

Reliability Analysis of Complex Systems using Subset Simulations with Hamiltonian Neural Networks

April 2024

hanging the World's Energy Future Idaho National

Denny Thaler, Franz Bamer, Bernd Markert, Michael Shields, Som LakshmiNarasimha Dhulipala

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/JOU-23-72489-Revision-0

Reliability Analysis of Complex Systems using Subset Simulations with Hamiltonian Neural Networks

Denny Thaler, Franz Bamer, Bernd Markert, Michael Shields, Som LakshmiNarasimha Dhulipala

April 2024

Idaho National Laboratory Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the U.S. Department of Energy Under DOE Idaho Operations Office Contract DE-AC07-05ID14517

Reliability Analysis of Complex Systems using Subset Simulations with Hamiltonian Neural Networks

³ Denny Thaler^{1,2*}, Somayajulu LN Dhulipala³, Franz Bamer¹, Bernd Markert¹, Michael D. Shields²

Institute of General Mechanics, RWTH Aachen University, 52062 Aachen, Germany

5 Department of Civil and Systems Engineering, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA

Computational Mechanics and Materials, Idaho National Laboratory, Idaho Falls, ID 83402, USA

7 Abstract

4

6

We present a new Subset Simulation approach using Hamiltonian neural network-based Monte Carlo 8 sampling for reliability analysis. The proposed strategy combines the superior sampling of the Hamiltonian 9 Monte Carlo method with computationally efficient gradient evaluations using Hamiltonian neural networks. 10 This combination is especially advantageous because the neural network architecture conserves the Hamil-11 tonian, which defines the acceptance criteria of the Hamiltonian Monte Carlo sampler. Hence, this strategy 12 achieves high acceptance rates at low computational cost. Our approach estimates small failure probabilities 13 using Subset Simulations. However, in low-probability sample regions, the gradient evaluation is particu-14 larly challenging. The remarkable accuracy of the proposed strategy is demonstrated on different reliability 15 problems, and its efficiency is compared to the traditional Hamiltonian Monte Carlo method. We note 16 that this approach can reach its limitations for gradient estimations in low-probability regions of complex 17 and high-dimensional distributions. Thus, we propose techniques to improve gradient prediction in these 18 particular situations and enable accurate estimations of the probability of failure. The highlight of this 19 study is the reliability analysis of a system whose parameter distributions must be inferred with Bayesian 20 inference problems. In such a case, the Hamiltonian Monte Carlo method requires a full model evaluation 21 for each gradient evaluation and, therefore, comes at a very high cost. However, using Hamiltonian neural 22 networks in this framework replaces the expensive model evaluation, resulting in tremendous improvements 23 in computational efficiency. 24

 ${}_{25} \quad Keywords: \ \ Subset \ Simulation; \ Hamiltonian \ Neural \ Networks; \ Hamiltonian \ Monte \ Carlo; \ Rare \ event$

²⁶ simulation; Bayesian inference

27 **1. Introduction**

Engineers are responsible for designing reliable structures, considering the cost and consequences of 28 structural damage/failure. To this end, methods such as the first-order reliability method (FORM) [1] and 29 second-order reliability method (SORM) [2] have become standard procedures to estimate the probability of 30 failure. However, structures require different thresholds for the probability of failure. Irreplaceable parts and 31 those whose failure will have dire consequences, i.e., components in space structures or nuclear power plants, 32 need to be constructed such that failure is extremely unlikely. Since structures generally behave nonlinearly 33 before they fail and distributions for uncertain variables may deviate significantly from Gaussian, evaluating 34 the failure probability is usually not straightforward. That is, the failure probability cannot be solved 35 analytically such that numerical and/or statistical reliability methods are required [3]. 36

The Monte Carlo method is the benchmark statistical method since it provides an unbiased estimate of the failure probability. However, it requires a huge number of model evaluations to provide a reasonable estimate, especially if the probability of failure is very low. Therefore, it is necessary to significantly speed up this method. Methods aimed at significantly decreasing computational cost generally fall into three

*Corresponding author. Prephrmeilauthttesst that sets in the constant of the transformed that the the constant of the cons

major categories: (i) methods that speed up each sample evaluation, (ii) methods that improve sampling 41 efficiency and reduce the number of necessary model evaluations, and (iii) methods that combine categories 42 (i) and (ii). To speed up model evaluations, it is common to employ model order reduction strategies. 43 e.g., [4–10] or construct a surrogate model that approximates the model response, e.g., [11–18]. Other 44 approaches combine these strategies in a multi-fidelity framework wherein data are fused from reduced order 45 models and high-fidelity simulations [19–22]. To improve sampling efficiency, numerous strategies have been 46 developed. Established methods include importance sampling [23], Latin Hypercube sampling [24], and 47 Subset Simulation [25], among others. The highest computational benefits can be achieved by combining 48 the speed up of sample evaluation and reducing model evaluations. For example, model order reduction has 49 been combined with importance sampling and subset simulations [26, 27] and Kriging and neural network 50 surrogate models have been integrated with importance sampling [28, 29]. 51 In this paper, we will focus on a specific aspect of the second strategy, namely efficient sampling. Our 52 approach is rooted in the widely-used Subset Simulation method, which has become a standard approach for 53

Monte Carlo-based reliability analysis [25, 30–32]. Subset Simulation requires Markov Chain Monte Carlo 54 (MCMC) methods to sample from the conditional distributions at each level of the method. Initially, this 55 was done with a modification to the random walk-based Metropolis-Hastings sampler [33, 34] to perform 56 component-wise steps [25]. Since that time, numerous schemes have been proposed that use enhanced 57 MCMC methods to improve efficiency, including repeated generation of pre-candidate states [35], delayed 58 rejection [36], and conditional sampling [37]. Zuev et al. also proposed a Bayesian postprocessor for Subset 59 Simulation to further estimate the probability density function to quantify the uncertainty of the failure 60 probability [38]. 61

The aforementioned methods generally consider Subset Simulation formulated in the standard Gaussian 62 space. However, performing Subset Simulation in the original physical space is often desirable because an 63 isoprobabilistic transformation to standard Gaussian is not easily obtained. In the original space, however, 64 65 Subset Simulation can be complicated by strong nonlinear dependence among variables coupled with strong non-Gaussianity and degeneracy of high-dimensional distributions. Two recent approaches have specifically 66 aimed to address this issue. Shields et al. [39] applied the random walk-based affine invariant ensemble 67 "Stretch" sampler in Subset Simulation for problems where sampling the conditional distributions may be 68 difficult. Meanwhile, Wang et al. [40] proposed algorithms that leverage Hamiltonian Monte Carlo (HMC) 69 to generate samples by simulating the evolution of time-reversible Hamiltonian dynamics using a symplectic 70 numerical integrator. This work was further extended to use Riemannian manifold HMC in [41]. 71

The major disadvantage of HMC compared to Metropolis-Hastings and other random walk methods is 72 the computational cost. The proposal of a new state in HMC requires simulating the evolution of time-73 reversible Hamiltonian dynamics numerically using symplectic integration. Therein, evaluating gradients of 74 the Hamiltonian causes the greatest computational effort. To improve the efficiency, Strathmann et al. [42] 75 proposed a gradient-free approach based on exponential kernels. Furthermore, Broccardo et al. combined the 76 HMC with Gaussian process modeling [43]. Li et al. [44] proposed to use neural networks to approximate the 77 numerical gradient in HMC, while Levy et al. [45] proposed to learn a neural network operator that serves 78 as an efficient kernel for the HMC method. These neural network methods, however, were not constrained 79 by the Hamiltonian dynamics. Greydanus et al. [46] proposed the Hamiltonian neural network (HNN), a 80 physics-informed neural network that conserves energy over long trajectories. Recently, Dhulipala et al. [47] 81 proposed to use HNNs for efficient HMC sampling for Bayesian inference and its integration with an advanced 82 version of HMC called the No-U-Turn Sampler (NUTS). Dhulipala et al. [47] also proposed a modified version 83 of HNNs called latent HNNs. This acceleration of HMC with HNNs has been further verified in [48]. 84 In this paper, we propose to integrate pre-trained HNNs with HMC, termed Hamiltonian Neural Network 85

Monte Carlo (H_{NN}MC), to accelerate Subset Simulations. The proposed method provides the same robust probability of failure estimates using the standard HMC methods, earlier proposed by Wang et al. [40], while reducing the cost of the Hamiltonian integration considerably. We achieve greater than 20 times speedup in the propagation of the Hamiltonian trajectories, which makes conditional sampling in Subset Simulation far more efficient than the standard HMC. Furthermore, we provide strategies to overcome the limitations

 $_{91}$ of pretraining the HNNs that make extensions of the proposed $H_{NN}MC$ algorithm possible for even more

⁹² challenging problems such as those where Bayesian inference and reliability analysis are conducted together.

⁹³ The final example of this paper explores this special case in which model evaluations are required at every

 $_{94}$ HMC step. However, the proposed $H_{NN}MC$ predicts the gradient directly, such that model evaluations are

not needed for Hamiltonian integration. In fact, model evaluation is only required once to accept or reject
 the conditional state, as in the conventional standard Subset Simulation, resulting in a significant speed-up.

the conditional state, as in the conventional standard Subset Simulation, resulting in a significant speed-up. Section 2 briefly introduces Subset Simulations before the $H_{NN}MC$ method is presented in Section 3.

⁹⁸ Using H_{NN}MC within Subset Simulations is proposed in Section 4 followed by a proof of concept in Section 5.

⁹⁹ Section 6 demonstrates the strategy on systems with uncertain parameters. Finally, Section 7 concludes the

¹⁰⁰ results of the proposed strategy.

¹⁰¹ 2. Subset Simulation for Reliability Analysis

To determine whether structural failure occurs, one must define a limit state based on engineering decisions. The limit state function $g(\mathbf{x})$ is defined as a function of the random vector \mathbf{x} describing uncertainties in the system and its inputs such that $g(\mathbf{x}) \leq 0$ corresponds to failure of the system. The probability of failure is then determined by the multi-dimensional integral of the probability density function $f_X(\mathbf{x})$ over the failure domain [49]:

$$P_F = P(F) = \int_{\substack{g(\mathbf{x}) \le 0}} \cdots \int_{g(\mathbf{x}) \le 0} f_X(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,, \tag{1}$$

where F is the failure region corresponding to the event $g(\mathbf{x}) \leq 0$. The equation can be reformulated using an indicator function $\mathbb{I}_F(\mathbf{x})$, for which $\mathbb{I}_F(\mathbf{x}) = 1$ if $\mathbf{x} \in F$ and $\mathbb{I}_F(\mathbf{x}) = 0$ otherwise, as follows:

$$P_F = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathbb{I}_F(\mathbf{x}) f_X(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,. \tag{2}$$

¹⁰⁹ Using standard Monte Carlo methods, the probability of failure can be estimated by the expectation of ¹¹⁰ the indicator function, written as:

$$P_F = E[\mathbb{I}_F(\mathbf{x})] \approx \frac{1}{m} \sum_{k=1}^m \mathbb{I}_F(\mathbf{x}^{(k)}) .$$
(3)

where $\mathbf{x}^{(k)}$ are *m* independent samples drawn from $f_X(\mathbf{x})$. In general, this method gives accurate and robust probability of failure estimates, but it converges slowly – requiring large numbers of samples in which

 $m \propto \frac{1}{P_F}$. For small P_F , this is clearly problematic.

