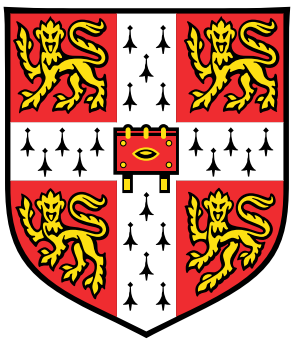
Optimal Importance Sampling in Quantum Monte Carlo for Lattice Models



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

August 2021

To my family.

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
August 2021

Acknowledgements

First of all, I would like to thank Zala for her love, support, and for helping me get through the roughest parts of my MPhil journey. I would also like to thank my family: grandparents for their words of wisdom, parents for showing nothing but support for my ambitions, and brother for being the most reliable friend.

Secondly, I would like to acknowledge everyone that helped make studying at Cambridge a reality for me. My primary and secondary school teachers who inspired me, as well as the faculty at FMF for providing the foundation of my education. I am most grateful to dr. Kosec for allowing me to work with him and for everything I learned from him and dr. Slak at IJS. I want to thank dr. Kosec and prof. Prosen, for believing in me and writing my recommendation letters.

Finally, I thank prof. Austen Lamacraft, not just for his supervision, guidance, and advice, but for all the coffees and discussions of Physics at St. Johns, which were exactly how I imagined Cambridge would be.

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

Nomenclature

Other Symbols

$\mathbb{E}$	Expectation
"$\mathcal{F}$"	σ-field (algebra)
$\mathbb{F}$	Filtration
D_{KL}	Kullback-Liebler divergence
$\mathbb{P}$	Measure
$\mathcal{N}$	The Gaussian distribution
$\{X\}$	Random variable
Γ	Transition rate matrix
$\mathbb{R}$	The set of real numbers
S	State space of a Markov process
$\{X_t\}$	Stochastic process
Cov	Covariance
Var	Variance
W_t	Wiener process, mathematical Brownian motion

Acronyms / Abbreviations

cdf	Cumulative density function
CNN	Convolutional Neural Network

DTMC Discrete time Markov Chain

DL Deep Learning

DMC Diffusion Quantum Monte Carlo

DTMC Discrete time Markov Chain

e.g. Exempli gratia ("for the sake of an example")

FP Fokker-Planck

GAN General Adversarial Network

GFMC Green's function Quantum Monte Carlo

i.e. Id est ("it is")

i.i.d Independent and identically distributed

MC Monte Carlo

ML Machine Learning

NN Neural Network

p.b.c Periodic boundary condition

pdf Probability density function

QMC Quantum Monte Carlo

RN Radon-Nikodym

SDE Stochastic Differential Equations

s.p. Stochastic process

s.t. Such that

VAE Variational Autoencoder

VMC Variational Quantum Monte Carlo

w.r.t With respect to

Chapter 1

Introduction

1.1 Thesis Contributions

1.2 Thesis Structure

Chapter 2

On the quantum many-body problem

This chapter discusses the quantum many-body problem and numerical approaches to its solution. We begin by introducing lattice models and their significance, before briefly overviewing the Schrödinger equation and Feynman path integral formulations of the problem. We then turn our attention towards solution procedures, providing a primer on popular methods with an emphasis on stochastic, Monte Carlo, approaches. Finally, we highlight recent usage of machine learning methods in this field.

2.1 Lattice models

TODO Add basic boson fermion distinction somewhere

2.1.1 The Schrödinger equation and the Feynman path integral

The dynamics of a quantum mechanical system are described with the Schrödinger equation

$$i\hbar \frac{\mathrm{d}}{\mathrm{d}t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle, \tag{2.1}$$

a linear partial differential equation. The state of the quantum system $|\Psi\rangle$ is a vector in a separable Hilbert space $\mathcal{H}$, and the square of its absolute value, e.g. $|\Psi(x,t)|^2$, at each point is interpreted as a probability density function (pdf). The Hamiltonian operator is the sum of kinetic and potential energies $\hat{H} = \hat{T} + \hat{V}$ of the system. Throughout this thesis we will be interested only in the ground state of the system, this is the state with lowest energy and it is the state we find the system in at zero-temperature. Instead of using the time-dependent

formulation in eq. (2.1), we use the *stationary* Schrödinger equation

$$\hat{\mathrm{H}}|\Psi\rangle = E|\Psi\rangle, \tag{2.2}$$

an eigenvalue equation, with the lowest energy E_0 corresponding to the ground state $|\Psi_0\rangle$. From this point onward, we use Hartree atomic units $m_e = e = \hbar = a_0 = \mathrm{E}_h = k_e = 1$.

Alternatively to the Schrödinger equation one can use an integral Green's function representation to express the wavefunction Ψ at some future time t given initial condition $\Psi(x', t')$ as

$$\Psi(x,t) = \int \mathcal{K}\left(x,t;x',t'\right)\Psi\left(x',t'\right)\mathrm{d}x'. \tag{2.3}$$

The *propagator* $\mathcal{K}\left(x,t;x',t'\right)$ is the kernel of the Schrödinger equation

$$\left(i\frac{\partial}{\partial t} - H_x\right)\mathcal{K}\left(x,t;x',t'\right) = i\delta\left(x-x'\right)\delta\left(t-t'\right). \tag{2.4}$$

It is related to the fundamental solution or Green's function as

$$\mathcal{G}\left(x,t;x',t'\right) = \frac{1}{i}\Theta\left(t-t'\right)\mathcal{K}\left(x,t;x',t'\right), \tag{2.5}$$

where Θ is the Heaviside function and δ is the Dirac delta. The same propagator can also be expressed using the Feynman path integral

$$\mathcal{K}\left(x,t;x',t'\right) = \int_{\substack{q(t)=x\\ q(t')=x'}} \exp\left(i\int_{t'}^{t}\mathcal{L}(q,\dot{q},t)dt\right)\mathcal{D}[q(t)], \tag{2.6}$$

where $\mathcal{L}$ is the classical Lagrangian function of the system, and the path integral is over all paths that satisfy the endpoint conditions $q(t) = x, q(t') = x'$. This formulation of QM is closely related to the approaches we will develop in the later chapters.

2.1.2 Examples of lattice models

TODO, mention curse of dimensionality

$$\hat{\sigma}_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}_i \quad \hat{\sigma}_i^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}_i \quad \hat{\sigma}_i^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}_i \tag{2.7}$$

Transverse-field Field Ising model

TODO

$$\hat{H}_{\text{Ising}} = -J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_i \sigma_i^x \tag{2.8}$$

Heisenberg model

TODO

$$\hat{H}_{\text{Heisenberg}} = -\frac{1}{2} \sum_{j=1}^{N} \left[J_x \hat{\sigma}_j^x \hat{\sigma}_{j+1}^x + J_y \hat{\sigma}_j^y \sigma_{j+1}^y + J_z \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z + h \hat{\sigma}_j^z \right] \tag{2.9}$$

Bose-Hubbard model

TODO

$$\hat{H}_{\text{BH}} = -t \sum_{\langle i,j \rangle} \hat{b}_i^\dagger \hat{b}_j + \frac{U}{2} \sum_i \hat{n}_i (\hat{n}_i - 1) - \mu \sum_i \hat{n}_i \tag{2.10}$$

2.2 Approaches to the quantum many-body problem

The quantum many-body problem, which amounts to solving the $3N$-dimensional Schrödinger equation, underpins a large part of quantum chemistry, condensed matter physics and materials science. The problem is notoriously hard to solve and very few systems with analytical solutions exist, most of them constrained in some artificial way such that they lend themselves to mathematical analysis. Examples include the Hookium atom, an analogue of helium where two electrons interact with the nucleus via a Hookean potential, Spherium, a system of two electrons confined to the surface of a sphere, and the Luttinger liquid of fermions in a one-dimensional conductor. Great efforts have been made in the nearly 100 years since the conception of the Schrödinger equation, in developing both analytical and numerical techniques to produce insights into quantum systems. Perhaps the most impactful was the development of various approximate methods that solve the many-body problem with our limited computational resources. While there is ongoing work on quantum simulators and computers that could greatly speed-up solving quantum problems [27, 18], we here discuss methods one can use with a classical computer. The commonality of all mentioned methods is that they try to tame the exponential growth of the underlying Hilbert space w.r.t the system size, they differ in how they achieve this. The three most common assumptions/simplifications to the many-body problem employed in condensed matter and quantum chemistry are the Born-Oppenheimer approximation, that electronic motion is instantaneous compared to the nuclear motion, the use of chemical basis sets, which transforms the PDE into an algebraic problem, and usually neglecting relativistic effects.

Hartree-Fock

One of the most common approaches to the many-body problem is to replace the original interacting many-body problem with a set of independent-particle problems with effective potential. **Hartree-Fock** (HF) approaches solve an auxiliary system of independent electrons in a self-consistent field and assume that the wave function (for fermions) can be represented as a single Slater determinant. The HF method does not include electron correlation, which makes it a good approximation only in systems where correlation contributions are small.

Post-Hartree-Fock methods

Post-HF methods, such as Coupled Cluster, Configuration interaction and Møller-Plesset theory include correlation by considering a linear combination of Slater determinants. They can be extremely accurate but come at a high computational cost.

Density Funcitonal Theory

Alternatively **Density Functional Theory** (DFT) reformulates the many-body electron problem in terms of the 3-dimensional electron density $n(\mathbf{r})$, which is found by minimising the total energy functional $E[n(\mathbf{r})]$ [35]. In practice this is done by solving the Kohn-Sham auxiliary system. DFT is in theory exact, however only if the true energy functional $E[n(\mathbf{r})]$ is known. As this is not the case, much research has been done in constructing different energy functionals with varying degrees of accuracy, starting with local functionals e.g. LSDA and continuing towards more heavily parameterised, non-local formulations. DFT provides a good trade-off between accuracy and computation time, it is used extensively for simulating large systems as linear scaling variants of DFT exist [68].

Dynamical Mean Field Theory

DMFT [34] is a framework that is specialised in solving strongly correlated systems. It is intuitively similar to Weiss Mean Field Theory in classical statistical physics. The main idea is to map an intractable lattice problem into an impurity model in an effective medium, a many-body local problem which can be solved with any standard approach (QMC, DFT, exact diagonalisation, etc.). This mapping between lattice and impurity model is exact, the approximation comes in neglecting spatial fluctuations of the lattice self-energy Σ, the contribution to energy due to particle interaction with medium. DMFT assumes that Σ is a function of frequency and not momentum $\Sigma(k, \omega) = \Sigma(\omega)$, which only holds in the infinite coordination case. Time fluctuations are taken into account, i.e. the effective medium is not static in DMFT, which is an advantage over other static mean field theories.

Density Matrix Renormalization group

DMRG [80] is considered the state of the art method for solving one-dimensional lattice problems, it has been widely adopted in condensed matter physics, first used to solve the system of a spin-0 particle in a box. It is an iterative method based on the renormalization group [81], and uses matrix product states as the variational ansatz. The method has also been extended for time evolution of systems [26], and higher dimensions [78].

2.2.1 Stochastic methods - Quantum Monte Carlo

Quantum Monte Carlo is a class of methods that uses statistical sampling to directly deal with high-dimensional integration that arises from working with the many-body wave function. QMC methods are among the most accurate achieving chemical accuracy for smaller systems [28], and can in principle achieve any degree of statistical precision sought. A large ecosystem of QMC methods exists, and they have been adapted to study almost any quantum system imaginable, from discrete to continuous state space, fermionic and bosonic systems, as well as both finite and zero temperatures. Even though QMC methods are not computationally the cheapest, they have reasonable storage requirements as the wave function does not need to be stored directly. Moreover, the high computational cost of QMC methods can be aided by paralellisation and use of hardware acceleration, as the core calculation in is repetitive and usually involves generating (pseudo)-random numbers, performing a simple calculation and in the end averaging over the results.

Variational quantum Monte Carlo (VMC)

The most straightforward QMC approach is based on the variational principle, which provides a clear path towards a solution to the ground state problem. Simply use a *trial wave function* Ψ_T to parameterise the ground state and optimise the parameters of Ψ_T to reach the lowest-energy state. This lowest variational state should capture the behaviour of the ground state if the ansatz is expressive enough. The flexibility of easily defining the trial wave function for a variety of different problems and the ease of its evaluation is a clear advantage over methods described in the previous section. Moreover, given that the variational wave function should encapsulate the main aspects of the system studied it provides intuition into the system itself. Development of trial functions has played a key role in the applicability of VMC, famous examples of trial wave functions include the Slater-Jastrow and Backflow wave functions. The drawback of VMC is that the variational wave function might contain a bias that cannot be avoided through optimisation of the parameters alone, see Fig. 2.1. VMC necessarily contains two steps, first is the estimation of the variational energy and second is the optimisation of the parameters. Any expectation of an operator $\hat{O}$ can be expressed in terms of the trial wave function as

$$\langle \hat{O} \rangle = \frac{\langle \Psi_T \mid \hat{O} \mid \Psi_T \rangle}{\langle \Psi_T \mid \Psi_T \rangle} = \frac{\sum_x \langle \Psi_T \mid x \rangle \langle x \mid \hat{O} \mid \Psi_T \rangle}{\sum_x \langle \Psi_T \mid x \rangle \langle x \mid \Psi_T \rangle}, \tag{2.11}$$

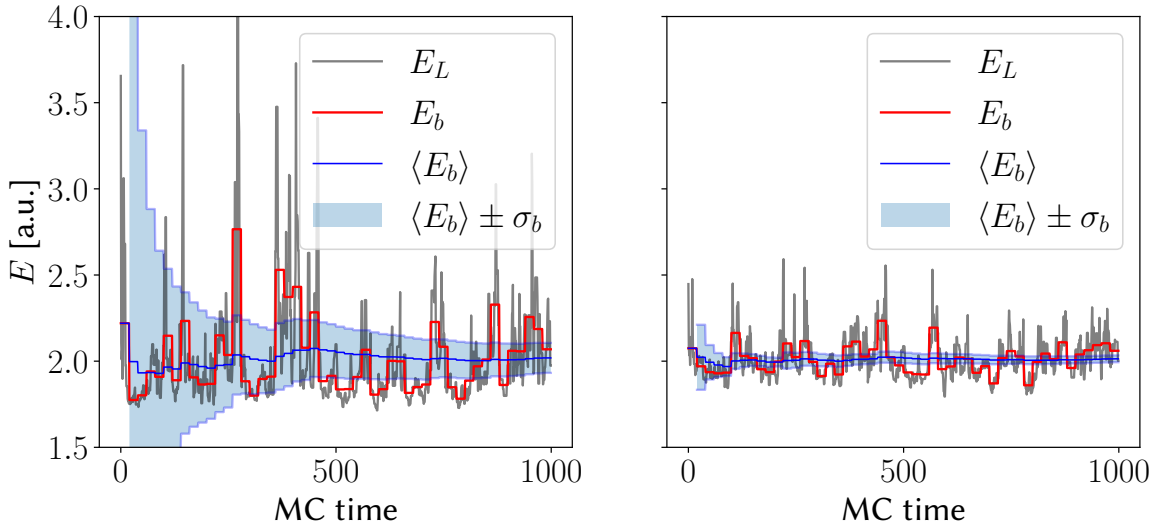


Fig. 2.1 **Ansatz quality in VMC**. Appropriateness of the variational wave function limits the quality of VMC, a poor choice of ansatz results in typical spikes in local energy and biased result (**left**), as well as slower convergence than a good trial wave function (**right**). Figures show the local energy E_L, reblocked average energy $\langle E_b \rangle$ and variance σ_b of a VMC simulation of Hookium.

where $|\, x\rangle$ are orthogonal and normal states of the Hilbert space. If we rewrite the above expression as

$$\langle \hat{O} \rangle = \frac{\sum_x |\, \Psi_T(x) \,|^2 \, \hat{O}_L(x)}{\sum_x |\, \Psi_T(x) \,|^2}, \tag{2.12}$$

with $\hat{O}_L$ being the *local operator*

$$\hat{O}_L(x) = \frac{\langle x \,|\, \hat{O} \,|\, \Psi_T \rangle}{\langle x \,|\, \Psi_T \rangle}, \tag{2.13}$$

we can interpret $|\Psi(x)|^2 / \sum_x |\Psi(x)|^2$ as a probability. Meaning that eq. (2.12) can be estimated as an average of the local operator $\hat{O}_L$

$$\langle \hat{O} \rangle \approx \frac{1}{M} \sum_{m=1}^{M} \hat{O}_L(x_m), \tag{2.14}$$

sampled from this probability distribution. The sampling can be done using Markov Chain Monte Carlo (MCMC). The second step of the procedure is variational optimisation of the trial wave function, where the optimal parameters of the approximation are found by minimising the *cost function*. The straightforward choice of the variational energy E_V as a cost function turns out to be inferior to minimizing the *variance* of the energy σ_E [28]. This

is because σ_E obeys the *zero-variance* property, meaning that if Ψ_T is an exact eigenvalue of the Hamiltonian

$$\hat{H}|\Psi_T\rangle = E_V|\Psi_T\rangle, \tag{2.15}$$

then the local energy E_L is constant and equal to E_V

$$E_L(x) = \Psi_T(x)^{-1}\hat{H}\Psi_T(x) = \Psi_T(x)^{-1}E_V\Psi_T(x) = E_V, \tag{2.16}$$

irrespective of the sampled configuration x and hence has zero variance. The zero-variance property has important consequences for numerical stability of optimisation, it means that energy variance minima are robust to finite sampling. Minimizing the variance of energy drives the trial wave function towards eigenstates of the Hamiltonian. Moreover, the statistical error of any expectation value $\langle\hat{O}\rangle$ is proportional to the variance of $\hat{O}$, making low variance doubly desirable. There are several approaches to updating the parameters each iteration, gradient descent, stochastic reconfiguration [69], and the linear method [54] are just a few examples. It is crucial that the methods are robust to statistical noise and converge quickly as the MC step can be expensive to perform. Moreover they are only as good as the estimates of the energy (variance) gradients w.r.t the parameters.

The first application of VMC was to the ground state ^{4}He [50] and it was later extended for studying many-body fermionic systems [17]. Time-dependant variants of this method exist [8] and VMC has been used to study non-equilibrium properties of bosonic [14, 13], and fermionic [38] systems.

