generated only once. Every estimate of  $a$  is generated by drawing a configuration uniformly from this library and calculating  $a^{(i)}$  from Eq. (24). Hence,  $a$  can be estimated for each new value of  $\lambda$  using the same pre-sampled reference states.

The effectiveness of the method depends critically on the choice of the reference parameters  $\lambda_{\text{ref}}$ . Recall that for the series of Eqs. (22) and (26) to converge, we assumed  $|a^{(i)}(\lambda)| < 1$  for all  $i$ , an assumption that is guaranteed by choosing a reference with

$$U(\mathbf{r}; \lambda_{\text{ref}}) < U(\mathbf{r}; \lambda) + \frac{\ln 2}{\beta} \quad (29)$$

for each sampled configuration  $\mathbf{r}$ . Since the series should converge throughout the design sampling process, we furthermore want Eq. (29) to hold for all designs  $\lambda$ . The reference energy can be made sufficiently low in two ways. First, we can seek as  $\lambda_{\text{ref}}$  the parameters  $\lambda$  that minimize  $U(\mathbf{r}; \lambda)$  for all configurations  $\mathbf{r}$ , but a globally optimal  $\lambda_{\text{ref}}$  may not exist. Indeed, if the minimizing  $\lambda$  depends on the particular configuration  $\mathbf{r}$ , it is necessary to sample configurations according to a shifted reference energy  $U(\mathbf{r}; \lambda_{\text{ref}}) + U_0$ . In that case,  $\lambda_{\text{ref}}$  could be any  $\lambda$  (low energy is better), and the constant offset  $U_0$  is chosen such that

$$U_0 < \min_{\mathbf{r}} (U(\mathbf{r}; \lambda) - U(\mathbf{r}; \lambda_{\text{ref}})) + \frac{\ln 2}{\beta}. \quad (30)$$

These conditions on the reference energy ensure series convergence, but the guarantee comes with a computational cost. By shifting to a lower offset energy, the series truncates after more terms, a trend that is easily rationalized in the  $U_0 \rightarrow -\infty$  limit. Then, every  $a^{(n)}$  tends to 1, so  $\Pi^{(n)}$  is very slow to decay below  $R$ . Consequently, the series seldom chooses to terminate. A rapidly truncated convergent series thus demands a reference energy that is as high as possible without ever violating Eq. (30).

#### E. Biasing for faster rates with multiple trajectories

Section II A illustrated how to sample in proportion to a transition rate  $k_{AB}(\lambda)$ . Suppose, however, that the vast design space has a large design entropy. Many designs with slow rates would overwhelm the probability of sampling one of the comparatively few designs with fast rates. For the sampling procedure to discover those designs with anomalously fast transition rates, it generally requires a stronger bias in favor of fast rates. For example, one could sample designs in proportion to  $k_{AB}(\lambda)^L$  for some  $L$  greater than one. If  $L$  is an integer, this more strongly biased distribution can be sampled in analogy with Sec. II A by making use of  $L$  independent reactive

trajectories,<sup>42</sup> collectively sampling the distribution

$$P(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda) = \frac{1}{\mathcal{N}^L} \prod_{i=1}^L h_A(\mathbf{x}_i(0)) h_B(\mathbf{x}_i(t_{\text{obs}})), \quad (31)$$

$$\times \frac{e^{-\beta H(\mathbf{x}_i(0); \lambda)}}{Z_A(\lambda)} P(\vec{\mathbf{x}}_i | \lambda, \mathbf{x}_i(0)). \quad (32)$$

Assuming the same timescale separation that led to Eq. (9), integration over the trajectories indeed leaves the targeted marginal distribution

$$P(\lambda) = \int \mathcal{D}\vec{\mathbf{x}}_1 \int \mathcal{D}\vec{\mathbf{x}}_2 \dots \int \mathcal{D}\vec{\mathbf{x}}_L P(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda) = \left[ \frac{t_{\text{obs}}}{\mathcal{N}} k_{AB}(\lambda) \right]^L. \quad (33)$$

The multiple-trajectory joint distribution of Eq. (32) can be sampled with Metropolis–Hastings MCMC by proposing changes from  $(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda) \rightarrow (\vec{\mathbf{x}}'_1, \vec{\mathbf{x}}'_2, \dots, \vec{\mathbf{x}}'_L, \lambda')$  according to some calculable generation probability  $P_{\text{gen}}$ . The changes are conditionally accepted with acceptance probability,

$$P_{\text{acc}} = \min \left[ 1, \frac{P(\vec{\mathbf{x}}'_1, \vec{\mathbf{x}}'_2, \dots, \vec{\mathbf{x}}'_L, \lambda')}{P(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda)} \times \frac{P_{\text{gen}}(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda' \rightarrow \vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda)}{P_{\text{gen}}(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda \rightarrow \vec{\mathbf{x}}'_1, \vec{\mathbf{x}}'_2, \dots, \vec{\mathbf{x}}'_L, \lambda')} \right]. \quad (34)$$

Similar to the case of a single trajectory, one finds that the ratio of probabilities contains problematic ratios of reciprocal partition functions,

$$\frac{P(\vec{\mathbf{x}}'_1, \vec{\mathbf{x}}'_2, \dots, \vec{\mathbf{x}}'_L, \lambda')}{P(\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \dots, \vec{\mathbf{x}}_L, \lambda)} = \frac{Z_A(\lambda')^{-L}}{Z_A(\lambda)^{-L}} \times \prod_{i=1}^L \frac{e^{-\beta H(\mathbf{x}'_i(0); \lambda')}}{e^{-\beta H(\mathbf{x}_i(0); \lambda)}} \times \prod_{i=1}^L \frac{h_A(\mathbf{x}'_i(0)) h_B(\mathbf{x}'_i(t_{\text{obs}}))}{h_A(\mathbf{x}_i(0)) h_B(\mathbf{x}_i(t_{\text{obs}}))} \frac{P(\vec{\mathbf{x}}'_i | \lambda', \mathbf{x}'_i(0))}{P(\vec{\mathbf{x}}_i | \lambda, \mathbf{x}_i(0))}. \quad (35)$$

As before, we replace the reciprocal partition functions by unbiased estimates. Specifically,  $Z_A(\lambda)^{-L}$  is replaced by the product of  $L$  independent estimates of  $Z_A(\lambda)^{-1}$ , each computed as described in Sec. II B,

$$Z_A(\lambda)^{-L} \rightarrow \prod_{l=1}^L \widehat{Z_A(\lambda)^{-1}}. \quad (36)$$

A demonstration that the unbiased estimates may be used in the acceptance probabilities follows in analogy to Eq. (13). For each of the  $L$  estimates, one introduces a lift to include some noise variables  $\eta_l$ .

### III. RESULTS

To illustrate the design sampling with noisy estimates, we numerically studied the rate of escape from an energy well as a function of the well depth  $\epsilon$ . The potentially high-dimensional design  $\lambda$

of Sec. II is just the scalar  $\epsilon$  for this application. The toy problem was chosen to be sufficiently simple that brute force rate calculations  $k_{AB}(\epsilon)$  could also be collected to ensure that the procedure sampled designs—in this case well depths  $\epsilon$ —according to  $P(\epsilon) \propto k_{AB}(\epsilon)$ . Through numerical sampling, we confirmed that use of the noisy estimates  $\widehat{Z_A(\epsilon)^{-1}}$  do not bias the sampling. We further demonstrate that, however inconvenient to estimate, the partition function terms cannot be responsibly neglected; doing so yields a notable bias.

The specific toy model is the escape of a particle from a Lennard-Jones well while evolving with underdamped Langevin dynamics in three-dimensional space. The energy well takes the familiar form

$$U(r; \epsilon) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right], \quad (37)$$

where  $r$  is distance from the particle to the origin and  $\sigma$  is the particle radius. We take the  $A$  region to be the bottom of the well, defined by positions with  $0.85 \leq r/r_{\text{min}} \leq 1.4$ , with  $r_{\text{min}} = 2^{1/6}\sigma$  being the location of the potential energy minimum. The  $B$  region is the unbound state, reached once  $r$  exceeds  $4\sigma$ . At every moment of time, the particle experiences forces from this potential energy as well as a drag force and random fluctuating force from the underdamped Langevin dynamics. Hence,

$$\dot{\mathbf{p}}(t) = -\nabla U(\mathbf{r}(t); \epsilon) - \frac{\gamma}{m} \mathbf{p}(t) + \boldsymbol{\xi}(t), \quad (38)$$

where  $\gamma$  is a friction coefficient and  $\boldsymbol{\xi}$  is a white noise with  $\langle \boldsymbol{\xi} \rangle = \mathbf{0}$  and  $\langle \boldsymbol{\xi}(t) \boldsymbol{\xi}(t') \rangle = 2\gamma k_B T \delta(t - t')$ . We allow our tunable design parameter  $\epsilon$  to vary between  $7 k_B T$  and  $12 k_B T$ , a parameter regime chosen to ensure that escape is a rare event. For convenience, we nondimensionalize the problem by setting  $\sigma = m = \beta = 1$  and  $\gamma = 0.5$ . This system is illustrated in Fig. 1.

