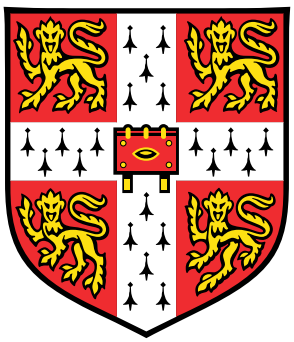
Optimal Importance Sampling in Quantum Monte Carlo for Lattice Models



Blaž Stojanovič

Supervisor: Prof. A. Lamacraft

Department of Physics

This dissertation is submitted for the degree of
Master of Philosophy in Scientific Computing

St. John's College May 2021

I would like to dedicate this thesis to …

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 15,000 words including appendices, figure legends, and tables.

Blaž Stojanovič
May 2021

Acknowledgements

And I would like to acknowledge ...

s

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

CNN	Convolutional Neural Network
DL	Deep Learning
DMC	Diffusion Quantum Monte Carlo
GAN	General Adversarial Network
ML	Machine Learning
NN	Neural Network
QMC	Quantum Monte Carlo
VAE	Variational Autoencoder
VMC	Variational Quantum Monte Carlo

Chapter 1

Introduction

1.1 Thesis Contributions

1.2 Thesis Structure

Chapter 2

Background

2.1 Lattice models

2.1.1 Historical introduction

I think starting from the magnetism point of view might be the best way to go, slowly lead into the field of condensed matter theory and lattice models.

2.1.2 The Schrödinger equation and the Feynman path integral

The wavefunction

$$\Psi(r_1, \ldots, r_N) \tag{2.1}$$

the Schrödinger equation

$$i\frac{\partial \psi(\boldsymbol{r}, t)}{\partial t} = \hat{H}\psi(\mathbf{r}, t) \tag{2.2}$$

for a single particle in an external potential $\hat{V}(\boldsymbol{r})$ the Hamiltonian is

$$\hat{H}\phi(\boldsymbol{r}) = -\frac{1}{2}\nabla^2\phi(\boldsymbol{r}) + \hat{V}(\boldsymbol{r})\phi(\boldsymbol{r}). \tag{2.3}$$

Alternatively to the Schrödinger equation one can use an integral Green's function representation to express the wavefunction ψ at some future time t_2 given initial condition $\psi(\boldsymbol{r}, t_1)$ as

$$\psi(\boldsymbol{r}_2, t_2) = \int \mathcal{K}(\boldsymbol{r}_2, t_2; \boldsymbol{r}_1, t_1)\, \psi(\boldsymbol{r}_1, t_1)\, \mathrm{d}\boldsymbol{r}_1. \tag{2.4}$$

The solution to equation

$$\left(i\frac{\partial}{\partial t_2} - H_{\boldsymbol{r}_2}\right)\mathcal{K}(\boldsymbol{r}_2, t_2; \boldsymbol{r}_1, t_1) = i\delta(\boldsymbol{r}_1 - \boldsymbol{r}_2)\,\delta(t_1 - t_2) \tag{2.5}$$

and the *propagator* $\mathcal{K}(\boldsymbol{r}_2, t_2; \boldsymbol{r}_1, t_1)$ is expressed using the Feynman path integral

$$\mathcal{K}(\boldsymbol{r}_2, t_2; \boldsymbol{r}_1, t_1) = \int_{\substack{r(t_1)=r_1 \\ r(t_2)=r_2}} \mathcal{D}\boldsymbol{r}(t) \exp\left(i\int_{t_1}^{t_2} \mathcal{L}(\boldsymbol{r}, \dot{\boldsymbol{r}})dt\right), \tag{2.6}$$

where $\mathcal{L}$ is the classical Lagrangian function of the system

$$L(\boldsymbol{r}, \dot{\boldsymbol{r}}) = \frac{1}{2}\dot{\boldsymbol{r}}^2 - \hat{V}(\boldsymbol{r}), \tag{2.7}$$

and the integral is over all paths that satisfy the endpoint conditions.

