



**MACHINE LEARNING APPLICATION  
TO THE SIMULATION AND ANALYSIS  
OF LATTICE FIELD THEORIES**

**BY  
CHARALAMBOS TSIOUTIS**

**MASTER OF SCIENCE IN SIMULATION  
AND DATA SCIENCE**

**2023**

# Abstract

The core objective of this research was to explore the application of a flow-based Monte Carlo algorithm, specifically RealNVP, to mitigate autocorrelation issues that are present in state-of-the-art algorithms such as the Metropolis and Hybrid Monte Carlo. Based on the path integral formulation, we initially implemented Metropolis-Hastings and Hybrid Monte Carlo methods for both harmonic and anharmonic oscillators, establishing a baseline for comparison with the RealNVP algorithm.

In our study we showcased that RealNVP can produce ensembles that accurately represent the theory, offering significant improvements in terms of autocorrelation times and accuracy. However the application of this method to more quantitative questions such as how the method scales with the physical volume, higher dimensions, and dependence on the model parameters such as the magnitude of the anharmonic potential requires further investigation.



## VALIDATION PAGE

**Master of Science Candidate:** Charalambos Tsioutis

**Title of the thesis:** Machine learning application to the simulation and analysis of lattice field theories.

*This thesis was submitted towards the fulfillment of the requirements for the degree of Master of Science at the Graduate School of The Cyprus Institute and was validated on November 16th, 2023 by the Academic Committee of The Cyprus Institute.*

**Examination Committee:**

Associate Professor Giannis Koutsou

Assistant Professor Mihalīs Nicolaou

Simone Bacchio

## DECLARATION

This thesis was submitted towards the fulfillment of the requirements for the award of a Master of Science Degree from The Cyprus Institute. It is the product of my own original work, unless otherwise mentioned through references, notes or other statements.

SIGNATURE

---

CHARALAMBOS TSIOUTIS

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Dr. Giannis Koutsou, for his continual support and guidance throughout the development of this research project.

His expertise and discerning feedback have been invaluable, guiding me towards the successful completion of this thesis.

# Contents

List of Figures	ii
List of Tables	iv
Introduction	v
1 Path Integral Formulation	1
1.1 The Quantum Harmonic oscillator . . . . .	3
1.2 The Quantum Anharmonic Oscillator . . . . .	4
2 Monte Carlo Methods	5
2.1 Metropolis-Hastings algorithm . . . . .	6
2.2 Hybrid Monte Carlo . . . . .	6
2.3 Practical considerations in Monte Carlo methods . . . . .	7
2.3.1 Thermalization . . . . .	7
2.3.2 Autocorrelations . . . . .	8
2.3.3 Jackknife Error Analysis . . . . .	9
3 Flow-based Generative Models	9
3.1 RealNVP Model . . . . .	10
4 Results	12
4.1 Computational Setup . . . . .	12
4.2 Metropolis-Hastings Monte Carlo Method . . . . .	13
4.3 Hybrid Monte Carlo Method . . . . .	14
4.4 Flow-based Monte Carlo . . . . .	17
4.5 Model Comparison . . . . .	21
5 Conclusions	25
Bibliography	28

# List of Figures

1	Thermalization process example from the Metropolis-Hastings algorithm simulation. The simulation is for a lattice size $L = 30$ and acceptance rate 80%. The initial configuration is a vector of size $L$ with zero values. The plotted data are the simulation results for each sweep for the observable $\langle x^2 \rangle$ . . . . .	8
2	Ground state energy continuum limit calculation for a) Harmonic and b) Anharmonic Oscillator ( $\lambda = 1$ ) using the Metropolis-Hastings algorithm. The filled circles represent the values calculated from the Metropolis-Hastings algorithm with their corresponding errors. The dashed line represents a linear fit of the data, and the shaded part represents the fitting uncertainty. . . . .	14
3	Ground state energy continuum limit calculation for a) Harmonic Oscillator and b) Anharmonic Oscillator ( $\lambda = 1$ ) using the Hybrid Monte Carlo algorithm. The filled circles represent the values calculated from the Hybrid Monte Carlo algorithm with their corresponding errors. The dashed lines in a) represent linear and constant fit of the data, in b) represent linear fit. The shaded part represents the fitting uncertainty. . . . .	17
4	Loss versus epochs (training steps) for the RealNVP model of the Quantum a) Harmonic and b) Anharmonic Oscillator. We used the same learning rate and architecture as well as the same number of epochs. Different colors represent different simulation lattice sizes as indicated in the legend. . . . .	18
5	Acceptance rate versus epochs (training iterations) for the RealNVP model. Different line colors represent simulations for different lattice sizes $L$ . For better representation we used the moving average in our data with window 10. . . . .	19
6	Ground state energy continuum limit calculation for a) Harmonic Oscillator and b) Anharmonic Oscillator using the RealNVP algorithm. The filled circles represent the values calculated from the RealNVP algorithm with their corresponding errors. The black dashed line represents a linear fit of the data, the grey dashed line the constant fit and the shaded part represents the fitting uncertainty. The errors that do not appear in the graph are smaller than the data circles. . . . .	21
7	$\langle x^2 \rangle$ distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	22
8	$\langle x^2 \rangle$ distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	22



9	$\langle x^4 \rangle$ distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	22
10	$\langle x^4 \rangle$ distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	23
11	The action distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	23
12	The action distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes $L = 30, 60, 100$ . . . . .	23
13	Autocorrelation times $t_{int}$ for the three methods used (Metropolis-Hastings, Hybrid Monte Carlo, and RealNVP) relative to lattice spacing $\delta\tau$ . a) Acceptance rates: Metropolis-Hastings $0.79 - 0.81$ , $0.72 - 0.85$ , RealNVP: $0.20 - 0.25$ . b) Acceptance rates: Metropolis-Hastings: $0.80 - 0.82$ , Hybrid Monte Carlo: $0.73 - 0.88$ , RealNVP: $0.20 - 0.25$ . The $t_{int}$ axis is logarithmic for presentation purposes. . . . .	24
14	The dots represent the estimate value of ground state energy for the three simulations from their best fit with their corresponding errors. Inner bar: Statistical error, Outer error bar: Statistical and Systematic error. Best fit for Harmonic Oscillator: i) Metropolis: Linear fit, 7 data points ii) HMC: Quadratic fit, 4 data points iii) RealNVP: Quadratic fit, 6 data points. Best fit for Anharmonic Oscillator: i) Metropolis: Linear fit, 6 data points ii) HMC: Linear fit, 3 data points iii) RealNVP: Quadratic fit, 8 data points. Theoretical value is 0.5 and the approximate values is 0.8038 [1], [2]. . . .	25

## List of Tables

1	Summary of the symbols and terminology used throughout the simulation procedure. . . . .	12
2	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the Metropolis-Hastings Quantum Harmonic Oscillator, alongside the autocorrelation times ( $t_{int}$ ). . . . .	15
3	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the Metropolis-Hastings Anharmonic Oscillator, along with autocorrelation times $t_{int}$ . . . . .	15
4	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the Hybrid Monte Carlo Harmonic Oscillator algorithm, along with autocorrelation times $t_{int}$ . . . . .	16
5	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the Hybrid Monte Carlo Anharmonic Oscillator, along with autocorrelation times $t_{int}$ . . . . .	17
6	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the RealNVP Monte Carlo model for the Harmonic Oscillator, along with autocorrelation times $t_{int}$ . . . . .	20
7	Ground state energy $E_0$ as a function of lattice spacing $\delta\tau$ with associated errors for the RealNVP Anharmonic Oscillator, along with autocorrelation times $t_{int}$ . . . . .	20

# Introduction

The path integral formulation of quantum mechanics has fundamentally altered our understanding of quantum systems. Pioneered by Richard Feynman in 1948 [3], this formulation interprets quantum mechanics through the lens of a classical action, drawing a striking bridge between the classical and quantum realms [4]. Initially, path integrals were more of a theoretical marvel, offering an elegant yet seemingly impractical approach to quantum mechanics due to the complexity of evaluating a continuum of all possible paths [5].

However, the advent of computational methods significantly altered this landscape. With the introduction of computers into the scientific realm, it became feasible to practically evaluate path integrals. Specifically, the Markov Chain Monte Carlo (MCMC) methodology [6] allowed for numerical approximations of quantum systems, revolutionizing fields such as lattice field theory [7], [8]. Despite its groundbreaking applicability, Monte Carlo simulations come with their own set of challenges, such as autocorrelation between configurations within the same chain and critical slowing down.

To address these limitations inherent in traditional Monte Carlo methods, advanced techniques and models have been developed to mitigate issues like critical slowing down. In this project, we apply the state-of-the-art Hybrid Monte Carlo (HMC) algorithm [9]. This model employs both the deterministic approach, a molecular dynamics integration, and the stochastic approach, Monte Carlo chain, offering a comprehensive mechanism to improve the efficiency and correlations of the simulations.

In the wake of recent advancements in machine learning, a surge of research has been dedicated to the application of flow-based MCMC approaches to physical quantum systems, including lattice field systems [10]. Utilizing models like RealNVP [11] (real valued non-volume preserving (RealNVP) transformations for density estimation), these machine learning algorithms can produce lattice field configurations with potentially reduced autocorrelation through improved sampling techniques. Our research provides a presentation of these flow-based machine learning models, critically discussing their limitations and possible solutions, especially in the context of the quantum Anharmonic oscillators.

It is clear, through our research, that the practical application of these advanced techniques is rooted in a deep understanding of the path integral formulation. Therefore, the ensuing chapter will delve into the theoretical underpinnings of path integrals, setting the foundational groundwork for the computational explorations that follow.

In subsequent chapters, we present the theoretical background on Harmonic and Anharmonic oscillators, involving the formulation of Hamiltonian and Lagrangian expressions for both systems and their connection to the computational process. Following this, we detail the Monte Carlo computational methods employed in this research,

describing the methodology behind path integral sampling and the approximation of observables in Harmonic and Anharmonic quantum oscillators.

Next, we present the results derived through the application of these methodologies, discussing not only the insights gained but also the limitations in terms of accuracy and performance of the three methodologies used following the concluding chapter, in which we summarize the key findings of our research.