We split our results into four parts. First, we consider a joint Monte Carlo sampling of designs and particle positions to demonstrate that the noisy estimate for the reciprocal partition function can be adequately incorporated into a Monte Carlo acceptance ratio. We next perform the joint sampling of designs and trajectories to extract the dependence of the rate constant on the well depth in the range  $\epsilon \in [7, 12]$ . We demonstrate the enhanced preference for faster rate constants that comes from simultaneously sampling multiple reactive trajectories. Finally, we illustrate the methodology on a more complex molecular cluster.

#### A. Monte Carlo sampling with reciprocal partition function estimates

As detailed in Sec. II B, the Monte Carlo sampling over trajectories and well depths involves the computation of an acceptance ratio, Eq. (15), containing  $\widehat{Z_A(\epsilon)^{-1}}$ . We note, however, that the reciprocal partition function enters this ratio not because of the trajectory sampling but rather due to the sampling of the initial condition for the trajectory. To evaluate the consequences of estimating  $\widehat{Z_A(\epsilon)^{-1}}$ , we first chose to study a simpler subproblem: simultaneous sampling of well depths and initial positions (as opposed to full trajectories).

We constructed MCMC moves that transition from an old position and well depth,  $\mathbf{r}$  and  $\epsilon$ , to a new position and depth,  $\mathbf{r}' = \mathbf{r} + \Delta \mathbf{r}$  and  $\epsilon' = \epsilon + \Delta \epsilon$ . The symmetric proposal is generated by drawing independent Gaussian variables  $\Delta r$  and  $\Delta \epsilon$ , each with zero mean and variance  $10^{-4}$ . When a trial move generated  $\epsilon'$  outside

the range,<sup>7,12</sup> the move was rejected. Otherwise, the moves were accepted in one of three different ways: according to the exact partition function

$$P_{\text{acc}}^{(\text{exact})} = \min \left[ 1, h_A(\mathbf{r}') \frac{\tilde{Z}_A(\epsilon')^{-1} e^{-\beta U(\mathbf{r}'; \epsilon')}}{\tilde{Z}_A(\epsilon)^{-1} e^{-\beta U(\mathbf{r}; \epsilon)}} \right], \quad (39)$$

according to the estimated partition function

$$P_{\text{acc}}^{(\text{est})} = \min \left[ 1, h_A(\mathbf{r}') \frac{\tilde{Z}_A(\epsilon')^{-1} e^{-\beta U(\mathbf{r}'; \epsilon')}}{\tilde{Z}_A(\epsilon)^{-1} e^{-\beta U(\mathbf{r}; \epsilon)}} \right], \quad (40)$$

or neglecting the partition function altogether

$$P_{\text{acc}}^{(\text{ignored})} = \min \left[ 1, h_A(\mathbf{r}') \frac{e^{-\beta U(\mathbf{r}'; \epsilon')}}{e^{-\beta U(\mathbf{r}; \epsilon)}} \right]. \quad (41)$$

We note that for Eq. (40), one must retain the old estimate for the reciprocal partition function after a rejection rather than recomputing a new estimate. This need follows from the fact that we formally consider a random walk through the  $\eta$  coordinates that produced the noisy estimate, as described in Sec. II B.

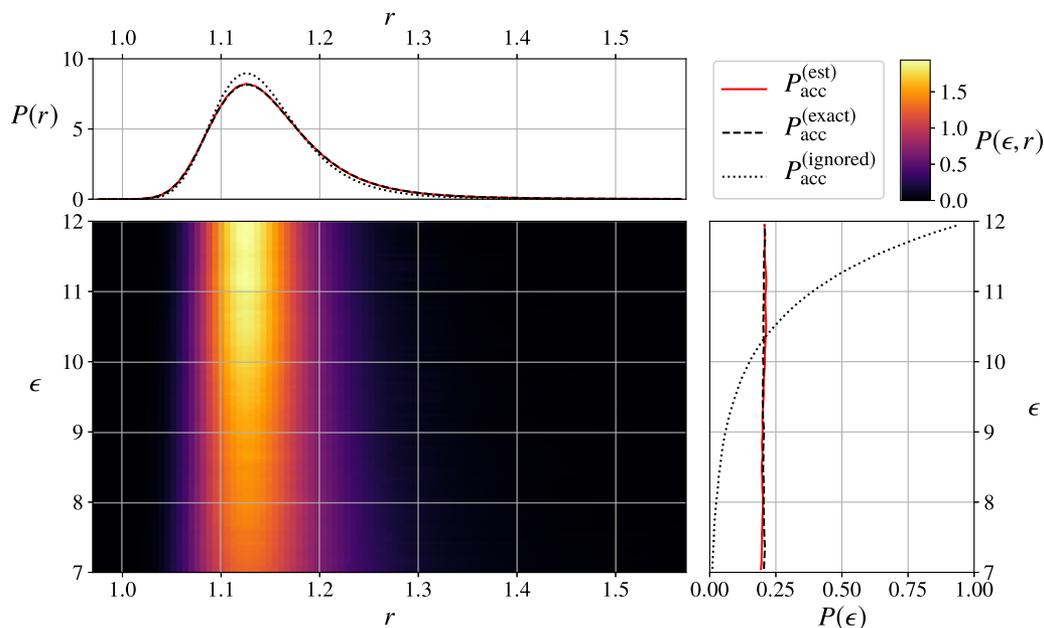
Equations (39) and (40) should both sample marginal distributions for  $\epsilon$  that are uniform, while Eq. (41) samples  $P(\epsilon) \propto Z_A(\epsilon)$ .

We confirmed these marginal distributions numerically by sampling particle positions in the Lennard-Jones well, a comparison made possible by the ease of numerically computing the exact partition function

$$\tilde{Z}_A(\epsilon) = 4\pi \int_{0.85r_{\text{min}}}^{1.4r_{\text{min}}} dr r^2 e^{-\beta U(r; \epsilon)} \quad (42)$$

for this toy model, with  $U(r; \epsilon)$  given by Eq. (37).

Figure 2 gives the joint probability density from sampling in  $\epsilon$  and  $r$  space using reciprocal partition function estimation [Eq. (40)]. The marginal distributions in  $\epsilon$  and  $r$  are shown along their respective axes and give comparisons with sampling using the exact partition function [Eq. (39)] and ignoring the partition function contributions [Eq. (41)]. We can see that using unbiased reciprocal partition function estimation works well, matching the results obtained from the exact partition functions. Both the estimated and exact approaches also result in a uniform marginal distribution across  $\epsilon$ , as expected. By carefully constructing an unbiased estimation procedure, we have recovered the proper sampling without having to laboriously calculate an exact partition function at every sampled value of  $\epsilon$ . The marginal distribution of  $\epsilon$  also shows the effect of the reciprocal partition function ratio on the sampling procedure. By ignoring this ratio, we sample  $\epsilon$  in proportion to  $\tilde{Z}_A(\epsilon)$ , which introduces a bias that prefers higher values of  $\epsilon$ . The non-uniform distribution reflects the fact that trajectory sampling would show a



**FIG. 2.** The joint probability densities of  $\epsilon$  and  $r$  for the sampling procedure described by Eq. (40). Marginal probability densities for  $\epsilon$  and  $r$  are given along their respective axes from the joint distribution. Marginal densities show data from the sampling procedure using reciprocal partition function estimates [Eq. (40)] (red), the exact partition function [Eq. (39)] (dashed black), and incorrectly ignoring the reciprocal partition function contributions [Eq. (41)] (dotted black). Distributions were each collected from  $5 \times 10^8$  Monte Carlo trial moves, and the estimated reciprocal partition functions were computed with a roulette parameter of  $R = 0.75$  and a reference value of  $\epsilon_{\text{ref}} = 12$ . Marginal densities for the estimation procedure and the exact partition function agree well, and the marginal density of  $\epsilon$  is uniform, as expected. Ignoring the reciprocal partition function introduces a non-negligible bias; rather than sampling  $\epsilon$  uniformly, it is sampled in proportion to  $\tilde{Z}_A(\epsilon)$ .

preference for some designs not only because of the propensity to react but also due to the ease of generating initial configurations. From Fig. 2, it is clear that this bias can be significant.

