arXiv:2008.06334v1 [physics.data-an] 11 Aug 2020

Learning with rare data: Using active importance sampling to optimize objectives dominated by rare events

Grant M. Rotskoff
Dept. of Chemistry, Stanford University, Stanford, CA 94305[*]

Eric Vanden-Eijnden
Courant Institute, New York University, New York, NY 10012[†]

Deep neural networks, when optimized with sufficient data, provide accurate representations of high-dimensional functions; in contrast, function approximation techniques that have predominated in scientific computing do not scale well with dimensionality. As a result, many high-dimensional sampling and approximation problems once thought intractable are being revisited through the lens of machine learning. While the promise of unparalleled accuracy may suggest a renaissance for applications that require parameterizing representations of complex systems, in many applications gathering sufficient data to develop such a representation remains a significant challenge. Here we introduce an approach that combines rare events sampling techniques with neural network optimization to optimize objective functions that are dominated by rare events. We show that importance sampling reduces the asymptotic variance of the solution to a learning problem, suggesting benefits for generalization. We study our algorithm in the context of learning dynamical transition pathways between two states of a system, a problem with applications in statistical physics and implications in machine learning theory. Our numerical experiments demonstrate that we can successfully learn even with the compounding difficulties of high-dimension and rare data.

I. RARE EVENTS AND IMPORTANCE SAMPLING

Deep neural networks (DNNs) have become an essential tool for a diverse set of problems in data science and, increasingly, the physical sciences. Underlying the impressive breadth of applications are the uncommonly robust approximation properties of DNNs, especially in high-dimensional settings where standard tools from numerical analysis break down. It has not gone without notice that many compelling questions in statistical physics require precise knowledge of high-dimensional functions [1] which can be challenging to represent and compute, suggesting that machine learning may have a transformative role to play.

Sampling is at the core of a large class of problems in both statistical mechanics and machine learning but is often rendered intractable by the need for precise knowledge about where and how to sample. Computing the expectation of a physical observable $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}$ requires making an empirical expectation of a Gibbs' distribution

$$\mathbb{E}\mathcal{A} = Z^{-1} \int_{\mathbb{R}^d} \mathcal{A}(\boldsymbol{x}) e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x}, \tag{1}$$

where $Z = \int_{\mathbb{R}^d} e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x}$, β is interpreted as the inverse temperature and V is the potential energy of the system, which we assume is known analytically and can be evaluated efficiently, a common setting for problems in classical statistical mechanics. Finally, Z is the normalization constant, also known as the partition function—it cannot be evaluated explicitly except in the most trivial cases. The seemingly basic task of computing expectations underlies many problems, for example, computing reaction rates and identifying physically important ”transition” states in molecular systems. Nevertheless, estimating (1) remains perhaps *the* central challenge for modeling and studying complex chemical and biophysical systems.

Here we focus on the particular problem of computing the statistics of transitions between two metastable states of a high-dimensional probably distribution. This problem is motivated by chemical reactions or other molecular

[*] rotskoff@stanford.edu; code: https://github.com/rotskoff-group/learning-committor
[†] eve2@cims.nyu.edu

transitions but has much broader relevance. In particular, with the resurgence of stochastic sampling schemes in the context of optimization [2–4], the approach we describe here can be used to study dynamical properties of sampling schemes in the nonconvex optimization setting.

The potential energy functions and free energy surfaces that arise in applications in machine learning, chemistry, and biophysics are typically highly nonconvex and may have hundreds of thousands of degrees of freedom. While it should be emphasized that, when sampling to compute expectations, the goal is not to identify the global minimum of the free energy, but rather to evaluate (1), such systems still present enormous computational challenges. The algorithms typically employed in Markov Chain Monte Carlo sampling schemes rely on local updates to the configuration $\boldsymbol{x}$, in turn leading to slow mixing between metastable states of the distribution. This leads to difficulty in estimating (1) when there is metastability, as the transition time between two states is exponential in the height of the free energy barrier between them [5, 6].

Many techniques have been proposed to overcome the inherent complexity of this task (cf. [6, 7]). Fundamentally, most of these techniques require the introduction of a "collective variables" map which reduces the high-dimensional sampling problem to a much lower dimensional setting. When the dimensionality is significantly reduced, computing expectations becomes tractable by using importance sampling techniques that require discretizing space. However, there is no *a priori* guarantee that the chosen collective variables map projects the original space without loss of important information. The approach we take here avoids the complex task of designing an appropriate collective variables map by using a neural network to parameterize a map from the full configuration space.

Importance sampling and other variance reduction techniques have appeared in a variety of contexts in machine learning. Csiba and Richtarik [8] described and analyzed an algorithm that does importance sampling of the training set to adaptively select minibatches and accelerate gradient descent. Their work formalizes an approach, represented in a large body of work [9–11], to aims reduce to the variance in the gradients when optimizing using stochastic gradient descent. In a separate line of inquiry, Fan *et al.* [12] uses importance sampling to perform approximate Bayesian inference in continuous time Bayesian networks. Our setting differs substantially from these works, as we are principally concerned with problems in which the data set is sampled on-the-fly from a Boltzmann distribution. Furthermore, we require importance sampling for the learning to be tractable at all, whereas the aforementioned works seek to accelerate optimization in otherwise tractable learning problems. Our theoretical results suggest that these previously studied approaches benefit generalization.

Identifying reaction pathways and sampling reactive trajectories is a central problem in statistical mechanics, with decades of work. Transition path sampling methods are perhaps the most closely related to our approach [13, 14]. Our applications are heavily influenced by the perspective of potential theory [15] and the notion of the committor function (discussed in detail below) [5, 16]. Khoo *et al.* [17] first considered the problem of learning committor functions from the perspective of solving high-dimensional PDEs but did not address the sampling issues that can arise in computing the objective. Our work most closely follows that of Li *et al.* [18], who also examined the problem of optimizing a representation of the committor using neural networks on low-dimensional landscapes. Our work extends this approach in several important ways: first, our algorithm is an *active* approach—the importance sampling directly uses the committor function meaning that there is feedback between the optimization and the data collection. In high-dimensional systems in which selecting a reaction coordinate presents a challenging design problem, our approach is crucial for effective sampling because we avoid explicitly constructing a reaction coordinate.

II. LOSS FUNCTIONS FOR REACTION PATH DYNAMICS DOMINATED BY RARE EVENTS

In many problems in chemical physics, we hope to calculate the probability a physical system to make a spontaneous excursion from a configuration in phase space $\hat{A} \subset \mathbb{R}^d \times \mathbb{R}^d$ to a distinct state in $\hat{B} \subset \mathbb{R}^d \times \mathbb{R}^d$. In general, it is not possible to directly compute these transition probabilities. Two confounding factors are at play: first, the state space may very high dimensional in nontrivial cases; secondly, the computational resources required to observe a transition from $\hat{A}$ to $\hat{B}$ in dynamical simulations are prohibitive when the transitions are infrequent. One classic approach to this problem relies on the Markovian nature of the dynamics. It can be shown that, for a Markovian dynamics, the infinitesimal generator has a "spectral gap" between a set of low-lying real eigenvalues associated with transitions between metastable states [15, 20]. However, in complex systems, there still may be hundreds or thousands of metastable states, and solving the associated eigenvalue problem for a given transition is neither tractable in practice nor very informative [21].

In the case that $\hat{A}$ and $\hat{B}$ can be specified, an alternative is to quantify the probability that a trajectory will make a dynamical transition from the metastable state $\hat{A}$ to another $\hat{B}$ by computing the "committor function" $\hat{q} : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$, which gives the probability that a trajectory starting at $(\boldsymbol{x}, \boldsymbol{v})$ first reaches $\hat{B}$ before $\hat{A}$:

$$\hat{q}(\boldsymbol{x}, \boldsymbol{v}) := \mathbb{P}^{(\boldsymbol{x},\boldsymbol{v})}(t_{\hat{B}} < t_{\hat{A}}). \tag{2}$$

We consider this problem in the following setting: we take a physical system with coordinates $(\boldsymbol{x}, \boldsymbol{v}) \in \mathbb{R}^d \times \mathbb{R}^d$ whose evolution is governed by the Langevin equation

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = -\nabla V(\boldsymbol{x}) - \gamma \boldsymbol{v} + \sqrt{2\beta^{-1}}\boldsymbol{\eta}. \end{cases} \tag{3}$$

Here $V : \mathbb{R}^d \to [0, \infty)$ is a potential energy function, $\beta > 0$, which controls the magnitude of the fluctuations, is typically interpreted as the inverse temperature in physical systems, γ is a friction coefficient, and $\boldsymbol{W}_t$ is a Wiener process. This dynamics is ubiquitously used to model molecular dynamics in the condensed phase but has also been proposed as a heuristic model for stochastic optimization methods like SGD [19] and sampling-based optimization schemes [2].

