Proceedings of Machine Learning Research vol 107:1–24, 2021

# Active Importance Sampling for Variational Objectives Dominated by Rare Events: Consequences for Optimization and Generalization

**Grant M. Rotskoff** ROTSKOFF@STANFORD.EDU
*Stanford University, Stanford, CA, USA*

**Andrew R. Mitchell** AMITCHE2@STANFORD.EDU
*Stanford University, Stanford, CA, USA*

**Eric Vanden-Eijnden** EVE2@CIMS.NYU.EDU
*Courant Institute, New York City, NY, USA*

## Abstract

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 training 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 solving high-dimensional PDEs that admit a variational formulation, 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.

**Keywords:** Partial Differential Equations; Importance Sampling; Rare Events; Backward Kolmogorov Equation; Variational Monte Carlo

## 1. Introduction

Deep neural networks (DNNs) have become an essential tool for a diverse set of problems in data science and, increasingly, the physical sciences Carleo et al. (2019). The uncommonly robust approximation properties of DNNs undergird the successes of deep learning in seemingly disparate problems LeCun et al. (2015). The power of approaches based on deep learning is evident in high-dimensional settings where most classical tools from numerical analysis break down, due to the curse of dimensionality Donoho and Johnstone (1989). Many compelling questions in statistical physics require precise knowledge of high-dimensional functions, objects which can be challenging to represent and compute, suggesting that machine learning may have a transformative role to play.

Of course, challenges arise when using machine learning techniques in the physical sciences that do not appear in conventional settings. Unlike in computer vision and natural language processing, curated data sets are not typically available for physical problems that we intend to solve *de novo*.

© 2021 G.M. Rotskoff, A.R. Mitchell & E. Vanden-Eijnden.

As a result, we must generate the data either experimentally or computationally that we use to train our models.

Of particular interest in this context are problems involving high-dimensional partial differential equations (PDE) that can be formulated as variational minimization problems. Many PDEs of interest in statistical mechanics and quantum mechanics admit such a variational principle, and they lend themselves naturally to solution by machine learning techniques since the objective function can serve as a loss to train a neural network used to represent the solution. How to generate data to evaluate this objective constitutes, perhaps, the core challenge in problems of this type because the data that dominates the objective may be rare if sampled naively. In this work, we address this sampling problem.

**Neural networks for variational PDEs.** Consider a PDE whose solution can be found via the minimization problem

$$\min_{f \in \mathcal{F}} \mathcal{I}(f) \tag{1}$$

Here

$$\mathcal{I}(f) = \int_\Omega \mathcal{L}(\boldsymbol{x}, f) d\nu(\boldsymbol{x}), \tag{2}$$

where $\Omega \subset \mathbb{R}^d$, $\nu$ is some positive measure, and $\mathcal{L}(\boldsymbol{x}, f)$ is some Lagrangian depending on $\boldsymbol{x}$ as well as $f$ and its derivatives: Typical examples are

$$\mathcal{L}(\boldsymbol{x}, f) = \tfrac{1}{2}|\nabla f(\boldsymbol{x})|^2 + V(\boldsymbol{x})|f(\boldsymbol{x})|^2, \qquad d\nu(\boldsymbol{x}) = d\boldsymbol{x}, \tag{3}$$

where $V : \Omega \to \mathbb{R}$ is some potential, which gives the time-independent Schrödinger equation if we impose $\int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x} = 1$, or

$$\mathcal{L}(\boldsymbol{x}, f) = \tfrac{1}{2}|\nabla f(\boldsymbol{x})|^2, \qquad d\nu(\boldsymbol{x}) = e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \qquad (\beta > 0) \tag{4}$$

which gives the time-independent backward Kolomogorov equation if we impose some boundary conditions.

Variational Monte Carlo (VMC) procedures Toulouse et al. (2016) have been used to compute solutions to PDEs that admit this formulation. In this context, solutions are often computed using the Ritz method, which essentially amounts to optimizing the weights of specified, hand-chosen basis elements. Methods based on neural networks Eigel et al. (2019); E and Yu (2017) offer an alternative to VMC which may need less *a priori* information about the solution by relying on the approximation power of these networks.

**Data acquisition and importance sampling.** Training a neural network to represent the solution of the PDE by minimizing (1) requires estimating the integral (2). Because there is no data set given beforehand, the most straightforward implementation samples data points on $\Omega$ from the measure $\nu$ properly normalized. While natural, this approach is by no means optimal and it could even fail if the expectation of $\mathcal{L}(\boldsymbol{x}, f)$ is dominated by events that are rare on $\nu$: a simple example illustrating this point is shown in Fig. 1. If $\Omega$ is high dimensional, the variance of a simple estimator using unbiased samples from $\nu$ will typically be large compared to its mean squared, and some form of importance sampling will therefore be required. If we were interested in estimating the loss $I(f)$, it is well-known that the optimal way to draw samples would be to use the reweighted measure $d\tilde{\nu}(\boldsymbol{x}) = I^{-1}(f)|\mathcal{L}(\boldsymbol{x}, f)|d\nu(\boldsymbol{x})$ and reweight the samples consistently using $I(f)|\mathcal{L}(\boldsymbol{x}, f)|^{-1}$ Awad et al.

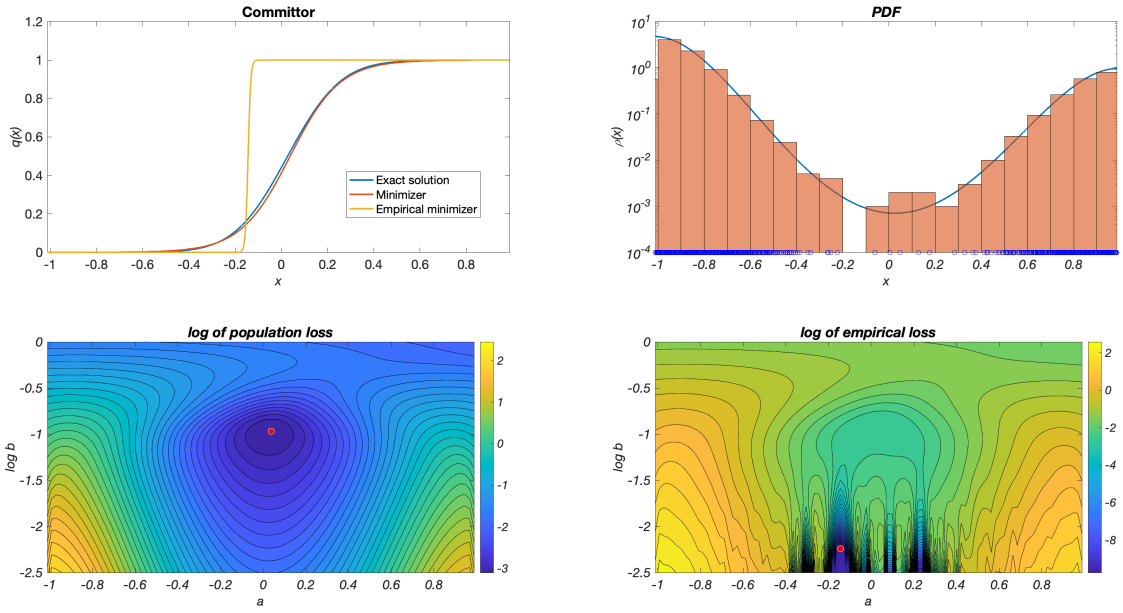


Figure 1: Simple illustrative example where the objective $I(q) = \int_{x_1}^{x_2} |q'(x)|^2 e^{-\beta V(x)} dx$, with $V(x) = (1-x^2)^2 + x/10$, $\beta = 8$, and $x_1$, $x_2$ at the minima of $V(x)$, is minimized over sigmoid functions with two parameters $a$ and $b$, controlling respectively their location and width – in this example, functions of this type do a good job at capturing the minimizer if $a$ and $b$ are properly are adjusted. Top left panel: the minimizers of the population and the empirical losses are compared to the actual solution; top right panel: the histogram of the data acquired by drawing $10^4$ independent samples (shown as circles) from the Gibbs distribution restricted on $x \in [x_1, x_2]$ is compared to the exact density; bottom panels: the population loss (left) is compared to the empirical loss (right) estimated on the data. Because the data is somewhat sparse near the maximum of $V(x)$ that dominates the objective, the empirical loss does a bad job at capturing the features of the population loss – note in particular the different scale and the added ruggedness in the empirical loss. As a result, the function optimized over this empirical loss differs significantly from the minimizer of the population loss that approximates the exact solution well, leading to a large generalization error. Note that here we sampled the measure and identified the minimizers by brute-force: in more complicated situations there is the added difficulty of performing this sampling, and minimizing the empirical loss by SGD. For more details on this example, see Appendix A

(2013). The difficulties with this approach are that the reweighted measure $\tilde{\nu}$ may not be easy to sample, and the reweighting factor involves the unknown value $I(f)$.

We show below that an importance sampling strategy can, however, be applied to reduce the variance of the estimator for the loss (as our training procedure relies on data generated at every training step) associated with variational problems of the type (1). We carry this out using umbrella sampling combined with replica exchange. These methods are widely used in applications from statistical mechanics because they offer a remedy to the problem of an objective dominated by rare data, but they are often rendered intractable becuase their practical design requires precise knowledge about where and how to sample. In our context, however, we can use the current estimate of the solution to inform and enhance the sampling in regions of the domain that contribute to the objective. The efficiency of such *active* learning approaches will be demonstrated below.

