Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

A DEEP LEARNING APPROACH FOR RARE EVENT SIMULATION IN DIFFUSION PROCESSES

Henrik Hult[1], Aastha Jain[2], Sandeep Juneja[3], Pierre Nyquist[4], and Sushant Vijayan[5]

[1]Department of Mathematics, KTH Royal Institute of Technology, Stockholm, SWEDEN
[2]Centre DLDS, Ashoka University, Haryana, INDIA
[3]Department of Computer Science, Ashoka University, Haryana, INDIA
[4]Department of Mathematical Sciences, Chalmers and University of Gothenburg, SWEDEN
[5]School of Technology and Computer Science, TIFR Mumbai, INDIA

ABSTRACT

We address the challenge of estimating rare events associated with stochastic differential equations using importance sampling. The importance sampling zero variance measure in these settings can be inferred from a solution to the Hamilton-Jacobi-Bellman partial differential equation (HJB-PDE) associated with a value function for the underlying process. Guided by this equation, we use a neural network to learn the zero variance change of measure. To improve performance of our estimation, we pursue two new ideas. First, we adopt a loss function that combines three objectives which collectively contribute to improving the performance of our estimator. Second, we embed our rare event problem into a sequence of problems with increasing rarity. We find that a well chosen schedule of rarity increase substantially speeds up rare event simulation. Our approach is illustrated on Brownian motion, Orstein-Uhlenbeck (OU) process, Cox–Ingersoll–Ross (CIR) process as well as Langevin double-well diffusion.

1 INTRODUCTION

Rare event estimation is of crucial importance in settings where the consequences of the rare event occurrence are catastrophic and/or of great significance. Standard Monte Carlo methods face either prohibitively high computational costs or poor accuracy when analyzing these events, owing to their extremely low probabilities. Variance reduction using importance sampling has found remarkable success in rare event estimation (Asmussen and Glynn 2007; Vanden-Eijnden and Weare 2012). Importance sampling is frequently utilized for estimating rare transition probabilities or functions thereof in diffusion processes (see (Juneja and Shahabuddin 2006) for an introduction). This involves simulating the process under an alternate measure, which accentuates paths leading to the rare event. The event is analysed under this new measure and the resulting output is unbiased using the likelihood ratio, which equals the Radon-Nikodym derivative of the original measure with respect to the new one. However, its effectiveness is often limited to well-structured models such as those involving random walks. For more complex processes, the success of importance sampling has been limited and there has been a strong need to extend the technique to more general settings. This paper contributes to that endeavour for rare events associated with diffusions.

The optimal measure in importance sampling minimizes the variance of the estimator, ideally to zero, by carefully assigning higher probabilities to the most likely paths to the rare event. This, when done properly, ensures unbiased and efficient estimates of the rare event probability. For continuous-time processes, estimating the optimal zero variance measure amounts to estimating the optimal control linked with a suitable value function related to the process. Approximately solving the HJB-PDEs linked with the diffusion processes enables the estimation of the value function (Pavliotis 2014), and consequently, the optimal control. However, solving HJB-PDEs, even approximately, is challenging and becomes intractable for high-dimensional processes. (Dupuis and Wang 2007) showed that subsolutions are sufficient for

designing efficient Monte Carlo methods linked to the HJB-PDEs. However, constructing subsolutions that give rise to efficient algorithms is a difficult task for most stochastic systems. As a result, there is growing interest in employing deep learning-based methods to approximately solve these PDEs (Han et al. 2017; Nakamura-Zimmerer et al. 2021). (Nüsken and Richter 2021) addressed rare events for diffusions by considering a performance measure associated with the zero variance measure satisfying an HJB-PDE. They demonstrated that efficiently estimating the solution to this PDE using deep learning yields a change of measure that performs well in rare event estimation. However, their approach fails to give good estimates for very small probabilities such as of order 10^{-8}. Building upon their methodology, our work employs an adaptive learning-based approach to efficiently estimate rare probabilities in different diffusion processes.

We embed the rare event problem into a sequence of problems with increasing rarity. At each stage, we estimate the optimal change of measure for the threshold corresponding to the current level of rarity. This optimal measure is estimated by approximately solving the associated HJB-PDE and identifying the minimizer of the variance of the rare event estimator. We use a novel loss function to minimize at each stage. This loss function comprises of three separate constraints imposed on the solution to the HJB-PDE. Subsequently, in the next level, we generate the SDE under this newly derived measure. Our approach demonstrates significant reductions in running times and the number of simulation runs needed to generate rare event estimates with lower relative error. We showcase the efficacy of our approach on Brownian motion, Ornstein-Uhlenbeck process, CIR process as well as Langevin double-well diffusion. We also empirically observe that at least for the double well diffusion, while our estimator is in the ball-park of the correct estimate (and better than that proposed by (Nüsken and Richter 2021)), it may differ from the true value by a constant. This suggests that further work is needed to make the proposed ideas more broadly applicable.

The rest of the paper is organized as follows. Section 2 gives background information on the problem. Section 3 outlines our adaptive-learning approach. Section 4 gives details about our numerical experiments. Section 5 presents the results. Section 6 offers concluding remarks.