# 1 Path Integral Formulation

A fundamental question in quantum physics revolves around the evolution of the state of a particle. To derive the path integral formulation, we start with the initial-value problem of the Schrödinger equation:

$$i\hbar \frac{\partial \psi(x, t)}{\partial t} = \hat{H} \psi(x, t) \quad (1)$$

where  $\hat{H}$  is the Hamiltonian operator,  $\psi(x, t)$  is the wave function of the system and  $\hbar$  is the reduced Planck's constant given by  $\hbar = \frac{h}{2\pi}$ , where  $h$  is Planck's constant.. The general solution of the equation using the *evolution operator*  $e^{-i\hat{H}t/\hbar}$  is:

$$\psi(x, t) = e^{-i\hat{H}t/\hbar} \psi(x, 0). \quad (2)$$

In Dirac's bra-ket notation [12], the matrix elements of the evolution operator between any two initial and final positions are:

$$\langle x_f | e^{-i\hat{H}(t_f - t_i)/\hbar} | x_i \rangle = \langle x_f, t_f | x_i, t_i \rangle. \quad (3)$$

The matrix elements of this equation describe the evolution of a system with a Hamiltonian  $\hat{H}$ . The evolution of the wave function is given by:

$$\psi(x_f, t_f) = \langle x_f, t_f | \psi \rangle \quad (4)$$

$$= \int dx_i \langle x_f, t_f | x_i, t_i \rangle \langle x_i, t_i | \psi \rangle \quad (5)$$

$$= \int dx_i \langle x_f, t_f | x_i, t_i \rangle \psi(x_i, t_i). \quad (6)$$

For the general Hamiltonian of a particle:

$$\hat{H} = \frac{\hat{p}^2}{2m} + V(\hat{x}), \quad (7)$$

where  $m$  the mass of the particle,  $V$  the potential and  $\hat{p}, \hat{x}$  the momentum and position operators, the evolution of the system is evaluated as:

$$\langle x_f, t_i + \delta t | x_i, t_i \rangle = \langle x_f | e^{-i\hat{H}(\delta t)/\hbar} | x_i \rangle \quad (8)$$

$$= \int dp \langle x_f | p \rangle \langle p | e^{-i\hat{H}(\delta t)/\hbar} | x_i \rangle. \quad (9)$$

For small time step  $\delta t$  we can expand the exponential to first order:

$$\langle p|e^{-i\hat{H}\delta t/\hbar}|x_i\rangle = \langle p|1 - \frac{i\hat{H}\delta t}{\hbar} + \mathcal{O}(\delta t)^2|x_i\rangle. \quad (10)$$

The matrix elements of the operators are evaluated as:

$$\langle p|1|x_i\rangle = \langle p|x_i\rangle \quad (11)$$

$$\langle p|\hat{p}^2|x_i\rangle = \langle p|\hat{p}^2|p\rangle \langle p|x_i\rangle = p^2 \langle p|x_i\rangle \quad (12)$$

$$\langle p|V(\hat{x})|x_i\rangle = \langle p|x_i\rangle \langle x|V(\hat{x})|x_i\rangle = V(x_i) \langle p|x_i\rangle. \quad (13)$$

We can write now the evolution Equation (10) as an approximation:

$$\langle p|e^{-i\hat{H}\delta t/\hbar}|x_i\rangle \approx \left(1 - \frac{ip^2\delta t}{2m\hbar} - \frac{i\delta t}{\hbar}V(x_i)\right) \langle p|x_i\rangle \quad (14)$$

$$\approx \exp\left(-\frac{i}{\hbar}\left(\frac{p^2\delta t}{2m} + V(x_i)\delta t\right)\right) \langle p|x_i\rangle. \quad (15)$$

We can use the Fourier tranformation of the momentum representation:

$$\langle p|x_i\rangle = \frac{e^{-ipx_i/\hbar}}{\sqrt{2\pi\hbar}}, \quad (16)$$

and by replacing in Equation (10) we obtain:

$$\langle p|e^{-i\hat{H}\delta t/\hbar}|x_i\rangle = \frac{1}{\sqrt{2\pi\hbar}} \exp\left(-\frac{i}{\hbar}\left(px_i + \frac{p^2\delta t}{2m} + V(x_i)\delta t\right)\right). \quad (17)$$

Using the equations (8) and (14) we get:

$$\langle x_f, t_i + \delta t|x_i, t_i\rangle = \sqrt{\frac{m}{2\pi i\hbar\delta t}} \exp\left(\frac{i}{\hbar}\left(\frac{m(x_f - x_i)^2}{2\delta t} - V(x_i)\delta t\right)\right). \quad (18)$$

The exponential term of the Equation (18) is the classical Lagrangian  $L\delta t$ , and the short term evolution can be reduced to:

$$\langle x_f, t_i + \delta t|x_i, t_i\rangle = \sqrt{\frac{m}{2\pi i\hbar\delta t}} e^{iL\delta t/\hbar}. \quad (19)$$

To evaluate the evolution of the system for any initial and final positions, we divide the time interval into time slices of duration  $\delta t$ . Replacing our findings into Equation (3) we get:

$$\begin{aligned} \langle x_f, t_f|x_i, t_i\rangle &= \int \cdots \int \langle x_f, t_f|x_{N-1}, t_{N-1}\rangle \langle x_{N-1}, t_{N-1}|x_{N-2}, t_{N-2}\rangle \\ &\quad \cdots \langle x_2, t_2|x_1, t_1\rangle \langle x_1, t_1|x_i, t_i\rangle dx_1 dx_2 \cdots dx_{N-1}. \end{aligned} \quad (20)$$

For each time slice we can apply the Equation (18) as:

$$\langle x_f, t_f | x_i, t_i \rangle = \int \prod_{n=1}^{N-1} dx_n \exp \left( \frac{i\delta t}{\hbar} \sum_{n=1}^{N-1} L(t_n) \right). \quad (21)$$

As  $N$  approaches  $\infty$  and  $\delta t$  nears zero, while keeping their product  $N\delta t$  constant, the integral across discrete time points converges to an integral over all possible trajectories between the starting and ending points. This is represented by:

$$\langle x_f, t_f | x_i, t_i \rangle = \int \mathcal{D}x(t) e^{-iS/\hbar}. \quad (22)$$

Here,  $\mathcal{D}x(t) = \prod_{n=1}^{N-1} dx_i$ . Moreover, as  $N$  tends toward  $\infty$ , the path action  $S$  simplifies to:

$$S = \int_{t_i}^{t_f} L(x(t)) dt = \int_{t_i}^{t_f} \left( \frac{m}{2} \left( \frac{dx}{dt} \right)^2 - V(x(t)) \right) dt. \quad (23)$$

To ensure that the associated path quantities are real and restrict overly large contributions we introduce an imaginary time transformation  $t \rightarrow -i\tau$  where  $\tau$  is real. Now, equations (22) and (23) become:

$$\langle x_f, \tau_f | x_i, \tau_i \rangle = \int \mathcal{D}x(\tau) e^{-S/\hbar} \quad (24)$$

$$S = \int_{\tau_i}^{\tau_f} L(x(\tau)) d\tau = \int_{\tau_i}^{\tau_f} \left[ \frac{m}{2} \left( \frac{dx}{d\tau} \right)^2 + V(x(\tau)) \right] d\tau. \quad (25)$$

## 1.1 The Quantum Harmonic oscillator

For a particle with mass  $m$  in a Harmonic oscillator potential defined by force constant  $k$ , we have the following Hamiltonian:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2, \quad (26)$$

where  $\omega = \sqrt{\frac{k}{m}}$  represents the natural frequency of the system. The Lagrangian of the system is:

$$L_i = \frac{m}{2} \left( \frac{x_{i+1} - x_i}{\delta\tau} \right)^2 + \frac{m\omega^2 x_i^2}{2}. \quad (27)$$

Based on the Lagrangian, we can formulate the Euclidean action:

$$S_e = \sum_{n=1}^{N-1} \left[ \frac{m}{2} \left( \frac{x_{i+1} - x_i}{\delta\tau} \right)^2 + \frac{m\omega^2 x_i^2}{2} \right]. \quad (28)$$

The Hamiltonian operator  $\hat{H}$  will have energy eigenvalues  $E_n$  when it operates on the corresponding wave function  $\psi_n(x)$ , such that:

$$\hat{H}\psi_n(x) = E_n\psi_n(x). \quad (29)$$

The quantum Harmonic oscillator has analytical solution that can be calculated through the energy eigenvalues:

$$E_n = \hbar\omega \left( n + \frac{1}{2} \right), \quad n = 0, 1, 2, \dots \quad (30)$$

Here,  $n$  is the quantum number corresponding to the energy level, and  $\hbar$  is the reduced Planck constant.

By applying the Virial theorem [13] in the Hamiltonian (26) we can express the energy of the ground state  $E_0$  in terms of correlation functions of the position  $x$ . The Virial theorem holds for the quantum Harmonic oscillator since the potential  $V(x) = \frac{1}{2}m\omega^2x^2$  is a central force potential, meaning it only depends on the position  $x$ . Initially we get from the Virial theorem that,

$$\langle T \rangle = \frac{1}{2} \left\langle x \frac{dV}{dx} \right\rangle, \quad (31)$$

where  $\langle T \rangle$  is the expectation value of the kinetic energy, and  $\langle x \frac{dV}{dx} \rangle$  is the expectation value for  $x \frac{dV}{dx}$ . For the potential  $V(x) = \frac{1}{2}m\omega^2x^2$ , we obtain:

$$\langle T \rangle = \frac{1}{2} \langle x (m\omega^2x) \rangle, \quad (32)$$

The ground state energy  $E_0$  can then be expressed as:

$$E_0 = m\omega^2 \langle x^2 \rangle \quad (33)$$

where  $\langle x \rangle$  is the expectation value of the position.

## 1.2 The Quantum Anharmonic Oscillator

While the Harmonic oscillator is a good check of our methodology, it is a well-known system which can also be solved analytically. We add a quartic term to the potential of our system, which yields the Anharmonic Oscillator. This system is inherently intriguing as, unlike the Harmonic oscillator, the energy eigenvalues of the quantum Anharmonic Oscillator cannot be determined through analytical solutions. The potential of the system is given by:

$$V(x) = \frac{1}{2}m\omega^2x^2 + \lambda x^4. \quad (34)$$



In this case, the Hamiltonian for the system becomes:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2 + \lambda\hat{x}^4. \quad (35)$$

The corresponding Lagrangian can be expressed as:

$$L_i = \frac{m}{2} \left( \frac{x_{i+1} - x_i}{\delta\tau} \right)^2 - \frac{1}{2}m\omega^2 x_i^2 - \lambda x_i^4. \quad (36)$$

And, the Euclidean action for the anharmonic oscillator becomes:

$$S_e = \sum_{n=1}^{N-1} \left[ \frac{m}{2} \left( \frac{x_{i+1} - x_i}{\delta\tau} \right)^2 + \frac{1}{2}m\omega^2 x_i^2 + \lambda x_i^4 \right]. \quad (37)$$

As in the Harmonic oscillator, the Anharmonic Oscillator have a central force potential thus we can apply the Virial theorem in the Hamiltonian (35) to express the energy of the ground state  $E_0$  in terms of correlation functions of the position  $x$ . The ground state energy  $E_0$  can then be expressed as:

$$\begin{aligned} E_0 &= \langle T \rangle + \langle V \rangle \\ &= m\omega^3 \langle x^2 \rangle + 3\lambda \langle x^4 \rangle. \end{aligned} \quad (38)$$

where  $\langle x \rangle$  is the expectation value of the position.