Projector QMC (PMC) techniques

PMC is a class of QMC methods which are in essence nothing more than stochastic implementations of the power method to obtain the dominant eigenvector of a matrix or a kernel function [31]. Their distinct advantage over VMC is that they are not constrained by our parametrisation of the trial wave function, as they can describe arbitrary probability distributions. PMC methods are based on the imaginary Schrödinger equation

$$\partial_t|\Psi_t\rangle = -\hat{H}|\Psi_t\rangle. \tag{2.17}$$

Its formal solution, the time propagation of an initial wave function $|\Psi_0\rangle$ at $t = 0$, is written as

$$|\Psi_t\rangle = e^{-\hat{H}t}|\Psi_0\rangle. \tag{2.18}$$

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 $\hat{H}$

$$e^{-\hat{H}t} = \sum_n e^{-E_n t} |\Phi_n\rangle\langle\Phi_n|, \tag{2.19}$$

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 E_0 as

$$\lim_{t\to\infty} |\Psi_t\rangle \propto e^{-E_0 t} |\Phi_0\rangle, \tag{2.20}$$

where $|\Phi_0\rangle$ is the corresponding state of E_0. This of course holds if the eigenstates of $\hat{H}$ are all positive, which can be achieved by shifting the potential by a constant energy E_c, which doesn't change the ground state wave function. The basic step of a PMC simulation is the projection step, where an existing ensemble of configurations is projected into a new one, this projection $\hat{P}$ is done in such a way that eq. (2.20) is satisfied

$$|\Phi_0\rangle = \lim_{n\to\infty} \hat{P}^n |\Psi_0\rangle. \tag{2.21}$$

Flavours of PMC differ in the choice of $\hat{P}$, the most popular Diffusion Monte Carlo (DMC) [28, 61] works with the time-dependent Green's function $G(x', t'; x, t)$ of eq. (2.17)

$$\Psi(x,t) = \int \mathcal{G}\left(x,t;x',t'\right) \Psi\left(x',t'\right) \mathrm{d}x', \tag{2.22}$$

while Green's function MC (GFMC) [41, 42] uses the time integrated version of the Green's function

$$\Psi^{(n+1)}(x) = E \int \mathcal{G}\left(x,x'\right) \Psi^{(n)}\left(x'\right) \mathrm{d}x'. \tag{2.23}$$

Both formulations are exact, but need some additional approximations to be made practical, as Green's functions are not known for a general system. In DMC the Green's function

$$\mathcal{G}(x', t'; x, t) = \langle x \mid e^{-(t-t')[\hat{T}+\hat{V}-E_c]} \mid x' \rangle, \tag{2.24}$$

is approximated for short times $\tau = t - t'$ using Trotter-Suzuki formula

$$\mathcal{G}(x' \longrightarrow x; \tau) = \underbrace{(2\pi\tau)^{-3N/2} e^{-\frac{(x-x')^2}{2\tau}}}_{\text{ordinary diffusion}} \cdot \underbrace{e^{-\tau\left[V(\mathbf{R})+V(\mathbf{R}')-2E_c\right]/2}}_{\text{reweighting} \equiv \text{birth/death}} + \mathcal{O}(\tau^3), \tag{2.25}$$

where the kinetic term is recognised to be ordinary diffusion. In practice eq. (2.25) is implemented as a simulation of a diffusion process, but instead of weighting the paths of the walkers, the potential contribution to $\mathcal{G}$ is interpreted as a probability of a walker to either branch or die, which is numerically more stable. This stochastic process converges to the ground state for sufficiently long times, see Fig. 2.2. Reptation quantum Monte Carlo [61] (RMC) is an alternative formulation which only uses a single walker, and instead of branching and dying the MC moves mutate the path of that single walker. Using a

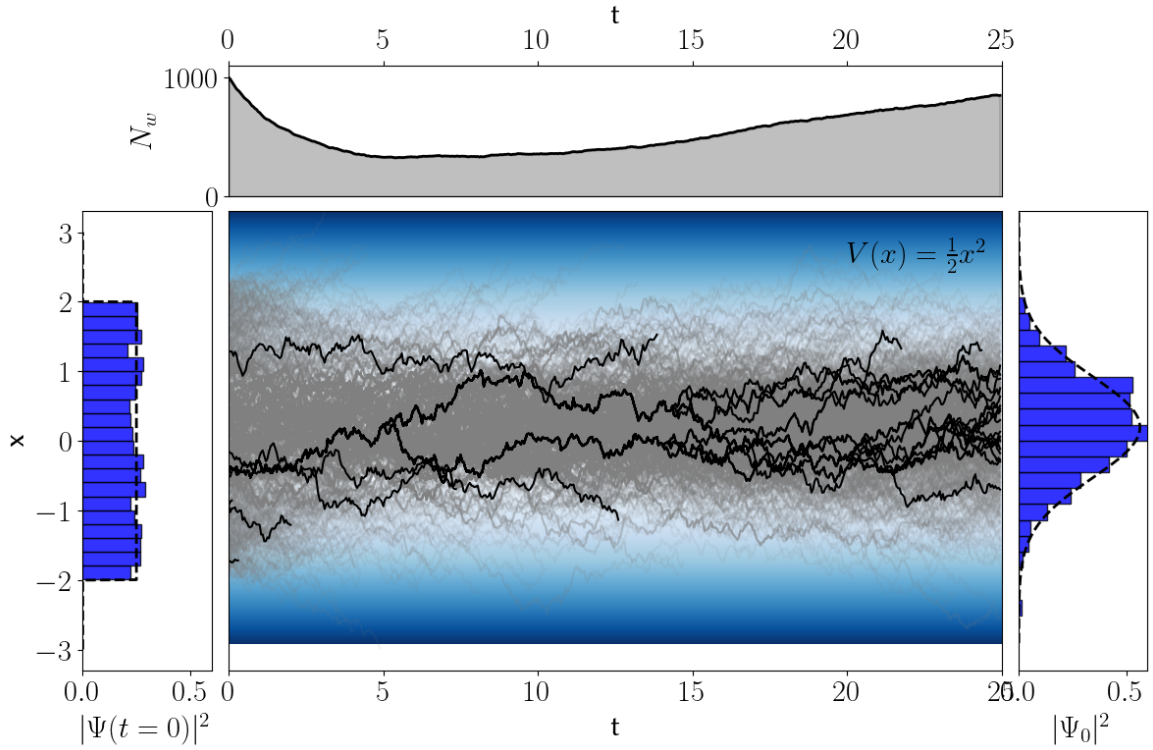


Fig. 2.2 **Diffusion Monte Carlo simulation of harmonic oscillator**, starting with N_w = 1000 walkers, τ = 0.05, E_c = 0.25 and uniformly sampling their initial positions from $(-2, 2)$ (**left**). The number of walkers at each step decreases rapidly before slowly increasing (**top**) the number of walkers is controlled by adjusting E_c. Walker paths, with a few highlighted in black to emphasise birth/death process (**middle**), diffuse into the approximate ground state of the HO $u_0(x) = \frac{1}{\pi^{\frac{1}{4}}} e^{-\frac{1}{2}x^2}$ (**right**).

trial wave function Ψ_T as a guiding function for importance sampling is an important improvement over vanilla DMC. This introduces a *drift* into the diffusion process, which leads the walkers into regions of large values of Ψ_T and greatly improves the statistical efficiency of the method. The guiding wave function is usually obtained by means of VMC. So far we have conveniently assumed that the wave function is positive everywhere in the

domain, this not generally true, e.g. in fermionic systems, and poses a problem for PMC methods.

The sign problem

Projector Monte Carlo methods can only operate with positive distributions, and as such they fall apart when applied to fermionic or frustrated systems [31]. A straightforward modification to the sampling scheme allows us to sample from a mixed-sign distribution. We sample from the distribution normally when it is positive, but sample from its absolute value and change the sign of the observable, when it is negative. The issue with this approach is that the population of configurations is split between positive and negative regions, the averages over both are comparable in size and cancel out, leading to a large statistical error compared to the observable. We refer to the accompanying exponential decrease [31] in sampling efficiency with system size and temperature, **the sign problem**. Its general solution was shown to be NP-hard [77], and as it is believed that P ≠ NP, this implies that no *general* polynomial-time solutions exist. However, this does not mean that the problem cannot be avoided in special cases, the search for solutions is still an area of active research [37, 5, 3]. In practice the sign problem is remedied by the *fixed-node* [4] or *constrained-path* [83] approximation. Fixed-node imposes a boundary condition into the projection such that the projected state shares the nodal surface with the trial wave function. The projected state is now only exact when the nodal surface is exact.

2.3 Machine Learning and the quantum many-body problem

With recent growing interest in Machine Learning there came a wave of research that applies ML methods to the natural sciences. As it pertains to the quantum many-body problem, most of the work is focused on exploiting the expressive nature of ML models, such as Restricted Boltzmann Machines (RBM) [16] or Deep Neural Networks [12] (DNN), to efficiently represent quantum states. These approaches fall into the VMC framework, and have been used for lattice models [16], both fermionic [55] and bosonic [64]. Notably, special NN architectures have been used to achieve higher accuracies than coupled cluster calculations on a variety of atoms and small molecules [59, 71]. In lattice models Convolutional Neural Networks (CNN) have been shown to struggle to converge to nontrivial sign structures of frustrated systems, but modelling the phase and amplitude with separate networks helps in this regard [73].

The expressiveness of RBM has also been analysed in depth [15], and contrasted to Tensor Network States [20]. Very recently an application of the NN ansatz in DMC with fixed-node approximation [82] was used to improve earlier work results of the FermiNet [59].

Alternatively to above approaches, which all operate in the Schrödinger picture, reinforcement learning has been used to solve the many-body problem in the path integral representation [6, 29]. ML has also found place in mean field methods, perhaps most notably for learning the exchange and correlation functionals in DFT [24].

Chapter 3

Feynman-Kac: connecting Quantum Mechanics and Stochastic Processes

In this chapter we will provide a bridge between the quantum many-body problem discussed in the previous chapter and stochastic processes. This will entail introducing the Feynman-Kac formula and relating it to the Fokker-Planck equation and optimal control formulations of QM. Moreover, a probabilistic view of the cost function will lead us to proposals for loss functions that can be used to learn optimal transition rates and consequently sample the ground state.

The field of stochastic processes is a vast body of work, approached from different angles by mathematicians, physicists and engineers. A necessary consequence of this is that the literature ranges from extremely thorough and rigorous [62, 63] to more applied and intuitive [66]. For this reason, the mentioned discussion will be preceded by an overview of the mathematical notation, lemmas and results from stochastic processes and measure theory that underpin some core ideas of this thesis. To avoid including a whole textbook of material on measure and stochastic processes some concepts will not be rigorously defined, the text will point to relevant literature where this is the case.

3.1 Stochastic processes

3.1.1 Fundamentals

This brief, more formal, discussion of stochastic processes is based mostly upon classic texts [25, 62, 63] and borrows some intuitions from [66]. The most basic quantity that we will need is the **probability space**.

Definition 3.1.1 (Probability space)**.** *The probability space is a tuple* $(\Omega, \mathcal{F}, \mathbb{P})$*, where* Ω *is the* ***sample space****,* $\mathcal{F}$ *is a* σ*-****field****, and* $\mathbb{P}$ *is the* ***measure****.*

The sample space is simply the set of all possible outcomes. A canonical example would be the roll of a 6-sided dice, $\Omega = \{1,2,3,4,5,6\}$. Without measure $\mathbb{P}$, the tuple $(\Omega, \mathcal{F})$ is termed a **measurable space**.

Definition 3.1.2 (σ-field)**.** *A* σ*-field* $\mathcal{F}$ *on a set* Ω*, is a nonempty collection of subsets of* Ω *that includes* Ω *itself, is closed under complement, i.e. if* $A \in \mathcal{F}$ *then* $A^c \in \mathcal{F}$*, and is closed under countable unions,* $\cup_i A_i \in \mathcal{F}$ *if* $A_i \in \mathcal{F}$ *is a countable union of sets.*

The main utility of the σ-field to us is its use in defining measures. We want to be able to assign a non-negative real number to all subsets of Ω, as well as the size of the union of the disjoint sets to be the sum of their individual sizes. This is not always possible, a counterexample for the real line being Vitali sets. The collection $\mathcal{F}$, must thus only include *measurable* sets, which are precisely the ones that satisfy the constraints imposed by the σ-field.

Definition 3.1.3 (Measure)**.** *A non-negative countably additive set function* $\mu : \mathcal{F} \longrightarrow \mathbb{R}$ *that satisfies*

i) $\mu(A) \geq \mu(\emptyset) = 0$ *for all* $A \in \mathcal{F}$

ii) if $A_i \in \mathcal{F}$ *is a countable sequence of disjoint sets, then* $\mu(\cup_i A_i) = \sum_i \mu(A_i)$

is a ***measure****.*

If $\mu(\Omega) = 1$, then μ is a **probability measure** and will be denoted by $\mathbb{P}$. With this notion we are now able to define a random variable (r.v) and a stochastic process (s.p.).

Definition 3.1.4 (Random variable)**.** *A* ***random variable*** X *defined on* Ω *is a real-valued measurable function* $X(\omega)$*,* $X : \Omega \longrightarrow \mathbb{R}^d$*.*

For a function to be measurable, we require that its preimage X^{-1} is in the σ-field $\mathcal{F}$

$$X^{-1}(B) = \{\omega : X(\omega) \in B\} \in \mathcal{F}, \tag{3.1}$$

and that this holds for every Borel set B in the Borel σ-field [1] of $\mathbb{R}^d$, which is simply the smallest σ-field that contains all measurable sets in $\mathbb{R}^d$.

[1] For a proper definition of the Borell set see ch. 3 of [65].

A random variable X induces a probability measure μ on $\mathbb{R}^d$ called its **distribution**, this is done by setting $\mu(A) = P(X \in A)$ for Borel sets A. Moreover, the distribution is usually given in terms of a **distribution function** $F(x)$

$$F(x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \leq x\}) = \mathbb{P}(X \leq x), \tag{3.2}$$

and X is said to have a **density function** $f(x)$ if $F(x)$ can be written as

$$F(x) = \int_{-\infty}^{x} f(y)\mathrm{d}y. \tag{3.3}$$

In essence, the random variable provides a connection between the less familiar probability measure $\mathbb{P}$ and the cumulative distribution function (CDF).

3.1.2 Stochastic process

Definition 3.1.5 (Stochastic process)**.** *Given a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *and a measurable (state) space* $(E, \mathcal{E})$*, we define the collection* $\{X_t : t \in T\}$ *of set* T *indexed and* $(E, \mathcal{E})$ *valued random variables a* ***stochastic process****.*

By far the most common case for the index set T, is time $T = \mathbb{R}^+$. Such s.p's are called *temporal*, examples include the model of velocity of a Brownian particle under influence of friction, in Fig. 3.1, or the Black-Scholes model. Nevertheless, the index set is not limited to time, as is often the case with Gaussian Process regression [60]. In this thesis we will mostly deal with temporal s.p's of the kind that do not "see into the future". This notion is formalized using **filtrations**. A filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in T}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is just an increasing sequence or order of σ-fields

$$\mathcal{F}_s \subset \mathcal{F}_t \text{ if } 0 \leq s \leq t < \infty \tag{3.4}$$

The filtration associated to a process that records its "past behaviour" at each time is called the **natural filtration**.

Definition 3.1.6 (Adapted process)**.** *A process* $\{X_t\}$ *is said to be* ***adapted to the filtration*** $(\mathcal{F}_t)_{t\in T}$ *if the random variable* $X_t : \Omega \longrightarrow E$ *is* $\mathcal{F}_t$*-measurable function for each* $t \in T$*.*

A process that is *non-anticipating*, i.e. depends only on the past and present, is adapted to the filtration $(\mathcal{F}_t)_{t\in T}$.

Definition 3.1.7 (Brownian motion)**.** ***Brownian motion*** *or a non-anticipating* ***Wiener process*** *is a stochastic process* W_t*, with the following properties:*

i) $W_0 = 0$

ii) W_t *is almost surely continuous in* t

iii) W_t *has independent increments*

iv) $W_t - W_s \sim \mathcal{N}(0, t-s)$ *for* $0 \leq s \leq t$

A realisation of Brownian motion can be found in Fig. 3.1.

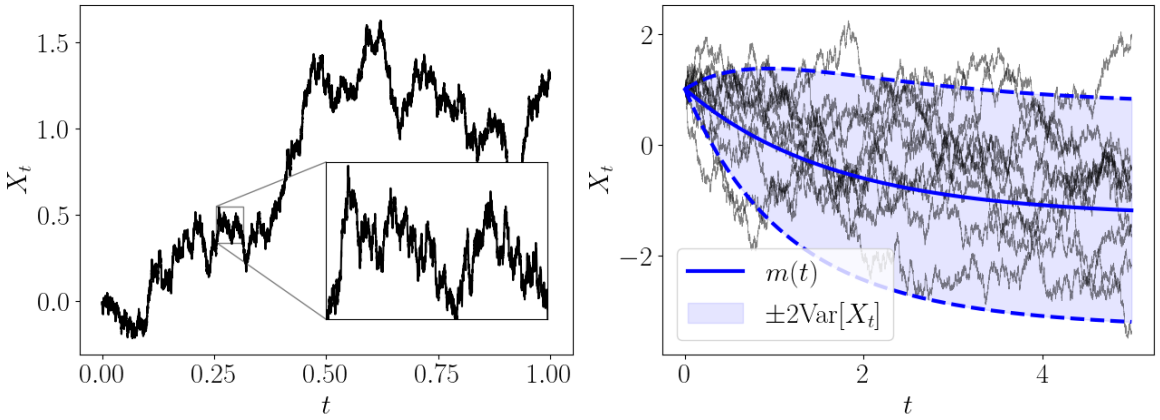


Fig. 3.1 **Brownian motion and Ornstein–Uhlenbeck process. left:** A single realisation of the Brownian process. **right:** Mean, variance and 10 samples of the Ohrnstein-Uhlenbeck process with $\theta = 0.6, \sigma = 1.1, X_0 = 1.0, \mu = -1.3$, integrated using Euler-Maruyama method.

3.1.3 Integrals

In order to proceed and define stochastic differential equations (SDE's) and the Radon-Nikodyn derivative, we must spend some time discussing various integrals we will use. In particular, alongside the usual Riemann integral, we will need three more types of integrals, which we will briefly describe without mathematical derivation. The simplest kind of integral we will introduce is the integral of a stochastic process

$$I = \int_0^t X_t \mathrm{d}t. \tag{3.5}$$

The simple appearance of the integral is deceiving as the integrand is a realisation of a $\mathcal{F}_t$-adapted stochastic process $\{X_t\} : \Omega \times T \longrightarrow \mathbb{R}^d$, meaning that I itself is a random variable. However, since each realisation of X_t is almost surely continuous, I can be expanded as a Riemann sum, which converges under mean-squared norm to I, so long as the mean

$\mathbf{m}(t) = \mathbb{E}[X_t]$ and covariance $\mathbf{k}(t,s) = \mathrm{Cov}(X_t, X_s)$ are continuous. In practice, computing the mean and covariance of I is usually enough to understand the resulting stochastic process. Importantly, integrals of continuous functions of s.p's $h(X_t)$, $h : \mathbb{R} \longrightarrow \mathbb{R}$ can be computed in a similar manner.

The second type of integrals we need to consider, are integrals with respect to a s.p, known as **Itô integrals**

$$Y_t = \int_0^t H_s \mathrm{d}X_s, \tag{3.6}$$

where both H_s and X_s are stochastic processes. The result integral Y_t is itself a stochastic process which resides in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, filtered by $(\mathcal{F}_t)_{t\in T}$. The integral can be formalised by putting slight constraints on what sort of stochastic processes X_s and H_t can be, expanding Y_t as a Riemann sum and proving convergence. Details of this procedure can be found in [63].

Finally we must define the **Lebesgue-Stieltjes integral** [32], which we need to properly define expectations of stochastic processes.

Definition 3.1.8 (Lebesgue-Stieltjes Integral)**.** *Given probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *and measurable function* $f : \Omega \longrightarrow \mathbb{R}$*, the* ***Lebesgue-Stieltjes integral***

$$I = \int_A f(x) \mathrm{d}\mathbb{P}(x), \tag{3.7}$$

is the Lebesgue integral[2] *with respect to measure* $\mathbb{P}$*,* $A \in \mathcal{F}$*.*

With it we can define expectations in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ as

$$\underset{\mathbb{P}}{\mathbb{E}}[f(x)] = \int_\Omega f(x) d\mathbb{P}(\boldsymbol{x}). \tag{3.8}$$

For a newcomer to stochastic processes this formulation may seem redundant, can we not just calculate expectations using a Riemann integral and the PDF? We can, and when the distribution $\mathbb{P}$ can be expressed in terms of the PDF (3.3), the Lebesgue integral can be interpreted in this way. However, stochastic processes need not admit a PDF, that is when the Lebesgue-Stieltjes integral is necessary.

