Machine Learning as an Alternative Wavefunction Ansatz to Improve Variational Monte Carlo

Student Members

Alberto Acevedo

Michael Curry

Brett Leroux

Academic Mentor

Dr. Shantanu Joshi

Industry Mentor

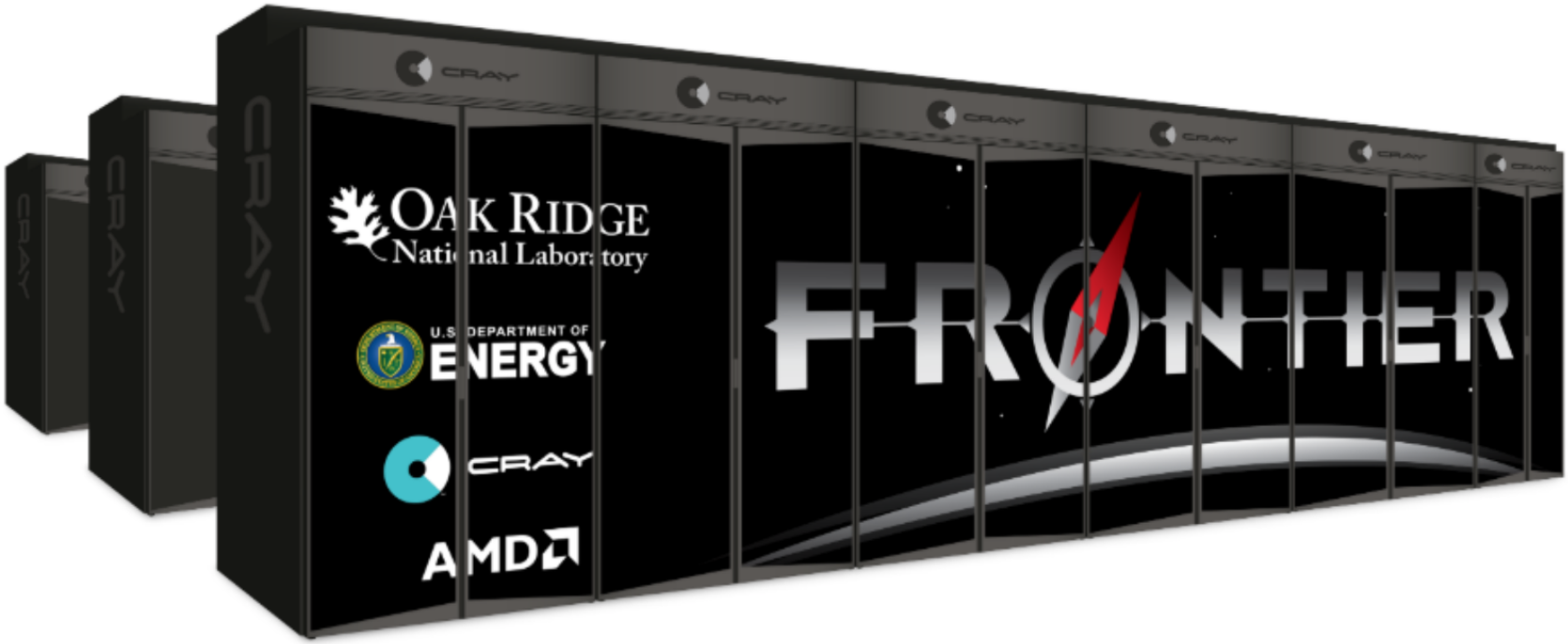
Dr. Nicholas Malaya

Intro

1. AMD and Quantum Monte Carlo
2. Variational Quantum Monte Carlo: Introduction
3. N electron systems
   - The need for antisymmetry
   - Comparison of Slater and Vandermonde determinants
4. Deep-learning-based ansatz: PauliNet
   - Description of PauliNet
   - Proposed improvement: Vandermonde with PauliNet
5. Conclusions and new directions

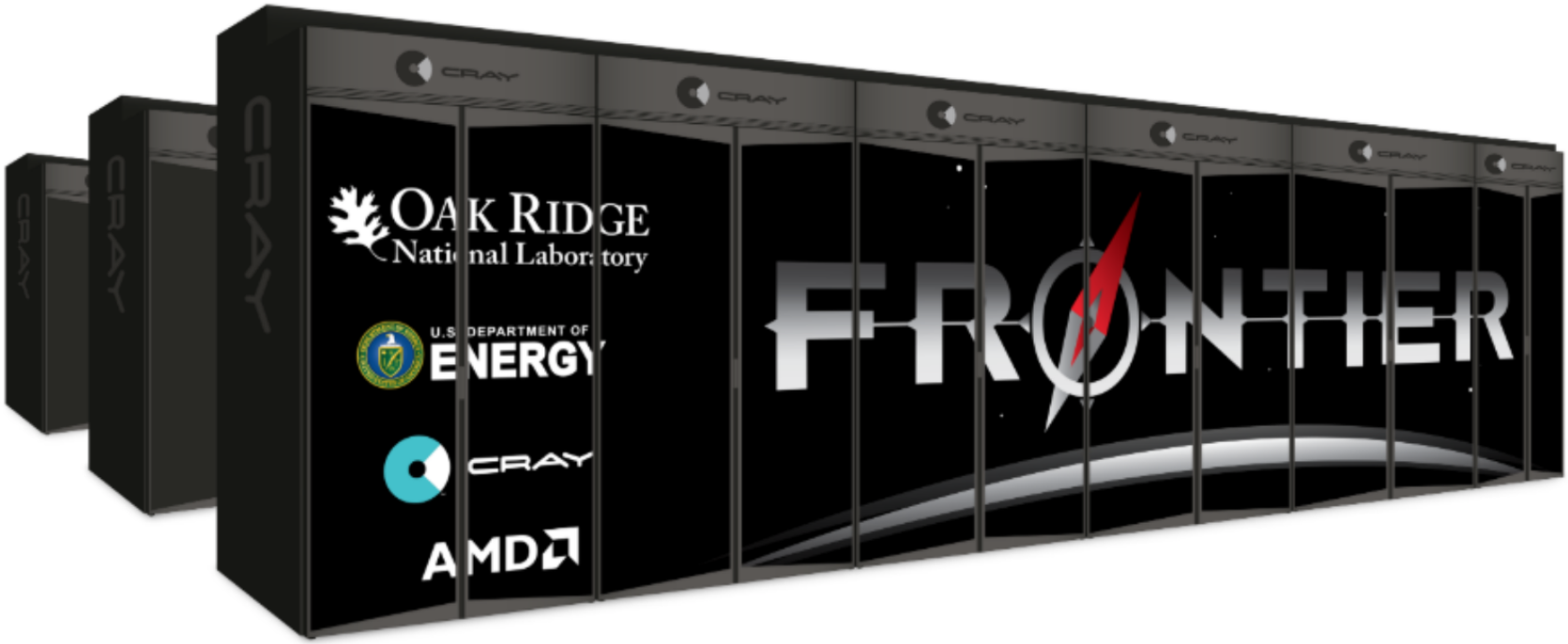
Outline for section 1

AMD and Frontier (exascale computing)