## 2 Monte Carlo Methods

Having discussed the theoretical background and formulated the Hamiltonian and Lagrangian for both the Harmonic and Anharmonic quantum oscillators, we now turn our attention to the numerical methods used for simulation and specifically Monte Carlo approaches. These stochastic techniques allow us to sample the path integral and thus approximate observables in both Harmonic and Anharmonic Quantum Oscillators.

The simulations are performed on a discrete time lattice with  $N_\tau$  time slices, imposing periodic boundary conditions. To acquire statistics for the observables, multiple particle trajectories  $x_1, \dots, x_N$  are generated, with each coordinate being a real number.

For computational purposes, we introduce dimensionless variables. Setting  $\hbar = 1 = c$  all variables are expressed in terms of the lattice spacing  $\delta\tau$  [14]:

$$\tilde{m} = m\delta\tau, \quad \tilde{\omega} = \omega\delta\tau, \quad \tilde{x}_i = \frac{x_i}{\delta\tau}. \quad (39)$$

This way, the dimensionless actions for the Harmonic and Anharmonic quantum

oscillators are given for the Harmonic oscillator:

$$\tilde{S}_{\text{Harmonic}} = \sum_{i=1}^{N_\tau} \left[ \frac{1}{2} \tilde{m} (\tilde{x}_{i+1} - \tilde{x}_i)^2 + \frac{1}{2} \tilde{m} \tilde{\omega}^2 \tilde{x}_i^2 \right], \quad (40)$$

and for the Anharmonic Oscillator:

$$\tilde{S}_{\text{Anharmonic}} = \sum_{i=1}^{N_\tau} \left[ \frac{1}{2} \tilde{m} (\tilde{x}_{i+1} - \tilde{x}_i)^2 + \frac{1}{2} \tilde{m} \tilde{\omega}^2 \tilde{x}_i^2 + \tilde{\lambda} \tilde{m}^2 \tilde{\omega}^3 \tilde{x}_i^4 \right], \quad (41)$$

where  $\tilde{\lambda} = \lambda \delta \tau^4$ .

## 2.1 Metropolis-Hastings algorithm

In the Metropolis-Hastings algorithm, our goal is to generate paths by updating a single site at a time based on the Metropolis-Hastings algorithm [6]. The input of the algorithm is an array with  $N_\tau$  sites and the parameters:  $\tilde{m}, \tilde{\omega}, \tilde{\lambda}$  and real number  $h$ . One sweep visits  $N_\tau$  sites in random order. For a given site  $i$  each Metropolis update will implement the following steps:

1. Generate a random number  $u$  from a uniform distribution in the interval  $[-h, h]$ .
2. Proposed new position:  $\tilde{x}_i \rightarrow \tilde{x}'_i = \tilde{x}_i + u$ .
3. Calculate the difference in the action  $\delta \tilde{S}$  between the new and old value.
4. The acceptance probability of the new site will be  $\min(1, e^{-\delta \tilde{S}})$ .

The probability  $\min(1, e^{-\delta \tilde{S}})$  ensures that the proposals that decrease the action are always accepted. A trial that would increase the action is accepted with probability  $e^{-\delta \tilde{S}}$  [15]. For each sweep we compute the acceptance ratio, and by adjusting the value of  $h$  we can tune the acceptance ratio.

## 2.2 Hybrid Monte Carlo

An alternative method to Metropolis-Hastings is the Hybrid Monte Carlo (HMC) algorithm [9]. This algorithm, rather than updating one site at a time, involves parallel updates of fields at all lattice sites with the acceptance check performed for the whole configuration.

The implementation of the algorithm requires the introduction of 'computer' time parameter  $\tau$ , and a Hamiltonian in order to compute the evolution of the vector  $\vec{x}(\tau)$ , in this time. Namely, we add conjugate momenta  $\pi(\tau)$  and define the Hamiltonian:

$$H(\vec{x}, \vec{\pi}) \equiv \frac{1}{2} \vec{\pi}^2 + S(\vec{x}) \quad (42)$$

where  $S(\vec{x})$  is the action of the system. From the Hamiltonian we get the equations of motion:

$$\dot{\vec{x}} = \frac{\delta H}{\delta \vec{\pi}} = \vec{\pi} \quad (43)$$

$$\dot{\vec{\pi}} = -\frac{\delta H}{\delta \vec{x}} = -\frac{\delta S}{\delta \vec{x}} \quad (44)$$

To create a new configuration for  $\vec{x}$ , we select randomly an initial momenta  $\vec{\pi}$  from a gaussian  $\sim N(0, 1)$  and then we evolve the system deterministically using the equations of motion. To achieve that we integrate the equations of motion by discretizing using a step  $\delta\tau$  up until the total integration time  $\tau_0$ . We use the leap frog integration that can be described as :

- Initial half step:

$$\vec{\pi}\left(\frac{\delta\tau}{2}\right) = \vec{\pi}(0) - \left[ \frac{\delta S(0)}{\delta \vec{x}} \right] \frac{\delta\tau}{2}$$

- Perform  $n = \frac{\tau_0}{\delta\tau}$  steps in  $\vec{x}$  and  $n - 1$  steps in  $\vec{\pi}$  of the form:

$$\vec{x}(\tau + \delta\tau) = \vec{x}(\tau) + \vec{\pi}\left(\tau + \frac{\delta\tau}{2}\right) \delta\tau$$

$$\vec{\pi}\left(\tau + \frac{\delta\tau}{2}\right) = \vec{\pi}\left(\tau - \frac{\delta\tau}{2}\right) - \left[ \frac{\delta S(\tau)}{\delta \vec{x}} \right] \frac{\delta\tau}{2}$$

- Final half step:

$$\vec{\pi}(\tau_0) = \vec{\pi}\left(\tau_0 - \frac{\delta\tau}{2}\right) - \left[ \frac{\delta S(\tau_0)}{\delta \vec{x}} \right] \frac{\delta\tau}{2}$$

The derived configuration of  $\vec{x}$ , is accepted with probability  $\min(1, e^{\delta H})$ . This distribution satisfies the ergodicity of a Markov process since :

$$P(\vec{x})P(\vec{x} \rightarrow \vec{x}') = P(\vec{x}')P(\vec{x}' \rightarrow \vec{x}) \quad (45)$$

## 2.3 Practical considerations in Monte Carlo methods

### 2.3.1 Thermalization

Thermalization is a crucial process in Monte Carlo methods. It aims to bring the system to a state where its properties reflect those of a thermal equilibrium. The thermalization process typically begins from an initial configuration, which can be a completely ordered state (a "cold" start), a random distribution (a "hot" start), or a configuration assumed to be close to a thermalized state. However, the initial steps of thermalization

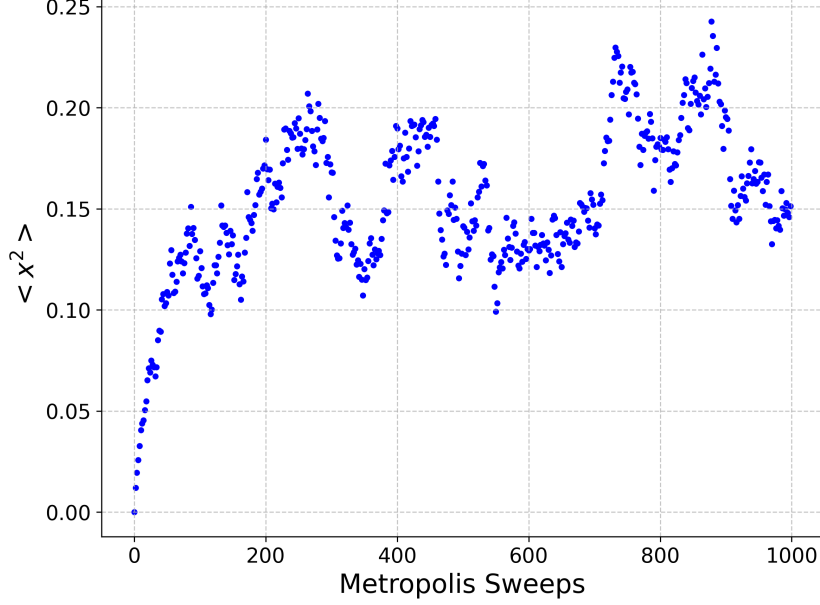


Figure 1: Thermalization process example from the Metropolis-Hastings algorithm simulation. The simulation is for a lattice size  $L = 30$  and acceptance rate 80%. The initial configuration is a vector of size  $L$  with zero values. The plotted data are the simulation results for each sweep for the observable  $\langle x^2 \rangle$ .

are not representative of the desired probability distribution and must be discarded to ensure accurate results. For the simulations we will present here the number of sweeps required for thermalization is determined through trial runs. In the trial runs we compute the average of the observable quantity  $\langle x^2 \rangle$ , that represents the energy of the oscillator, and exclude the necessary configurations from the analysis to ensure the system has reached a thermalized state. An example for the thermalization process for initial vector zero is presented in Figure 1. From this figure, we conclude that the first 50 to 100 sweeps are not thermalized and they are ignored throughout our analysis.

### 2.3.2 Autocorrelations

In order to perform any statistical analysis, we must ensure that the constructed configurations - measurements of the Markov process are independent. The configurations produced by our algorithms are in principle correlated, since each configuration is proposed based on the the previous.

To quantify the correlations between our samples we compute an autocorrelation function. The autocorrelation for an observable  $\hat{O}$ , denoted as  $\rho_O(t)$ , is defined as the correlation between values of  $\hat{O}$  at different Monte Carlo times  $t_{MC}$ :

$$\rho_O(t) = \frac{\langle (\hat{O}(t') - \langle \hat{O} \rangle) (\hat{O}(t' + t) - \langle \hat{O} \rangle) \rangle_{t'}}{\langle (\hat{O} - \langle \hat{O} \rangle)^2 \rangle}, \quad (46)$$

where  $\langle \cdot \rangle$  represents the average value,  $\langle \hat{O} \rangle$  is the mean value of the observable, and  $t'$  denotes the Monte Carlo time up to a maximum value of  $t_{\max} - t$ . The normalization is chosen such that  $\rho_O(0) = 1$ .