2 PROBLEM FORMULATION

We consider the problem of computing the expectation $p_\varepsilon(a,T) = E[\exp\{-\varepsilon^{-1} f(X_T,a)\}]$ for some small $\varepsilon \geq 1$, where $\{X_t\}$ is the unique strong solution to an SDE of the form,

$$dX_t = b(X_t,t)dt + \sigma(X_t,t)\sqrt{\varepsilon}dB_t, \qquad X_0 = x_0, \tag{1}$$

where $b(X_t,t)$ is the drift coefficient driving the process X_t, $\sigma(X_t,t)$ is the diffusion coefficient, and B_t denotes the standard d-dimensional Brownian motion. According to (Oksendal 2013), we work under the following assumptions on b and σ to ensure strong unique solutions to the above SDE: $b(x,t) : R^d \times [0,T] \to R^d$ and $\sigma(x,t) : R^d \times [0,T] \to R^d$ are measurable functions satisfying

$$|b(x,t)| + |\sigma(x,t)| \leq C(1+|x|),$$
$$|b(x,t) - b(y,t)| + |\sigma(x,t) - \sigma(y,t)| \leq D|x-y|,$$

for some constants C and D. Here $f(X_T^\varepsilon, a)$ is defined such that $\exp\{-\varepsilon^{-1} f(X_T^\varepsilon, a)\}$ is concentrated around $x = a$. For instance, in case of Langevin double well diffusion, let the drift $b(X_t) = -\kappa\nabla_x(X_t^2 - 1)^2$, and we define $f(X_T) = (X_T - 1)^2$, so that $\exp\{-f(X_T)\}$ is close to 1 in the neighbourhood of $X_T = a = 1$. In this case, our performance measure is closely related to the probability of transition between the wells positioned at $X_0 = -1$ and $X_T = 1$, separated by a potential barrier of height κ. Through $f(X_T)$, we model the terminal cost incurred by the particle in reaching the state $X_T = a$ from a starting state $X_t = x$. More generally, to model the cost associated with the rare event problem of a particle crossing a large threshold a, we can define $f(X_T) = 0$ for $X_T \geq a$, and $f(X_T) \to \infty$ otherwise. This enables defining $\exp\{-f(X_T)\}$ as an indicator function, that takes the value 1 when the particle crosses the threshold, and 0 otherwise.

Consequently, the expectation in this framework aligns directly with the empirical probability estimate. The Monte-Carlo estimate of the expectation using N samples is given by:

$$\hat{p}_\varepsilon(a,T) = \frac{1}{N}\sum_{j=1}^{N} \exp\{-\varepsilon^{-1} f(X_{jT},a)\}$$

where each X_{jT} is an independent sample of X_T. The estimator is unbiased, i.e. $\mathbb{E}[\hat{p}_\varepsilon(a,T)] = p_\varepsilon(a,T)$. The relative error (defined as the ratio of standard deviation and probability) corresponding to the estimator is given as:

$$\delta(\hat{p_\varepsilon}) = \sqrt{\frac{\mathbb{E}[\exp\{-\frac{2}{\varepsilon} f(X_T)\}]}{\mathbb{E}[\exp\{-\frac{1}{\varepsilon} f(X_T)\}]^2} - 1}$$

To generate a reliable estimate of $\hat{p}_\varepsilon$, one needs a large number of samples $N(\propto \frac{1}{p_\varepsilon})$ if the event is rare. Since this is computationally expensive, the variance of the estimator tends to be relatively high for an extremely low probability value, which leads to the relative error blowing up, i.e. $\delta(\hat{p}_\varepsilon) \to \infty$ as $p_\varepsilon \to 0$. There is therefore a prohibitive computational cost associated with naive Monte Carlo estimation.

As mentioned in the introduction, importance sampling is commonly utilized for obtaining low-variance probability estimates in rare event scenarios. This involves simulating paths under an alternate measure, where the rare event happens more frequently and the resultant output can be unbiased using the likelihood ratio. Under the importance sampling measure $\mathbb{Q}^u$, the SDE and the associated Brownian motion will experience adjusted drifts. The system dynamics will now be driven by the control u, and the modified SDE can be written as:

$$\begin{aligned} dX_t^u &= (b(X_t^u,t) + \sigma(X_t^u,t)u(X_t^u,t)\sqrt{\varepsilon})dt + \sigma(X_t^u,t)\sqrt{\varepsilon}dB_t^u, \\ B_t^u &= B_t - \int_0^t u(X_t^u,t)dt \end{aligned} \tag{2}$$

Under the new measure $\mathbb{Q}^u$, B_t^u is the standard Brownian motion. The importance sampling estimator can be written as:

$$\hat{\rho}_\varepsilon(a,T) = \frac{1}{N}\sum_{j=1}^{N} \exp\{-\varepsilon^{-1} f(X_{jT}^u,a)\}\frac{d\mathbb{P}}{d\mathbb{Q}^u} \tag{3}$$

where X_t^u is the unique strong solution to the controlled SDE in Equation (2), and $\frac{d\mathbb{P}}{d\mathbb{Q}^u}$ is the Radon-Nikodym derivative obtained using the Girsanov theorem (Üstünel and Zakai 2013) as:

$$\frac{d\mathbb{P}}{d\mathbb{Q}^u} = \exp\left(-\int_0^T u(X_t,t)dB_t - \frac{1}{2}\int_0^T |u(X_t,t)|^2 dt\right)$$

Note that $\mathbb{E}_{\mathbb{Q}^u}[\hat{\rho}_\varepsilon(a,T)] = \mathbb{E}_{\mathbb{P}}[\hat{p}_\varepsilon(a,T)]$, i.e. $\hat{\rho}_\varepsilon(a,T)$ is also an unbiased estimator. The control u needs to be chosen such that the variance of the estimator under the measure $\mathbb{Q}^u$ is minimized. See (4), for $t=0, x=X_0$, the solution $u=u^*$ to the equation is unique and under u^*, the variance of $\hat{\rho}$ is 0 (Nüsken and Richter 2021). Finding the zero-variance change of measure is as difficult as finding the original performance measure, so in our algorithm, we resort to finding a control that approximately minimizes variance on empirically generated sample paths.