**Reactive events and committor.** As a specific application of practical interest that illustrates the general issues outlined above, we will focus on optimizing an objective of type (4) for a target function known as the "committor function," or committor in short. The committor is useful to identify reaction pathways and sample reactive trajectories in problems displaying metastability, a central question in statistical mechanics with decades of work behind it. In this context, the committor describes the probability that a configuration will "react", by transiting from one metastable basin to another under the stochastic dynamics of the system under consideration. Parameterizing the committor accurately (as many functions related to rare transitions in applications in condensed matter physics) requires samples from configurations that are rare under the Boltzmann distribution, a fact emphasized by the ubiquity of importance sampling methods for free energy calculations. With this in mind, calculating the committor epitomizes why a naive sampling strategy will not succeed in general and importance sampling is necessary.

**Related works.** Importance sampling and other variance reduction techniques have appeared in a variety of contexts in machine learning. Csiba and Richtarik (2018) 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 Nesterov (2012); Roux et al. (2012); Johnson and Zhang (2013), that aims to reduce the variance in the gradients when optimizing using stochastic gradient descent. In a separate line of inquiry, Fan et al. (2010) 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.

Our work parallels a line of inquiry in the Quantum Monte Carlo literature which has demonstrated the utility of neural network ansatze for electronic structure problems Han et al. (2019); Hermann et al. (2020); Pfau et al. (2020). Though the physical setting is quite different from the one we consider here, these works also rely on a strategy in which the data is collected online and there is feedback between training and data collection. Regarding the application to metastability, transition path sampling methods are perhaps the most closely related to our approach Bolhuis et al. (2002); Maragliano et al. (2006); E et al. (2005). Our applications are heavily influenced by the perspective of potential theory Bovier et al. (2002) and transition path theory E and Vanden-Eijnden (2006, 2010), which use the notion of the committor function (discussed in detail below) to charac-

terize metastability. Khoo et al. (2018) 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 approach most closely follows that of Li et al. (2019), 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 uses 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.

**Main contributions.** First, under very general assumptions, we show that importance sampling asymptotically improves the generalization error. Next, we describe an algorithm for *active* importance sampling that enables variance reduction for the estimator of the loss function, even in high-dimensional settings. Finally, we demonstrate numerically that this algorithm performs well both on low and high-dimensional examples and that, even when the total amount of data is fixed, optimizing the variational objective fails when importance sampling is not used.

## 2. Online learning and generalization error

Suppose we parametrize the function $f$ entering the objective function in (1) using a neural network, i.e. we set $f(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\theta})$, where $f(\cdot, \boldsymbol{\theta})$ is the network output and $\boldsymbol{\theta} \in \mathbb{R}^N$ collectively denotes all the parameters entering this network. This turns (1) into an objective function for the parameters $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta}) = I(f(\cdot, \boldsymbol{\theta})) = \int_\Omega \ell(\boldsymbol{x}, \boldsymbol{\theta}) d\nu(\boldsymbol{x}) \tag{5}$$

where

$$\ell(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{x}, f(\cdot, \boldsymbol{\theta})) \tag{6}$$

In the jargon of machine learning, $L(\boldsymbol{\theta})$ is called the population loss or risk, and in practice, it must be estimated using an empirical estimate. The simplest choice for the empirical loss is

$$L_n(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{7}$$

where $\{\boldsymbol{x}_i\}_{i=1}^n$ are *iid* drawn from $\nu$ (assuming this positive measure is normalized). This offers the possibility to optimize the parameters $\boldsymbol{\theta}$ by gradient descent (GD), i.e. using

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha \nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta}^k), \qquad k = 0, 1, 2, \ldots \tag{8}$$

where $\boldsymbol{\theta}^k$ denote the successive updates of the parameters starting from some initial $\boldsymbol{\theta}^0$ and $\alpha > 0$ is some time step (learning rate). In situations in which no data set is available beforehand, it is customary to use online learning, i.e. to generate new independent batches of data $\{\boldsymbol{x}_i\}_{i=1}^n$ after each (or a few) step(s) of the GD update. In (8): each $\{\boldsymbol{x}_i\}_{i=1}^n$ is called a minibatch, and the update in (8) is the widely used stochastic gradient descent (SGD) algorithm.

In this setup, the main issue becomes how to assess the quality of an approximation of the minimizer(s) of the population risk that we obtain using SGD. To phrase this question more precisely,

let us denote by $\{\bar{\boldsymbol{\theta}}^k\}_{k\in\mathbb{N}_0}$ the successive update of the parameters by GD over the population risk, i.e.

$$\bar{\boldsymbol{\theta}}^{k+1} = \bar{\boldsymbol{\theta}}^k - \alpha\nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k), \qquad k = 0, 1, 2, \dots \tag{9}$$

Let us assume that: Given some initial value $\bar{\boldsymbol{\theta}}^0$, the GD update in (9) converges towards a local minimizer of the population risk, $\boldsymbol{\theta}^* = \lim_{k\to\infty} \bar{\boldsymbol{\theta}}^k$, satisfying

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*) = 0, \qquad H^* = \nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*) \quad \text{is positive-definite} \tag{10}$$

Note that this assumption does not specify how good the local minimizer $\boldsymbol{\theta}^*$ is (i.e. how close $L(\boldsymbol{\theta}^*)$ is from $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$) but it requires that $L(\boldsymbol{\theta})$ be strictly convex in the vicinity of $\boldsymbol{\theta}^*$. Similar assumptions have been used to study SGD as variational inference Mandt et al. (2017). This assumption implies:

**Proposition 1** *The sequence $\{\boldsymbol{\theta}^k\}_{k\in\mathbb{N}_0}$ obtained using the SGD update in (8) starting from $\boldsymbol{\theta}^0 = \bar{\boldsymbol{\theta}}^0$ and using an independent batch of data $\{\boldsymbol{x}_i\}_{i=1}^n$ drawn from $\nu$ at every step is such that*

$$\lim_{k\to\infty}\lim_{n\to\infty} n\mathbb{E}_D[L(\boldsymbol{\theta}^k) - L(\boldsymbol{\theta}^*)] = \tfrac{1}{2}\alpha\,\mathrm{tr}[C^*H^*] \tag{11}$$

*Here $\mathbb{E}_D$ denotes expectation over all the batches used to compute the sequence $\boldsymbol{\theta}^k$, and $C^*$ is the $N \times N$ tensor that solves*

$$H^*C^* + C^*H^* - \alpha C^*H^*C^* = B^* \tag{12}$$

*where $B^*$ is the covariance of $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \boldsymbol{\theta}^*)$ (using $\nabla L(\boldsymbol{\theta}^*) = 0$)*

$$B^* = \int_\Omega \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \boldsymbol{\theta}^*)[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \boldsymbol{\theta}^*)]^T d\nu(\boldsymbol{x}) \tag{13}$$

The proof of this proposition is given in Appendix B. Essentially, it amounts to linearizing the sequence $\{\boldsymbol{\theta}^k\}_{k\in\mathbb{N}_0}$ from SGD around $\{\bar{\boldsymbol{\theta}}^k\}_{k\in\mathbb{N}_0}$ from GD: the resulting sequence is the discretized version of an Ornstein-Ulhenbeck process that can be analyzed exactly.

Even though the statement in (11) is only asymptotic in $n$ and $k$, it suggests that for large $k$ and large $n$, we will have

$$\mathbb{E}_D L(\boldsymbol{\theta}^k) = L(\boldsymbol{\theta}^*) + \tfrac{1}{2}n^{-1}\alpha\,\mathrm{tr}[C^*H^*] + \text{higher order corrections in } n \tag{14}$$

Therefore, if we can guarantee that $\boldsymbol{\theta}^*$ is a good local minimizer of the loss (which has to do with the choice of network architecture and how well-tailored it is to the problem at hand, the choice of the initial $\bar{\boldsymbol{\theta}}^0$, etc.), (11) indicates that the error made by learning using SGD rather than GD can be controlled by: (i) increasing the size $n$ of the batches, (ii) decreasing the learning rate $\alpha$, or (iii) reducing $\mathrm{tr}[C^*H^*]$. The first two observations are standard and are at the core of the Robbins-Monro stochastic approximation procedure Robbins and Monro (1951). The third observation is also not surprising from the proof of Proposition 1 which shows that $n^{-1}\alpha C^*$ is asymptotic covariance of the update $\boldsymbol{\theta}^k$ from the SGD sequence around its mean $\bar{\boldsymbol{\theta}}^k$.

Interestingly, reducing $\mathrm{tr}[C^*H^*]$ essentially amounts to reducing $\mathrm{tr}\,B^*$ with $B^*$ defined in (13). Indeed, to leading order in $\alpha$, we see from (12) that

$$\mathrm{tr}[C^*H^*] = \tfrac{1}{2}\,\mathrm{tr}\,B^* + O(\alpha) \tag{15}$$

Reducing $\mathrm{tr}\,B^*$ is precisely what we show how to do next using importance sampling.