The autocorrelation function  $\rho$  drops exponentially as:

$$\rho(t) \sim e^{-t/t_{int}} \quad (47)$$

where  $t_{int}$  is the time scale of decorrelation of the measurements  $\hat{O}$  and it is called the autocorrelation time. To calculate the autocorrelation time we fit  $\rho$  and then we use the autocorrelation time to determine how many configurations to skip in the Markov Chain. By skipping the correlated sweeps, we reduce the correlation between our samples and thus our error analysis can proceed assuming uncorrelated measurements.

### 2.3.3 Jackknife Error Analysis

Due to the correlated nature of the Markov chain algorithms we employ the jackknife method to estimate errors and the expectation values of our observables. To overcome the problems arising from non-independent samples, this method divides a sample into blocks or bins. To implement this method we perform the following:

- The data is split into  $nb$  bins, each containing  $b = n - (n/nb)$  elements.
- The bin  $j$  contains the part of the sample obtained after omitting the contents of the  $j$ -th bin from the full sample  $O_0, O_1, \dots, O_{n-1}$ .
- We calculate the average value of  $O$  in each bin, resulting in  $O_0^b, O_1^b, \dots, O_{nb-1}^b$ .

The jack-knife error in the measurement of  $O$  is evaluated as:

$$(\delta O)^2 = \sum_{j=0}^{nb-1} (O_j^b - \langle O^b \rangle)^2 = nb (\langle (O^b)^2 \rangle - \langle O^b \rangle^2) \quad (48)$$

The expectation values are calculated as:

$$\langle O \rangle = \frac{1}{n} \sum_{i=0}^{n-1} O_i \quad (49)$$

## 3 Flow-based Generative Models

In this chapter we introduce a flow-based MCMC that can be used to generate lattice field configurations. Utilizing normalizing flows [16], we transform a simple base distribution into a targeted, complex distribution through efficient invertible mappings. This approach may improve the Markov chain's autocorrelation properties, yielding

a more efficient sampling method. By incorporating Metropolis-Hastings steps, we ensure that the generated samples approach the desired probability distribution [10]. This technique allows for systematic improvements in the autocorrelation time through model training, only by a model evaluation and action computation at each step of the chain. The main advantage of these models is that they do not need pre-existing samples from the desired probability distribution for training since the generated samples are produced by the model itself.

The proposed MCMC process creates field configurations  $x \in \mathbb{R}^D$ , following a target probability distribution given by

$$p(x) = \frac{e^{-S(x)}}{Z}, \quad Z = \int \prod_{j=1}^L dx_j e^{-S(x)}, \quad (50)$$

where  $j$  denotes the  $L$  components of  $x$ ,  $S(x)$  defines the action of the system, and  $Z$  is the partition function. The field configuration  $x$  is a vector composed of  $L$  real components on a finite, discrete spacetime lattice. The generated Monte Carlo steps are stochastic and the probabilities for each possible transition  $x \rightarrow x'$  must be non-negative and normalized. Each possible transition from a starting configuration  $x$  to any other configuration  $x'$  must be possible in a finite number of steps, satisfying the conditions of ergodicity [17].

To ensure that the generated samples  $p(x)$  are approaching the desired distribution, we incorporate Metropolis-Hastings steps [6] into our MCMC procedure. The Metropolis-Hastings acceptance criterion is

$$A(x_{i-1}, x'_i) = \min \left( 1, \frac{\tilde{p}(x_{i-1})p(x'_i)}{p(x_{i-1})\tilde{p}(x'_i)} \right). \quad (51)$$

If the criterion is satisfied then,  $x_i = x'_i$ , otherwise  $x_{(i)} = x_{i-1}$ . This method defines the transition probabilities of the Markov chain and ensures that the generated configurations are accurate representations of the target distribution.

### 3.1 RealNVP Model

RealNVP (Real-valued Non-Volume Preserving) [11] presents a machine learning approach, used to define the proposal distribution  $\tilde{p}(x)$  for a generative Metropolis-Hastings algorithm. The proposed normalizing flow model has the capability to produce samples along with their associated probability densities.

Given a random variable  $z$  with a known probability density  $r(z)$ , we can construct a new random variable using a smooth, invertible mapping function  $f$  such as  $z = f^{-1}(x)$ .

Considering the definition of the probability distribution,

$$\int p(x)dx = \int r(z)dz = 1$$

we can calculate the probability density of the new variable  $p(x)$  as:

$$p(x) = r(z) \left| \frac{dz}{dx} \right| = r(f^{-1}(x)) \left| \frac{df^{-1}(x)}{dx} \right|. \quad (52)$$

To utilize this idea to our models we generalize for mappings,  $f^{-1} : \mathbb{R}^L \rightarrow \mathbb{R}^L$  through the multivariable change-of-variables theorem. The transformation will then give:

$$\tilde{p}_f(x) = r(f(x)) \left| \det \frac{\partial f(x)}{\partial x} \right|. \quad (53)$$

where the transformation function  $f$  is invertible and has a computable Jacobian matrix and  $\det \frac{\partial f(x)}{\partial x}$  is the Jacobian determinant of the function  $f$ . This function is constructed using several invertible affine coupling layers. In this setup, an  $L$ -dimensional vector  $x$  is split into two halves,  $x_a$  and  $x_b$ , each having  $(L/2)$  dimensions. Then a single coupling layer, labeled as  $g_i$ , transforms  $x$  into a new vector  $z$  with components defined as:

$$g_i^{-1}(x) = \begin{cases} z_a = x_a \\ z_b = x_b \odot e^{s_i(x_a)} + t_i(x_a) \end{cases} \quad (54)$$

where  $s_i(x_a)$  and  $t_i(x_a)$  are neural networks that represent scale and translation operations. This transformation is favorable as the Jacobian matrix, which describes it, has a lower-triangular form, simplifying the computation of its determinant. The determinant can be calculated as:

$$\begin{aligned} \left| \det \frac{\partial g_i^{-1}(x)}{\partial x} \right| &= \det \begin{bmatrix} \mathbf{I}_{L/2} & \mathbf{0} \\ \frac{\partial z_b}{\partial x_a} & \text{diag}(e^{s_i(x_a)}) \end{bmatrix} \\ &= \prod_{j=1}^{L/2} e^{[s_i(x_a)]_j} = \exp \left( \sum_{j=1}^{L/2} [s_i(x_a)]_j \right), \end{aligned} \quad (55)$$

where  $j$  indexes the  $(L/2)$  output elements of  $s_i$ . This form does not require computing the inverse of the neural networks, nor does it involve calculating their Jacobian, thereby allowing  $s_i$  and  $t_i$  to be arbitrarily complex while still making the Jacobian determinant easily computable. The full function  $f$  is constructed by stacking several coupling layers, alternating the role of  $x_a$  and  $x_b$  in each layer:

$$f(x) = g_1(g_2(\dots g_n(x) \dots)). \quad (56)$$

The training of our model involves the minimization of shifted Kullback-Leibler (KL) divergence loss function, that can present a measure of the difference between the target distribution  $p(x)$  and the generated distribution  $\tilde{p}_f(x)$ . The used loss function is expressed as:

$$L(\tilde{p}_f) = D_{KL}(\tilde{p}_f||p) - \log Z = \int \tilde{p}_f(x)(\log \tilde{p}_f(x) + S(x))dx \quad (57)$$

where  $S(x)$  is the action of the physical system. The optimization of this loss function is performed using stochastic gradient descent.

## 4 Results

### 4.1 Computational Setup

Parameter	Explanation
$\tau$	Imaginary time
$\delta\tau$	Lattice spacing on the discrete imaginary time lattice
$N$	Total number of Metropolis-Hastings sweeps
$L$	Total site number of the time lattice ( lattice size )
$N_\tau$	Number of time slices
$\lambda$	Coupling constant for the Anharmonic Oscillator
$\tilde{m}$	Dimensionless mass
$\tilde{\omega}$	Dimensionless frequency
$\tilde{x}_i$	Dimensionless position on the discrete time lattice
$\tilde{S}$	Dimensionless Euclidean action
$t_{int}$	Autocorrelation time
$epoch$	Training Iterations for Machine Learning model
$r_{acc}$	Acceptance rate

Table 1: Summary of the symbols and terminology used throughout the simulation procedure.

In this section, we provide an overview of our computational setup, detailing the methods and outcomes for computing the observable of the ground energy ( $E_0$ ) of



the Harmonic and Anharmonic oscillators using three distinct computational methods: Metropolis-Hastings Monte Carlo, Hybrid Monte Carlo (HMC), and flow-based Monte Carlo with RealNVP. The theoretical value for the ground state energy of the Harmonic oscillator is 0.5 as calculate from Eq. (30). Since the Anharmonic Oscillator, as we discussed in section 1.2, does not have analytical solution, we compare our results for the ground energy with other approximation methods [1],[2]. The approximate value of quantum Anharmonic Oscillator with a quartic term has been found 0.8038.

Our simulations are conducted on a discrete time of  $L$  time slices with periodic boundary conditions, such that:

$$\begin{aligned} x(L) &= x(0), \\ x(-1) &= x(L-1). \end{aligned}$$

For all simulations, we use the Euclidean time formalism ( $\tau = it$ ), as discussed in Section 1, and fix the total time to  $L \times \delta\tau = 6$ . In order to have true comparison between the models, ensembles of  $4, 5 \times 10^6$  lattice configurations were generated for all methods.

This section conducts a thorough comparison of these models, assessing their performance, which includes examining autocorrelation times, and the evaluation of the obtained results. In Table 1 there is an overview of the terminology used throughout our analysis.[18]

In terms of computational resources, the Metropolis-Hastings and Hybrid Monte Carlo simulations were conducted on CPU, while the RealNVP model on GPU (NVIDIA GTX 1080).

## 4.2 Metropolis-Hastings Monte Carlo Method

In the implementation of the Metropolis-Hastings Monte Carlo algorithm, we initially conducted trials to determine the range  $h$  using the algorithm described in Section 2.1. Our target acceptance rate was set to be close to 80%. We repeated this procedure for various numbers of sites and different coupling constants  $\lambda$ . Subsequently, we simulated the particle positions with the Metropolis-Hastings algorithm ( $N_\tau \times N_{\text{sweeps}}$  elements) for both Harmonic and Anharmonic oscillators.

For the Anharmonic Oscillator, we specifically set the coupling constant  $\lambda$  to 1. We did not explore other values of  $\lambda$  because our primary focus was on testing the applicability to the Anharmonic case rather than comparing different values of  $\lambda$ .

We visually inspected the observable  $\langle x^2 \rangle$  to check the thermalization process. Sweeps that were not thermalized were discarded, and we proceeded by constructing the autocorrelation function for each lattice spacing  $\delta\tau$ . The plot was fitted with an exponential

function to calculate the autocorrelation time  $t_{int}$ . We skipped appropriate sweeps and used the jackknife method to compute the ground state energy  $E_0$  for each  $\delta\tau$ .