3.1.4 Stochastic Differential Equations

In this thesis we will refer to a SDE as an informal notation of an Itô integral equation or **Itô process**.

[2] For proper definition of the Lebesgue integral see ch. 1 of [65].

Definition 3.1.9 (Itô process). *Given deterministic functions* $v : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^d$ *and* $\sigma : \mathbb{R}^d \times \mathbb{R}^+ \longrightarrow \mathbb{R}^{d\times d}$*, we define the* ***Itô process*** X_t *as the sum of Itô and Lebesgue integrals*

$$X_{t+s} - X_t = \int_t^{t+s} \sigma(X_u, u)\,\mathrm{d}W_u + \int_t^{t+s} v(X_u, u)\,\mathrm{d}u, \tag{3.9}$$

where W_t *is a Brownian motion.*

In simplified notation we can write (3.9) as

$$\mathrm{d}X_t = \sigma(X_t, t)\,\mathrm{d}W_t + v(X_t, t)\,\mathrm{d}t, \tag{3.10}$$

this is what we refer to as an SDE, an example can be found in Fig. 3.1. The functions v and σ, we will refer to as the **drift** and **volatility** of the Itô process respectively. The most intuitive interpretation of a SDE is in terms of the time evolution of the PDF of the process X_t. It is described by the **Fokker-Planck equation** [3]

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\sum_{i=1}^{N} \frac{\partial}{\partial x_i}\left[\mu_i(\mathbf{x}, t)p(\mathbf{x}, t)\right] + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2}{\partial x_i \partial x_j}\left[D_{ij}(\mathbf{x}, t)p(\mathbf{x}, t)\right], \tag{3.11}$$

where $p(\mathbb{x}, t)$ is the PDF of the solution to the SDE and $\mathbf{D} = \frac{1}{2}\boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ is the diffusion tensor. Finally we state without proof a consequence of Itô calculus, most commonly named **Itô's rule** or **lemma**, it is the stochastic calculus equivalent of the chain rule

Lemma 3.1.1 (Itô's lemma). *Given an Itô process* X_t *as given by* (3.9) *and a twice differentiable scalar function* $f(X_t, t)$*, then the Itô process for* f *is*

$$\mathrm{d}f = \frac{\partial f}{\partial t}\mathrm{d}t + \sum_i \frac{\partial f}{\partial x_i}\mathrm{d}x_i + \frac{1}{2}\sum_{ij} \frac{\partial^2}{\partial x_i \partial x_j} f\,\mathrm{d}x_i \mathrm{d}x_j, \tag{3.12}$$

when compared to ordinary calculus we notice an additional quadratic term.

3.1.5 Radon-Nikodym Derivative and Girsanov theorem

To perform importance sampling we perform a change of measure in an integral

$$\int_A f(x)\mathrm{d}\mathbb{P}(x) = \int_A f(x)\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x)\mathrm{d}\mathbb{Q}(x). \tag{3.13}$$

[3] Derivation in [66].

The function that measures the rate of change of density of one measure w.r.t another is the **Radon-Nikodym derivative** $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x)$.

Theorem 3.1.2 (Radon-Nikodym theorem)**.** *Let $\mathbb{P}$ and $\mathbb{Q}$ be probability measures on the measurable space $(\Omega, \mathcal{F})$, then the measurable function **Radon-Nikodym derivative** $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x) : \Omega \longrightarrow [0, \infty)$ exists and*

$$\mathbb{P}(A) = \int_A \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x)\mathrm{d}\mathbb{Q}(x), \tag{3.14}$$

for set $A \subseteq \mathcal{F}$.

The RN derivative will also be useful in defining the KL divergence between two **path measures**. Properly defining the path measure would bring a lot of notational overhead, it is enough to think of it as a measure on the **path space**, i.e all possible paths of a SDE, for rigour see [47]. Finally, we state the **Girsanov theorem** that is often used for transforming or removing drift functions of SDE, it is the RN derivative between an Itô process and one with $v = 0$ and $\sigma = 1$, i.e. Brownian motion.

Theorem 3.1.3 (Girsanov Theorem)**.** *Given Itô process*

$$\mathrm{d}X_t = \mathrm{d}W_t + v(X_t, t)\mathrm{d}t \quad \textit{and} \quad X_0 = 0 \tag{3.15}$$

and Brownian motion $\mathrm{d}Y_t = \mathrm{d}W_t$, the RN derivative of their respective path measures $\mathbb{P}$ and $\mathbb{P}_0$ is

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}_0} = \exp\left(-\frac{1}{2}\int_0^t |v(X_s, s)|^2 \,\mathrm{d}s + \int_0^t v(X_s, s)^\top \mathrm{d}W_s\right) \tag{3.16}$$

This *change in dynamics* as we will call it later is true in the sense, that expectations for an arbitrary functional $h(\cdot)$ of the path from 0 to t are

$$\mathop{\mathbb{E}}_{\mathbb{P}}[h(X_t)] = \mathop{\mathbb{E}}_{\mathbb{P}_0}\left[\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{P}_0} h(Y_t)\right]. \tag{3.17}$$

For a more general case and proof see [66].

3.1.6 Markov processes

We now shift our view to a special kind of s.p's, ones that satisfy the **Markov property** called **Markov processes** or **Markovian**. The property is sometimes referred to as *memorylessness*, as the future of a Markov process depends only on the present state. We can classify the processes based on the system's **state-space** S, which can be either discrete (countable) or continuous, and the **time indexing** of the system, either discrete-time

$\{X_n\}_{n\geq 0}$ or continuous-time $\{X_t\}_{t\geq 0}$. A taxonomy is given in Table 3.1. We will not specifically discuss Markov processes in continuous state-space, but it is important to note that any Itô process with time-homogenous drift $v = v(X_t)$ and volatility $\sigma = \sigma(X_t)$ is Markovian.

From now on we refer to Markov processes in countable state-space as **Markov chains**. We base our discussion on [62] and [56].

Table 3.1 **Taxonomy of Markov processes**

	Countable state-space	Continuous state-space
Discrete time	**index:** $\{X_n\}_{n\geq 0}$, $n \in \mathbb{Z}^+$ **state-space:** countable set I **define:** stochastic $\{P\}_{ij}$ **example:** DTMC	**index:** $\{X_n\}_{n\geq 0}$, $n \in \mathbb{Z}^+$ **state-space:** general state-space Ω **define:** stochastic kernel K **example:** Harris Chain
Continuous time	**index:** $\{X_t\}_{t\geq 0}$, $t \in \mathbb{R}^+ = [0,\infty)$ **state-space:** countable set I **define:** rate $\{\Gamma\}_{ij}$ equiv. to jump chain $\{J_n\}_{n\geq 0}$ and hold times $\{S_n\}_{n\geq 1}$. **example:** CTMC	**index:** $\{X_t\}_{t\geq 0}$, $t \in \mathbb{R}^+ = [0,\infty)$ **state-space** general state-space Ω **define:** stochastic kernel K **example:** Diffuson process

Discrete-time Markov Chains

The simplest and most common Markov process is a Markov chain in discrete time, an example of it can be found in Fig. 3.2. Its state-space is a countable set I and we call each $i \in I$ a **state**. We define a distribution λ in a familiar way

$$\lambda = \{\lambda_i : i \in I\} \quad \text{where} \quad \forall i : \ 0 \leq \lambda_i < \infty \quad \text{and} \quad \sum_{i\in I} \lambda_i = 1. \tag{3.18}$$

We can now set λ as a distribution of some random variable $X : \Omega \longrightarrow I$ as

$$\lambda_i = \mathbb{P}(X = i) = \mathbb{P}(\{\omega : X(\omega) = i\}), \tag{3.19}$$

where we are still working in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. A discrete-time Markov chain is defined in terms of its **transition matrix** $P = \{p_{ij} : i,j \in I\}$, which is a **stochastic matrix** meaning all of its rows $\{p_{ij} : j \in I\}$ are distributions.

Definition 3.1.10 (Discrete-time Markov chain). *A discrete time stochastic process* $\{X_n\}_{n\geq 0}$ *is a* ***discrete-time Markov chain*** *with initial distribution* λ *and transition matrix* P *if for* $i_1, \ldots, i_n + 1 \in I$ *and* $n \geq 0$

i) $\mathbb{P}(X_0 = i_1) = \lambda_{i_1}$

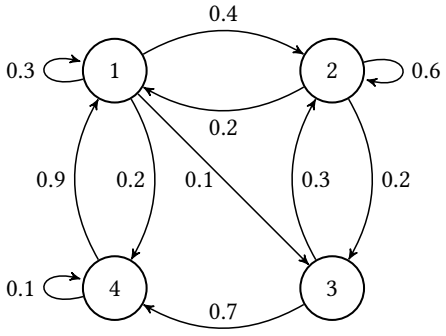
ii) $\mathbb{P}(X_{n+1} = i_{n+1} \mid X_0 = i_0, \ldots, X_n = i_n) = p_{i_n i_{n+1}}$

Rewriting the second condition above, it is clear that the Markov chain is without memory

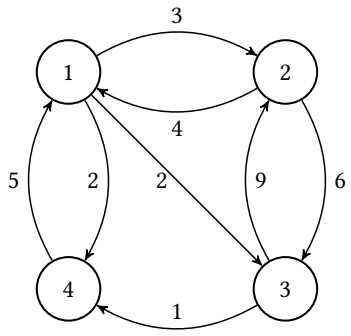
$$\mathbb{P}(X_{n+1} = i_{n+1} \mid X_0 = i_1, \ldots, X_n = i_n) = \mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n). \tag{3.20}$$

Intuitively we understand the discrete-time Markov chain as a system changing its state at discrete time intervals, each time choosing the next state according to the row of the Transition matrix corresponding to the current state.

a) Discrete-time Markov Chain



b) Continuous-time Markov Chain



$$P = \begin{pmatrix} 0.3 & 0.4 & 0.1 & 0.2 \\ 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0.3 & 0 & 0.7 \\ 0.1 & 0 & 0 & 0.1 \end{pmatrix}$$

$$\Gamma_{ii} = \sum_{j \neq i} -\Gamma_{ij} \quad \longrightarrow \quad \Gamma = \begin{pmatrix} -7 & 3 & 2 & 2 \\ 4 & -10 & 6 & 0 \\ 0 & 9 & 10 & 1 \\ 5 & 0 & 0 & -5 \end{pmatrix}$$

Fig. 3.2 **Discrete and continuous time Markov Chains. left:** Discrete-time Markov Chain defined by P. **right:** Continuous-time Markov Chain defined by Γ.

Continuous-time Markov Chains

Defining a Markov chain in continuous time is slightly trickier as describing the system with a stochastic matrix does no longer suffice because transition probabilities become zero when considering an infinitesimal time. Instead a continuous-time Markov Chain (CTMC) is characterised by a **rate matrix** or **infinitesimal generator matrix** Γ defined on the set I. A rate matrix has the following three properties

i) $0 \leq \Gamma_{ii} < \infty, \quad \forall i$

ii) $\Gamma_{ij} \geq 0, \quad \forall i \neq j$

iii) $\sum_{j \in I} \Gamma_{ij} = 0, \quad \forall i$

While the CTMC can be interpreted in a number of ways, we shall use the so called **jump chain** and **holding times** representation, see also Fig. 3.3. We can think of a CTMC as a

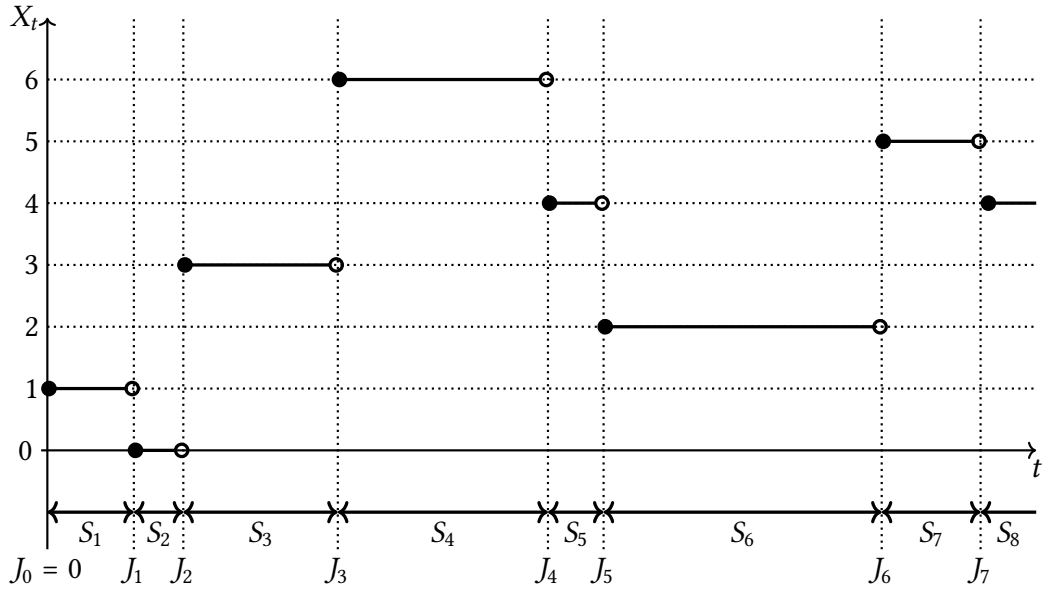


Fig. 3.3 **Jump chain and Holding times.** A discrete space Markov process $\{X_t\}_{t \geq 0}$ in continuous time. The holding times S_n are independent exponential random variables and the transition probabilities at jump times J_n are given with the jump matrix Π. Inspired by [56].

series of discrete jumps, where the system remains in each state for a certain holding time. This suggests that we can construct the CTMC from a discrete-time chain with stochastic matrix Π, which we will call the **jump matrix**, and a set of independent random variables $\{S_n\}$ which determine the holding times. We construct matrix Π by rescaling rows of Γ so they add up to one, putting a 0 on the diagonal

$$\begin{aligned} \Pi_{ij} &= \begin{cases} \Gamma_{ij}/\Gamma_{ii} & \text{if } j \neq i \text{ and } \Gamma_{ii} \neq 0 \\ 0 & \text{if } j \neq i \text{ and } \Gamma_{ii} = 0 \end{cases} \\ \Pi_{ii} &= \begin{cases} 0 & \text{if } \Gamma_{ii} \neq 0 \\ 1 & \text{if } \Gamma_{ii} = 0. \end{cases} \end{aligned} \tag{3.21}$$

In order for the process to possess the Markov property, the distribution of holding times $\{S_n\}$ must be exponential [56],

$$S_{n+1} \sim \text{Exp}(-\Gamma_{ii}(X_n)), \tag{3.22}$$

with exponential parameters being $-\Gamma_{ii}$ where i is the current state. Processes with different holding time distributions are called **semi-Markov**. The jump times $\{J_n\}$ are simply

$$J_n = S_1 + \ldots + S_n. \tag{3.23}$$

Definition 3.1.11 (Continuous-time Markov chain)**.** *A stochastic process $\{X_t\}_{t\geq 0}$ on set I is a **continuous-time Markov chain** if its jump chain $\{Y_n\}_{n\geq 0}$ is a discrete-time Markov chain and its holding times $\{S_n\}_{n\geq 1}$ are independent exponential random variables $S_n \sim \text{Exp}(-\Gamma_{ii}(X_n))$.*

An equivalent formulation is in terms of **competing exponentials**. Transitions $\Gamma_{j\to k}$ from j to k are defined as independent exponential random variables $\tau_{j\to k}$

$$\tau_{j\to k} \sim \text{Exp}(\Gamma_{jk}), \quad j \neq k \tag{3.24}$$

the next state is then chosen as

$$Y_{n+1} = \text{argmin}_k \, \tau_{j\to k}. \tag{3.25}$$

The chain $\{Y_n\}_{n\geq 0}$ along with times

$$S_n = \min_k \tau_{j\to k}, \tag{3.26}$$

gives the full description of the CTMC. With this formulation in mind we now interpret Γ_{ii} as the rate ob *leaving* current state and Γ_{ij} as the rate of *going* from i to j.

3.2 The Feynman-Kac formula

The Feynman path integral formulation introduced in chapter 2 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 [40] provided a rigorous formulation of the *real-valued* case of the Feynman path integral, and the resulting **Feynman-Kac formula** eq. (3.29) provides a bridge between *parabolic* partial differential equations and stochastic processes.

3.2.1 Feynman-Kac in continuous state space

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

$$\hat{H} = -\frac{\mathrm{d}^2}{\mathrm{d}x^2} + V(x) \tag{3.27}$$

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

$$\partial_t |\psi_t\rangle = -\hat{H}|\psi_t\rangle. \tag{3.28}$$

In close analogy to arguments presented in the DMC section 2.2.1, Kac noticed that the kinetic term of the Lagrangian in eq. (2.6) could be interpreted as a measure on Brownian walks, and a solution to the imaginary time Schrödinger equation can be written as

$$\psi(x,t) = \mathop{\mathbb{E}}_{X\sim \text{ Brownian with } X_t=x} \left[\exp\left(-\int_0^t V(X_\tau,\tau)\,\mathrm{d}\tau \right) \psi(X_0,0) \right], \tag{3.29}$$

where only the **endpoint** at time t of the Brownian process fixed, whereas the starting point at time $t = 0$ is not. $\psi(x,0)$ encodes the initial condition into this representation. When there is no external potential $V(x) = 0$, the Schrödinger equation in imaginary time is the diffusion equation and the Feynman-Kac solution is simply

$$\begin{aligned} \psi(x,t) &= \mathop{\mathbb{E}}_{X\sim \text{ Brownian with } X_t=x} [\psi(X_0,0)] \\ &= \frac{1}{\sqrt{2\pi t}} \int e^{-(x-x')^2/2t} \psi_0\left(x'\right) \mathrm{d}x' \end{aligned} \tag{3.30}$$

An illustration of the Feynman-Kac approach to the problem with no external potential $V(x)$ in 1D is depicted in Fig. 3.4. The role of the potential in the Feynman-Kac formula is to weight the Brownian paths, in turn defining the Feynman-Kac *path measure* $\mathbb{P}_{\mathrm{FK}}$ it is related to the Brownian measure $\mathbb{P}_0$ by the Radon-Nykodym derivative

$$\frac{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}{\mathrm{d}\mathbb{P}_0} = \mathcal{N} \exp\left(-\int V(X_t)\, dt \right), \tag{3.31}$$

where $\mathcal{N}$ is a normalizing constant. Intuitively we can understand the measure as assigning more weight to Brownian paths that spend more time in the attractive region ($V(x) < 0$) than in repulsive regions ($V(x) > 0$), this is illustrated in Fig. 3.5.

