Research in Industrial Projects for Students

Sponsor

AMD

Final Report

Using Machine Learning as an Alternative Wave Function Ansatz to Improve Variational Monte Carlo

Student Members

Brett Leroux (Project Manager), *University of California, Davis*
beleroux@ucdavis.edu

Alberto Acevedo, *University of Arizona*
albertoacevedo@math.arizona.edu

Michael Curry, *University of Maryland*
curry@cs.umd.edu

Academic Mentor

Shantanu Joshi, sjoshi1@g.ucla.edu

Sponsoring Mentors

Nicholas Malaya, Nicholas.Malaya@amd.com

19 August 2020

This project was jointly supported by AMD and NSF Grant DMS 1440415.

Abstract

Solutions to the Schrödinger equation can be used to predict the electronic structure of molecules and materials and therefore infer their complex physical and chemical properties. One area of application is the design of semiconductors for computer chips. Variational Quantum Monte Carlo (VMC) is a technique that can be used to solve the weak form of the Schrödinger equation. Applying VMC to systems with N electrons involves evaluating the determinant of an N by N matrix. The evaluation of this determinant scales as $O(N^3)$ and is the main computational cost in the VMC process. In this work we propose an alternative VMC technique based on the Vandermonde determinant. The Vandermonde determinant is a product of pairwise differences and so evaluating it scales as $O(N^2)$. Therefore, our approach reduces the computational cost by a factor of N.

We implemented VMC using the new low cost approach in PyTorch and compared its use in approximating the ground state energy of various quantum systems against existing techniques, starting with the one-dimensional particle in a box and moving on to more complicated atomic systems with multiple particles. We also use the Vandermonde determinant as a part of PauliNet, a deep-learning architecture for VMC. While the new method obtains a reasonable approximation for wavefunctions of atomic systems, it does not reach the accuracy of the Hartree-Fock method that relies on the standard determinant. We observed that while the use of neural networks in VMC can result in highly accurate solutions, further new approaches are needed to best balance computational cost with accuracy.

Acknowledgments

We are very thankful for the support and encouragement of the IPAM staff and crew, in particular for Dr. Susana Serna (Director of RIPS, IPAM), Dr. Christian Ratsch (Deputy Director, IPAM), Dr. Dimitri Shlyakhtenko (Director of IPAM) and the IPAM staff (Neli Petrosyan, Parama Sigurdsen, Kayleigh Steele, David Medina, and Jim Kimmick). We are thankful to IPAM for hosting this G-RIPS program and supporting our research and the NSF DMS grant DMS-1440415 for making this possible. A special thanks is reserved to our academic mentor Dr. Shantanu Joshi whose perpetual guidance has been fundamental to our progress.

We would also like to express gratitude to AMD, in particular to our industry mentor Dr. Nicholas Malaya for his insight and perspective on scientific problems. Many thanks for sponsoring and for funding this research.

Contents

Abstract **3**

Acknowledgments **5**

1 Introduction **11**

2 Primer on Quantum Mechanics **13**
2.1 Schrödinger's equation 13
2.2 The complexity of the quantum many-body problem 14
2.3 Pauli exclusion principle 14

3 Primer on Variational Monte Carlo for quantum systems **17**
3.1 Introduction 17
3.2 A Test Case: Variational Monte Carlo on the Hydrogen Atom 18
3.3 Antisymmetry for multi-electron systems 19

4 PauliNet **21**
4.1 Introduction 21
4.2 Description of PauliNet architecture 21
4.3 Additional details 24
4.4 Neural net architectural design choices: physical constraints 25

5 Vandermonde and Slater Ansatz on Simple Systems **27**
5.1 Introduction 27
5.2 Antisymmetry: Slater and Vandermonde determinants 27
5.3 Physical Systems 28
5.4 Experimental results 28

6 Modifications to PauliNet **35**
6.1 Vandermonde determinants 35

7 Challenges **39**
7.1 Introduction 39
7.2 Enforcement of boundary conditions 39
7.3 Negative results for convolutional neural networks 39
7.4 Improved basis functions for compatibility with the Vandermonde determinant 40

8 Conclusion **43**

REFERENCES

Selected Bibliography Including Cited Works **45**

List of Figures

3.1 Gradient descent for the hydrogen atom converges to $\alpha = 1$. 19
3.2 Hydrogen atom energy estimates for various parameter values 19

4.1 A schematic diagram of the PauliNet architecture taken from [10]. 22

5.1 The ground state and first excited state for a particle in a 1-D box. The two wave functions are orthogonal. 29
5.2 Orthogonal particle in a box wave functions 30
5.3 Two particles in a box. Gradient descent for the Slater Ansatz 31
5.4 Convergence plot for the two-fermions-in-a-box system near the ground state energy 31
5.5 Convergence curve for the Helium atom problem with the traditional Slater ansatz with and without a cusp term. 32
5.6 Convergence curve for the Helium atom problem with the Vandermonde ansatz with and without a cusp term. 33
5.7 Convergence curves for Slater and Vandermonde ansatz for Lithium. 33

6.1 Two slices of the PauliNet wavefunction for Lithium hydride 37
6.2 Two slices of the PauliNet wavefunction for Berrylium. 37
6.3 Norm of difference in wavefunction outputs evaluated at 3000 random points, every 50 training iterations. 37

Chapter 1

Introduction

Advanced Micro Devices (AMD) is a semiconductor company that designs microprocessors for the consumer and enterprise markets. Founded in 1969, it is one of the companies that originally turned the southern Bay Area into "Silicon Valley". Recently, AMD has won the contract in conjunction with Cray to build the Oak Ridge National Laboratory's upcoming exascale supercomputer, Frontier.

Supercomputers are essential tools for complex scientific discoveries. The simulations performed on these machines span a wide range of applications across fundamental science, such as cosmology or turbulent flows, and applied engineering, such as material science and storm surge modeling. At this time, high performance computing is undergoing a radical transformation from largely homogeneous clusters of CPUs to heterogeneous machines with specialized accelerators, particularly GPUs. This paradigm shift requires that algorithms and software evolve to leverage the specialized hardware in these systems.

Coming online in 2021, Frontier is expected to be the largest supercomputer ever constructed and will exceed 1.5 exaflops of peak computational performance. This will be achieved using both CPUs and GPUs designed by AMD. Ensuring application readiness from Day-0 for a massively parallel, heterogeneous machine of this size is a challenging task. One application targeted for Frontier is QMCPACK [13]. QMCPACK is a quantum Monte Carlo code that leverages stochastic integration to solve the weak formulation of the Schrödinger equation. Solutions to Schrödinger's equation describe not only molecular, atomic, and subatomic systems, but also macroscopic systems, and are used to predict the electronic structure of chemical, physical, and material problems of interest. An important problem is the simulation of spin glass systems used in quantum annealing. Such applications are expected to massively aid the process of quantum computation. The QMCPACK application is expected to scale to the entire machine.

The proposed problem. QMCPACK uses the variational Monte Carlo method to estimate the ground state of quantum systems. Variational Monte Carlo is a scheme whereby the wavefunction of the system is represented approximately using an initial guess also known as the *ansatz*. Samples are drawn from the ansatz using a Monte Carlo sampling scheme followed by an optimization process over the parameters of the ansatz to minimize the wave function energy on the observed samples. This process continues until the ground state is approximately reached.

Since the the ansatz in this situation is a parametric function, it is natural to wonder whether techniques from machine learning can be used in designing an ansatz for VMC. While QMCPACK does not currently use machine learning techniques, there are several promising recent works demonstrating the use of deep neural networks as powerful function

approximators in the context of the Variational Monte Carlo algorithm used by QMCPACK [19, 10]. It is an open question whether or not techniques like these will ultimately become standard, and whether QMCPACK and other mainstream, high performance computing libraries will begin using them in lieu of their existing ansatzes, but in our view, this is a real possibility.

Our team's approach.

We aim to investigate the feasibility of machine learning approaches to produce new and powerful forms of ansatzes. The use of neural networks to generate ansatzes requires a careful design of the network architecture to enforce various physically motivated properties. We focus on one of these in particular: the need to enforce a property known antisymmetry due to the *Pauli exclusion principle*. The traditional method for enforcing antisymmetry uses a technique known as the Slater determinant, which incurs a significant computational cost for the ansatz in use. Here, we focus on a different technique for enforcing antisymmetry by using the Vandermonde determinant.

We explore two avenues to investigate this technique. First, we build our own variational Monte Carlo solution in order to explore the properties of the Slater and Vandermonde determinants in simple, well-understood systems. Second, we adapt PauliNet, a complex neural-network-based ansatz from the literature [10] by modifying it to use the Vandermonde determinant.

This report is organized as follows. In chapter 2, we provide some necessary background information on quantum mechanics. Chapter 3 outlines how the variational Monte Carlo technique is applied to quantum systems. We illustrate the technique by applying it to one of the simplest quantum systems, the hydrogen atom. At the end of this chapter we provide more details about antisymmetry and the Slater determinant. Chapter 4 outlines the existing PauliNet architecture and provides some brief background on neural networks. Chapter 5 defines and describes the Vandermonde determinant and demonstrates our comparisons of Vandermonde and Slater determinants using our own implementations on simple systems. Chapter 6 demonstrates our modifications of PauliNet, in particular the use of the Vandermonde determinant, and the challenges required to ensure enforcement of correct physical conditions. Finally, chapter 7 discusses the challenges and limitations of our approach, and conjectures about future improvements to the neural network based ansatz.

Chapter 2

Primer on Quantum Mechanics

2.1 Schrödinger's equation

The Schrödinger equation is a partial differential equation used to deduce the dynamics of any isolated quantum system. It is given by

$$i\hbar\partial_t \left|\psi(t)\right\rangle = H|\psi(t)\rangle\,, \tag{2.1}$$

where the *kets* i.e. $|\psi(t)\rangle$, are elements of an appropriate Hilbert space and H is a Hermitian operator known as the Hamiltonian of the respective system. More formally, $|\psi(t)\rangle \in \mathcal{H}$ is a separable Hilbert space that can be either finite or infinite dimensional. Assuming that H is time independent, the solution of the differential equation above is simply

$$\left|\psi(t)\right\rangle = U(t)\left|\psi(0)\right\rangle, \tag{2.2}$$

where

$$U(t) = e^{\frac{-it}{\hbar}H}. \tag{2.3}$$

This operator is referred to as the time evolution operator and $|\psi(0)\rangle$ is the initial state of the system.