Of course to sample the joint distribution in Fig. 2, we need to generate estimates for the reciprocal partition functions. The first step in doing so is to choose an appropriate reference  $\epsilon_{\text{ref}}$  following the considerations of Sec. II D. For the range  $\epsilon \in [7, 12]$ , a value of  $\epsilon_{\text{ref}} = 12$  gives the lowest energy at any value of  $r$  due to the monotonically decreasing energy of the Lennard-Jones form with increasing  $\epsilon$ . There is no need for an offset  $U_0$ . Using standard MCMC for fixed  $\epsilon_{\text{ref}}$ , we generated 10 000 independent samples of position  $\mathbf{r}$  and corresponding reference energy  $U(\mathbf{r}; \epsilon_{\text{ref}})$  to construct a library. As described in Sec. II D, we generated  $a^{(i)}$  terms in the expansion of Eq. (28) for arbitrary  $\epsilon$  by uniformly randomly drawing samples from the library and evaluating Eq. (24). The estimates of Fig. 2 were generated with a roulette parameter of  $R = 0.75$ .

Figure 3(a) shows that the reciprocal partition function remains unbiased regardless of the choice of  $R$ , but that  $R$  impacts the estimator's noise. The figure also highlights that the stochastic truncation of the series expansion is essential and that termination at a fixed expansion length would yield biased estimates. Likewise, the reciprocal of our stochastic series is a *biased* estimate of the partition function itself, as shown in Fig. 3(b). This bias is expected and not problematic; our MCMC scheme required unbiased  $\tilde{Z}_A^{-1}$ , not unbiased  $\tilde{Z}_A$ . While the procedure formally works irrespective of  $R$  or the distance from the reference potential, the cost of the estimation procedure varies. Figure 3(c) shows that decreasing  $R$  decreases the probability of truncation, resulting in a stochastic sum with more terms. A series with more terms is better converged since it effectively averages over many more values of  $a^{(i)}$ , but the decrease in the noise comes at a computational expense.

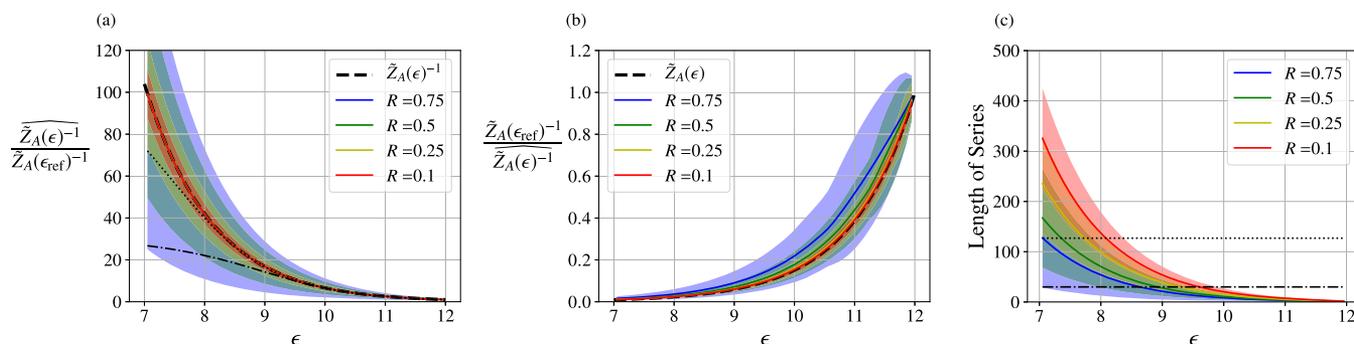
Tuning  $R$  to select an optimal trade-off between noise and computational cost is a complicated affair. One advantage of using

estimates to sample is that one can get by with noise, potentially very large noise without introducing bias, suggesting that one should favor very cheap, noisy estimates. However, very noisy estimates can cause the Markov chain to get stuck in  $\eta$  variables that produce an overly favorable estimate. Practical implementations require care—in choosing  $R$ , in selecting a reference potential, and in preventing stuck Markov chains—but our calculations serve as a demonstration of the principle that the noisy estimates of reciprocal partition functions can be computed and productively employed.

## B. Simultaneously sampling trajectory space and design space

Having demonstrated the ability to sample the design and the initial condition, we now want to bias designs so as to favor fast rates. Section II laid out two routes to sample  $P(\epsilon) \propto k_{AB}(\epsilon)$ . If we could compute  $\tilde{Z}_A^{-1}$  exactly, we could sample designs and trajectories according to Eq. (7) with a MCMC procedure that updates a trajectory  $\vec{x}$  and a design  $\epsilon$ . Otherwise, we could also generate estimates for the reciprocal of the partition function to sample Eq. (13). As in Sec. III A, we consider the Lennard-Jones escape problem because it is reasonable to implement both routes—using exact and estimated reciprocal partition functions—as a demonstration of validity.

Both routes require a random generation of new designs and trajectories,  $P_{\text{gen}}(\vec{x}, \epsilon \rightarrow \vec{x}', \epsilon')$ . We make these proposals in two steps. First, we symmetrically generate  $\epsilon' = \epsilon + \Delta\epsilon$  as in Sec. III A. Next, we use  $\epsilon'$  to generate a new trajectory via a “shooting move” that re-evolves the stochastic dynamics forward and backward in time from a randomly selected time.<sup>3</sup> The combined move is conditionally accepted according to Eq. (11) (exact) or to Eq. (14) (est), both of which require an explicit calculation of the ratio of generation probabilities, as well as a measure of the Boltzmann probability of the new trajectory's initial condition. That Boltzmann

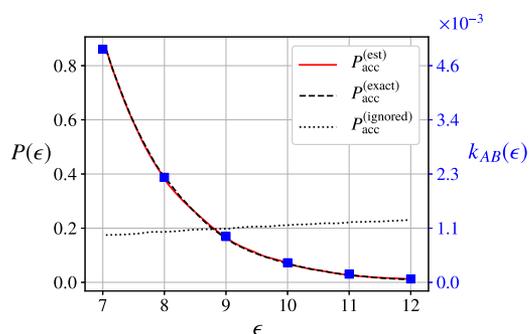


**FIG. 3.** (a) Unbiased estimates of the reciprocal partition function  $\tilde{Z}_A(\epsilon)^{-1}$ . (b) Biased estimates of the partition function  $\tilde{Z}_A(\epsilon)$  constructed by taking the reciprocal of  $\tilde{Z}_A(\epsilon)^{-1}$ . Both estimates are scaled by the value of the exact partition function at the reference  $\epsilon_{\text{ref}} = 12$ , and the exact partition function is given as a thick dashed black line. (c) The series length when using the Booth method as a function of the well depth  $\epsilon$ . Different colors represent different roulette parameters with the solid lines giving the average value and the shaded areas showing the standard deviation. To emphasize the importance of stochastic truncation, biased results are shown for series truncated at fixed lengths of 30 and 127 terms, plotted as dashed-dotted and dotted lines, respectively. Average estimates of the reciprocal partition function are indistinguishable from the exact result in (a), demonstrating the unbiased nature of the Booth estimation procedure. The average estimate values only approach the exact values in the limit  $R \rightarrow 0$  in (b), showing that the reciprocal of our unbiased estimates for  $\tilde{Z}_A^{-1}$  does not yield an unbiased estimate of  $\tilde{Z}_A$ . From (c), we can see that the cost of the estimation method increases as  $R$  decreases and also as the value of  $\epsilon$  strays further from the reference  $\epsilon_{\text{ref}} = 12$ .

probability of the initial condition is handled as in Sec. III A—we either compute it exactly (exact) or we generate an unbiased estimate (est). Unlike Sec. III A, we now must compute the ratio of the trajectory generation probabilities, a ratio that can be computed explicitly in terms of the random noise terms for the stochastic dynamics. Appendix A provides details of the trajectory generation probability based on the underdamped Langevin integrator of Athènes and Adjanor.<sup>43</sup>

To confirm that the trajectory sampling approach yields the rate of unbinding, we additionally computed the rate constant as a function of well depth  $k_{AB}(\epsilon)$  by brute force. For those brute force calculations, we used MCMC to initialize the Lennard-Jones particle's position and momentum from the equilibrium Boltzmann distribution and then propagated the trajectory<sup>43</sup> with a time step of  $\Delta t = 0.005$ . Once the system reached  $B$ , we recorded the elapsed time  $\tau$  and repeated the procedure. In total, we sampled  $10^6$  realizations of  $\tau$  and calculated an estimate of  $k_{AB}$  as  $1/\langle\tau\rangle$ . In Fig. 4, we overlaid the rate constant data on top of the marginal distributions of  $\epsilon$  taken from trajectory sampling.