2.1 Optimal Control as Solution to HJB-PDE

Finding the optimal control u can be viewed through the optimization problem of minimizing the second moment of the probability estimate:

$$V^{\varepsilon}(t,x) = \inf_{u} \mathbb{E}_{\mathbb{P}} \left[\frac{d\mathbb{P}}{d\mathbb{Q}^u} e^{-\frac{2}{\varepsilon} f(X_T)} | X_t = x \right]. \tag{4}$$

This can be expressed as the value function for the underlying process X_t (Boué and Dupuis 1998), and therefore satisfies the associated HJB-PDE,

$$\inf_{u} \left[\partial_t V + V \frac{u^2}{\varepsilon} + (b + \sigma u \sqrt{\varepsilon}) \partial_x V + \frac{1}{2} \partial_{xx} V \sigma^2 \varepsilon \right] = 0.$$

Since V typically takes extremely small values, we use a more stable substitute for it by setting,

$$W^{\varepsilon}(t,x) = -\varepsilon \log V^{\varepsilon}(t,x). \tag{5}$$

With this notation $V^{\varepsilon}(t,x) = \exp\{-\varepsilon^{-1} W^{\varepsilon}(t,x)\}$, which implies that

$$\begin{aligned}
\partial_t V^{\varepsilon}(t,x) &= -\frac{1}{\varepsilon} V^{\varepsilon}(t,x) \partial_t W^{\varepsilon}(t,x), \\
\partial_x V^{\varepsilon}(t,x) &= -\frac{1}{\varepsilon} V^{\varepsilon}(t,x) \partial_x W^{\varepsilon}(t,x), \\
\partial_{xx} V^{\varepsilon}(t,x) &= -\frac{1}{\varepsilon} V^{\varepsilon}(t,x) \partial_{xx} W^{\varepsilon}(t,x) + \frac{1}{\varepsilon^2} V^{\varepsilon}(t,x) (\partial_x W^{\varepsilon}(t,x))^2.
\end{aligned}$$

This leads to the following PDE for W^{ε}:

$$\inf_{u} \left[u^2 - \partial_t W^{\varepsilon} - (b + \sigma u \sqrt{\varepsilon}) \partial_x W^{\varepsilon} + \frac{(\partial_x W^{\varepsilon})^2 \sigma^2}{2} - \frac{\varepsilon \sigma^2 \partial_{xx} W^{\varepsilon}}{2} \right] = 0,$$
$$W^{\varepsilon}(T,x) = 2f(x).$$

By minimizing pointwise over $u(t,x)$ we find that the optimal control is given by $u^* = \frac{\sigma \sqrt{\varepsilon} \partial_x W^{\varepsilon}}{2}$ and we arrive at the following PDE for W^{ε}:

$$-\partial_t W^{\varepsilon} - b \partial_x W^{\varepsilon} + \frac{(\partial_x W^{\varepsilon})^2 \sigma^2}{2} - \frac{\varepsilon \sigma^2 \partial_{xx} W^{\varepsilon}}{2} = 0, \tag{6}$$
$$W^{\varepsilon}(T,x) = 2f(x). \tag{7}$$

As $\varepsilon \to 0$, $W^{\varepsilon}(t,x) \to W(t,x)$, where W solves the limiting Hamilton-Jacobi equation:

$$\partial_t W + b \partial_x W - \frac{(\partial_x W)^2 \sigma^2}{2} = 0,$$
$$W(T,x) = 2f(x).$$

The above PDE can be solved approximately to estimate the optimal control u^*, that minimizes each $V^{\varepsilon}(t,x)$. This estimate can be used to arrive at a change of measure for efficiently estimating the expectation through importance sampling.

2.2 Estimating W Using a Neural Network

In order to obtain an approximation, we encode W^ε using a neural network. The change of measure $d\mathbb{Q}^u/d\mathbb{P}$ is then obtained via a Girsanov transformation, by taking the control $u^\varepsilon = \frac{\sigma\sqrt{\varepsilon}\partial_x W^\varepsilon}{2}$, and samples are generated under $\mathbb{Q}^u$. The network parameters (denoted by θ) are updated iteratively using a stochastic approximation scheme. This scheme involves optimizing a loss function that combines three objectives: minimizing the variance of the estimator, enforcing the HJB-PDE, and satisfying the terminal condition. Note that the second moment of the estimator is minimized by maximizing the initial value, $W^\varepsilon(0,x_0)$, or equivalently, minimizing $-W^\varepsilon(0,x_0)$. To enforce the HJB-PDE, it is particularly important to enforce it along the most likely paths leading to the rare event. To this end, we consider a term of the form:

$$\int_0^T \frac{1}{2}\left(\mathbb{E}_{\mathbb{Q}^u}[\mathscr{L}^\varepsilon W(t,X(t))^2]\right)dt = 0,$$

where $\mathscr{L}$ represents the infinitesimal generator involved in the HJB-PDE:

$$\mathscr{L}^\varepsilon W = -\partial_t W^\varepsilon - b\partial_x W^\varepsilon + \frac{(\partial_x W^\varepsilon)^2\sigma^2}{2} - \frac{\varepsilon\sigma^2\partial_{xx}W^\varepsilon}{2}$$