We can write a partial differential equation (PDE) for the probability $q(\boldsymbol{x}, \boldsymbol{v})$ that a trajectory under the dynamics (3) that starts at $(\boldsymbol{x}, \boldsymbol{v})$ reaches $\hat{B}$ before $\hat{A}$. This is the following backward Kolmogorov equation [5]

$$\begin{cases} L\hat{q} = 0 & \text{for } (\boldsymbol{x}, \boldsymbol{v}) \notin \hat{A} \cup \hat{B} \\ \hat{q}(\boldsymbol{x}, \boldsymbol{v}) = 0 & \text{for } (\boldsymbol{x}, \boldsymbol{v}) \in \hat{A} \\ \hat{q}(\boldsymbol{x}, \boldsymbol{v}) = 1 & \text{for } (\boldsymbol{x}, \boldsymbol{v}) \in \hat{B}. \end{cases} \tag{4}$$

where L is the infinitesimal generator of the process defined by (3):

$$Lf = \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f - \nabla_{\boldsymbol{x}} V \cdot \nabla_{\boldsymbol{v}} f - \gamma \boldsymbol{v} \cdot \nabla_{\boldsymbol{v}} f + \beta^{-1} \Delta_{\boldsymbol{v}} f. \tag{5}$$

The probability $\hat{q} : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ is known as the committor function because its level sets are surfaces of constant probability to reach $\hat{B}$. In the context of chemical reaction dynamics, the backward Kolmogorov equation is a PDE in very high dimension, meaning that is not possible to obtain a solution analytically or using traditional numerical methods based e.g. on finite elements.

A. Defining a loss function for learning the committor

Our goal is to define a parametric representation of the committor function and an objective function that enables us to optimize the parameters. Neural networks offer flexibility to the representation and relative ease of optimization, making this a natural choice for a representation of the committor. While solving the BKE is intractable, the committor satisfies a straightforward variational principle that can be employed as an objective function. Using (4), the committor function satisfies

$$\hat{C}[\hat{q}] = \hat{Z}^{-1} \int_{\mathbb{R}^d \times \mathbb{R}^d} |L\hat{q}(\boldsymbol{x}, \boldsymbol{v})|^2 e^{-\beta \mathcal{H}(\boldsymbol{x}, \boldsymbol{v})} d\boldsymbol{x} d\boldsymbol{v} = 0 \tag{6}$$

where $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = \frac{1}{2}|\boldsymbol{v}|^2 + V(\boldsymbol{x})$ is the Hamiltonian and $\hat{Z} = \int_{\mathbb{R}^d \times \mathbb{R}^d} e^{-\beta \mathcal{H}(\boldsymbol{x}, \boldsymbol{v})} d\boldsymbol{x} d\boldsymbol{v}$ is the partition function, here included to interpret $\hat{C}[\hat{q}]$ as the canonical expectation of $|L\hat{q}(\boldsymbol{x}, \boldsymbol{v})|^2$. Hence, to find $\hat{q}$ we can minimize $\hat{C}[\hat{q}]$ over all functions satisfying the boundary conditions in (4),

$$\inf_{\hat{q}} \hat{C}[\hat{q}] \quad \text{with} \quad \hat{q} = 0 \ \text{ in } \hat{A}, \quad \hat{q} = 1 \ \text{ in } \hat{B} \tag{7}$$

Integrating over the momenta we reduce $\hat{C}[\hat{q}]$ to an objective function for $q(\boldsymbol{x})$:

$$\inf_{q} C[q] \quad \text{subject to} \quad q = 0 \ \text{ in } A, \quad q = 1 \ \text{ in } B \tag{8}$$

where

$$C[q] = \int_{\mathbb{R}^d} |\nabla q(\boldsymbol{x})|^2 d\mu(\boldsymbol{x}) \qquad \text{with} \quad d\mu(\boldsymbol{x}) = Z^{-1} e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \tag{9}$$

and the reactant state $A \in \mathbb{R}^d$ and the product state $B \in \mathbb{R}^d$ are now sets defined in configuration space.

In the optimization procedure below, it is more tractable to penalize deviations from the boundary conditions rather than impose them as constraints. Consequently, we use Lagrange multipliers to ensure that the committor has the right values on the initial and target states. The objective function we use is thus

$$C_\lambda[q] = \int_{\mathbb{R}^d} |\nabla q(\boldsymbol{x})|^2 d\mu(\boldsymbol{x}) + \lambda \int_A |q(\boldsymbol{x})|^2 d\mu(\boldsymbol{x}) + \lambda \int_B |1 - q(\boldsymbol{x})|^2 d\mu(\boldsymbol{x}). \tag{10}$$

Using a neural network representation of the committor with parameter set $\{\boldsymbol{\theta}\}_{i=1}^n$, the problem becomes to minimize C_λ over this set. We describe an alternative formulation of the committor (cf. [22]) in Appendix C which can be solved with distinct boundary conditions. In Appendix D we discuss how to use collective variables maps.

III. IMPORTANCE SAMPLING IMPROVES GENERALIZATION ERROR

The loss function (10) requires data that may be exponentially rarer than the metastable states A and B. To estimate it efficiently, straightforward Monte Carlo sampling does not suffice. In practice, the importance sampling procedure we outline below reduces the variance of the estimator by ensuring that regions of $\mathbb{R}^d$ with low probability are well-sampled. Importantly, this reduction in variance benefits the generalization error. We prove two results that demonstrate the utility of importance sampling. First, consider the empirical risk minimization (ERM) problem

$$\boldsymbol{\theta}_M = \text{argmin}_{\boldsymbol{\theta}} M^{-1} \sum_{m=1}^{M} \ell(\boldsymbol{x}_m, \boldsymbol{\theta}) \equiv \text{argmin}_{\boldsymbol{\theta}} L_M(\boldsymbol{\theta}) \qquad \boldsymbol{x}_i \sim \mu \tag{11}$$

which we would like to compare to minimizer of the population risk

$$\boldsymbol{\theta}_* = \text{argmin}_{\boldsymbol{\theta}} \int_{\mathbb{R}^d} \ell(\boldsymbol{x}, \boldsymbol{\theta}) d\mu(\boldsymbol{x}) \equiv \text{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{12}$$

Importance sampling reduces the variance of our estimate for the population risk, L. This has implications for the expected value of the minimizer; specifically, variance reduction in the empirical loss reduces the asymptotic variance of the generalization error. Specifically, we prove

Proposition III.1. *Denote by $\boldsymbol{\theta}_M$ the ERM obtained from canonical sampling and $\tilde{\boldsymbol{\theta}}_M$ the ERM obtained from importance sampling. If*

$$\forall \boldsymbol{\theta} \quad var(\nabla \tilde{L}_M) \leq var(\nabla L_M) \tag{13}$$

and $\nabla\nabla L_M$ and $\nabla\nabla \tilde{L}_M$ are PSD (which can be guaranteed with regularization), then

$$\mathbb{E}_\mu(\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*)^2 \leq \mathbb{E}_\mu(\boldsymbol{\theta}_M - \boldsymbol{\theta}_*)^2 \tag{14}$$

The proof is given in Appendix A.

The assumptions we make above are not guaranteed to hold in the overparameterized regime, where local strong convexity near the minimizer may be difficult to ensure. This in mind, we also prove that importance sampling improves an upper bound on the asymptotic variance of the loss at convergence in the overparameterized regime. Our result relies on the mean-field perspective for neural networks, in which we take limit $n \to \infty$ where n is the number of parameters [23–26]. In this setting, we write the function representation

$$q(\boldsymbol{x}) = \sigma\left(\int_D \varphi(\boldsymbol{x}, \boldsymbol{z}) d\gamma(\boldsymbol{z})\right) \tag{15}$$

where γ is a signed Radon measure on the space of parameters for the nonlinearity φ and σ is a sigmoidal function. Using this representation, in Appendix A we prove,