Subset Simulation, proposed by Au and Beck [25], aims to reduce the number of samples by defining P_F through a nested sequence of failure regions $F_1 \supset F_2 \supset \cdots \supset F_n = F$ so that $F = \bigcap_{i=1}^k F_i$, $k = 1, \ldots, n$. The probability of failure is then evaluated using conditional probabilities as follows:

$$P(F) = P(F_1) \prod_{i=1}^{n-1} P(F_{i+1} \mid F_i) .$$
(4)

The conditional probabilities are set to be larger values, typically around 0.1, and are estimated using Monte Carlo simulation by drawing samples from the conditional distributions at each level. These samples are drawn using various MCMC algorithms to be constrained within the conditional subsets. As previously mentioned, a great deal of research has focused on using different MCMC algorithms for this task. Of particular interest in this work is the use of HMC, which we discuss next.

¹¹⁹ 3. Hamiltonian Neural Networks in Hamiltonian Markov Chain

In this section, we first provide a brief overview of the Hamiltonian Monte Carlo method. Next, we introduce Hamiltonian Neural Networks (HNNs), a type of physics-informed neural network specifically designed to learn Hamiltonian dynamics. We then show how HNNs can be integrated into HMC to accelerate the sampling process.

124 3.1. Hamiltonian Monte Carlo

Hamiltonian Monte Carlo makes the analogy between sampling and a set of particles moving through a probability distribution, $\pi(\mathbf{x})$, with no external forces (i.e., only conservative forces). Under these conditions, the moving particles are governed by the following principles of Hamiltonian mechanics, wherein the state of the particles can be fully described by the position vector \mathbf{q} and the momentum vector \mathbf{p} . Since the system is conservative, its energy (Hamiltonian) is constant and composed of the kinetic and potential energy, Kand U, which depend only on the position and momentum. The Hamiltonian is written as follows:

$$H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + K(\mathbf{p}) .$$
(5)

The state of the system evolves with time using Hamilton's equations relating the position \mathbf{q} and the momentum \mathbf{p} as:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}} ,$$

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}} .$$
(6)

 $_{131}$ Solving this system of equations in time results in a constant energy (Hamiltonian) trajectory in the (\mathbf{p}, \mathbf{q})

phase space. Importantly for HMC, these equations describe a system that is reversible and symplectic (i.e.,
conserves energy and preserves volume). The importance of these properties for HMC has been detailed in
other works [40, 50] and will not be discussed further here.

HMC leverages Hamiltonian dynamics by treating the sample **x** drawn from $\pi(\mathbf{x})$ as a set of particles having positions **q** in the Hamiltonian trajectory (i.e., $\mathbf{x} \equiv \mathbf{q}$). We then define the joint probability density between the position **q** and momentum **p** to take the following form:

$$\pi(\mathbf{p}, \mathbf{q}) \propto e^{-H(\mathbf{p}, \mathbf{q})} = e^{-U(\mathbf{q})} e^{-K(\mathbf{p})} .$$
(7)

¹³⁸ We then define the potential energy by:

$$U(\mathbf{q}) = -\log \pi(\mathbf{q}) , \qquad (8)$$

¹³⁹ and the kinetic energy by:

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}\mathbf{M}^{-1}\mathbf{p} , \qquad (9)$$

which results in the position following the target distribution $\pi(\mathbf{q})$ and the momentum independently following a Gaussian distribution with covariance \mathbf{M} . Selecting \mathbf{M} to be a non-diagonal, positive semi-definite covariance matrix results in the Riemannian Manifold HMC, which has been demonstrated to improve the acceptance rate for non-Gaussian distributions but requires determining the best \mathbf{M} matrix for a given problem [41, 51]. However, in this work, as is often done, we select $\mathbf{M} = \alpha \mathbf{I}$ where \mathbf{I} is the identity matrix, and α is a scalar value, which corresponds to the momenta being independent Gaussian random variables.

To perform HMC, we then consider the current state of the Markov chain as the position **q** and assign a Gaussian random momentum **p**, which completely defines a state in phase space such that Eq. (6) defines a reversible trajectory of constant energy (Hamiltonian). Hamilton's equations (Eq. (6)) are then solved numerically using a symplectic integrator. Here, we use the synchronized leapfrog scheme, which first ¹⁵⁰ updates the positions written as:

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \frac{\Delta t}{m_i} \mathbf{p}(t) + \frac{\Delta t^2}{2m_i} \dot{\mathbf{p}}(t) = \mathbf{q}(t) + \frac{\Delta t}{m_i} \mathbf{p}(t) - \frac{\Delta t^2}{2m_i} \frac{\partial H}{\partial \mathbf{q}(t)} , \qquad (10)$$

where Δt is the integration time step and $\dot{\mathbf{p}}(t) = -\frac{\partial H}{\partial \mathbf{q}(t)}$. Then, the momenta are updated by:

$$\mathbf{p}(t+\Delta t) = \mathbf{p}(t) + \frac{\Delta t}{2} \left(\dot{\mathbf{p}}(t) + \dot{\mathbf{p}}(t+\Delta t) \right) = \mathbf{p}(t) - \frac{\Delta t}{2} \left(\frac{\partial H}{\partial \mathbf{q}(t)} + \frac{\partial H}{\partial \mathbf{q}(t+\Delta t)} \right), \tag{11}$$

¹⁵² where again $\dot{\mathbf{p}}(t) = -\frac{\partial H}{\partial \mathbf{q}(t)}$.

A new state $(\mathbf{q}^*, \mathbf{p}^*)$ is determined by integrating forward L time steps for a total time of $t_f = L\Delta t$. Because the integration is performed numerically, there is some error that causes a deviation in the Hamiltonian (i.e., H is not strictly conserved). Therefore, a Metropolis-Hastings type acceptance-rejection criterion is introduced that accepts the new state with probability

$$\alpha = \min\left[1, \exp(H(\mathbf{q}, \mathbf{p}) - H(\mathbf{q}^*, \mathbf{p}^*))\right] .$$
(12)

The HMC process is highly sensitive to both L and Δt such that large Δt can cause large integration errors and thus high rejection rates. Meanwhile, small L can result in undesirable random-walk-type behavior, while large L can result in unnecessarily high computation costs. To avoid the need to manually tune the parameters, one may apply methods such as the No-U-Turns Sampler (NUTS) [52]. However, in the context of Subset Simulation, Wang et al. [40] proposed methods for setting t_f , which will be discussed in Section 4. Here, we apply strategies similar to NUTS to generate the samples from the original distribution and rely on methods of Wang et al. for sampling from conditional distributions at subsequent levels.

¹⁶⁴ For completeness, the Hamiltonian Monte Carlo method is detailed in Algorithm 1.

 Algorithm 1 Hamiltonian Monte Carlo

 Initial state: q, Hamiltonian: H, Leapfrog steps: L, Step size: Δt
 $\mathbf{p}_0 \leftarrow \mathcal{N}(0, 1)$
 $\mathbf{q}_0 \leftarrow q$
 $\mathbf{p}_{\frac{1}{2}} \leftarrow \mathbf{p}_0 - \frac{\Delta t}{2} \frac{\partial H}{\partial \mathbf{q}}$

 for $1 \leq n < L$ do

 $\mathbf{q}_n \leftarrow \mathbf{q}_{n-1} + \Delta t p_n$
 $\mathbf{p}_{n+\frac{1}{2}} \leftarrow \mathbf{p}_{n-\frac{1}{2}} - \Delta t \frac{\partial H}{\partial \mathbf{q}}$

 end for

 $\mathbf{p}_L \leftarrow p_{L-1} - \frac{\Delta t}{2} \frac{\partial H}{\partial q}$
 $\mathbf{q}^* \leftarrow \mathbf{q}_L$
 $\mathbf{p}^* \leftarrow -\mathbf{p}_L$
 $\alpha = \min[1, \exp(-H(\mathbf{q}^*, \mathbf{p}^*) + H(\mathbf{q}_0, \mathbf{p}_0))]$

 if $\alpha \geq \mathcal{U}(0, 1)$ then

 return $(\mathbf{q}, \mathbf{p}^*)$

 else

 return $(\mathbf{q}_0, \mathbf{p}_0)$

 end if

165 3.2. Hamiltonian Neural Networks

¹⁶⁶ HMC requires the computation of the gradient of the system $\left(\frac{\partial H}{\partial \mathbf{q}}\right)$ at each time step, which can be ¹⁶⁷ computationally expensive, and each proposal of HMC requires many time steps. To reduce this cost, gradient ¹⁶⁸ evaluation can be accelerated using a surrogate model. Li et al. [44] showed that standard feedforward neural ¹⁶⁹ networks, such as that shown in Fig. 1(a), result in a high speed-up. However, recent enhancements in ¹⁷⁰ physics-informed neural networks that conserve energy promise high accuracy in long-run time integration ¹⁷¹ of Hamiltonian systems. One example of particular interest is the Hamiltonian Neural Network (HNN) [46]. ¹⁷² The idea of HNNs is to learn the Hamiltonian $H(\mathbf{q}, \mathbf{p})$ such that energy is conserved – that is, Hamilton's ¹⁷³ equations are satisfied. This is achieved by constructing a neural network that learns the parameters θ by ¹⁷⁴ solving

$$\operatorname{argmin}_{\theta} \left\| \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} - \frac{\partial H_{\theta}}{\partial \mathbf{p}} \right\|^{2} + \left\| \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} + \frac{\partial H_{\theta}}{\partial \mathbf{q}} \right\|^{2}, \tag{13}$$

where the gradients $\frac{\partial H_{\theta}}{\partial \mathbf{p}}$ and $\frac{\partial H_{\theta}}{\partial \mathbf{q}}$ are obtained directly from the neural network through automatic differentiation as illustrated in Fig. 1(b). Recently, a variation of HNNs, termed Latent HNNs (L-HNNs), was proposed in which the network output is a set of *d* latent variables $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots, \lambda_d\}$ where *d* is the number of random variables such that

$$H_{\theta} = \sum_{i=1}^{d} \lambda_i , \qquad (14)$$

which has been shown to improve the expressivity of the network [47]. The L-HNN structure is shown in Fig. 1(c). Next, we will briefly discuss the nuances of training these different neural networks and their implications for HMC.