To satisfy the terminal condition, we aim to minimize $E_{\mathbb{Q}^u}\left[(W(T,X^\varepsilon) - 2f(X_T^\varepsilon))^2\right]$, and to minimize the variance of the estimator, we aim to maximize $W(0,x_0)$. We represent the loss function as a weighted combination of these three objectives:

$$L(\theta) = -k_1 W(0,x_0) + k_2\int_0^T \frac{1}{2}\left(\mathbb{E}_{\mathbb{Q}^u}[\mathscr{L}^\varepsilon W(t,X(t))^2]\right)dt + k_3 E_{\mathbb{Q}^u}\left[(W(T,X^\varepsilon) - 2f(X_T^\varepsilon))^2\right]. \tag{8}$$

We choose the weights k_1,k_2,k_3 using a discretized grid search over the possible ranges. This is explained in Section 5.2. In practice the expectations in the gradient of the loss function are approximated by samples and the time integral by a discrete sum. The parameters of the network are updated via repeated application of gradient descent with a learning rate η on the loss computed on batches of generated SDE paths, as follows:

$$\theta_{k+1} = \theta_k - \eta\nabla_\theta L(\theta_k).$$

3 PROPOSED ALGORITHM

Following the approach outlined in Section 2, we utilize importance sampling for estimating the probability of interest. The approximation to zero-variance change of measure is derived via a stochastic approximation scheme, optimizing a mixture of loss functions defined in (8).

Escalating level of rarity: Instead of solving the problem directly for a large threshold a, we adopt a sequential approach. We transform the original problem into a series of equivalent problems with progressively increasing levels of rarity, controlled by the value of ε. Starting with a large $\varepsilon = \varepsilon_0$, we iteratively decrease it until $\varepsilon = 1$. For each value of ε in the schedule, we determine the optimal change of measure using the stochastic approximation scheme described in Section 2. In our analysis in Section 5.3 it is more convenient to look at $\frac{1}{\varepsilon}$ instead of ε and consider an increasing schedule in $\frac{1}{\varepsilon}$. Our findings suggest that an optimal increase in schedule for $\frac{1}{\varepsilon}$ should be linear, meaning $\frac{1}{\varepsilon}$ increases from 0 to 1 in equal increments.

Stochastic Approximation: We use a neural network to approximate W^ε (defined in Equation 5). The initial layer of the network takes pairwise inputs (t,x), representing the position of particle x at time $t \le T$. The output layer yields $W^\varepsilon(t,x)$ corresponding to the input. We initialize all network parameters to 0 and start with $\varepsilon_0 = 10$. For the current ε, we generate N trajectories for the given process. These trajectories serve as pairwise input to the neural network. We define a number of epochs E and a batch size B, dividing the entire data into batches of size B. The empirical loss in Equation 8 is computed on each batch. We

use batch training to get stable convergence of the parameters and leverage parallel computations for a large dataset. Utilizing PyTorch's automatic differentiation tool, we efficiently compute terms in the loss function involving the derivative of output W^{ε} with respect to the input (x,t). Subsequently, we update the network parameters via gradient descent on the computed loss. We repeat this process for each ε over E epochs. Specifically, at each level i, we utilize the previously estimated optimal control u_{i-1} along with the current $\varepsilon = \varepsilon_i$ to generate N trajectories. The loss is computed on these trajectories, and the parameters are updated using gradient descent. W^{ε_i} is generated as an output from the neural network the optimal control can then be estimated as $u^{\varepsilon_i} = \frac{\sigma\sqrt{\varepsilon_i}\partial_x W^{\varepsilon_i}}{2}$. After the final level, we obtain the estimate of optimal control u^* corresponding to the original problem. We generate N trajectories under the final ε, and $u = u^*$, and compute the importance sampling estimator for the expectation as defined in Equation 3.

This approach benefits from the reduced number of samples we need to generate at each level, as each smaller embedded problem is no longer a rare event. The steps are summarised in Algorithm (3.1).

Algorithm 3.1 Neural Network Approximation to W

Algorithm parameters: Number of levels (k), Number of trajectories (N)
Initialize Neural Network: L layers
    Input: $(x,t) \in R^{d\times 1}$ ▷ $(d\times 1)$ dimensional input of particle state x at time t
    Layer 1: Input $\rightarrow H_1$
    Layer i: $H_{i-1} \rightarrow H_i$
    ...
    Layer L: $H_{L-1} \rightarrow$ Output
    Output: $W^{\varepsilon}(t,x) \in R^d$ ▷ W is the output from the NN
Initialize parameters $\theta_0 = 0$, $\varepsilon_0 = 10, u_0 = 0$
NN training parameters: Number of epochs (E), Batch size (B)
while $i \leq k$ **do**
    Generate N trajectories for $u = u_{i-1}, \varepsilon = \varepsilon_i$

$$dX_t^u = (b(X_t^u,t) + \sigma(X_t^u,t)u(X_t^u,t)\sqrt{\varepsilon})dt + \sigma(X_t^u,t)\sqrt{\varepsilon}dB_t^u$$

    while $j \leq E$ **do**
        for m in N/B **do** ▷ Loss calculation and weight update using a batch B
            Obtain $W_\theta^b(t,x_b)$ as the network output for each sample x_b in the batch B

$$\hat{L}(\theta) = -k_1 W_\theta^b(0,x_0) + \frac{k_2}{B}\sum_{b=1}^{B}(\mathscr{L}^{\varepsilon_i}W_\theta^b(t,x_b)^2)dt + \frac{k_3}{B}(\sum_{b=1}^{B}\left[(W_\theta^b(T,x_b) - 2f(x_b))^2\right])$$