Figure 4 shows that using our reciprocal partition function estimation procedure works well as it matches the data collected when using an exact partition function. Furthermore, by overlaying the rate constant data, we show that we are indeed sampling  $\epsilon$  in proportion to the rate constant,  $P(\epsilon) \propto k_{AB}(\epsilon)$ . Consequently, the random walkers executing both the exact and estimation approaches spend most of their time sampling designs with fast rates  $\epsilon \approx 7$ . In contrast, the random walker that ignores the partition function factor in its acceptance ratio spends most of its time sampling the slow designs, those with  $\epsilon \approx 12$ . These data confirm that the partition function term can be very significant in rate design problems and that it can be computed approximately in a manner that avoids sampling bias.



**FIG. 4.** (Left, black axis) Marginal probability densities of  $\epsilon$  from simultaneous sampling of the design space  $\epsilon$  and of unbinding trajectories. Sampling using the exact reciprocal partition functions [Eq. (11)] gives the dashed black curve and sampling with estimated reciprocal partition functions [Eq. (14)] gives the solid red curve. (Right, blue axis) Brute force estimations of the average rate constants of unbinding  $k_{AB}(\epsilon)$  as a function of  $\epsilon$  given as blue squares. Each trajectory sampling procedure used  $5 \times 10^8$  trial attempts, and the estimation procedure used a roulette parameter  $R = 0.75$  and a reference value of  $\epsilon_{\text{ref}} = 12$ . Each rate calculation consisted of  $10^6$  independent trials. To overlay  $P(\epsilon)$  and  $k_{AB}(\epsilon)$ , the proportionality constant was fitted by least squares regression.

### C. Sampling with stronger bias for fast rates

As discussed in Sec. II E, visiting designs in proportion to their rates is a relatively weak preference in favor of fast rates. In high-dimensional design spaces, the design entropy overwhelms that weak preference, so we sought a way to turn up the bias by sampling  $P(\epsilon) \propto k(\epsilon)^L$  with  $L$  independent reactive trajectories. Implementing this scheme follows quite directly from Sec. III B. The principal difference is that a MC move in  $\epsilon$  and  $\vec{x}$  now becomes a move in  $\epsilon, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_L$ . Specific computational details are discussed in Appendix A.

Figure 5 summarizes the result of sampling with  $L = 1, 2,$  and  $3$ . The fastest rate of escape occurs with the shallowest allowed well,  $\epsilon = 7$ , but with  $L = 1$ , there is still appreciable probability of seeing  $\epsilon$  fluctuate to suboptimal values above 9 or 10. By increasing  $L$ , the density near the optimal  $\epsilon$  grows. Figure 5(a) shows that sampling using the estimated reciprocal partition functions remains unbiased for  $L > 1$ . The more stringent confirmation that  $P(\epsilon) \propto k(\epsilon)^L$ , or equivalently that  $\log P(\epsilon)$  vs  $\log k(\epsilon)$  has slope  $L$ , is plotted in Figs. 5(b)–5(d).

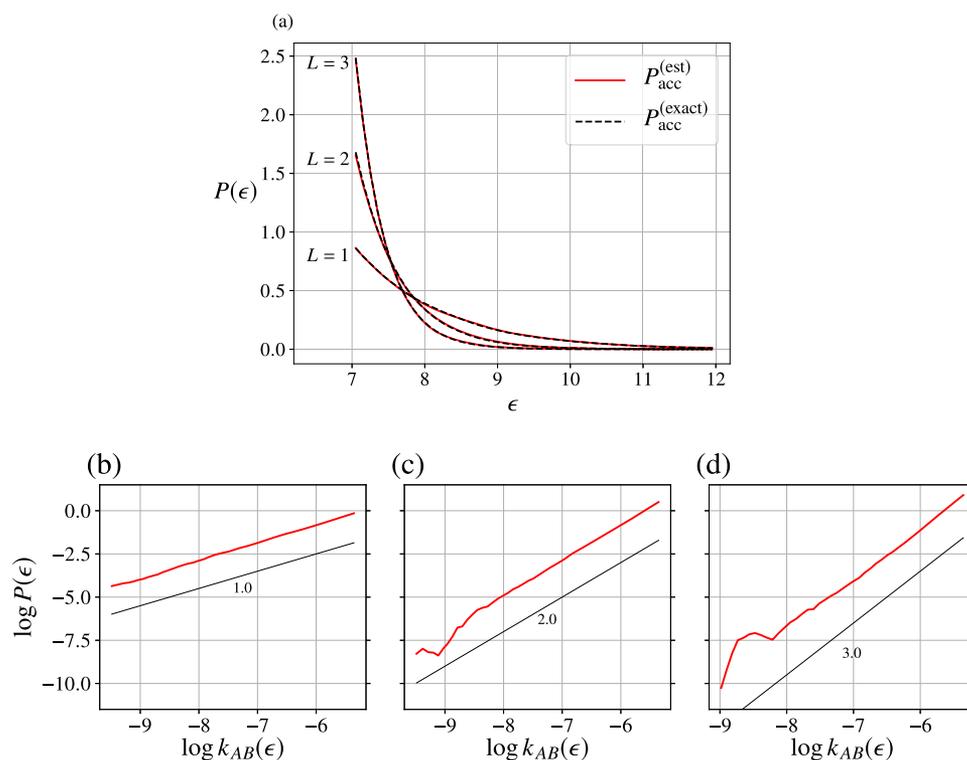
### D. A more complex example

While the Lennard-Jones unbinding system is useful for demonstrating correctness, the power of the sampling approach is most applicable to more complex problems with high-dimensional state spaces and design spaces. To illustrate such an application, we introduce a five-particle system shown in Fig. 6 that consists of one particle metastably trapped inside a tetrahedron of four other particles. Given enough time, the trapped particle escapes from the tetrahedron with a rate constant that depends on the various interactions between particles. Whereas the Lennard-Jones unbinding example supported a single trivial reaction pathway, competing reaction pathways are possible with multiple particles. In this respect, this toy model serves as a more realistic example that builds toward chemical applications.

In addition to increasing the dimensionality of the state space, we now consider a three-dimensional design space that regulates interparticle interactions. One of those design degrees of freedom adjusts a spring constant  $k$  for a harmonic potential acting between pairs of the tetrahedron particles. When separated by a distance  $r_{ij}$ , this pair potential contributes  $U_{\text{harmonic}}(r_{ij}; k) = \frac{1}{2}kr_{ij}^2$  to the energy. All five particles additionally interact through modified Lennard-Jones potentials,

$$U_{\text{LJ}}(r_{ij}; \epsilon_R, \epsilon_A) = 4\epsilon_R \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - 4\epsilon_A \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6. \quad (43)$$

Here,  $\sigma_{ij}$  is the mean of the Lennard-Jones radii of particles  $i$  and  $j$ . We call the potentials modified Lennard-Jones potentials because we allow attractive and repulsive interactions to be, respectively, tuned via  $\epsilon_A$  and  $\epsilon_R$ . In principle, those  $\epsilon_A$  and  $\epsilon_R$  parameters could be distinct for each pairwise interaction, but we wanted to limit to a three-dimensional design space in order to visually compare with brute force calculations. Thus, we held fixed  $\epsilon_A = 0$  and  $\epsilon_R = 10$  between tetrahedron particles and took  $\epsilon_A$  and  $\epsilon_R$  between the trapped particle and each tetrahedron particle to be the two remaining design parameters. All other parameters were held fixed:  $k_B T = 1$ , and