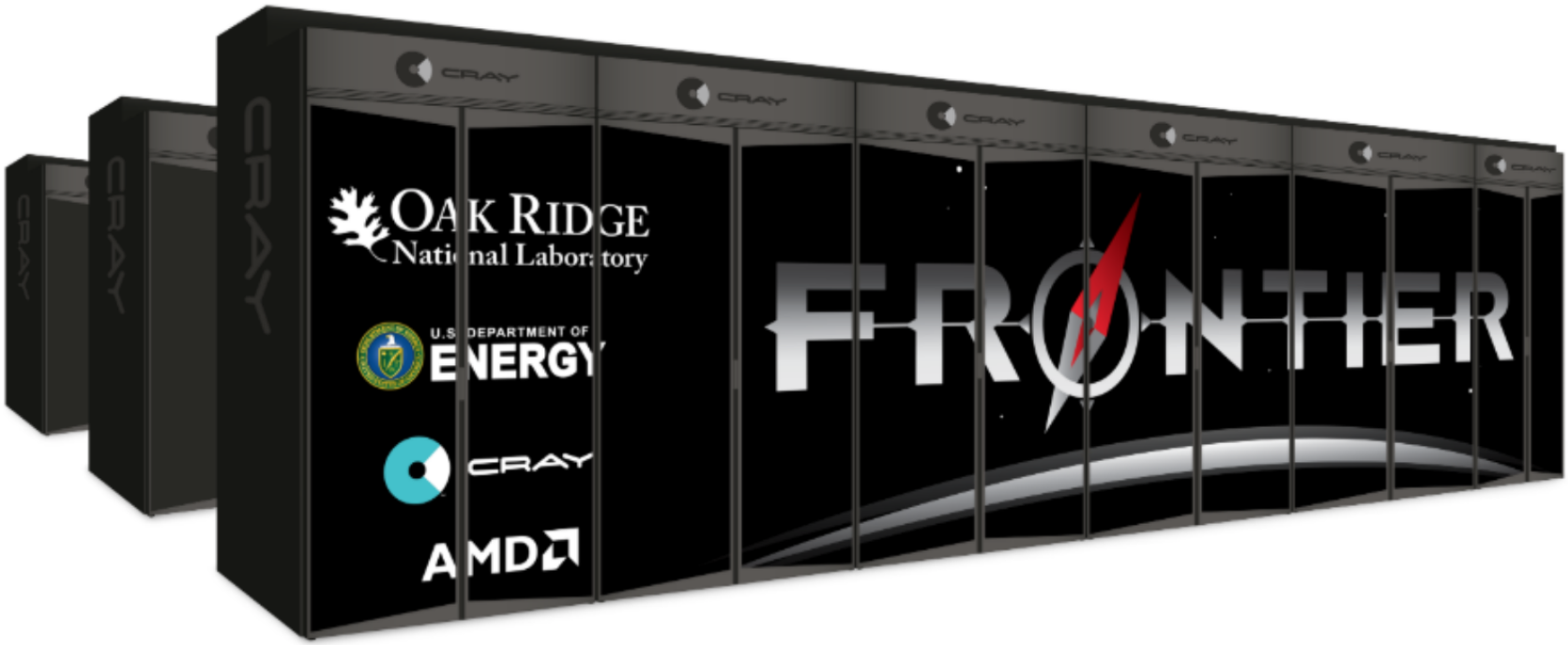
- Epyc CPUs
- Radeon GPUs

AMD and Frontier (exascale computing)