            $\theta_{m+1} = \theta_m - \eta_m \nabla_\theta \hat{L}(\theta_m)$ ▷ Gradient Descent Update.
            Consider another randomly selected batch B among N trajectories
        end for
        Epoch number $e \rightarrow e+1$ for another pass over the N trajectories
    end while
    $u^i = \frac{\sigma\sqrt{\varepsilon_i}\partial_x W^{\varepsilon_i}}{2}$ ▷ Control derived from NN output
    Compute $p(X_T > a) = \sum_{i=1}^{N} I(X_T^{u^i} > a)\frac{d\mathbb{P}}{d\mathbb{Q}^{u^i}}$, ▷ Exceedance probability for $\varepsilon = \varepsilon_i$
end while

4 NUMERICAL EXPERIMENTS

We conduct experiments on the following diffusion processes: Brownian motion, Ornstein-Uhlenbeck, Cox-Ingersoll-Ross, and Langevin double well diffusion process. The process dynamics for each of these under an external control u is specified below:

1. Brownian motion: $dX_t^\varepsilon = \sqrt{\varepsilon}(udt + dB_t^u)$
2. Ornstein Uhlenbeck: $dX_t^\varepsilon = -\gamma X_t^\varepsilon dt + \sqrt{\varepsilon}udt + \sigma\sqrt{\varepsilon}dB_t$
3. Cox-Ingersoll-Ross process: $dX_t^\varepsilon = \alpha(\beta - X_t^\varepsilon)dt + \sqrt{\varepsilon}udt + \sigma\sqrt{\varepsilon X_t^\varepsilon}dB_t$
4. Double well potential: $dX_t^\varepsilon = -\kappa\nabla_x((X_t^\varepsilon)^2 - 1)^2 dt + \sqrt{\varepsilon}udt + \sqrt{2\varepsilon}dB_t$

Specifically, for OU process we set $\gamma = 0.01, \sigma = 0.1$. For CIR, we set $\alpha = 0.1, \beta = 0.1, \sigma = 0.15$. These parameters were chosen arbitrarily to get probabilities of the desired order.

4.1 Data Generation

To discretize the SDE over a time interval T, we employ the Euler-Maruyama scheme (Milstein 2013), which is expressed as:

$$X_{t+1} = X_t + (b + \sigma u\sqrt{\varepsilon})\Delta t + \zeta\sigma\sqrt{\varepsilon\Delta t},$$

where ζ follows a normal distribution with mean 0 and standard deviation 1. We initialize $X_0 = 0$ for all processes except the Langevin diffusion, where we initialize $X_0 = -1$. Additionally, we discretize the equation in steps of $\Delta t = 0.01$, upto $T = 10$.

4.2 Neural Network Training

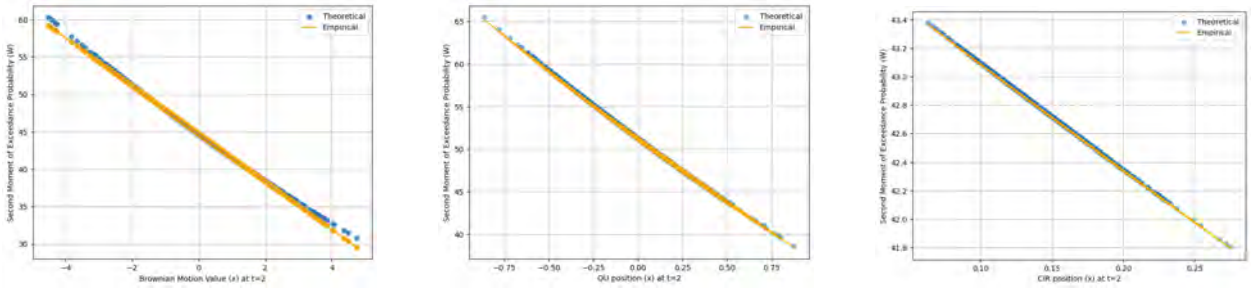
We implemented neural networks with 2-hidden layers, each having a dimension of $n = 5$ and employing a tanh activation function. These networks are capable of processing pairwise input representing the spatial position of a particle, denoted as $x \in \mathbb{R}^d$, at time $t \in [0, T]$. The output of the neural network is denoted as $W(t,x)$, as defined in Equation (5). To train these networks effectively, we utilize a loss function that combines three objectives, as specified in Equation (8). In our experiments, we generate $N = 10^4$ trajectories for each value of ε, and we operate with a batch size of 64 for stochastic gradient descent. We conduct 100 epochs with an early stopping parameter set to 20, and a learning rate of $\eta = 0.01$. We opt for the Adam optimizer for gradient descent. Following each epoch, we calculate the terms involving the derivative of the output with respect to the input in the loss function using the automatic differentiation functionality provided by PyTorch. We follow a linear schedule for ε and vary it uniformly from $\varepsilon = 10$ to 1 in steps decided by number of levels (k).

5 RESULTS

5.1 Verification of Estimates of Probability and Value Function

We compare the true probabilities and estimates obtained through our approach for the four processes across various levels of rarity. With the exception of Langevin double well diffusion, we determine the true probability analytically by examining the density function at time $t \leq T$. For Langevin double well diffusion, we estimate the true probability within a 95% confidence interval using Monte Carlo simulations executed on a High-Performance Computing machine. These comparative results are presented in Table (1). The estimates in the table represent the average of 20 independent runs, along with the 95% confidence interval assuming normal distribution of probabilities. Additionally, we report the empirical variance estimated from the neural network as $V^\varepsilon = \exp\{-\frac{W^\varepsilon}{\varepsilon}\}$. Across all processes except Langevin double well diffusion, we observe that the true probability lies within the 95% confidence interval of the estimated probabilities. In Figure 1, we illustrate the values of W obtained empirically as output from the neural network, alongside the theoretical value whenever feasible. The latter is computed as the second moment of the exceedance probability. Notably, we observe a close match between the neural network approximation to W^ε and the analytically computed value.