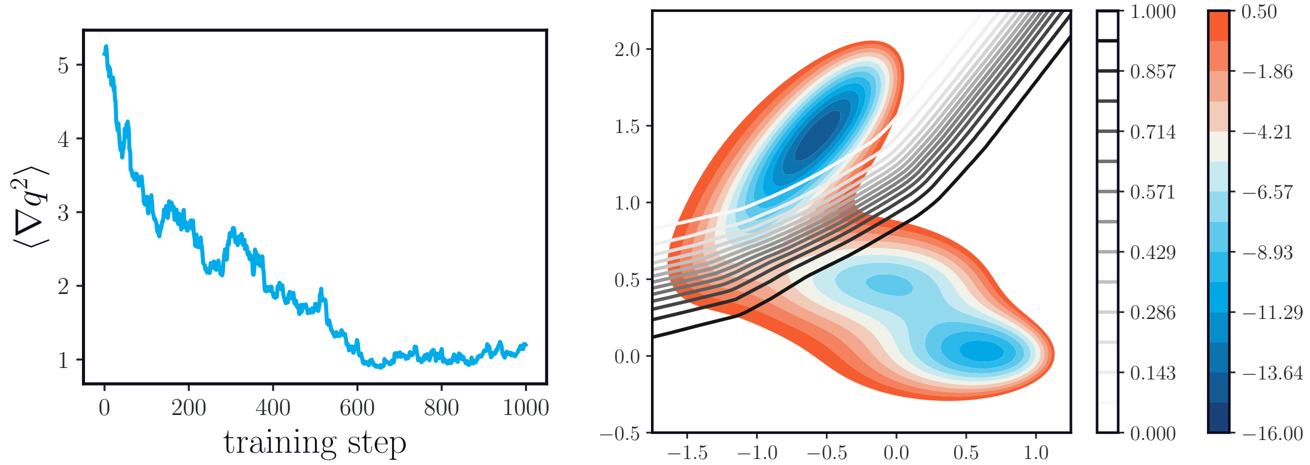

Figure 1. Simple illustrative experiment; the potential energy function is a 2D mixture of Gaussians and the committor is represented as a single hidden layer neural network. Left: decay of the variational loss function as a function of training time for the Müller-Brown potential (26), angle brackets denote expectation with respect to the associated Gibbs measure. Right: the potential function is shown as a contour plot. The isocommittor lines are shown from white to black. Notably, level set $q = 0.5$ coincides with the saddle, as expected.

Proposition III.2. *Let $\mathcal{L}[\gamma]$ denote the convex population risk as a functional of the signed Radon measure γ where q is represented as (15); let γ_* be the minimizer. $\mathcal{L}_P$ denotes the corresponding convex empirical risk functional with minimizer γ_P. Denote by $D_\gamma \mathcal{F}$ the functional derivative of a functional $\mathcal{F}$ with respect to γ. Then,*

$$\mathcal{L}[\gamma_P] \leq \\ 4 \sup_{\boldsymbol{z},\boldsymbol{z}',t} \| D_\gamma^2 \Delta \mathcal{L}_P(\boldsymbol{z}, \boldsymbol{z}', \gamma_* + t(\gamma_P - \gamma_*)) \| |\gamma_*|_{\mathrm{TV}}^2, \tag{16}$$

where $\Delta \mathcal{L}_P = \mathcal{L}[\gamma] - \mathcal{L}_P[\gamma]$ and $|\gamma|_{\mathrm{TV}}$ denotes the total variation norm of the measure. This bound is tightened by importance sampling because the variance term is diminished.

This result is perhaps best illustrated in the case where $q(\boldsymbol{x})$ does not have the sigmoid function and we take the mean-squared error as the loss. In this case,

$$D_\gamma^2 \Delta \mathcal{L}_P(\boldsymbol{z}, \boldsymbol{z}', \gamma) = \frac{1}{P} \sum_{p=1}^{P} \nabla \varphi(\boldsymbol{x}_p, \boldsymbol{z}) \cdot \nabla \varphi(\boldsymbol{x}_p, \boldsymbol{z}'); \tag{17}$$

note that there is no dependence on γ. This expression emphasizes that lower variance in the empirical expectation over μ will improve the bound.

IV. ALGORITHMS: EMPIRICAL LOSS AND IMPORTANCE SAMPLING

In practice we need to estimate the canonical expectations $\langle \cdot \rangle_\beta$ in (10) using some data set: that is, we need to replace the population loss (10) by an empirical loss. The simplest way to proceed is to replace the canonical expectation of any function $f(\boldsymbol{x})$ by its empirical average over a set of points $\{\boldsymbol{x}_m\}_{m=1}^M$ sampled from $Z^{-1} e^{-\beta V(\boldsymbol{x})}$, i.e. use

$$\mathbb{E}_\mu f \approx \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{x}_m), \qquad \boldsymbol{x}_m \sim \mu \tag{18}$$

This procedure gives an empirical loss that is an unbiased estimator of the population loss and converges to it almost surely as $M \to \infty$. However, the empirical loss constructed this way is a poor estimator of the population loss in general, in the sense that its variance is large compared to the square of its mean. The reason is that the expectation $\langle |\nabla q|^2 \rangle_\beta$ is dominated by configurations that are unlikely for $Z^{-1}e^{-\beta V(\boldsymbol{x})}$. This is evident if we consider a double-well potential, in which the barrier between the two minima of the potential is sampled exponentially rarely in the height of the barrier.

To bypass this difficulty we will estimate the loss via importance sampling. Given any set of nonnegative functions $W_l(\boldsymbol{x}) \geq 0$ with $l = 1, \dots, L$ such that

$$\forall \boldsymbol{x} \in \mathbb{R}^d \quad : \quad \sum_{l=1}^{L} W_l(\boldsymbol{x}) = 1, \tag{19}$$

we can write

$$\mathbb{E}_\mu f = \sum_{l=1}^{L} \int_{\mathbb{R}^d} f(\boldsymbol{x}) W_l(\boldsymbol{x}) d\mu(\boldsymbol{x}) \equiv \sum_{l=1}^{L} \langle f \rangle_l w_l \tag{20}$$

where we defined the expectation

$$\langle f \rangle_l = Z_l^{-1} \int_{\mathbb{R}^d} f(\boldsymbol{x}) W_l(\boldsymbol{x}) d\mu(\boldsymbol{x}) \quad \text{where} \quad Z_l = \int_{\mathbb{R}^d} W_l(\boldsymbol{x}) d\mu(\boldsymbol{x}) \tag{21}$$

as well as the weights

$$w_l = \mathbb{E}_\mu W_l \tag{22}$$

In addition, by using $f(\boldsymbol{x}) = W_{l'}(\boldsymbol{x})$ in this expression, we deduce that the weights satisfy the eigenvalue problem [27]

$$w_{l'} = \sum_{l=1}^{L} w_l p_{ll'}, \quad l' = 1, \dots, L, \quad \text{subject to} \quad \sum_{l=1}^{L} w_l = 1, \tag{23}$$

where we defined

$$p_{ll'} = \langle W_{l'} \rangle_l \tag{24}$$

In practice, we can sample $Z_l^{-1} W_l(\boldsymbol{x}) d\mu(\boldsymbol{x})$ e.g. by Metropolis-Hastings Monte-Carlo on the potential $U(\boldsymbol{x}) - \beta^{-1} \log W_l(\boldsymbol{x})$ and compute expectations in this ensemble as

$$\langle f \rangle_l \approx \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{x}_{m,l}), \qquad \boldsymbol{x}_{m,l} \sim Z_l^{-1} W_l(\boldsymbol{x}) d\mu(\boldsymbol{x}) \tag{25}$$

This allows us to estimate $\langle f \rangle_l$ in (23) as well as $p_{ll'}$ in (24): knowledge of the latter quantity enables us to solve the eigenvalue problem in (23) to find the weights w_l, and finally estimate $\mathbb{E}_\mu f$ via (20). We discuss specific choices of the windowing function in App. B 3.

Putting everything together, in Algorithm IV.1, we describe the most straightforward implementation of our approach. Algorithm IV.1 is sequential; a version in which we evolve $\boldsymbol{x}_{m,l}$ and $\boldsymbol{\theta}$ concurrently would allow for significant wallclock speed-ups.