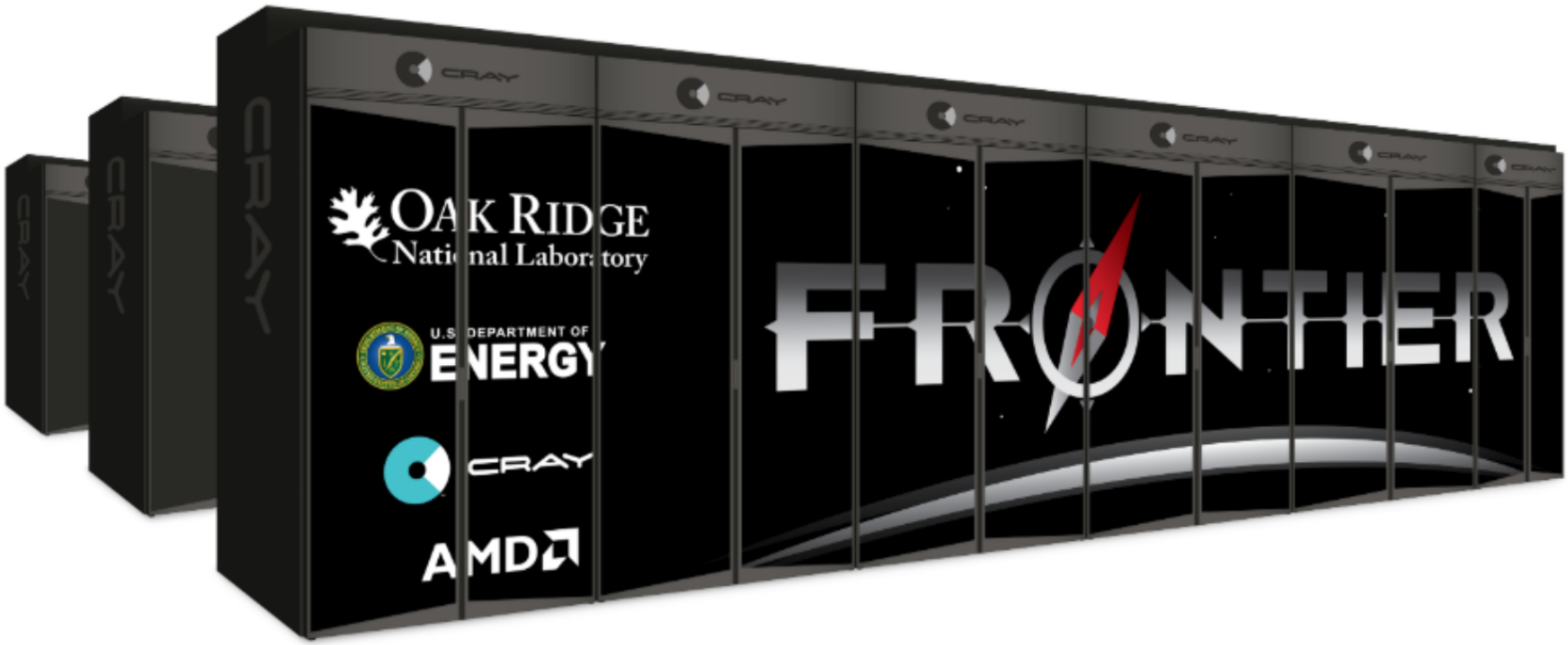


- Epyc CPUs
- Radeon GPUs
- In 2021: Frontier will become the largest supercomputer ever constructed.

QMCPACK


CRAY
OAK RIDGE
National Laboratory
U.S. DEPARTMENT OF
ENERGY
CRAY
AMD
FRONTIER

QMCPACK



- Uses **Quantum Monte Carlo** methods to solve the Schrödinger equation $\hat{H}\Psi = E\Psi$.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.
- For a molecule with N electrons, the wavefunction takes the form $\Psi(r_1, r_2, \ldots, r_N)$.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.
- For a molecule with N electrons, the wavefunction takes the form $\Psi(r_1, r_2, \ldots, r_N)$.
  - r_i is the position of electron i in three-dimensional space.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.
- For a molecule with N electrons, the wavefunction takes the form $\Psi(r_1, r_2, \ldots, r_N)$.
  - r_i is the position of electron i in three-dimensional space.
  - Ψ is a function of $3N$ variables.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.
- For a molecule with N electrons, the wavefunction takes the form $\Psi(r_1, r_2, \ldots, r_N)$.
  - r_i is the position of electron i in three-dimensional space.
  - Ψ is a function of $3N$ variables.
  - $|\Psi(r_1, r_2, \ldots, r_N)|^2$ gives the **probability** that the electrons are in a given configuration $r_1, r_2, \ldots, r_N$.

Significance of Quantum Monte Carlo

- **Wavefunctions** describe molecular, atomic, and subatomic systems.
- For a molecule with N electrons, the wavefunction takes the form $\Psi(r_1, r_2, \ldots, r_N)$.
  - r_i is the position of electron i in three-dimensional space.
  - Ψ is a function of $3N$ variables.
  - $|\Psi(r_1, r_2, \ldots, r_N)|^2$ gives the **probability** that the electrons are in a given configuration $r_1, r_2, \ldots, r_N$.
- Knowing the configuration of electrons can help infer chemical properties of molecules and materials.

The Pauli exclusion principle

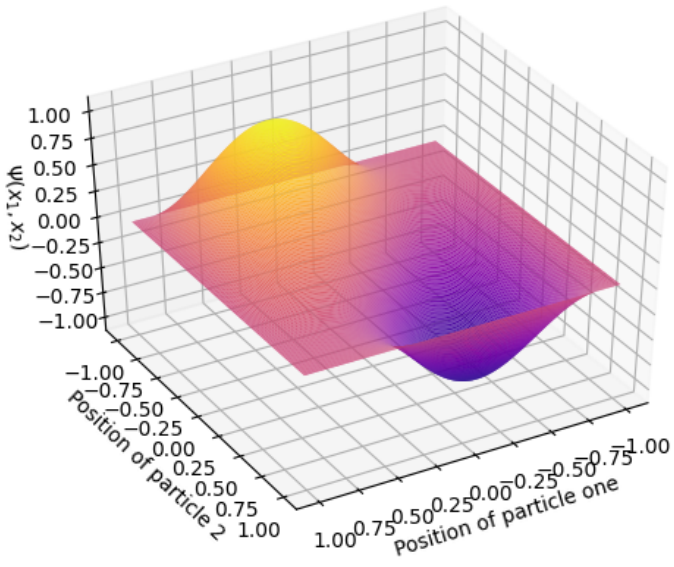
- The Pauli exclusion principle is an empirical law that states that two interacting fermions (electrons are fermions) cannot occupy the same quantum state.

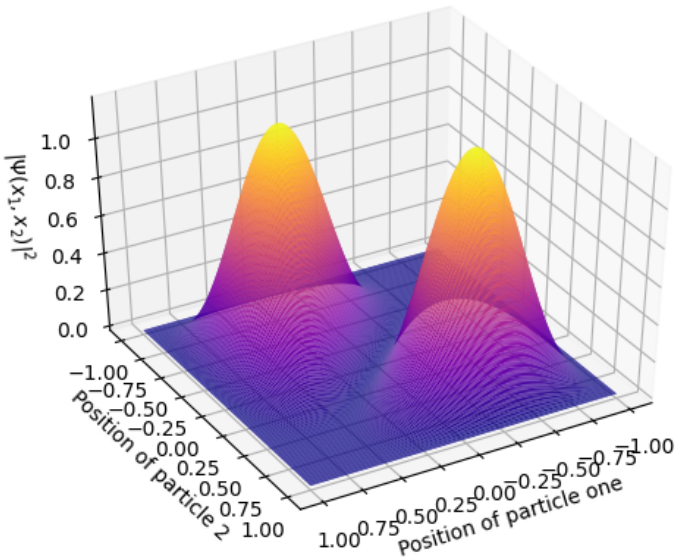
The Pauli exclusion principle

- The Pauli exclusion principle is an empirical law that states that two interacting fermions (electrons are fermions) cannot occupy the same quantum state.
- In mathematical terms, the wavefunction must obey **antisymmetry**. This means that **exchanging two particles must flip only the sign of the wavefunction**:

$$\Psi(\cdots, r_i, \cdots, r_j, \cdots) = -\Psi(\cdots, r_j, \cdots, r_i, \cdots)$$