We begin our results discussion by analyzing the continuum limit of the ground state energy, showcased in Figure 2. Here, we observe the effects of varying lattice spacings  $\delta\tau$ , on the calculation of the ground state energy. The spline fit in the referred figure is an indicator of the result for our simulation since it can be used as an estimate of the energy by extrapolation. Figure 2a shows the improvement on the accuracy for the calculation of the ground state energy by reducing the time lattice spacing until the last results where the autocorrelation limitations take over. In the Anharmonic Oscillator 2b, although it presents lower jackknife error, the lattice spacing affects the results in a more dramatic way.

Next, table 2 and table 3 detail the outcomes for both the Harmonic and Anharmonic oscillators, indicating the lattice size  $L$ , the ground state energy  $E_0$ , calculated jackknife errors and autocorrelation time. Larger autocorrelation times  $t_{int}$  result in increasing jackknife errors since the size of each ensemble is decreasing for higher  $t_{int}$ .

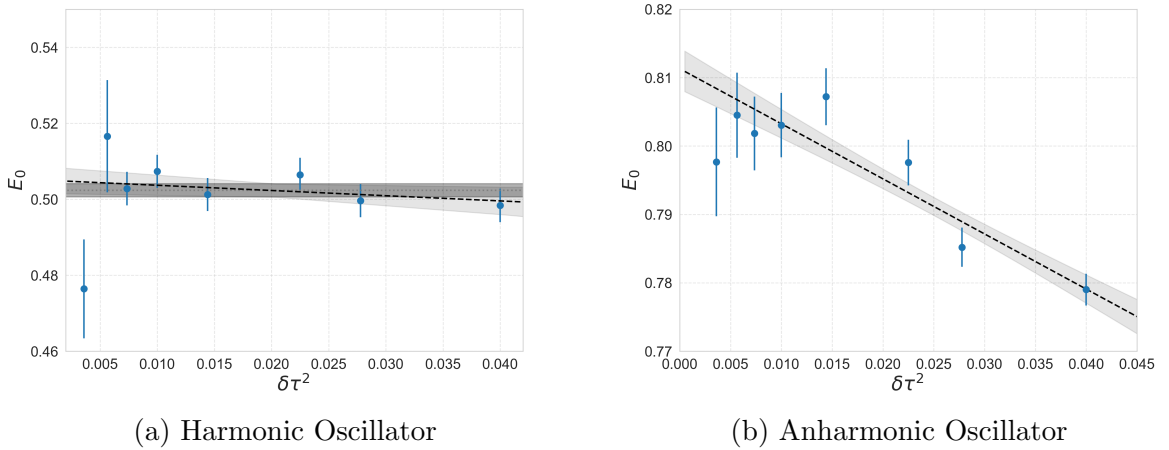


Figure 2: Ground state energy continuum limit calculation for a) Harmonic and b) Anharmonic Oscillator ( $\lambda = 1$ ) using the Metropolis-Hastings algorithm. The filled circles represent the values calculated from the Metropolis-Hastings algorithm with their corresponding errors. The dashed line represents a linear fit of the data, and the shaded part represents the fitting uncertainty.

### 4.3 Hybrid Monte Carlo Method

Following the implementation of the Metropolis-Hastings Monte Carlo we proceed with the Hybrid Monte Carlo (HMC) algorithm. In this simulation, we undertook the following steps:

1. **Integration Time Selection:** We began by selecting an appropriate integration

Metropolis-Hastings Monte Carlo Harmonic Oscillator Results			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.4984(44)	167.67
36	0.1667	0.4996(43)	262.22
40	0.15	0.5065(45)	294.65
50	0.12	0.5012(43)	399.21
60	0.1	0.5073(44)	719.27
70	0.0857	0.5028(44)	905.55
80	0.075	0.517(15)	1019.56
100	0.06	0.476(13)	1663.98

Table 2: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the Metropolis-Hastings Quantum Harmonic Oscillator, alongside the autocorrelation times ( $t_{int}$ ).

Metropolis-Hastings Anharmonic Oscillator Results			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.7790(23)	36.47
36	0.1667	0.7852(29)	55.14
40	0.15	0.7976(33)	71.29
50	0.12	0.8072(42)	108.47
60	0.1	0.8030(47)	142.99
70	0.0857	0.8018(54)	189.88
80	0.075	0.8045(62)	256.97
100	0.06	0.7977(80)	410.62

Table 3: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the Metropolis-Hastings Anharmonic Oscillator, along with autocorrelation times  $t_{int}$ .

time for the leapfrog algorithm. The chosen integration time was adjusted to maintain an acceptance rate above 70% for each lattice size  $L$ .

2. **Simulation:** We conducted simulations for both the Harmonic and Anharmonic oscillators, specifically for the case where  $\lambda = 1$ .
3. **Thermalization:** To ensure accurate results, we monitored the observable  $\langle x^2 \rangle$  to visualize the thermalization process. Sweeps that did not exhibit thermalization were discarded.
4. **Autocorrelation Time:** For each lattice spacing  $\delta\tau$ , we constructed the autocorrelation function and fitted it with an exponential function to calculate the autocorrelation time.

5. **Ground Energy Calculation:** We used the jackknife method to compute the ground energy ( $E_0$ ) for each lattice spacing. This analysis was performed for both the Harmonic and Anharmonic oscillators.

Initially, we report our calculations in table 4 and table 5 for both Harmonic and Anharmonic oscillators. From the report of the calculations it is evident that the autocorrelation time  $t_{int}$  is decreased relative to the Metropolis-Hastings algorithm yet continues to play a substantial role in determining the ground state energy.

Moving forward, we present the continuum limit of the ground state energy as depicted in Figure 3 for both Harmonic and Anharmonic oscillators. In Figure 3a, we observe that the precision in calculating the ground state energy increases with the reduction of the time lattice spacing, until the point where the autocorrelation effects become dominant. Despite this, the Harmonic oscillator results are notably close to the theoretical value of 0.5, while the jackknife errors still increase for smaller lattice spacing's.

Hybrid Monte Carlo Harmonic Oscillator Algorithm Results			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.5003(4)	6.20
36	0.1667	0.5017(14)	10.80
40	0.15	0.5022(14)	14.68
50	0.12	0.5007(14)	29.23
60	0.1	0.5032(14)	50.22
70	0.0857	0.5040(14)	81.10
80	0.075	0.5102(45)	123.61
100	0.06	0.5066(44)	248.66

Table 4: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the Hybrid Monte Carlo Harmonic Oscillator algorithm, along with autocorrelation times  $t_{int}$ .

Hybrid Monte Carlo Anharmonic Oscillator Results			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.7832(12)	7.34
36	0.1667	0.7979(41)	10.78
40	0.15	0.7950(39)	11.12
50	0.12	0.7875(38)	17.07
60	0.1	0.7972(38)	26.64
70	0.0857	0.7954(38)	38.70
80	0.075	0.8032(39)	54.45
100	0.06	0.7941(38)	86.56

Table 5: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the Hybrid Monte Carlo Anharmonic Oscillator, along with autocorrelation times  $t_{int}$ .

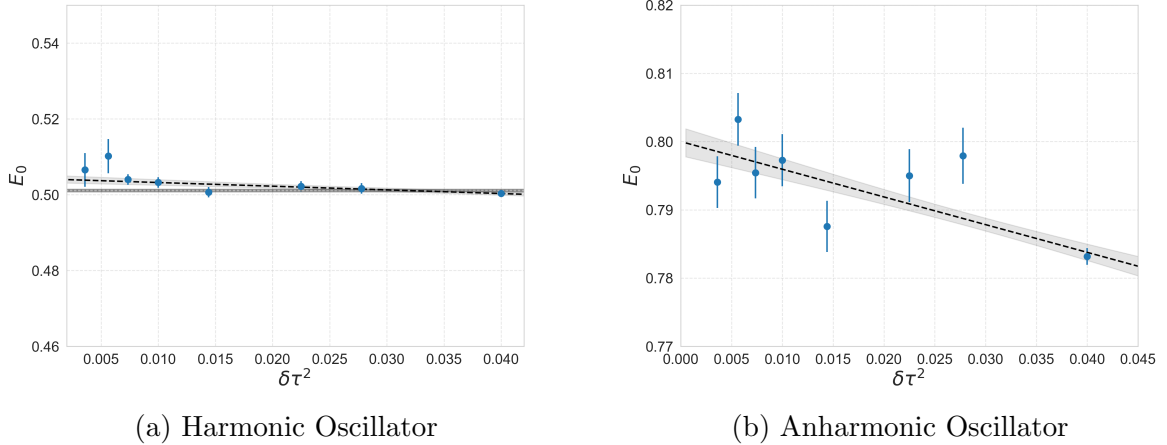


Figure 3: Ground state energy continuum limit calculation for a) Harmonic Oscillator and b) Anharmonic Oscillator ( $\lambda = 1$ ) using the Hybrid Monte Carlo algorithm. The filled circles represent the values calculated from the Hybrid Monte Carlo algorithm with their corresponding errors. The dashed lines in a) represent linear and constant fit of the data, in b) represent linear fit. The shaded part represents the fitting uncertainty.

#### 4.4 Flow-based Monte Carlo

From the implementation of both the Metropolis-Hastings and Hybrid Monte Carlo methods, it is apparent that the autocorrelation issue persists for smaller lattice sizes

despite efforts to mitigate this problem. In an attempt to find a solution, we implemented the RealNVP algorithm as detailed in Section 3.1.

In our setup, the RealNVP employs six affine coupling layers outlined in Equation 54. Each neural network ( $s_i, t_i$ ) responsible for scale and translation operations is structured with three fully connected layers, facilitated by ReLU activation functions. For the parameter optimization we used a Stochastic Gradient Descent optimizer with a fixed learning rate of  $5 \times 10^{-3}$ . This setup proved to be efficient for both Harmonic and Anharmonic oscillators. We chose the prior distribution to be Gaussian, denoted as  $\mathcal{N}(0, 1)$ , while the target distribution is determined by the Euclidean action, for the Harmonic (40) and Anharmonic (41) oscillators, respectively.

During each iteration in the training process, a tensor with random Gaussian values is generated representing the prior distribution  $\tilde{p}(x)$ . This tensor undergoes transformations through the affine coupling layers. The new distribution from the transformation and the target distribution are used to calculate the Kullback-Leibler (KL) loss which guides the training of the neural networks through gradient descent optimization. Back-propagation is used to compute the gradients of the loss with respect to each parameter in the neural networks. The optimizer updates the weights of the neural networks based on the calculated gradients to minimize the loss.