Algorithm IV.1 (Active Importance Sampled Variational Stochastic Gradient Descent)**.**

```
Input: Energy function $V : \mathbb{R}^d \to \mathbb{R}$, initial $\boldsymbol{\theta}$
while $\langle \nabla_{\boldsymbol{\theta}} \mathcal{C}[q] \rangle > \epsilon_{\text{tol}}$ do
    for $l = 1, \dots, L$ do
```

```
        for $m = 1, \dots, M$ do
            Sample $\boldsymbol{x}_{m,l} \sim Z_L^{-1} e^{-\beta V(\boldsymbol{x})} W_l(\boldsymbol{x})$
        Compute $p_{l,l'} = \frac{1}{M} \sum_{m=1}^{M} W_{l'}(\boldsymbol{x}_{m,l})$ for $l' = 1, \dots, L$
        Compute
            $C_\lambda^l[q] = \frac{1}{M} \sum_{m=1}^{M} \Big( \nabla_{\boldsymbol{\theta}_i} \nabla_{\boldsymbol{x}} q_{m,l} \nabla_{\boldsymbol{x}} q_{m,l} + \lambda q_{m,l} \nabla_{\boldsymbol{\theta}_i} q_{m,l} 1_A(\boldsymbol{x}_{m,l}) - \lambda (1 - q_{m,l}) \nabla_{\boldsymbol{\theta}_i} q_{m,l} 1_B(\boldsymbol{x}_{m,l}) \Big)$
        where we denote $q_{ml} = q(\boldsymbol{x}_{m,l})$.
    Solve (23) for $w_l$, $l = 1, \dots, L$
    Compute $\nabla_{\boldsymbol{\theta}} \mathcal{C}[q] = \frac{1}{L} \sum_{l=1}^{L} C_\lambda^l[q] w_l$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \Delta t \nabla_{\boldsymbol{\theta}} \mathcal{C}[q]$
Return $\boldsymbol{\theta}$
```

V. NUMERICAL EXPERIMENTS

As a proof of concept, we optimize the committor function on the well-studied Müller-Brown potential [28]. We consider the dynamics (3) for a 2D system evolving in a Gaussian mixture potential

$$V_{\mathrm{MB}}(\boldsymbol{x}) = \sum_{k=1}^{4} A_k \exp\left(-(\boldsymbol{x} - \mu_k)^T \Sigma_k^{-1} (\boldsymbol{x} - \mu_k) \right) \tag{26}$$

with

$$\begin{aligned} A &= (-200, -100, -170, 15) \\ \mu &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}, \begin{pmatrix} -0.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \\ \Sigma^{-1} &= \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \begin{pmatrix} 6.5 & -5.5 \\ -5.5 & 6.5 \end{pmatrix}, \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \end{aligned} \tag{27}$$

Our results, shown in Fig. 1, demonstrate the importance sampling is sufficient to decrease the loss of the objective, which we compute along the transition path samples gathered during the optimization.

Unlike standard approaches to computing the committor (e.g., finite elements), the algorithm outlined here also succeeds when the input space is high-dimensional. As a non-trivial test of robustness, we consider the energy functional

$$E[\rho] = \int_{[0,1]^2} \left(\tfrac{1}{2} D |\nabla \rho(\boldsymbol{z})|^2 + \tfrac{1}{4} (1 - |\rho(\boldsymbol{z})|^2)^2 \right) d\boldsymbol{z} \tag{28}$$

and the associated stochastic PDE

$$\partial_t \rho = D \Delta \rho + \rho - \rho^3 + \text{spatio-temporal white noise} \tag{29}$$

If we impose the Dirichlet boundary conditions

$$\rho = +1, \quad \text{for} \quad z_1 = 0, 1, \qquad \rho = -1, \quad \text{for} \quad z_2 = 0, 1, \tag{30}$$

and take both D and the noise amplitude small enough the system display metastability. That is, the equation

$$D \Delta \rho + \rho - \rho^3 = 0 \tag{31}$$

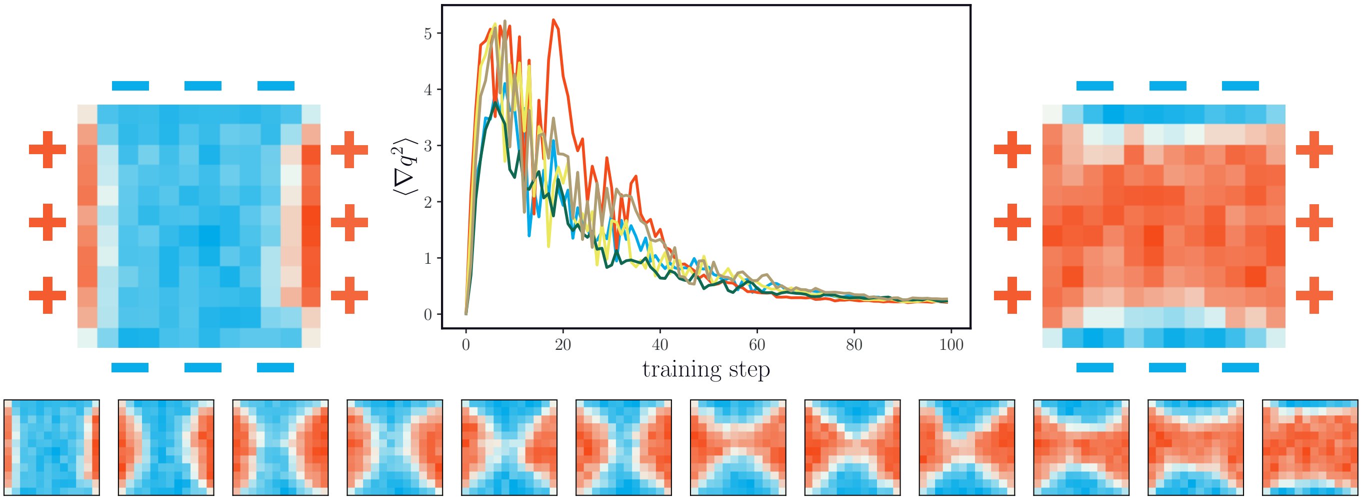

Figure 2. Top left and right: The two metastable solutions of (33) with Dirichlet boundary conditions ($\rho = 1$ at the left and right boundaries, $\rho = -1$ at the top and bottom boundaries). Top center: Decay of the loss of as a function of training step for 10 runs of the optimization with random initial conditions. Bottom: A sample transition path obtained by sampling the biased ensemble with ($q = 0, \dots, 1$). The path shows the characteristic nucleation pathway for a transition between the two metastable states, with the expected hourglass shape.

has two solutions that are stable under the deterministic dynamics: these solution are either mostly $\rho = 1$ in the domain, with boundary layer of size $D^{-1/2}$ near $z_2 = 0, 1$, or mostly $\rho = -1$, with boundary layer of size $D^{-1/2}$ near $z_1 = 0, 1$. These two solutions are depicted in Fig. 2.

If we discretize the problem on a lattice with spacing $h = 1/N$ and introduce

$$\rho_{i,j} = \rho(ih, jh), \qquad i, j = 0, \dots, N \tag{32}$$

we arrive at the SDE

$$d\rho_{i,j} = \left(\rho_{i,j} - \rho_{i,j}^3 + D(\Delta_N \rho)_{i,j}\right) dt + \sqrt{2\beta^{-1}} h^{-1} d\eta_{i,j} \tag{33}$$

Here $\eta_{i,j}$ is set of independent Wiener processes, Δ_N is the discrete Laplacian,

$$(\Delta_N \rho)_{i,j} = h^{-2} \left(\rho_{i+1,j} + \rho_{i-1,j} + \rho_{i,j+1} + \rho_{i,j-1} - 4\rho_{i,j}\right), \tag{34}$$

and the boundary conditions read

$$\begin{aligned} \rho_{i,j} &= 1, \qquad \text{for} \ \ i = 0, N, \quad j = 1, \dots, N-1, \\ \rho_{i,j} &= -1, \qquad \text{for} \ \ j = 0, N, \quad i = 1, \dots, N-1. \end{aligned} \tag{35}$$

We can also set $\rho_{0,0} = \rho_{0,N} = \rho_{N,0} = \rho_{N,N} = 0$. This model is in the class for we would like to solve the committor. Here the energy is the discrete equivalent to (28):

$$V(\boldsymbol{\rho}) = h^{-2} \sum_{i,j=1}^{N} \left(\tfrac{1}{2} D |(\nabla_N \rho)_{i,j}|^2 + \tfrac{1}{4}(1 - |\rho_{i,j}|^2)^2\right) \tag{36}$$

where ∇_N is the discrete gradient so that

$$|(\nabla_N \rho)_{i,j}|^2 = h^{-2} \left(\rho_{i+1,j} - \rho_{i,j}\right)^2 + h^{-2} \left(\rho_{i,j+1} - \rho_{i,j}\right)^2. \tag{37}$$

The discrete Laplacian has the effect of aligning neighboring lattice sites. The example we consider has a 256-dimensional state space. As shown in Fig. 2, the shows the characteristic pathway for a transition between the two metastable states, with the expected hourglass shape as transition state [29] that can also be identified by the minimum action method in this specific example [30, 31]. It should be noted that the initial increase in the loss function arises due to an initial representation of the transition path that is not consistent with the dynamics of the model and that once representative configurations are sampled, the estimate of the loss improves.

VI. FUTURE WORK

The approach we propose here enables optimization in contexts in which the loss function is dominated by data that is exceedingly rare with respect to its equilibrium measure. While we have both theoretical and numerical evidence that this approach is effective for high-dimensional problems and improves generalization, further evidence from physics applications would bolster our current findings. In particular, we must test our approach on more complicated systems, like those typically arising in biophysics. In such systems, there may be multiple pathways connecting two metastable states, a complication that we did not investigate thoroughly here.