Comparison with (Nüsken and Richter 2021): Our approach achieves smaller relative errors in single-dimensional processes and shows an average reduction of approximately 34% in running time compared to



(a) Brownian (b) Ornstein-Uhlenbeck (c) CIR process

Figure 1: Comparison of numerically computed and neural network approximation to $W(t,x)$ at $t = 2$. Numerically, W is computed as the second moment of the exceedance probability.

the findings in (Nüsken and Richter 2021) for the same process parameters. Importantly, our method allows us to estimate extremely small probabilities, down to the order of 10^{-8} and below, which their methodology struggles to handle. The sequential escalation of rarity levels in our algorithm proves effective, enabling us to observe successes at each level and update parameters accordingly, even when dealing with problems characterized by high thresholds.

Process	**True Probability** (p)	**Estimated Probability** ($\hat{p}$)	$\sqrt{V^{\varepsilon}(0,x_0)}$
Brownian (a=1)	0.784×10^{-3}	$(0.784 \pm 0.003) \times 10^{-3}$	0.710×10^{-3}
Brownian (a=1.5)	1.038×10^{-6}	$(1.038 \pm 0.003) \times 10^{-6}$	1.025×10^{-6}
Brownian (a=2)	1.244×10^{-10}	$(1.241 \pm 0.011) \times 10^{-10}$	1.301×10^{-10}
OU ($a = 1.5$)	3.229×10^{-6}	$(3.258 \pm 0.026) \times 10^{-6}$	3.029×10^{-6}
OU ($a = 2$)	9.027×10^{-10}	$(8.982 \pm 0.126) \times 10^{-10}$	8.662×10^{-10}
CIR ($a = 1$)	1.796×10^{-5}	$(1.802 \pm 0.037) \times 10^{-5}$	1.649×10^{-5}
CIR ($a = 1.5$)	4.556×10^{-8}	$(4.793 \pm 0.163) \times 10^{-8}$	4.223×10^{-8}
CIR ($a = 2$)	1.003×10^{-10}	$(1.376 \pm 0.225) \times 10^{-10}$	1.611×10^{-10}
Double well ($\kappa = 5, d = 1$)	$(1.868 \pm 0.268) \times 10^{-4}$	$(3.534 \pm 0.118) \times 10^{-4}$	3.829×10^{-4}
Double well ($\kappa = 8, d = 1$)	$(3.552 \pm 0.368) \times 10^{-8}$	$(9.682 \pm 3.738) \times 10^{-8}$	7.137×10^{-8}
Double well ($\kappa = 5, d = 10$)	$(4.735 \pm 0.134) \times 10^{-3}$	$(3.118 \pm 0.732) \times 10^{-3}$	4.053×10^{-3}

Table 1: Comparison of true and estimated probabilities for different processes. With the exception of Langevin double well diffusion, true probabilities for all other processes are calculated numerically. For Langevin diffusion, the true probability is estimated using naive Monte Carlo samples, within the 95% confidence interval. Empirical values of $\sqrt{V^{\varepsilon}(0,x_0)}$ are also reported as obtained from the output of the neural network. V^{ε} is the second moment of probability and therefore a good approximation for the square of probability.

5.2 Grid Search for Optimal Weights in the Loss Function

The weights assigned to the three different objectives in the loss function, see (8), are likely to be critical to solution's accuracy. To illustrate this in a simple setting, we conduct an empirical grid search to find close to optimal distribution of weights assigned to these terms in case of Brownian motion. Specifically, we explore the parameter space where $k_3 = 1 - k_2 - k_1$ and each k_i ranges from 0 to 1. We traverse this grid with increments of 0.05 and evaluate the relative errors for all such triplets k_1, k_2, k_3. In Figure 2, we plot the relative errors for different weight distributions, keeping k_1 fixed, and varying k_2 from 0 to 1 in increments of 0.05. We find that we achieve low relative errors when a non-zero weight is given to all the 3

terms in the loss function, and especially when the term involving HJB-PDE is given the maximum weight. In particular, lowest errors are achieved in case of Brownian motion for $k_1 = 0.35, k_2 = 0.45, k_3 = 0.2$. We also observe that the solution does not converge when $k_1 = 0$, and one of k_2 or k_3 is also 0.

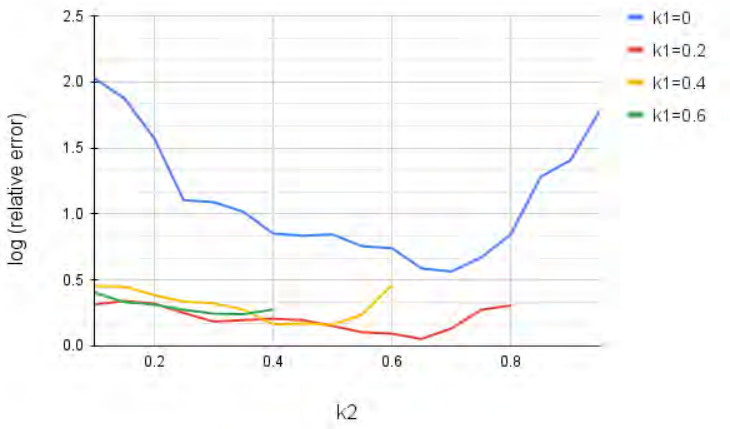


Figure 2: Relative errors for different weight distributions in the loss in (8) for Brownian motion ($a = 2$). We achieve low relative errors when a non-zero weight is given to all the three terms in the loss function, especially when k_2 is given the maximum value. In particular, the lowest errors are achieved in the case of Brownian motion for $k_1 = 0.35$, $k_2 = 0.45$, $k_3 = 0.2$. We also observe that the solution does not converge when $k_1 = 0$, and one of k_2 or k_3 is also 0.