Periodically, the Metropolis-Hastings criterion is applied to new data sets to assess the acceptance rate and evaluate the model's performance over time. For each lattice size an acceptance rate over 20% was targeted. It is important to note that the minimization of the loss is not an optimal indicator of the model's performance, as it could converge to a 'bad' minima. Using the weights and the parameters of the trained model we construct the ensemble of our simulation.

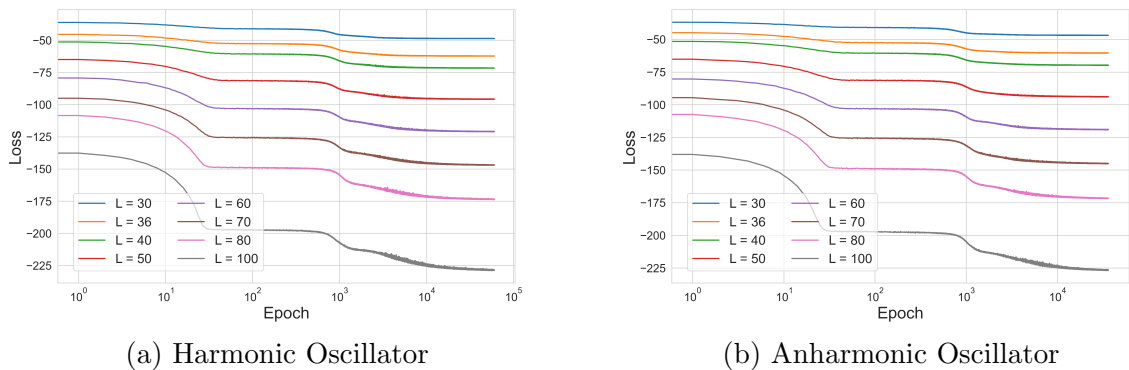


Figure 4: Loss versus epochs (training steps) for the RealNVP model of the Quantum a) Harmonic and b) Anharmonic Oscillator. We used the same learning rate and architecture as well as the same number of epochs. Different colors represent different simulation lattice sizes as indicated in the legend.

To estimate the performance of our models for each lattice size we initially plotted

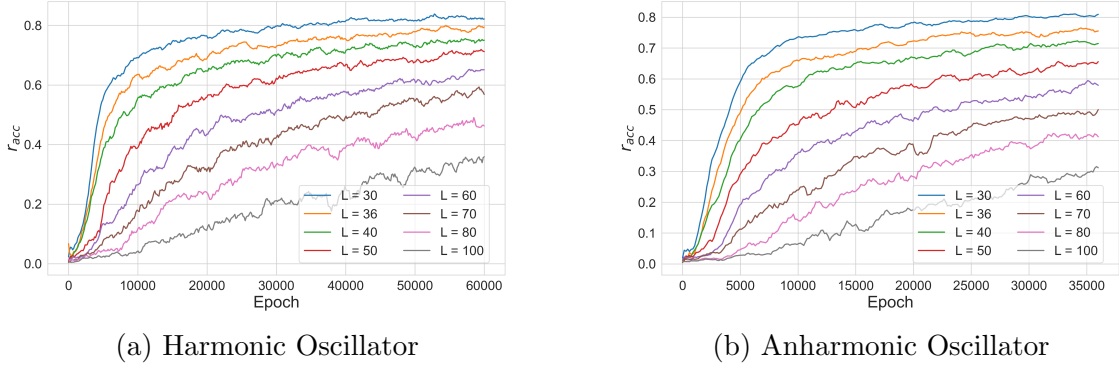


Figure 5: Acceptance rate versus epochs (training iterations) for the RealNVP model. Different line colors represent simulations for different lattice sizes  $L$ . For better representation we used the moving average in our data with window 10.

the loss values of the simulation as we show in Figure 4 for both Harmonic and Anharmonic oscillators. This figure displays two distinct points of inflection, for all lattice sizes, indicating changes on the training rate of the algorithm. The points appear for the same epoch index for all the lattice sizes,  $50$  and  $10^3$  for both Harmonic and Anharmonic Oscillator.

To further assess the model's performance, we display the acceptance rate  $r_{acc}$  of the Metropolis-Hastings criterion versus epochs in Figure 5. In contrast to loss graph, the accuracy rate shows high dependency on the lattice size. Larger lattice sizes require more epochs to achieve a sufficient acceptance rate. The graph also hints at a potential maximum acceptance rate for each lattice size, since each line appears to stabilize after a certain number of epochs. This findings can be justified since the feature space of the algorithm is increasing with  $L$ .

Moreover, the graph indicates that the Anharmonic Oscillator reach at a satisfactory acceptance rate in fewer epochs compared to the Harmonic oscillator. This observation suggest that the added quartic term in the Anharmonic Oscillator assists the training process.

Following the performance estimation we perform the analysis of the produced data in terms of autocorrelations and thermalization processes as in the previous methods. The results of the simulation are presented in Table 6 and Table 7 followed by the calculation of the continuum limit in Figure 6. These results showcase a noticeable improvement in terms of accuracy. The autocorrelation times  $t_{int}$ , relative to the other methods, are dramatically decreased and remained at same orders of magnitude for all lattice sizes. Jack-Knife errors also present considerable decrease.

<b>RealNVP Harmonic Oscillator Model Results</b>			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.5017(25)	0.659
36	0.1667	0.4998(8)	0.763
40	0.15	0.5048(24)	0.808
50	0.12	0.4979(24)	0.954
60	0.1	0.5012(24)	1.420
70	0.0857	0.5013(8)	2.507
80	0.075	0.5023(8)	4.526
100	0.06	0.4962(24)	10.754

Table 6: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the RealNVP Monte Carlo model for the Harmonic Oscillator, along with autocorrelation times  $t_{int}$ .

<b>RealNVP Anharmonic Oscillator Results</b>			
$L$	$\delta\tau$	$E_0(\pm\delta E_0)$	$t_{int}$
30	0.2	0.7811(4)	7.34
36	0.1667	0.7880(4)	10.78
40	0.15	0.7905(4)	11.12
50	0.12	0.7954(12)	17.07
60	0.1	0.7968(12)	26.64
70	0.0857	0.7979(12)	38.70
80	0.075	0.8019(12)	54.45
100	0.06	0.7990(39)	86.56

Table 7: Ground state energy  $E_0$  as a function of lattice spacing  $\delta\tau$  with associated errors for the RealNVP Anharmonic Oscillator, along with autocorrelation times  $t_{int}$ .



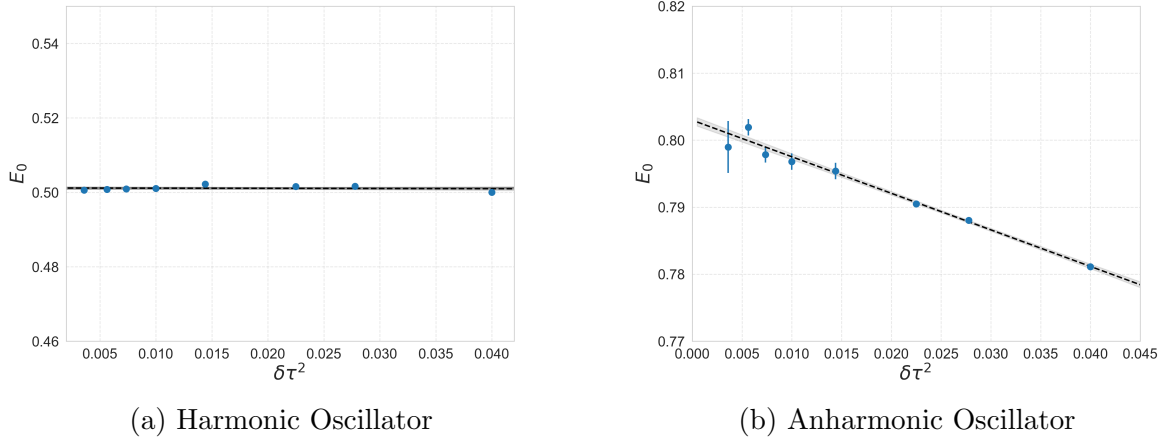


Figure 6: Ground state energy continuum limit calculation for a) Harmonic Oscillator and b) Anharmonic Oscillator using the RealNVP algorithm. The filled circles represent the values calculated from the RealNVP algorithm with their corresponding errors. The black dashed line represents a linear fit of the data, the grey dashed line the constant fit and the shaded part represents the fitting uncertainty. The errors that do not appear in the graph are smaller than the data circles.

## 4.5 Model Comparison

In this chapter, we conduct a comparison of the three models: Metropolis-Hastings, Hybrid Monte Carlo, and RealNVP Monte Carlo. Firstly, we need to ensure that the employed methods yield results that are closely aligned. To confirm that the generated ensembles represent our theory, we present the distributions of various observables.

The distributions for the observable  $\langle x^2 \rangle$  for both the Harmonic and Anharmonic Oscillator are illustrated in Figure 7 and Figure 8 for all three models. Moreover, we exhibit the distributions of the observable  $\langle x^4 \rangle$  in Figure 9 and Figure 10, along with the distributions of the action  $S(x)$  for both oscillators in Figure 11 and Figure 12. Examining these figures verifies that all the models produce similar distributions for all observables, confirming that the simulations produce credible configurations.