While our algorithm and code can easily accept a neural network architecture, we used very simple neural networks for the examples in this paper. Finding architectures that are well-adapted to a given physical system remains an important challenge. Additionally, there are natural improvements to the implementation of our algorithm: adaptive windowing, more sophisticated reweighting schemes, and exploiting the "embarrassingly parallel" structure of the computation to obtain computational speed-ups.

We focused on the computation of committor functions, but the approach we outline is much more general. Applications of importance sampling to reinforcement learning and optimization of generative models are among the most compelling future directions.

[1] G. Carleo, I. Cirac, K. Cranmer, and L. Daudet, Machine learning and the physical sciences*, Rev. Mod. Phys. **91**, 39 (2019).

[2] Y.-A. Ma, Y. Chen, C. Jin, N. Flammarion, and M. I. Jordan, Sampling can be faster than optimization, Proceedings of the National Academy of Sciences **116**, 20881 (2019).

[3] W. Mou, N. Flammarion, M. J. Wainwright, and P. L. Bartlett, An Efficient Sampling Algorithm for Non-smooth Composite Potentials, arXiv:1910.00551 [cs, stat] (2019), arXiv:1910.00551 [cs, stat].

[4] W. Mou, Y.-A. Ma, M. J. Wainwright, P. L. Bartlett, and M. I. Jordan, High-Order Langevin Diffusion Yields an Accelerated MCMC Algorithm, arXiv:1908.10859 [cs, math, stat] (2020), arXiv:1908.10859 [cs, math, stat].

[5] W. E and E. Vanden-Eijnden, Transition-Path Theory and Path-Finding Algorithms for the Study of Rare Events, Annual Review of Physical Chemistry **61**, 391 (2010).

[6] D. Chandler, *Introduction to Modern Statistical Mechanics* (Oxford University Press, USA, 1987).

[7] A. Barducci, M. Bonomi, and M. Parrinello, Metadynamics, WIREs Computational Molecular Science **1**, 826 (2011).

[8] D. Csiba and P. Richtarik, Importance Sampling for Minibatches, Journal of Machine Learning Research , 21 (2018).

[9] Y. Nesterov, Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems, SIAM Journal on Optimization **22**, 341 (2012).

[10] N. L. Roux, M. Schmidt, and F. R. Bach, A stochastic gradient method with an exponential convergence _Rate for finite training sets, in *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012) pp. 2663–2671.

[11] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Inc., 2013) pp. 315–323.

[12] Y. Fan, J. Xu, and C. R. Shelton, Importance Sampling for Continuous Time Bayesian Networks, Journal of Machine Learning Research , 2115 (2010).

[13] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Transition path sampling: Throwing ropes over rough mountain passes, in the dark., Annu. Rev. Phys. Chem. **53**, 291 (2002).

[14] L. Maragliano, A. Fischer, E. Vanden-Eijnden, and G. Ciccotti, String method in collective variables: Minimum free energy paths and isocommittor surfaces, J. Chem. Phys. **125**, 024106 (2006).

[15] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein, Metastability and Low Lying Spectra in Reversible Markov Chains, Communications in Mathematical Physics **228**, 219 (2002).

[16] W. E. and E. Vanden-Eijnden, Towards a Theory of Transition Paths, Journal of Statistical Physics **123**, 503 (2006).

[17] Y. Khoo, J. Lu, and L. Ying, Solving for high dimensional committor functions using artificial neural networks, arXiv (2018), arXiv:1802.10275v1.

[18] Q. Li, B. Lin, and W. Ren, Computing Committor Functions for the Study of Rare Events Using Deep Learning, The Journal of Chemical Physics **151**, 054112 (2019), arXiv:1906.06285.

[19] S. Yaida, Fluctuation-dissipation relations for stochastic gradient descent, arXiv:1810.00004 [cs, stat] (2018), arXiv:1810.00004 [cs, stat].

[20] B. Gaveau and L. S. Schulman, Theory of nonequilibrium first-order phase transitions for stochastic dynamics, Journal of Mathematical Physics **39**, 1517 (1998).

[21] M. Cameron and E. Vanden-Eijnden, Flows in Complex Networks: Theory, Algorithms, and Application to Lennard–Jones Cluster Rearrangement, Journal of Statistical Physics **156**, 427 (2014).

[22] J. Lu and E. Vanden-Eijnden, Exact dynamical coarse-graining without time-scale separation, The Journal of Chemical Physics **141**, 044109 (2014).

[23] L. Chizat and F. Bach, On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport, arXiv:1805.09545 [cs, math, stat] (2018), arXiv:1805.09545 [cs, math, stat].

[24] S. Mei, A. Montanari, and P.-M. Nguyen, A mean field view of the landscape of two-layer neural networks, Proceedings of the National Academy of Sciences **115**, E7665 (2018).

[25] G. M. Rotskoff and E. Vanden-Eijnden, Trainability and Accuracy of Neural Networks: An Interacting Particle System Approach, arXiv:1805.00915 [cond-mat, stat] (2018), arXiv:1805.00915 [cond-mat, stat].

[26] J. Sirignano and K. Spiliopoulos, Mean Field Analysis of Neural Networks, arXiv (2018), arXiv:1805.01053v1.

[27] E. H. Thiede, B. Van Koten, J. Weare, and A. R. Dinner, Eigenvector method for umbrella sampling enables error analysis, The Journal of chemical physics **145**, 084115 (2016).

[28] K. Müller and L. D. Brown, Location of saddle points and minimum energy paths by a constrained simplex optimization procedure, Theoretica chimica acta **53**, 75 (1979).

[29] R. V. Kohn, F. Otto, M. G. Reznikoff, and E. Vanden-Eijnden, Action minimization and sharp-interface limits for the stochastic Allen-Cahn equation, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences **60**, 393 (2007).

[30] W. E, W. Ren, and E. Vanden-Eijnden, Minimum action method for the study of rare events, Communications on pure and applied mathematics **57**, 637 (2004).

[31] M. Heymann and E. Vanden-Eijnden, The geometric minimum action method: A least action principle on the space of curves, Comm. Pure Appl. Math. **61**, 1052 (2008).

Appendix A: Variance reduction improves generalization

1. Proof of Prop 3.1

Consider the empirical risk minimization (ERM) problem for data sampled iid from some measure $\{\boldsymbol{x}_m\}_{m=1}^M \sim \mu$. In this problem, we seek to minimize the empirical risk (or loss)

$$L_M(\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=1}^{M} \ell(\boldsymbol{x}_m, \boldsymbol{\theta}) \tag{A.1}$$

where ℓ is a convex function that measures the discrepancy between the target function and the neural network representation at $\boldsymbol{x}$. For example,

$$\ell(\boldsymbol{x}, \boldsymbol{\theta}) = |f_*(\boldsymbol{x}) - f(\boldsymbol{x}, \boldsymbol{\theta})|. \tag{A.2}$$

The generalization error is measured by the "population loss"

$$L(\boldsymbol{\theta}) = \int_\Omega \ell(\boldsymbol{x}, \boldsymbol{\theta}) d\mu(\boldsymbol{x}). \tag{A.3}$$

In the algorithm we propose for optimizing committor functions, the importance sampling reduces the variance of our estimator of the empirical loss. We show that this reduction in variance for the estimator of $L(\boldsymbol{\theta})$ leads to better generalization. Let us denote by $\tilde{\mu}$ the measure that we use to perform importance sampling. Defining

$$\tilde{L}_M(\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=1}^{M} \ell(\tilde{\boldsymbol{x}}_m, \boldsymbol{\theta}) \quad \tilde{\boldsymbol{x}}_m \sim \frac{d\mu}{d\tilde{\mu}}\,\tilde{\mu} \tag{A.4}$$

we have $\mathbb{E}\tilde{L}_M(\boldsymbol{\theta}) = L(\boldsymbol{\theta})$ and we can assume that

$$\forall \boldsymbol{\theta}, \quad \mathrm{var}_{\tilde{\mu}}(\tilde{L}_M(\boldsymbol{\theta})) \le \mathrm{var}_{\mu}(L_M(\boldsymbol{\theta})). \tag{A.5}$$

Denoting by $\boldsymbol{\theta}_M$, $\tilde{\boldsymbol{\theta}}_M$, and $\boldsymbol{\theta}_*$ minimizers of L_M, $\tilde{L}_M$, and L, respectively, a simple asymptotic argument demonstrates that

$$\mathbb{E}_{\tilde{\mu}}|\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*|^2 \le \mathbb{E}_{\mu}|\boldsymbol{\theta}_M - \boldsymbol{\theta}_*|^2, \tag{A.6}$$

meaning that importance sampling reduces the asymptotic variance of the ERM. This argument requires several assumptions so we state it as a proposition