The superposition principle says that our initial state can take the following general form,

$$\psi(0)\rangle = \sum_i c_i|\phi_i\rangle_S, \tag{2.4}$$

where c_i are complex numbers and $\{|\phi_i\rangle\}_i$ is an orthonormal basis for the system's Hilbert space $\mathcal{H}$, consisting of eigenvectors of some Hermitian operator over $\mathcal{H}$. Knowledge of the spectrum of H is important if one is to make sense of the time evolution Equation (2.2), which is an eigenvalue problem $H|\psi\rangle = E|\psi\rangle$. For problems with a continuum configuration space this eigenvalue problem is more concretely a second order differential equation.

As an example, for the Hydrogen atom the Hamiltonian is given by

$$H = -\frac{\hbar^2}{2\mu}\nabla^2 - \frac{e^2}{r}.$$

The corresponding time independent Schrodinger equation is thus obtained as

$$-\frac{\hbar^2}{2\mu}\nabla^2\Psi(\vec{r}) - \frac{e^2}{r}\Psi(\vec{r}) = E\Psi(\vec{r}),$$

with $\Psi \in L^2(\mathbb{R})$. This is one of the few systems where an analytic solution is known. The Hydrogen atom is essentially a one body problem since, if the position of the nucleus is fixed, the only relevant atomic particle is the electron.

In general, the N-body electron problem for the time independent case will yield a differential equation of the form

$$\sum_{i=1}^{N}(-\frac{\hbar^2}{2m}\nabla_i^2) + V_{e-n}(\vec{r_1},...,\vec{r_N}) + V_{e-e}(\vec{r_1},...,\vec{r_N})\}\Psi(\vec{r_1},...,\vec{r_N}) = E\Psi(\vec{r_1},...,\vec{r_N}) \quad (2.5)$$

2.2 The complexity of the quantum many-body problem

The Hilbert space of a quantum system with N "distinguishable" particles (a system with N electrons) can be described by the tensor product of the N corresponding Hilbert spaces.

$$\mathcal{H}^N = \bigotimes_{i=1}^{N} \mathcal{H}_i.$$

For a single particle in three dimensional space, the state is described by a complex-valued wave function $\Psi(\vec{r})$ of the position $\vec{r}$ of the particle, while N distinguishable particles are described by a complex-valued wave function $\Psi(\vec{r_1},...,\vec{r_N})$ of each of the respective positions of the particles. For a continuum configuration as seen here we have infinite dimensional Hilbert spaces. Computationally what this means is that we must approximate the Hilbert spaces associated with each of the individual particles with a finite dimensional Hilbert space. Assuming we take a d dimensional subspace for each of the N particles we are left with a d^N dimensional subspace. This exponential scaling of the Hilbert spaces is a huge hurdle and very quickly it becomes impossible to work with higher dimensional N body systems.

Even in the simplest of cases, where the Hilbert spaces of each of our N particles is a two dimensional Hilbert space we quickly reach the computational limits of computers.

Let $N = 40$ and let $dim(\mathcal{H}_i) = 2$. This means that the dimension of the N body system total Hilbert space will have dimension $2^{40} \approx 10^{12}$. Then a complex vector representing our space would take over 2 terabytes of space in memory. Having access to greater computational power is essential to tackle large N body problems. However, if we limit ourselves to modest values of N we still run into hurdles when trying to find the ground state energy and its corresponding wavefunction.

The Hilbert space we are searching within is very large and an indirect method of distilling the correct ground state energy and wavefunction is preferable. As we will see in the next chapter, variational Monte Carlo is such a method. Based on physical symmetries we may pose a parametrized ansatz that structurally encapsulates the physics we expect the corresponding system to exhibit and optimize over the parameters using Monte Carlo methods. If our ansatz is good, quick convergence to the true ground state of the studied system can be hoped for.

2.3 Pauli exclusion principle

Fermionic systems obey a property known as the Pauli exclusion principle, an empirical law that states that two or more identical fermions cannot occupy the same quantum state

within a quantum system simultaneously. The particles that follow this rule have half-integer spin, a subject beyond the scope of this paper but the common denominator of all Fermionic systems following Pauli's exclusion principle. One very important particle with half-integer spin is the electron. It is due to this Pauli exclusion property that electrons form the so-called electron shells around a given nucleus to yield all of the elements of matter.

Because matter is essentially made up of electrons one needs to make sure to incorporate the Pauli exclusion principle into the framework of any scheme designed to study atomic, molecular and nanoscale bulk matter type problems. Later we will see that an effective way to force the Pauli exclusion principle is via the requirement that our solutions to the Schrödinger equation be antisymmetric. To elucidate on the requirement of antisymmetry consider a two-fermion system and assume that the solution to the corresponding ground state energy eigenvalue/eigenstate problem had the form $\Psi(x_1, x_2) = \Psi_a(x_1)\Psi_b(x_2)$. Here Ψ_a and Ψ_b are the states of fermions one and two respectively. We observe that the wave-function is nonzero when both fermions are in the same position, more importantly we note that the wave function will also be nonzero, in general, if both particles are taken to be in the state Ψ_1. This clearly violates the exclusion principle. Now, if we were to rewrite our wave function as $\Psi(x_1, x_2) = \Psi_a(x_1)\Psi_b(x_2) - \Psi_a(x_2)\Psi_b(x_1)$ then we have $\Psi(x_1, x_2) = 0$ for any input whenever a=b. Furthermore the antisymmetry property yields $\Psi(x_1, x_2) = -\Psi(x_2, x_1)$. This antisymmetry property becomes important, and investigating ways to impose it on otherwise general and flexible and parametric ansatz is the main subject of our research.

Chapter 3

Primer on Variational Monte Carlo for quantum systems

3.1 Introduction

One of the techniques used by QMCPACK to estimate the ground state of a quantum system is variational Monte Carlo. We assume that we have access to the Hamiltonian H that characterizes the system. We would like to approximate the ground state wavefunction ψ which is the solution of the time-independent Schrödinger's equation

$$H\left|\psi\right\rangle = E\left|\psi\right\rangle \tag{3.1}$$

for the lowest-energy state of the system. Then, roughly speaking, the variational Monte Carlo (VMC) procedure consists of the following steps (as described in [19]):

1. Start with some initial variational ansatz $|\psi_\alpha(x)\rangle$ from some class of functions parameterized by α.

2. Using Markov-chain Monte Carlo (MCMC) techniques, sample from the probability distribution implied by this wavefunction: $\pi(x) \propto ||\psi_\alpha(x)\rangle|^2$.

3. Minimize the expected value of the energy of the system

$$\mathbb{E}[E] = \frac{\langle\psi_\alpha|H|\psi_\alpha\rangle}{\langle\psi_\alpha|\psi_\alpha\rangle} = \frac{\int \psi_\alpha^*(x) H \psi_\alpha(x) dx}{\int \psi_\alpha^*(x)\psi_\alpha(x)dx}. \tag{3.2}$$

   Using the samples $x_i \sim \pi(x)$, we can approximate this expected value by computing

$$\frac{1}{N}\sum_{i=1}^{N} \psi_\alpha^*(x) H \psi_\alpha(x) \tag{3.3}$$

   and approximate the variance by computing

$$\frac{1}{N}\sum_{i=1}^{N} (\psi_\alpha^*(x) H \psi_\alpha(x) - \mathbb{E}[E])^2 \tag{3.4}$$

   and optimize the parameters α of the ansatz to minimize some combination of these losses.

3.1.1 Variational Ansatz

The most common family of variational ansatzes is the Slater-Jastrow family [6]. In brief, the idea is to organize a set of individual wavefunctions, each with its own parameters, into a matrix, whose determinant is computed. The overall wavefunction for the system is represented as a weighted sum of a few such determinants, multiplied by an additional term (the Jastrow factor) which represents pairwise interactions. However, more general parametric function approximators can be used for this purpose as long as they respect certain required symmetries; [19] proposes a neural network architecture that achieves this goal. There is more discussion of the Slater determinant in Section 3.3 below.

3.1.2 Markov-Chain Monte Carlo Sampling

Here we provide some more details on how machine learning can be used in the sampling process. Markov-Chain Monte Carlo methods sample a sequence of points such that their stationary distribution converges to a target distribution π. In cases where π is not explicitly known, we still need to know how to sample from its unnormalized probability density. Getting the next sample point is a two step process [9, 1]:

1. A point is sampled from some **proposal distribution** around the current point, typically denoted Q.
2. An **acceptance ratio** is computed, and the point is accepted with probability

$$\min\left(1.0, \frac{\pi(x_{t+1})}{\pi(x_t)}\frac{Q(x_{t+1} \to x_t)}{Q(x_t \to x_{t+1})}\right), \tag{3.5}$$

   otherwise the point is rejected. This is known as the Metropolis-Hastings method.

Traditionally, the proposal distribution Q is relatively simple – a Gaussian, or else derived from the target density, as in Hamiltonian Monte Carlo. The proposal distribution Q can be chosen very freely, and as long as the Metropolis-Hastings rejection step remains correct, the chain will still converge to the target.

3.2 A Test Case: Variational Monte Carlo on the Hydrogen Atom

The hydrogen atom is a special example of a quantum system because it is one of the few systems which admits a simple closed-form analytic solution. For this reason, it is a good testing ground for quantum Monte Carlo methods. First we review the known analytic solution of the Schrödinger equation for the hydrogen atom. Then we verify our implementation of variational Monte Carlo by using it to recover the known analytic solution for the hydrogen atom.

The time-independent Schrödinger equation for the hydrogen atom is

$$\hat{H}\Psi = -\frac{\hbar^2}{2\mu}\nabla^2\Psi - \frac{e^2}{r}\Psi = E\Psi.$$

The ground state energy of the hydrogen atom is known to be $E = -13.6eV$. In different units, the energy is $\lambda = -1/2$ with the exact ground state radial wavefunction $u(r) = re^{-r}$.

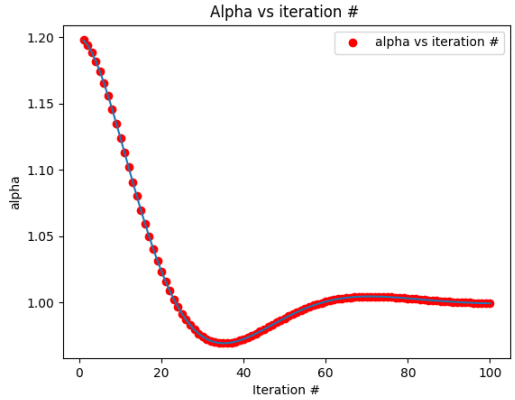

Figure 3.1: Gradient descent for the hydrogen atom converges to $\alpha = 1$.