5.3 Choosing the ε Increase Schedule to Control Rarity

We provide an illustrative heuristic argument to justify a linearly increasing schedule for increasing rarity in the simple Brownian motion setting. Consider the problem of estimating $P(B_T \geq Ta)$ where B_T denotes the standard Brownian motion observed at time T and a is a positive value. The above probability is identical to $P(N(0,1) \geq a\sqrt{T})$. We consider a sequence of problems $P(N(0,1) \geq \beta_m a\sqrt{T})$ for $m = 1,\ldots,n$ and $0 < \beta_1 < \beta_2 < \ldots \beta_n = 1$ for estimating this probability, where $\beta = \frac{1}{\varepsilon}$. We consider the semi-ideal setting where at each stage, the algorithm learns the correct exponential twist accurately. Thus after stage m, it learns that the optimal change of measure corresponds to generating a sample of $N(0,1)$ using $N(\beta_m, 1)$ (see (Asmussen and Glynn 2007) for discussion on exponentially twisting distributions, and their asymptotic optimality for random walks when the mean of the random walk is set to the exceedance probability threshold). Since $N(0,1)$ can be expressed as a sum of n independent $N(0, \frac{1}{n})$ random variables, the asymptotic optimality ideas remain relevant in our context. Now, further suppose that in the next stage, computational effort is proportional to the number of paths that cross the threshold $\beta_{m+1} a\sqrt{T}$. Each path roughly achieves this with probability $\frac{1}{2\pi(\beta_{m+1}-\beta_m)a\sqrt{T}} \exp(-\frac{1}{2}(\beta_{m+1}-\beta_m)^2 a^2 T)$. Further, assume that at each stage, a large and a fixed number of successful samples are needed to estimate the change of measure for the next stage. We see that the computational effort needed at each stage is roughly proportional to $\exp(\frac{1}{2}(\beta_{m+1}-\beta_m)^2 a^2 T)$. Thus, in this toy setting, finding the optimal escalation schedule roughly boils down to minimising (set $\varepsilon_0 = 0$),

$$\sum_{m=0}^{n-1} \exp\left(\frac{1}{2}(\beta_{m+1}-\beta_m)^2 a^2 T\right)$$

Observe that $\exp(cx^2)$ is a convex function of x for $c > 0$. Thus, by Jensen's inequality (for a random variable that takes each value $\beta_{m+1} - \beta_m$ with probability $1/n$)

$$\sum_{m=0}^{n-1} \exp\left(\frac{1}{2}(\beta_{m+1} - \beta_m)^2 a^2 T\right) \geq n \exp\left(\frac{1}{2}\left(\frac{\beta_n - \beta_0}{n} a\right)^2 T\right).$$

The RHS is achievable by a linear schedule achieved by setting each $\beta_m = \frac{m}{n}$. Thus the minimum value for a given n equals

$$n \exp\left(\frac{1}{2}\left(\frac{a}{n}\right)^2 T\right).$$

Using calculus, the optimal n can be seen to equal $a\sqrt{T}$. In practice the correct change of measure is only approximately learnt. To control the fluctuation in it, one may need to have a finer grid corresponding to a larger n. Our experiments on Brownian motion for $a = 2, T = 10$ support the above arguments. In Figure 3, we show the comparison of relative errors for schedules that have ε decreasing linearly vis-à-vis an exponentially decreasing schedule. We observe that while the errors are always ≥ 1, the linear schedule always results in lower relative errors for all values of the number of levels. Empirically, the optimal number of levels seems to be larger than the theoretical optimum, i.e. $a\sqrt{T}$. The latter however has been calculated under the assumption that we estimate the optimal measure at each level. We find that with better estimates of the measure in each level, the number of optimal levels start decreasing toward the theoretical optimum. This is shown in Figure 4, where it is indicated that more epochs result in better estimates and lower number of optimal levels.

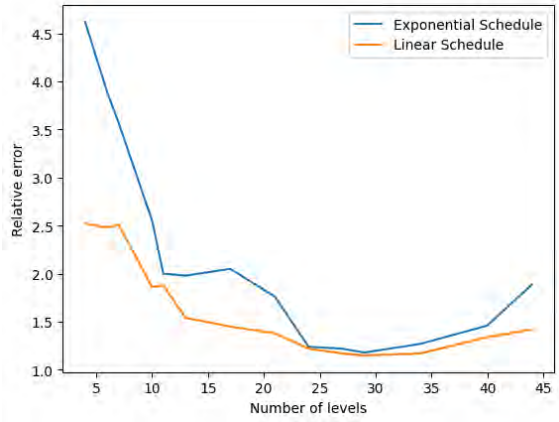


Figure 3: Comparison of relative errors for a linearly decreasing and exponentially decreasing schedule of ε for different number of levels. The linear schedule always results in lower relative errors.