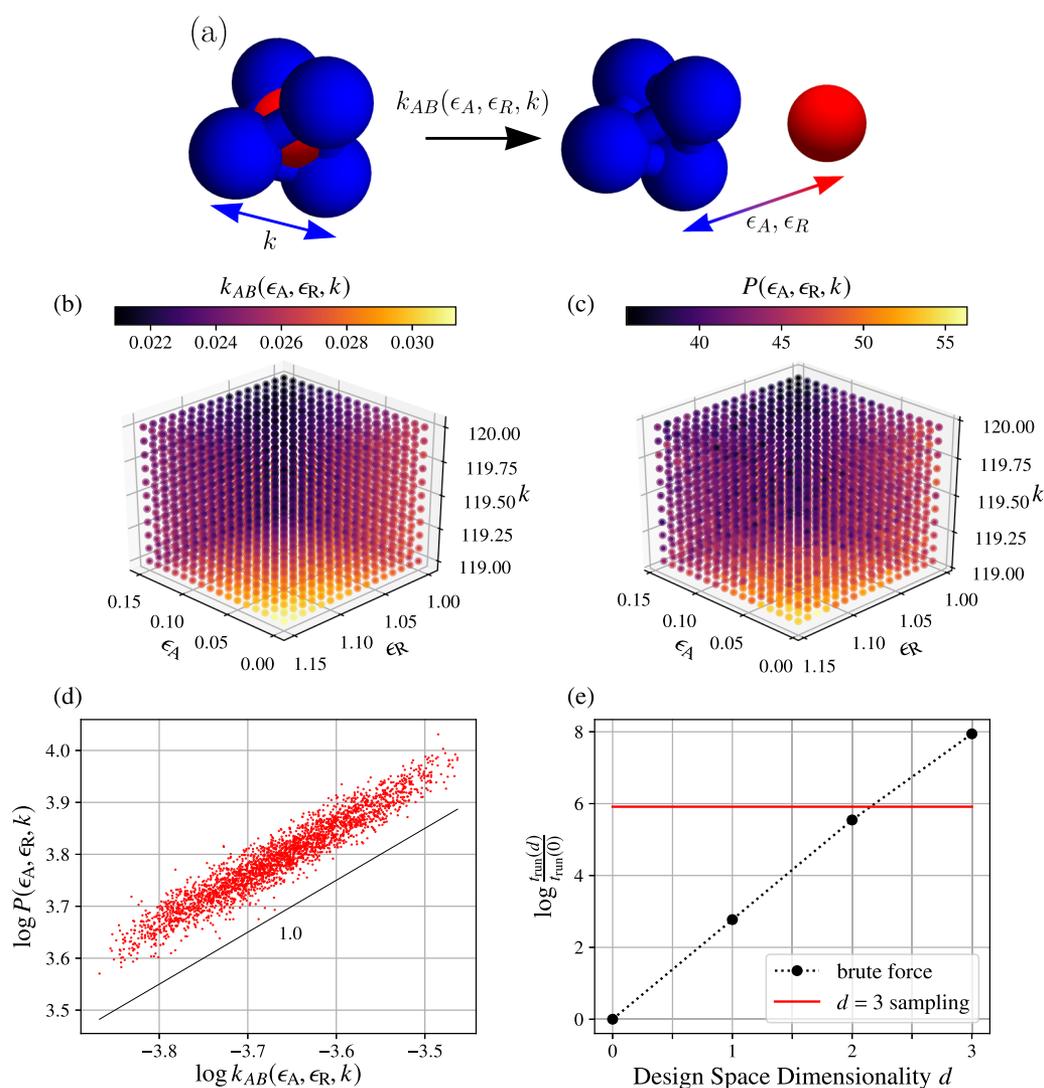
**FIG. 5.** (a) Marginal probability densities of  $\epsilon$  from joint sampling of the design space  $\epsilon$  and of  $L$  independent unbinding trajectories, as described in Sec. II E. Accounting for the reciprocal partition function exactly is shown in black dashed line, and using the reciprocal partition function estimations is shown in red solid line. (b) The marginal probability densities from the estimation procedure in (a) are plotted against the brute force calculated rate constants (blue squares in Fig. 4) on a log-log plot for (b)  $L = 1$ , (c)  $L = 2$ , and (d)  $L = 3$ . Power law scalings are provided in solid black line as reference. The brute force rate constants were fitted to a high-order polynomial for interpolation purposes. Each trajectory sampling procedure used  $5 \times 10^8$  trial attempts, and the estimates used a roulette parameter of  $R = 0.75$  and a reference value of  $\epsilon_{\text{ref}} = 12$ . From (a), we can see that increasing the number of independent trajectories in the sampling creates a stronger preference for the fastest unbinding rate, which occurs at  $\epsilon = 7$ . From (b), (c), and (d), we conclude that we are sampling according to  $P(\epsilon) \propto k_{AB}(\epsilon)^L$ . The power law scaling at low values of  $k_{AB}$  (high values of  $\epsilon$ ) is noisy for larger  $L$  values because slow values of  $k_{AB}$  are rendered particularly rare by using a large number of independent trajectories.

$\gamma = 0.5$ ,  $\Delta t = 0.005$ , unit mass for all particles, unit radii for all tetrahedron particles, and a trapped particle radius of 0.45. The design  $\lambda = \{k, \epsilon_A, \epsilon_R\}$  was allowed to sample, restricted to the domain with  $\epsilon_R \in [1, 1.15]$ ,  $\epsilon_A \in [0, 0.15]$ , and  $k \in [119, 120]$ . Trial moves in parameter space were generated via  $\epsilon'_A = \epsilon_A + \Delta\epsilon_A$ ,  $\epsilon'_R = \epsilon_R + \Delta\epsilon_R$ , and  $k' = k + \Delta k$ , where  $\Delta\epsilon_A$ ,  $\Delta\epsilon_R$ , and  $\Delta k$  were drawn from Gaussian distributions with 0 mean and variances of  $10^{-4}$ ,  $10^{-4}$ , and  $10^{-2}$ , respectively. The criteria for detecting reactant and product states were based on the distance between the center of mass of the tetrahedron and the trapped particle. When that distance is below 0.25, the configuration is in the reactant state  $A$ , and when it exceeds 1.1, the configuration is in the product state  $B$ .

Figure 6 demonstrates the merits of the Monte Carlo methodology for this three-dimensional design problem. The brute force rate calculations of Fig. 6(b) reveal relative rates between designs, which are essentially indistinguishable from the design sampling of Fig. 6(c). In both cases, the rate of the trapped particle's escape is

maximized when the tetrahedron is the least rigid (low  $k$ ) and when the interactions between the tetrahedron particles and the trapped particle are most repulsive and least attractive (high  $\epsilon_R$ , low  $\epsilon_A$ ). The proportionality between the brute force rate and design sampling is made quantitative by comparing  $\log P(\lambda)$  with  $\log k_{AB}(\lambda)$  in Fig. 6(d).

The computational benefit of the sampling procedure becomes apparent in this higher-dimensional parameter space. Figure 6(e) makes this benefit clear by comparing run times for computing rates in  $d$  dimensions with  $d = 0$  corresponding to a single design,  $d = 1$  corresponding to  $\lambda = \{\epsilon_A\}$ ,  $d = 2$  corresponding to  $\lambda = \{\epsilon_A, \epsilon_R\}$ , and  $d = 3$  corresponding to  $\lambda = \{\epsilon_A, \epsilon_R, k\}$ . The cost for performing calculation on a grid like this scales exponentially, whereas the sampling procedure's cost depends only on the number of sampled designs, not the design dimensionality. Because of the poor scaling of grids, the sampling procedure eventually wins out as dimensionality increases, a crossover that occurred after  $d = 2$  for our implementation.



**FIG. 6.** (a) A tetrahedral cluster consisting of five Lennard-Jones particles releases the trapped red particle from its blue cage with a rate that depends on three parameters of the pairwise interactions. (b) Brute force rate calculations for a grid of designs, estimated as in the Lennard-Jones unbinding example as the reciprocal of the average escape time of  $2 \times 10^4$  independent underdamped Langevin trajectories. (c) Binned probability density for design sampling, revealing relative rate constants that match the brute force calculation. Design sampling utilized trajectories with  $M = 1000$  time steps, reciprocal partition function parameter  $R = 0.75$ , and reference design  $\epsilon_A = 0.15$ ,  $\epsilon_R = 1$ , and  $k = 119$  that was sampled by  $10^6$  reference configurations. Trajectories were sampled as described in Appendix A. (d) A quantitative comparison of the brute force rates and the sampled designs. As in Fig. 5(b), the unit slope on the log-log scale reflects equality up to a proportionality constant. (e) Sampled designs can be generated with a cost independent of design dimensionality  $d$ , while brute force rate computations scale exponentially with  $d$ . Although we never aimed to fully optimize the sampling strategy, the runtime  $t_{\text{run}}$  to generate (d) was already below that of (c) for the three-dimensional design problem.