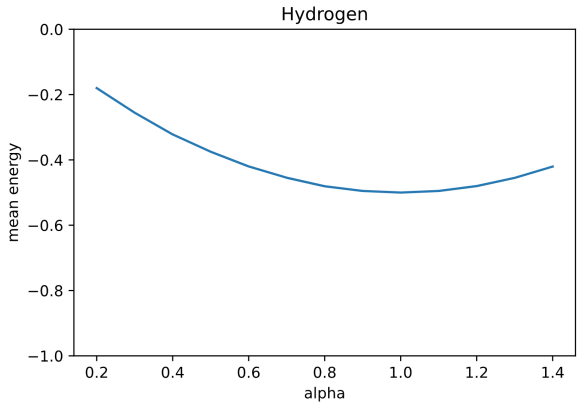

Figure 3.2: Hydrogen atom energy estimates for various parameter values

In applying VMC to the hydrogen atom, we use the ansatz $\Psi(\alpha, r) = \alpha r e^{-\alpha r}$. So the correct parameter value is $\alpha = 1$. We can confirm this solution using our implementation in two ways. First, we estimate the energy of the system for various values of α. This is shown in Figure 3.2. One can see that the lowest energy is around $\alpha = 1$. Second, Figure 3.1 shows how gradient descent converges to the correct value $\alpha = 1$.

3.3 Antisymmetry for multi-electron systems

Next we discuss general ideas in the design of an ansatz for many-electron systems. Much of this discussion is adapted from [11].

The one-electron hydrogen ansatz above satisfies a number of desirable physical properties:

- The electron cannot collide with the nucleus – the wavefunction vanishes as the nucleus is approached.
- As the electron distance goes to infinity, the wavefunction decays to 0.

As we move to studying many-electron systems, we want to preserve these properties for each electron, as well as ensuring some new properties hold:

- It should satisfy **electron-electron cusp** conditions, meaning two electrons should be unlikely to collide.
- As discussed in Section 2.3, the ansatz should satisfy **antisymmetry**: if two electrons of the same spin have their states exchanged, only the sign of the wavefunction should flip: $\Psi(\cdots, \vec{r}_i, \cdots, \vec{r}_j, \cdots) = -\Psi(\cdots, \vec{r}_j, \cdots, \vec{r}_i, \cdots)$.

A naive ansatz for many-electron systems is known as the Hartree product. In this case, there is one separate wavefunction ansatz φ_i for each electron: it can take a form similar to the hydrogen ansatz above – the unrealistic assumption is that electrons do not interact with each other. Then the ansatz for the whole system becomes:

$$\Psi(r_1, \cdots, r_N) = \prod_i \varphi_i(r_i) \tag{3.6}$$

Unfortunately, this ansatz does not satisfy antisymmetry. A classic approach to remedy this problem is known as the Slater determinant. One still has N wavefunctions φ_j, but they are arranged in a matrix whose determinant is given by

$$\Psi(r_1, \cdots, r_N) = \begin{vmatrix} \varphi_1(r_1) & \varphi_2(r_1) & \cdots & \varphi_N(r_1) \\ \varphi_1(r_2) & \varphi_2(r_2) & \cdots & \varphi_N(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(r_N) & \varphi_2(r_N) & \cdots & \varphi_N(r_N) \end{vmatrix}. \tag{3.7}$$

Exchanging two particles exchanges two rows of this matrix; this switches the sign by the determinant property exactly as desired. Because each term in the determinant is a product of basis functions, any boundary and nuclear cusp conditions enforced by the φ_j are preserved. This determinant technique can be extended by multiplying the entire expression by functions that are symmetric in the r_i, known in this context as a Jastrow factor. This can be used to increase the flexibility of the ansatz and enforce further physical constraints.

The Slater determinant is powerful and it is used in almost all ansatzes for many-electron quantum systems, including PauliNet and those implemented in QMCPACK. However, it is not the only approach for enforcing antisymmetry. Later in the report, we will discuss another technique, the Vandermonde determinant.

Chapter 4

PauliNet

4.1 Introduction

PauliNet [10] is a neural network architecture for solving N-body electron systems, i.e. molecular and atomic systems. It uses deep learning techniques to approximate electronic wavefunctions. Although it uses neural networks, it is also physically inspired and borrows ideas from traditional techniques. In particular, it effectively incorporates the Hartree-Fock method which is a baseline analytic approximate solution to a simplified version of the appropriate Schrödinger equation. Below, we provide a high-level overview of the architecture of PauliNet, and the procedure for training it to minimize the ground-state energy. Then, we give more detailed explanations of specific important properties of its architecture.

4.2 Description of PauliNet architecture

PauliNet uses a Slater-Jastrow-backflow ansatz. Ansatzes of this general form have long been popular for variational Monte Carlo, and are widely used including in QMCPACK. PauliNet follows this basic structure but replaces various parts of it with neural networks. It constructs the ansatz in the following form:

$$\begin{aligned} \Psi_\theta(r) &= e^{\gamma(r)+J_\theta(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r)) \\ \tilde{\varphi}_\mu(r)_i &= \varphi_\mu(r_i) f_{\mu,\theta}(r)_i \end{aligned} \tag{4.1}$$

This expression is described in more detail below. A schematic diagram of the PauliNet architecture is in Figure 4.1.

- The μ denotes an index when there are multiple copies of certain basis sets or other functions.
- $\varphi_\mu(r)$ are Hartree-Fock functions, a well-known approximate solution derived by making simplifying assumptions to the physical problem until it becomes analytically tractable; they are essentially treated as inputs. They are modified to enforce nuclear cusp constraints, and they naturally enforce decay of the wavefunction to zero as electrons stray far from the nucleus.
- The terms J_θ and $f_{\mu,\theta}$ are learnable neural networks; the $f_{\mu,\theta}$ are the backflow factors which model electron-electron interactions and scale the basis functions; the J_θ is the

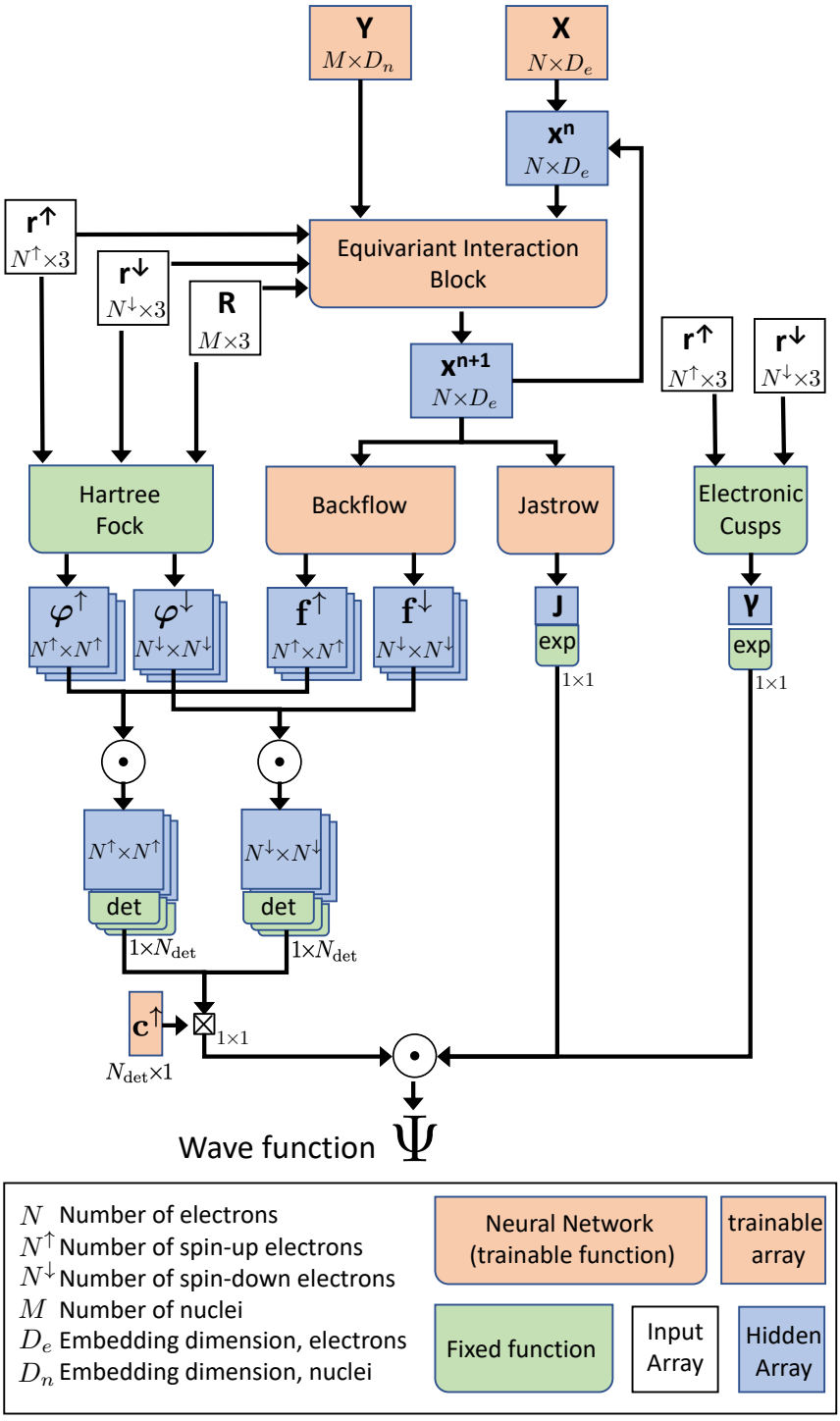

Figure 4.1: A schematic diagram of the PauliNet architecture taken from [10].

Jastrow factor which models global correlations between electrons but does not affect antisymmetry.

- The γ are fixed, physical cusp constraints: as electrons approach each other the wavefunction must obey certain physical behavior.

- The rescaled φ are combined using Slater determinants, split according to spin up ($\uparrow$) or spin down ($\downarrow$) electrons (antisymmetry only applies to electrons with the same spin).

- There may be multiple copies of each Slater determinant, which are indexed by p and combined in a weighted sum. We focus on the single-determinant case for most of our work, however.

- Implicit in the implementation of Equation (4.1) is the use of SchNet [21] to transform electron coordinate inputs into a 128-dimensional vector for input into f_θ and J_θ. This is discussed in section 4.3.1.

The architecture combines physically motivated inputs and architectural choices with flexible transformations represented by general function approximators.

4.2.1 PauliNet training procedure

PauliNet uses the same variational Monte Carlo procedure discussed in the previous chapter. First, points are drawn using an MCMC approach – specifically, Langevin Monte Carlo [24]. Then, stochastic gradient descent is performed using these points to minimize the energy for a few steps, and the process is repeated. In particular, the so-called "score function" estimator is used to approximate the gradient of the mean local energy with respect to the parameters:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{\boldsymbol{r}\sim|\Psi_\theta(\boldsymbol{r})|^2}\left[E_{\mathrm{loc}}[\Psi_\theta](\boldsymbol{r})\right] \\ \nabla_\theta\mathcal{L}(\theta) &= 2\mathbb{E}_{r\sim|\Psi_\theta|^2}\left[(E_{\mathrm{loc}}[\Psi_\theta](\boldsymbol{r}) - \mathcal{L}(\theta))\nabla_\theta \log|\Psi_\theta|\right] \end{aligned} \tag{4.2}$$

The expectation in these expressions is calculated by taking the mean on sampled points; in particular the second expression approximated in this way is an unbiased estimator for the true gradient.