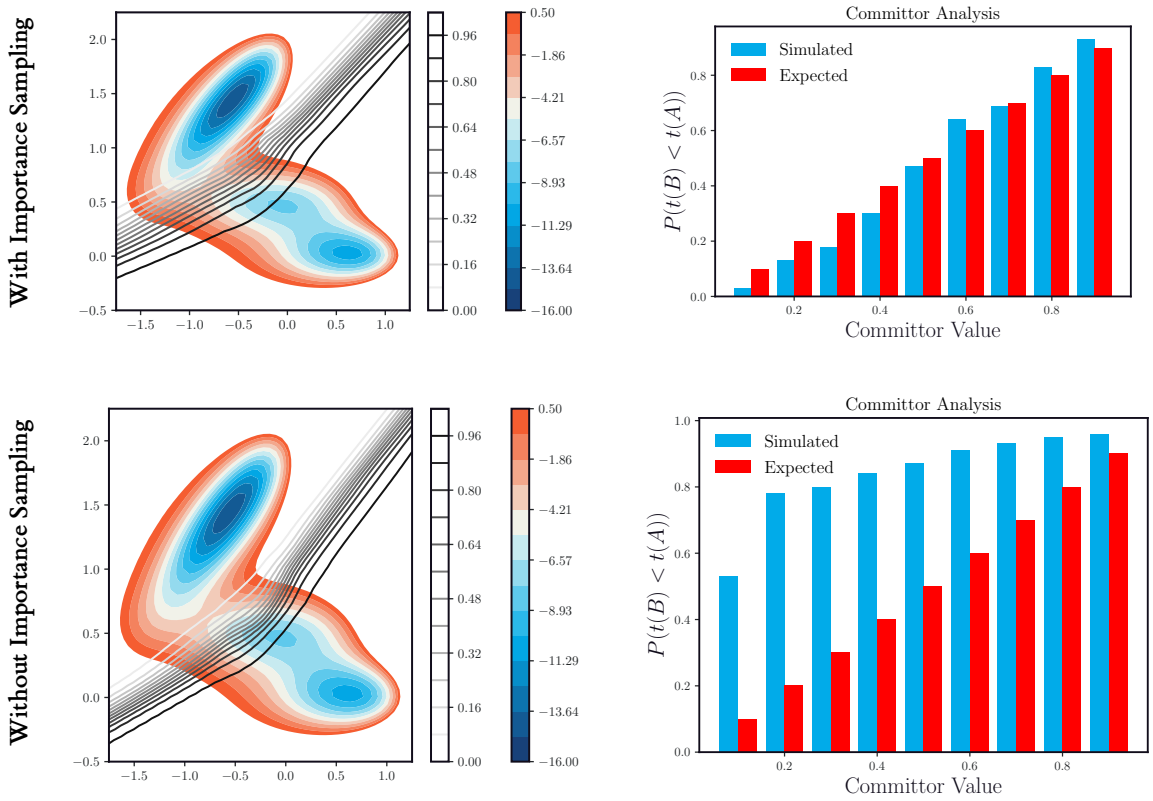


Figure 2: Simple illustrative experiment; the potential energy function is a 2D mixture of Gaussians and the committor is represented as a single hidden layer neural network. Top (results using our active importance sampling algorithm): Filled contours illustrate the Müller-Brown potential (38) with the isocommittor lines are shown from white to black. Notably, level set $q = 0.5$ coincides with the saddle, as expected. Right: We verify that this solution is consistent with the expected committor function by sampling 100 configurations for each window $0.1, 0.2, \ldots$ and running Langevin trajectories to compute estimate the committor probability. The fraction of trajectories reaching $B$ before $A$ is close to the expected value. Bottom: Without importance sampling, the optimization converges to a representation of the committor $q$ with poor performance. Left: The contours of the committor fail to localize to the transition region. Right: The committor analysis shows that without importance sampling, the results deviate strongly from the expected probabilities.

## 3. Active learning with umbrella sampling and replica exchange

To reduce the variance of the estimator for $L$, we will use an importance sampling strategy that combines umbrella sampling Torrie and Valleau (1977) (cf. stratification Dinner et al. (2020)) with replica exchange Swendsen and Wang (1986); Fukunishi et al. (2002). The first method uses windowing functions to enhance the sampling in otherwise rarely sampled regions of the data distribution; the second allows for exchange between these windows to accelerate sampling even further. Both these methods are widely used: the novelty lies in the way we actively define the windowing functions using the current estimate of the target function $f$ by its network representation $f(\cdot, \boldsymbol{\theta})$. In Appendix C we discuss an active importance sampling scheme based on direct reweighting which could be used to reduce the variance in the estimate of the gradient of the loss.

### 3.1. Umbrella sampling

Let us denote these windowing functions as a set of non-negative functions $W_l(\boldsymbol{x}) \geq 0$ with $l = 1, \ldots, L$ such that

$$\forall \boldsymbol{x} \in \Omega \quad : \quad \sum_{l=1}^{L} W_l(\boldsymbol{x}) = 1, \tag{16}$$

Given any test function $\phi : \Omega \to \mathbb{R}$, we can write

$$\mathbb{E}_\nu \phi = \sum_{l=1}^{L} \int_{\mathbb{R}^d} \phi(\boldsymbol{x}) W_l(\boldsymbol{x}) d\nu(\boldsymbol{x}) \equiv \sum_{l=1}^{L} w_l \, \mathbb{E}_l \phi \tag{17}$$

where we defined the expectation

$$\mathbb{E}_l \phi = Z_l^{-1} \int_{\mathbb{R}^d} \phi(\boldsymbol{x}) W_l(\boldsymbol{x}) d\nu(\boldsymbol{x}) \quad \text{where} \quad Z_l = \int_{\mathbb{R}^d} W_l(\boldsymbol{x}) d\nu(\boldsymbol{x}) \tag{18}$$

as well as the weights

$$w_l = \mathbb{E}_\nu W_l. \tag{19}$$

By choosing $\phi(\boldsymbol{x}) = W_{l'}(\boldsymbol{x})$ in this expression, we deduce that the weights satisfy the eigenvalue problem Thiede et al. (2016)

$$w_{l'} = \sum_{l=1}^{L} w_l p_{ll'}, \quad l' = 1, \ldots, L, \quad \text{subject to} \quad \sum_{l=1}^{L} w_l = 1, \tag{20}$$

where we defined

$$p_{ll'} = \langle W_{l'} \rangle_l \tag{21}$$

In practice, we can sample $Z_l^{-1} W_l(\boldsymbol{x}) d\nu(\boldsymbol{x})$ by Metropolis-Hastings Monte-Carlo on a potential biased by $-\log W_l(\boldsymbol{x})$ and compute expectations in this ensemble as

$$\mathbb{E}_l \phi \approx \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_{i,l}), \qquad \boldsymbol{x}_{i,l} \sim Z_l^{-1} W_l(\boldsymbol{x}) d\nu(\boldsymbol{x}) \tag{22}$$

This allows us to estimate $\mathbb{E}_l \phi$ in (20) as well as $p_{ll'}$ in (21): knowledge of the latter quantity enables us to solve the eigenvalue problem in (20) to find the weights $w_l$, and finally estimate $\mathbb{E}_\nu \phi$ via (17).

### 3.2. Replica Exchange

The sampling can be accelerated by using replica exchange between the ensembles in the different windows, which alleviates potential problems due to metastability within these windows. This amounts to constructing a stochastic process on the extended phase space $\Omega^L \times P_L$, where $P_L$ is the permutation group of the $L$ replica, such that its joint equilibrium measure on this extended space is given by

$$d\nu_\sigma(d\boldsymbol{x}_1, \dots, d\boldsymbol{x}_L) = \frac{1}{L!} \prod_{l=1}^{L} Z_l^{-1} W_{\sigma(l)}(\boldsymbol{x}_l) d\nu(\boldsymbol{x}_l) \tag{23}$$

where $\sigma \in P_L$ denotes a permutation of the indices $\{1, 2, \dots, L\}$ and $\sigma(l)$ is the index on which $l$ is mapped by this permutation. A process in detailed-balance with respect to $\nu_\sigma$ can be implemented e.g. in the form of a stochastic switching process Lu and Vanden-Eijnden (2019). In practice, we sample the measures $Z_l^{-1} W_l(\boldsymbol{x}) d\nu(\boldsymbol{x})$ independently in each windows for intervals of duration $t_{\text{swap}}$, then attempt to exchange configurations in two windows $l$ and $m \neq l$ uniformly picked in $\{1, \dots, L\}$ and $\{1, \dots, L\} \setminus \{l\}$ respectively, and accept this move with a Metropolis acceptance probability consistent with (23). Assuming that $d\nu(\boldsymbol{x}) = Z^{-1} \exp(-V(\boldsymbol{x})) d\boldsymbol{x}$ where $V : \Omega \to \mathbb{R}$ is some potential and $Z = \int_\Omega \exp(-V(\boldsymbol{x})) d\boldsymbol{x}$, and denoting $V_l(\boldsymbol{x}) = V(\boldsymbol{x}) - \log W_l(\boldsymbol{x})$, this acceptance probability reads

$$\text{acc}(\boldsymbol{x}_l, \boldsymbol{x}_m) = \min \left[ 1, \exp \left( - V_l(\boldsymbol{x}_m) - V_m(\boldsymbol{x}_l) + V_l(\boldsymbol{x}_l) + V_m(\boldsymbol{x}_m) \right) \right]. \tag{24}$$

### 3.3. Adaptive choice of the windows

As of yet, we have not specified the windowing functions $W_l(\boldsymbol{x})$. Because the $W_l(\boldsymbol{x})$ determine where the samples concentrate a "good" choice of these functions is crucial for the success of the sampling scheme. Here we propose to make this choice adaptive to the function $f(\boldsymbol{x}, \boldsymbol{\theta})$ that is being optimized, by dividing space into regions where $f(\boldsymbol{x}, \boldsymbol{\theta})$ takes specific values. To this end, let

$$\sigma(u) = \frac{1}{1 + e^{-u}} \tag{25}$$

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

$$W_l(\boldsymbol{x}) = \sigma\left(k(f(\boldsymbol{x}, \boldsymbol{\theta}) - u_{l-1})\right) - \sigma\left(k(f(\boldsymbol{x}, \boldsymbol{\theta}) - u_l)\right), \qquad l = 1, \dots, L \tag{26}$$

In the applications considered below $f$ is a probability and hence its range is restricted to $[0, 1]$. As a result we have

$$\begin{aligned} \forall \boldsymbol{x} \in \Omega \quad : \quad \sum_{l=1}^{L} W_l(\boldsymbol{x}) &= \sigma\left(k(f(\boldsymbol{x}, \boldsymbol{\theta}) - u_0)\right) - \sigma\left(k(f(\boldsymbol{x}, \boldsymbol{\theta}) - u_L)\right) \\ &\geq \sigma(-k u_0) - \sigma(k(1 - u_L)) \end{aligned} \tag{27}$$

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 non-negative functions $W_l(\boldsymbol{x})$ can be made to satisfy (16) to arbitrary precision exponentially fast in $ak$. The functions $W_l(\boldsymbol{x})$ are also peaked around $f(\boldsymbol{x}, \theta) = \frac{1}{2}(u_l + u_{l-1})$ 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

**Algorithm 1:** Importance Sampled Variational Stochastic Gradient Descent.

**Data:** Lagrangian $\ell(\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{x}, f(\cdot, \boldsymbol{\theta}))$, initial $\boldsymbol{\theta}$, $n \in \mathbb{N}$, $L \in \mathbb{N}$, $\alpha > 0$, $k > 0$, $u_0 < \ldots < u_L$.