Proposition A.1. *Assume that L, L_M, and $\tilde{L}_M$ are strictly convex (which can be guaranteed with regularization) and that there exist $0 < \sigma_1 \le \sigma_2 < \infty$ such that*

$$\sigma_1\, Id \preceq \nabla\nabla L(\boldsymbol{\theta}_*) \preceq \sigma_2\, Id \tag{A.7}$$

If the importance sampling guarantees that

$$\forall \boldsymbol{\theta} \quad : \quad \sigma_2 var_{\tilde{\mu}}(\nabla \tilde{L}_M(\boldsymbol{\theta})) \le \sigma_1 var_{\mu}(\nabla L_M(\boldsymbol{\theta})) \tag{A.8}$$

then

$$\mathbb{E}_{\tilde{\mu}}|\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*|^2 \le \mathbb{E}_{\mu}|\boldsymbol{\theta}_M - \boldsymbol{\theta}_*|^2 \tag{A.9}$$

We establish this result by asymptotic expansion around the population loss minimizer $\boldsymbol{\theta}_*$, which is unique by the assumption of strict convexity of L. Assume that for M large enough we can approximate

$$L_M(\boldsymbol{\theta}) = L_M(\boldsymbol{\theta}_*) + \nabla L_M(\boldsymbol{\theta}_*)\cdot(\boldsymbol{\theta} - \boldsymbol{\theta}_*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)^T \nabla\nabla L_M(\boldsymbol{\theta}_*)(\boldsymbol{\theta} - \boldsymbol{\theta}_*) \tag{A.10}$$

we see that the minimizer for L_M can be written

$$\boldsymbol{\theta}_M = \boldsymbol{\theta}_* - (\nabla\nabla L_M(\boldsymbol{\theta}_*))^{-1}\nabla L_M(\boldsymbol{\theta}_*) \tag{A.11}$$

where we ensure the invertibility of Hessian with the strict convexity assumption. Therefore

$$\mathbb{E}_\mu|\boldsymbol{\theta}_M - \boldsymbol{\theta}_*|^2 = \mathbb{E}_\mu|(\nabla\nabla L_M(\boldsymbol{\theta}_*))^{-1}\nabla L_M(\boldsymbol{\theta}_*)|^2. \tag{A.12}$$

which, for $M \gg 1$, we can approximate as

$$\mathbb{E}_\mu|\boldsymbol{\theta}_M - \boldsymbol{\theta}_*|^2 \approx \mathbb{E}_\mu|(\nabla\nabla L(\boldsymbol{\theta}_*))^{-1}\nabla L_M(\boldsymbol{\theta}_*)|^2 \geq \sigma_2^{-1}\mathbb{E}_\mu|\nabla L_M(\boldsymbol{\theta}_*)|^2. \tag{A.13}$$

where we used (A.7). Using the same calculation for the variance $\mathbb{E}_{\tilde{\mu}}|\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*|^2$ we obtain

$$\mathbb{E}_{\tilde{\mu}}|\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*|^2 \leq \sigma_1^{-1}\mathbb{E}_{\tilde{\mu}}|\nabla\tilde{L}_M(\boldsymbol{\theta}_*)|^2. \tag{A.14}$$

and combining the last two equations gives

$$\mathbb{E}_{\tilde{\mu}}|\tilde{\boldsymbol{\theta}}_M - \boldsymbol{\theta}_*|^2 \leq \mathbb{E}_\mu|\boldsymbol{\theta}_M - \boldsymbol{\theta}_*|^2 \tag{A.15}$$

since (A.8) implies

$$\sigma_2\mathbb{E}_{\tilde{\mu}}|\nabla\tilde{L}_M(\boldsymbol{\theta}_*)|^2 \leq \sigma_1\mathbb{E}_\mu|\nabla L_M(\boldsymbol{\theta}_*)|^2 \tag{A.16}$$

2. Proof of Prop. 3.2

To state the proposition, we rely on the mean-field picture of wide neural networks, following [23, 25]. In this formulation, we express the neural network as

$$q(\boldsymbol{x}) = \int_D \varphi(\boldsymbol{x}, \boldsymbol{z})d\gamma(\boldsymbol{z}) \tag{A.17}$$

where φ is a nonlinear function (the unit) and γ is a signed Radon measure defined on the parameter space D. When using the boundary conditions to optimize the committor function described in the main text, we use a thresholding function to ensure that the range of q is in $[0, 1]$. The following argument can be adapted to that case, but it is clearer to rely on the boundary conditions described in App. C and use the representation (A.17).

We let $\mathcal{L}$ be the convex population risk functional with minimizer γ_* and minimum value zero, $\mathcal{L}_*[\gamma_*] = 0$. Similarly, we let $\mathcal{L}_M$ be the convex empirical risk functional with minimizer γ_M and minimum value zero, $\mathcal{L}_M[\gamma_M] = 0$. Denoting by D_γ the functional derivative, the conditions of optimality require that

$$D_\gamma\mathcal{L}_M[\boldsymbol{z}, \gamma_M] \geq 0, \qquad D_\gamma\mathcal{L}[\boldsymbol{z}, \gamma_*] \geq 0 \tag{A.18}$$

with equality for all $\boldsymbol{z} \in \operatorname{supp}\gamma_M$ for the first relation, and for all $\boldsymbol{z} \in \operatorname{supp}\gamma_*$ for the second. Note that γ_* is also a minimizer of the empirical loss, $\mathcal{L}_M[\gamma_*] = 0$, since $\mathcal{L}_M \geq 0$ and $\mathbb{E}_\mu\mathcal{L}_M[\gamma_*] = \mathcal{L}[\gamma_*] = 0$. Therefore

$$\forall\boldsymbol{z} \in \operatorname{supp}\gamma_M \;:\; D_\gamma\mathcal{L}_M[\boldsymbol{z}, \gamma_M] = 0 \qquad \forall\boldsymbol{z} \in \operatorname{supp}\gamma_* \;:\; D_\gamma\mathcal{L}_M[\boldsymbol{z}, \gamma_*] = D_\gamma\mathcal{L}[\boldsymbol{z}, \gamma_*] = 0 \tag{A.19}$$

The proof proceeds in two steps. First, by convexity,

$$\mathcal{L}[\gamma_M] \leq \int D_\gamma\mathcal{L}[\boldsymbol{z}, \gamma_M](d\gamma_M(\boldsymbol{z}) - d\gamma_*(\boldsymbol{z})) \tag{A.20}$$

which, after taking the supremum over $\boldsymbol{z}$, leads to the upper bound

$$\mathcal{L}[\gamma_M] \leq 2\sup_{\boldsymbol{z}}|D_\gamma\mathcal{L}[\boldsymbol{z}, \gamma_M]||\gamma_*(\boldsymbol{z})|_{\mathrm{TV}} \tag{A.21}$$

where we have assumed that $|\gamma_M|_{\text{TV}} \leq |\gamma_*|_{\text{TV}}$ which can be ensured with an appropriate regularization.

Next, using (A.19), we observe

$$D_\gamma \mathcal{L}[\boldsymbol{z}, \gamma_M] = D_\gamma \mathcal{L}[\boldsymbol{z}, \gamma_M] - D_\gamma \mathcal{L}_M[\boldsymbol{z}, \gamma_M] \equiv D_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \gamma_M]. \tag{A.22}$$

As a result

$$\begin{aligned}
&D_\gamma \mathcal{L}_M[\boldsymbol{z}, \gamma_* + \gamma_M - \gamma_*] \\
&= D_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \gamma_* + \gamma_M - \gamma_*] \\
&= D_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \gamma_*] + \int_0^1 dt \int D^2_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \boldsymbol{z}', \gamma_* + t(\gamma_M - \gamma_*)] d\gamma_*(\boldsymbol{z}') d\gamma_M(\boldsymbol{z}') \\
&= \int D^2_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \boldsymbol{z}', \gamma_* + t_m(\gamma_M - \gamma_*)] d\gamma_*(\boldsymbol{z}') d\gamma_M(\boldsymbol{z}')
\end{aligned} \tag{A.23}$$

where we used (A.19) and we fixed $t_m \in [0, 1]$ by the mean value theorem to get the third equality. This implies

$$|D_\gamma \mathcal{L}_M[\boldsymbol{z}, \gamma_M]| \leq \sup_{\substack{\boldsymbol{z}' \\ t \in [0,1]}} \|D^2_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \boldsymbol{z}', \gamma_* + t(\gamma_M - \gamma_*)]\| |\gamma_*|^2_{\text{TV}} \tag{A.24}$$

Combining with (A.21) we have

$$\mathcal{L}[\gamma_M] \leq 4 \sup_{\substack{\boldsymbol{z}, \boldsymbol{z}' \\ t \in [0,1]}} \|D^2_\gamma \Delta \mathcal{L}_M[\boldsymbol{z}, \boldsymbol{z}', \gamma_* + t(\gamma_M - \gamma_*)]\| |\gamma_*|^2_{\text{TV}} \tag{A.25}$$

The term involving $\|D^2_\gamma \Delta \mathcal{L}_M\|$ is suppressed with importance sampling, which completes the argument.