The local energy here is computing by allowing the Hamiltonian to act on the wavefunction evaluated at a specific sample point. The Hamiltonian for an atomic system in general takes the following form, where i, j index electrons and I, J index nuclei with atomic number Z_I (this expression is from [19], see also the equivalent Equation (2.5)):

$$H = -\frac{1}{2}\sum_i \nabla_i^2 + \sum_{i>j}\frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} - \sum_{i,I}\frac{Z_I}{|\boldsymbol{r}_i - \boldsymbol{R}_I|} + \sum_{I>J}\frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|} \tag{4.3}$$

The terms, from left to right, are the kinetic energy (involving the Laplacian ∇_i^2 with respect to $\boldsymbol{r}_i$), the electron-electron potential energy, the electron-nucleus potential energy, and the nucleus-nucleus potential energy.

All gradients and the Laplacian operators ∇_i^2 here are computed using the automatic differentiation facilities in PyTorch.

4.3 Additional details

4.3.1 Description of modified SchNet

The modified SchNet used as a module within PauliNet takes in the set of electron coordinates and outputs a feature representation in an arbitrary space ($\mathbb{R}^{128}$ in practice) for each electron that reflects the interactions between them. It is a "message-passing" architecture: during a single evaluation of the wavefunction, each electron starts with a feature vector in a current state, and over some number of repeated rounds, its features are updated by incorporating information passed from the feature representations of other electrons.

In particular, at iteration n, each electron has some feature representation $\boldsymbol{x}_i^n$. Then at each round, $\boldsymbol{x}_i^{n+1}$ is generated according to the following rule, where $\boldsymbol{z}_i$ represent "messages" incoming to electron i, split according to incoming information from same spin electrons $(+)$, opposite spin electrons $(-)$ and the atomic nuclei (a):

$$\boldsymbol{x}_i^{n+1} = \boldsymbol{x}_i^n + \sum_{\pm} g_\theta^{(n,\pm)}(\boldsymbol{z}_i^{n,\pm}) + g_\theta^{(n,a)}(\boldsymbol{z}_i^{n,a}) \tag{4.4}$$

All terms with a θ are functions with learnable parameters. The messages from other electrons (split into same and opposite spin) are generated by taking the distance from each electron j, using a function e (defined in Equation (4.7)) to transform these distances into d-dimensional feature vectors (for our purposes, $d = 32$), and passing these distance features along with the current electron features $\boldsymbol{x}_j$ through neural networks:

$$\boldsymbol{z}_i^{(n+1,\pm)} = \sum_{j\neq i,\pm} w_\theta^{(n,\pm)}(e(|\boldsymbol{r}_i - \boldsymbol{r}_j|)) \odot h_\theta^n(\boldsymbol{x}_j^(n)) \tag{4.5}$$

A similar procedure is used to produce the messages from each atomic nucleus to the electron, except that the distance from the electron to each nucleus is considered instead, and there are fixed, learnable features $\boldsymbol{Y}_{\theta,I}$ per nucleus.

$$\boldsymbol{z}_i^{(n+1,a)} = \sum_{\text{nucleus } I} w_\theta^{(n,a)}(e(|\boldsymbol{r}_i - \boldsymbol{R}_I|)) \odot \boldsymbol{Y}_{\theta,I} \tag{4.6}$$

The fixed function e (adapted from [25]) that takes scalar distances to a d-dimensional has each dimension k of the output vector defined as follows (where r_c is an arbitrary cutoff parameter, and q_k spans $(0, 1)$ at even intervals for each k):

$$\begin{aligned} e_k(r) &= r^2 e^{-r-(r-\mu_k)^2/\sigma_k^2} \\ \mu_k &= r_c q_k^2 \\ \sigma_k &= \frac{1}{7}(1 + r_c q_k) \end{aligned} \tag{4.7}$$

To summarize, the components of the SchNet module that have learnable parameters are as follows:

- The message-receiving networks g_θ^+, g_θ^-, and g_θ^a, which take averaged messages from all electrons to the space of feature representations.
- The networks h_θ which takes feature representations to the space of messages, and w_θ^+, w_θ^-, w_θ^a, which takes distance representations to the space of messages.
- The fixed but trainable nucleus features $\boldsymbol{Y}_{\theta,I}$

- Additionally, the initial electron features before message passing begins are trainable and denoted $\boldsymbol{X}_{\theta,i}$.

After L rounds of message passing, one is left with informative feature representations $\boldsymbol{x}_i^L$ for each electron. These are the actual inputs used to the Jastrow and backflow networks, so that we compute $J_\theta(\sum_i \boldsymbol{x}_i^L)$ and $f_\theta(\boldsymbol{x}_i^L)_i$.

4.3.2 Enforcing antisymmetry

PauliNet uses the Slater determinant to enforce antisymmetry of the wavefunction. Asymptotically, computing the determinant of an N electron system is an $O(N^3)$ operation – this dominates computational costs when sampling, evaluating the local energy, or performing gradient descent. It may be possible to use other ways to enforce antisymmetry which are simpler and more computationally efficient. It may be also possible that ignoring the antisymmetry altogether does not harm effectiveness too much in certain systems.

The authors of [7] suggest replacing the commonly used Slater determinant with the Vandermonde determinant. The Vandermonde determinant takes the following general form:

$$\prod_{i<j}(\phi_i(X) - \phi_j(X)).$$

The authors of [7] note that the complexity of computing a Slater determinant is $O(N^3)$ while the complexity of computing a Vandermonde determinant is $O(N^2)$. Therefore, replacing the Slater determinants used by PauliNet with Vandermonde determinants could be a useful step towards a more efficient ansatz. This is precisely what we achieve in this work.

4.4 Neural net architectural design choices: physical constraints

The f_θ and J_θ are neural networks, as is the SchNet which generates their input, which in general may learn any arbitrary function. However, other parts of the PauliNet ansatz are carefully designed to enforce certain physical properties. The neural networks used in the ansatz thus must be architecturally constrained to avoid disturbing these properties. We discuss several of these issues below.

4.4.1 Electron cusps

The underlying Hartree-Fock orbitals φ enforce the nuclear cusp and boundary conditions. When multiplying them by the backflow factors f, we do not want to interfere with them. Likewise the term γ enforces the electron-electron cusp conditions; we do not want the Jastrow factor J to interfere with these. As such, it is necessary to ensure that where one electron is close to another or to the nucleus, the gradient of J_θ and f_θ with respect to the electron coordinates are zero: this is guaranteed simply by Equation (4.7), which ensures that the underlying feature representations and their derivatives are zero when the input distance is zero.

4.4.2 Invariance and equivariance

There are more physical properties that must be preserved. In order for the Slater determinant to properly enforce antisymmetry, the underlying functions must be permutation equivariant with respect to the electrons: in other words, swapping two electrons must only swap their respective outputs, but leave the values themselves the same. The Jastrow factor, on the other hand, is external to the determinant, so in order to preserve antisymmetry of the whole wavefunction it must be permutation invariant as well.

4.4.3 SchNet and permutation equivariance

The authors of PauliNet design their SchNet to ensure that permutation equivariance remains true of its outputs. In other words, swapping $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, the input electron coordinates, will swap the positions of $\boldsymbol{x}_i^L$ and $\boldsymbol{x}_j^L$ but will leave the values the same.

Because the backflow network f_θ is applied to each electron individually, the outputs of f_θ and therefore $\tilde{\varphi}$ will also be permutation equivariant. And because the inputs to J_θ are summed, its output will be the same regardless of their order, as long as the values do not change; thus it is also permutation invariant as desired.

In building and modifying the ansatz, these issues of permutation invariance and equivariance, and enforcement of boundary conditions, are important design considerations.

Chapter 5

Vandermonde and Slater Ansatz on Simple Systems

5.1 Introduction

As a testbed for specifying different ansatzes, we implemented our own variational Monte Carlo pipeline. This uses a simple Metropolis-Hastings sampler, and the standard "score function" stochastic gradient descent estimator, also used in PauliNet (Equation (4.2)). We experiment with a number of simple physical systems. In particular, we focus on systems where antisymmetry is a concern.

5.2 Antisymmetry: Slater and Vandermonde determinants

As mentioned in Section 3.3, when the Pauli exclusion principle is in play, it is standard to require that the wavefunction Ψ obey antisymmetry. The traditional way to enforce this is to employ a Slater determinant. However, computing a determinant has a complexity of $O(N^3)$, which is undesirable.

An alternate method of enforcing antisymmetry is to use the Vandermonde determinant. This has been proposed, for instance, in [7]. In brief, the Vandermonde determinant takes as input a map $(\varphi(r_1), \varphi(r_2), \cdots, \varphi(r_N))$ and takes the determinant of the following matrix:

$$\det_V = \begin{vmatrix} 1 & \varphi(r_1)^1 & \varphi(r_1)^2 & \cdots & \varphi(r_1)^{N-1} \\ 1 & \varphi(r_2)^1 & \varphi(r_2)^2 & \cdots & \varphi(r_2)^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi(r_N)^1 & \varphi(r_N)^2 & \cdots & \varphi(r_N)^{N-1} \end{vmatrix} \tag{5.1}$$

This is a determinant, and assuming the vector of φ is permutation equivariant (as it is in our applications), exchanging two inputs will exchange rows of the matrix, swapping signs of the determinant, so that antisymmetry is enforced.

The Vandermonde determinant has a nice property, however: one does not need to write out the full matrix and actually take its determinant; instead, it is possible to use the following expression, requiring only $O(N^2)$ operations:

$$\det_V = \prod_{i<j}(\varphi(r_i) - \varphi(r_j)), \tag{5.2}$$

Thus the Vandermonde determinant is a potentially efficient technique for enforcing antisymmetry with a lower computational cost than the widely-used Slater determinant.

5.3 Physical Systems

For our experiments, we considered the following physical systems:

- To verify our solutions, we considered the toy problem of two fermions in a box – a slight variation on a textbook problem in quantum mechanics.
- We also considered n-electron systems: Helium (2 electrons) and Lithium (3 electrons).