The standard feedforward neural network approach takes as input the position and momentum (\mathbf{q}, \mathbf{p}) and 182 returns the gradients $\left(\frac{d\mathbf{q}}{dt}, \frac{d\mathbf{p}}{dt}\right)$ directly. Hence, training data consists of trajectories of Hamilton's equations, 183 where each time step provides a training point. In HNNs, the training data are the same, but the network 184 is constructed to learn a function H_{θ} intended to approximate the Hamiltonian. By establishing the loss 185 function in Eq. (13), the network learns H_{θ} such that energy is conserved by comparing the gradients of $\frac{\partial H_{\theta}}{\partial \mathbf{p}}$ and $\frac{\partial H_{\theta}}{\partial \mathbf{q}}$ with the gradients $(\frac{d\mathbf{q}}{dt}, \frac{d\mathbf{p}}{dt})$ in the training data. Finally, the L-HNN takes advantage of the fact that the gradient is a linear operation and instead predicts a set of d latent variables from which H_{θ} 186 187 188 can be computed by Eq. (14), thus improving expressivity, particularly for high-dimensional problems. We 189 therefore notice that, although the training data are identical, the HNNs and L-HNNs will have far superior 190 performance – especially for long trajectories – because they are designed to conserve energy. 191

¹⁹² 3.3. The $H_{NN}MC$ Algorithm

Integration of HNNs into HMC is straightforward but can have significant computational benefits. This method, which we refer to as $H_{NN}MC$ simply replaces the standard numerical gradient evaluation in the leapfrog integration (Eqs. (10) and (11)) with HNN approximated gradients $\frac{\partial H_{\theta}}{\partial \mathbf{p}}$ and $\frac{\partial H_{\theta}}{\partial \mathbf{q}}$. This way, the Hamiltonian trajectories necessary for generating the next state of the Markov chain can be computed without the expensive numerical gradients that would typically be required when performing the conventional approach. The resulting algorithm is provided in Algorithm 2.

Of course, the HNN requires sufficient training data, and in its application to HMC, this training data is generated *a priori*. Consequently, the cost of the $H_{NN}MC$ is incurred up-front, and the subsequent chain propagation comes at little cost. Generally, this up-front cost is far less than the cost of numerical integration and it becomes increasingly efficient as the number of time steps and the number of samples increases.

203 3.4. H_{NN}MC Illustration

In this section, we briefly demonstrate the use of HNNs in HMC. To achieve good results, it is crucial to accurately reproduce the Hamiltonian system. First, we show that the HNN accurately conserves the Hamiltonian for a 1D bimodal distribution. We then show that the HNN gradients produce accurate Hamiltonian trajectories for a 2D distribution with correlations.

For the first demonstrative example, we choose a bimodal Gaussian with $\mu = [0,3]$ and $\sigma = [1,1]$ as the target distribution, where the second peak has a higher weight, having a pdf given by:

$$p(q) = \frac{\sigma}{\sqrt{2\pi}} \left(\frac{1}{4} \exp\left(-0.5\left(\frac{q-\mu_1}{\sigma}\right)^2\right) + \frac{3}{4} \exp\left(-0.5\left(\frac{q-\mu_2}{\sigma}\right)^2\right) \right) . \tag{15}$$



Figure 1: Comparison of the neural network architectures: (a) Scheme of a standard feedforward neural network; (b) Hamiltonian Neural Network architecture with an in-graph gradient, cf. [46]; (c) Scheme of latent Hamiltonian Neural Networks, cf. [47].

The training data are generated by simulating relatively long training trajectories from a few initial 210 samples. For this distribution, 40 trajectories are simulated with a trajectory length of 20 and a step 211 size of 0.05. Hence, the training set includes 1.6×10^4 gradients. This modest amount of data leads to 212 a sufficient coverage of the sample domain taking into account that the distribution is one-dimensional 213 and has a relatively simple density function. The HNN consists of two hidden layers with ten neurons in 214 each layer. This relatively simple architecture can achieve high training and test accuracy. For this one-215 dimensional problem, the accurate prediction of the HNN can be seen in Figure 2(a) by comparing the 216 simulated trajectories with those using traditional gradient calculation in phase space. The Hamiltonian is 217 well preserved using the HNN, as the constant energy levels are simulated accurately. Figure 2(b) shows the 218 distribution estimated from $5\,000$ samples generated using the $H_{NN}MC$ matches the true probability density 219 with high accuracy. 220

For the second illustrative example, we apply $H_{NN}MC$ for a two-dimensional correlated Gaussian distribution that has the mean $\boldsymbol{\mu} = [1, -1]^T$ and the covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1. \end{bmatrix}$, with pdf given by:

$$p(\mathbf{q}) = \frac{\exp\left(-0.5\left(\mathbf{q}-\boldsymbol{\mu}\right)^{T}\boldsymbol{\Sigma}^{-1}\left(\mathbf{q}-\boldsymbol{\mu}\right)\right)}{\sqrt{(2\pi)^{2}\left\|\boldsymbol{\Sigma}\right\|}} .$$
 (16)

Algorithm 2 Hamiltonian Neural Network Monte Carlo

Initial state: \mathbf{q} , Hamiltonian: H, Leapfrog steps: L, Step size: Δt , Hamiltonian Neural Network prediction: HNN(), Neural network input: $\mathbf{x} = \mathbf{q} \oplus \mathbf{p}$ $\mathbf{p}_0 \leftarrow \mathcal{N}(0, 1)$ $\mathbf{q}_0 \leftarrow \mathbf{q}$ $\mathbf{p}_{\frac{1}{2}} \leftarrow \mathbf{p}_0 - \frac{\Delta t}{2} \text{HNN}(\mathbf{x})$ for $1 \le n < L$ do $\mathbf{q}_n \leftarrow \mathbf{q}_{n-1} + \Delta t \mathbf{p}_n$ $\mathbf{p}_{n+\frac{1}{2}} \leftarrow \mathbf{p}_{n-\frac{1}{2}} - \Delta t \text{HNN}(\mathbf{x})$ end for $\mathbf{p}_L \leftarrow \mathbf{p}_{L-1} - \frac{\Delta t}{2} \text{HNN}(\mathbf{x})$ $\mathbf{q}^* \leftarrow \mathbf{q}_L$ $\mathbf{p}^* \leftarrow -\mathbf{p}_L$ $\alpha = \min[1, \exp(-H(\mathbf{q}^*, \mathbf{p}^*) + H(\mathbf{q}_0, \mathbf{p}_0))]$ if $\alpha \ge \mathcal{U}(0, 1)$ then return $(\mathbf{q}^*, \mathbf{p}^*)$ else return $(\mathbf{q}_0, \mathbf{p}_0)$ end if



Figure 2: Application of $H_{NN}MC$ for a bimodal Gaussian mixture distribution. (a) Trajectories in phase space of constant Hamiltonian. The trajectories are simulated using the gradients of the Hamiltonian (HMC) or the predicted values (HNN); (b) The distribution of 5000 samples using the $H_{NN}MC$ compared with the true probability density function.

The generated samples of the correlated distribution using $H_{NN}MC$ are shown in Figure 3(a). To compare the gradient evaluations, one trajectory is simulated using the standard Hamiltonian gradients as well as the predicted gradients using HNNs. To show the accuracy of long-run trajectories, the length is chosen larger than the required trajectories during sampling. We observe that the HNN trajectory follows the original one and conclude that the gradient prediction for this trajectory is accurate. For both variables, q_1 and q_2 , the probability density functions are shown in Figure 3(b). Both match the true probability density function of the Gaussian shifted to the respective mean.



Figure 3: Two-dimensional correlated Gaussian distribution: (a) 5000 generated samples along with a long trajectory that moves through the distribution using traditional and HNN predicted gradients. (b) Probability density of the variables using 5000 samples compared to the true probability density function.

231 4. H_{NN}MC for Subset Simulation

In this section, we propose a methodology to integrate $H_{NN}MC$ into Subset Simulation. The framework closely follows the method proposed by Wang et al. [40] for the use of HMC in Subset Simulation, so we begin by discussing this approach.

235 4.1. HMC for Subset Simulation

Two primary considerations must be addressed when applying HMC for conditional sampling in Subset Simulation. The first is the fundamental question of sampling from the *conditional* distribution. To account for the truncation of the distribution that occurs when defining each conditional level, Wang et al. point out that integrating the conditional distribution into the Hamiltonian as follows:

$$H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + K(\mathbf{p}) = -\log(\pi(\mathbf{q}|F_j)) + \frac{1}{2}\mathbf{p}\mathbf{M}^{-1}\mathbf{p}$$

= $-\log(\pi(\mathbf{q})) + \frac{1}{2}\mathbf{p}\mathbf{M}^{-1}\mathbf{p} - \log(\mathbb{I}_F(\mathbf{q})) + \text{const.}$ (17)

240 creates a *potential barrier*. That is, when $\mathbb{I}_F(\mathbf{q}) = 0$, the Hamiltonian has infinite potential energy.

To account for this energy barrier, they proposed two algorithms. The first approach effectively ignores the barrier and applies a *post hoc* acceptence-rejection step. This scheme, referred to as Rejection Sampling HMC, simply computes the Hamiltonian trajectory to generate the next state \mathbf{q}^* and then rejects this state if $\mathbb{I}_F(\mathbf{q}^*) = 0$; that is if the new state does not lie in the current conditional level F_j .

The second, more complicated approach uses a novel algorithm to approximate the time at which the trajectory crosses the limit surface (potential barrier), calculates the momentum at this "hitting time," and bounces the trajectory off the potential barrier, after which the trajectory continues, and a new state \mathbf{q}^* is proposed. This state is then again rejected if $\mathbb{I}_F(\mathbf{q}^*) = 0$. This method, referred to as the Barrier Bouncing HMC, will not be discussed in detail here. In our applications, we apply a Rejection Sampling-based approach for simplicity, although there is no reason that the barrier-bouncing approach could not be applied in our setting.

The second issue in applying HMC for Subset Simulation is the appropriate selection of the trajectory length in the subsets. As stated previously, one could naively apply the HMC using Algorithm 3. However,

this may result in relatively long trajectories that wander out of the conditional level, yielding a high rejection

²⁵⁵ rate. On the other hand, selecting very short trajectories yields very high acceptance rates, causing a strong

²⁵⁶ correlation between samples. Wang et al. propose to aim for a relatively constant acceptance rate targeted

in the range between $a_{low} = 0.3$ and $a_{high} = 0.5$. To achieve this target acceptance rate, they proposed the

²⁵⁸ method in Algorithm 4. The initial trajectory length is set based on Algorithm 3.