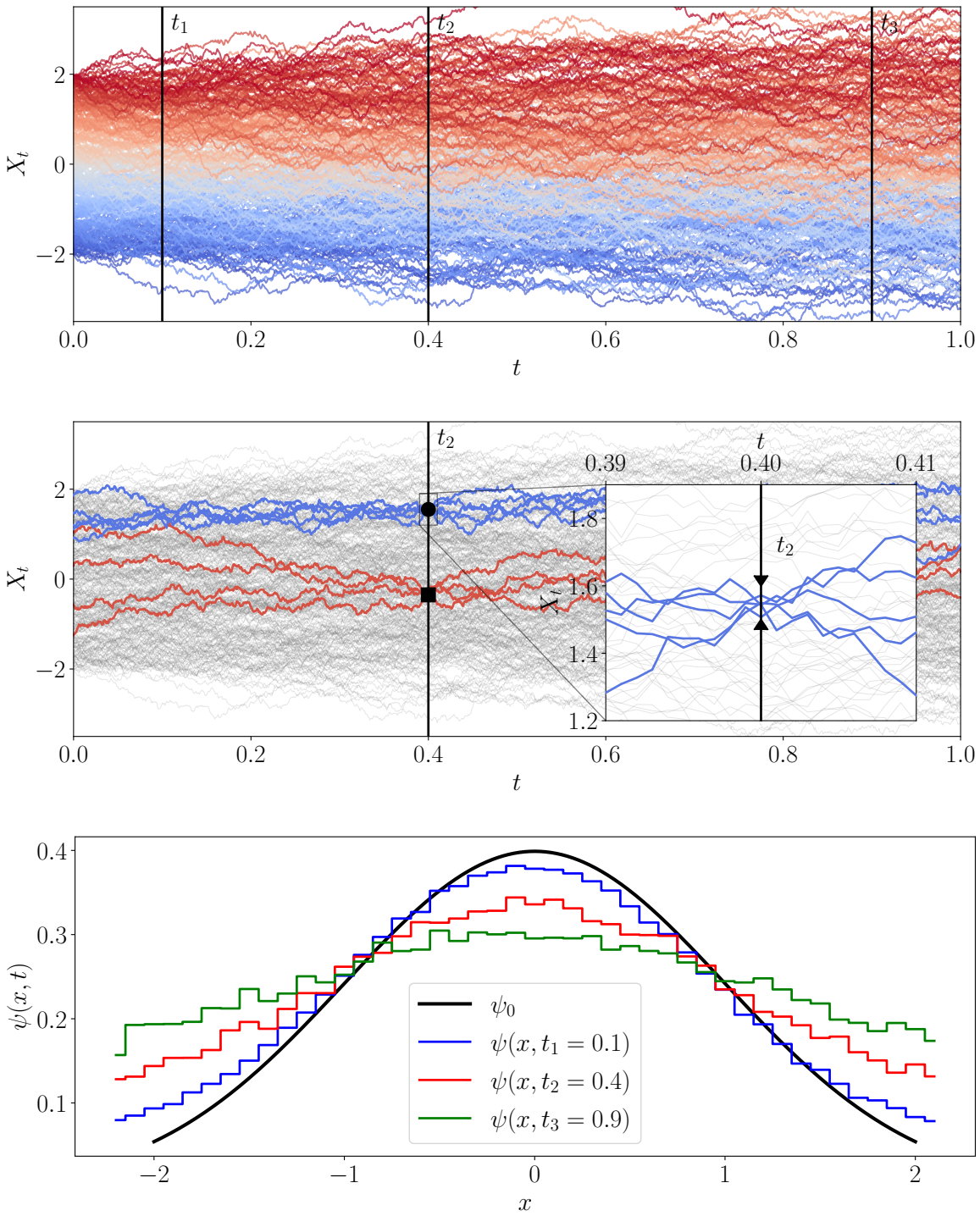


Fig. 3.4 **Feynman-Kac for a free particle in 1D. top:** $N = 400$ Brownian walks starting from different x_0, the color signifies initial position. In order to evaluate ψ between $x - \frac{\delta x}{2}$ and $x + \frac{\delta x}{2}$ at some time t we must first find Brownian paths that end there. **middle:** The paths that pass through at $x \in (1.5, 1.6)$ (blue) and through $x \in (-0.4, -0.3)$ (red) are colored, others are left in grey. **bottom:** Time evolution of the initial condition $\psi_0 = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$, by estimating $\mathbb{E}[\psi(X_0, 0)]$ from the filtered paths at each timestep.

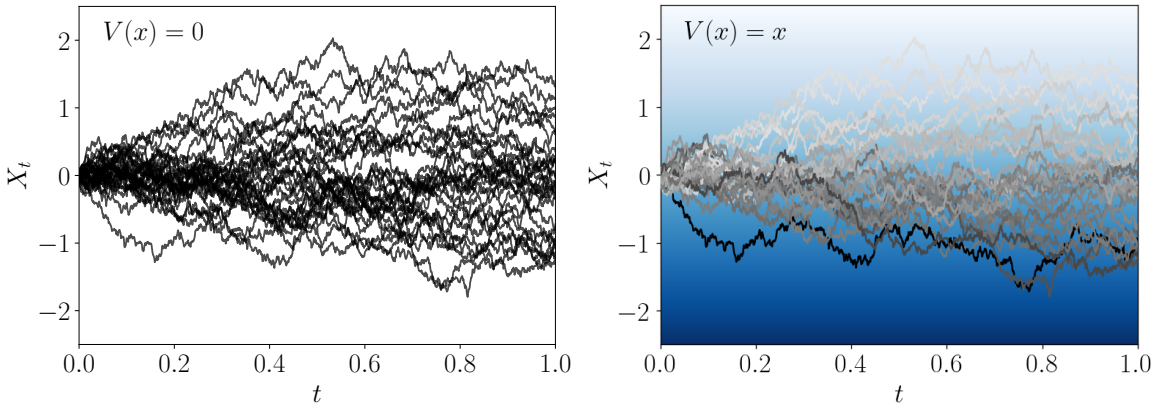


Fig. 3.5 **Feynman-Kac measure in a linear potential. left:** $N = 30$ Brownian paths. **right:** The paths colored (P(black) = 1, P(white) = 0) by their likelihood under the Feynman-Kac measure with $V(x) = x$.

Moreover, this new stochastic process is Markovian, meaning that a clear connection exists between the imaginary time Schrödinger equation and a SDE of form (3.10) with time-homogeneous σ and v. Indeed, in the continuous case we have discussed so far, the mapping between the Fokker-Planck equation (3.11) and the Schrödinger equation exists in the form of a similarity transform. Starting from the FP equation of a stochastic process with constant volatility $\sigma = 1$

$$\mathrm{d}X_t = \mathrm{d}W_t + v(X_t)\,\mathrm{d}t, \tag{3.32}$$

and drift $v(x) = -U'(x)$ given as a gradient of some potential function $U(x)$, the PDF $\rho(t,x)$ of the process is

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x}\left[\frac{\partial \rho}{\partial x} + U'(x)\rho\right]. \tag{3.33}$$

We can define the function

$$\psi(x,t) = \frac{\rho(x,t)}{\sqrt{\rho_0(x)}}, \tag{3.34}$$

with ρ_0 being the stationary distribution of the FP equation

$$\frac{\partial}{\partial x}\left[\frac{\partial \rho}{\partial x} + U'(x)\rho\right] = 0 \quad \longrightarrow \quad \rho_0(x) \propto \exp(-U(x)), \tag{3.35}$$

which satisfies the imaginary time Schrödinger equation (3.28) with the Hamiltonian

$$\hat{H} = -\frac{\partial^2}{\partial x^2} \overbrace{-\frac{U''}{2} + \frac{U'^2}{4}}^{\equiv V(x)}. \tag{3.36}$$

The ground state of this Hamiltonian has zero energy and is

$$\psi_0(x) = \sqrt{\rho_0(x)}. \tag{3.37}$$

In other words, the quantum ground state probability distribution $|\psi_0|^2$ is the same as classical stationary distribution ρ_0 of the stochastic process X_t in the literature referred to as the *Nelson's ground state process* [53, 2]. This connection is one that our computational method will exploit, as the ability to efficiently sample from the stochastic process with correct drift v is equivalent to sampling from the ground state of the quantum system. Even though the connection is simple, it comes with a caveat. Starting from the Schrödinger equation one needs to find the drift $v(x)$ and while the connection with the potential of the Hamiltonian is clear-cut in this simple example, this is not the case in many-body systems, i.e. the *inverse problem* of finding the stochastic process of a given Hamiltonian is difficult, and is one of the core problems approached in this thesis.

3.2.2 Stoquastic Hamiltonians and Feynman-Kac in discrete state space

Before we illustrate the connection between the imaginary time Schrödinger equation and CTMCs we must introduce **Stoquastic Hamiltonians** [10], a class of Hamiltonians which do not suffer from the sign problem.

Definition 3.2.1 (Stoquastic Hamiltonian)**.** *A k-local Hamiltonian $\hat{H} = \sum_i \hat{H}_i$ is **stoquastic** if there exists a local basis $\mathcal{B}$ in which off-diagonal matrix elements of terms $\hat{H}_i$ are zero or negative*

$$\langle x|\hat{H}|y\rangle \leq 0, \quad \forall x, y \in \mathcal{B} \quad \textit{with } x \neq y. \tag{3.38}$$

If we consider the matrix $e^{-\tau\hat{H}}$ for a non-positive $\hat{H}$, we see that every term in the expansion

$$e^{-\tau\hat{H}} = 1 - \tau\hat{H} + \frac{1}{2}(\tau\hat{H})^2 + \ldots \tag{3.39}$$

is a non-negative matrix, thus so is $e^{-\tau\hat{H}}$. In the infinite time limit, the ground state is projected out

$$\lim_{\tau\to\infty} e^{-\tau\hat{H}} = |\psi_0\rangle\langle\psi_0|, \tag{3.40}$$

and a global phase exists for which the ground state has non-negative amplitudes. Moreover, if $\hat{H}$ is irreducible then the ground state is node-less [Crosson]

$$\psi_0(x) > 0 \text{ for all } x \in \mathcal{B}. \tag{3.41}$$

We can decompose a stoquastic Hamiltonian into a rate matrix Γ, as defined in section 3.1.6, and a diagonal potential matrix V

$$H = -\Gamma + V. \tag{3.42}$$

The rates Γ are analogous to the Brownian motion in the continuum case, or can be interpreted as the kinetic contribution. They represent a CTMC which we will understand as *passive dynamics*. In terms of the Hamiltonian matrix,

$$\Gamma_{s\to s'} = \begin{cases} -H_{ss'} & \text{if } s \neq s \\ \sum_{s'\neq s} H_{ss'} & \text{if } s = s' \end{cases} \tag{3.43}$$

and the potential is

$$V(\boldsymbol{s}) = H_{ss} + \sum_{s'\neq s} H_{ss'}. \tag{3.44}$$

From now on we use $\longrightarrow$ notation to emphasise the transition between *adjacent* states $s \neq s'$ which satisfy $H_{ss'} \neq 0$. This decomposition allows us to define the Feynman-Kac formula in the discrete state space [63] as

$$\psi(s_t, t) = \mathop{\mathbb{E}}_{\Sigma_{[0,t]}\sim\Gamma}\left[\exp\left(-\int_0^t V(\boldsymbol{s}_{t'})\,\mathrm{d}t'\right)\psi(s_0, 0)\right]. \tag{3.45}$$

The expectation is now taken over the process driven by Γ and weighted by the potential V. We denote the trajectory as $\Sigma_{[0,t]}$, where $\Sigma_{t'}$ is the state of the system at time $t' \in [0, t]$. Analogous as with the Feynman-Kac formula in continuous state space, this defines a new CTMC with measure $\mathbb{P}_{\mathrm{FK}}$, which is related to passive dynamics with measure $\mathbb{P}_0$ by the RN derivative, eq. (3.31).

How exactly is this CTMC connected to the imaginary time Schrödinger equation? The connection exists, as in the continuous space, via a similarity transform. The difference being that instead of Fokker-Planck we use the master equation to describe the time evolution of the pdf P

$$\frac{\partial P(s)}{\partial t} = \sum_{s'\neq s}\left[\Gamma_{s'\to s}P(s') - \Gamma_{s\to s'}P(s)\right]. \tag{3.46}$$

The stationary state P_0 of the master equation satisfies detailed balance

$$\Gamma_{k\to j} = \exp\left(\frac{V_{s'} - V_s}{2}\right), \tag{3.47}$$

and is thus

$$P_0(s) \propto \exp(-V_s). \tag{3.48}$$

The wave function

$$\psi(s,t) = \frac{P_s(t)}{\sqrt{P_0(s)}}, \tag{3.49}$$

then satisfies the imaginary time Schrödinger equation with the Hamiltonian

$$\hat{H}_{s's} = \begin{cases} -P_0^{-\frac{1}{2}}(s)\Gamma_{s'\to s}P_0^{\frac{1}{2}}(s') = -1 & s' \neq s \\ \sum_{s'\neq s}\Gamma_{s\to s'} & s' = s. \end{cases} \tag{3.50}$$

Again this Hamiltonian has a zero-energy ground state

$$\psi_0(s) = \sqrt{P_0(s)} \propto \exp\left(-\frac{V_s}{2}\right), \tag{3.51}$$

we see that the quantum probability in the ground state $|\psi_0(s)|^2$ coincides with the stationary distribution of the CTMC. The connection in the direction from the Markov process to Hamiltonian is clear, but we are interested in the inverse, starting from the Hamiltonian and finding for the corresponding stochastic process. We now turn our attention towards defining a suitable optimisation objective that, when optimised, will yield the correct Markov process.

3.3 Control theoretic approach to QM and loss functions

3.3.1 Holland Cost in continuous space

In section 3.2.1 we explored a connection between the ground state of some quantum system and Itô processes. We have seen that the ground state probability $|\psi_0|^2$ is also the stationary distribution π of some stochastic process

$$\mathrm{d}X_t = \mathrm{d}W_t + v'(X_t)\,\mathrm{d}t, \tag{3.52}$$

where v' is the *optimal*/correct drift. Finding the correct drift in one-dimension was straightforward, but how to do it in higher dimensions ($W_t, X_t \in \mathbb{R}^n$ and $v : \mathbb{R}^n \to \mathbb{R}^n$) remained unanswered. Holland [36] formulated the search for optimal rates as a stochastic control problem with cost function

$$C[v] = \lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\int\left(\frac{1}{2}|v(X_t)|^2 + V(X_t)\right)\mathrm{d}t\right], \tag{3.53}$$

where the expectation is over the process in eq. (3.52). Its derivation and proof of an unique solution can be found in Appendix A.1. To see that the minimum of $C[v]$ really corresponds to the ground state, we first rewrite the cost in terms of the stationary distribution $\pi(x|v)$ of Itô process generated by the drift v

$$C[v] = \int \left[\frac{1}{2}|v(X_t)|^2 + V(X_t)\right] \pi(x \mid v)\mathrm{d}x. \tag{3.54}$$

The distribution π must satisfy the stationary Fokker-Planck equation

$$\frac{1}{2}\nabla^2\pi - \nabla\cdot(v\pi) = 0, \tag{3.55}$$

the equation holds for drift $v = \frac{\nabla\psi}{\psi}$ and distribution $\pi = \psi^2$, the cost is then

$$C[v] = \int \left[\frac{1}{2}(\nabla\psi)^2 + V(X_t)\,\psi^2\right] \mathrm{d}x. \tag{3.56}$$

For normalised ψ the integrand is the expected value of the quantum energy of the system, its minimum value λ is the ground state energy E_0 and is achieved for the ground state wave function ψ_0. We could directly use the Holland's cost to find the optimal rates v', but we will instead use the fact that trajectories sampled under the Feynman-Kac measure $\mathbb{P}_{\mathrm{FK}}$ coincide with ones sampled from the optimal drift, and base our variational approach on finding the FK measure.

3.3.2 Loss for continuous state space

To see that the path measures $\mathbb{P}_v$ of the process in eq. (3.32) and the Feynman-Kac measure $\mathbb{P}_{\mathrm{FK}}$ coincide for the optimal rates v', we first need a measure of similarity between probability distributions. For this purpose we will use the **Kullback–Leibler** divergence D_{KL}

$$D_{\mathrm{KL}}(p\|q) = \mathop{\mathbb{E}}_{p}\left[\log\left(\frac{p}{q}\right)\right], \tag{3.57}$$

the quantity is a divergence because of the asymmetry $D_{\mathrm{KL}}(p\|q) \neq D_{\mathrm{KL}}(q\|p)$. It is a strictly positive quantity $D_{\mathrm{KL}}(p\|q) \geq 0$ except for $p = q$ when $D_{\mathrm{KL}}(p\|q) = 0$. In order to obtain the KL divergence between $\mathbb{P}_v$ and $\mathbb{P}_{\mathrm{KL}}$ we first express the RN derivative of each path measure w.r.t. Brownian motion using Girsanov theorem (3.16). Both RN derivatives can then be

combined to express the logarithm Radon-Nikodym derivative or the *log-likelihood* ratio as

$$\log\left(\frac{d\mathbb{P}_v}{d\mathbb{P}_{\mathrm{FK}}}\right) = \tilde{\ell} - E_0 T + \log\left(\frac{\varphi_0(r_0)}{\varphi_0(r_T)}\right), \tag{3.58}$$

where we have defined $\tilde{\ell}$ as

$$\tilde{\ell} \equiv \int v(r_t)\,\mathrm{d}W_t + \int \left(\frac{1}{2}|v(r_t)|^2 + V(r_t)\right)\mathrm{d}t. \tag{3.59}$$

The boundary term depends the ground state φ_0 at initial r_0 and final r_T points in the trajectory, and originates from the normalisation constant $\mathcal{N}$ in the Radon-Nikodym derivative. The KL divergence is

$$D_{\mathrm{KL}}(\mathbb{P}_v \| \mathbb{P}_{\mathrm{FK}}) = \mathop{\mathbb{E}}_{\mathbb{P}_v}\left[\ell' - E_0 T + \log\left(\frac{\varphi_0(r_0)}{\varphi_0(r_T)}\right)\right]. \tag{3.60}$$

If we consider the above Kullback-Leibler divergence in the long time limit $T \longrightarrow \infty$, we see that the expectation of the first term of $\tilde{\ell}$ is zero. Moreover, if the marginal distributions $\psi_0(r_0)$ and $\psi_0(r_T)$ coincide, the boundary term vanishes as well and the KL divergence is equivalent to the Holland cost, thus vanishes for optimal drift. This means that finding the correct path measure is equivalent to finding the optimal drift, and in standard machine learning fashion, minimizing D_{KL} can be used to find the optimal rates. Sampling the trajectories becomes a matter of integrating a SDE and can be done with some standard approach, e.g. Euler-Mayurama method. The rates v_θ are parameterised and gradients of D_{KL} w.r.t θ can be obtained using stochastic backpropagation where the reparameterisation trick is used for each increment in the discretised SDE. The gradient of the boundary term is non-zero, but it can be estimated by expressing ψ_0 in terms of v_θ. Missing steps of the derivation can be found in Appendix A.2, more details in [6].