5.4 Experimental results

Two fermions in a box

The *two fermions in a box* system consists of two fermions (particles where the Pauli exclusion principle applies) in a box (finite volume infinite potential well) that the fermions are restricted to. The system is relatively simple and analytically tractable – the problem is that for the most naturally defined ansatzes the local energy does not depend on the particle configuration, making variational Monte Carlo totally unnecessary. As a concrete example, we consider the Schrödinger equation for the 1-D box case,

$$-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)\Psi(x_1, x_2) = E\Psi(x_1, x_2). \tag{5.3}$$

The time independent Schrödinger in this case amounts to the simple eigenvalue problem corresponding to the Laplacian. The Laplacian indeed has a discrete spectrum over a bounded domain. For a box of dimension 1-D and length L, the particles bounded to the domain [0, L], the eigenvalue/eigenvector pairs are the energies $E_{n_1,n_2} = \frac{\hbar^2\pi^2}{2L^2}(n_1^2 + n_2^2)$ with the corresponding states $\Psi_{n_1,n_2}(x_1, x_2) = \sqrt{\frac{2}{L}}\sin(\frac{n_1\pi x_1}{L})\sqrt{\frac{2}{L}}\sin(\frac{n_2\pi x_1}{L})$. Let us now calculate the corresponding local energy:

$$E_{loc} = \frac{1}{\Psi_{n_1,n_2}(x_1, x_2)}\{-\frac{\hbar^2}{2m}\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_1^2}\right)\Psi_{n_1,n_2}(x_1, x_2)\} = E_{n_1,n_2}. \tag{5.4}$$

This local energy is independent of configuration, therefore structurally for an ansatz will yield trivial results: VMC would be completely superfluous as samples from the wavefunction would serve no purpose. The same issues arise even after implementing antisymmetry via a Slater determinant. In order to test VMC, we devise a deliberately deficient ansatz, using both Slater and Vandermonde determinants. Below, we denote parameters by α_1, α_2, etc.

Letting our 1-D box be segment $[-1, 1]$ we have that $L = 2$. We may estimate the eigenstates with polynomial ansatz so long as the geometry is similar to that of the actual eigenstates, plotted on Figure 5.1, and the ansatz vanishes at the boundary, this is important

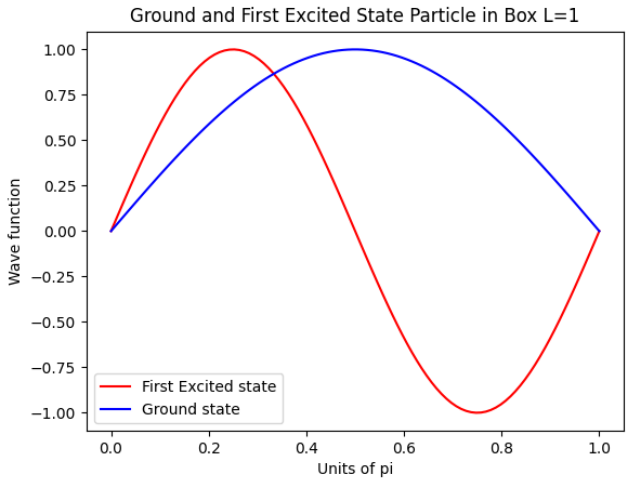

Figure 5.1: The ground state and first excited state for a particle in a 1-D box. The two wave functions are orthogonal.

since the particles being confined to the box have a zero probability of exiting it and hence a zero value for their corresponding wave function at the boundary.

For two fermions in a box we must make sure that the wave function satisfies antisymmetric conditions. Recall that the Pauli exclusion principle tells us that two particles can not occupy the same state. To enforce this we must find a function, $\Psi_{excited}(x_1)$ for particle one, that has the same geometry as the second excited state and use this in conjunction with the first excited state wave function,$\Psi_{ground}(x_2)$, to create a total ground state ansatz for our two particle system, i.e. $\Psi_{total} = \Psi_{excited}(x_1)\Psi_{ground}(x_2)$. A possible pair of ansatz are $\Psi_{excited}(x) = -(1-(x^2)^{\alpha_1})x$ and $\Psi_{ground}(x) = (1-(x^2)^{\alpha_2})$ both plotted in Figure 5.2 with $\alpha_1 = \alpha_2 = 2$. Finally, implementing a Slater determinant on the appropriate Slater matrix afforded by the above ansatz we may achieve antisymmetry.

Figure 5.3 shows the training results for the following ansatz, which is a 2-dimensional Slater determinant of the Slater matrix constructed from the ansatz $\Psi_{excited}$ and Ψ_{ground} discussed above:

$$\Psi(x_1, r_2) = (1-(x_1^2)^{\alpha_1})(1-(x_2^2)^{\alpha_2})x_2 - (1-(x_2^2)^{\alpha_1})(1-(x_1^2)^{\alpha_2})x_1 \tag{5.5}$$

Figure 5.4 shows the training results for another Slater-based ansatz, with slightly different basis:

$$\Psi(x_1, x_2) = (1-(x_1^2)^{\alpha_1})e^{-\alpha_2 x_2^2} - (1-(x_2^2)^{\alpha_1})e^{-\alpha_2 x_1^2} \tag{5.6}$$

as well as the results for a basic Vandermonde-based ansatz:

$$\Psi(x_1, x_2) = (1-x_1^2)^{\alpha_1}(1-x_2^2)^{\alpha_1}(\sin(\alpha_2 x1) - \sin(\alpha_2 x_2)). \tag{5.7}$$

In all cases we achieve convergence to the true ground state energy as seen in plots on Figure 5.3 and Figure 5.4. It is clear that using an ansatz that ensures that both of the particles are in states corresponding to different energy levels yields faster convergence. Figure 5.4 shows on the left this relatively rapid convergence to the true ground state energy, on the other hand, the Gaussian-type Slater ansatz on the left of Figure 5.4 converges about 20 iterations after our polynomial ansatz, Equation (5.5), has already converged. It is surprising nonetheless, that the ansatz used in the leftmost figure of Figure 5.4 converges.

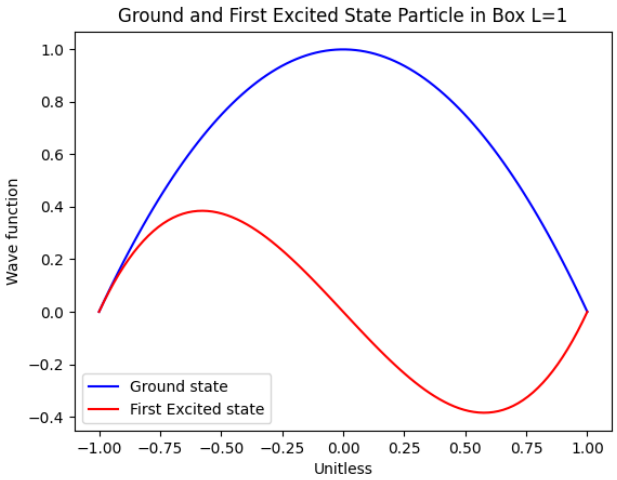

Figure 5.2: Orthogonal particle in a box wave functions. Here we see the respective ansatz for the first and excited state with their variational parameters α_1 and α_2 being taken at one. The desired feature here is orthogonality in order to aid us in the implementation of the Pauli exclusion principle, the amplitude of the ansatz for the first excited state not resembling that of the the true first excited state is unimportant.

The Gaussian terms do not satisfy the necessary boundary conditions but via VMC the parameters optimize to alleviate this issue. Despite the Vandermonde ansatz's ability to yield the fastest converging approximate energy it is the weakest of the ansatzes due to the relatively significant discrepancy between the attained approximate ground state energy and the true ground state energy.

N-electron systems

We now consider a few atomic systems. Due to the Pauli exclusion principle we will need to force antisymmetry via a physically inspired ansatz.

The traditional Helium Slater ansatz:

$$\Psi(r_1, r_2) = e^{-\alpha_1 r_1} e^{-\alpha_2 r_2} - e^{-\alpha_1 r_2} e^{-\alpha_2 r_1}. \tag{5.8}$$

This ansatz uses *Hydrogenic* type wavefunctions, $\phi(r) = e^{-\alpha r}$, as a basis for the construction of the Slater determinant. Indeed, if the two electrons orbiting the nucleus of the Helium atom did not interact with one another, each electron would simply behave as if it were part of a hydrogen atom.

In Figure 5.5 we see the convergence of our estimated local energy to the true ground state energy. We can go further and include some cusp terms that further restrict the electrons from coming near each other or near the nucleus. Some examples of cusp terms are:

- 1. $c_1(r_i) = e^{-\frac{\beta}{r}}$
- 2. $c_2(r_{ij}) = e^{-\frac{\beta}{r_{ij}}}$
- 3. $c_3 = e^{-\frac{r_{ij}}{2(1+\beta r_{ij})}}$

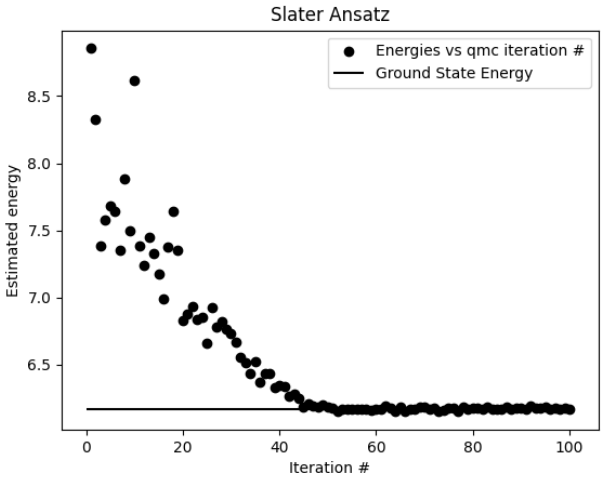

Figure 5.3: Two particles in a box. Gradient descent for the Slater Ansatz. Using the ansatz below for the ground state we estimate the ground state energy. $\Psi(x_1, r_2) = (1-(x_1^2)^{\alpha_1})(1-(x_2^2)^{\alpha_2})x_2 - (1-(x_2^2)^{\alpha_1})(1-(x_1^2)^{\alpha_2})x_1$.