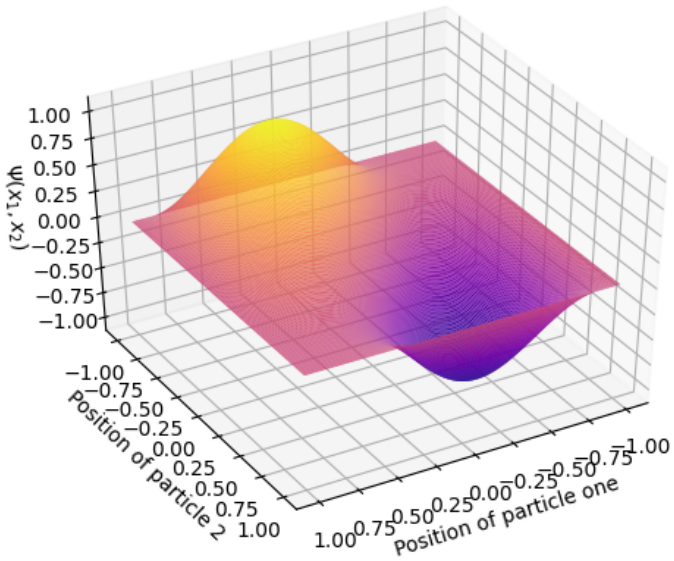
Example: two fermions in a 1-dimensional box

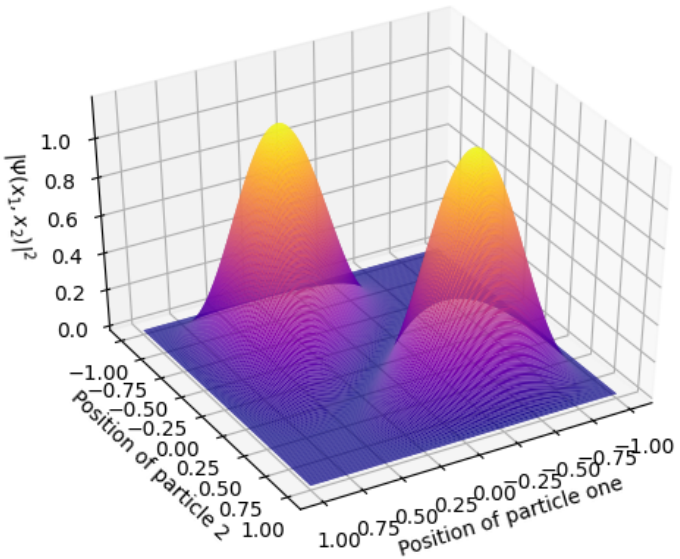




- Wavefunction $\Psi(x_1, x_2)$
- x_1, x_2 are positions of particles.
- $|\Psi(x_1, x_2)|^2$ is probability

Example: two fermions in a 1-dimensional box





- Wavefunction $\Psi(x_1, x_2)$
- x_1, x_2 are positions of particles.
- $|\Psi(x_1, x_2)|^2$ is probability

- Ψ must be **antisymmetric**: $\Psi(x_1, x_2) = -\Psi(x_2, x_1)$.

Approximating Wavefunctions

Approximating Wavefunctions

- Choose a class of wavefunctions $\Psi_\alpha(r)$ parameterized by α.

Approximating Wavefunctions

- Choose a class of wavefunctions $\Psi_\alpha(r)$ parameterized by α.
- Optimize alpha to find the ground state wavefunction.

Challenge of applying QMC

Need a well informed "guess" (ansatz) of the function $\Psi_\alpha(r)$ which best describes the system.

Challenge of applying QMC

Need a well informed "guess" (ansatz) of the function $\Psi_\alpha(r)$ which best describes the system.

- **Solution 1:** Use your knowledge of physics to design the ansatz.

Challenge of applying QMC

Need a well informed "guess" (ansatz) of the function $\Psi_\alpha(r)$ which best describes the system.

- **Solution 1:** Use your knowledge of physics to design the ansatz.
- **Solution 2** Use neural networks as function approximators.

Challenge of applying QMC

Need a well informed "guess" (ansatz) of the function $\Psi_\alpha(r)$ which best describes the system.

- **Solution 1:** Use your knowledge of physics to design the ansatz.
- **Solution 2** Use neural networks as function approximators.

Many methods e.g. PauliNet do both.

Our focus

- We aim to investigate the feasibility of **deep learning** to produce new and powerful forms of **ansatz**.
- 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 is a major part of the computational cost of ansatz that use it.
- We focus on a different technique for enforcing antisymmetry known as the **Vandermonde determinant**.

Slater determinants

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

Slater determinants

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

Computational cost: $O(N^3)$

Vandermonde determinants

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

Vandermonde determinants

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

$$= \prod_{i<j} (\varphi(r_i) - \varphi(r_j))$$

Vandermonde determinants

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

$$= \prod_{i<j} (\varphi(r_i) - \varphi(r_j))$$

Computational cost: $O(N^2)$

Outline for section 2

The ground state energy of the system

- Variational Monte Carlo is used to find the **ground state energy** of the system.

The ground state energy of the system

- Variational Monte Carlo is used to find the **ground state energy** of the system.
- All wavefunctions ϕ other than the true ground state Ψ_0 have higher energy: $\mathbb{E}_{\Psi_0}[E] \leq \mathbb{E}_{\phi}[E]$.

The ground state energy of the system

- Variational Monte Carlo is used to find the **ground state energy** of the system.
- All wavefunctions ϕ other than the true ground state Ψ_0 have higher energy: $\mathbb{E}_{\Psi_0}[E] \leq \mathbb{E}_{\phi}[E]$.
- The system's properties are defined by an operator H known as the Hamiltonian. Given H and the wavefunction Ψ, the expected energy of the system is:

$$\mathbb{E}[E] = \frac{\int \Psi_\alpha^*(\vec{r}) H \Psi_\alpha(\vec{r}) dV}{\int |\Psi_\alpha(\vec{r})|^2 dV}$$

We want to minimize this quantity.