**while** $\nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta}) > \epsilon_{\text{tol}}$ **do**

  **for** $l = 1, \ldots, L$ **do**

    **for** $i = 1, \ldots, n$ **do**

      Sample $\boldsymbol{x}_{i,l} \sim Z_l^{-1} W_l(\boldsymbol{x}) d\nu(\boldsymbol{x})$;

      Propose replica swaps;

    **end**

    Compute

$$p_{l,l'} = \frac{1}{n} \sum_{i=1}^{n} W_{l'}(\boldsymbol{x}_{i,l}) \qquad \text{for } l' = 1, \ldots, L, \quad \text{and}$$

$$\boldsymbol{G}_l[\boldsymbol{\theta}] = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_{i,l}, \boldsymbol{\theta})$$

  **end**

  Solve (20) for $w_l$, $l = 1, \ldots, L$;

  Compute

$$\nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta}) = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{G}_l(\boldsymbol{\theta}) w_l$$

  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta})$;

**end**

**Result:** $\boldsymbol{\theta}$

possible values for $f(\boldsymbol{x}, \boldsymbol{\theta})$. The $u_l$ can be spaced linearly, or, to concentrate sampling near the rapidly varying part of the committor function can be spaced geometrically away from $u_l = 1/2$.

An explicit scheme putting these steps together with SGD is described in Algorithm 1, where we provide a description of the most straightforward implementation of our approach. Algorithm 1 is sequential; a version in which we evolve $\boldsymbol{x}_{i,l}$ and $\boldsymbol{\theta}$ concurrently would allow for significant wallclock speed-ups.

## 4. Application: High-dimensional Backward Kolmogorov Equations (BKE)

Within the framework of statistical mechanics, the evolution of complex physical systems can be described by probability distributions and expectations that solve partial differential equations like the Fokker-Planck equation or the backward Kolmogorov equation (BKE). Because systems of practical interest are often high-dimensional, these PDEs are typically not solved directly—rather we resort to Monte-Carlo sampling methods or molecular dynamics simulations to estimate the system distribution. Our aim here is to investigate whether we can bypass these sampling methods, and go back to solving the relevant PDEs, using tools from ML.

### 4.1. The metastability problem

We will focus on one specific problem often encountered in practice: how to analyze the dynamics of systems displaying metastability—i.e. evolution that occurs on a wide range of very different time-scales. Consider in particular a physical system with coordinates $\boldsymbol{X}_t \in \mathbb{R}^d$ whose evolution is governed by the Langevin equation

$$d\boldsymbol{X}_t = -\nabla V(\boldsymbol{X}_t)dt + \sqrt{2\beta^{-1}}d\boldsymbol{W}_t. \tag{28}$$

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, and $\boldsymbol{W}_t$ is a Wiener process. This dynamics, or its variant with momentum included, 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 Yaida (2018) and sampling-based optimization schemes Ma et al. (2019).

In the context of a system with a dynamics governed by (28), metastability arises when the system remains confined in some region of its phase space for very long periods of time and seldom makes a transition to another such region. In general, it is not possible to directly observe the transitions between these metastable states using trajectories generated by (28) because the state space is very high dimensional in nontrivial cases and these transitions are by definition very infrequent.

Solving a high-dimensional PDE offers an alternative, in principle. Indeed metastability can be characterized mathematically as the property that the spectrum of the infinitesimal generator associated with (28) has a "spectral gap" between a set of low-lying eigenvalues with small magnitude compared to the rest of them—these low lying eigenvalues specify the rates of transition between metastable states, while the associated eigenfunctions describe their mechanism Bovier et al. (2002); Gaveau and Schulman (1998). The eigenvalue/eigenfunction pairs solve the minimization problem

$$\lambda_k = \min_{\varphi} Z^{-1} \int_{\mathbb{R}^d} |\nabla \varphi_k(\boldsymbol{x})|^2 e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x}, \qquad k \in \mathbb{N}_0 \tag{29}$$

where $Z = \int_{\mathbb{R}^d} e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x}$ and successive eigenfunctions are obtained by requiring that they be orthonormal to the previous ones: starting from $\varphi_0 = 1$ with $\lambda_0 = 0$, for $k \in \mathbb{N}$, we impose

$$Z^{-1} \int_{\mathbb{R}^d} \varphi_k(\boldsymbol{x}) \varphi_{k'}(\boldsymbol{x}) e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} = \delta_{k,k'} \quad \text{for } k' = 0, \ldots, k \tag{30}$$

This gives $0 = \lambda_0 < \lambda_1 < \cdots$.

While the minimization problem in (29) fits the framework of (1), in complex systems there may be hundreds or thousands of metastable states, and only a few of them are actually relevant Cameron and Vanden-Eijnden (2014). In this context, it is preferable to focus on one transition of interest at a time. This can be achieved using the potential theoretic framework to metastability Bovier et al. (2002); Bovier (2006) or transition path theory (TPT) E and Vanden-Eijnden (2006, 2010), and this is the approach we will focus on next.

### 4.2. Potential approach via BKE

Suppose we want to quantify the average rate and mechanism by which the solution to the Langevin equation (28) makes a transition from a state $A \subset \mathbb{R}^d$ to a distinct state $B \subset \mathbb{R}^d$. This can be

done by calculating the "committor function" $q : \mathbb{R}^d \to [0,1]$, which gives the probability that a trajectory starting at $\boldsymbol{x}$ first reaches $B$ before $A$:

$$q(\boldsymbol{x}) := \mathbb{P}^{\boldsymbol{x}}(t_B < t_A) \tag{31}$$

where $t_A = \inf\{t : x(t) \in A\}$ and similarly for $t_B$. Under the dynamics (28), the committor $q(\boldsymbol{x})$ solves the backward Kolmogorov equation

$$\begin{cases} (Lq)(\boldsymbol{x}) = 0 & \text{for } \boldsymbol{x} \notin A \cup B \\ q(\boldsymbol{x}) = 0 & \text{for } \boldsymbol{x} \in A \\ q(\boldsymbol{x}) = 1 & \text{for } \boldsymbol{x} \in B. \end{cases} \tag{32}$$

where $-L$ is the infinitesimal generator of the process defined by (28):

$$Lq = \nabla V \cdot \nabla q - \beta^{-1} \Delta q. \tag{33}$$

It can be shown that, with appropriate choice of $A$ and $B$, $q(\boldsymbol{x})$ can be asymptotically related to a eigenfunction $\varphi_k$ in the low-lying part of the spectrum as $\varphi_k(\boldsymbol{x}) = aq(\boldsymbol{x}) + b$ for some appropriate choice of $a$ and $b$—we refer the interested reader to Bovier (2006) for details. Here we will focus on using the active learning method we propose to solve the backward Kolmogorov equation in (32) in high dimension, i.e. in a setup where we would not be able to solve it using classical numerical PDE methods such as the finite element method. Specifically, our goal in the next sections is to define a parametric representation of the committor function by a neural network and an objective function that enables us to optimize the parameters in this network via active learning with importance sampling.

### 4.3. Variational loss functions for learning the committor

The committor satisfies a Ritz-type variational principle (3) that the can be employed directly as an objective function: That is, the solution to the BKE (32) is the minimizer of

$$\inf_q C(q) \quad \text{subject to } \; q = 0 \; \text{ in } A \; \text{ and } \; q = 1 \; \text{ in } B \tag{34}$$

where

$$C(q) = \int_{\mathbb{R}^d} |\nabla q(\boldsymbol{x})|^2 d\nu(\boldsymbol{x}) \qquad \text{with} \quad d\nu(\boldsymbol{x}) = Z^{-1} e^{-\beta V(\boldsymbol{x})} d\boldsymbol{x} \tag{35}$$

In the optimization procedure below, it is more tractable to penalize deviations from the boundary conditions rather than impose them as constraints. Consequently, we add penalty terms in (35) to ensure that the committor has the right values on $A$ and $B$ and the objective function we will use is

$$C_\lambda(q) = \int_{\mathbb{R}^d} |\nabla q(\boldsymbol{x})|^2 d\nu(\boldsymbol{x}) + \lambda \int_A |q(\boldsymbol{x})|^2 d\nu(\boldsymbol{x}) + \lambda \int_B |1 - q(\boldsymbol{x})|^2 d\nu(\boldsymbol{x}) \tag{36}$$

where $\lambda > 0$ is an adjustable parameter. This objective function is of the type in (2) with

$$\mathcal{L}(\boldsymbol{x}, q) = |\nabla q(\boldsymbol{x})|^2 + \lambda |q(\boldsymbol{x})|^2 1_A(\boldsymbol{x}) + \lambda |1 - q(\boldsymbol{x})|^2 1_B(\boldsymbol{x}) \tag{37}$$

where $1_A(\boldsymbol{x})$ and $1_B(\boldsymbol{x})$ denote the indicator functions of sets $A$ and $B$, respectively.