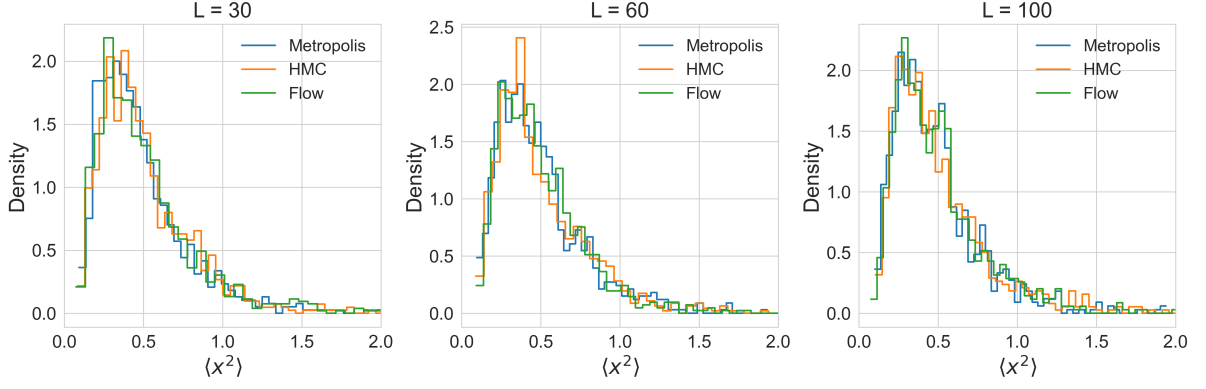


Figure 7:  $\langle x^2 \rangle$  distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

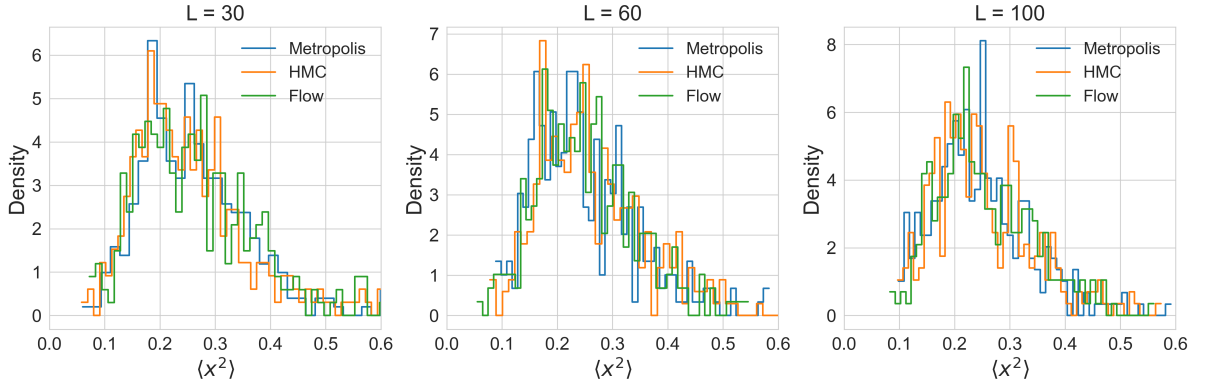


Figure 8:  $\langle x^2 \rangle$  distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

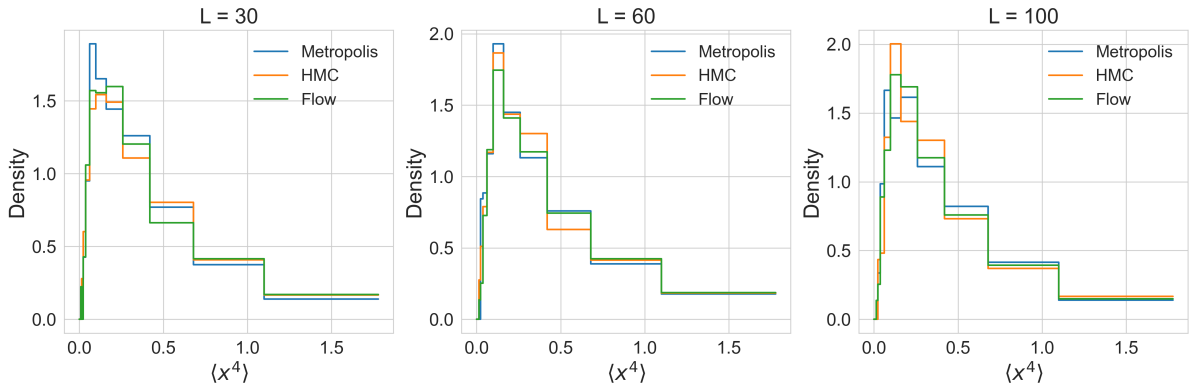


Figure 9:  $\langle x^4 \rangle$  distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

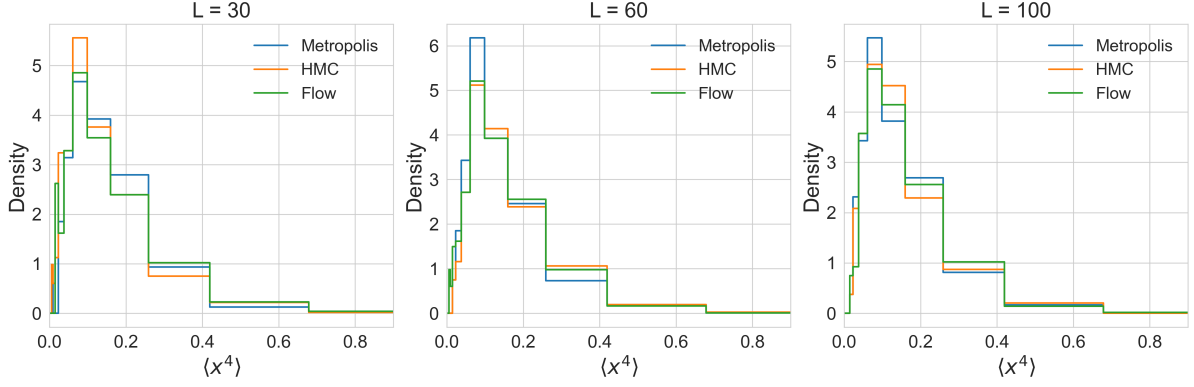


Figure 10:  $\langle x^4 \rangle$  distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

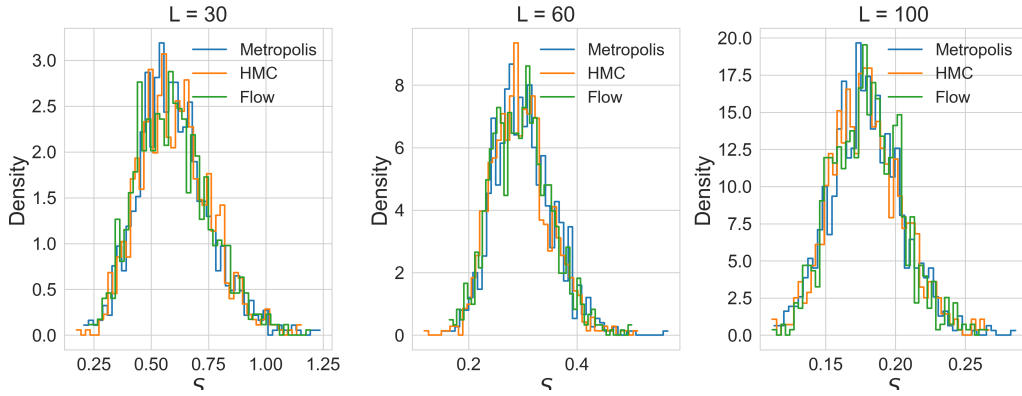


Figure 11: The action distribution of the Harmonic oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

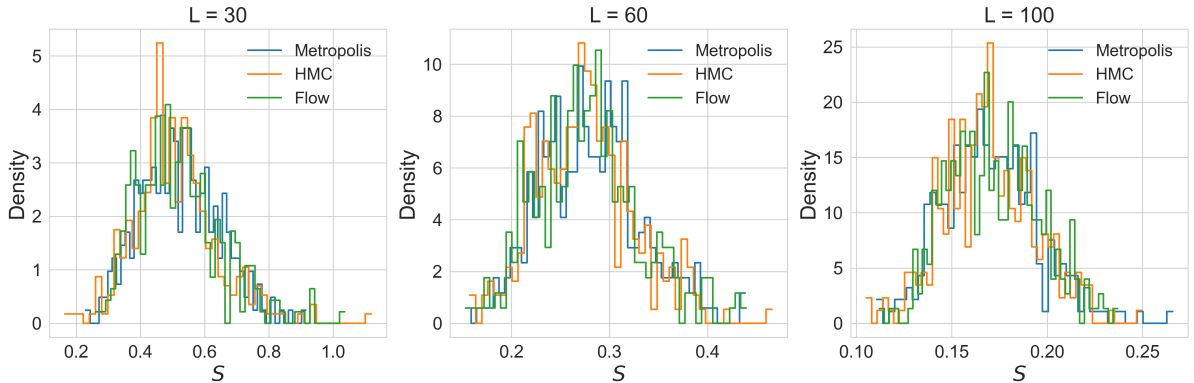


Figure 12: The action distribution of the Anharmonic Oscillator for the three methods Metropolis-Hastings, Hybrid Monte Carlo and RealNVP. Each graph represents different lattice sizes  $L = 30, 60, 100$ .

Moving forward from verifying the results through observable distributions, we next

focus on the autocorrelation time relative to lattice size, showcased in Figure 13. A clear improvement is noted in the autocorrelation time when progressing from the Metropolis-Hastings model to the Hybrid Monte Carlo, and further to the RealNVP. This trend is consistent for both Harmonic and Anharmonic oscillators. The improved results from Hybrid Monte Carlo can be attributed to its use of smarter proposals in the construction of the Markov chain. Meanwhile, RealNVP produces precise target distributions by training and transformations through affine coupling layers.

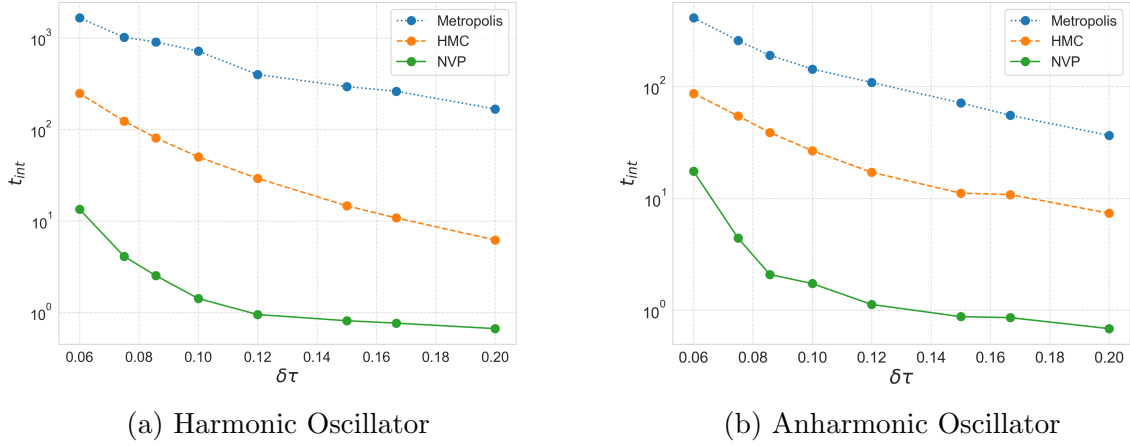


Figure 13: Autocorrelation times  $t_{int}$  for the three methods used (Metropolis-Hastings, Hybrid Monte Carlo, and RealNVP) relative to lattice spacing  $\delta\tau$ . a) Acceptance rates: Metropolis-Hastings 0.79 – 0.81, 0.72 – 0.85, RealNVP: 0.20 – 0.25. b) Acceptance rates: Metropolis-Hastings: 0.80 – 0.82, Hybrid Monte Carlo: 0.73 – 0.88, RealNVP: 0.20 – 0.25. The  $t_{int}$  axis is logarithmic for presentation purposes.