#### IV. DISCUSSION

In this manuscript, we have demonstrated how to sample design spaces with a preference for fast reaction rates. Our central focus has been the development of a practical Monte Carlo strategy that samples  $p(\lambda) \propto k_{AB}(\lambda)^L$  without bias. Our success studying toy models demands a level-headed assessment of whether the methodology will scale to more complex problems with still higher

dimensional  $\lambda$ . As a Monte Carlo strategy, there is reason to believe that high-dimensional spaces could be accessible, but here we must highlight two reasons for caution.

One concern is that our unbiased estimates of  $Z_A(\lambda)^{-1}$  came from a comparison against a single fixed reference with energy  $U(\mathbf{r}; \lambda_{\text{ref}}) + U_0$ . Like the estimation of free energy differences from importance sampled configurations, efficient computations rely on good overlap with the reference distribution. In our case, we require

the typical  $\mathbf{r}$  sampled by  $U(\mathbf{r}; \lambda)$  to be similar to those typically sampled by the reference potential. Adequate overlap was simple to achieve in the toy problems due to our focus on  $d \leq 3$  design dimensions. We expect greater difficulty in higher dimensions, where it may be necessary to generate unbiased estimates of  $Z_A(\lambda)^{-1}$  using samples from multiple different reference potentials. We expect that the multistate Bennett acceptance ratio method for unbiased free energy calculations likely guides the way toward simultaneously weighting multiple references.<sup>44</sup>

Another concern one might raise is that the sampling procedure directly provides an ensemble of decent designs rather than a single optimal design, yet often it is this optimal design that is desired. The situation is analogous with a finite-temperature canonical ensemble returning configurations that differ from the energy minima. To discover those minima, it is necessary to quench by progressively lowering the temperature. For the design sampling problem we showed that quench could be achieved by increasing  $L$ , the number of random walkers. However, the principal virtue of the method is that it allows one to importance sample design space, not just perform quenches. This importance sampling promises insight going beyond the identification of an optimal design. For example, it should be possible to assess how many near-optimal designs exist and by how much their performance erodes relative to the optimum.

## ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Comet cluster at the San Diego Supercomputer Center through startup allocation Grant No. TG-CHE190024. T.R.G. gratefully acknowledges a long history of useful conversations and guidance from Michael Grünwald and Phillip Geissler. T.R.G. and A.A. thank Geyao Gu for many productive discussions.

## APPENDIX A: GENERATION OF NEW TRAJECTORIES

The Monte Carlo procedure of Sec. II B expresses the probability of accepting a MC move in terms of a ratio of generation probabilities  $P_{\text{gen}}(\vec{\mathbf{x}}, \lambda \rightarrow \vec{\mathbf{x}}', \lambda') / P_{\text{gen}}(\vec{\mathbf{x}}', \lambda' \rightarrow \vec{\mathbf{x}}, \lambda)$ . Here, we discuss the details of the transition path sampling moves that allow us to compute the acceptance ratio and efficiently sample the trajectory space. We focus on sampling stochastic dynamics, specifically discretized underdamped Langevin dynamics, in which case shooting moves combined with the strategy of Crooks is effective.<sup>46</sup> The trajectory  $\vec{\mathbf{x}}$  consists of  $M + 1$  snapshots of phase space,  $\mathbf{x} = \{\mathbf{r}, \mathbf{p}\}$ , separated by time step  $\Delta t$ . It is convenient to have a compact notation, so the  $j$ th snapshot occurring at time  $j\Delta t$ ,  $\mathbf{x}(j\Delta t)$ , will simply be written as  $\mathbf{x}^j$ . The trajectory  $\vec{\mathbf{x}}$  is defined either by this set  $\{\mathbf{x}^j\}$  for all  $j = 0, \dots, M$  or alternatively by the initial condition  $\mathbf{x}^0$  and the set of all random numbers that cause the stochastic integrator to visit the subsequent points in phase space.

In Sec. III B, we employed the Langevin integrator of Athènes and Adjanor,<sup>43</sup> in which case each degree of freedom requires two Gaussian random variables (noises) to be drawn for each time step. Similar to the velocity Verlet algorithm, the position coordinates defined at integer time steps are computed using velocities that are defined at integer and half-integer times. The two noises for each

degree of freedom can also be associated with fractional time steps, so the update of particle  $i$  can be computed as

$$\begin{aligned} \mathbf{p}_i^{j+\frac{1}{2}} &= \mathbf{p}_i^j e^{-\frac{\gamma \Delta t}{2m_i}} + \xi_i^j \frac{\Delta t}{2} + \xi_i^{j+\frac{1}{2}}, \\ \mathbf{r}_i^{j+1} &= \mathbf{r}_i^j + \mathbf{p}_i^{j+\frac{1}{2}} \frac{\Delta t}{m_i}, \\ \mathbf{p}_i^{j+1} &= \left[ \mathbf{p}_i^{j+\frac{1}{2}} + \xi_i^{j+1} \frac{\Delta t}{2} \right] e^{-\frac{\gamma \Delta t}{2m_i}} + \xi_i^{j+1}, \end{aligned} \quad (\text{A1})$$

where the Gaussian white noises  $\xi_i^{j+\frac{1}{2}}$  and  $\xi_i^{j+1}$  both have mean zero and variance  $m_i(1 - e^{-\frac{\gamma \Delta t}{m_i}})/\beta$ . The trajectory can then be expressed as  $\vec{\mathbf{x}} \equiv \{\mathbf{x}(0), \xi^{\frac{1}{2}}, \xi^1, \dots, \xi^{j+\frac{1}{2}}, \xi^{j+1}, \dots, \xi^{M-\frac{1}{2}}, \xi^M\}$ . We can also define the time-reversed trajectory as  $\vec{\mathbf{x}} \equiv \{\mathbf{x}(M\Delta t), \tilde{\xi}^M, \tilde{\xi}^{M-\frac{1}{2}}, \dots, \tilde{\xi}^{j+1}, \tilde{\xi}^{j+\frac{1}{2}}, \dots, \tilde{\xi}^1, \tilde{\xi}^{\frac{1}{2}}\}$ , where  $\tilde{\xi}$  are the random numbers that would give a reversed trajectory through phase space starting from the endpoint with reversed momenta.

The MC proposal  $\vec{\mathbf{x}}, \lambda \rightarrow \vec{\mathbf{x}}', \lambda'$  is thus constructed as follows. First, we generate  $\lambda \rightarrow \lambda'$  symmetrically, and if a  $\lambda'$  value is chosen outside the desired domain, then the entire move is rejected. Next, we randomly choose a “shooting point”  $m \in [0, M]$  along the trajectory with uniform probability. From this point, we modify the time-reversed random numbers from that point backward to the beginning of the trajectory,  $\tilde{\xi}^{j+\frac{1}{2}} \rightarrow \tilde{\xi}^{j+\frac{1}{2}}$  and  $\tilde{\xi}^{j+1} \rightarrow \tilde{\xi}^{j+1}$  for  $j = 0, 1, \dots, m - 1$ . Similarly, we modify the forward time random numbers from that point to the end of the trajectory,  $\xi^{j+\frac{1}{2}} \rightarrow \xi^{j+\frac{1}{2}}$  and  $\xi^{j+1} \rightarrow \xi^{j+1}$  for  $j = m, m + 1, \dots, M - 1$ . We generate trial move noises by a linear combination of the old noise and a new one,  $\xi' = \alpha \xi + \sqrt{1 - \alpha^2} \zeta$  and  $\tilde{\xi}' = \alpha \tilde{\xi} + \sqrt{1 - \alpha^2} \tilde{\zeta}$ <sup>34,47</sup>. Here,  $\zeta$  and  $\tilde{\zeta}$  are the new noises, drawn from the same zero mean,  $m_i(1 - e^{-\frac{\gamma \Delta t}{m_i}})/\beta$  variance Gaussian distribution that the Langevin integrator noises naturally sample. The parameter  $\alpha$  allows us to control the decorrelation between the current and trial trajectory. Starting from the random point  $m$ , we integrate the trajectories backward in time using the proposed backward random numbers  $\tilde{\xi}'$  and forward in time using the proposed forward random numbers  $\xi'$  to generate the trial trajectory  $\vec{\mathbf{x}}'$ . The resulting trajectory can then be converted to its equivalent complete forward and reverse time representations as random numbers,  $\xi'$  and  $\tilde{\xi}'$ , respectively, using the form of the integrator in Eq. (A1).