3.3.3 Todorov Cost in discrete state space

We now turn our attention towards finding a variational approach in discrete state space and continuous time. We rely on foundational work done on linearly solvable Markov Decision Processes[4] (MDP) by Todorov [75, 76] and its applications to lattice problems [29]. The main idea is to reinterpret the dynamics of the imaginary time Schrödinger equation in the Todorov MDP framework by treating control as a modification of the passive dynamics

[4]For a general discussion of MDP's see [72]

of the system. The imaginary time Schrödinger equation can be written as

$$\frac{\partial \psi(s_j,t)}{\partial t} = \underbrace{-\sum_{s_k \neq s_j} \Gamma_{s_j \to s_k} \left[\psi(s_k,t) - \psi(s_j,t)\right]}_{\text{passive dynamics}} - V_{s_j}\psi(s_j,t) \tag{3.61}$$

to emphasise the decomposition of the Hamiltonian eq. (3.42) into passive dynamics and the potential. Akin to the transformation in discrete space, we use $\psi(s_j,t) = \exp[-u(s_j,t)]$, to arrive at an alternative form

$$-\frac{\partial u(s_j,t)}{\partial t} = \min_{\Gamma^{(v)}} \left[\ell(s_j,v) + \sum_{s_k} \Gamma^{(v)}_{s_j \to s_k} (u(s_k,t) - u(s_j,t)) \right], \tag{3.62}$$

where $\Gamma^{(v)}$ are the rates of the modified CTMC, and $\ell(s_j,v)$ is the cost per time associated with state s_j and parameters v. This is a form of Bellman equation, a very general concept in control theory and reinforcement learning, related to the Hamilton-Jacobi equation in physics, it is used to find the optimal actions i.e. how we should choose v at each moment w.r.t the cost associated with each state and parameters $\ell(s_j,v)$. The function $u(s_j,t)$ is the cost-to-go function, it is the cumulative cost obtained starting from state s_j at time t and acting optimally afterwards. The optimal parameters $v(t)$ are computed greedily with each step and guarantee minimum $u(s_j,t)$, most easily illustrated by looking Δt into the past, where the cost-to-go is given by the cost of remaining in the same state $\ell(s_j,t)\Delta t$ and the changes in costs-to-go when transitioning to other states weighted by the probability of making the transition $\Gamma^{(v)}_{s_j \to s_k}$

$$u_j(t-\Delta t) = u_j(t) + \Delta t \min_{\Gamma^{(v)}} \left[\ell(j,v) + \sum_k \Gamma^{(v)}_{j \to k} (u(k,t) - u(j,t)) \right]. \tag{3.63}$$

The cost $\ell(s_j,v)$ introduced by Todorov is of the form

$$\ell(j,v) = q(j) + D_{\mathrm{KL}}(v(\cdot \mid j) \| p(\cdot \mid j)), \tag{3.64}$$

where the first term encodes how undesirable different states are, and the second measures how costly the deviation from passive dynamics is, the agent (controlled rates) pay a price for reshaping the environment (passive rates), for more rigour see Appendix A.3. In the case of the imaginary time Schrödinger equation, these terms are

$$q(s_j) = V(s_j)\Delta t, \tag{3.65}$$

and

$$D_{\mathrm{KL}} = \mathop{\mathbb{E}}_{\Sigma_{[0,t]}=k_t\sim\Gamma^{(v)}}\left[\sum_n D_{\mathrm{IS}}\left(\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}, \Gamma_{k^{(n)}\to k^{(n+1)}}\right)\right], \tag{3.66}$$

where $\Sigma_{[0,t]}$ is the trajectory of states sampled from controlled dynamics $\Gamma^{(v)}$ which visits states $k^{(n)}$ for $t_{n-1} < t < t_n$ and D_{IS} is the Itakura-Saito divergence, see Appendix A.3.

3.3.4 Loss for discrete state space

As we have done in the continuous space, we will compare the cost to the Kullback-Liebler divergence of the two path measures $\mathbb{P}_v$ and $\mathbb{P}_{\mathrm{FK}}$ to see that the optimal rates $\Gamma^{(v)}$ coincide with sampling from the Feynman-Kac measure, full derivations in A.4. The logarithm Radon-Nikodym between the controlled rates and Feynman-Kac measure is

$$\log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_{FK}}(k(t))\right) = \tilde{\ell} + \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right) - E_0\,T - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right), \tag{3.67}$$

where we have defined $\tilde{\ell}$ as

$$\tilde{\ell} = \int\left[V(k(t)) + \sum_{l\neq k(t)}\left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l}\right)\right]\mathrm{d}t. \tag{3.68}$$

The KL divergence can be written as

$$\begin{aligned} D_{\mathrm{KL}}(\mathbb{P}_v \mid \mathbb{P}_{\mathrm{FK}}) = \mathop{\mathbb{E}}_{P_v}\Bigg[\int V(k(t)) + \sum_{l\neq k(t)}\left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l}\right) \\ + \Gamma^{(v)}_{k(t)\to l}\log\left(\frac{\Gamma^{(v)}_{k(t)\to l}}{\Gamma_{k(t)\to l}}\right)dt - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right)\Bigg] - E_0\,T, \end{aligned} \tag{3.69}$$

which is the Todorov's cost and vanishes for optimal rates, meaning that finding the these rates is equivalent to finding the Feynman-Kac measured process. One might be tempted to use D_{KL} as an optimisation objective, but this is not possible because its gradient is biased due to the boundary term $\log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right)$. We now propose two alternatives.

Loss no. 1: Variance of $\tilde{\ell}$

The variance of quantity $\tilde{\ell}$

$$\tilde{\ell} = \log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_{FK}}(k(t))\right) - \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right) + E_0\,T + \log\left(\frac{\varphi\left(k^{(N)}\right)}{\varphi\left(k^{(0)}\right)}\right) \tag{3.70}$$

can be used as a loss function. Kolmogorov's criterion tells us that

$$\log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right) = \log\left(\frac{\varphi(k^{(n+1)})}{\varphi(k^{(n)})}\right), \tag{3.71}$$

combined with the fact that the log Radon-Nikodym derivative is 0 for correct transition rates $\Gamma^{(v)} = \Gamma'$ means that $\mathrm{Var}[\tilde{\ell}]$ vanishes in that case, moreover it is bounded below by zero making it a suitable objective function.

Loss no. 2: Variance of log Radon-Nikodym derivative with fixed endpoints

While the gradients of $D_{\mathrm{KL}}(\mathbb{P}_v \mid \mathbb{P}_{\mathrm{FK}})$ are not accessible, we can still make use of the logarithm RN derivative in eq. (3.67). It vanishes for optimal rates and while we can treat $E_0\,T$ as constant, we have no easy way to approximate the wave function φ in the boundary term which in general does not have zero variance. We can exploit the fact that log RN must be constant on all trajectories and consider a batch of trajectories with fixed endpoints, in this case the variance of the boundary term vanishes

$$\operatorname*{Var}_{\substack{\Sigma_{[0,t]}\sim r,\\ \text{with fixed } k^{(0)} \text{ and } k^{(N)}}} \left[\tilde{\ell} + \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right) - E_0\,T - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right)\right] \underset{\Gamma^{(v)}=\Gamma'}{\longrightarrow} 0. \tag{3.72}$$

The trajectories can be sampled using arbitrary rates r, but for convenience $\Gamma^{(v)}$ can be used. How to obtain a batch poses a technical challenge which is to be solved on a system by system basis, usually by generating a single trajectory and perturbing it in such a way that one obtains many paths with fixed endpoints.

Chapter 4

Methodology

This chapter presents the computational method developed in this thesis. We start by introducing each component of the method separately before presenting how they fit together into a single procedure. We discuss neural networks, automatic differentiation, gradient estimation, gradient-based optimisation, as well as importance sampling. Finally, we present the actual implementation in *JAX*, and discuss computational considerations and intricacies of thereof.

4.1 Neural Networks

4.1.1 The Multilayer Perceptron

The **multilayer perceptron** (MLP), also referred to as a **deep feedforward network** or simply a **deep neural network** (DNN), is the paradigmatic model of deep learning and serves as a foundation of more advanced models. In essence it is nothing more than a mapping of inputs to outputs

$$f(\mathbf{x};\ \theta) : \mathbb{R}^{\text{in}} \longrightarrow \mathbb{R}^{\text{out}}, \tag{4.1}$$

which is structured in a certain way and depends on parameters θ. The mapping is a composition of vector-valued functions f which are called *layers* of the network, and with each layer we associate variational parameters $\mathbf{w}$, or simply the weights. The MLP can be described with a directed acyclic graph which details the compositions of the layers. The simplest and most common is a chain of compositions, Fig 4.1b,

$$f_{\text{MLP}} = \left(f^{(n)} \circ f^{(n-1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}\right)(\mathbf{x}), \tag{4.2}$$

where the input **x** passes through *hidden layers* before the *output layer* outputs the result. The length of this chain is the *depth* of the network, and the dimensionality of hidden layers is the *width* of the network. We can interpret each transformation $f^{(i)}$ as consisting of a unit/node/neuron for each input dimension, which is vector-to-scalar transformation, Fig 4.1a.

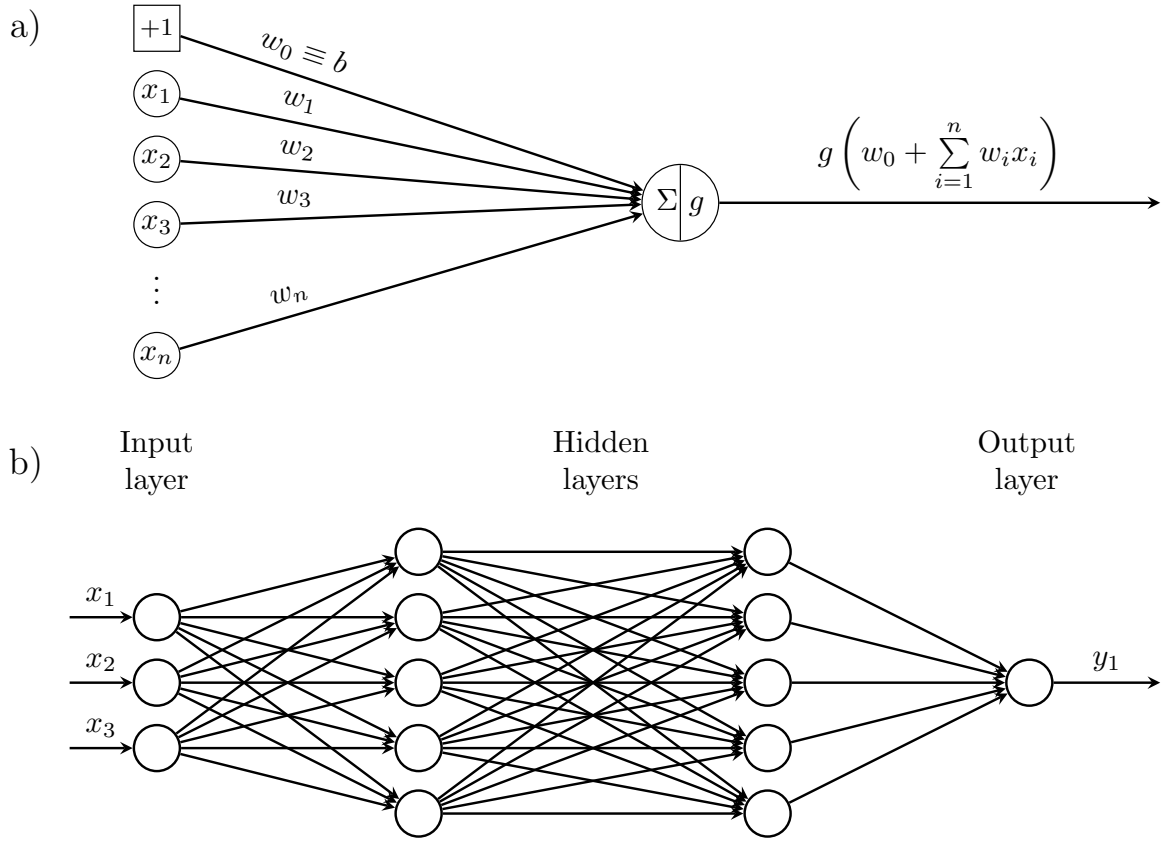


Fig. 4.1 **Multilayer Perceptron. top:** A node outputs the activation function σ evaluated at the weighted average over outputs from the previous layer and bias. **bottom:** A MLP with two hidden layers ($f^{(1)} : \mathbb{R}^3 \to \mathbb{R}^4$, $f^{(2)} : \mathbb{R}^4 \to \mathbb{R}^4$, $f^{(3)} : \mathbb{R}^4 \to \mathbb{R}$).

The simplest layer would be a linear one, composed of

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{x}^\mathsf{T}\mathbf{w} + b. \tag{4.3}$$

However, a linear neural network famously cannot learn the XOR function [51], and in practice a nonlinearity or *activation* function $g(\cdot)$ is used in each node to bolster the network's representational power

$$f(\mathbf{x}; \mathbf{w}, b) = g\left(\mathbf{x}^\mathsf{T}\mathbf{w} + b\right). \tag{4.4}$$

A variety of activations have been used, namely tanh(·) and the logistic sigmoid $\sigma(\cdot)$, but have since been displaced by the use of the **rectified linear unit** or ReLu(·), which is advantageous for training, for the output layer we will use the *softplus*, which fulfils the requirement of being positive everywhere, Fig. 4.2. A multilayer perceptron is a **universal function approximator**, meaning that with at least one hidden layer it can approximate any Borel measurable function mapping from a finite-dimensional space to another with desired accuracy, given sufficient hidden units [48]. This means that a large enough network will be able do represent the rates, however this provides no guarantee that learning the correct rates will be efficient or possible.

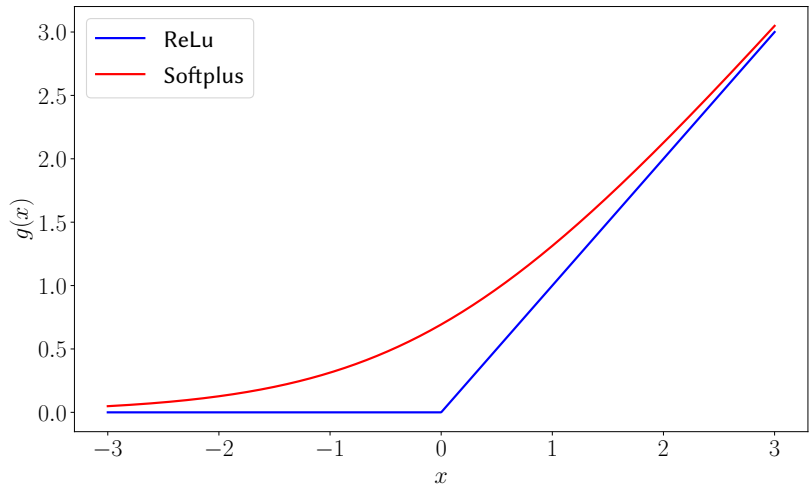


Fig. 4.2 **Activation functions.** ReLu and Softmax nonlinearities.

4.1.2 Convolutional Neural Networks

Convolutional neural networks (CNN) [46] are neural network architectures specialised to work with grid/image inputs. They make use of three powerful concepts to increase performance, **sparse interactions**, **parameter sharing** and **equivariant representations** [30]. At the core of a CNN is the *convolutional* layer in which the input I is convolved with the *kernel* K as

$$S_{i,j} = (I * K)_{i,j} = \sum_m \sum_n I_{m,n} K_{i-m,j-n}, \tag{4.5}$$

in two dimensions, Fig. 4.3a. Outputs of convolutions are referred to as *feature maps*. In practice the kernel K is much smaller than the input, meaning that each neuron is connected only to a small fraction of neurons in the previous layer, hence the layers are sparsely connected in contrast to the fully connected layer Fig. 4.3c. This decreases the number of weights and operations required. Moreover, the kernel is applied everywhere in the

input, meaning the weights are shared across all connections and need to be learned for the whole input as opposed to every single position in the input. This makes the convolutional layer much more memory efficient than the fully connected layer. Moreover, the nature of convolution in eq. (4.5) means that the convolutional layer is equivariant to translation, i.e. a translation of the input results in the same translation of the output. Altogether, a convolutional layer is a transformation acting on a batch of examples N_b with channels N_c

$$f : \mathbb{R}^{(N_b,N_w,N_h,N_c)} \longrightarrow \mathbb{R}^{(N_b,O_w,O_h,N_c)}. \tag{4.6}$$

The output dimensions also depend on the *stride* (how quick the kernel travels) N_S and *padding* N_P (how much the dimension of input is increased), Fig. 4.3a,

$$O_{\text{out}} = \frac{N_{\text{in}} - N_K + 2N_P}{N_S} + 1. \tag{4.7}$$

Alongside convolutional layers CNNs often employ *pooling* layers which downsample the input by combining a cluster of neurons into a single one. *Max* and *average* pooling are in common use, the pooling output is the max or average of the cluster respectively. A pooling layer can act globally, on the whole feature map, or locally.

The models we are interested in will be image-to-image networks that preserve the input shape, as the outputs will represent the rates corresponding to adjacent states. In the Ising model, the output at (i,j) is the rate associated with transition $s \longrightarrow s'$ where the spin at (i,j) is flipped.

Periodic CNN

The simplest model we will use is a CNN in which all layers preserve the shape of the input $N_{\text{in}} = O_{\text{out}}$. This means that the input into each layer needs to be padded

$$N_P = \frac{N_K - 1}{2}, \tag{4.8}$$

for $N_S = 1$. Given that the underlying lattice is chosen to be periodic, we use periodic instead of zero padding, see Fig. 4.3b. Hidden units use ReLu and the last layer uses softmax.

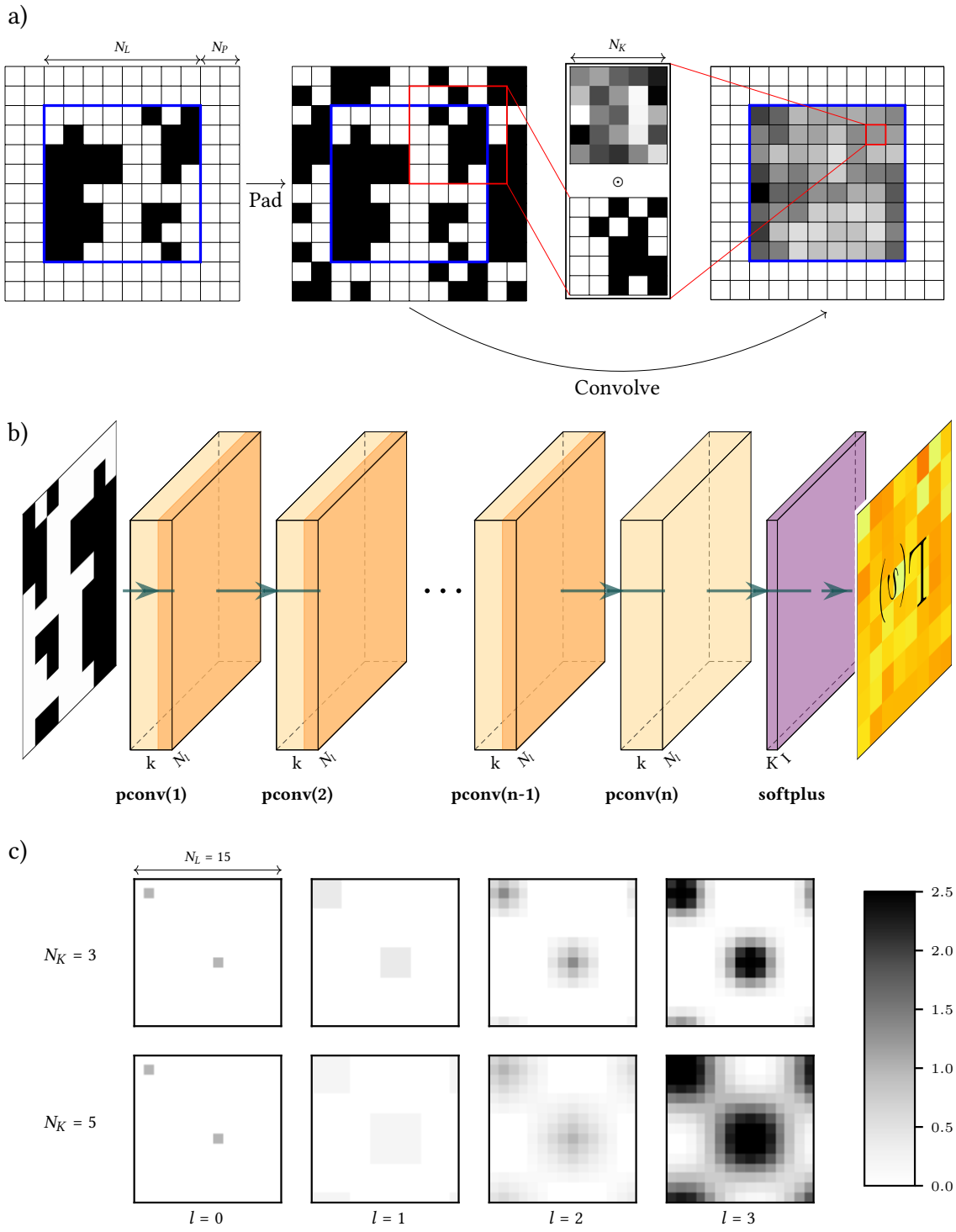


Fig. 4.3 **Periodic CNN top:** Each layer consists of periodic padding, and then convolving. The output shape matches the input shape. **middle:** The pCNN architecture, takes state s as input and outputs rates $\Gamma(s)$, similar to [29]. **bottom:** While the receptive field of a single node is reduced in a CNN, faraway nodes are still connected indirectly after enough layers, how many layers depends on the kernel size and stride. Figure shows successive applications of the convolutional layers for $N_s = 1$ and $N_K = 3$ or $N_K = 5$.