As discussed above, it is natural to model the minimizer of this cost functional as a neural network. Given some representation $f(\cdot, \boldsymbol{\theta})$ with parameter set $\{\boldsymbol{\theta}\}_{i=1}^{n}$, the problem becomes to minimize $C_\lambda$ over this set. We discuss the specific architectures that we use in applications below, but any neural network architecture is admissible within in this scheme, provided that it gives an output in the range $[0, 1]$, which is simple to achieve in practice by passing the output through a sigmoidal function (Appendix D). Even this condition can be relaxed: We describe an alternative formulation of the committor (cf. Lu and Vanden-Eijnden (2014)) in Appendix E which can be solved with distinct boundary conditions. The scheme we have described here could be implemented using symmetry functions or collective variables, which we leave for future work.

### 4.4. Numerical Experiments

#### 4.4.1. MÜLLER-BROWN POTENTIAL

As a proof of concept, we optimize the committor function on the well-studied Müller-Brown potential Müller and Brown (1979). We consider the dynamics (28) for a 2D system evolving in a Gaussian mixture potential

$$V_{\mathrm{MB}}(\boldsymbol{x}) = \sum_{k=1}^{4} A_k \exp\left(-(\boldsymbol{x} - \nu_k)^T \Sigma_k^{-1} (\boldsymbol{x} - \nu_k)\right) \tag{38}$$

with

$$\begin{aligned} A &= (-200, -100, -170, 15) \\ \nu &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}, \begin{pmatrix} -0.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \\ \Sigma^{-1} &= \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \begin{pmatrix} 6.5 & -5.5 \\ -5.5 & 6.5 \end{pmatrix}, \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \end{aligned} \tag{39}$$

Our results, shown in Fig. 2, demonstrate the importance sampling is required to converge a robust estimate of the committor.

While the contours of the committor provide a reasonable guide, a "committor analysis" gives a more precise test of convergence. To carry this analysis out, we sampled 100 distinct initial configurations from each window (where $q = 0.1, 0.2, \dots$) and then ran unbiased Langevin dynamics to compute $\min(t(A), t(B))$. Histograms of this calculation show that active importance sampling leads to trajectories that reach $B$ before $A$ with the expected probabilities. However, without importance sampling estimate of $q$ performs poorly.

We represent the committor as a single-hidden layer ReLU network with $m = 100$ units. The output of the ReLU network is passed through a sigmoidal function to compress the range because the committor represents a probability. To initialize the representation, we take a discretized linear interpolation between the center of basins $A$ and $B$ and optimize the representation to match the normalized distance along this path. At each optimization step, we collect 50 samples from each of the 10 windows in $q$-space. We use this sample to estimate the gradient after reweighting, as described above. We run the optimization for a total of 1000 optimization steps using 50 samples per window per optimization step with a restraint of $k = 100$ in the windowing function.

To make a systematic comparison, we ran a control experiment in which we used a single unbiased trajectory (that is, no importance sampling) to carry out the optimization. The total number of samples from this trajectory (10000 optimization steps with 50 samples per step) was chosen to

be equal to the total amount of data collected in our importance sampling optimization. As shown in Fig. 2, this approach does not succeed.

#### 4.4.2. ALLEN-CAHN-TYPE SYSTEM

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 will consider the following example building on a discretized version of the Allen-Cahn equation in two-dimension. Let us start from

$$\partial_t \rho = D\Delta\rho + \rho - \rho^3, \qquad \rho : [0,\infty) \times [0,1]^2 \to \mathbb{R} \tag{40}$$

with the Dirichlet boundary conditions

$$\rho(t, z_1, z_2) = +1, \quad \text{for } z_1 = 0, 1, \qquad \rho(t, z_1, z_2) = -1, \quad \text{for } z_2 = 0, 1, \tag{41}$$

The Allen-Cahn equation is the gradient flow in $L^2$ over the energy functional

$$E(\rho) = \int_{[0,1]^2} \left(\tfrac{1}{2} D|\nabla\rho(\boldsymbol{z})|^2 + \tfrac{1}{4}(1 - |\rho(\boldsymbol{z})|^2)^2\right) d\boldsymbol{z} \tag{42}$$

If we take $D$ small enough, this energy admits two minimizers, which are also the stable fixed points of (40) that solve

$$D\Delta\rho + \rho - \rho^3 = 0 \tag{43}$$

These fixed points 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. 3.

To build the model that we will actually study, let us discretize (40) on a lattice with spacing $h = 1/(N-1)$ and introduce

$$\rho_{i,j} = \rho(ih, jh), \qquad i, j = 1, \ldots, N \tag{44}$$

We also add some additive noise to the discretized equation to arrive at the Langevin equation

$$d\rho_{i,j} = \left(\rho_{i,j} - \rho_{i,j}^3 + D(\Delta_N \rho)_{i,j}\right) dt + \sqrt{2\beta^{-1}} h^{-1} dW_{i,j} \tag{45}$$

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

$$(\Delta_N \rho)_{i,j} = h^{-2} \left(\rho_{i+1,j} + \rho_{i-1,j} + \rho_{i,j+1} + \rho_{i,j-1} - 4\rho_{i,j}\right), \tag{46}$$

and the boundary conditions read

$$\begin{aligned} \rho_{i,j} &= 1, \qquad \text{for } i = 0, N+1, \quad j = 1, \ldots, N, \\ \rho_{i,j} &= -1, \qquad \text{for } j = 0, N+1, \quad i = 1, \ldots, N. \end{aligned} \tag{47}$$

We also set $\rho_{0,0} = \rho_{0,N+1} = \rho_{N+1,0} = \rho_{N+1,N+1} = 0$.

If we take $D$ and $\beta^{-1}$ small enough, the Langevin equation (45) displays metastability: the solution stays confined for long period of times in regions near the fixed points of the deterministic

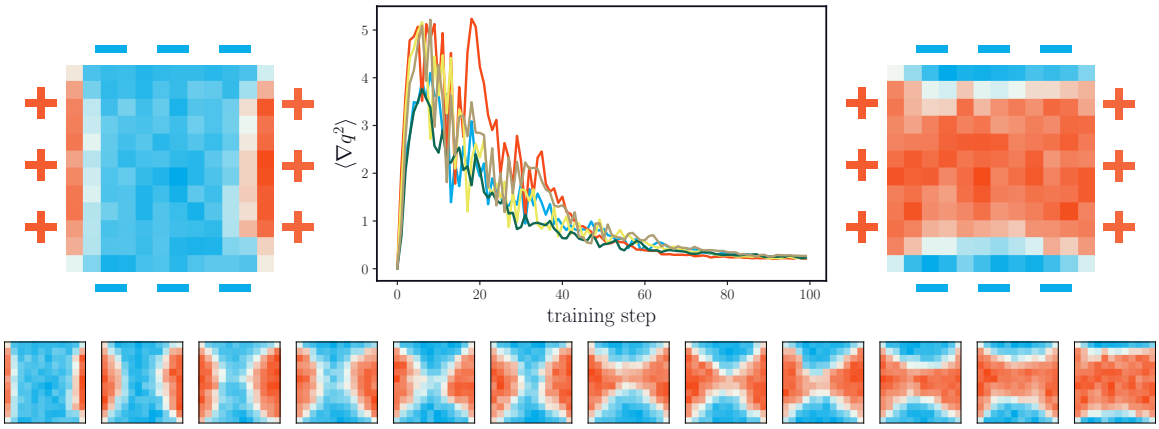


Figure 3: Top left and right: The two metastable solutions of (45) with Dirichlet boundary conditions ($\rho = 1$ at the left and right boundaries, $\rho = -1$ at the top and bottom boundaries). Top center: Decay of the loss of as a function of training step for 10 runs of the optimization with random 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.

equation (obtained by setting $\beta^{-1} = 0$ in (45)) and only rarely make transition between these regions. This can be confirmed by looking at the equilibrium distribution of (45):

$$d\nu(\boldsymbol{\rho}) = Z^{-1} e^{-\beta V(\boldsymbol{\rho})} d\boldsymbol{\rho} \tag{48}$$

where we denote $\boldsymbol{\rho} = (\rho_{i,j})_{i,j=1}^{N}$, $Z = \int_{\mathbb{R}^{2N-2}} e^{-\beta V(\boldsymbol{\rho})} d\boldsymbol{\rho}$ and the potential $V(\boldsymbol{\rho})$ is the discrete equivalent to (42):

$$V(\boldsymbol{\rho}) = h^{-2} \sum_{i,j=1}^{N} \left( \tfrac{1}{2} D |(\nabla_N \rho)_{i,j}|^2 + \tfrac{1}{4} (1 - |\rho_{i,j}|^2)^2 \right) \tag{49}$$

where $\nabla_N$ is the discrete gradient so that

$$|(\nabla_N \rho)_{i,j}|^2 = h^{-2} \left( \rho_{i+1,j} - \rho_{i,j} \right)^2 + h^{-2} \left( \rho_{i,j+1} - \rho_{i,j} \right)^2 . \tag{50}$$

For small $\beta^{-1}$ the distribution (48) is nearly atomic on the two minimizers of $V(\boldsymbol{\rho})$ shown in Fig. 3., and the question is how do rare transitions occur between these metastable state and at which average rate. This question can be answered by solving the BKE for the committor associated with (45)

We solved this problem using the active learning method outlined before in a situation where $N = 12$, i.e. the state space is $12^2 = 144$ dimensional. We use a single hidden layer ReLU network with $m = 100$ neurons which is passed through a sigmoidal function at the output layer to ensure that the range of $q$ is $(0, 1)$. The network is initialized by linearly interpolating homogeneous configurations in magnetization space, which provides no *a priori* information about the spatial structure of the transition path. We carried out the optimization with 12 total windows, including

the boundary windows, with a learning rate of $5 \times 10^{-2}$ for 5000 steps with 25 sampling steps per window.

As shown in Fig. 3, this shows the characteristic pathway for a transition between the two metastable states, with the expected hourglass shape as transition state Kohn et al. (2007) that can also be identified by the string method E et al. (2002, 2007) or the minimum action method in this specific example E et al. (2004); Heymann and Vanden-Eijnden (2008). 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.