Furthermore, in Figure 14 we illustrate the estimate value for the ground state energy for each of our methods for both Harmonic and Anharmonic Oscillators, with their corresponding errors. The estimated value of the ground state energy  $E_0$  is derived from the best fit of the data obtained from each of our methods. For the selection of the best fit, we performed constant, linear and quadratic fit to the data using from three up to eight data points. From this selection of fits we chose the one that minimizes the  $\chi^2$  per degrees of freedom (number of data points - number of parameters). The inner error bars represent the statistical Jack-Knife errors in the estimation of the ground state energy. The outer error bars represent the statistical errors but also the systematic errors (Total error =  $\sqrt{\text{Systematic}^2 + \text{Statistical}^2}$ ). The systematic error is calculated as the weighted standard deviation of the  $E_0$  values obtained from constant, linear and quadratic fits of different number of data samples. Overall the Figure 14 illustrates that RealNVP provides more accurate results, proving the effectiveness of the methodology.

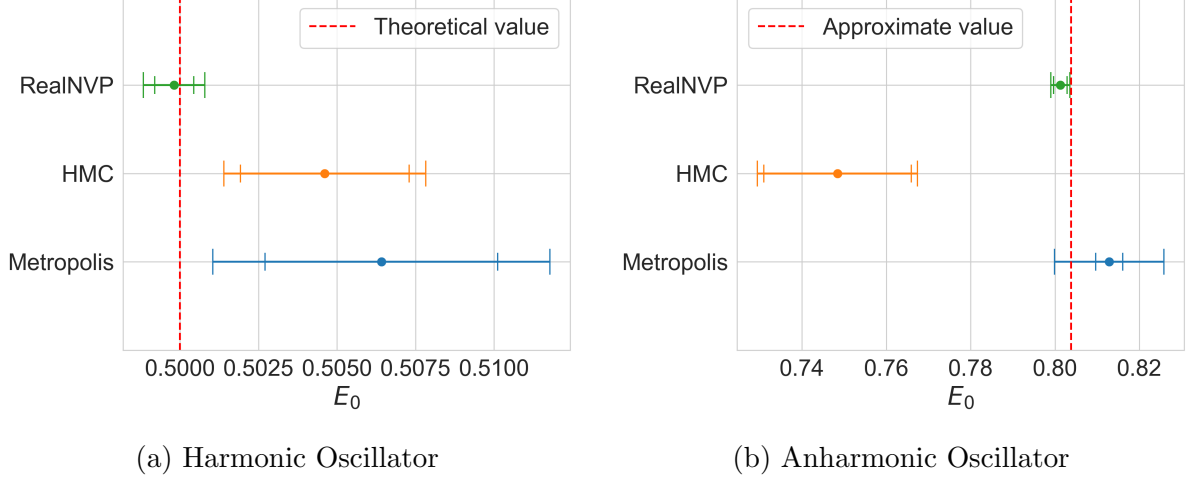


Figure 14: The dots represent the estimate value of ground state energy for the three simulations from their best fit with their corresponding errors. Inner bar: Statistical error, Outer error bar: Statistical and Systematic error. Best fit for Harmonic Oscillator: i) Metropolis: Linear fit, 7 data points ii) HMC: Quadratic fit, 4 data points iii) RealNVP: Quadratic fit, 6 data points. Best fit for Anharmonic Oscillator: i) Metropolis: Linear fit, 6 data points ii) HMC: Linear fit, 3 data points iii) RealNVP: Quadratic fit, 8 data points. Theoretical value is 0.5 and the approximate values is 0.8038 [1], [2].

## 5 Conclusions

In this study, we examined the implementation of RealNVP sampling as part of a generative Metropolis-Hastings algorithm for Monte Carlo simulations of Harmonic and Anharmonic Quantum Oscillators. The RealNVP method, compared to Hybrid Monte Carlo and Metropolis-Hastings Monte Carlo, confirmed that this sampling method can indeed produce ensembles according to the correct distribution of the theory.

We demonstrated that as the lattice spacing approaches zero, Metropolis-Hastings and Hybrid Monte Carlo show critical slowing down in terms of increased autocorrelation time and, consequently, increasing statistical errors at equal computational cost. The implementation of the flow-based Monte Carlo model presents notable advancements in mitigating these limitations when approaching the continuum limit, at the cost of increased training time (see Figure 5).

The two systems to which we applied the RealNVP method were chosen to include one with an analytical solution, the Quantum Harmonic Oscillator, as well as one for which an analytic solution is not available, namely the Quantum Anharmonic Oscillator. Both are simple, one-dimensional systems which allow quickly benchmarking our models and allow for investigating things such as hyperparameter optimization, changes of neural network architecture and scaling techniques, the dependence on the number of affine coupling layers, and model variations such as NICE [19], which preserves the original symmetrical attributes of a distribution by omitting the scale term, or more

advanced models like GLOW [20], which employs invertible convolutions, may prove to be beneficial in certain theoretical frameworks.

In conclusion, we have developed a flow-based approach for the Monte Carlo simulation of the Quantum Harmonic and Anharmonic Oscillators, which are relatively simpler systems than those recently employed in the literature, such as the scalar field theory [10, 21], and lattice gauge theories [22–26]. We showed that in principle we can construct uncorrelated samples with this method, while further investigation must follow in order to address more quantitative questions such as how the method scales with the physical volume, higher dimensions, and dependence on the model parameters such as the magnitude of the anharmonic potential. Our research can be used as a starting point for addressing these questions in the future.



## References

- [1] F. Hioe and E. Montroll, “Quantum theory of anharmonic oscillators. i. energy levels of oscillators with positive quartic anharmonicity,” *Journal of Mathematical Physics*, vol. 16, no. 9, pp. 1945–1955, 1975.
- [2] S. Mittal, M. J. Westbroek, P. R. King, and D. D. Vvedensky, “Path integral monte carlo method for the quantum anharmonic oscillator,” *European Journal of Physics*, vol. 41, no. 5, p. 055401, 2020.
- [3] R. P. Feynman, “Space-time approach to non-relativistic quantum mechanics,” *Rev. Mod. Phys.*, vol. 20, pp. 367–387, Apr 1948. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.20.367>
- [4] P. A. M. Dirac, *The Principles of Quantum Mechanics*. Oxford,: Clarendon Press, 1930.
- [5] R. Feynman and A. Hibbs, *Quantum Mechanics and Path Integrals*, ser. International Earth and Planetary Sciences Series. McGraw-Hill, 1965. [Online]. Available: <https://books.google.com.cy/books?id=14ApAQAAMAAJ>
- [6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. [Online]. Available: <http://link.aip.org/link/?JCP/21/1087/1>
- [7] C. Gattringer and C. Lang, *Quantum Chromodynamics on the Lattice: An Introductory Presentation*, ser. Lecture Notes in Physics. Springer Berlin Heidelberg, 2009. [Online]. Available: <https://books.google.com.cy/books?id=l2hZKnlYDxoC>
- [8] I. Montvay and G. Münster, *Quantum Fields on a Lattice*, ser. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1994.
- [9] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, “Hybrid monte carlo,” *Physics Letters B*, vol. 195, no. 2, pp. 216–222, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/037026938791197X>
- [10] M. Albergo, G. Kanwar, and P. Shanahan, “Flow-based generative models for markov chain monte carlo in lattice field theory,” *Physical Review D*, vol. 100, no. 3, aug 2019. [Online]. Available: <https://doi.org/10.1103/PhysRevD.100.034515>
- [11] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” 2017.



- [12] U.-R. Kim, D. Kim, and J. Lee, “Bra-ket representation of the inertia tensor,” *Journal of the Korean Physical Society*, vol. 77, no. 11, pp. 945–951, nov 2020. [Online]. Available: <https://doi.org/10.3938%2Fjkps.77.945>
- [13] E. Merzbacher, *Quantum Mechanics*. Wiley, 1998. [Online]. Available: [https://books.google.com.cy/books?id=6Ja\\_QgAACAAJ](https://books.google.com.cy/books?id=6Ja_QgAACAAJ)
- [14] M. Creutz and B. Freedman, “A statistical approach to quantum mechanics,” *Annals of Physics*, vol. 132, no. 2, pp. 427–462, 1981. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0003491681900749>
- [15] S. Marik, S. Naskar, and S. Banerjee, “Path integral approach to driven quantum harmonic oscillator using markov chain monte carlo methods,” 2022.
- [16] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” 2016.
- [17] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970. [Online]. Available: <http://www.jstor.org/stable/2334940>
- [18] S. Mittal, M. J. E. Westbroek, P. R. King, and D. D. Vvedensky, “Path integral monte carlo method for the quantum anharmonic oscillator,” *European Journal of Physics*, vol. 41, no. 5, p. 055401, aug 2020. [Online]. Available: <https://dx.doi.org/10.1088/1361-6404/ab9a66>
- [19] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” 2015.
- [20] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” 2018.
- [21] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, “Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models,” *Phys. Rev. Lett.*, vol. 126, no. 3, p. 032001, 2021.
- [22] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, “Equivariant flow-based sampling for lattice gauge theory,” *Phys. Rev. Lett.*, vol. 125, no. 12, p. 121601, 2020.
- [23] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, “Sampling using  $SU(N)$  gauge equivariant flows,” *Phys. Rev. D*, vol. 103, no. 7, p. 074504, 2021.

- [24] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, “Flow-based sampling for fermionic lattice field theories,” *Phys. Rev. D*, vol. 104, no. 11, p. 114507, 2021.
- [25] R. Abbott *et al.*, “Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions,” *Phys. Rev. D*, vol. 106, no. 7, p. 074506, 2022.
- [26] S. Bacchio, P. Kessel, S. Schaefer, and L. Vaitl, “Learning trivializing gradient flows for lattice gauge theories,” *Phys. Rev. D*, vol. 107, no. 5, p. L051504, 2023.
- [27] M. J. E. Westbroek, P. R. King, D. D. Vvedensky, and S. Dürr, “User's guide to monte carlo methods for evaluating path integrals,” *American Journal of Physics*, vol. 86, no. 4, pp. 293–304, apr 2018. [Online]. Available: <https://doi.org/10.1119%2F1.5024926>
- [28] R. Phythian, “The functional formalism of classical statistical dynamics,” *Journal of Physics A: Mathematical and General*, vol. 10, no. 5, p. 777, may 1977. [Online]. Available: <https://dx.doi.org/10.1088/0305-4470/10/5/011>
- [29] L. Del Debbio, J. M. Rossney, and M. Wilson, “Efficient modeling of trivializing maps for lattice  $\phi^4$  theory using normalizing flows: A first look at scalability,” *Phys. Rev. D*, vol. 104, no. 9, p. 094507, 2021.