Algorithm 3 Initialize trajectory length and set leapfrog steps

Step counter: j, Step size: Δt , Mean period: \overline{T} , Factor for division: kfor $i \leftarrow 1, N$ do $(\mathbf{q}, \mathbf{p}) \leftarrow (\mathbf{q}^{(i)}, \mathbf{p}^{(i)})$ j = 1 $(\mathbf{q}_+, \mathbf{p}_+) \leftarrow$ One leapfrog step (\mathbf{q}, \mathbf{p}) $(\mathbf{q}_-, \mathbf{p}_-) \leftarrow$ One leapfrog step $(\mathbf{q}, -\mathbf{p})$ while $\mathbf{p}_+^T(\mathbf{q}_+ - \mathbf{q}_-)$ or $\mathbf{p}_-^T(\mathbf{q}_- - \mathbf{q}_+)$ do j+=1 $(\mathbf{q}_+, \mathbf{p}_+) \leftarrow$ One leapfrog step $(\mathbf{q}_+, \mathbf{p}_+)$ $(\mathbf{q}_-, \mathbf{p}_-) \leftarrow$ One leapfrog step $(\mathbf{q}_-, \mathbf{p}_-)$ end while $T_i \leftarrow 2j\Delta t$ end for $\overline{T} \leftarrow \frac{\sum T_i}{N}$ $t_f \leftarrow \overline{T}/k$ $L \leftarrow \lfloor \frac{t_f}{\Delta t} \rceil$ \triangleright S

 \triangleright Set leapfrog steps to nearest integer

Algorithm 4 Adaptive rule for trajectory length

return $[\bar{T}, L]$

Subset number: j, Step size: Δt , Mean period: \overline{T} if $a_{low} < a < a_{high}$ then return No update for leapfrog steps else if $a_{low} < a$ then $t_f \leftarrow \frac{\overline{T}}{2\pi} \sin^{-1} \left(\sin \left(\frac{2\pi t_f}{\overline{T}} \right) \right) \exp((a - a_{low})/2))$ else if $a_{high} > a$ then $t_f \leftarrow \frac{\overline{T}}{2\pi} \sin^{-1} \left(\sin \left(\frac{2\pi t_f}{\overline{T}} \right) \right) \exp((a - a_{high})/2))$ end if end if $L \leftarrow \lfloor \frac{t_f}{\Delta t} \rceil$

 \triangleright Set Leapfrog steps to nearest integer

259 4.2. Proposed Approach

- $_{260}$ The proposed approach to integrate $H_{NN}MC$ into Subset Simulation proceeds as follows:
- 1. Pretrain an HNN by running some set of Hamiltonian trajectories.
- 262 2. Generate an initial set of samples according to the probability distribution $\pi(\mathbf{x})$ using H_{NN}MC.
- ²⁶³ 3. For each conditional level, draw samples using either the Rejection Sampling HMC or the Barrier
- Bouncing HMC. However, within the time integration for the Hamiltonian trajectories, compute the
- gradients of the Hamiltonian using the pre-trained HNN.

These conceptually simple steps are implemented using the pieces described above and can be augmented with certain optional enhancements as described next.

268 4.2.1. Online Error Monitoring

A general limitation of neural networks is that they interpolate well but often extrapolate inaccurately [53, 54]. Hence, circumstances arise where the HNN gradient predictions become inaccurate as a result of insufficient training data. This is particularly true in the tails of the distribution whose states have not previously been observed. For such cases, Dhulipala et al. [47] proposed an online error monitoring scheme based on slice sampling, which was originally proposed to monitor integration errors during the standard HMC. At the current state, a slice variable u is drawn from the Uniform distribution written as:

$$u \sim \mathcal{U}(0, \exp(H(\mathbf{q}_0, \mathbf{p}_0)))$$
, (18)

where $H(\mathbf{q}_0, \mathbf{p}_0)$ is the Hamiltonian at the current state. Defining a threshold $\Delta \varepsilon$, if

$$H\left(\mathbf{q}_{n},\mathbf{p}_{n}\right)\right)+\ln(u)>\Delta\varepsilon,$$
(19)

then the algorithm reverts to conventional numerical integration for the current time step. This increases computational cost but improves robustness, particularly for strongly non-Gaussian and high-dimensional

²⁷⁸ distributions where exploration is difficult.

279 4.2.2. Neural network updates

Aside from the error monitoring scheme, the neural network parameters can be updated during the Subset Simulations. For example, whenever a new subset level is reached, a share of the new samples can be used to create new training data using the traditional gradient calculation. Afterward, the data can be used to retrain the neural network. This procedure leads to higher accuracy in the region of interest and, therefore, reduces the number of rejections.

Another possible approach is to use the online error monitoring scheme and neural network updates together. In such cases, based on the history of online error monitoring, the scheme decides whether a neural network update is necessary, e.g., when an error monitoring threshold has been exceeded several times, neural network retraining may be triggered.

$_{289}$ 4.3. Benefits of $H_{NN}MC$ for Subset Simulation

The main benefit of using $H_{NN}MC$ is the computational savings compared to standard HMC. Our first examples will show the significant speed-up of the sampling while the limit state function of the chosen problems is rather cheap. In these examples, we demonstrate the performance of $H_{NN}MC$ on complex distributions and show that the HNNs are able to estimate gradients for rare events.

In general, most computational expense comes from model evaluation, i.e., the evaluation of the limit state function. To this end, Wang et al. state that "in practice, the main computational effort in Subset Simulation is usually the evaluation of limit-state functions, and each leapfrog step (except the last step) does not involve limit-state function evaluation, thus the additional cost introduced by using a relatively small Δt is often negligible" [40]. This is true for problems in which the probability density function is well-known a priori, as in the traditional Subset Simulation methods where all computations are performed using standard

normal random variables. However, this is not the case in the more general reliability problem in which the 300 probability density is known only implicitly (i.e. through the computational model), known only up to a 301 scale factor, or must be inferred from data. In such cases, for example in inference problems where the 302 system parameters are unknown, the model needs to be evaluated to obtain the gradients required for HMC. 303 Consequently, each leapfrog step becomes extremely expensive. H_{NN}MC, on the other hand, is capable of 304 learning the Hamiltonian gradients that involve limit state function (model) evaluation. This way, model 305 evaluation is only required during HNN training and in the last (acceptance/rejection) step. This makes 306 $\mathrm{H}_{\mathrm{NN}}\mathrm{MC}\text{-}\mathrm{based}$ Subset Simulation substantially faster computationally. 307

308 5. Proof of concept

In this section, we provide a rigorous proof of concept presenting several numerical examples of reliability 309 problems using the proposed method. For the chosen examples, we use the reliability index $\beta = \Phi^{-1}(P_F)$ and 310 compare the results for accuracy with the modified Metropolis-Hastings (MMH) sampler [25] and for compu-311 tational performance with existing HMC methods [40]. Using the reliability index, instead of the probability 312 of failure itself, allows us to make meaningful comparisons of the coefficient of variation, which can break down 313 for P_F [39]. The problems are implemented in Python using the UQPy package [55, 56]. Parts of the code for 314 latent Hamiltonian Neural Networks have been adopted from the GitHub repository Bayesian inference with 315 Hamiltonian Neural Networks (BIhNNs) available at (https://github.com/IdahoLabResearch/BIhNNs). The 316 computational time of the examples was assessed on an Apple MacBook Pro equipped with an M1 chip and 317 16 GB of memory. 318

For all examples, we use 10^3 samples from the original distribution and 10^3 samples in each conditional 319 level within the rejection sampling Subset Simulation as proposed by Wang et al. [40]. The conditional 320 probability is fixed at 0.1. Unless otherwise specified, 4×10^5 gradient evaluations of the Hamiltonian are 321 used to generate the training data set for the HNN. However, complex and high-dimensional distributions 322 may require a larger training set for the HNN. The HNN is composed of three hidden layers and 100 neurons 323 in each layer. In all cases, these settings give reasonable results. However, we have not performed a more 324 rigorous network optimization on the HNN. Thus, performance may be improved through a better network 325 design aimed at optimizing the learning rate. This problem is beyond the scope of this paper since it is 326 independent of the approaches proposed here. 327

³²⁸ 5.1. Linear Limit State with Degenerate Gaussian Distributions

The first numerical example aims to explore the performance of $H_{NN}MC$ -based Subset Simulation for problems where the distribution form is simple, but sampling is difficult due to the degeneracy of the distribution. That is, the distribution is defined in a space with dimension D, but its support lies primarily on a space with dimension d < D. For this, we consider a random vector X with multivariate normal distributions having mean vector and covariance matrix given by:

$$\boldsymbol{\mu} = [0, 0, \cdots, 0, 0]; \qquad \mathbf{C} = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \rho \\ \rho & \dots & \rho & 1 \end{bmatrix}.$$
(20)

where the correlation ρ is modified to control the dependence between the random variables. Importantly, as $\rho \to 1$ the distribution degenerates and becomes effectively one-dimensional, sampling using standard MCMC methods (e.g. Metropolis-Hastings) becomes a challenge. The linear limit state function is given by $g(\mathbf{x}) = \beta \sqrt{\sigma_{max}n} - \sum_{i=1}^{n} x_i$, where $\beta = \{3, 4, 5\}$ is the reliability index, n is the problem dimension, and σ_{max} is the largest eigenvalue of **C**. The reliability problem is illustrated for n = 2 in Figure 4, which shows the contours of the joint pdf and the limit state function for different values of β and correlation of (a) $\rho = 0$, (b) $\rho = 0.75$, and (c) $\rho = 0.95$.



Figure 4: Illustration of the reliability problem for a bi-variate normal distribution with linear limit state function for different correlations: (a) $\rho = 0$; (b) $\rho = 0.75$; (c) $\rho = 0.95$. (d) Illustration of the samples generated from Subset Simulation using the proposed H_{NN}MC for a two-dimensional uncorrelated standard normal random vector having linear limit state function with $\beta = 5$.

The generated samples for one Subset Simulation run using the proposed H_{NN}MC with n = 2, $\beta = 5$, and $\rho = 0$ are shown in Figure 4(d). The acceptance rate of the samples within the conditional levels varies between 0.96 and 0.99 for the HMC acceptance criteria and between 0.16 and 0.52 for the subset rejection criteria. The predicted probability of failure is 3.12×10^{-8} which corresponds to $\beta = 5.41$.

This example serves as a first demonstration that the proposed $H_{NN}MC$ can be used to estimate small failure probabilities such that we can test it on more challenging cases where existing methods encounter difficulties. Considering different correlations ρ and dimensions n, we demonstrate its performance by repeating each case 100 times to estimate the coefficient of variation.

The results of the Subset Simulations using MMH and $H_{NN}MC$ are summarized in Fig. 5, which shows the 349 mean value and coefficient of variation of beta for different values of ρ (plotted in log scale in terms of $1-\rho$) 350 for low (n = 2), medium (n = 10), and high-dimensional (n = 100) cases. As expected, the MMH gives good 351 estimates for the failure probability for uncorrelated random variables, regardless of the dimension. However, 352 as the correlation increases the MMH quickly loses accuracy, especially in high dimensions. In contrast, the 353 H_{NN}MC sampling strategy gives very good results in almost all problems and only starts to lose accuracy 354 when n = 100 and $\beta \ge 4$. For the two and ten-dimensional correlated distributions, the H_{NN}MC sampler 355 gives very accurate results such that solid lines in the upper plots of Figure 4 (mean values) are almost flat at 356 the correct β value. Furthermore, the coefficients of variation of the estimators from H_{NN}MC are relatively 357 low and only increase mildly with dimension. Meanwhile, for the 100-dimensional distribution, the CoV 358 from MMH drops to zero as the correlation increases meaning that the method gives the wrong answer with 359 little to no variability. 360

Notably, for this example, the primary objective is to demonstrate that the $H_{NN}MC$ can be used to achieve accurate reliability estimates. Computational efficiency is not the primary concern. Nonetheless, the computational efficiency of using the proposed $H_{NN}MC$ method in Subset Simulations is compared with standard HMC in Appendix A.

365 5.2. Two-dimensional Rosenbrock distribution

For this example, we study a problem with two random variables following a Rosenbrock distribution having the unscaled probability density function written as:

$$p(\mathbf{x}) \propto \frac{k(x_2 - x_1^2)^2 + (1 - x_1)^2}{20}$$
 (21)



Figure 5: Results of 100 Subset Simulations using MMH and $H_{NN}MC$ for correlated Gaussian distributions: the mean reliability indices are compared in the upper plots while the lower plots show the coefficient of variation for (a) $\beta = 3$, (b) $\beta = 4$, and (c) $\beta = 5$.