## 5. Conclusion and 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.

In some sense, the promise of machine learning for solving committor equations can be conceptualized by interpreting these problems as classification problems. In the examples we consider, the primary task directly resembles binary classification in which the network is attempting to find a dividing surface between classes in a high dimensional space. The isocommittor surface is defined by the dynamical fate of points in this space and collecting data to adequately resolve the location of the boundary is typically impossible without importance sampling.

While our algorithm and code can easily employ any 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 Kearnes et al. (2016). 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.

The class of PDEs that we consider here could be generalized to include Ritz-type objectives with forcing terms, as well. Problems that are driven away from the equilibrium Gibbs distribution pose significant challenges for existing sampling techniques and represent an important target for future work.

## References

Hernan P. Awad, Peter W. Glynn, and Reuven Y. Rubinstein. Zero-Variance Importance Sampling Estimators for Markov Process Expectations. *Mathematics of Operations Research*, 38(2):358–388, May 2013. ISSN 0364-765X, 1526-5471. doi: 10.1287/moor.1120.0569.

Peter G Bolhuis, David Chandler, Christoph Dellago, and Phillip L Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annu. Rev. Phys. Chem.*, 53(1):291–318, 2002. doi: 10.1146/annurev.physchem.53.082301.113146.

Anton Bovier. Metastability: A potential theoretic approach. *Proceedings of the International Congress of Mathematicians*, page 20, 2006.

Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability and Low Lying Spectra in Reversible Markov Chains. *Communications in Mathematical Physics*, 228(2):219–255, June 2002. ISSN 0010-3616, 1432-0916. doi: 10.1007/s002200200609.

Maria Cameron and Eric Vanden-Eijnden. Flows in Complex Networks: Theory, Algorithms, and Application to Lennard–Jones Cluster Rearrangement. *Journal of Statistical Physics*, 156(3):427–454, August 2014. ISSN 0022-4715, 1572-9613. doi: 10.1007/s10955-014-0997-8.

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, and Laurent Daudet. Machine learning and the physical sciences*. *Rev. Mod. Phys.*, 91(4):39, 2019. doi: 10/ggd5qv.

Dominik Csiba and Peter Richtarik. Importance Sampling for Minibatches. *Journal of Machine Learning Research*, 19:21, 2018.

Aaron R. Dinner, Erik H. Thiede, Brian Van Koten, and Jonathan Weare. Stratification as a general variance reduction method for markov chain monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 8(3): 1139–1188, 2020. doi: 10.1137/18M122964X. URL https://doi.org/10.1137/18M122964X.

David L. Donoho and Iain M. Johnstone. Projection-based approximation and a duality with kernel methods. *Ann. Statist.*, 17(1):58–106, 03 1989. doi: 10.1214/aos/1176347004. URL https://doi.org/10.1214/aos/1176347004.

Weinan E and Eric Vanden-Eijnden. Towards a Theory of Transition Paths. *Journal of Statistical Physics*, 123(3):503–523, May 2006. ISSN 0022-4715, 1572-9613. doi: 10/b3pwgn.

Weinan E and Eric Vanden-Eijnden. Transition-Path Theory and Path-Finding Algorithms for the Study of Rare Events. *Annual Review of Physical Chemistry*, 61(1):391–420, March 2010. ISSN 0066-426X, 1545-1593. doi: 10.1146/annurev.physchem.040808.090412.

Weinan E and Bing Yu. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *arXiv:1710.00211 [cs, stat]*, September 2017.

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66(5):052301, August 2002. doi: 10.1103/PhysRevB.66.052301.

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. Minimum action method for the study of rare events. *Communications on pure and applied mathematics*, 57(5):637–656, 2004.

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. Finite temperature string method for the study of rare events. *J. Phys. Chem. B*, 109(14):6688–6693, 2005.

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *The Journal of Chemical Physics*, 126(16):164103, 2007. doi: 10.1063/1.2720838. URL https://doi.org/10.1063/1.2720838.

Martin Eigel, Reinhold Schneider, Philipp Trunschke, and Sebastian Wolf. Variational Monte Carlo—bridging concepts of machine learning and high-dimensional partial differential equations. *Advances in Computational Mathematics*, 45(5):2503–2532, December 2019. ISSN 1572-9044. doi: 10.1007/s10444-019-09723-8.

Yu Fan, Jing Xu, and Christian R Shelton. Importance Sampling for Continuous Time Bayesian Networks. *Journal of Machine Learning Research*, 11:2115–2140, 2010.

Hiroaki Fukunishi, Osamu Watanabe, and Shoji Takada. On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *The Journal of Chemical Physics*, 116(20):9058–9067, May 2002. ISSN 0021-9606. doi: 10.1063/1.1472510.

Bernard Gaveau and L. S. Schulman. Theory of nonequilibrium first-order phase transitions for stochastic dynamics. *Journal of Mathematical Physics*, 39(3):1517–1533, March 1998. ISSN 0022-2488, 1089-7658. doi: 10.1063/1.532394.

Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, December 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108929.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4349. doi: 10.1038/s41557-020-0544-y.

Matthias Heymann and Eric Vanden-Eijnden. The geometric minimum action method: A least action principle on the space of curves. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 61(8):1052–1117, 2008.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, August 2016. ISSN 1573-4951. doi: 10.1007/s10822-016-9938-8.

Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving for high dimensional committor functions using artificial neural networks. *arXiv*, February 2018.

Robert V Kohn, Felix Otto, Maria G Reznikoff, and Eric 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(3):393–438, 2007.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. doi: 10.1038/nature14539.

Qianxiao Li, Bo Lin, and Weiqing Ren. Computing Committor Functions for the Study of Rare Events Using Deep Learning. *The Journal of Chemical Physics*, 151(5):054112, August 2019. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.5110439.

Jianfeng Lu and Eric Vanden-Eijnden. Exact dynamical coarse-graining without time-scale separation. *The Journal of Chemical Physics*, 141(4):044109, July 2014. ISSN 0021-9606, 1089-7690. doi: 10/gf3xf4.

Jianfeng Lu and Eric Vanden-Eijnden. Methodological and computational aspects of parallel tempering methods in the infinite swapping limit. *Journal of Statistical Physics*, 174(3):715–733, 2019.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, October 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1820003116.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

Luca Maragliano, Alexander Fischer, Eric Vanden-Eijnden, and Giovanni Ciccotti. String method in collective variables: Minimum free energy paths and isocommittor surfaces. *J. Chem. Phys.*, 125(2):024106, July 2006. doi: 10.1063/1.2212942.

Klaus Müller and Leo D Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica chimica acta*, 53(1):75–93, 1979.

Yu. Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, January 2012. ISSN 1052-6234, 1095-7189. doi: 10.1137/100802001.

David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schr\"odinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, September 2020. doi: 10.1103/PhysRevResearch.2.033429.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence _Rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

Robert H. Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57:2607–2609, Nov 1986. doi: 10.1103/PhysRevLett.57.2607. URL https://link.aps.org/doi/10.1103/PhysRevLett.57.2607.

Erik H Thiede, Brian Van Koten, Jonathan Weare, and Aaron R Dinner. Eigenvector method for umbrella sampling enables error analysis. *The Journal of chemical physics*, 145(8):084115, 2016.

G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187 – 199, 1977. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(77)90121-8. URL http://www.sciencedirect.com/science/article/pii/0021999177901218.

Julien Toulouse, Roland Assaraf, and Cyrus J. Umrigar. Chapter Fifteen - Introduction to the Variational and Diffusion Monte Carlo Methods. In Philip E. Hoggan and Telhat Ozdogan, editors, *Electron Correlation in Molecules – Ab Initio Beyond Gaussian Quantum Chemistry*, volume 73 of *Advances in Quantum Chemistry*, pages 285–314. Academic Press, 2016. doi: 10.1016/bs.aiq.2015.07.003.

Sho Yaida. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv:1810.00004 [cs, stat]*, December 2018.

## Appendix A. One-dimensional example

To illustrate the necessity of importance sampling for objectives dominated by rare events, consider the one-dimensional committor problem associated with transitions between the minima located at $x = x_1$ and $x = x_2$ of the potential $V(x) = (1 - x^2) + x/10$, i.e. the minimization of

$$\int_{-1}^{1} |q'(x)|^2 e^{-\beta V(x)} dx \tag{A.1}$$

The minimizer of this objective function subject to $q(x_1) = 0$, $q(x_2) = 1$ is

$$q(x) = \frac{\int_{x_1}^{x} e^{\beta V(y)} dy}{\int_{x_1}^{x_2} e^{\beta V(y)} dy} \tag{A.2}$$

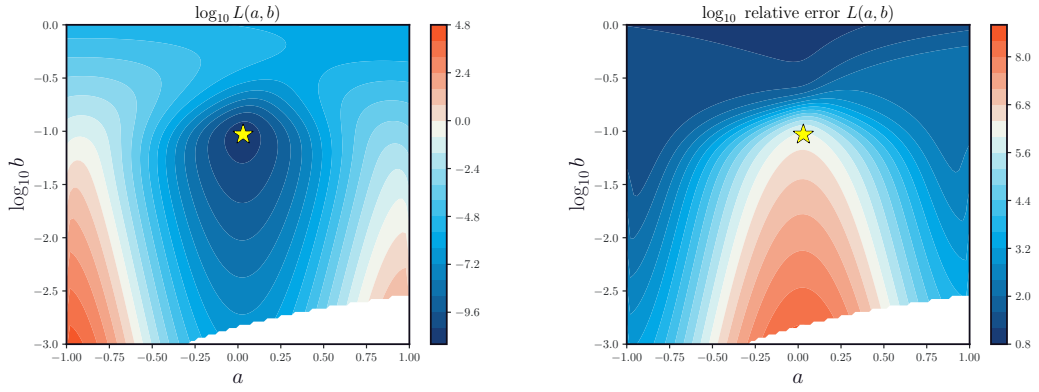


Figure 4: The loss landscape $L(a, b)$ in (A.4) and the relative error on the estimator ( = std/loss) when the data is drawn from the Gibbs distribution with density $Z^{-1}_{[x_1,x_2]} = e^{-\beta V(x)}$ restricted on $x \in [x_1, x_2]$. At the minimum of the loss (located at the red dot), this relative error is about 60. Here $\beta = 1/8$ (i.e. the energy barrier is $8k_BT$) and the optimal parameters are $a \approx 0.04$ and $b \approx 0.11$.