The reversed proposal move,  $\vec{\mathbf{x}}', \lambda' \rightarrow \vec{\mathbf{x}}, \lambda$ , occurs when  $\lambda' \rightarrow \lambda$  is generated, the same shooting point is chosen, and  $\zeta' = (\zeta - \alpha \xi')/\sqrt{1 - \alpha^2}$  and  $\tilde{\zeta}' = (\tilde{\zeta} - \alpha \tilde{\xi}')/\sqrt{1 - \alpha^2}$  are chosen from a Gaussian distribution to map from  $\xi'$  back to  $\xi$ . The relative probability of the forward move to the reverse is thus

$$\begin{aligned} & \frac{P_{\text{gen}}(\vec{\mathbf{x}}, \lambda \rightarrow \vec{\mathbf{x}}', \lambda')}{P_{\text{gen}}(\vec{\mathbf{x}}', \lambda' \rightarrow \vec{\mathbf{x}}, \lambda)} \\ &= \exp \left[ \sum_{i=1}^N \frac{\beta}{2m_i(1 - e^{-\frac{\gamma \Delta t}{m_i}})} \left( \sum_{j=0}^{m-1} \left[ (\tilde{\zeta}_i^{j+\frac{1}{2}})^2 + (\tilde{\zeta}_i^{j+1})^2 - (\tilde{\zeta}_i^{j+\frac{1}{2}})^2 - (\tilde{\zeta}_i^{j+1})^2 \right] \right. \right. \\ & \quad \left. \left. + \sum_{j=m}^{M-1} \left[ (\zeta_i^{j+\frac{1}{2}})^2 + (\zeta_i^{j+1})^2 - (\zeta_i^{j+\frac{1}{2}})^2 - (\zeta_i^{j+1})^2 \right] \right) \right], \end{aligned} \quad (\text{A2})$$

where  $m_i$  is the mass of particle  $i$ . Although this ratio appears to be cumbersome, it is straightforward to compute in terms of all of the noise variables  $\xi$  and  $\tilde{\xi}$ , and it is then used in the acceptance probabilities of Eqs. (11) and (14).

Those acceptance probabilities can be further simplified by recognizing a cancellation in the product of Eq. (A2) and the  $P(\vec{x}'|\lambda', \mathbf{x}'(0))/P(\vec{x}|\lambda, \mathbf{x}(0))$  term of Eqs. (12) and (15). For example, after some algebra, Eq. (11) becomes

$$P_{\text{acc}} = \min \left[ 1, h_A(\mathbf{x}'(0))h_B(\mathbf{x}'(t_{\text{obs}})) \frac{Z_A(\lambda')^{-1} e^{-\beta H(\mathbf{x}'(0); \lambda')}}{Z_A(\lambda)^{-1} e^{-\beta H(\mathbf{x}(0); \lambda)}} \right. \\ \left. \times \exp \left[ \sum_{i=1}^N \frac{\beta}{2m_i \left(1 - e^{-\frac{\gamma \Delta t}{m_i}}\right)} \sum_{j=0}^{m-1} \left[ (\xi_i^{j+\frac{1}{2}})^2 + (\xi_i^{j+1})^2 - (\xi_i^{j+\frac{1}{2}})^2 \right. \right. \right. \\ \left. \left. \left. - (\xi_i^{j+1})^2 + (\tilde{\xi}_i^{j+\frac{1}{2}})^2 + (\tilde{\xi}_i^{j+1})^2 - (\tilde{\xi}_i^{j+\frac{1}{2}})^2 - (\tilde{\xi}_i^{j+1})^2 \right] \right] \right]. \quad (\text{A3})$$

After the algebraic simplification, only contributions from the first  $m - 1$  steps of the trajectory remain in Eq. (A3). Even this simplified expression looks daunting, but is easily computed by storing the random numbers of the current and trial trajectories. The long sum of squares of  $\xi$  terms has the physical interpretation of a heat flow between the system and the thermostat.<sup>43</sup> We want to weigh the trajectories based on their probability of occurring from forward-time integration, but we generated a portion of the trajectory ( $j = 0$  to  $j = m - 1$ ) from reversed-time integration. To compute that trajectory's likelihood in the forward-time trajectory ensemble, we must reweight by an exponential of the heat. We also note that because trial trajectories are always generated from previous successful trajectories, the terms  $h_A(\mathbf{x}(0))$  and  $h_B(\mathbf{x}(t_{\text{obs}}))$  are always 1.

For the simple Lennard-Jones binding system of Secs. III B and III C, we used  $\alpha = 0.99$  and  $M = 1000$  with a time step of  $\Delta t = 0.005$  for an observation time of  $t_{\text{obs}} = M\Delta t = 5$ . For the particle escaping from a tetrahedral cage of Sec. III D, we used  $\alpha = 0.9$  and  $M = 1000$ , again with a time step of  $\Delta t = 0.005$  for an observation time of  $t_{\text{obs}} = 5$ . For both systems, we built an initial trajectory by interpolating a starting state at the potential energy minima with Boltzmann momenta and an ending state just inside our respective definitions of the product state. To build a starting trajectory, one could run natural dynamics with an integrator until a suitable trajectory is isolated or do linear interpolation between a starting and ending point, as we have done. This linear interpolation leads to an unphysical starting trajectory, but sampling quickly moves away from it to more natural trajectories.

When sampling with multiple independent trajectories in order to bias more severely toward faster rates, our overall acceptance ratio is a product of individual ratios with the form of Eq. (A3). For example, when the exact partition function is known, the acceptance probability using  $L$  trajectories  $\vec{x}'_l$  with  $l = 1, 2, \dots, L$  is

$$P_{\text{acc}} = \min \left[ 1, \frac{Z_A(\lambda')^{-L}}{Z_A(\lambda)^{-L}} \prod_{l=1}^L h_A(\mathbf{x}'_l(0))h_B(\mathbf{x}'_l(t_{\text{obs}})) \frac{e^{-\beta H(\mathbf{x}'_l(0); \lambda')}}{e^{-\beta H(\mathbf{x}_l(0); \lambda)}} \right. \\ \left. \times \exp \left[ \sum_{i=1}^N \frac{\beta}{2m_i \left(1 - e^{-\frac{\gamma \Delta t}{m_i}}\right)} \sum_{j=0}^{m-1} \left[ (\xi_{i,l}^{j+\frac{1}{2}})^2 + (\xi_{i,l}^{j+1})^2 - (\xi_{i,l}^{j+\frac{1}{2}})^2 \right. \right. \right. \\ \left. \left. \left. - (\xi_{i,l}^{j+1})^2 + (\tilde{\xi}_{i,l}^{j+\frac{1}{2}})^2 + (\tilde{\xi}_{i,l}^{j+1})^2 - (\tilde{\xi}_{i,l}^{j+\frac{1}{2}})^2 - (\tilde{\xi}_{i,l}^{j+1})^2 \right] \right] \right]. \quad (\text{A4})$$

When those partition functions are not known, they must be estimated as discussed in Sec. II E. In either case, the  $L$  trajectories share the same length and parameter set  $\lambda$ , but they vary in their independent starting configurations and in their noises  $\xi_i$ .

## APPENDIX B: ROULETTE PROCEDURE FOR PRODUCING UNBIASED SERIES ESTIMATES

To obtain unbiased estimates of the partition function, we must truncate the infinite series

$$S = 1 + a^{(1)}(1 + a^{(2)}(1 + \dots)) = 1 + \sum_{i=1}^{\infty} \prod_{j=1}^i a^{(j)} \quad (\text{B1})$$

using a roulette procedure, adapted from Ref. 33. We introduce a roulette parameter  $R$  and calculate individual values of  $a^{(n)}$  sequentially, as described in Sec. II C. For the  $n$ th term in the expansion, if the running product  $\Pi^{(n)}$  of Eq. (27) is less than the parameter  $R$ , then a roulette game is played with the series. The series will continue with survival probability  $q^{(n)} = \Pi^{(n)}/R$  and truncate with probability  $1 - q^{(n)}$ . If the series continues, then we scale  $a^{(n)}$  by the survival probability  $q^{(n)}$  and continue the procedure for sample  $n + 1$ . If the series truncates, then  $a^{(n)} \rightarrow 0$  and the estimate of the series is complete. Alternatively, when the running product is not less than  $R$ , the series continues without scaling  $a^{(n)}$ . We compactly express the various cases by noting that sample  $n$  survives with probability  $q^{(n)} = \min[1, \Pi^{(n)}/R]$  in which case it contributes the scaled contribution  $a^{(n)} \max[1, R/\Pi^{(n)}]$ .