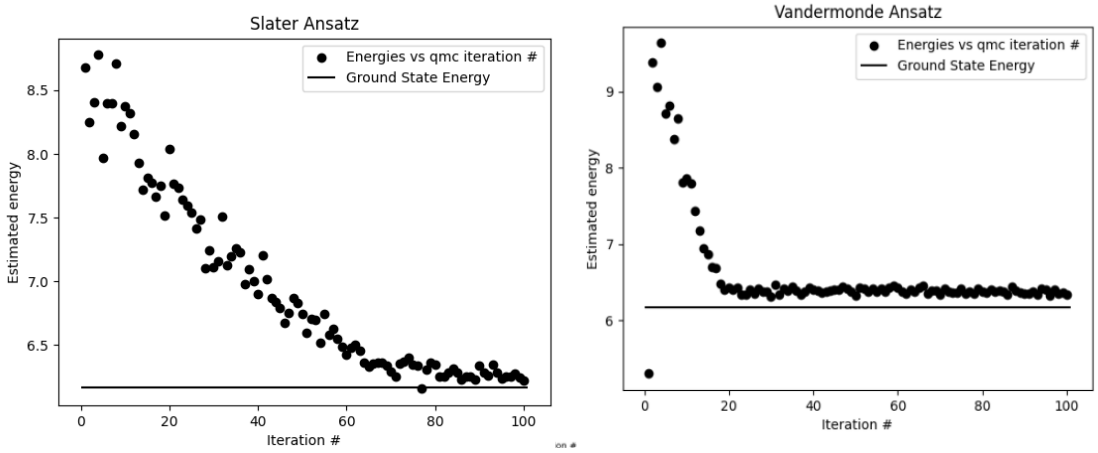

Figure 5.4: Convergence plot for the two-fermions-in-a-box system near the ground state energy

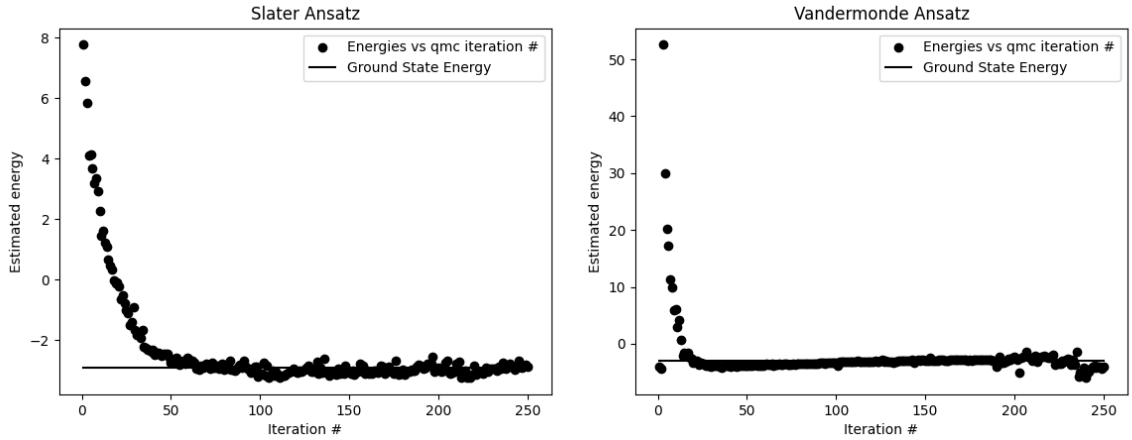

Figure 5.5: Convergence curve for the Helium atom problem with the traditional Slater ansatz with and without a cusp term. The traditional Slater ansatz, Equation (5.8) (left) and for the same ansatz with a cusp term $e^{-\frac{\beta}{r_{12}}}$ (right).

Here, r_i and r_{ij} refer to the electron-nuclear and electron-electron distances respectively. We implement a cusp term of the second kind and the result are shown on the right of Figure 5.5: the convergence of gradient descent is improved significantly. Less than half of the iterations are required to attain convergence.

We now conduct the same analysis for the Vandermonde type ansatz for Helium., which is as follows:

$$\Psi(r_1, r_2) = e^{-\alpha_1 r_1} e^{-\alpha_1 r_2} (e^{-\alpha_2 r_1} - e^{-\alpha_2 r_2}). \tag{5.9}$$

The component $e^{-\alpha_1 r_1} e^{-\alpha_1 r_2}$ is symmetric with respect to r_1 and r_2 while the term $(e^{-\alpha_2 r_1} - e^{-\alpha_2 r_2})$ yields the antisymmetry desired.

We can also add cusp terms, so long as they are symmetric functions, by multiplying the above Ψ by a term incorporating the electron-electron distance or the electron-nucleus distances as already shown above. For our cusp Vandermonde implementation we use the cusp term $e^{-\frac{\alpha_3}{r_{12}}}$.

The results can be seen in Figure 5.6. Once again there is a clear advantage in using the ansatz with the cusp term: despite the noise there is faster convergence. Approximately half of the iterations are needed in this case as opposed to the fifty iterations necessary for the case of the Vandermonde ansatz without the use of the cusp term. In the latter one even sees an increasing deviation from the true ground state energy as the iterations become larger. When comparing the Vandermonde ansatz to the Slater ansatz for Helium it is clear that the Slater ansatz are superior. The noisy fluctuations around the true ground state energy for the Vandermonde cases can be mitigated by taking an average and utilizing statistical methods. Computing the Slater determinant in the Helium case does not pose a significant computational hurdle but as N becomes large the determinant will be very computationally costly and a Vandermonde method with cusp terms leveraged with some statistical inference could provide a cheaper method to calculate ground state energies.

As a final test of the efficacy of the Vandermonde ansatz we perform gradient decent on the parameterized approximate energies afforded by both a Vandermonde and Slater determinant ansatz. Lithium Vandermonde is as follows

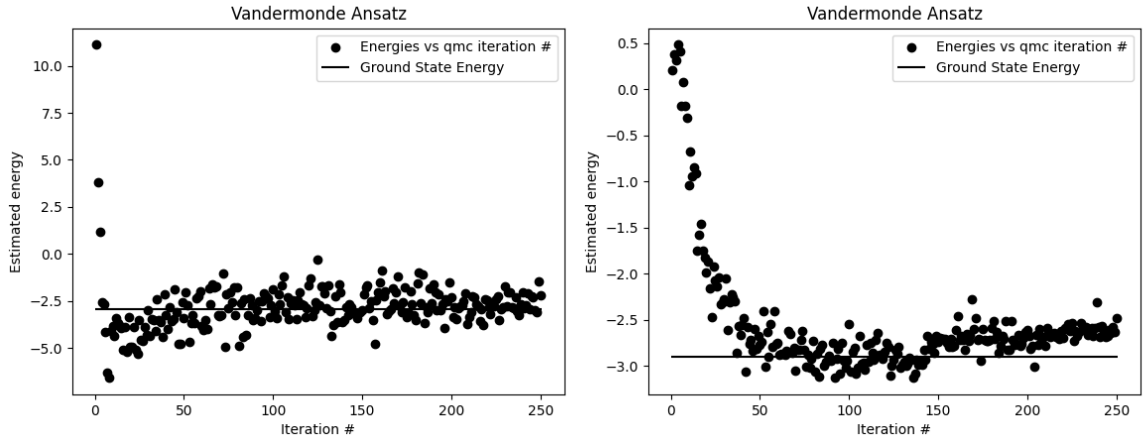

Figure 5.6: Convergence curve for the Helium atom problem with the Vandermonde ansatz with (left) and without (right) a cusp term.

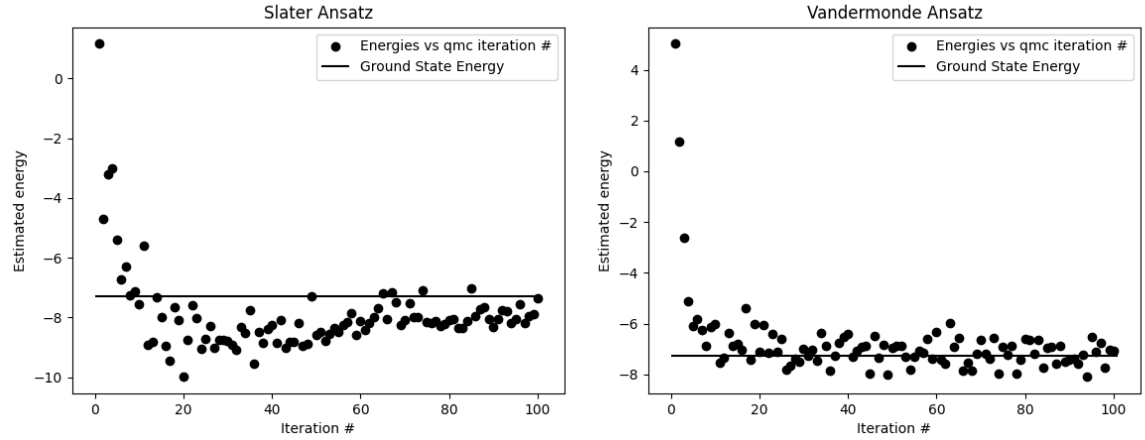

Figure 5.7: Convergence curves for Slater (left) and Vandermonde (right) ansatz for a Lithium (3-electron) system.

$$\Psi(r_1, r_2, r_3) = e^{-\alpha_1 r_1} e^{-\alpha_1 r_2} e^{-\alpha_1 r_3} \sum_{l=1}^{2} \left(\prod_{i<j} e^{-\alpha_{l+1} r_i} - e^{-\alpha_{l+1} r_j} \right). \tag{5.10}$$

The Slater determinant ansatz in this case is

$$\Psi(r_1, r_2, r_3) = \begin{vmatrix} e^{-\alpha_1 r_1} & e^{-\alpha_1 r_2} & e^{-\alpha_1 r_3} \\ e^{-\alpha_2 r_1} & e^{-\alpha_2 r_2} & e^{-\alpha_2 r_3} \\ e^{-\alpha_3 r_1} & e^{-\alpha_3 r_2} & e^{-\alpha_3 r_3} \end{vmatrix}. \tag{5.11}$$

The results can be seen in Figure 5.7. This result shows that we achieve faster convergence for the Vandermonde ansatz. We note that with more iterations we may see the Slater determinant ansatz further converge to the true ground state but this was not tested in our experiments. It also remains to be tested whether the variability around the true ground state energy for the Vandermonde ansatz is less than that of the Slater ansatz.

In the previous section we analyzed a variety of ansatzes for three different models that obeyed the Pauli exclusion principle. One major challenge was tuning the hyperparameters

of the VMC pipeline, i.e. learning rate, proposal distribution standard deviation, momentum, wavefucntion input restrictions, number of walkers and number of steps for the Monte Carlo. Exploitation of known physical properties and symmetries or in some instances trial error was needed in order to find the correct initialization necessary to attain a convergent gradient decent.

Chapter 6

Modifications to PauliNet

6.1 Vandermonde determinants

The main motivation for replacing the Slater determinants with Vandermonde determinants in PauliNet is in reducing the computational cost of evaluating the local energy. In PauliNet, the cost of evaluating the wavefunction scales as $O(N^3)$ where N is the number of electrons. This is because the cost of computing the Slater determinants is N^3 and the determinant calculations dominate the cost of evaluating the wavefunction. Due to the kinetic energy, evaluating the local energy scales as N times the complexity of evaluating the wavefunction. Therefore, the computational cost of PauliNet scales asymptotically as $O(N^4)$.