For large $\beta$, this function is sigmoid-like with a sharp transition from 0 to 1 around $x = 0.1$. Suppose that we want to approximate it using the parametric representation

$$q(x; a, y) = \sigma((x - a)/b) \qquad \text{where} \qquad \sigma(z) = \frac{1}{1 + e^{-z}} \tag{A.3}$$

This function does not satisfy the boundary condition exactly, but for $a$ around 0.1 and $b$ small enough, it does a good job at representing the exact (A.2) (see the top left panel in Fig. 1). Accordingly, let us look at the loss function as a function of $(a, b)$ in this parameter range, viewed as an expectation of $|q'(x; a, b)|$ the Gibbs distribution with density $e^{-\beta V(x)}$ restricted to $x \in [x_1, x_2]$ and properly normalized on that interval:

$$L(a, b) = Z^{-1}_{[-x_1,x_2]} \int_{x_1}^{x_2} |q'(x; a, b)|^2 e^{-\beta V(x)} dx \qquad \text{with} \qquad Z^{=}_{[-x_1,x_2]} \int_{x_1}^{x_2} e^{-\beta V(x)} dx \tag{A.4}$$

where

$$|q'(x; a, b)|^2 = b^{-2}\sigma^2((x - a)/b)(1 - \sigma((x - a)/b)) \tag{A.5}$$

The population and empirical losses were shown in in the bottom panels of Fig. 1: the latter was obtained by drawing $10^4$ independent samples from $Z^{-1}_{[-x_1,x_2]}e^{-\beta V(x)}$ using a rejection method, resulting in the empirical distribution shown in the top right panel of Fig. 1. Here we compute an additional quantity: the variance of the estimator for the population loss if we use data sampled from $Z^{-1}_{[x_1,x_2]}e^{-\beta V(x)}$. The result (together with the population loss) is shown in Fig. 4: when $beta$ is large so that the energy barrier is also large in units of $k_BT$ (here $\beta = 1/8$, so that the barrier is $8k_BT$), the relative error on the loss becomes large in the regions close to the minimum of this loss.

Note that in this one-dimensional example, adding a regularizing term to the empirical loss improves its predictions. However this strategy will not be generically applicable to higher dimensional situations.

## Appendix B. Variance reduction improves generalization

**Proof** [Proof of Proposition 1] Recall that the discrete time updates of the stochastic gradient descent dynamics are obtained from:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha\nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta}^k), \qquad k = 0, 1, 2, \dots \tag{B.1}$$

Since the miniibatches are draw independently at every step and $\nabla_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta})$ is an unbiased estimator of $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$, in law (B.1) is equivalent to

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^k) + \frac{\alpha}{\sqrt{n}}\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}^k), \qquad k = 0, 1, 2, \dots \tag{B.2}$$

where $\eta$ is a random function with mean zero, $\mathbb{E}_\nu \eta(\boldsymbol{\theta}) = 0$, and covariance

$$\mathbb{E}_\nu \eta(\boldsymbol{\theta})\eta(\boldsymbol{\theta}') = \mathbb{E}_\nu \ell(\boldsymbol{x}, \boldsymbol{\theta})\ell(\boldsymbol{x}, \boldsymbol{\theta}') - L(\boldsymbol{\theta})L(\boldsymbol{\theta}'). \tag{B.3}$$

Let us introduce $\tilde{\boldsymbol{\theta}}^k_n$ defined as

$$\tilde{\boldsymbol{\theta}}^k_n = \sqrt{\frac{n}{\alpha}}(\boldsymbol{\theta}^k - \bar{\boldsymbol{\theta}}^k) \tag{B.4}$$

where $\{\bar{\boldsymbol{\theta}}^k\}_{k\in\mathbb{N}_0}$ are the update from the GD scheme in (9) so that

$$\tilde{\boldsymbol{\theta}}^{k+1}_n = \tilde{\boldsymbol{\theta}}^k - \sqrt{\alpha n}\big(\nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k + \sqrt{\alpha/n}\tilde{\boldsymbol{\theta}}^k_n) - \nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k)\big) + \sqrt{\alpha}\nabla_{\boldsymbol{\theta}}\eta(\bar{\boldsymbol{\theta}}^k + \sqrt{\alpha/n}\tilde{\boldsymbol{\theta}}^k_n). \tag{B.5}$$

for $k = 0, 1, 2, \dots$. Taking the limit as $n \to \infty$ shows that for each $k$ $\tilde{\boldsymbol{\theta}}^k_n to \tilde{\boldsymbol{\theta}}^k$, where $\{\boldsymbol{\theta}^k\}_{k\in\mathbb{N}_0}$ is the solution of the updating scheme

$$\tilde{\boldsymbol{\theta}}^{k+1} = \tilde{\boldsymbol{\theta}}^k - \alpha H^k\tilde{\boldsymbol{\theta}}^k + \sqrt{\alpha}\boldsymbol{b}^k, \qquad k = 0, 1, 2, \dots \tag{B.6}$$

where $H^k = \nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k)$ and $\{\boldsymbol{b}^k\}_{k\in\mathbb{N}_0}$ are random vector, independent for different $k$, with mean zero and covariance

$$B^k = \mathbb{E}_\nu \boldsymbol{b}^k(\boldsymbol{b}^k)^T = \mathbb{E}_\nu \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \bar{\boldsymbol{\theta}}^k)(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \bar{\boldsymbol{\theta}}^k))^T - \nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k)(\nabla_{\boldsymbol{\theta}} L(\bar{\boldsymbol{\theta}}^k))^T \tag{B.7}$$

which we assume to be non-zero when the data set is finite.

Next note that the limiting sequence $\{\tilde{\boldsymbol{\theta}}^k\}_{k\in\mathbb{N}_0}$ can be used to deduce that

$$\lim_{n\to\infty} n\mathbb{E}_D[L(\boldsymbol{\theta}^k) - L(\bar{\boldsymbol{\theta}}^k)] = \tfrac{1}{2}\operatorname{tr}[C^k H^k] \qquad \text{where} \quad C^k = \mathbb{E}_D\tilde{\boldsymbol{\theta}}^k(\tilde{\boldsymbol{\theta}}^k)^T \tag{B.8}$$

From (B.6), the tensor $C^k$ satisfies

$$C^{k+1} = C^k - \alpha H^k C^k - \alpha C^k H^k + \alpha^2 H^k C^k H^k + \alpha B^k \tag{B.9}$$

with $C^0 = 0$ which follows from $\tilde{\boldsymbol{\theta}}^0 = 0$ since $\boldsymbol{\theta}^0 = \bar{\boldsymbol{\theta}}^0$. By Assumption 2, as $k \to \infty$, $H^k \to H^*$, which is the positive-definite tensor defined in (10), and $B^k \to B^*$, which is the tensor defined in (13). This guarantees that $\lim_{k\to\infty} C^k = C^*$, where $C^*$ is the solution to (12). From (B.8), it also implies that

$$\lim_{k\to\infty}\lim_{n\to\infty} n\mathbb{E}_D[L(\boldsymbol{\theta}^k) - L(\bar{\boldsymbol{\theta}}^k)] = \lim_{k\to\infty}\lim_{n\to\infty} n\mathbb{E}_D[L(\boldsymbol{\theta}^k) - L(\boldsymbol{\theta}^*)] = \tfrac{1}{2}\operatorname{tr}[C^* H^*] \tag{B.10}$$

which establishes (11) and ends the proof. ■

Note that, from (B.6), the $k$th iterate of $\tilde{\boldsymbol{\theta}}^k$ is (using $\tilde{\boldsymbol{\theta}}^0 = 0$ with follows from $\boldsymbol{\theta}^0 = \bar{\boldsymbol{\theta}}^0$)

$$\tilde{\boldsymbol{\theta}}^k = \sqrt{\alpha}\sum_{p=0}^{k-1}\prod_{q=0}^{p}(1 - \alpha H^q)\boldsymbol{b}^{k-p} \tag{B.11}$$

from which we can get more detailed information about the statistics of the sequence. Note also that, in the limit as $\alpha \to 0$, (B.6) reduces to an SDE similar to that of an Ornstein-Uhlenbeck process.