4.1.3 Group-Equivariant CNN

The basic convolutional layer is translation equivariant, providing an advantage when we expect the same equivariant behaviour between the input and output. Alongside translational symmetry we can take advantage of other symmetries present in the lattice model. Recent work [21, 11] in the field of **geometric deep learning** has shown how to construct layers equivariant to arbitrary group symmetry, in the case of discrete grid-like data referred to as **group-equivariant** convolutional layers. At the core of *group convolution* is moving the filter using group action (rotation, translation, etc.). Adopting notation of Bronstein [11], we write a group convolution as the inner product of the input x and a filter transformed by group element $\mathfrak{g} \in \mathfrak{G}$ via group representation $\rho(\mathfrak{g})$ as $\rho(\mathfrak{g})\theta_u = \theta_{\mathfrak{g}^{-1}u}$,

$$(x \star \theta)(\mathfrak{g}) = \sum_{u \in \Omega} x_u \rho(\mathfrak{g}) \theta_u. \tag{4.9}$$

We can separate the ordinary convolution from additional group transformations by writing

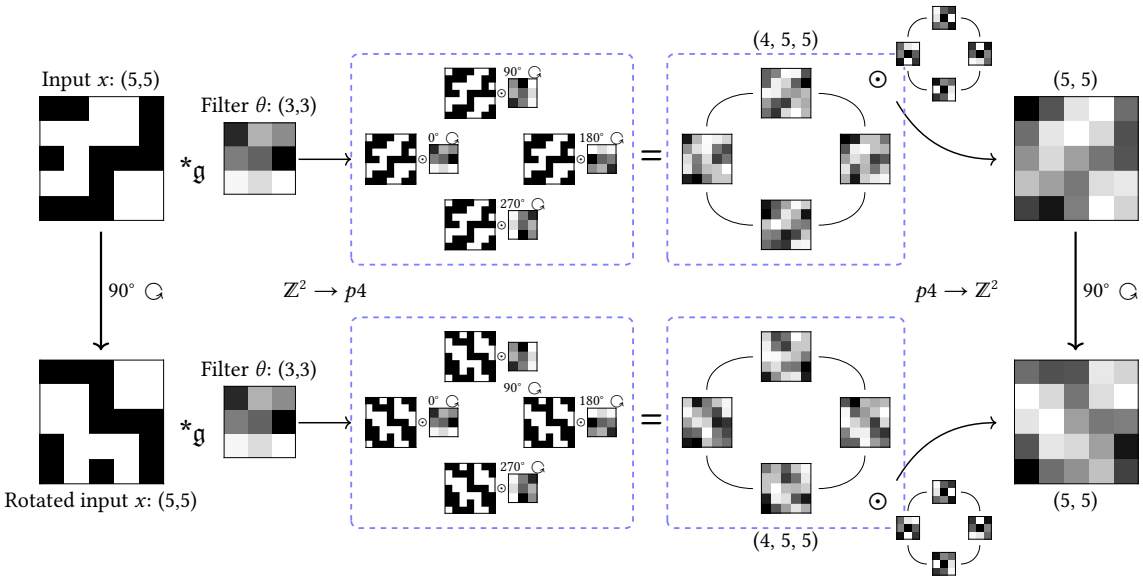


Fig. 4.4 **g-CNN layer.** Comparison of rotated inputs passing through two $p4$ equivariant layers. The first layer $\mathbb{Z}^2 \to p4$ produces a structured output, one convolution per each rotation of the filter θ, the second layer $p4 \to \mathbb{Z}^2$ is an elementwise convolution between the structured output and rotations of the second filter. The layers are equivariant to $\frac{\pi}{2}$ rotation.

a general group element $\mathfrak{g} \in \mathfrak{G}$ as a composition of a translation $\mathfrak{t}$ and rotation $\mathfrak{r}$, $\mathfrak{g} = \mathfrak{t}\mathfrak{r}$, and

using $\rho(\mathfrak{t}\mathfrak{r}) = \rho(\mathfrak{t})\rho(\mathfrak{r})$

$$(x \star \theta)(\mathfrak{t}\mathfrak{r}) = \sum_{u \in \Omega} x_u \rho(\mathfrak{t})\rho(\mathfrak{r})\theta_u = \sum_{u \in \Omega} x_u (\rho(\mathfrak{r})\theta)_{u-\mathfrak{t}}, \tag{4.10}$$

This yields a standard convolution with a transformed filter $\rho(\mathfrak{r})\theta$, meaning that we can implement the group-equivariant layer by first transforming the filter and performing convolution with each of these transformations, for details see Fig. 4.4 and [21].

4.2 Gradient-based optimisation

There are three components needed to perform gradient based optimisation. **Automatic differentiation** (AD), the ability to efficiently and reliably differentiate through computations. **Gradient estimation** methods for estimating gradients of expectations of functions, and an **optimisation algorithm**.

4.2.1 Automatic differentiation

The demand for automatic evaluation of derivatives is today greater than ever, and automatic differentiation has seen wide adoption within the scientific computing and machine learning communities [49, 74, 7], with the development of high profile libraries such as *PyTorch* [57], *TensorFlow* [1], or *JAX* [9]. Not to be confused with numerical or symbolic differentiation, **automatic differentiation** provides a way to mechanically find derivatives of functions expressed as a computer program, with certain complexity guarantees [58]. While modern implementations employ a variety of tricks, AD has two basic *modi operandi* of calculating partial derivatives of $f(x_1, x_2, \ldots, x_n) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ at $\mathbf{a}$. We start by representing the computation $y = f(\mathbf{x})$ as an *evaluation trace* of elementary operations, composed of inputs, intermediate values, and outputs:

- **Forward accumulation mode** [79]: In forward mode, when evaluating the partial derivative w.r.t x_j, we associate each intermediate value v_i with the partial derivative $\frac{\partial v_i}{\partial x_j}$. We then apply the chain rule to each operation in the evaluation trace, producing the derivative trace. A single forward pass, in tandem passing both primals v_i and their tangents $\frac{\partial v_i}{\partial x_j}$ through the trace, produces the output of the function, as well as the desired partial derivative. Forward mode AD produces one *column* of the Jacobian $\mathbf{J}_f$ at each pass, and is suited for functions with $n \ll m$.

- **Reverse accumulation mode** [70]:

AD is most commonly implemented in one of two ways, either by *operator overloading*, abstracting away the derivative part of the calculation, e.g. in *TensorFlow* of *JAX* [1, 9]. Or alternatively via *source code transformation*, where a new function is constructed by altering the source code, see *Zygote* [39].

4.2.2 Gradient estimation

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 [19, 67]. Different estimators differ in from and their properties, variance being one of the most important. In their review [52] Mohamed et al. categorise MC gradient estimators into three categories

Score-function estimator

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

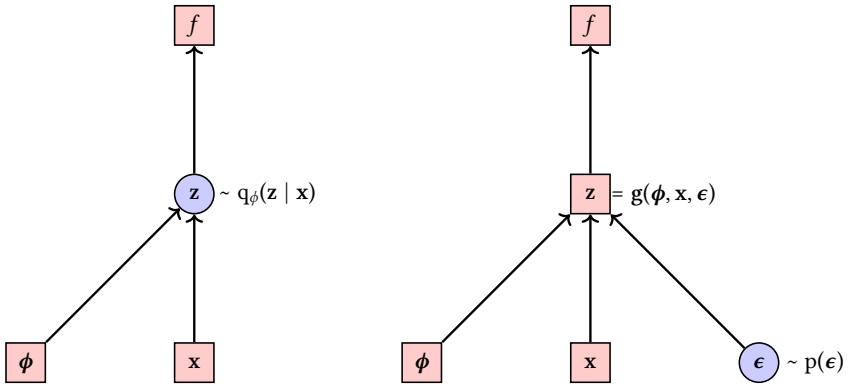
$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[f(\mathbf{x})] &= \nabla_{\boldsymbol{\theta}} \int p_{\boldsymbol{\theta}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[f(\mathbf{x}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})] . \end{aligned} \tag{4.11}$$

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. Importantly it is applicable to both discrete and continuous distribution, but has the drawback of having high variance.

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

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x})}[f(\mathbf{x})] &= \nabla_{\boldsymbol{\theta}} \int p_{\boldsymbol{\theta}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= \nabla_{\boldsymbol{\theta}} \int p(\boldsymbol{\epsilon}) f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})) d\boldsymbol{\epsilon} \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} f(g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}))] . \end{aligned} \tag{4.12}$$



(a) Original (b) Reparametrized

Fig. 4.5 **The reparametrization trick**, adapted from [43]. Stochasticity of the **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.

This was the gradient estimator originally used in the VAE implementation [44] there named as the *reparametrization trick*, see also Figure 4.5. 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.

Measure-valued gradient estimator

Which exploits the properties of signed-measures, is beyond the scope of this report.

1. measure-valued
2. how does this apply to our objective?

4.2.3 Optimisation algorithms

4.3 Monte Carlo Importance Sampling

The most common application of Monte Carlo methods is evaluation of integrals in high dimensional space. There MC has a distinct advantage over quadrature methods, as the statistical error decreases with the square root of samples irregardless of the dimensionality of the problem. Integrals of a function $g(\mathbf{R})$

$$I = \int g(\mathbf{R}) \mathrm{d}\mathbf{R}, \tag{4.13}$$

where $\mathbf{R}$ is the *configuration* of the system or simply a *walker*, can be integrated by use of an *importance function* $\mathrm{P}(\mathbf{R})$, where $\int d\mathbf{R}\mathrm{P}(\mathbf{R}) = 1$ and $\mathrm{P}(\mathbf{R}) \geq 0$. The integral can be rewritten in the form

$$\int g(\mathbf{R})\mathrm{d}\mathbf{R} = \int \frac{g(\mathbf{R})}{\mathrm{P}(\mathbf{R})}\mathrm{P}(\mathbf{R})\mathrm{d}\mathbf{R} = \int f(\mathbf{R})\mathrm{P}(\mathbf{R})\mathrm{d}\mathbf{R}, \tag{4.14}$$

where $f(\mathbf{R}) = g(\mathbf{R})/\mathrm{P}(\mathbf{R})$. The importance function $\mathrm{P}(\mathbf{R})$ can be interpreted as a probability density. If we generate an infinite number of random uncorrelated configurations $\mathbf{R}_m$ from the distribution $\mathrm{P}(\mathbf{R})$, the sample average is a good estimator of the integral I

$$I = \lim_{M\to\infty}\left\{\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{R}_m)\right\}, \tag{4.15}$$

and for an approximation with a finite number of samples

$$I \approx \frac{1}{M}\sum_{m=1}^{M} f(\mathbf{R}_m). \tag{4.16}$$

Under conditions where the central limit theorem holds [28], the estimator is normally distributed with variance σ_f^2/M, which can also be estimated from the samples as

$$\frac{\sigma_f^2}{M} \approx \frac{1}{M(M-1)}\sum_{m=1}^{M}\left[f(\mathbf{R}_m) - \frac{1}{M}\sum_{n=1}^{M} f(\mathbf{R}_n)\right]^2. \tag{4.17}$$

4.4 Metropolis-Hastings Algorithm

The integration technique from the previous section relies on our ability to obtain samples from a probability distribution $\mathrm{P}(\mathbf{R})$. In the case of QMC these distributions are high-dimensional and cannot be directly sampled from. Moreover their normalisations are usually not known. The Metropolis-Hastings algorithm [33], see Algorithm 1, avoids direct sampling from the distribution $\mathrm{P}(\mathbf{R})$ and is insensitive to its normalisation. It uses a Markov process whose stationary distribution $\pi(\mathbf{R})$ is the same as $\mathrm{P}(\mathbf{R})$ to generate a sequence of configurations $\{\mathbf{R}_n\}_\mathrm{P}$ that are drawn from $\mathrm{P}(\mathbf{R})$. A Markov process is completely defined with its transition probability $\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')$, which is the probability of transitioning from state $\mathbf{R}$ to state $\mathbf{R}'$. For the process to have a unique stationary distribution two conditions must be met, the process must be *ergodic* and it must obey *detailed balance*

$$\mathrm{P}(\mathbf{R})\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{P}(\mathbf{R}')\mathrm{P}(\mathbf{R}' \longrightarrow \mathbf{R}), \tag{4.18}$$

rewritten as

$$\frac{\mathrm{P}(\mathbf{R})}{\mathrm{P}(\mathbf{R}')} = \frac{\mathrm{P}(\mathbf{R}' \longrightarrow \mathbf{R})}{\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')}. \tag{4.19}$$

The right transition probability $\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')$ is not known, but we can express it with a trial move transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$ which we sample and acceptance probability $\mathrm{A}(\mathbf{R} \longrightarrow \mathbf{R}')$ as

$$\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')\mathrm{A}(\mathbf{R} \longrightarrow \mathbf{R}'). \tag{4.20}$$

For equation (4.19) to hold, the acceptance probability must be

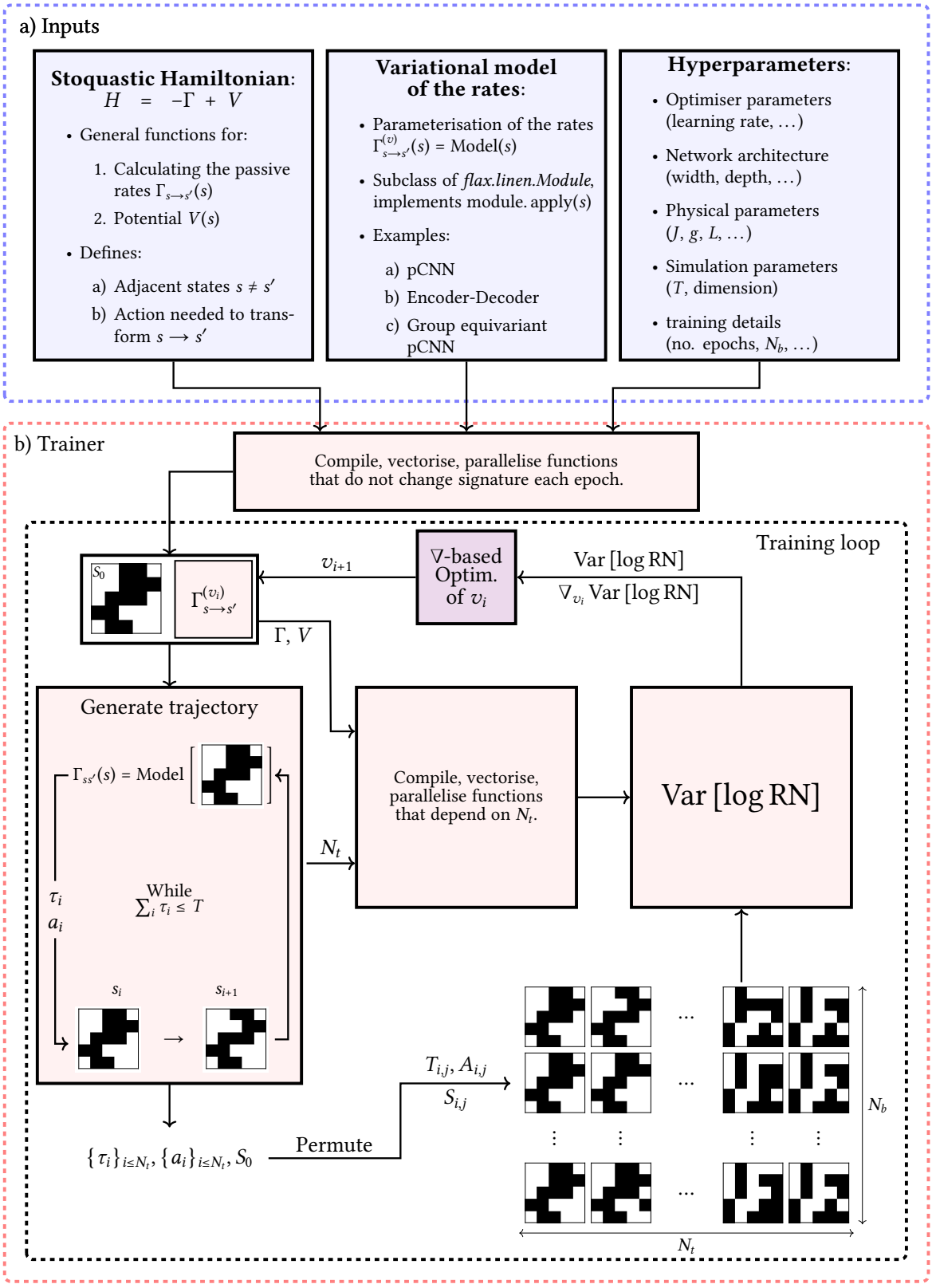
$$A\left(\mathbf{R} \longrightarrow \mathbf{R}'\right) = \min\left(1, \frac{\mathrm{T}\left(\mathbf{R}' \longrightarrow \mathbf{R}\right)\mathrm{P}\left(\mathbf{R}'\right)}{\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')\mathrm{P}(\mathbf{R})}\right). \tag{4.21}$$

Thus to sample from any probability distribution we need only have the ability to calculate probabilities $\mathrm{P}(\mathbf{R})$ and to sample from a trial transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$. The efficiency of the algorithm depends on the amount of trial moves that we reject. All trial moves would be accepted if $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{P}(\mathbf{R}')$, which would just mean sampling from P directly and is the very problem we are trying to solve with Metropolis-Hastings.