2.1.3 Examples of lattice models

$$\hat{\sigma}_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}_i \quad \hat{\sigma}_i^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}_i \quad \hat{\sigma}_i^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}_i \tag{2.8}$$

Transverse Field Ising model

$$\hat{H}_{\text{Ising}} = -J\sum_{\langle i,j\rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h\sum_i \sigma_i^x \tag{2.9}$$

Heisenberg model

$$\hat{H}_{\text{Heisenberg}} = -\frac{1}{2}\sum_{j=1}^{N}\left[J_x\hat{\sigma}_j^x\hat{\sigma}_{j+1}^x + J_y\hat{\sigma}_j^y\sigma_{j+1}^y + J_z\hat{\sigma}_j^z\hat{\sigma}_{j+1}^z + h\hat{\sigma}_j^z\right] \tag{2.10}$$

Bose-Hubbard model

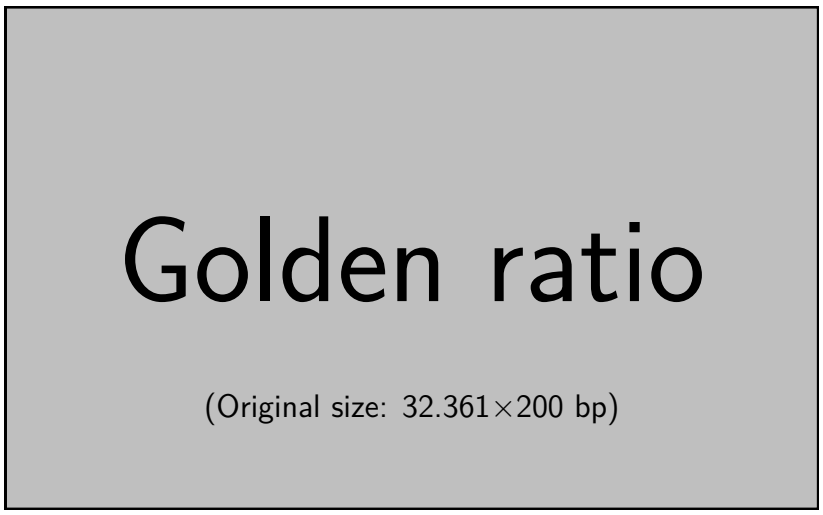
$$\hat{H}_{\text{BH}} = -t\sum_{\langle i,j\rangle} \hat{b}_i^\dagger \hat{b}_j + \frac{U}{2}\sum_i \hat{n}_i(\hat{n}_i - 1) - \mu\sum_i \hat{n}_i \tag{2.11}$$

2.2 Feynman-Kac: connecting Quantum Mechanics and Stochastic Processes

2.2.1 Stochastic processes

Introduce minimal necessary basics to understand the following section/discussions. This includes

- Random variables
- Markov chains
- Transition matrices
- Brownian walk, measures on the Brownian walk.



2.2.2 Feynman-Kac formula

The Feynman path integral formulation (2.6) was extensively used by physicists for decades, even in the absence of a formal mathematical formulation which is hard to define because of the difficulties with defining an appropriate measure on the path space. Kac [2] provided a rigorous formulation of the *real-valued* case of the Feynman path integral, and the resulting *Feynman-Kac* formula provides a bridge between *parabolic* partial differential equations and stochastic processes.

To illustrate the Feynman-Kac formula let us consider a single particle with Hamiltonian

$$\hat{H} = -\frac{\mathrm{d}^2}{\mathrm{d}x^2} + V(x) \tag{2.12}$$

and the Schrödinger equation in *imaginary time*, which is of the elliptic type,

$$\partial_t |\psi_t\rangle = -\hat{H}|\psi_t\rangle. \tag{2.13}$$

Its formal solution, the time propagation of an initial wave function $|\phi_0\rangle$ at $t = 0$, is written as

$$|\psi_t\rangle = e^{-\hat{H}t}|\psi_0\rangle. \tag{2.14}$$

From the spectral decomposition of the operator $e^{-\hat{H}t}$ in terms of eigenstates $|\phi_n\rangle$ and eigen-energies E_n of the Hamiltonian $\hat{H}$

$$e^{-\hat{H}t} = \sum_n e^{-E_n t}|\phi_n\rangle\langle\phi_n|, \tag{2.15}$$

it follows that the term corresponding to the ground state of the system $|\phi_0\rangle$ decays the slowest. Thus starting in some initial state and propagating for a long imaginary time it leads into the ground state with the decay rate giving the ground state energy as

$$\lim_{t\to\infty} |\psi_t\rangle \propto e^{-E_0 t}|\phi_0\rangle, \tag{2.16}$$