The distribution has a banana or boomerang shape, and depending on the factor k, the distribution can be wide or very thin. We define a linear limit state function given by $g(\mathbf{x}) = 120 - x_2 - 3x_1$. For this problem, the MMH algorithm can give reliable failure probabilities only for $k \leq 1$ [39].

We first consider the Rosenbrock distribution with k = 1 having "true" reliability index of $\beta = 2.706$ 371 $(P_F = 3.4 \times 10^{-3})$ obtained from Monte Carlo simulations. Samples from one Subset Simulation using 372 $H_{\rm NN}MC$ are shown in Figure 6(a) yielding very accurate results with $\beta = 2.808 \ (P_F = 2.48 \times 10^{-3})$. Using 373 the same architecture for 100 Subset Simulations trials yields the mean reliability index $\beta = 2.953$ and 374 coefficient of variation $CV(\beta) = 0.0932$. To further test the approach, we increase k to 10, where the 375 MMH algorithm fails to adequately sample from the failure region, and, therefore, the Subset Simulation 376 with MMH breaks down. Figure 6(b) shows one subset using the H_{NN}MC approach yielding $\beta = 3.039$ 377 $(P_F = 1.18 \times 10^{-3})$, while the "true" value is $\beta = 2.6755$ $(P_F = 3.73 \times 10^{-3})$. The mean reliability index 378 from 100 trials is $\bar{\beta} = 3.435$ and the coefficient of variation is $CV(\beta) = 0.127$, which compare favorably to 379 state-of-the-art methods such as the affine invariant stretch sampler, which gives $\bar{\beta} = 3.466$, $CV(\beta) = 0.138$ 380 [39]. However, using the standard HMC achieves better results for k = 10, implying the limitations of the 381 $H_{NN}MC$ for complex distributions, e.g., for the Rosenbrock distribution with k = 100, discussed in the next 382 section. Some additional remarks on computational cost for this example are provided in Appendix A. 383



Figure 6: Subset results for the Rosenbrock distribution using $H_{NN}MC$: (a) Samples from each subset with k = 1; (b) Samples from each subset with k = 10.

5.3. Limitations of the $H_{NN}MC$ in Subset Simulations for multi-dimensional Gauss and Rosenbrock distributions

The proposed $H_{NN}MC$ algorithm reaches its limitations for strong correlation with high dimension and 386 low probability of failure, e.g., the 100-dimensional problem with a correlation of $\rho = 0.999$ and $\beta = 5$. 387 The same applies to Rosenbrock distributions with high k. While the HMC method still gives reasonable 388 389 results, the proposed $H_{NN}MC$ faces higher rejection rates in the rare event space since it is particularly challenging to learn the steep gradients of the Hamiltonian correctly. In Section 4.2.1 and 4.2.2, we propose 390 two strategies to mitigate this issue: (i) updating the neural network parameters in each subset and (ii) an 391 online error monitoring scheme that deploys conventional HMC as needed. The limitations and strategies 392 are demonstrated in Appendix B where both approaches are applied to the two-dimensional Rosenbrock 393 distribution with k = 100. 394

³⁹⁵ 5.4. Additional examples for proof of concept

Additional examples related to structural dynamics are provided in the appendix. The reliability problems focus on system parameters, following different distributions in Appendix C and uncertainty introduced within the load with high dimensional problems in Appendix D. As shown in these appendices, the proposed H_{NN}MC Subset Simulation provides accurate estimations of the failure probability in both examples.

400 6. Subset Simulations for systems with uncertain parameters

The major advantage of the proposed method arises when model evaluations are necessary for Hamiltonian gradient evaluation. As highlighted by Wang et al. [40], this is not generally the case for reliability problems on deterministic systems having uncertain inputs with well-known distributions. However, when performing reliability analysis on a system whose parameters must be inferred from data, we cannot avoid performing model evaluations at each step of HMC. That is, model evaluation is *necessary* for gradient evaluation.

Furthermore, we cannot use automatic differentiation tools since model evaluation is required. Therefore, we use a central difference scheme to calculate the numerical gradients. Hence, the evaluation requires multiple model evaluations for each gradient computation, which becomes computationally excessively expensive for high dimensional problems.

To illustrate this case, we apply the proposed $H_{NN}MC$ -based Subset Simulation approach to assess the reliability of a single-degree-of-freedom system having a Bouc-Wen hysteretic material model (illustrated in

Figure 7a) whose parameters must be inferred from data. The equation of motion of the highly nonlinear system is written as [55]:

$$m\ddot{u}(t) + c\dot{u}(t) + kr(t) = -m\ddot{u}_g(t) , \quad \text{with}$$
(22)

$$\dot{r}(t) = \dot{u} - \beta |\dot{u}| |r(t)|^{n-1} r(t) - \gamma \dot{u}(t) |r(t)|^n .$$
(23)

where k is the spring stiffness and $n \beta$ and γ are the hysteresis parameters of the Bouc-Wen model. Alternatively, we parameterize the system equivalently by n, r_0 and δ , where $r_0 = \sqrt[n]{\frac{1}{\beta + \gamma}}$ and $\delta = \frac{\beta}{\beta + \gamma}$.



Figure 7: Bouc-Wen model for Bayesian Inference

The parameters of the spring and the Bouc-Wen material model are assumed to be unknown and must be learned from data. These data are generated by creating an artificial measurement of the system response to the scaled El Centro ground motion record [57] shown in Figure 7(b) with parameters $k = 1 \frac{\text{N}}{\text{m}}$, $r_0 =$ 2.5 cm, $\delta = 1$, and n = 2. The original system uses small viscous damping, which introduces a small error since the inference model assumes no damping. The simulated noisy response measurement, including 5% root mean squared noise, is shown in Figure 7(c).

Using the observed data, we first employ traditional HMC to infer the joint distribution of the material parameters. This HMC comes at a significant computational cost since each leapfrog step requires evaluation of the model. We draw a total of 5000 samples by employing 50 leapfrog steps for each proposal, gathering 250 000 gradient evaluation in total. We then use these gradient evaluations to train the HNN – hence the HMC cost is not wasted. For reference, the $H_{NN}MC$ is tested on the Bayesian inference problem first, where we again generate 5000 samples from the posterior parameter distribution. These samples, which are



Figure 8: Subset Simulations for Bayesian inference; Bottom left: Drawn samples using $H_{NN}MC$ in Bayesian inference sampling; Diagonal: Histograms of 5000 samples using $H_{NN}MC$; Top right: Subset Simulations using $H_{NN}MC$ with 1000 samples.

424 generated extremely fast because they do not require model evaluation for the Hamiltonian gradients, are

shown in the bottom left plots of Figure 8, and histograms for each variable are shown on the diagonal.

The observed results show that the $H_{NN}MC$ distributions are consistent with those observed from previous studies [55].

⁴²⁸ Next, we conduct reliability analysis using the inferred joint parameter distribution. The limit state

function corresponds to the exceedance of a displacement limit, $u_{lim} = 5$, in the time range $t \in [0, 40]$ sec under the same El Centro excitation and is expressed as:

$$g(\mathbf{x}) = u_{lim} - \max_{t \in [0,40]} |u(\mathbf{x},t)| .$$
(24)

where **x** is the vector of uncertain system parameters. We apply the H_{NN}MC sampling strategy within Subset Simulation using 1000 samples per subset. Repeating the analysis for 100 trials, we observe a mean reliability index of $\bar{\beta} = 4.35$ with coefficient of variation $CV(\beta) = 0.048$. A representative sample set from Subset Simulation with samples colored by subset is shown in the upper off-diagonal plots in Figure 8.

The H_{NN}MC results are in agreement with Subset Simulations performed with HMC using the traditional 435 gradient updates, which produced a mean reliability index of $\beta = 4.44$ with coefficient of variation $CV(\beta) =$ 436 0.039 from ten independent trials. However, the cost of Subset Simulation with $H_{NN}MC$ is dramatically 437 lower. Traditional HMC requires that the system response be evaluated at every time step of the Hamiltonian 438 trajectory. Therefore, one Subset Simulation takes $t_{\rm HMCSS} = 579549$ sec (> 6 days). As a result, we could 439 only afford 10 independent trials with standard HMC. Because the trained HNN automatically computes the 440 gradients, and no model evaluation is needed, The $H_{NN}MC$ method takes only $t_{HMCSS} = 4509$ sec (≈ 1.25 441 hours) per Subset Simulation – an improvement of more than 100x. 442

443 7. Conclusion

In this paper, we developed Hamiltonian Neural Network Markov Chain sampling for reliability analysis using Subset Simulations. The proposed strategy reveals high efficiency and accurate estimates for the probability of failure.

Hamiltonian Neural Networks are particularly powerful for gradient predictions of long trajectories [46],
 so marriage with Hamiltonian Monte Carlo is highly beneficial. As a result of this, the acceptance rate of
 the non-random walk proposals remains high.

The main benefit of using this approach is to decrease the computational efforts for the gradient calcula-450 tion during the sampling. For the examples in this paper, the new approach is at least 20 times faster when 451 performing Subset Simulations. However, a fair metric for the speed-up of the approach also considers the 452 training procedure of the Hamiltonian Neural Network, which includes the traditional gradient calculation 453 during the generation of the training set. The gradient evaluations required for the Subset Simulations are – 454 dependent on the settings – far more than the number of samples for the training set. Notably, the strategy 455 becomes increasingly beneficial if the same distribution is used for several simulations since the training 456 procedure is only required once. 457

In the case of very complex and high-dimensional distributions, such as the Rosenbrock and the highly correlated 100-dimensional Gaussian distribution, the approach reaches its limitations in higher subset levels. The most reliable solution for this problem is to include an online error monitoring scheme. Although this extension weakens the computational efficiency of the H_{NN}MC subset method proposed, online error monitoring enables comparable results to the standard HMC method, even in the most unfavorable situations, i.e., for very complex distributions. However, for most updates, the online error scheme is not required.

The H_{NN}MC proposed in this paper is particularly valuable when applied to problems where the underlying distribution is *a priori* unknown. For such Bayesian inference problems, the numerical gradient calculation becomes extremely expensive when applying the traditional HMC method since the underlying model has to be evaluated. These gradient evaluations are done by the Hamiltonian Neural Network in a fractional amount of time, enabling outstandingly fast and efficient sampling. Thus, the algorithm proposed in this paper reveals the highest speed up in the Bayesian inference framework since it decreases the computational effort to 1% compared to the time necessary using the traditional HMC sampler.

471 CRediT authorship contribution statement

Denny Thaler: Conceptualization, Methodology, Software, Data curation, Writing- Original draft preparation, Funding acquisition. **Somayajulu LN Dhulipala**: Methodology, Writing- Reviewing and Editing, Funding acquisition. Franz Bamer: Conceptualization, Supervision, Writing- Reviewing and
 Editing. Bernd Markert: Writing- Reviewing and Editing. Michael D. Shields: Conceptualization,
 Methodology, Supervision, Writing- Reviewing and Editing, Funding acquisition.