Algorithm 1: Metropolis-Hastings

Result: A set of configurations $\{\mathbf{R}_n\}_\mathrm{P}$ sampled from P
Initialize walker at random configuration $\mathbf{R}$;
while *no. samples less than N* **do**
  Generate new configuration $\mathbf{R}'$ with transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$;
  Accept the move $(\mathbf{R} \longrightarrow \mathbf{R}')$ with probability
  $A\left(\mathbf{R} \longrightarrow \mathbf{R}'\right) = \min\left(1, \frac{\mathrm{T}(\mathbf{R}' \longrightarrow \mathbf{R})\mathrm{P}(\mathbf{R}')}{\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')\mathrm{P}(\mathbf{R})}\right)$;
  Append $\mathbf{R}$ to the set of configuration;
end

4.5 *Qptimal sampling*: optimal sampling in lattice models


a) Inputs
Stoquastic Hamiltonian:
$H = -\Gamma + V$
• General functions for:
1. Calculating the passive rates $\Gamma_{s\to s'}(s)$
2. Potential $V(s)$
• Defines:
a) Adjacent states $s \neq s'$
b) Action needed to transform $s \to s'$
Variational model of the rates:
• Parameterisation of the rates $\Gamma^{(v)}_{s\to s'}(s)$ = Model(s)
• Subclass of *flax.linen.Module*, implements module.apply(s)
• Examples:
a) pCNN
b) Encoder-Decoder
c) Group equivariant pCNN
Hyperparameters:
• Optimiser parameters (learning rate, …)
• Network architecture (width, depth, …)
• Physical parameters (J, g, L, …)
• Simulation parameters (T, dimension)
• training details (no. epochs, N_b, …)
b) Trainer
Compile, vectorise, parallelise functions that do not change signature each epoch.
Training loop
S_0
$\Gamma^{(v_i)}_{s\to s'}$
v_{i+1}
∇-based Optim. of v_i
Var [log RN]
∇_{v_i} Var [log RN]
Γ, V
Generate trajectory
$\Gamma_{ss'}(s)$ = Model
τ_i
a_i
While $\sum_i \tau_i \leq T$
s_i
s_{i+1}
N_t
Compile, vectorise, parallelise functions that depend on N_t.
Var [log RN]
$T_{i,j}, A_{i,j}$
$S_{i,j}$
$\{\tau_i\}_{i\leq N_t}, \{a_i\}_{i\leq N_t}, S_0$
Permute
N_b
N_t

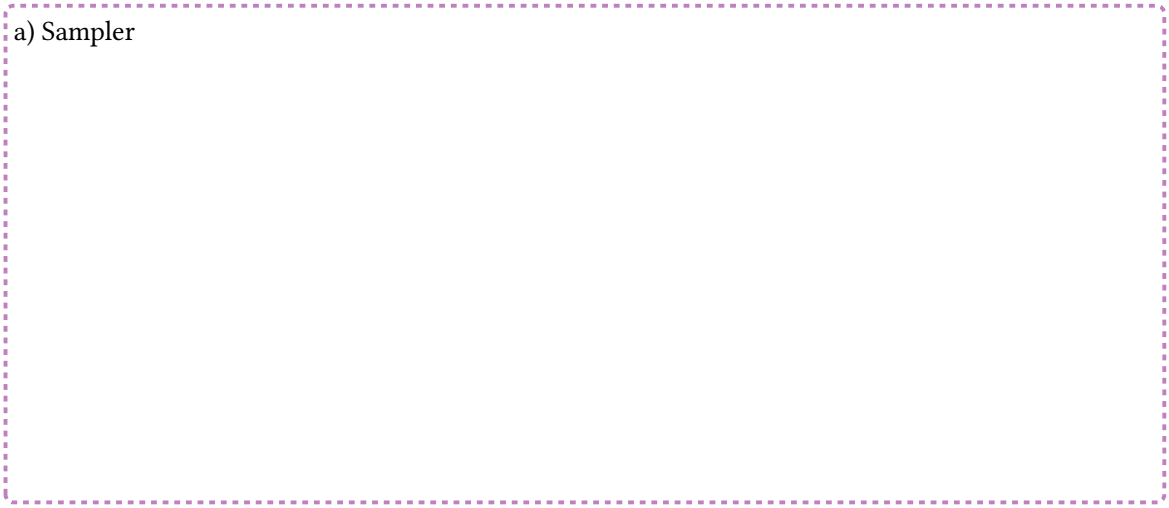


Fig. 4.5 **Implementation details**

Chapter 5

Experiments and Results

5.1 Stoquastic lattice models

5.1.1 Transverse-field Ising model

If we transform the transverse-field Ising model Hamiltonian in eq.(2.8) into the z-spin basis, where s =

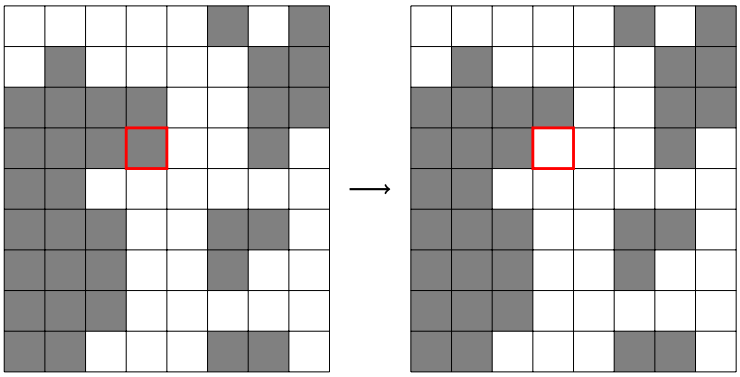


Fig. 5.1 **Ising passive process**

5.1.2 Heisenberg model

Heisenberg ferromagnet

$$\hat{H}_{\mathrm{F}} = -\frac{1}{2}\sum_{j}\left[\hat{\sigma}_j^x\hat{\sigma}_{j+1}^x + \hat{\sigma}_j^y\hat{\sigma}_{j+1}^y + \hat{\sigma}_j^z\hat{\sigma}_{j+1}^z\right] \tag{5.1}$$

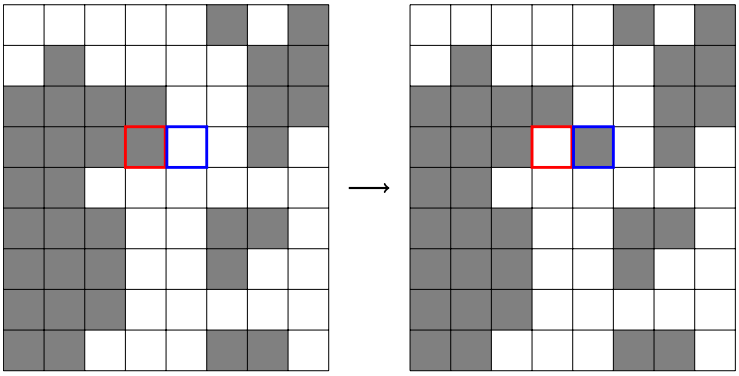


Fig. 5.2 **XY passive process**

The XY model.

$$\begin{aligned}\hat{H}_{XY} = -\sum_j \left[\hat{\sigma}_j^x \hat{\sigma}_{j+1}^x + \hat{\sigma}_j^y \hat{\sigma}_{j+1}^y\right] &= H_{\mathrm{F}} + \frac{1}{2}\sum_j \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z \\ &= -\mathcal{W} + \sum_j \left[n_j\left(1-n_{j+1}\right) + n_{j+1}\left(1-n_j\right)\right]\end{aligned} \tag{5.2}$$

$$\psi_{s_{1:N}}(t) = \mathop{\mathbb{E}}_{\Sigma_{[0,t]}\sim\mathrm{SEP}\ \text{with}\ \Sigma_t = s_{1:N}} \left[\exp\left(-\int_0^t dt' \sum_j \left[n_j\left(1-n_{j+1}\right) + n_{j+1}\left(1-n_j\right)\right]\right)\psi_{\Sigma_0}(0)\right] \tag{5.3}$$

5.2 Learning the rates

How does simulation time T and batch size N_b affect the learning of the rates?

How do architectural choices of the pCNN limit learning the rates?

How does the size of lattice play a role?

Can we improve learning by scaling the output of the pCNN?

Also mention reshuffling.

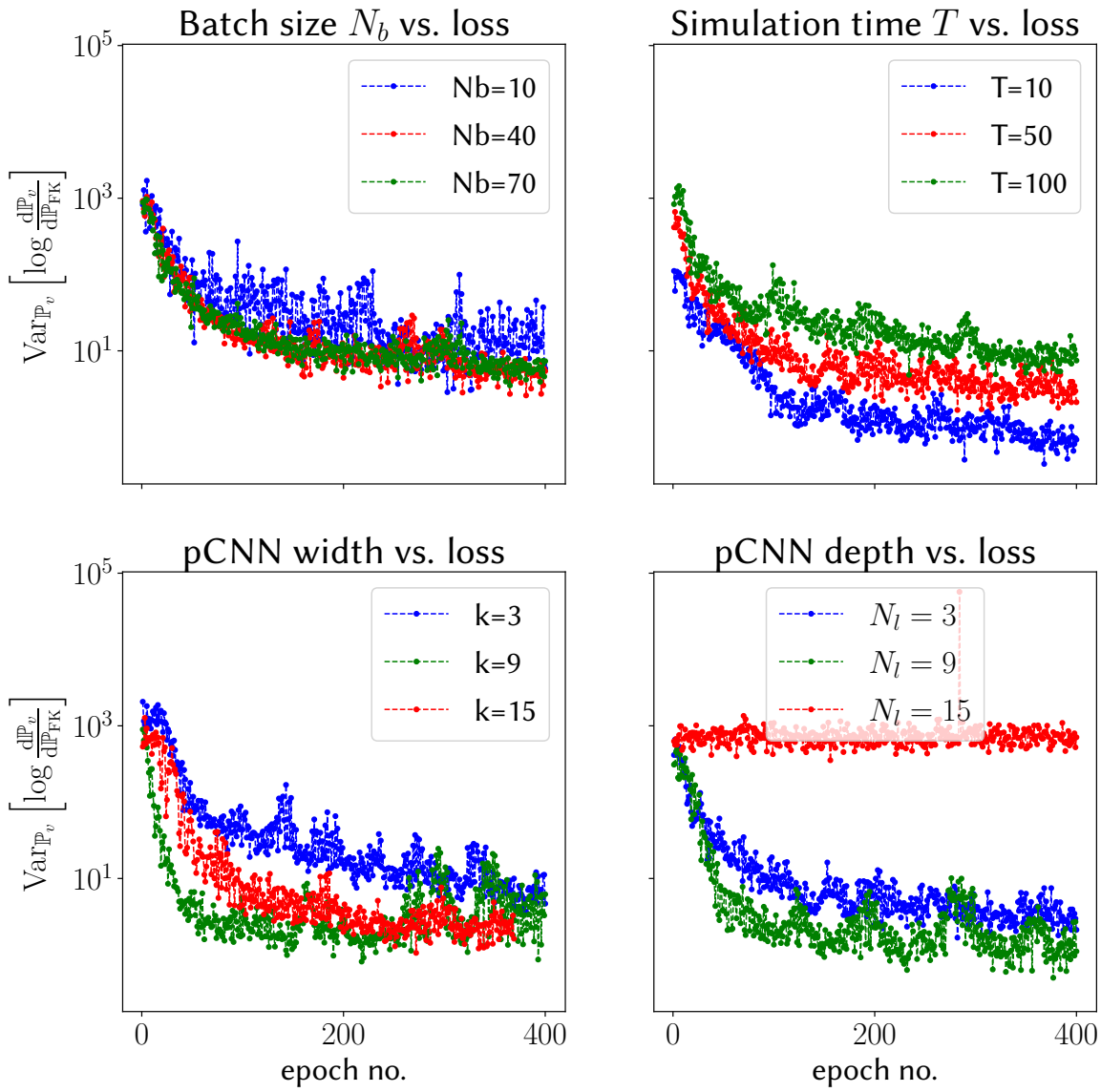


Fig. 5.3 **Initial rate training experiments**

What role does symmetry play in all of this?

5.3 Importance sampling

5.3.1 Ising model

5.3.2 XY model

Chapter 6

Discussion

6.1 Direction for further work

6.2 Remarks

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

[2] Albeverio, S., Ho/egh-Krohn, R., and Streit, L. (1977). Energy forms, hamiltonians, and distorted brownian paths. *Journal of Mathematical Physics*, 18(5):907–917.

[3] Alexandru, A., Basar, G., Bedaque, P. F., and Warrington, N. C. (2020). Complex paths around the sign problem. *arXiv preprint arXiv:2007.05436*.

[4] Anderson, J. B. (1975). A random-walk simulation of the schrödinger equation: H+ 3. *The Journal of Chemical Physics*, 63(4):1499–1503.

[5] Assaraf, R., Caffarel, M., and Khelif, A. (2007). The fermion monte carlo revisited. *Journal of Physics A: Mathematical and Theoretical*, 40(6):1181.

[6] Barr, A., Gispen, W., and Lamacraft, A. (2020). Quantum ground states from reinforcement learning. In *Mathematical and Scientific Machine Learning*, pages 635–653. PMLR.

[7] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18.

[8] Becca, F. and Sorella, S. (2017). *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press.

[9] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

[10] Bravyi, S., Divincenzo, D. P., Oliveira, R. I., and Terhal, B. M. (2006). The complexity of stoquastic local hamiltonian problems. *arXiv preprint quant-ph/0606140*.

[11] Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

[12] Cai, Z. and Liu, J. (2018). Approximating quantum many-body wave functions using artificial neural networks. *Physical Review B*, 97(3):035116.

[13] Carleo, G., Becca, F., Sanchez-Palencia, L., Sorella, S., and Fabrizio, M. (2014). Light-cone effect and supersonic correlations in one-and two-dimensional bosonic superfluids. *Physical Review A*, 89(3):031602.

[14] Carleo, G., Becca, F., Schiró, M., and Fabrizio, M. (2012). Localization and glassy dynamics of many-body quantum systems. *Scientific reports*, 2(1):1–6.

[15] Carleo, G., Nomura, Y., and Imada, M. (2018). Constructing exact representations of quantum many-body systems with deep neural networks. *Nature communications*, 9(1):1–11.

[16] Carleo, G. and Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606.

[17] Ceperley, D., Chester, G. V., and Kalos, M. H. (1977). Monte carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081.

[18] Childs, A. M. (2010). On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603.

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

[20] Clark, S. R. (2018). Unifying neural-network quantum states and correlator product states via tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 51(13):135301.

[21] Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

[Crosson] Crosson, E. Discussion on stoquastic hamiltonians.

[23] Dai Pra, P. and Pavon, M. (1990). On the markov processes of schroedinger, the feynman-kac formula and stochastic control. In *Realization and Modelling in System Theory*, pages 497–504. Springer.

[24] Dick, S. and Fernandez-Serra, M. (2020). Machine learning accurate exchange and correlation functionals of the electronic density. *Nature communications*, 11(1):1–10.

[25] Durrett, R. (2019). *Probability: theory and examples*, volume 49. Cambridge university press.

[26] Feiguin, A. E. and White, S. R. (2005). Time-step targeting methods for real-time dynamics using the density matrix renormalization group. *Physical Review B*, 72(2):020404.

[27] Feynman, R. P. (2018). Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press.

[28] Foulkes, W., Mitas, L., Needs, R., and Rajagopal, G. (2001). Quantum monte carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33.

[29] Gispen, W. and Lamacraft, A. (2020). Ground states of quantum many body lattice models via reinforcement learning. *arXiv preprint arXiv:2012.07063*.

[30] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*. MIT press Cambridge.

[31] Gubernatis, J., Kawashima, N., and Werner, P. (2016). *Quantum Monte Carlo Methods: Algorithms for Lattice Models*. Cambridge University Press.

[32] Halmos, P. R. (2013). *Measure theory*, volume 18. Springer.

[33] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*.

[34] Held, K. (2007). Electronic structure calculations using dynamical mean field theory. *Advances in physics*, 56(6):829–926.

[35] Hohenberg, P. and Kohn, W. (1964). Inhomogeneous electron gas. *Physical review*, 136(3B):B864.

[36] Holland, C. J. (1977). A new energy characterization of the smallest eigenvalue of the schrödinger equation. *Communications in Pure Applied Mathematics*, 30:755–765.

[37] Hutcheon, M. (2020). Stochastic nodal surfaces in quantum monte carlo calculations. *Physical Review E*, 102(4):042105.

[38] Ido, K., Ohgoe, T., and Imada, M. (2015). Time-dependent many-variable variational monte carlo method for nonequilibrium strongly correlated electron systems. *Physical Review B*, 92(24):245106.

[39] Innes, M. (2018). Don't unroll adjoint: Differentiating ssa-form programs. *arXiv preprint arXiv:1810.07951*.

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

[41] Kalos, M. (1962). Monte carlo calculations of the ground state of three-and four-body nuclei. *Physical Review*, 128(4):1791.

[42] Kalos, M. (1966). Stochastic wave function for atomic helium. *Journal of Computational Physics*, 1(2):257–276.

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

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

[45] Kipnis, C. and Landim, C. (1998). *Scaling limits of interacting particle systems*, volume 320. Springer Science & Business Media.

[46] LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, 19:143–155.

[47] Léonard, C. (2014). Some properties of path measures. In *Séminaire de Probabilités XLVI*, pages 207–230. Springer.

[48] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.

[49] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J. R. (2013). Automatic differentiation adjoint of the reynolds-averaged navier-stokes equations with a turbulence model. In *21st AIAA Computational Fluid Dynamics Conference*, page 2581.

[50] McMillan, W. L. (1965). Ground state of liquid he 4. *Physical Review*, 138(2A):A442.

[51] Minsky, M. and Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. MIT press.

[52] 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.

[53] Nelson, E. (1967). Dynamical theories of brownian motion, princeton univ. *Press, Princeton, NJ*.

[54] Nightingale, M. and Melik-Alaverdian, V. (2001). Optimization of ground-and excited-state wave functions and van der waals clusters. *Physical review letters*, 87(4):043401.

[55] Nomura, Y., Darmawan, A. S., Yamaji, Y., and Imada, M. (2017). Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152.

[56] Norris, J. R. and Norris, J. R. (1998). *Markov chains*. Number 2. Cambridge university press.

[57] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[58] Pearlmutter, B. A. (2016). Automatic differentiation: History and headroom. Workshop on the future of gradient-based machine learning software, NIPS.

[59] Pfau, D., Spencer, J. S., Matthews, A. G., and Foulkes, W. M. C. (2020). Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429.

[60] Rasmussen, C., Williams, C., Press, M., Bach, F., and (Firm), P. (2006). *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press.

[61] Reynolds, P. J., Tobochnik, J., and Gould, H. (1990). Diffusion quantum monte carlo. *Computers in Physics*, 4(6):662–668.

[62] Rogers, L. C. G. and Williams, D. (1994). Diffusions, markov processes and martingales, volume 1: Foundations. *John Wiley & Sons, Ltd., Chichester*, 7.

[63] Rogers, L. C. G. and Williams, D. (2000). *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press.

[64] Saito, H. (2017). Solving the bose–hubbard model with machine learning. *Journal of the Physical Society of Japan*, 86(9):093001.