In general, the expected energy cannot be computed analytically. An approximation is

$$\mathbb{E}[E] \approx \frac{1}{N} \sum_{i=1}^{N} E_L(x_i)$$

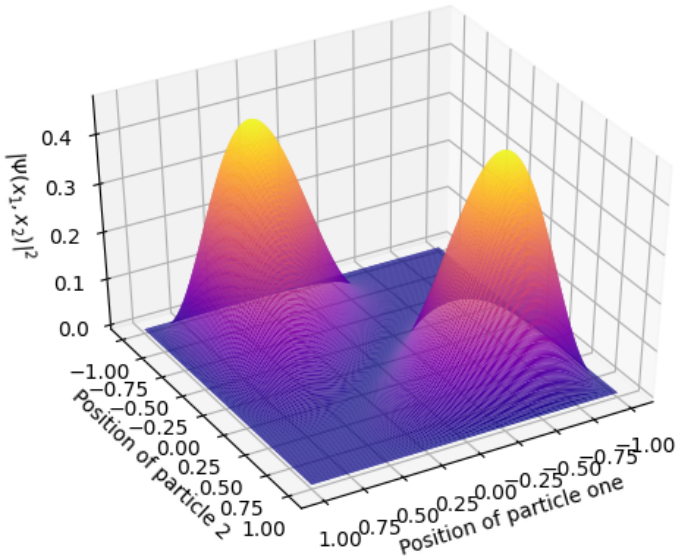
- E_L is the **local energy** associated with a particle in a single, specific configuration.
- $\{x_i\}$ is a set of random samples from $|\Psi_\alpha(\vec{r})|^2$. These are drawn using Markov-chain Monte Carlo (MCMC).

Sampling with Markov-Chain Monte Carlo

Using the two particles in a box ansatz as an example. We set $(\alpha_1, \alpha_2) = (1.95, 0.95)$.

$\Psi_{\alpha_1,\alpha_2}(x_1, x_2) = (1 - x_1^{2\alpha_1})(1 - x_2^{2\alpha_2})x_2 - (1 - x_2^{2\alpha_1})(1 - x_1^{2\alpha_2})x_1$

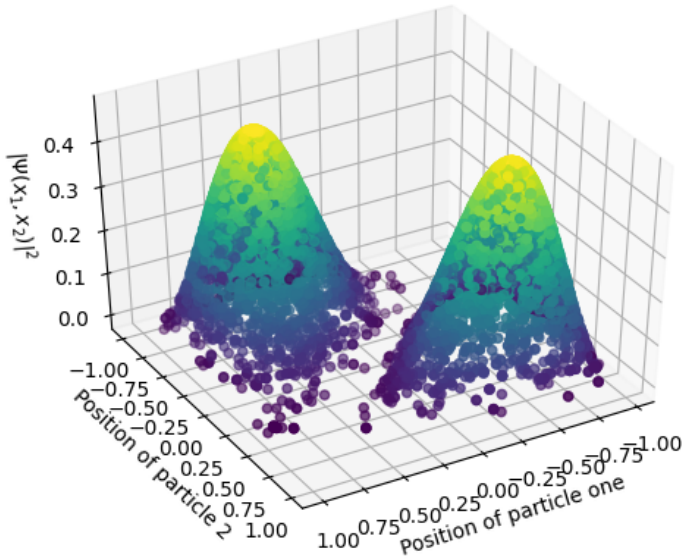
Sampling with MCMC 4500 times.



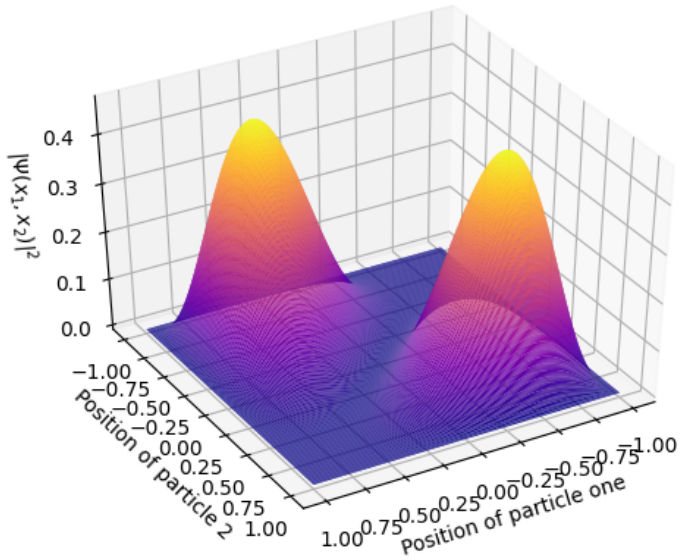
Actual $|\Psi_{\alpha_1,\alpha_2}(x_1, x_2)|^2$.

Sampling with Markov-Chain Monte Carlo

Using the Two particles in a box Ansatz as an example. We set $(\alpha_1, \alpha_2) = (1.95, 0.95)$.
$\Psi_{\alpha_1,\alpha_2}(x_1, x_2) = (1 - x_1^{2\alpha_1})(1 - x_2^{2\alpha_2})x_2 - (1 - x_2^{2\alpha_1})(1 - x_1^{2\alpha_2})x_1$

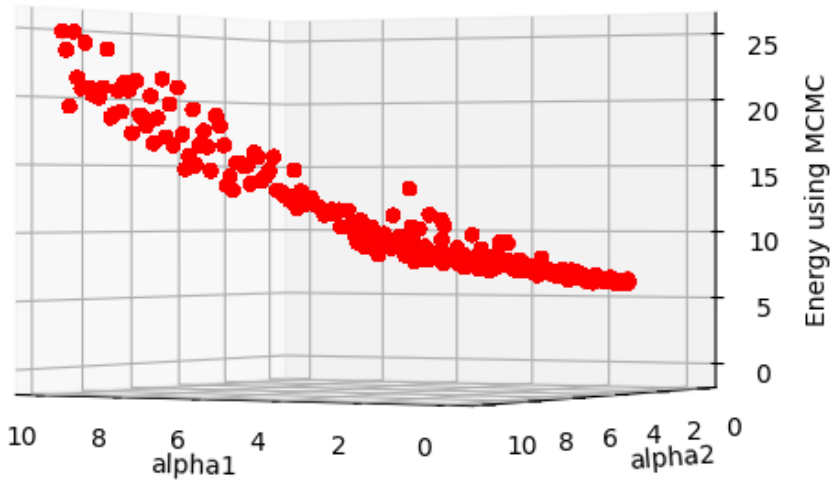


Sampling with MCMC 10000 times.