## Appendix C. Active sampling by reweighting

The results of Sec. 2 indicate that the variance of the estimator for the gradient of the population loss dominates the generalization error. In view of this, at every step of SGD, instead of sampling the original measure $\nu$, an option is to sample a modified measure $\tilde{\nu}$ and reweight the samples in the estimator accordingly, in such a way as to minimize the variance of this estimator. To make this concrete let $g(\boldsymbol{x}) = d\tilde{\nu}/d\nu$ be the Radon-Nikodym derivative of $\tilde{\nu}$ with respect to $\nu$, assume that $g$ is positive everywhere, and let $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n$ be a batch of independent samples draw from $\tilde{\nu}$. Then

$$\frac{1}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}}\ell(\tilde{\boldsymbol{x}}_i, \boldsymbol{\theta})g^{-1}(\boldsymbol{x}_i) \tag{C.1}$$

is an unbiased estimator for the gradient of population loss and the choice of $\tilde{\nu}$ that minimizes the variance of this estimator, i.e. minimizes

$$\int_\Omega |\nabla_{\boldsymbol{\theta}}\ell(\tilde{\boldsymbol{x}}, \boldsymbol{\theta})|^2 g^{-2}(\boldsymbol{x})d\tilde{\nu}(\boldsymbol{x}) = \int_\Omega |\nabla_{\boldsymbol{\theta}}\ell(\tilde{\boldsymbol{x}}_i, \boldsymbol{\theta})|^2 g^{-1}(\boldsymbol{x}_i)d\nu(\boldsymbol{x}), \tag{C.2}$$

is

$$d\tilde{\nu}(\boldsymbol{x}) = g(\boldsymbol{x})d\nu(\boldsymbol{x}) \qquad \text{with} \qquad g(\boldsymbol{x}) = \frac{|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x}, \boldsymbol{\theta})|}{\mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot, \boldsymbol{\theta})|} \tag{C.3}$$

An obvious difficulty with this estimator is that the reweighting factor $g(\boldsymbol{x})$ contains the factor $\mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot, \boldsymbol{\theta})|$ which we do not know. Still, in the context of optimization by SGD, it is useful since any unknown constant entering the gradient of the loss can be absorbed in the learning rate. To see why consider the following scheme: Starting from some initial value $\tilde{\boldsymbol{\theta}}^0$, update these parameters using the iteration rule

$$\tilde{\boldsymbol{\theta}}^{k+1} = \tilde{\boldsymbol{\theta}}^k - \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\nabla_{\boldsymbol{\theta}}\ell(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{\theta}}^k)}{|\nabla_{\boldsymbol{\theta}}\ell(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{\theta}}^k)|}, \qquad k = 0, 1, 2, \ldots \tag{C.4}$$

where the batch $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n$ contains independent samples from

$$d\tilde{\nu}_k(\boldsymbol{x}) = \tilde{Z}_k^{-1}|\nabla_\theta\ell(\boldsymbol{x}, \tilde{\boldsymbol{\theta}}^k)|d\nu(\boldsymbol{x}) \quad \text{with} \quad \tilde{Z}_k = \mathbb{E}_\nu|\nabla_\theta\ell(\cdot, \tilde{\boldsymbol{\theta}}^k)|. \tag{C.5}$$

Note that this measure can be sampled by the Metropolis-Hastings method or the Metropolis-adjusted Langevin algorithm without requiring to know its normalization factor $\tilde{Z}_k$. Under Assumption 2 we can prove the following equivalent of (11)

**Proposition C.1** *The sequence $\{\tilde{\boldsymbol{\theta}}^k\}_{k\in\mathbb{N}_0}$ obtained using the SGD update in (C.4) starting from $\tilde{\boldsymbol{\theta}}^0 = \bar{\boldsymbol{\theta}}^0$ and using an independent batch of data $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n$ drawn from $\tilde{\nu}^k$ at every step is such that*

$$\lim_{k\to\infty}\lim_{n\to\infty} n\mathbb{E}_D[\tilde{L}_n(\boldsymbol{\theta}^k) - L(\boldsymbol{\theta}^*)] = \tfrac{1}{2}\alpha\,\mathrm{tr}[\tilde{C}^* H^*] \tag{C.6}$$

*where $\mathbb{E}_D$ denotes expectation over all the batches used to compute the sequence $\boldsymbol{\theta}^k$, and $C^*$ is the $N\times N$ tensor that solves*

$$H^*\tilde{C}^* + \tilde{C}^* H^* - \alpha\tilde{C}^* H^*\tilde{C}^* = \tilde{B}^* \tag{C.7}$$

*Here $\tilde{B}^*$ is*

$$\tilde{B}^* = \int_\Omega \frac{\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x},\boldsymbol{\theta}^*)[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x},\boldsymbol{\theta}^*)]^T}{|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{x},\boldsymbol{\theta}^*)|^2} d\tilde{\nu}_*(\boldsymbol{x}) \tag{C.8}$$

*where*

$$d\tilde{\nu}_*(\boldsymbol{x}) = \tilde{Z}_*^{-1}|\nabla_\theta\ell(\boldsymbol{x},\boldsymbol{\theta}_*)|d\nu(\boldsymbol{x}) \quad \textit{with} \quad \tilde{Z}_* = \mathbb{E}_\nu|\nabla_\theta\ell(\cdot,\tilde{\boldsymbol{\theta}}_*)|. \tag{C.9}$$

The proof of this proposition is similar to that of Proposition 1. For small $\alpha$, this shows again that the error will be controlled by $\mathrm{tr}\,\tilde{B}^*$, which is now trivially given by

$$\mathrm{tr}\,\tilde{B}^* = 1 \tag{C.10}$$

This result may look surprising but it is a consequence of the fact that, by using (C.4) we have effectively absorbed in the learning rate the unknown factor $\mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot,\boldsymbol{\theta})|$ entering the weights $g(\boldsymbol{x})$ defined in (C.3). If we had not done this, $\mathrm{tr}\,\tilde{B}^*$ in (C.10) would be replaced by $|\mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot,\boldsymbol{\theta})||^2$; this provides a point of comparison with the scheme discussed in Proposition 1, since from (13) $\mathrm{tr}\,B^* = \mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot,\boldsymbol{\theta})|^2 \geq |\mathbb{E}_\nu|\nabla_{\boldsymbol{\theta}}\ell(\cdot,\boldsymbol{\theta})||^2$. Therefore we would reduce the variance.

Coming back to the scheme defined by (C.4), one feature that makes it somewhat academic is that we still need to sample $\tilde{\nu}_k$: while this can in principle be done via the Metropolis-Hastings method or the Metropolis-adjusted Langevin algorithm, we have no guarantees that this sampling will be fast—for example, even if $\tilde{\nu}$ has a density $\rho(\boldsymbol{x})$ with respect to the Hausdorff measure on $\Omega$, there is no guarantee that its potential $-\log\rho(\boldsymbol{x})$ will be convex or even that it will have a single minimum. For these reasons, we instead implement the alternative active importance sampling strategy based on umbrella sampling and replica exchange which we deem more robust and more widely applicable.

## Appendix D. Approximation of the committor with a neural network

### D.1. Representation

Neural networks (NN) offer flexibility to the representation and relative ease of optimization, making them a natural choice for a representation of the committor. For example, if we use a single hidden layer neural network with nonlinearity $\varphi$ (e.g., ReLU) passed through a thresholding function $\sigma$ (e.g., a sigmoid function, $\sigma(z) = 1/(1+e^{-z})$) to ensures that $q(\boldsymbol{x}) \in [0,1],\ \forall \boldsymbol{x}\in\mathbb{R}^d$, this amounts to taking

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

where we use $\boldsymbol{\theta}_i$ with $i = 1,\dots,n$ to denote the parameters in each neural units and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n)$ to denote all of them collectively. In practice, the architecture of the neural network will be substantially more intricate than the single hidden layer network (D.1).

### D.2. Computing the gradients

Optimization of the neural network representation of the committor (D.1) by gradient descent (GD) requires estimating the gradient of the objective function with respect to the parameters. For example, if we use (D.1) in the Lagrangian defined in (37), we have

$$\tfrac{1}{2}\nabla_{\boldsymbol{\theta}_i}\mathcal{L}(\boldsymbol{x},q) \equiv \tfrac{1}{2}\nabla_{\boldsymbol{\theta}_i}\ell(\boldsymbol{x},\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}_i}\nabla_{\boldsymbol{x}}q\nabla_{\boldsymbol{x}}q + \lambda q\nabla_{\boldsymbol{\theta}_i}q1_A - \lambda(1-q)\nabla_{\boldsymbol{\theta}_i}q1_B \tag{D.2}$$

Noting that, with $\sigma(z) = 1/(1+e^{-z})$,

$$\nabla_{\boldsymbol{x}}\sigma(f(\boldsymbol{x})) = \sigma(f(\boldsymbol{x}))\left(1-\sigma(f(\boldsymbol{x}))\right)\nabla_{\boldsymbol{x}}f(\boldsymbol{x}) \tag{D.3}$$

and similarly for $\nabla_{\boldsymbol{\theta}}$ we can derive explicit expressions for the factors at the right hand side of (D.2). In particular, we see that

$$\nabla_{\boldsymbol{x}}q(\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{n}q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\sum_{i=1}^{n}\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_{\phi}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i), \tag{D.4}$$

$$\nabla_{\boldsymbol{\theta}_i}q(\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{n}q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}_i}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i) \tag{D.5}$$

and

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i}\nabla_{\boldsymbol{x}}q(\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{n}\nabla_{\boldsymbol{\theta}_i}q(\boldsymbol{x},\boldsymbol{\theta})(1-2q(\boldsymbol{x},\boldsymbol{\theta}))\sum_{j=1}^{n}\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_{\phi}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_j)\\ + \frac{1}{n}q(\boldsymbol{x},\boldsymbol{\theta})(1-q(\boldsymbol{x},\boldsymbol{\theta}))\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\nabla_{\phi}\nabla_{\boldsymbol{\theta}_i}\varphi(\phi(\boldsymbol{x}),\boldsymbol{\theta}_i).\end{aligned} \tag{D.6}$$

## Appendix E. 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 Lu and Vanden-Eijnden (2014),

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

where $\tau > 0$ is a parameter, and $\delta(\boldsymbol{x}-\boldsymbol{a})$ and $\delta(\boldsymbol{x}-\boldsymbol{b})$ denote the Dirac delta distribution centered at $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively. Given a solution to (E.1), it is straightforward to verify that the committor between the sets

$$\begin{aligned}A &= \{\boldsymbol{x}|\tilde{q}(\boldsymbol{x}) \le \tilde{q}_-\} \ni \boldsymbol{a}\\ B &= \{\boldsymbol{x}|\tilde{q}(\boldsymbol{x}) \ge \tilde{q}_+\} \ni \boldsymbol{b}\end{aligned}$$

is given by

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

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

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

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

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, 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 two points $\boldsymbol{a} \in A$ and $\boldsymbol{b} \in B$.