Appendix B: Approximation of the committor with a neural network

1. Representation

In our experiments we use single hidden layer ReLU networks: the simplicity of the networks is meant to demonstrate the generic capabilities of the methodology, as we are not relying on sophisticated architectures. We define a parametric representation of the committor function and an objective function that enables us to optimize the parameters. Neural networks (NN) offer flexibility to the representation and relative ease of optimization, making this a natural choice for a representation of the committor. To this end, we proceed in two distinct steps.

First we will work under the assumption that $\hat{q}(\boldsymbol{x}, \boldsymbol{v})$ can be well approximated by a function of the position alone, i.e. we set

$$\hat{q}(\boldsymbol{x}, \boldsymbol{v}) \approx q(\boldsymbol{x}), \tag{B.1}$$

Second we represent $q(\boldsymbol{x})$ by a neural network taking the input $\boldsymbol{x}$ through some predefined map $\phi : \mathbb{R}^d \to \mathbb{R}^k$ with $k \leq d$. For example, we can use

$$q(\boldsymbol{x}) = \sigma \left[\frac{1}{n} \sum_{i=1}^n \varphi(\phi(\boldsymbol{x}), \boldsymbol{\theta}_i) \right] \tag{B.2}$$

which represents a single hidden layer neural network with φ a nonlinear function (e.g., ReLU) passed through a thresholding function σ (e.g., a sigmoid function) to ensures that $q(\boldsymbol{x}) \in [0, 1]$, $\forall \boldsymbol{x} \in \mathbb{R}^d$. In practice, the architecture of the neural network will be substantially more intricate than the single hidden layer network (B.2).

In the optimization procedure below, it is more tractable to penalize deviations from the boundary conditions rather than impose them as constraints. Consequently, we use a strong Lagrange multipliers to ensure that the committor has the right values on the initial and target states. The objective function we use is thus

$$\begin{aligned} C_\lambda[q] &= Z^{-1}\int_{\mathbb{R}^d} |\nabla q(\boldsymbol{x})|^2 e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \\ &+ \lambda Z^{-1}\int_A q(\boldsymbol{x})^2 e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} + \lambda Z^{-1}\int_B \big(1-q(\boldsymbol{x})\big)^2 e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \\ &\equiv \left\langle |\nabla q|^2\right\rangle_\beta + \lambda\left\langle |q|^2 1_A\right\rangle_\beta + \lambda\left\langle |1-q|^2 1_B\right\rangle_\beta \end{aligned} \tag{B.3}$$

where $\langle\cdot\rangle_\beta$ denotes canonical expectation with respect to $Z^{-1}e^{-\beta V(\boldsymbol{x})}$, and 1_A and 1_B are the indicator functions of A and B, respectively. Using the parametric representation of the committor in (B.2) the problem becomes to find a set of $\{\boldsymbol{\theta}\}_{i=1}^n$ that minimize C_λ.

2. Computing the gradients

Optimization of the neural network representation of the committor (B.2) by gradient descent (GD) requires estimating the gradient of the objective function with respect to the parameters. Denote

$$q(\boldsymbol{x},\boldsymbol{\theta}) = \sigma\left[\frac{1}{n}\sum_{i=1}^n \varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i)\right] \tag{B.4}$$

so that

$$\tfrac{1}{2}\nabla_{\boldsymbol{\theta}_i} C_\lambda[q] = \langle \nabla_{\boldsymbol{\theta}_i}\nabla_{\boldsymbol{x}} q \nabla_{\boldsymbol{x}} q\rangle_\beta + \lambda\left\langle q\nabla_{\boldsymbol{\theta}_i} q 1_A\right\rangle_\beta - \lambda\left\langle (1-q)\nabla_{\boldsymbol{\theta}_i} q 1_B\right\rangle_\beta \tag{B.5}$$

Noting that

$$\nabla_{\boldsymbol{x}}\sigma(f(\boldsymbol{x})) = \sigma(f(\boldsymbol{x}))\left(1-\sigma(f(\boldsymbol{x}))\right)\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{B.6}$$

and similarly for $\nabla_{\boldsymbol{\theta}}$ we can derive explicit expressions for the quantities that need to be averaged. In particular, we see that

$$\nabla_{\boldsymbol{x}} q(\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{n} q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\sum_{i=1}^n \nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_\phi\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i), \tag{B.7}$$

$$\nabla_{\boldsymbol{\theta}_i} q(\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{n} q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}_i}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i) \tag{B.8}$$

and

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_i}\nabla_{\boldsymbol{x}} q(\boldsymbol{x},\boldsymbol{\theta}) &= \frac{1}{n}\nabla_{\boldsymbol{\theta}_i} q(\boldsymbol{x},\boldsymbol{\theta})(1-2q(\boldsymbol{x},\boldsymbol{\theta}))\sum_{j=1}^n \nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_\phi\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_j) \\ &+ \frac{1}{n} q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_\phi\nabla_{\boldsymbol{\theta}_i}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i), \end{aligned} \tag{B.9}$$

3. Choice of the windowing functions

It remains to specify the function $W_l(\boldsymbol{x})$. Here we propose to do this in a way that is adaptive to $q(\boldsymbol{x},\boldsymbol{\theta})$ itself. To this end, let

$$\sigma(u) = \frac{1}{1+e^{-u}} \tag{B.10}$$

and given $u_0 < u_1 < u_2 < \cdots < u_L$ and some $k > 0$, let

$$W_l(\boldsymbol{x}) = \sigma\left(k(q(\boldsymbol{x}, \boldsymbol{\theta}) - u_{l-1})\right) - \sigma\left(k(q(\boldsymbol{x}, \boldsymbol{\theta}) - u_l)\right), \qquad l = 1, \ldots, L \tag{B.11}$$

Since $q(\boldsymbol{x}, \theta) \in [0, 1]$ by construction we then have

$$\begin{aligned} \forall \boldsymbol{x} \in \mathbb{R}^d \quad : \quad \sum_{l=1}^{L} W_l(\boldsymbol{x}) &= \sigma\left(k(q(\boldsymbol{x}, \boldsymbol{\theta}) - u_0)\right) - \sigma\left(k(q(\boldsymbol{x}, \boldsymbol{\theta}) - u_L)\right) \\ &\geq \sigma(-k u_0) - \sigma(k(1 - u_L)) \end{aligned} \tag{B.12}$$

That is if we take k large enough and pick $u_0 = -a$ and $u_L = 1 + a$ with $a > 0$ such that $ka \gg 1$, the nonnegative functions $W_l(\boldsymbol{x})$ can be made to satisfy (19) to arbitrary precision exponentially fast in ak. The functions $W_l(\boldsymbol{x})$ are also peaked around $q(\boldsymbol{x}, \theta) = \frac{1}{2}(u_l + u_{l-1})$ which means that by taking enough values of u_l between $u_0 = -a$ and $u_L = 1 + a$ we can cover all the range of possible values for $q(\boldsymbol{x}, \boldsymbol{\theta})$.

Appendix C: Alternative formulation of the committor and boundary conditions

The variational problem of determining the committor function can be reinterpreted via a solution to the following PDE [22],

$$L\tilde{q} = \tau e^{\beta V(\boldsymbol{x})} \left[\delta(\boldsymbol{x} - \boldsymbol{a}) - \delta(\boldsymbol{x} - \boldsymbol{b})\right]. \tag{C.1}$$

Given a solution to (C.1), it is straightforward to verify that the committor between sets

$$\begin{aligned} A &= \{\boldsymbol{x} | \tilde{q}(\boldsymbol{x}) \leq \tilde{q}_-\} \\ B &= \{\boldsymbol{x} | \tilde{q}(\boldsymbol{x}) \geq \tilde{q}_+\} \end{aligned}$$

is given by

$$q(\boldsymbol{x}) = \frac{\tilde{q}(\boldsymbol{x}) - \tilde{q}_-}{\tilde{q}_+ - \tilde{q}_-} \tag{C.2}$$

for $\boldsymbol{x} \in (A \cup B)^c$.

We can use the variational optimization algorithm Alg.IV.1 to compute $\tilde{q}$ where we penalize the cost functional to obtain the loss function,

$$C_\lambda[\tilde{q}] = C[\tilde{q}] + \lambda[\tilde{q}(\boldsymbol{a}) - \tilde{q}(\boldsymbol{b})]. \tag{C.3}$$

In practice, we mollify the objective by replacing the δ-masses with Gaussians centered at $\boldsymbol{a}$ and $\boldsymbol{b}$ using the mean in the penalized objective above.