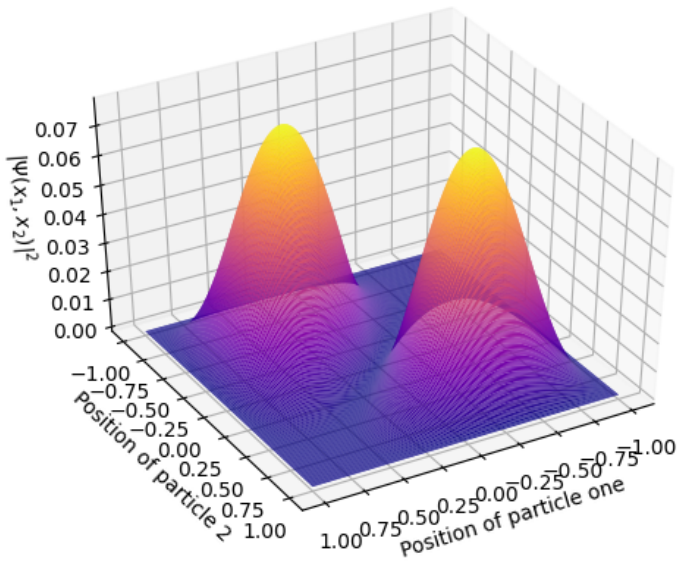
Actual $|\Psi_{\alpha_1,\alpha_2}(x_1, x_2)|^2$.

Gradient descent on the approximated energy

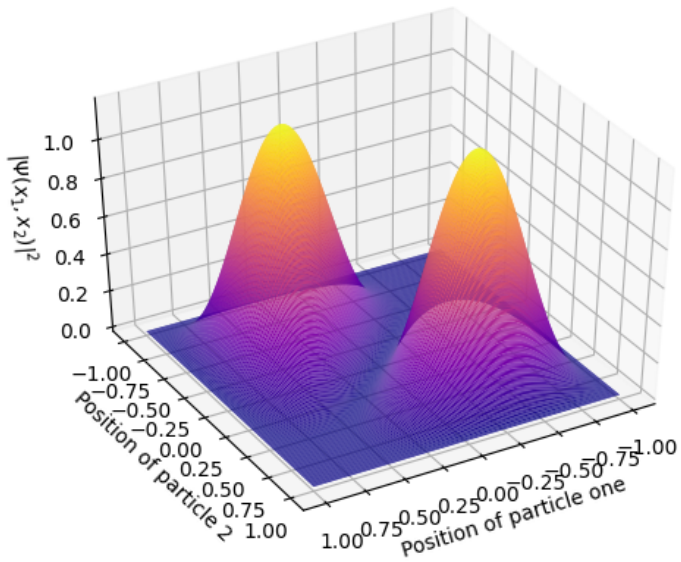


GD to approximate ground state energy of the two fermions in a box model. $E_0 = 6.168$ in our units. Converges to $\alpha_1 = 0.876$ and $\alpha_2 = 0.52$.

Ansatz with optimal alphas vs Exact solution



Using $\alpha_1 = 0.876$ and $\alpha_2 = 0.52$ with our ansatz.



Exact solution.

Outline for section 3

N-electron systems

We focus on the study of N-electron systems: that is, atoms and molecules with electrons surrounding the nucleus. The Hamiltonian (which defines the energy) of systems like is:

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

- Kinetic energy of electrons: defined in terms of Laplacian of the wavefunction.
- Electron-nucleus energy
- Electron-electron energy

Pauli exclusion principle

Recall the: **Pauli exclusion principle:** Two identical fermions cannot occupy the same state.

- $|\Psi(\cdots, r_i, \cdots, r_j, \cdots)|^2 = |\Psi(\cdots, r_j, \cdots, r_i, \cdots)|^2$.
- $\Psi(\cdots, r_i, \cdots, r_j, \cdots) = -\Psi(\cdots, r_j, \cdots, r_i, \cdots)$

A first attempt at an ansatz

Suppose we have single-electron basis functions φ_j. A first attempt at an ansatz might be:

$$\Psi(r_1, \cdots, r_N) = \prod_i \varphi_i(r_i)$$

A first attempt at an ansatz

Suppose we have single-electron basis functions φ_j. A first attempt at an ansatz might be:

$$\Psi(r_1, \cdots, r_N) = \prod_i \varphi_i(r_i)$$

This is known as the Hartree product ansatz, but it does not enforce antisymmetry when r_i, r_j switched.

One can extend the idea of the Hartree product ansatz as follows:

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

One can extend the idea of the Hartree product ansatz as follows:

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

Because this is a determinant, permuting r_i, r_j swaps rows and therefore swaps sign – antisymmetry is enforced perfectly.

The Vandermonde determinant

Suppose we have some basis function φ applied to each electron configuration. The Vandermonde determinant is defined as:

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

This is a determinant: swapping r_i will switch sign, can enforce antisymmetry.

The Vandermonde determinant

Suppose we have some basis function φ applied to each electron configuration. The Vandermonde determinant is defined as:

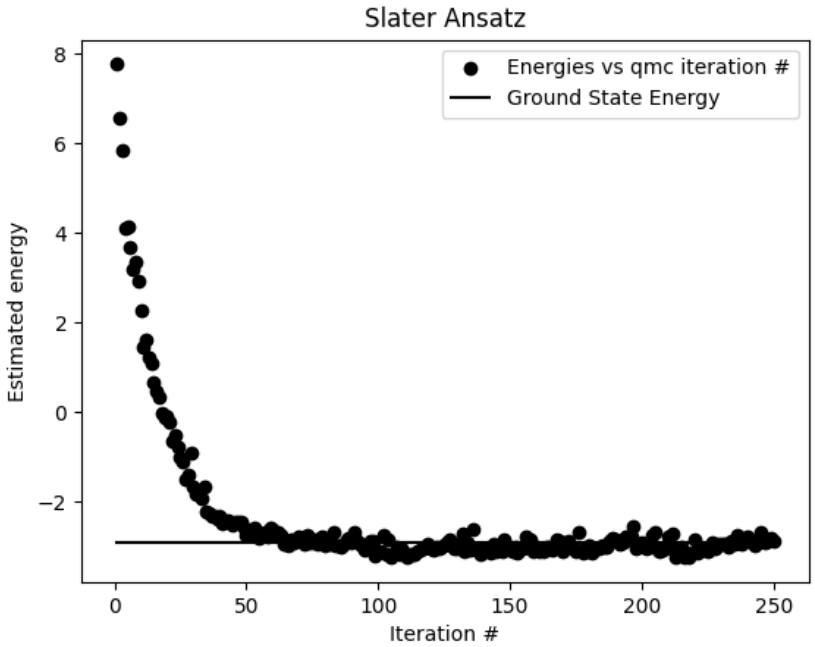
$$\det_V = \begin{vmatrix} 1 & \varphi(r_1)^1 & \varphi(r_1)^2 & \cdots & \varphi(r_1)^{N-1} \\ 1 & \varphi(r_2)^1 & \varphi(r_2)^2 & \cdots & \varphi(r_2)^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi(r_N)^1 & \varphi(r_N)^2 & \cdots & \varphi(r_N)^{N-1} \end{vmatrix}$$