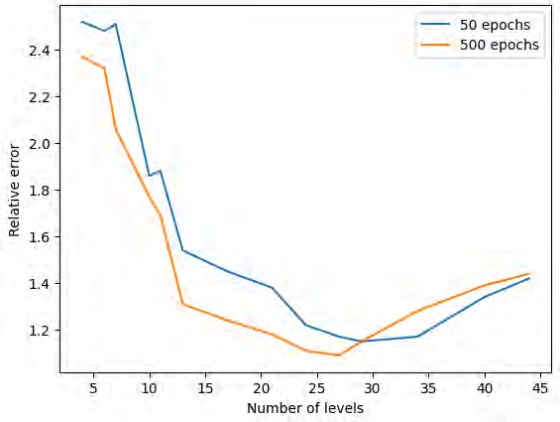


Figure 4: Comparison of relative errors under the linear schedule for different number of epochs across varying levels. More training results in lower relative errors upto a certain number of levels.

6 CONCLUSION

We considered the problem of devising an importance sampling methodology for a rare event problem associated with a diffusion process taking a rare excursion. We embedded the problem in a sequence of problems with increasing rarity. Each problem has an ideal zero variance solution that corresponds to solving an HJB-PDE. We approximately solve this PDE using deep learning methods with a mixture of carefully selected loss functions. Solution from each less rare problem is used to help solve the more rare problem at the next stage. We implemented the proposed methods on rare events associated with the

Brownian motion, O-U process, CIR process as well as for double well problem with a Langevin diffusion. We find that our results are quite efficient and accurate for Brownian motion as well as the O-U and the CIR process. For the double-well problem our results begin to lose accuracy although they still perform better than other proposed methods in the literature. This suggests that while the proposed approach is promising, much further research is needed both conceptually and empirically, to be able to solve rare event problems associated with diffusion processes in generality.

7 ACKNOWLEDGEMENTS

We thank the Indo-Sweden joint project entitled 'Large Deviations, Rare-Event Simulation and Machine Learning: Importance Sampling using Neural Networks' for supporting this research.

REFERENCES

Asmussen, S. and P. W. Glynn. 2007. *Stochastic Simulation: Algorithms and Analysis*, Volume 57. Springer.

Boué, M. and P. Dupuis. 1998. "A Variational Representation for Certain Functionals of Brownian Motion". *The Annals of Probability* 26(4):1641–1659.

Dupuis, P. and H. Wang. 2007. "Subsolutions of an Isaacs Equation and Efficient Schemes for Importance Sampling". *Mathematics of Operations Research* 32(3):723–757.

Han, J., A. Jentzen, *et al*. 2017. "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations". *Communications in mathematics and statistics* 5(4):349–380.

Juneja, S. and P. Shahabuddin. 2006. "Rare-event Simulation Techniques : An Introduction and Recent Advances". *Handbooks in operations research and management science* 13:291–350.

Milstein, G. N. 2013. *Numerical Integration of Stochastic Differential Equations*, Volume 313. Springer Science & Business Media.

Nakamura-Zimmerer, T., Q. Gong, and W. Kang. 2021. "Adaptive Deep Learning for High-Dimensional Hamilton-Jacobi-Bellman Equations". *SIAM Journal on Scientific Computing* 43(2):A1221–A1247.

Nüsken, N. and L. Richter. 2021. "Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space". *Partial differential equations and applications* 2(4):48.

Oksendal, B. 2013. *Stochastic Differential Equations: an Introduction with Applications*. Springer Science & Business Media.

Pavliotis, G. A. 2014. "Stochastic Processes and Applications". *Texts in Applied Mathematics* 60.

Üstünel, A. S. and M. Zakai. 2013. *Transformation of Measure on Wiener Space*. Springer Science & Business Media.

Vanden-Eijnden, E. and J. Weare. 2012. "Rare Event Simulation of Small Noise Diffusions". *Communications on Pure and Applied Mathematics* 65(12):1770–1803.

AUTHOR BIOGRAPHIES

HENRIK HULT is a professor of mathematical statistics and the head of division of Probability, Mathematical Physics and Statistics at KTH Royal Institute of Technology. His research interests lie in applied probability and statistics including in machine learning, Monte Carlo methods, mathematical finance and life sciences. His email address is hult@kth.se and his website is https://people.kth.se/~hult/.

AASTHA JAIN is a pre-doctoral fellow at CDLDS at Ashoka University. Her research interests are in applied probability and machine learning. Her email address is aastha.jain@ashoka.edu.in.

SANDEEP JUNEJA is a professor of computer science and the director for Centre for Data, Learning and Decision Sciences at Ashoka University. His research interests lie in applied probability including in sequential learning, mathematical finance, Monte Carlo methods, game theoretic analysis of queues and epidemiological modelling. He is currently the area editor for Operations Research in simulation. Earlier he has been on editorial boards of Stochastic Systems, Mathematics of Operations Research, Management Science and ACM TOMACS. His email address is sandeep.juneja@ashoka.edu.in and his website is https://www.tcs.tifr.res.in/~sandeepj/.

PIERRE NYQUIST is an Associate Professor in the Department of Mathematical Sciences at Chalmers University of Technology and the University of Gothenburg. His research interests lie in applied probability, including large deviations, stochastic control, Monte Carlo methods, and machine learning. His e-mail address is pnyquist@chalmers.se and his website

is https://pierrenyq.github.io.

SUSHANT VIJAYAN is a fifth year PhD student at the School of Technology and Computer Science (STCS) at TIFR, Mumbai. His research interests include sequential decision problems, applied probability and stochastic processes. His email address is sushant.vijayanq@tifr.res.in.