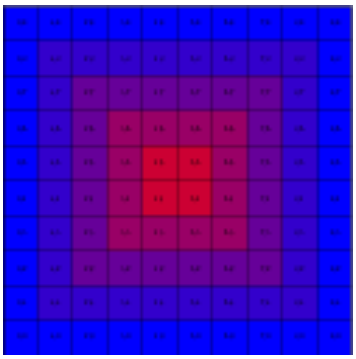
where E_0 is the ground state energy and $|\phi_0\rangle$ is the corresponding state. Kac noticed that the kinetic term of the Lagrangian in (2.6) could be interpreted as a measure on Brownian walks, and a solution to the imaginary time Schrödinger equation can be written as

$$\psi(x,t) = \mathop{\mathbb{E}}_{X\sim \text{ Brownian with } X_t=x}\left[\exp\left(-\int_0^t V(X_\tau,\tau)\,\mathrm{d}\tau\right)\psi(X_0,0)\right], \tag{2.17}$$

where only the **endpoint** at time t of the Brownian process fixed, whereas the starting point at time $t = 0$ is not, $\psi(x,0)$

2.2.3 Stoquastic Hamiltonians and Lattice-model representations

general Stochquastic hamiltonian -> Feynman kac for general process



Quantum Ising model

Heisenberg model

Bose-Hubbard model

2.3 Quantum Monte Carlo

Nekje povej, kaksen complexity imajo QMC problemi za fermione in bozone.

2.3.1 Overview

2.4 Machine Learning

2.4.1 Convolutional Neural Networks

2.4.2 Overview of ML approaches to the Quantum many-body problem

Chapter 3

Methodology

3.1 Numerical Methods

3.1.1 Discretizing the Feynman-Kac formula

3.1.2 Automatic differentiation

3.1.3 Gradient based optimisation

3.2 *Qptimal sampling*: optimal sampling in lattice models

Chapter 4

Results

4.1 Transverse Field Ising model

4.2 Heisenberg model

4.3 Bose-Hubbard model

Chapter 5

Conclusion

5.1 Direction for future work

5.2 Remarks

References

[1] Chriss, N. A. and Chriss, N. (1997). *Black Scholes and beyond: option pricing models*. McGraw-Hill.

[2] Kac, M. (1949). On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):1–13.

[3] Kingma, D. P. (2017). Variational inference & deep learning: A new synthesis.

[4] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[5] Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2020). Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62.

[6] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Appendix A

Jax ecosystem

Appendix B

Gradient estimation

In order to perform gradient descent on the ELBO objective, we need to be able to evaluate its gradients with respect to parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Taking the gradient w.r.t generative parameters $\boldsymbol{\theta}$ is straightforward, because we can change the order of the expectation operator and the gradient, leaving us with

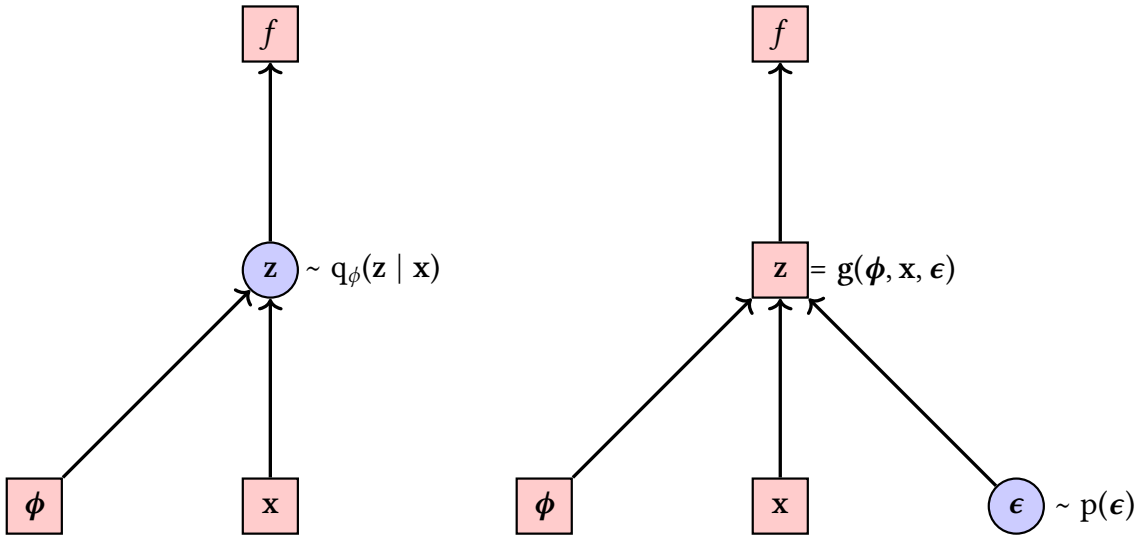
$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \right] \\ &\simeq \nabla_{\boldsymbol{\theta}} \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \right) \\ &= \nabla_{\boldsymbol{\theta}} \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right), \end{aligned} \tag{B.1}$$