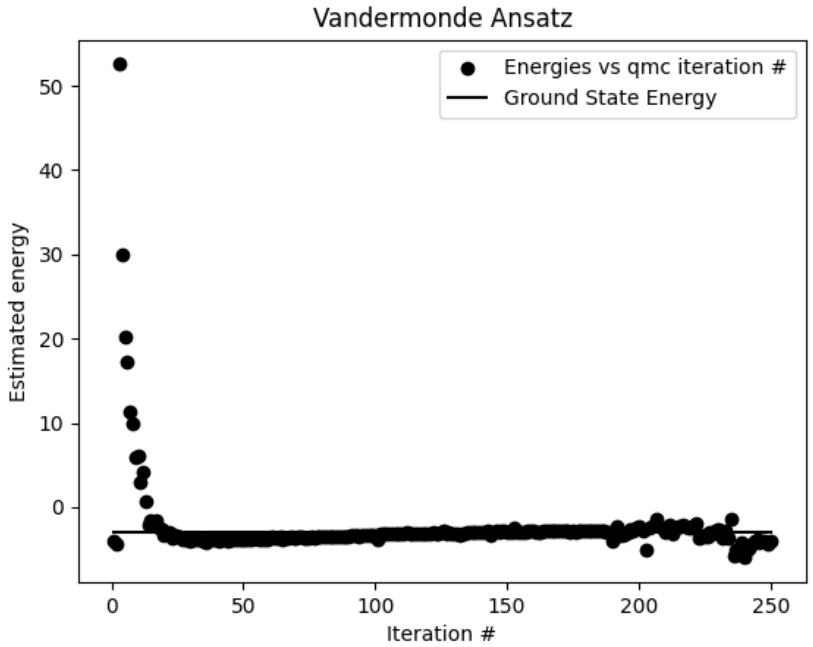
This is a determinant: swapping r_i will switch sign, can enforce antisymmetry.

But it can be computed using a more efficient expression.

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

Helium (2 electrons)
Slater vs Vandermonde

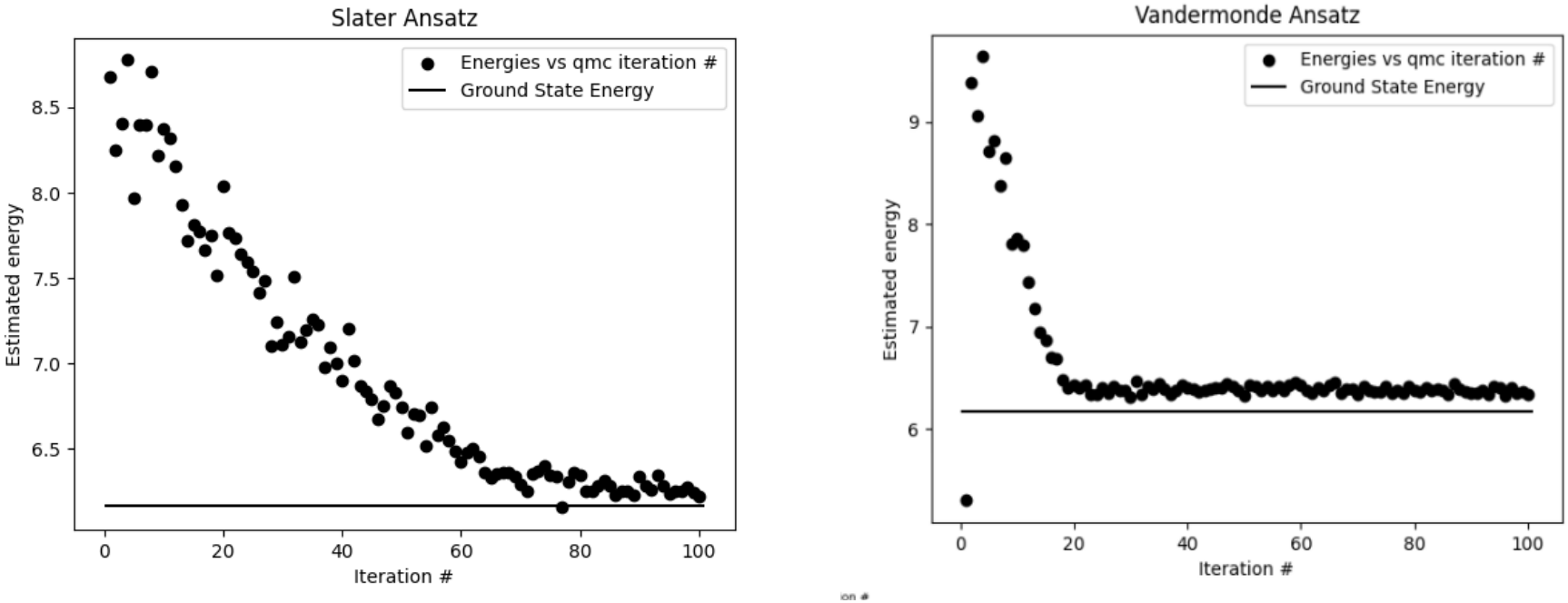




$$\Psi_{slat}(r_1, r_2) = e^{-\alpha_1 r_1} e^{-\alpha_2 r_2} - e^{-\alpha_1 r_2} e^{-\alpha_2 r_1}$$

$$\Psi_{Van}(r_1, r_2) = e^{-\alpha_1 (r_1 + r_2)} (e^{-\alpha_2 r_1} - e^{-\alpha_2 r_2})$$

Two particles in a box: Vandermonde and Slater



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

Outline for section 4

- It's quite natural to think of VMC as machine learning – we are minimizing a nonlinear function Ψ_α by gradient descent.

PauliNet

- It's quite natural to think of VMC as machine learning – we are minimizing a nonlinear function Ψ_α by gradient descent.
- PauliNet is one of a few recent deep learning architectures for approximation of wavefunctions.

PauliNet

- It's quite natural to think of VMC as machine learning – we are minimizing a nonlinear function Ψ_α by gradient descent.
- PauliNet is one of a few recent deep learning architectures for approximation of wavefunctions.
- The structure of the ansatz has a widely-used design – what is known as a "Slater-Jastrow-Backflow" ansatz. (QMCPACK uses a similar form.)