477 Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

480 Acknowledgement

⁴⁸¹ Denny Thaler was supported by a fellowship of the German Academic Exchange Service (DAAD) during
⁴⁸² his research stay at Johns Hopkins University. Somayajulu Dhulipala and Michael Shields were supported
⁴⁸³ through INL's Laboratory Directed Research and Development (LDRD) Program under U.S. Department
⁴⁸⁴ of Energy (DOE) Idaho Operations Office contract no. DE-AC07-05ID14517. Franz Bamer was supported
⁴⁸⁵ by the Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the
⁴⁸⁶ German State (MKW) under the Excellence Strategy of the Federal Government and the Länder.

487 References

- [1] A. M. Hasofer, N. C. Lind, Exact and invariant second-moment code format, Journal of the Engineering
 Mechanics Division 100 (1) (1974) 111–121. doi:10.1061/JMCEA3.0001848.
- [2] H. U. Köylüoğlu, S. R. Nielsen, New approximations for sorm integrals, Structural Safety 13 (4) (1994)
 235-246. doi:https://doi.org/10.1016/0167-4730(94)90031-0.
- ⁴⁹² URL https://www.sciencedirect.com/science/article/pii/0167473094900310
- ⁴⁹³ [3] O. Ditlevsen, H. Madsen, Structural Reliability Methods, John Wiley & Sons Ltd, 1996.
- [4] G. Kerschen, J.-c. Golinval, A. F. VAKAKIS, L. A. BERGMAN, The method of proper orthogonal
 decomposition for dynamical characterization and order reduction of mechanical systems: An overview,
 Nonlinear Dynamics 41 (1) (2005) 147–169. doi:10.1007/s11071-005-2803-2.
- ⁴⁹⁷ URL https://doi.org/10.1007/s11071-005-2803-2
- [5] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized
 decomposition, Archives of Computational Methods in Engineering 18 (4) (2011) 395–404. doi:10.
 1007/s11831-011-9064-7.
- ⁵⁰¹ URL https://doi.org/10.1007/s11831-011-9064-7
- [6] F. Bamer, C. Bucher, Application of the proper orthogonal decomposition for linear and nonlinear structures under transient excitations, Acta Mechanica 223 (12) (2012) 2549–2563. doi:10.1007/s00707-012-0726-9.
- ⁵⁰⁵ URL https://doi.org/10.1007/s00707-012-0726-9
- [7] H. Jensen, A. Muñoz, C. Papadimitriou, E. Millas, Model-reduction techniques for reliability-based
 design problems of complex structural systems, Reliability Engineering & System Safety 149 (2016)
 204-217. doi:https://doi.org/10.1016/j.ress.2016.01.003.
- 509 URL https://www.sciencedirect.com/science/article/pii/S0951832016000120
- [8] F. Bamer, B. Markert, An efficient response identification strategy for nonlinear structures subject to nonstationary generated seismic excitations, Mechanics Based Design of Structures and Machines
 45 (3) (2017) 313-330. arXiv:https://doi.org/10.1080/15397734.2017.1317269, doi:10.1080/ 15397734.2017.1317269.
- ⁵¹⁴ URL https://doi.org/10.1080/15397734.2017.1317269

- [9] K. Skandalos, S. Chakraborty, S. Tesfamariam, Seismic reliability analysis using a multi-fidelity sur-
- rogate model: Example of base-isolated buildings, Structural Safety 97 (2022) 102222. doi:https:
- 517 //doi.org/10.1016/j.strusafe.2022.102222.
- URL https://www.sciencedirect.com/science/article/pii/S0167473022000340
- [10] F. Bamer, A. K. Amiri, C. Bucher, A new model order reduction strategy adapted to nonlinear problems in earthquake engineering, Earthquake Engineering & Structural Dynamics 46 (4) (2017) 537 559. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.2802, doi:https://doi.
 org/10.1002/eqe.2802.
- ⁵²³ URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.2802
- [11] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, J. M. McFarland, Efficient global reliability
 analysis for nonlinear implicit performance functions, AIAA journal 46 (10) (2008) 2459–2468.
- ⁵²⁶ [12] B. Echard, N. Gayton, M. Lemaire, Ak-mcs: an active learning reliability method combining kriging ⁵²⁷ and monte carlo simulation, Structural Safety 33 (2) (2011) 145–154.
- [13] D. Thaler, F. Bamer, B. Markert, A comparison of two neural network architectures for fast structural response prediction, PAMM 21 (1) (2021) e202100137. arXiv:https://onlinelibrary.wiley.com/ doi/pdf/10.1002/pamm.202100137, doi:https://doi.org/10.1002/pamm.202100137.
 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202100137
- [14] V. Sundar, M. D. Shields, Surrogate-enhanced stochastic search algorithms to identify implicitly defined
 functions for reliability analysis, Structural Safety 62 (2016) 1–11. doi:https://doi.org/10.1016/j.
 strusafe.2016.05.001.
- URL https://www.sciencedirect.com/science/article/pii/S0167473016300054
- [15] R. Schöbi, B. Sudret, S. Marelli, Rare event estimation using polynomial-chaos kriging, ASCE-ASME
 Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 3 (2) (2017)
 D4016002.
- [16] D. Thaler, F. Bamer, B. Markert, A machine learning enhanced structural response prediction strategy due to seismic excitation, PAMM 20 (1) (2021) e202000294. arXiv:https://onlinelibrary.wiley.
 com/doi/pdf/10.1002/pamm.202000294, doi:https://doi.org/10.1002/pamm.202000294.
- URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202000294
- [17] M. Li, R.-Q. Wang, G. Jia, Efficient dimension reduction and surrogate-based sensitivity analysis for
 expensive models with high-dimensional outputs, Reliability Engineering & System Safety 195 (2020)
 106725. doi:https://doi.org/10.1016/j.ress.2019.106725.
- 546 URL https://www.sciencedirect.com/science/article/pii/S0951832018315758
- [18] F. Bamer, D. Thaler, M. Stoffel, B. Markert, A monte carlo simulation approach in non-linear structural dynamics using convolutional neural networks, Frontiers in Built Environment 7 (2021). doi:10.3389/
 fbuil.2021.679488.
- URL https://www.frontiersin.org/articles/10.3389/fbuil.2021.679488
- [19] B. Peherstorfer, T. Cui, Y. Marzouk, K. Willcox, Multifidelity importance sampling, Computer Methods
 in Applied Mechanics and Engineering 300 (2016) 490–509.
- [20] B. Kramer, A. N. Marques, B. Peherstorfer, U. Villa, K. Willcox, Multifidelity probability estimation via fusion of estimators, Journal of Computational Physics 392 (2019) 385–402.
- [21] T. Pham, A. A. Gorodetsky, Ensemble approximate control variate estimators: Applications to multifi delity importance sampling, SIAM/ASA Journal on Uncertainty Quantification 10 (3) (2022) 1250–1292.

- ⁵⁵⁷ [22] S. L. Dhulipala, M. D. Shields, B. W. Spencer, C. Bolisetti, A. E. Slaughter, V. M. Labouré,
 P. Chakroborty, Active learning with multifidelity modeling for efficient rare event simulation, Journal
- of Computational Physics 468 (2022) 111506. doi:https://doi.org/10.1016/j.jcp.2022.111506.
- URL https://www.sciencedirect.com/science/article/pii/S002199912200568X
- ⁵⁶¹ [23] S.-K. Au, J. L. Beck, A new adaptive importance sampling scheme for reliability calculations, Structural ⁵⁶² safety 21 (2) (1999) 135–158.
- ⁵⁶³ [24] A. Olsson, G. Sandberg, O. Dahlblom, On latin hypercube sampling for structural reliability analysis,
 ⁵⁶⁴ Structural safety 25 (1) (2003) 47–68.
- ⁵⁶⁵ [25] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, ⁵⁶⁶ Probabilistic engineering mechanics 16 (4) (2001) 263–277.
- [26] A. Robens-Radermacher, J. F. Unger, Efficient structural reliability analysis by using a pgd model in an adaptive importance sampling schema, Advanced Modeling and Simulation in Engineering Sciences
 7 (1) (2020) 29.
- ⁵⁷⁰ [27] D. Thaler, M. D. Shields, B. Markert, F. Bamer, Model order reduction in subset simulations using the ⁵⁷¹ proper orthogonal decomposition, PAMM n/a (n/a) (2023) e202300053.
- ⁵⁷² [28] B. Echard, N. Gayton, M. Lemaire, N. Relun, A combined importance sampling and kriging reliability
 ⁵⁷³ method for small failure probabilities with time-demanding numerical models, Reliability Engineering
 ⁵⁷⁴ & System Safety 111 (2013) 232–240.
- ⁵⁷⁵ [29] N.-C. Xiao, H. Zhan, K. Yuan, A new reliability method for small failure probability problems by
 ⁵⁷⁶ combining the adaptive importance sampling and surrogate models, Computer Methods in Applied
 ⁵⁷⁷ Mechanics and Engineering 372 (2020) 113336.
- [30] S. K. Au, J. L. Beck, Subset simulation and its application to seismic risk based on dynamic analysis,
 Journal of Engineering Mechanics 129 (8) (2003) 901–917. doi:10.1061/(ASCE)0733-9399(2003)129:
 8(901).
- [31] S. Au, J. Ching, J. Beck, Application of subset simulation methods to reliability benchmark problems,
 Structural Safety 29 (3) (2007) 183-193, a Benchmark Study on Reliability in High Dimensions. doi:
 https://doi.org/10.1016/j.strusafe.2006.07.008.
- 584 URL https://www.sciencedirect.com/science/article/pii/S0167473006000385
- [32] F. Cérou, P. Del Moral, T. Furon, A. Guyader, Sequential monte carlo for rare event estimation,
 Statistics and Computing 22 (3) (2012) 795–808. doi:10.1007/s11222-011-9231-6.
 URL https://doi.org/10.1007/s11222-011-9231-6
- [33] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of state cal culations by fast computing machines, The Journal of Chemical Physics 21 (6) (1953) 1087–1092.
 doi:10.1063/1.1699114.
- ⁵⁹¹ URL https://doi.org/10.1063/1.1699114
- [34] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika
 57 (1) (1970) 97-109. doi:10.1093/biomet/57.1.97.
- ⁵⁹⁴ URL https://doi.org/10.1093/biomet/57.1.97
- ⁵⁹⁵ [35] A. Santoso, K. Phoon, S. Quek, Modified metropolis-hastings algorithm with reduced chain correlation ⁵⁹⁶ for efficient subset simulation, Probabilistic Engineering Mechanics 26 (2) (2011) 331–341.
- ⁵⁹⁷ [36] F. Miao, M. Ghosn, Modified subset simulation method for reliability analysis of structural systems,
 ⁵⁹⁸ Structural Safety 33 (4-5) (2011) 251–260.