where $\simeq$ denotes an unbiased estimator. This reversing of the order of operations is not possible when taking gradients w.r.t variational parameters $\boldsymbol{\phi}$ because the expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}$ is performed w.r.t the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. The gradient could be estimated with a vanilla Monte Carlo estimator, but it has very high variance and is not practical [4].

The problem of stochastic gradient estimation of an expectation of a function is a well studied problem that transcends machine learning and has a variety of applications [1, 6]. Different estimators differ in from and their properties, variance being one of the most important. In their review [5] Mohamed et al. categorise MC gradient estimators into three categories

1. *Score-function estimator*: The score function is a logarithm of a probability distribution w.r.t to distributional parameters. It can be used as a gradient estimator

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[f(\mathbf{x})] &= \nabla_{\boldsymbol{\theta}} \int p_{\boldsymbol{\theta}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})} \left[f(\mathbf{x}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \right]. \end{aligned} \tag{B.2}$$



(a) Original (b) Reparametrized

Fig. B.1 The reparametrization trick, adapted from [3]. The stochasticity of the $\mathbf{z}$ node is pushed out into a separate input to the same node, resulting in deterministic gradients w.r.t $\boldsymbol{\phi}$ through the node.

The score-function estimator is compatible with any cost function, it requires that the measure $p_{\boldsymbol{\theta}}(\mathbf{x})$ is differentiable and easy to sample. Very importantly it is applicable to both discrete and continuous distribution, but has a drawback of having high variance.

2. *Pathwise estimator*: Continuous distributions can be sampled either directly by generating samples from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ or indirectly, by sampling from a simpler base distribution $p(\boldsymbol{\epsilon})$ and transforming the variate through a deterministic path $g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$. Using this, it is possible to move the source of randomness in such a way that the objective is differentiable. In essence this approach pushes the parameters of the measure into the cost function which is then differentiated. The estimator is

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[f(\mathbf{x})] &= \nabla_{\boldsymbol{\theta}} \int p_{\boldsymbol{\theta}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= \nabla_{\boldsymbol{\theta}} \int p(\boldsymbol{\epsilon}) f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})) d\boldsymbol{\epsilon} \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}))]. \end{aligned} \tag{B.3}$$

This was the gradient estimator originally used in the VAE implementation [4] there named as the *reparametrization trick*, see also Figure B.1. In many cases the transformation paths are so simple they can be implemented in one line of code,

referred to as *one-liners*. The pathwise-estimator can only be used on differentiable cost functions, but is easy to implement and crucially has lower variance than the score-function estimator.

3. *Measure-valued gradient estimator*, which exploits the properties of signed-measures, is beyond the scope of this report.

Unbiased gradient estimation makes training on the ELBO objective possible. During training the VAE learns a mapping between a complicated observed space of $\mathbf{x}$ and the latent space of $\mathbf{z}$, which is usually defined to be relatively simple. The *decoder* network or the *generative* part of the VAE $p_{\theta}(\mathbf{x}|\mathbf{z})$ learns to map from the latent space to the data space, while the *inference* or *encoder* network $q_{\phi}(\mathbf{z}|\mathbf{x})$ approximates the true posterior $p_{\theta}(\mathbf{z}|\mathbf{x})$ of this process, see Figure.

Appendix C

Probability distributions