In contrast, evaluating the Vandermonde determinants scales as $O(N^2)$. Therefore, replacing the Slater determinants in PauliNet with Vandermonde determinants reduces the computational cost from $O(N^4)$ to $O(N^3)$.

In the next sections we outline our implementation of the Vandermonde determinant in PauliNet. We compare performance on various atomic systems including Beryllium, Boron, Lithium hydride and H_2. Finally we provide a visual comparison of the wavefunctions obtained by using PauliNet with the Vandermonde determinant and PauliNet with the Slater determinant.

6.1.1 General approach

There are two Slater determinants in Equation (4.1). Instead of taking a Slater determinant, our modification to PauliNet first takes a product of the φ_i and then takes the Vandermonde determinant of the product. That is, letting $\tilde{\varphi} = \prod_{i=1}^{N} \tilde{\varphi}_i(r)$, we set

$$\det_V := \prod_{i<j} (\tilde{\varphi}(r_i) - \tilde{\varphi}(r_j)),$$

and replace every Slater determinant in PauliNet with the corresponding Vandermonde determinant $\det_V$.

6.1.2 Enforcement of cusp and other conditions

As mentioned in Section 4.4.2, in order for antisymmetry to be enforced, the $\tilde{\varphi}$ must be permutation equivariant. The same is true for the Vandermonde determinant: this motivates our use of the products $\prod_{i=1}^{N} \tilde{\varphi}_i(r)$, which incorporate information from the whole basis

while ensuring permutation equivariance – exchanging two electrons will exchange only the positions of their respective $\tilde{\varphi}$ without changing values.

While this construction can ensure that antisymmetry is enforced, it has a drawback compared to the Slater determinant. If any electron violates the boundary conditions, its $\tilde{\varphi}$ goes to zero: then every term in the Slater determinant will go to zero, while the same is not true of the Vandermonde determinant. After observing and debugging unexpected samples of configurations with infinite or zero energy, we realized that it was necessary to have a separate term outside the Vandermonde determinants enforcing these properties.

Since the underlying Hartree-Fock φ functions enforce these conditions, we simply select a single φ_μ and multiply the entire wavefunction by $\prod_{i=1}^{N} |\varphi_\mu(r_i)|$ – thus ensuring that if any electron violates cusp or boundary conditions, the entire wavefunction still goes to zero. This term, which falls outside of the determinants, is permutation invariant as required.

6.1.3 Description of experiments

Experimental parameters

We used the same architectural designs as the original PauliNet paper (see Table 2 in [10] for the full set of parameters), with a few exceptions. We only used a single determinant for both the Slater and PauliNet cases. All network sizes and dimensionalities of feature representations remained the same, and we used the same basis functions. We found it necessary to lower the maximum learning rate to 0.005 from the default 0.01. We ran for 600 or 800 iterations, less than for the results collected originally in [10].

Results and Comparisons

With the Vandermonde determinants, convergence was achieved but to an energy relatively far from the correct answer – the Slater determinants reached much closer. To visualize the wavefunction during training, we made plots of the wavefunction sliced along a single coordinate of a single electron.

Figure 6.1 shows the evolution of the wavefunction for LiH, a system with four electrons (one for hydrogen and three for lithium). Figure 6.2 shows the same evolution Beryllium, a single-atom system with four electrons.

In both of these systems, we find that the Slater determinant reaches a lower energy – much closer to the true values in both cases. We also make plots of the difference in wavefunction outputs every 50 training iterations: this is a measure of how much the wavefunction shifts during training. The results are shown in Figure 6.3. We find that PauliNet with the Slater determinant continues to train, while the Vandermonde determinant trains quickly at first but appears to get "stuck" in a local minimum far from the true ground state energy.

The relative accuracy of various methods is most readily understood by comparing the estimation of the ground-state energy. In particular, we compared our estimation of the ground state energy to the estimation obtained by the Hartree-Fock method and the original PauliNet. PauliNet uses the Hartree-Fock method as a baseline solution and as a starting point for the variational Monte Carlo method. It first performs a Hartree-Fock calculation to produce one electron orbitals $\varphi_\mu(r_i)$ (see Equation (4.1)) which are then used as inputs to PauliNet. After performing variational Monte Carlo, PauliNet significantly surpasses the accuracy of Hartree-Fock. The authors of PauliNet demonstrate this in [10] by comparing the accuracy of their solution after 7000 training steps to the Hartree-Fock method and another deep learning based VMC method.

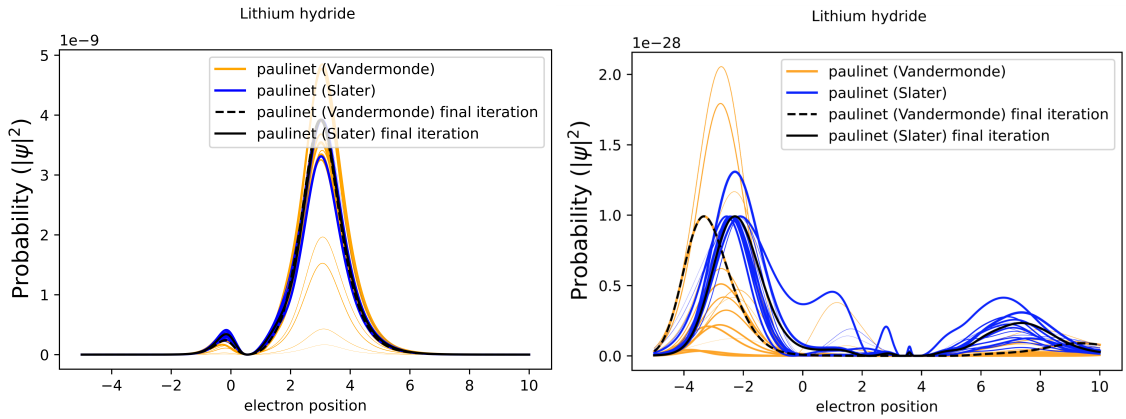

Figure 6.1: Two slices of the PauliNet wavefunction for Lithium hydride. Two different slices of the wave function are plotted every 50 steps over the course of 800 total training steps. Thinner lines are earlier time steps, thicker are later. The ground state energy estimation at the final iteration is -7.1686 for PauliNet with the Vandermonde determinant, and -8.0619 for PauliNet with the Slater determinant. The true ground state energy is -8.07.

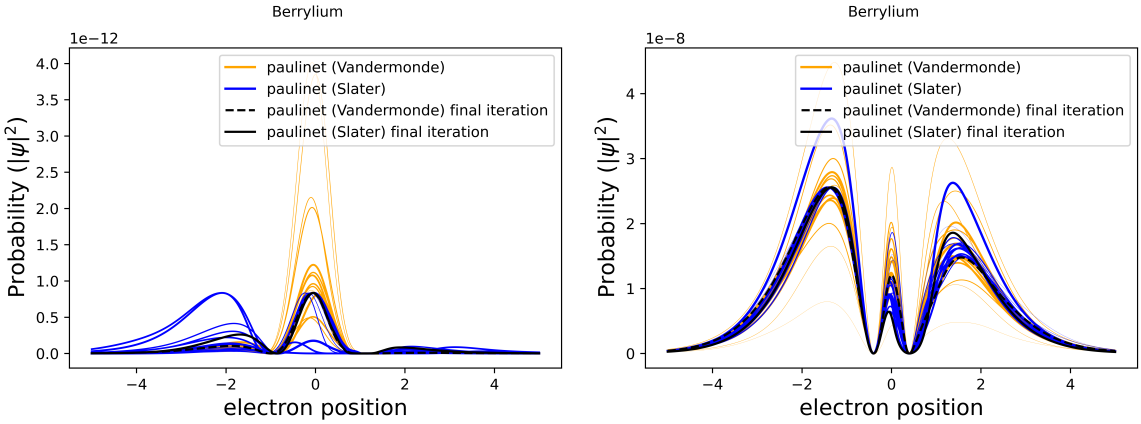

Figure 6.2: Two slices of the PauliNet wavefunction for Berrylium. Two different slices of the wave function are plotted every 50 steps over the course of 800 total training steps. Thinner lines are earlier time steps, thicker are later. The ground state energy estimation at the final iteration is -13.2703 for PauliNet with the Vandermonde determinant, and -14.6479 for PauliNet with the Slater determinant. The true ground state energy is -14.6674.

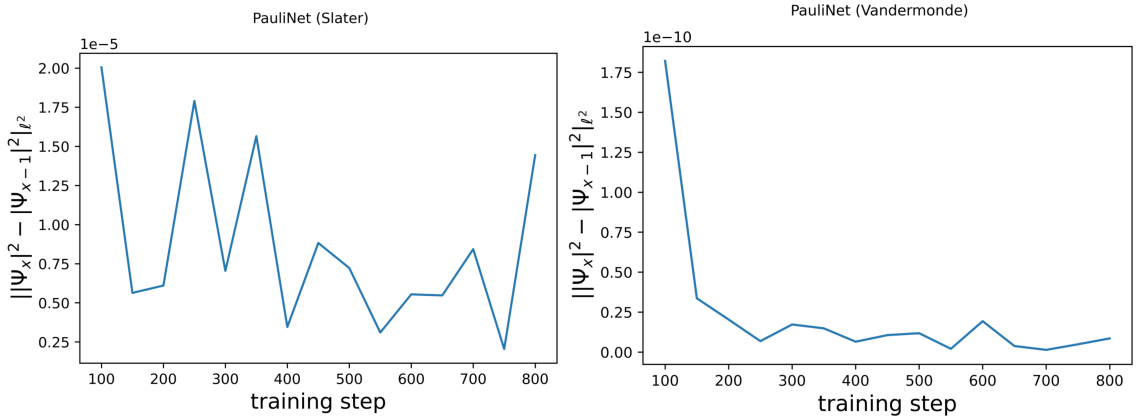

Figure 6.3: Norm of difference in wavefunction outputs evaluated at 3000 random points, every 50 training iterations.

Unfortunately, even after 7000 training steps, our modification with the Vandermonde determinant in general does not exceed Hartree-Fock accuracy. Others who used similar constructions [8] also reported poor results. In order to present our method as a viable low-cost alternative to PauliNet, we would at least want to surpass the accuracy of Hartree-Fock for most systems. We believe that there may be other modifications that one could make in order to achieve this level of accuracy while still maintaining the N^3 asymptotic complexity. This possibility is discussed in Section 7.4.

Chapter 7

Challenges

7.1 Introduction

In this section, we discuss challenges involved in modifying the PauliNet ansatz: some which we were able to overcome, and some which we leave to future work. Particular challenges included preserving the enforcement of boundary conditions, discussed in the previous section. We also discuss experimental attempts to modify the architectures of the Jastrow and backflow networks, and conjecture about what might be needed to achieve commensurate performance using the Vandermonde determinant.