This formulation offers several advantages compared to the formulation discussed in the main text. First, because the range of $\tilde{q}$ is all of $\mathbb{R}$, there is no need to use thresholding functions that could affect the magnitude of gradients and hence the rate of convergence of the optimization. Secondly, in order to use the penalized objective of the main text, we must draw samples from the metastable states A and B. If those states are difficult to sample, the boundary conditions here require knowledge of only a single point within A and B.

Appendix D: Using collective variables maps

While we do not pursue this strategy in the numerical examples considered here, collective variables are often useful in complex molecular systems, especially when physical insight into the reaction mechanism can be deployed. Identifying a low-dimensional subspace that describes the reaction can help alleviate the issue of sampling high dimensional systems. This approach may also be useful to remove symmetries that are irrelevant for the reaction in question. To do so, we use the standard notion of a reaction coordinate, which is a map

$$\phi : \mathbb{R}^d \to \mathbb{R}^k; \quad \phi(\boldsymbol{x}) \mapsto \boldsymbol{z} \tag{D.1}$$

with $k \ll d$. In some sense, the committor is *the* reaction coordinate for the system, but it remains too unwieldy to be directly useful. Instead, we hope to approximate the committor; in what follows, we will use two distinct approximations:

$$\hat{q}(\boldsymbol{x}, \boldsymbol{v}) \approx q(\boldsymbol{x}), \tag{D.2}$$

and

$$\hat{q}(\boldsymbol{x}, \boldsymbol{v}) \approx q(\boldsymbol{z}) \quad \text{with} \quad \boldsymbol{z} = \phi(\boldsymbol{x}). \tag{D.3}$$

In the second case (D.3), we use the collective variables map to eliminate degrees of freedom in addition to reducing the dimensionality by neglecting the momenta. Our goal at this point is to find a computationally tractable scheme for identifying q.

Using (D.2), we find that the objective function in collective variables can be written

$$\begin{aligned} \mathcal{C}[q] &= C \int_{\mathbb{R}^d} |\nabla q(\phi(\boldsymbol{x}))|^2 e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \\ &= C \int_{\mathbb{R}^d} |\nabla \phi(\boldsymbol{x}) \cdot \nabla q(\boldsymbol{z})|^2 e^{-\beta V(\boldsymbol{x})} \delta(\boldsymbol{z} - \phi(\boldsymbol{x})) d\boldsymbol{x} \\ &= C \int_{\mathbb{R}^k} \langle \nabla q(\boldsymbol{z}), M(\boldsymbol{z}) \nabla q(\boldsymbol{z}) \rangle \, e^{-\beta F(\boldsymbol{z})} d\boldsymbol{z}, \end{aligned} \tag{D.4}$$

where we have defined

$$F(\boldsymbol{z}) = -\beta^{-1} \log \int_{\mathbb{R}^d} \delta(\boldsymbol{z} - \phi(\boldsymbol{x})) e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \tag{D.5}$$

which can be viewed as the potential of mean force for the collective variable $\boldsymbol{z} \in \mathbb{R}^k$ and C is a constant that will not affect the optimization. Further, the change of coordinates necessitates including the metric term

$$M(\boldsymbol{z}) = \frac{\int \nabla \phi(\boldsymbol{x}) \cdot \nabla \phi(\boldsymbol{x}) e^{-\beta V(\boldsymbol{x})} \delta(\phi(\boldsymbol{x}) - \boldsymbol{z}) d\boldsymbol{x}}{\int \delta(\phi(\boldsymbol{x}) - \boldsymbol{z}) e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x}}. \tag{D.6}$$

Importantly, we can relate the solution of the optimization in collective variables space to the solution in the original state space. To do so, we examine the minimizers of the objective written with collective variables by studying its Euler-Lagrange equation. The objective D.4 can be expressed as

$$\begin{aligned} \tilde{\mathcal{C}}_\lambda[q] = \frac{1}{2} \int_0^1 \int_{\mathbb{R}^k} &\langle \nabla q(\boldsymbol{z}), M(\boldsymbol{x}) \nabla q(\boldsymbol{z}) \rangle \, e^{-\beta F(\boldsymbol{z}) + \beta G(q(\boldsymbol{z}))} du \\ + \frac{\lambda}{2} \int_0^1 \int_A &q(\boldsymbol{z})^2 e^{-\beta F(\boldsymbol{z}) + \beta G(q(\boldsymbol{z}))} \delta(q(\boldsymbol{z}) - u) d\boldsymbol{z} \; du \\ + \frac{\lambda}{2} \int_0^1 \int_B &\big(1 - q(\boldsymbol{z})\big)^2 e^{-\beta F(\boldsymbol{z}) + \beta G(q(\boldsymbol{z}))} \delta(q(\boldsymbol{z}) - u) d\boldsymbol{z} \; du \end{aligned} \tag{D.7}$$

where we have introduced

$$G(u) = -\beta^{-1} \log \int_{\mathbb{R}^k} e^{-\beta F(\boldsymbol{z})} \delta(q(\boldsymbol{z}) - u) d\boldsymbol{z}. \tag{D.8}$$

This objective can be interpreted as the integral of a conditional expectation (not explicitly writing the integrals that contain the boundary conditions)

$$\begin{aligned} \tilde{\mathcal{C}}_\lambda[q] &= \int_0^1 \frac{\int_{\mathbb{R}^k} \langle \nabla q(\boldsymbol{z}), M(\boldsymbol{x})\nabla q(\boldsymbol{z})\rangle\, e^{-\beta F(\boldsymbol{z})}\delta(q(\boldsymbol{z})-u)d\boldsymbol{z}}{e^{-\beta G(u)}} du + \text{B.C.} \\ &\equiv \int_0^1 \mathbb{E}_u \left\langle \nabla q(\boldsymbol{z}), M(\boldsymbol{z})\nabla(\boldsymbol{z})\right\rangle du + \text{B.C.} \end{aligned} \tag{D.9}$$

Let us first consider the case where $\phi \equiv id$; that is, no collective variables. An optimum of $\tilde{\mathcal{C}}_\lambda$ solves a different Euler-Lagrange equation compared to a minimizer of $\mathcal{C}_\lambda$, which we can easily see satisfies

$$\Delta\hat{q} - \beta\nabla V \cdot \nabla\hat{q} = 0 \tag{D.10}$$

However, it suffices to minimize (D.9) to find a minimizer of (10) because we have an explicit relation between the two. To see this, we compute the variation of the objective function with respect to q. The Euler-Lagrange equation satisfied by (D.7)—neglecting the boundary conditions for clarity—reads

$$\begin{aligned} 0 &= \frac{\delta\tilde{\mathcal{C}}_\lambda}{\delta q} \\ &= \frac{\delta}{\delta q}\int \nabla(q+\delta q)\cdot\nabla(q+\delta q)e^{-\beta V(\boldsymbol{x})+\beta G(q(\boldsymbol{x}))}\delta(q(\boldsymbol{x})-u)d\boldsymbol{x}\; du \\ &= \frac{\delta}{\delta q}\int \delta q\big[\nabla V\cdot\nabla q - \beta G'(q)|\nabla q|^2 - \Delta q\big]e^{-\beta V(\boldsymbol{z})+\beta G(q)}d\boldsymbol{z}\; du \\ &+ \frac{1}{2}\frac{\delta}{\delta q}\int \delta q G'(q)|\nabla q|^2 e^{-\beta F(\boldsymbol{z})+\beta G(q)}d\boldsymbol{z}\; du, \end{aligned} \tag{D.11}$$

yielding

$$\Delta q - \nabla V\cdot\nabla q + \frac{\beta}{2}G'(q)|\nabla q|^2 = 0. \tag{D.12}$$

Hence, we see that the solution of (D.12) is related to the solution of the original committor's Euler Lagrange equation (D.10). In particular, suppose that q were a reparameterization of $\hat{q}$ in the sense that there is a map $\hat{q} = f(q)$. Then, using (D.10), we have that

$$\frac{f''(q)}{f'(q)}|\nabla q|^2 - \nabla V\cdot\nabla q + \Delta q = 0 \tag{D.13}$$

meaning that we can recover $\hat{q}$ given a solution of (D.12). This amounts to determining f which can be computed by observing

$$\frac{d}{dq}\log f' = \frac{\beta}{2}G'(q) \tag{D.14}$$

so that

$$f(q) = \frac{\int_0^q e^{\frac{\beta}{2}G(u)}du}{\int_0^1 e^{\frac{\beta}{2}G(u)}du} \tag{D.15}$$

where we used the fact that $f(0) = 0$ and $f(1) = 1$ to determine the constant of integration.