- [37] I. Papaioannou, W. Betz, K. Zwirglmaier, D. Straub, Mcmc algorithms for subset simulation, Probabilistic Engineering Mechanics 41 (2015) 89–103.
- [38] K. M. Zuev, J. L. Beck, S.-K. Au, L. S. Katafygiotis, Bayesian post-processor and other enhancements
 of subset simulation for estimating failure probabilities in high dimensions, Computers & structures 92
 (2012) 283–296.
- [39] M. D. Shields, D. G. Giovanis, V. Sundar, Subset simulation for problems with strongly non-gaussian, highly anisotropic, and degenerate distributions, Computers & Structures 245 (2021) 106431.
- [40] Z. Wang, M. Broccardo, J. Song, Hamiltonian monte carlo methods for subset simulation in reliability analysis, Structural Safety 76 (2019) 51–67.
- [41] W. Chen, Z. Wang, M. Broccardo, J. Song, Riemannian manifold hamiltonian monte carlo based subset simulation for reliability analysis in non-gaussian space, Structural Safety 94 (2022) 102134.
- [42] H. Strathmann, D. Sejdinovic, S. Livingstone, Z. Szabo, A. Gretton, Gradient-free hamiltonian monte
 carlo with efficient kernel exponential families, Advances in Neural Information Processing Systems 28
 (2015).
- [43] M. Broccardo, Z. Wang, S. Marelli, J. Song, B. Sudret, Hamiltonian monte carlo-based subset simulation
 using gaussian process metamodeling, in: 19th IFIP WG7.5 Conference on Reliability and Optimization
 of Structural Systems, 2018.
- [44] L. Li, A. Holbrook, B. Shahbaba, P. Baldi, Neural network gradient hamiltonian monte carlo, Computational Statistics 34 (1) (2019) 281–299. doi:10.1007/s00180-018-00861-z.

⁶¹⁸ URL https://doi.org/10.1007/s00180-018-00861-z

- [45] D. Levy, M. D. Hoffman, J. Sohl-Dickstein, Generalizing hamiltonian monte carlo with neural networks,
 International Conference on Learning Representations (2018).
- [46] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, CoRR abs/1906.01563 (2019).
 arXiv:1906.01563.
- 623 URL http://arxiv.org/abs/1906.01563
- [47] S. L. Dhulipala, Y. Che, M. D. Shields, Efficient bayesian inference with latent hamiltonian neural
 networks in no-u-turn sampling, Journal of Computational Physics 492 (2023) 112425. doi:https:
 //doi.org/10.1016/j.jcp.2023.112425.
- URL https://www.sciencedirect.com/science/article/pii/S002199912300520X
- [48] D. Thaler, S. L. N. Dhulipala, F. Bamer, B. Markert, M. D. Shields, Enhanced hamiltonian monte
 carlo simulations using hamiltonian neural networks, PAMM 22 (1) (2023) e202200188. arXiv:https:
 //onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202200188, doi:https://doi.org/10.1002/
 pamm.202200188.
- URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202200188
- [49] C. Bucher, Computational Analysis of Randomness in Structural Mechanics, 1st Edition, Vol. 3 of
 Structures and Infrastructures Book Series, CRC Press, 2009.
- [50] R. M. Neal, Mcmc using hamiltonian dynamics, in: S. Brooks, A. Gelman, G. Jones, X.-L. Meng (Eds.),
 Handbook of markov chain monte carlo, Chapman and Hall/CRC, New York, 2011, Ch. 5, pp. 113–160.
- ⁶³⁷ [51] M. Betancourt, A conceptual introduction to hamiltonian monte carlo (2018). arXiv:1701.02434.
- [52] M. D. Hoffman, A. Gelman, et al., The no-u-turn sampler: adaptively setting path lengths in hamiltonian
 monte carlo., J. Mach. Learn. Res. 15 (1) (2014) 1593–1623.

- [53] D. Thaler, M. Stoffel, B. Markert, F. Bamer, Machine-learning-enhanced tail end prediction of structural response statistics in earthquake engineering, Earthquake Engineering & Structural Dynamics 50 (8) (2021) 2098-2114. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.3432,
- doi:https://doi.org/10.1002/eqe.3432.
- URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3432

[54] D. Thaler, L. Elezaj, F. Bamer, B. Markert, Training data selection for machine learning-enhanced monte
 carlo simulations in structural dynamics, Applied Sciences 12 (2) (2022). doi:10.3390/app12020581.
 URL https://www.mdpi.com/2076-3417/12/2/581

- [55] A. Olivier, D. G. Giovanis, B. Aakash, M. Chauhan, L. Vandanapu, M. D. Shields, Uqpy: A general purpose python package and development environment for uncertainty quantification, Journal of Computational Science 47 (2020) 101204. doi:https://doi.org/10.1016/j.jocs.2020.101204.
 URL https://www.sciencedirect.com/science/article/pii/S1877750320305056
- [56] D. Tsapetis, M. D. Shields, D. G. Giovanis, A. Olivier, L. Novak, P. Chakroborty, H. Sharma,
 M. Chauhan, K. Kontolati, L. Vandanapu, et al., Uqpy v4. 1: Uncertainty quantification with python,
 SoftwareX 24 (2023) 101561.
- [57] Center for Engineering Strong Motion Data, Northridge, California 1940-05-18 2037 PST, Tech. rep.,
 Enineering Data Center (1940).
- [58] A. D. Kiureghian, M. D. Stefano, Efficient algorithm for second‐order reliability analysis,
 Journal of Engineering Mechanics 117 (12) (1991) 2904–2923. doi:10.1061/(ASCE)0733-9399(1991)
 117:12(2904).
- [59] M. Shinozuka, G. Deodatis, Simulation of Stochastic Processes by Spectral Representation,
 Applied Mechanics Reviews 44 (4) (1991) 191-204. arXiv:https://asmedigitalcollection.
 asme.org/appliedmechanicsreviews/article-pdf/44/4/191/5435905/191_1.pdf, doi:10.1115/
 1.3119501.
- ⁶⁶⁴ URL https://doi.org/10.1115/1.3119501

$_{665}$ Appendix A. Computational efficiency of $H_{NN}MC$ compared on proof of concept examples

666 Appendix A.1. Degenerate Gaussian

The computational cost of H_{NN}MC for Subset Simulation is compared with standard HMC based on the 667 number of Hamiltonian gradient evaluations for cases with n = 100, $\rho = 0$, and $\beta = 3,5$ in Table A.1. Clearly, 668 both H_{NN}MC and HMC are capable of drawing quality samples in this uncorrelated high-dimensional case. 669 The $H_{NN}MC$ only requires traditional gradient evaluations for the training data. Here, we used 4×10^5 670 gradient evaluations. It can then be directly deployed for all problems based on this distribution without 671 any additional computations. Meanwhile, the traditional gradient updates using HMC highly depend on the 672 settings of the algorithm. Using as few as possible gradient evaluations, i.e., one chain with 100 burn-in steps. 673 results in the number of gradient evaluations presented in Figure A.1. Hence, the proposed method only 674 leads to a speed-up if the probability of failure becomes low. Of course, the $H_{NN}MC$ sampler is significantly 675 more efficient when used for several trials since the HNN does not need to be retrained for each trial. 676

Regarding computational time, the HNN prediction and the Hamiltonian gradient evaluation are compared for the uncorrelated Gaussian distribution ($\rho = 0$) with 100 variables. The numerical evaluation of 10³ gradients takes 2 896 ms, whereas the HNN performs this task in 16 ms which is a reduction in cost by a factor of more than 180. Assuming the HNN is already trained for this distribution, full Subset Simulations using the H_{NN}MC run in 489 sec on average, while the traditional HMC takes 30 543 sec on average.

Table A.1: Comparison of the required Hamiltonian gradient evaluations for Subset Simulations using HMC and $H_{NN}MC$ for correlated Gaussian random variables with a linear limit state having n = 100, $\rho = 0$, and $\beta = 3, 5$.

| Hamiltonian | | | | number | average |
|---------------------------------------|-----------------|---------|--------|---------|-----------------|
| gradient | training | initial | subset | of | for one |
| evaluations | data | set | | subsets | simulation |
| $\mathrm{H}_{\mathrm{NN}}\mathrm{MC}$ | 4×10^5 | 0 | 0 | 2.8/7.5 | 4×10^5 |
| HMC $(\beta = 3)$ | 0 | 91516 | 79856 | 2.8 | 315113 |
| HMC $(\beta = 5)$ | 0 | 91075 | 78 282 | 7.5 | 678190 |

682 Appendix A.2. Rosenbrock

The speed-up of the method for this example in terms of traditional gradient calculations depends on 683 the number of samples used for the neural network training. For this distribution, using $8 \cdot 10^5$ gradients for 684 training led to good accuracy during the supervised learning procedure. The traditional approach depends 685 on the settings for the Hamiltonian Monte Carlo, especially since the initial set requires a relatively high 686 burn length for this distribution to find adequate initial samples. The number of leapfrog steps is calculated 687 using Algorithm 3 with a fixed step length $\Delta t = 0.01$. Starting with 20 independent samples from the 688 Gaussian distribution and using a burn length of 500 results in 881619 traditional gradient calculations. 689 The number of gradient evaluations could be further reduced if only one single chain is used, reducing the 690 number of gradients required for the initial set to 120220. However, within each subset, the chains operate 691 independently, meaning that the number of required gradients is solely influenced by the trajectory and 692 the step length. For the chosen example, the algorithm evaluated seven levels with the following gradient 693 evaluations [205 200, 410 364, 407 220, 483 184, 471 440, 539 776, 550 668], which is 3 067 852 evaluations for all 694 subset levels. Thus, the number of required conventional gradient evaluations is dependent on the number 695 of chains chosen for the evaluation of the initial subset, but at least 4 times higher compared to the HNN 696 approach. 697

To compare the computational time of the two approaches, we used 10^3 samples in each subset, and 698 the initial set of these examples requires the update of the 10^3 samples in each run. The classic gradient 699 evaluation takes 226.8 ms, while the neural network requires only 10.9 ms. Thus, the neural network approach 700 is approximately 20 times faster for gradient evaluations. The evaluation of a whole Subset Simulation takes 701 15 872 sec on average for HMC and 735 sec on average using $H_{NN}MC$. Increasing the number of samples 702 for the new strategy achieves better accuracy than using the HMC. Even though sampling is still faster – 703 using 10 times more samples, 10^4 instead of 10^3 takes 5 393 sec on average – the evaluation of the limit state 704 function is in general the most expensive part of the simulation. 705

⁷⁰⁶ Appendix B. Limitations for Rosenbrock distribution with k = 100

This part of the appendix focuses on the proposed strategies to deal with the limitations of the approach 707 based on the Rosenbrock distribution with k = 100. First, we observe that using the same training procedure 708 as in previous cases (4×10^5 initial gradient evaluations) results in very few samples in the tails of this very 709 thin distribution. As a result, the HNN is likely to produce poor gradient predictions in this area – which 710 is critical for reliability analysis. Hence, we use long trajectories (800 leapfrog steps with a step size of 711 (0.05) during the creation of training data assembled using 10^3 initial samples without using an acceptance-712 rejection criterion. Figure Appendix B.1(a) shows the results of one Subset Simulation using $H_{NN}MC$. The 713 pre-trained HNN on 8×10^5 gradient evaluations gives accurate predictions within the first three sets; the 714 acceptance rate is above 90%. However, the rate drops rapidly due to the poor training in the tails (e.g., 715 the acceptance rate is 0.05 in subset 8 and does not exceed 0.25 in any subset after the third). Hence, the 716 samples reach the failure region only after 15 subsets, which significantly underestimates the probability of 717 failure. However, these observations represent only one trial. For further analysis, we used 10 independently 718 trained architectures to create 100 Subset Simulations, i.e., ten trials with each trained architecture. The 719

results of these runs are summarized in Table B.2. The standard HNN approach gives a mean beta value of $\bar{\beta} = 8.014$, which is not accurate relative the reference value $\beta = 2.715$ [39].