7.2 Enforcement of boundary conditions

As mentioned in Section 6.1.2, the Slater determinant nicely preserves certain boundary conditions in the basis wavefunctions, while the Vandermonde determinant does not. As such, to enforce boundary conditions, we had to additionally premultiply by some of the basis functions. It is possible that some other approach to enforcing the boundary conditions might have resulted in better performance.

7.3 Negative results for convolutional neural networks

In addition to working with the Vandermonde determinant, we attempted to make additional modifications to the PauliNet architecture. The architecture includes feedforward neural networks in several places, particularly the Jastrow and backflow terms. Since memory is always an important limiting factor in deep learning, we considered replacing some of the feedforward layers with convolutional layers which have many fewer parameters and are widely used across deep learning.

Our results showed a worsening of performance after these modifications. Training was about 10 times slower despite the convolutional networks having roughly 1/1000 as many parameters. This made training and experimentation unreasonably slow. We suspected that the issue might be the cost of computing the Laplacian involved in the local energy (Equation (4.3)) – computing second derivatives is an unusual operation for deep learning.

Input size	linear	5	9	33	65
64	0.01	0.05	0.05	0.05	0.05
128	0.03	0.14	0.12	0.12	0.13
256	0.07	0.29	0.29	0.31	0.35

Table 7.1: Average runtime in seconds for computing Laplacian on 50 batches of 5 inputs for convolutional networks of different kernel sizes. Linear denotes linear network.

Input size	linear	5	9	33	65
64	7.660e-05	3.110e-04	2.590e-04	2.730e-04	2.770e-04
128	6.500e-05	3.010e-04	3.260e-04	3.200e-04	3.190e-04
256	7.560e-05	3.730e-04	3.600e-04	3.780e-04	3.850e-04

Table 7.2: Average runtime in seconds for computing forward pass on 50 batches of 5 inputs for convolutional networks of different kernel sizes. Linear denotes linear network

7.3.1 Experiments: speed of computing Laplacians for convolutional networks

To investigate this issue further, we kicked off a simple experiment on some Google Colab resources (CPU only). We compared a simple linear architecture, in which each linear layer halves the dimensionality of the previous until a 1-dimensional output is reached, to a convolutional architecture with max pooling where each pair of convolutional/pooling layers have the same output sizes. We vary the kernel size of the convolutional architecture. We sampled 250 random points of each size (split into batches of 5) and tested the runtime to perform a forward pass and to compute the Laplacian (using the same method as used for PauliNet). Results are in Tables 7.1 and 7.2.

Regardless of the size of the convolution kernel, computing both the forward pass and the Laplacian were significantly slower for the convolutional network. However, the computation of the Laplacian significantly magnified the performance difference.

This suggests that when considering architectural changes to PauliNet, a key consideration must be the speed of computing second derivatives with the new architecture – the Laplacian must be computed at every single sampled point in order to determine the local energy.

7.4 Improved basis functions for compatibility with the Vandermonde determinant

A possible limitation of our adaptations to PauliNet might be the use of the Hartree-Fock-derived basis functions as predetermined inputs to the ansatz. The core Hartree-Fock ansatz itself uses a Slater determinant, so the parameters of the basis functions might be particularly well-tuned for use in conjunction with Slater determinants. The flexible neural networks of the rest of the PauliNet architecture are still capable of learning, but when using the Vandermonde determinant they might have to overcome this limitation.

As such, one possible avenue for improvement might be replacing the basis functions φ with ones somehow better suited to the Vandermonde determinant. The authors of PauliNet plausibly claim that part of the success of their method is the fact that it begins with the

physically plausible Hartree-Fock functions. It might be possible to initialize by solving some analogue to the Hartree-Fock wavefunction that uses a Vandermonde determinant instead of a Slater determinant, potentially yielding better performance.

Chapter 8

Conclusion

Variational Monte Carlo is a widely-used algorithm for approximating the ground state energy of quantum systems. While more complex variations of Monte Carlo approaches can give better performance, VMC is widely used and implemented in many popular high-performance codes including QMCPACK. Variational Monte Carlo relies on the use of a flexible parametric ansatz to represent the wavefunction.

Recently, several authors including [19, 10] have proposed to use deep-learning-based ansatzes in the variational Monte Carlo pipeline. These techniques show promise – using the relatively simple VMC algorithm, they are able to achieve performance commensurate with more complex algorithms. In this work, we focused on the PauliNet architecture from [10].

For many-electron systems, most common ansatzes (including both PauliNet and QMCPACK) use a construction known as the Slater determinant to enforce antisymmetry. For an N-electron system, this requires taking the determinant of an $N \times N$ matrix, an $O(N^3)$ operation which in some cases is the asymptotically dominant term for computational cost. There is however another possible construction known as the Vandermonde determinant. The authors of [7] show that the Vandermonde determinant can function as a universal approximator for antisymmetric functions, suggesting that in principle, it could work as well as the Slater determinant as part of a flexible deep-learning-based ansatz. Yet it is potentially only an $O(N^2)$ operation.

To investigate the feasibility of replacing the Slater determinant with the Vandermonde determinant, we have followed two approaches:

- We implemented our own variational Monte Carlo pipeline on which we tested simple Slater and Vandermonde based ansatzes on simple physical systems.
- We modified PauliNet to use the Vandermonde determinant in place of its Slater determinants.

In general, we found that the Vandermonde determinant does not work as well as the Slater determinant. In our own implementation, its performance is tolerable, suggesting that as systems scale up and the asymptotic difference in evaluation cost becomes relevant, the tradeoff in computational cost could be reasonable. Within PauliNet, the use of the Vandermonde determinant results in a wavefunction that, while somewhat viable, performs significantly worse than PauliNet with Slater determinants.

There are significant avenues for improvement that could be investigated in the future. There are also several cusp terms that could be experimented with as well as strategies for

selecting the hyperparameters in our implementation. As mentioned before, our pipeline requires fine tuning of its hyperparameters in order to attain convergence. As the dimensionality of a quantum system grows, finding a good set of initialization parameters becomes a difficult task.

With regards to PauliNet, a major challenge during development was ensuring the appropriate enforcement of physical boundary conditions – this falls out naturally when using the Slater determinant but required extra effort to guarantee using the Vandermonde determinant. It is possible that different techniques for enforcing these conditions might give better performance than the ones we used. Additionally, it is possible that the input Hartree-Fock basis functions are not well-adapted for the Vandermonde determinant, and that deriving new basis functions could also significantly enhance performance.

Finally, in this work we only focused on the time independent Scrödinger equation. Furthermore, we only considered the ground states of each of the systems when we tested our VMC implementation. Future work could potentially explore to what extent our VMC pipeline and PauliNet would be able to approximate higher order excited states, and how this work could be adapted for solving the time dependent Schrödinger equation with VMC methods.

Selected Bibliography Including Cited Works

[1] M. Betancourt, *A conceptual introduction to Hamiltonian Monte Carlo*, arXiv preprint arXiv:1701.02434, (2017).

[2] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[3] M. Caffarel and R. Assaraf, *A pedagogical introduction to quantum Monte-Carlo*, in Mathematical models and methods for ab initio quantum chemistry, Springer, 2000, pp. 45–73.

[4] A. Cross, *The IBM Q experience and QISKit open-source quantum computing software*, APS, 2018 (2018), pp. L58–003.

[5] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm*, arXiv preprint arXiv:1411.4028, (2014).

[6] W. Foulkes, L. Mitas, R. Needs, and G. Rajagopal, *Quantum Monte Carlo simulations of solids*, Reviews of Modern Physics, 73 (2001), p. 33.

[7] J. Han, Y. Li, L. Lin, J. Lu, J. Zhang, and L. Zhang, *Universal approximation of symmetric and anti-symmetric functions*, CoRR, abs/1912.01765 (2019).

[8] J. Han, L. Zhang, and E. Weinan, *Solving many-electron Schrödinger equation using deep neural networks*, Journal of Computational Physics, 399 (2019), p. 108929.

[9] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[10] J. Hermann, Z. Schätzle, and F. Noé, *Deep neural network solution of the electronic Schrödinger equation*, CoRR, abs/1909.08423 (2019).

[11] M. Hjorth-Jensen, *Computational physics*, Lecture notes, (2011).

[12] J. R. Johansson, P. D. Nation, and F. Nori, *QuTiP: An open-source python framework for the dynamics of open quantum systems*, Computer Physics Communications, 183 (2012), pp. 1760–1772.

[13] J. Kim, A. D. Baczewski, T. D. Beaudet, A. Benali, M. C. Bennett, M. A. Berrill, N. S. Blunt, E. J. L. Borda, M. Casula, D. M. Ceperley, et al., *QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids*, Journal of Physics: Condensed Matter, 30 (2018), p. 195901.

[14] J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, *Generalized unitary coupled cluster wave functions for quantum computation*, Journal of chemical theory and computation, 15 (2018), pp. 311–324.

[15] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu, *Self-learning Monte Carlo method and cumulative update in fermion systems*, Physical Review B, 95 (2017), p. 241104.

[16] R. M. Neal, *MCMC using Hamiltonian dynamics*, in Handbook of Markov chain Monte Carlo, S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds., Chapman & Hall/CRC Handb. Mod. Stat. Methods, CRC Press, Boca Raton, FL, 2011, pp. 113–162.

[17] K. Nordlund, *Lecture notes on quantum Monte Carlo*, 2006.

[18] F. Pagani, M. Wiegand, and S. Nadarajah, *An n-dimensional rosenbrock distribution for mcmc testing*, arXiv preprint arXiv:1903.09556, (2019).

[19] D. Pfau, J. S. Spencer, A. G. d. G. Matthews, and W. M. C. Foulkes, *Ab-initio solution of the many-electron Schrödinger equation with deep neural networks*, arXiv preprint arXiv:1909.02487, (2019).

[20] G. O. Roberts, A. Gelman, W. R. Gilks, et al., *Weak convergence and optimal scaling of random walk metropolis algorithms*, The annals of applied probability, 7 (1997), pp. 110–120.

[21] K. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, *Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions*, Nature communications, 10 (2019), pp. 1–10.

[22] H. Shen, J. Liu, and L. Fu, *Self-learning Monte Carlo with deep neural networks*, Physical Review B, 97 (2018), p. 205140.

[23] B. Tracey, D. Wolpert, and J. J. Alonso, *Using supervised learning to improve Monte Carlo integral estimation*, AIAA journal, 51 (2013), pp. 2015–2023.

[24] C. Umrigar, M. Nightingale, and K. Runge, *A diffusion Monte Carlo algorithm with very small time-step errors*, The Journal of chemical physics, 99 (1993), pp. 2865–2890.

[25] O. T. Unke and M. Meuwly, *PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges*, Journal of chemical theory and computation, 15 (2019), pp. 3678–3693.