As stated in the main text, the threshold for stochastic truncation after term  $n$ ,  $\Pi^{(n)}$ , is computed recursively as the running product of the absolute value of these scaled contributions,

$$\Pi^{(n)} = \left| a^{(n)} \prod_{i=1}^{n-1} \left| a^{(i)} \right| \max \left[ 1, \frac{R}{\Pi^{(i)}} \right] \right|, \quad (\text{B2})$$

where  $\Pi^{(1)} = |a^{(1)}|$ . Since  $a^{(n)}$  values can be scaled throughout the procedure and because  $a^{(n)} = 0$  if the series is truncated at the  $n$ th term, the infinite series (B1) is replaced by

$$S^{(n)} = 1 + \sum_{i=1}^{n-1} \prod_{j=1}^i \left| a^{(j)} \right| \max \left[ 1, \frac{R}{\Pi^{(j)}} \right]. \quad (\text{B3})$$

We now show that the scaling of terms, which up to now was introduced in an *ad hoc* manner, was constructed such that

the expectation value of the truncated sum will exactly equal the expected value of the infinite sum. Note that the expected value of the truncated series can be expressed as the sum over  $n$  of the probability

of reaching the  $n$ th term times the value of the truncated sum  $S^{(n)}$ . The algebra of that expectation value simplifies due to telescoping sums to give

$$\begin{aligned}
 \langle S \rangle &= (1 - q^{(1)})S^{(1)} + (1 - q^{(2)})q^{(1)}S^{(2)} + (1 - q^{(3)})q^{(2)}q^{(1)}S^{(3)} + \dots \\
 &= (1 - q^{(1)}) + (1 - q^{(2)})q^{(1)} \left( 1 + a^{(1)} \max \left[ 1, \frac{R}{\Pi^{(1)}} \right] \right) \\
 &\quad + (1 - q^{(3)})q^{(1)}q^{(2)} \left( 1 + a^{(1)} \max \left[ 1, \frac{R}{\Pi^{(1)}} \right] + a^{(1)}a^{(2)} \max \left[ 1, \frac{R}{\Pi^{(2)}} \right] \right) + \dots \\
 &= 1 + q^{(1)}a^{(1)} \max \left[ 1, \frac{R}{\Pi^{(1)}} \right] + q^{(1)}q^{(2)}a^{(1)}a^{(2)} \max \left[ 1, \frac{R}{\Pi^{(1)}} \right] \max \left[ 1, \frac{R}{\Pi^{(2)}} \right] + \dots \\
 &= 1 + \sum_{i=1}^{\infty} \prod_{j=1}^i a^{(j)} \min \left[ 1, \frac{\Pi^{(j)}}{R} \right] \max \left[ 1, \frac{R}{\Pi^{(j)}} \right] \\
 &= 1 + \sum_{i=1}^{\infty} \prod_{j=1}^i a^{(j)}.
 \end{aligned} \tag{B4}$$

The final equality follows because  $R > 0$ ,  $\Pi^{(i)} > 0$ , and for positive  $x$ ,  $\min[1, x] \max[1, x^{-1}] = 1$ . Consequently, we see that the expectation value of the truncated sums [Eq. (B4)] equals that of the infinite sum [Eq. (B1)]. In other words, the stochastic truncation scheme is unbiased. We note that the roulette procedure we have described is not a unique way to generate an unbiased stochastic truncation. It may be possible to design alternative roulette games that give a better trade-off between truncation speed and estimate noise.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in <http://doi.org/10.5281/zenodo.4150012>.<sup>45</sup>

## REFERENCES

- H. A. Kramers, *Physica* **7**, 284 (1940).
- P. Hänggi, P. Talkner, and M. Borkovec, *Rev. Mod. Phys.* **62**, 251 (1990).
- P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, *Annu. Rev. Phys. Chem.* **53**, 291 (2002).
- B. Peters, *Mol. Simul.* **36**, 1265 (2010).
- T. Yamamoto, *J. Chem. Phys.* **33**, 281 (1960).
- W. H. Miller, S. D. Schwartz, and J. W. Tromp, *J. Chem. Phys.* **79**, 4889 (1983).
- D. Chandler, *J. Chem. Phys.* **68**, 2959 (1978).
- J. A. Montgomery, Jr., D. Chandler, and B. J. Berne, *J. Chem. Phys.* **70**, 4056 (1979).
- D. Chandler, *J. Stat. Phys.* **42**, 49 (1986).
- D. Chandler and R. A. Kuharski, *Faraday Discuss. Chem. Soc.* **85**, 329 (1988).
- M. Borkovec and P. Talkner, *J. Chem. Phys.* **92**, 5307 (1990).
- G. Hummer, *J. Chem. Phys.* **120**, 516 (2004).
- L. R. Pratt, *J. Chem. Phys.* **85**, 5045 (1986).
- C. Dellago, P. Bolhuis, and P. L. Geissler, *Adv. Chem. Phys.* **123**, 1 (2002).
- P. G. Bolhuis, *J. Phys.: Condens. Matter* **15**, S113 (2002).
- C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, *J. Chem. Phys.* **108**, 1964 (1998).
- P. G. Bolhuis, C. Dellago, and D. Chandler, *Faraday Discuss.* **110**, 421 (1998).
- C. Dellago, P. G. Bolhuis, and D. Chandler, *J. Chem. Phys.* **110**, 6617 (1999).
- B. Peters and B. L. Trout, *J. Chem. Phys.* **125**, 054108 (2006).
- B. Peters, G. T. Beckham, and B. L. Trout, *J. Chem. Phys.* **127**, 034109 (2007).
- T. F. Miller III and C. Predescu, *J. Chem. Phys.* **126**, 144102 (2007).
- M. Grünwald, C. Dellago, and P. L. Geissler, *J. Chem. Phys.* **129**, 194101 (2008).
- T. S. van Erp, D. Moroni, and P. G. Bolhuis, *J. Chem. Phys.* **118**, 7762 (2003).
- D. Moroni, P. G. Bolhuis, and T. S. van Erp, *J. Chem. Phys.* **120**, 4055 (2004).
- T. S. van Erp and P. G. Bolhuis, *J. Comput. Phys.* **205**, 157 (2005).
- R. J. Allen, P. B. Warren, and P. R. ten Wolde, *Phys. Rev. Lett.* **94**, 018104 (2005).
- R. J. Allen, D. Frenkel, and P. R. ten Wolde, *J. Chem. Phys.* **124**, 024102 (2006).
- R. J. Allen, D. Frenkel, and P. R. ten Wolde, *J. Chem. Phys.* **124**, 194111 (2006).
- R. J. Allen, C. Valeriani, and P. R. ten Wolde, *J. Phys.: Condens. Matter* **21**, 463102 (2009).
- E. E. Borrero and F. A. Escobedo, *J. Chem. Phys.* **129**, 024115 (2008).
- E. E. Borrero and F. A. Escobedo, *J. Phys. Chem. B* **113**, 6434 (2009).
- D. M. Ceperley and M. Dewing, *J. Chem. Phys.* **110**, 9812 (1999).
- T. E. Booth, *Nucl. Sci. Eng.* **156**, 403 (2007).
- T. R. Gingrich and P. L. Geissler, *J. Chem. Phys.* **142**, 234104 (2015).
- P. Diaconis, S. Holmes, and R. M. Neal, *Ann. Appl. Probab.* **10**, 726 (2000).
- L. Lin, K. Liu, and J. Sloan, *Phys. Rev. D* **61**, 074505 (2000).
- C. Andrieu and G. O. Roberts, *Ann. Stat.* **37**, 697 (2009).
- E. P. Bernard, W. Krauth, and D. B. Wilson, *Phys. Rev. E* **80**, 056704 (2009).
- M. Vucelja, *Am. J. Phys.* **84**, 958 (2016).
- A.-M. Lyne, M. Girolami, Y. Atchadé, H. Strathmann, and D. Simpson, *Stat. Sci.* **30**, 443 (2015).
- G. Bhanot and A. D. Kennedy, *Phys. Lett. B* **157**, 70 (1985).
- T. R. Gingrich, G. M. Rotskoff, G. E. Crooks, and P. L. Geissler, *Proc. Natl. Acad. Sci. U. S. A.* **113**, 10263 (2016).
- M. Athènes and G. Adjanor, *J. Chem. Phys.* **129**, 024116 (2008).
- M. R. Shirts and J. D. Chodera, *J. Chem. Phys.* **129**, 124105 (2008).
- A. Albaugh and T. R. Gingrich (2020). "Estimating reciprocal partition functions to enable design space sampling." Zenodo, Dataset. <https://doi.org/10.5281/zenodo.4150011>
- G. E. Crooks and D. Chandler, *Phys. Rev. E* **64**, 026109 (2001).
- G. Stoltz, *J. Comput. Phys.* **225**, 491 (2007).