Figure Appendix B.1: Subset results for the Rosenbrock distribution k = 100 using Hamiltonian Neural Network sampling (a) with a Hamiltonian Neural Network, (b) with a latent Hamiltonian Neural Network, (c) with a Hamiltonian Neural Network and retraining in each set, (d) with a Hamiltonian Neural Network and online error monitoring, (e) Hamiltonian Monte Carlo sampling, and (f) Hamiltonian Neural Network with 10 000 samples.

| Approach | \bar{eta} | $\operatorname{Var}(\beta)$ | $CV(\beta)$ |
|--------------------------------------|-------------|-----------------------------|-------------|
| Standard HNN | 8.014 | 3.985 | 0.249 |
| Latent HNN | 6.519 | 0.841 | 0.140 |
| HNN with retraining | 5.615 | 1.658 | 0.229 |
| HNN with online error | 4.983 | 0.885 | 0.189 |
| HMC (10 runs) | 2.997 | 0.0932 | 0.032 |
| HNN (10 runs, 10^4 samples/subset) | 2.781 | 0.0313 | 0.064 |

Table B.2: Subset Simulation results for different $H_{NN}MC$ approaches. Summary from 100 trials using ten different HNN architectures and ten repeated trials for each approach.

To improve performance, we explore three strategies:

1. Using an enhanced HNN architecture, i.e., the latent variable HNN [47].

Updating the HNN parameters in each subset evaluation to achieve higher acceptance rates cf. Section 4.2.2.

3. Using an online error monitoring scheme to ensure good proposals for the new states, cf. Section 4.2.1.

⁷²⁷ All three approaches improve the estimation of the probability of failure, as shown in Figure Appendix ⁷²⁸ B.1(b)-(d) and Table B.2.

First, the enhancement of the architecture from HNN to latent HNN improves the prediction accuracy of the network [47]. Even though the latent HNN provides better gradient estimates, the results vary significantly with each simulation. The gradients are often overestimated in regions that are not trained well, as can be observed from Figure Appendix B.1(b). In the lower region around (0,0), we observe outliers stuck outside the high probability area. One interesting observation is the vast area covered by the second subset in orange. This approach slightly improves the mean value of beta but significantly reduces the variance (Table B.2).

The results can be further improved if more training data is used, particularly in the tails of the dis-736 tribution. This leads us to the second approach: retraining the HNN after each subset. The results for a 737 single Subset Simulation are shown in Figure Appendix B.1(c). Although the probability of failure esti-738 mate is a little lower compared to the latent HNN prediction in this particular illustration, we observe that 739 the outcome is better on average. Also, we observe higher acceptance rates within the subsets, gradually 740 evolving toward the failure region. Table B.2 shows that the mean beta value decreases to $\beta = 5.615$, which 741 (while not particularly accurate) is comparable to estimates from other state-of-the-art samplers, such as 742 the stretch sampling method [39]. However, this approach requires the evaluation of samples using the tra-743 ditional approach within each subset. Thus, the computational cost increases with the number of training 744 samples chosen for each subset. 745

The best results of the proposed approaches are achieved by using online error monitoring. Here, the 746 trajectory is corrected using traditional gradient calculations if a threshold is exceeded. Therefore, we 747 observe that no samples get stuck outside of the distribution, see Figure Appendix B.1(d). However, using 748 this scheme diminishes the computational savings of the approach as it requires expensive computational 749 gradient calculations when Hamiltonian conservation is violated. From Table B.2, the mean beta value is 750 $\beta = 4.983$, which is an improvement over other methods (except traditional HMC) for only 1000 samples 751 per subset [39]. Furthermore, the variance of the approach is also reduced. This approach delivers the most 752 reliable results for very complex distribution and low-probability regions. However, the threshold has to be 753 chosen carefully. If chosen too large, the proposals may still be inaccurate. If chosen too small, for most of 754 the samples, numerical gradient calculations will be used such that the approach ends up in the traditional 755 HMC. 756

⁷⁵⁷ Meanwhile, the traditional HMC approach produces accurate results with an average of $\beta = 2.997$ and ⁷⁵⁸ a coefficient of variation of $CV(\beta) = 0.032$ for ten Subset Simulations. Therefore, while the proposed im-⁷⁵⁹ provements show better performance for this very difficult problem, they are still not capable of reproducing ⁷⁶⁰ the accuracy of the full HMC approach. Another possibility for increasing the accuracy of Subset Simulations is to evaluate more samples in the subsets. Using 10⁴ samples in $H_{NN}MC$ results in a crucial improvement of the estimate. From 10 runs, the $H_{NN}MC$ results in an average estimate of $\beta = 2.781$ with a coefficient of variation of $CV(\beta) = 0.063$. The $H_{NN}MC$ with increased subset samples outperforms the traditional method regarding speed and accuracy. However, if the limit state functions become computationally more expensive, the sample increase will lead to significantly higher costs. Thus, to keep a balance between sampling efficiency and the number of samples required online error monitoring is likely to be preferred.

⁷⁶⁸ Appendix C. Spring damper system

⁷⁶⁹ In this example, the two-degree-of-freedom (dof) spring-mass-damper system shown in Figure Appendix

 T_{10} C.1 is considered. The primary-secondary system is characterized by the masses m_p and m_s , spring stiffnesses

 k_p and k_s , and damping coefficients c_p and c_s , where the subscripts p and s refer to the primary and secondary oscillators, respectively. These six uncertain parameters follow independent log-normal distributions with



Figure Appendix C.1: A two-dof system with uncertain parameters, cf [14].

772

mean values and coefficients of variation provided in Table C.3.

F

Table C.3: Mean values and coefficient of variation of the parameters of the two degrees of freedom system. All random variables of this example follow log-normal distributions.

| Parameter | Mean | CV |
|-----------|-----------------------------------|-----|
| m_p | $1.5\mathrm{kg}$ | 0.1 |
| m_s | $0.01\mathrm{kg}$ | 0.1 |
| k_p | $1.\frac{\mathrm{N}}{\mathrm{m}}$ | 0.2 |
| k_s | $0.01 \frac{\text{N}}{\text{m}}$ | 0.2 |
| ζ_p | 0.05 | 0.4 |
| ζ_s | 0.02 | 0.5 |
| F_s | $15\mathrm{N}$ | 0.1 |
| S_0 | $100\mathrm{N}$ | 0.1 |
| | | |

773

The reliability of the system is based on the spring capacity F_s of the secondary oscillator, where the highly nonlinear limit state function is given by [58]:

$$g(\mathbf{x}) = F_s - 3k_s \sqrt{\frac{\pi S_0}{4\zeta_s \omega_s^3}} \left[\frac{\zeta_a \zeta_s}{\zeta_p \zeta_s \left(4\zeta_a^2 + \theta^2\right) + \gamma \zeta_a^2} \frac{\left(\zeta_p \omega_p^3 + \zeta_s \omega_s^3\right) \omega_p}{4\zeta_a \omega_a^4} \right],$$
(C.1)

where $\omega_p = \sqrt{k_p/m_p}$ and $\omega_s = \sqrt{k_s/m_s}$ denote the natural frequencies and $\gamma = m_s/m_p$ the mass ratio. Furthermore, $\omega_a = (\omega_p + \omega_s)/2$ and $\zeta_a = (\zeta_p + \zeta_s)/2$ denote the average values for the eigenfrequency and damping parameters, and the factor Θ is defined as $\Theta = \omega_p - \omega_s/\omega_a$. The probability of failure is $P_F = 4.79 \times 10^{-3}$ ($\beta = 2.59$) determined from Monte Carlo simulation [14, 58].

We performed 100 Subset Simulations using the proposed $H_{NN}MC$ sampling. The resulting mean reliability index is $\bar{\beta} = 2.680$, and the coefficient of variation is $CV(\beta) = 0.0433$, which is very close to the correct value from MCS. We noticed that the acceptance rate of the subset drops below 0.3, which would lead to an update of the leapfrog steps. The acceptance rate of the HNN proposals remains above 0.9, indicating that the previously discussed online error monitoring and/or retraining are unnecessary. The traditional HMC sampling for ten Subset Simulations results in a similar estimate with mean reliability index $\bar{\beta} = 2.663$ and the coefficient of variation $CV(\beta) = 0.025$.

787 Appendix D. White noise excitation

This example considers a high-dimensional problem with 200 variables. A single-degree-of-freedom oscillator, shown in Figure Appendix D.1(a), is excited with a seismic loading [40]. The properties of the oscillator are fixed with mass $m = 6 \times 10^4$ kg, stiffness $k = 2 \times 10^7$ N/m and damping $c = 2m\zeta\sqrt{k/m}$ using a viscous damping ratio of $\zeta = 0.1$. The equation of motion is written as:

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = -m\ddot{u}_g(t) .$$
 (D.1)

Here, u(t) is the displacement of the oscillator, and \dot{u} and \ddot{u} are the velocity and acceleration of the mass, respectively. The randomness of this example is introduced by the ground acceleration \ddot{u}_g . The acceleration is simulated by a white noise process discretized in the frequency domain as [59]:

$$\ddot{u}_g = A \sum_{j=1}^{n/2} \left(x_j \cos \omega_j t + x_{j+n/2} \sin \omega_j t \right) , \qquad (D.2)$$

where $A = \sqrt{2S\Delta\omega}$ to account for the intensity of the white noise, chosen as $S = 0.01 \text{m}^2/\text{s}^3$. Figure Appendix D.1(b) shows one white noise sample.

The random vector **q** consists of n = 200 independent standard Gaussian random variables. The frequency points are given by $\omega_j = j\Delta\omega$ using n/2 = 100 points with a cut-off frequency of $\omega_{cut} = 15\pi$, which leads to $\Delta\omega = 0.15\pi$.

We use the same first-passage probability problems in the following, as demonstrated for the standard Hamiltonian Monte Carlo Subset Simulation [40]. The limit state function of the problem is written as:

$$g(\mathbf{x}) = u_{lim} - \max_{t \in [0,10]} u(\mathbf{x}, t) .$$
(D.3)

The results of 100 Subset Simulations for $u_{lim} = 0.02$ m, $u_{lim} = 0.025$ m and $u_{lim} = 0.03$ m in terms of the reliability index β are shown in Table D.4. In the last column, the results from the original proposal of Hamiltonian Monte Carlo for Subset Simulations are shown to compare the new method.

Table D.4: Subset Simulation results of 100 runs of the Hamiltonian Neural Network approach for the white noise excitation subjected to a single-degree-of-freedom system.

| Threshold | $ar{eta}$ | $\operatorname{Var}(\beta)$ | $CV(\beta)$ | HMC $[40]$ |
|--------------------|-----------|-----------------------------|-------------|------------|
| $0.020 \mathrm{m}$ | 2.355 | 0.0074 | 0.0365 | 2.471 |
| $0.025 \mathrm{m}$ | 3.802 | 0.0661 | 0.0676 | 3.786 |
| $0.030\mathrm{m}$ | 5.165 | 0.0775 | 0.0539 | 4.997 |



Figure Appendix D.1: White noise excitation problem; (a) single-degree-of-freedom oscillator; (b) exemplary created white noise; (c) examplary response of the oscillator.