[65] Salamon, D. (2016). *Measure and integration*. European Mathematical Society.

[66] Särkkä, S. and Solin, A. (2019). *Applied stochastic differential equations*, volume 10. Cambridge University Press.

[67] 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.

[68] Skylaris, C.-K., Haynes, P. D., Mostofi, A. A., and Payne, M. C. (2005). Introducing onetep: Linear-scaling density functional simulations on parallel computers. *The Journal of chemical physics*, 122(8):084119.

[69] Sorella, S. (1998). Green function monte carlo with stochastic reconfiguration. *Physical review letters*, 80(20):4558.

[70] Speelpenning, B. (1980). *Compiling fast partial derivatives of functions given by algorithms*. PhD thesis, University of Illinois at Urbana-Champaign.

[71] Spencer, J. S., Pfau, D., Botev, A., and Foulkes, W. M. C. (2020). Better, faster fermionic neural networks. *arXiv preprint arXiv:2011.07125*.

[72] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[73] Szabó, A. and Castelnovo, C. (2020). Neural network wave functions and the sign problem. *Physical Review Research*, 2(3):033075.

[74] Tamayo-Mendoza, T., Kreisbeck, C., Lindh, R., and Aspuru-Guzik, A. (2018). Automatic differentiation in quantum chemistry with applications to fully variational hartree–fock. *ACS central science*, 4(5):559–566.

[75] Todorov, E. (2007). Linearly-solvable markov decision problems. In *Advances in neural information processing systems*, pages 1369–1376.

[76] Todorov, E. (2009). Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483.

[77] Troyer, M. and Wiese, U.-J. (2005). Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical review letters*, 94(17):170201.

[78] Verstraete, F. and Cirac, J. I. (2004). Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*.

[79] Wengert, R. E. (1964). A simple automatic derivative evaluation program. *Communications of the ACM*, 7(8):463–464.

[80] White, S. R. (1992). Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863.

[81] Wilson, K. G. (1975). The renormalization group: Critical phenomena and the kondo problem. *Reviews of modern physics*, 47(4):773.

[82] Wilson, M., Gao, N., Wudarski, F., Rieffel, E., and Tubman, N. M. (2021). Simulations of state-of-the-art fermionic neural network wave functions with diffusion monte carlo. *arXiv preprint arXiv:2103.12570*.

[83] Zhang, S., Carlson, J., and Gubernatis, J. E. (1997). Constrained path monte carlo method for fermion ground states. *Physical Review B*, 55(12):7464.

Appendix A

Additional Derivations

A.1 Holland cost

Following Holland's [36] derivation of a stochastic control problem for the Schrödinger equation. We start by introducing the exponential transform

$$\psi(x) = \exp[-U(x)], \tag{A.1}$$

into the Schroödinger equation for $x \in \mathbb{R}^n$

$$H\psi = \left[-\frac{1}{2}\nabla^2 + V(x)\right]\psi = \lambda\psi, \tag{A.2}$$

to obtain

$$\frac{1}{2}\nabla^2 U - \frac{1}{2}(\nabla U)^2 + V(x) = \lambda. \tag{A.3}$$

We can reinterpret the second term in eq. (A.3) as a minimum over all vectors v for every $x \in \mathbb{R}^n$

$$\frac{1}{2}\nabla^2 U + \min_v \left[v \cdot \nabla U + \frac{1}{2}|v|^2 + V(x)\right] = \lambda. \tag{A.4}$$

We define v to be a Lipshitz continuous function, i.e. the drift $v(x)$. Each drift now generates an Itô process

$$\mathrm{d}X_t = \mathrm{d}W_t + v(X_t)\,\mathrm{d}t; \qquad \text{and} \qquad X_0 = x, \tag{A.5}$$

and we can define the cost function $C[v]$ for each v as

$$C[v] = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int \left(\frac{1}{2}|v(X_t)|^2 + V(X_t)\right)\mathrm{d}x\right]. \tag{A.6}$$

Holland [36] proves the following theorem.

Theorem A.1.1 (Holland cost function)**.** *The minimum* $\lambda = \min_v C[v]$*, where the minimum is taken over drift functions* $v : \Omega \longrightarrow \mathbb{R}^n$*,* $\Omega \subset \mathbb{R}^n$ *with Neumann boundary conditions* $\frac{\partial \psi}{\partial n} = 0$ *on* $\partial\Omega$*, is obtained only for* $v = \frac{\nabla\psi}{\psi} = -\nabla U$*.*

Proof. From eq. (A.4) follows the inequality

$$\frac{1}{2}\nabla^2 U + v \cdot \nabla U + \frac{1}{2}|v(x)|^2 + V(x) \geq \lambda, \tag{A.7}$$

we notice that the first two terms are an infinitesimal generator $[\mathcal{L}_G U](x)$ of the process in eq. (A.5), which is defined as

$$[\mathcal{L}_G \psi](x) \equiv \lim_{t \longrightarrow 0} \frac{\mathbb{E}_x[\psi(X_t)] - \psi(x)}{t}, \tag{A.8}$$

where $\mathbb{E}_x$ is over processes with initial condition $X_0 = x$. With this in mind, we evaluate eq. (A.7) on the process X_t and integrate over time to obtain

$$\frac{\mathbb{E}[U(X_T) - U(x)]}{T} + \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\frac{1}{2}|v(X_t)|^2 + V(X_t)\right) \mathrm{d}t\right] \geq \lambda. \tag{A.9}$$

In the $T \longrightarrow \infty$ limit, the first terms disappears, so long as U is bounded, and we are left with $C[v]$, meaning that we have proven the bound

$$C[v] \geq \lambda. \tag{A.10}$$

We see that the minimum λ is achieved for control function

$$v = \frac{\nabla\psi}{\psi} = -\nabla U, \tag{A.11}$$

in eq. (A.4), the uniqueness of this optimal v is due to the fact that if $v \neq \frac{\nabla\psi}{\psi}$ there cannot be a full equality in eq. (A.7). □

A.2 Probabilistic interpretation of Holland's cost function

This section follows Barr et al. [6] which is closely related to previous work [23]. To express the RN derivative we use the Girsanov theorem for both $\mathbb{P}_v$

$$\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_0} = \exp\left(\int v(X_t)\,\mathrm{d}X_t - \frac{1}{2}\int |v(X_t)|^2\,\mathrm{d}t\right), \tag{A.12}$$

and $\mathbb{P}_{\mathrm{FK}}$

$$\frac{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}{\mathrm{d}\mathbb{P}_0} = \mathcal{N} \exp\left(-\int V(X_t)\,\mathrm{d}t\right). \tag{A.13}$$

We combine both, and we have up to exponential accuracy $\mathcal{N} \sim e^{\lambda T}$

$$\log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}\right) = \log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_0}\frac{\mathrm{d}\mathbb{P}_0}{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}\right) = \int v(X_t)\,\mathrm{d}X_t + \int \left(-\frac{1}{2}|v(X_t)|^2 + V(X_t)\right)\mathrm{d}t - \lambda T, \tag{A.14}$$

finally we substitute $\mathrm{d}X_t = \mathrm{d}W_t + v(X_t)\mathrm{d}t$ to get

$$\begin{aligned}\log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}\right) &= \int v(X_t)\,\mathrm{d}W_t + \int |v(X_t)|^2\mathrm{d}t + \int \left(-\frac{1}{2}|v(X_t)|^2 + V(X_t)\right)\mathrm{d}t \\ &= \int v(X_t)\,dW_t + \int \left(\frac{1}{2}|v(X_t)|^2 + V(X_t)\right)\mathrm{d}t - \lambda T.\end{aligned} \tag{A.15}$$

A closer look at the normalisation constant $\mathcal{N}$ in eq. (A.13) gives rise to the boundary term. The normalisation is given by

$$\mathcal{N} = \frac{\tilde{\psi}(r_T, T)}{\bar{\psi}(r_0, 0)}, \tag{A.16}$$

where $\tilde{\psi}(r_t, t)$ is the solution to the backwards imaginary time Schrödinger equation [6], and is related to the distribution of the stochastic process in eq. (A.5) as

$$\pi(r, t) = \tilde{\psi}(r, t)\,\psi(r, t). \tag{A.17}$$

If we now take both distributions at initial time $\pi(r, 0)$ and terminal time $\pi(r, T)$ to be the ground state the normalisation constant becomes

$$\frac{\tilde{\psi}(r_T, T)}{\bar{\psi}(r_0, 0)} = e^{E_0 T}\frac{\varphi_0(r_T)}{\varphi_0(r_0)}, \tag{A.18}$$

and accounts for the boundary term and $E_0 T$ term in the Kullback-Leibler divergence.

A.3 Todorov cost

Optimal decision processes are formalised using Markov Decision processes, here we follow work of Todorov [75, 76] to fit the imaginary Schrödinger equation in this mould, we illustrate the concepts on the Ising model. An MDP is a 4-tuple (S, A, P_a, R_a):

- S: Is the *state space*, i.e. all possible configurations of the system s_k

- U: Is the *control space*, i.e. all possible single-spin flips in each configuration
- P_u: Is the probability $p(s^{(t+1)} = s'|s^{(t)} = s, u^{(t)} = u)$ of control u in state s leading to state s' in the next time step
- g: Is the cost received when moving from state s to s' due to u

The optimal decision/control problem is then given by the Bellman equation for the optimal cost-to-go function $v(s)$

$$v(s) = \min_u \left\{ \underbrace{\ell(s,u)}_{\text{immediate cost}} + \underbrace{\mathbb{E}_{s'\sim p(\cdot|s,u)}\left[v\left(x'\right)\right]}_{\text{expected cost of next state}} \right\}. \tag{A.19}$$

Todorov introduces a formalism where the agent does not perform specific symbolic actions (e.g. flips a certain spin) but is instead allowed to specify transition probabilities $u(s'|s)$. Formally this means

$$p\left(s' \mid s, u\right) = u\left(s' \mid s\right), \tag{A.20}$$

the agent reshapes the dynamics of the system as it wishes, but for this it pays a price depending on how much it changes the dynamics. In absence of controls u the system follows *passive dynamics* $p(s'|s)$ which correspond to the first term in eq. (3.42) of the stoquastic Hamiltonian. The cost is thus

$$\ell(s,u) = \underbrace{q(s)}_{\text{state cost}} + \underbrace{D_{\mathrm{KL}}((\cdot \mid s)\|p(\cdot \mid s))}_{\text{control cost}}. \tag{A.21}$$

Optimal control problem in this form can be linearised in terms of the *desirability* function $z(s,t) = \exp(-v(s,t))$, yielding optimal dynamics v'

$$v'(s_j \mid s_k) = \frac{p(s_j \mid s_k) z(s_j)}{\sum_{s_l} p(s_l \mid s_k) z(s_l)}, \tag{A.22}$$

with

$$z(s_k, t) = e^{-q(s_k)} \sum_{s_j} p(s_j \mid s_k) z(s_j, t+1) \tag{A.23}$$

and it is this linear equation, what we can connect to the imaginary time Schrödinger equation. We start by transforming the MDP into continuous time (transition probabilities

to rates $p, u \longrightarrow \Gamma, \Gamma^{(v)}$) as

$$p(s_j \mid s_k) = \begin{cases} 1 - \Delta t \sum_{s_l} \Gamma_{s_k \to s_l} & s_j = s_k \\ \Delta t \Gamma_{s_k \to s_j} & s_j \neq s_k \end{cases}, \quad \text{and} \quad u(s_j \mid s_k) = \begin{cases} 1 - \Delta t \sum_{s_l} \Gamma^{(v)}_{s_k \to s_l} & s_j = s_k \\ \Delta t \Gamma^{(v)}_{s_k \to s_j} & s_j \neq s_k \end{cases} \tag{A.24}$$

and setting $q(s_j) = \Delta t V(s_j)$ eq. (A.23) becomes

$$\begin{aligned} z(s_k, t) &= e^{-\Delta t V(s_k)} \left[\underbrace{z(s_k, t + \Delta t) - \Delta t \sum_{s_l \neq s_k} \Gamma_{s_k \to s_l} z(s_k, t + \Delta t)}_{\text{from } s_j = s_k} + \underbrace{\Delta t \sum_{s_j \neq s_k} \Gamma_{s_k \to s_j} z(s_j, t + \Delta t)}_{\text{from } s_j \neq s_k} \right] \\ &= \left[1 - \Delta t V(s_k) + \ldots \right] \cdot \left[z(s_k, t + \Delta t) - \sum_{s_j \neq s_k} \Gamma_{s_k \to s_j} \left[z(s_j, t + \Delta t) - z(s_k, t + \Delta t) \right] \right] \end{aligned} \tag{A.25}$$

keeping only the first order in Δt and dropping the unnecessary $\neq$ in the sum

$$\frac{z(s_k, t) - z(s_k, t + \Delta t)}{\Delta t} = V(s_k) z(s_k, t + \Delta t) - \sum_{s_j} \Gamma_{s_k \to s_j} \left[z(s_j, t + \Delta t) - z(s_k, t + \Delta t) \right] \tag{A.26}$$

taking limit $\Delta t \longrightarrow 0$ finally gives

$$-\frac{\mathrm{d} z(s_k, t)}{\mathrm{d} t} = V(s_k) z(s_k, t) - \sum_{s_j} \Gamma_{s_k \to s_j} \left[z(s_j, t) - z(s_k, t) \right], \tag{A.27}$$

it is the imaginary time Schrödinger equation (3.61).
The D_{KL} in the loss $\ell(s, v)$ is expressed in the same manner

$$\begin{aligned} D_{\mathrm{KL}}(v(\cdot \mid s_k) \| p(\cdot \mid s_k)) &= \left[1 - \Delta t \sum_{s_l} \Gamma^{(v)}_{s_k \to s_l} \right] \log \left[\frac{1 - \Delta t \sum_{s_l} \Gamma^{(v)}_{s_k \to s_l}}{1 - \Delta t \sum_{s_l} \Gamma_{s_k \to s_l}} \right] + \Delta t \sum_{s_j \neq s_k} \Gamma^{(v)}_{s_k \to s_j} \log \left[\frac{\Gamma^{(v)}_{s_k \to s_j}}{\Gamma_{s_k \to s_j}} \right] \\ &= \Delta t \sum_{s_j \neq s_k} \Gamma^{(v)}_{s_k \to s_j} \underbrace{\left(\log \left[\frac{\Gamma^{(v)}_{s_k \to s_j}}{\Gamma_{s_k \to s_j}} \right] + \frac{\Gamma_{s_k \to s_j}}{\Gamma^{(v)}_{s_k \to s_j}} - 1 \right)}_{\text{Itakura–Saito divergence } D_{\mathrm{IS}}(\Gamma^{(v)}_{s_k \to s_j}, \Gamma_{s_k \to s_j})}, \end{aligned} \tag{A.28}$$

If we consider an ensemble of systems in state s_k there is $\Delta t \Gamma^{(v)}_{s_k \to s_j}$ probability of transitioning $s_k \longrightarrow s_j$ in the next time increment, meaning that the ensemble contribution to the D_{KL} each time increment is $\sum_{s_j} \Delta t \Gamma^{(v)}_{s_k \to s_j} D_{\mathrm{IS}} \left(\Gamma^{(v)}_{s_k \to s_j}, \Gamma_{s_k \to s_j} \right)$, thus we can express the Kullback-Liebler

divergence as

$$D_{\mathrm{KL}} = \mathop{\mathbb{E}}_{\Sigma_{[0,t]}=k_t\sim\Gamma^{(v)}}\left[\sum_n D_{\mathrm{IS}}\left(\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}, \Gamma_{k^{(n)}\to k^{(n+1)}}\right)\right]. \tag{A.29}$$

A.4 Probabilistic interpretation of Todorov's cost function

To find the Radon-Nikodym derivative between $\mathbb{P}_v$ and $\mathbb{P}_{\mathrm{FK}}$ we proceed in the same way as in the continuous case, by first finding the respective RN derivatives with the passive process. From the discrete space Feynman-Kac formula (3.45) follows

$$\log\left(\frac{\mathrm{d}\mathbb{P}_0}{\mathrm{d}\mathbb{P}_{FK}}(k(t))\right) = \int V(k(t))dt - E_0 T - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right), \tag{A.30}$$

and by using the Girsanov theorem equivalent for discrete space[1] we obtain

$$\log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_0}(k(t))\right) = \int \sum_{l\neq k(t)}\left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l}\right)dt + \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right). \tag{A.31}$$

We combine both

$$\log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}\right) = \log\left(\frac{\mathrm{d}\mathbb{P}_v}{\mathrm{d}\mathbb{P}_0}\frac{\mathrm{d}\mathbb{P}_0}{\mathrm{d}\mathbb{P}_{\mathrm{FK}}}\right) = \tilde{\ell} + \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right) - E_0 T - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})}\right), \tag{A.32}$$

with

$$\tilde{\ell} = \int\left[V(k(t)) + \sum_{l\neq k(t)}\left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l}\right)\right]\mathrm{d}t. \tag{A.33}$$

To see that zero $D_{KL}(\mathbb{P}_v\|\mathbb{P}_{\mathrm{FK}})$ coincides with rates that minimize Todorov's cost, we need

$$\mathop{\mathbb{E}}_{\mathbb{P}_v}\left[\sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}}\right)\right] = \mathop{\mathbb{E}}_{\mathbb{P}_v}\left[\int\sum_{l\neq k(t)}\Gamma^{(v)}_{k(t)\to l}\log\left(\frac{\Gamma^{(v)}_{k(t)\to l}}{\Gamma_{k(t)\to l}}\right)\mathrm{d}t\right], \tag{A.34}$$

which holds because the expectation of a contribution of a single step of the trajectory $k^{(n)}\to k^{(n+1)}$ is equivalent to an ensemble average starting from the same state weighted by the probability of jump $\Gamma^{(v)}_{k(t)\to l}\Delta t$, this holds separately for each step in the trajectory and

[1] see proposition 2.6 in Appendix 1 of [45]

by writing $\sum_{t_i} \cdots \Delta t_i \longrightarrow \int \cdots \mathrm{d}t$ we obtain above equality. The KL divergence then becomes

$$D_{\mathrm{KL}}(\mathbb{P}_v \| \mathbb{P}_{\mathrm{FK}}) = \mathop{\mathbb{E}}_{\mathbb{P}_v}\left[\int V(k(t)) + \sum_{l \neq k(t)} \left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l} \right) \mathrm{d}t + \sum_n \log\left(\frac{\Gamma^{(v)}_{k^{(n)}\to k^{(n+1)}}}{\Gamma_{k^{(n)}\to k^{(n+1)}}} \right) - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})} \right) \right] - E_0 T \tag{A.35}$$

$$D_{\mathrm{KL}}(\mathbb{P}_v \| \mathbb{P}_{\mathrm{FK}}) = \mathop{\mathbb{E}}_{\mathbb{P}_v}\left[\int V(k(t)) + \sum_{l \neq k(t)} \left(\Gamma_{k(t)\to l} - \Gamma^{(v)}_{k(t)\to l} \right) + \Gamma^{(v)}_{k(t)\to l} \log\left(\frac{\Gamma^{(v)}_{k(t)\to l}}{\Gamma_{k(t)\to l}} \right) \mathrm{d}t - \log\left(\frac{\varphi(k^{(N)})}{\varphi(k^{(0)})} \right) \right] - E_0 T, \tag{A.36}$$

which agrees with Todorov.

Appendix B

On the distribution of logRN variance