PauliNet

- It's quite natural to think of VMC as machine learning – we are minimizing a nonlinear function Ψ_α by gradient descent.
- PauliNet is one of a few recent deep learning architectures for approximation of wavefunctions.
- The structure of the ansatz has a widely-used design – what is known as a "Slater-Jastrow-Backflow" ansatz. (QMCPACK uses a similar form.)
- But parts of this are replaced with flexible neural networks. The weights of these networks are what varies to minimize the energy.

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

- φ are Hartree-Fock functions (physically derived single-electron wavefunctions, treated as inputs)

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

- φ are Hartree-Fock functions (physically derived single-electron wavefunctions, treated as inputs)
- J_α is what is known as the Jastrow factor, represented as a neural network.

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

- φ are Hartree-Fock functions (physically derived single-electron wavefunctions, treated as inputs)
- J_α is what is known as the Jastrow factor, represented as a neural network.
- $f_{\mu,\alpha}$ are the backflow factors which scale the Hartree-Fock functions: another neural network.

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

- φ are Hartree-Fock functions (physically derived single-electron wavefunctions, treated as inputs)
- J_α is what is known as the Jastrow factor, represented as a neural network.
- $f_{\mu,\alpha}$ are the backflow factors which scale the Hartree-Fock functions: another neural network.
- $\gamma(r)$ enforces electron-electron cusp conditions.

PauliNet: the ansatz itself

For electrons with coordinates r_i, the ansatz is as follows:

$$\Psi_\alpha(r) = e^{\gamma(r)+J_\alpha(r)} \sum_p c_p \det(\tilde{\varphi}^\uparrow_{\mu_p}(r)) \det(\tilde{\varphi}^\downarrow_{\mu_p}(r))$$

$$\tilde{\varphi}_\mu(r)_i = \varphi_\mu(r_i) f_{\mu,\alpha}(r)_i$$

- φ are Hartree-Fock functions (physically derived single-electron wavefunctions, treated as inputs)
- J_α is what is known as the Jastrow factor, represented as a neural network.
- $f_{\mu,\alpha}$ are the backflow factors which scale the Hartree-Fock functions: another neural network.
- $\gamma(r)$ enforces electron-electron cusp conditions.
- Slater determinants are used (split by spin up/down of particles).

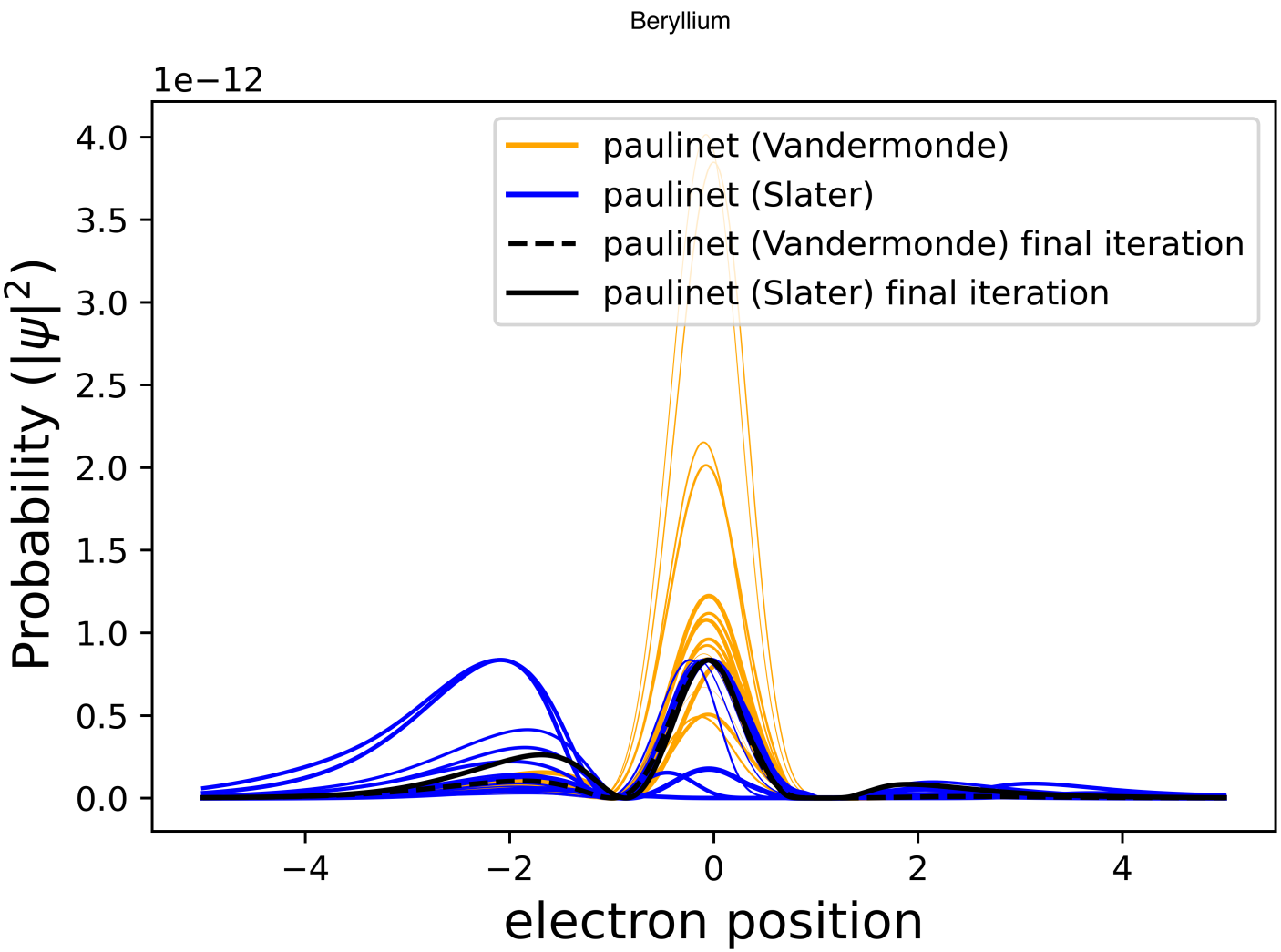
Modification to PauliNet: Vandermonde Determinants

- We replace the Slater determinants in the PauliNet ansatz with Vandermonde determinants.
- We define $\tilde{\varphi}(r_j) = \prod_{i=1}^{N} \tilde{\varphi}_i(r_j)$ and then take a Vandermonde determinant of $(\tilde{\varphi}(r_1), \tilde{\varphi}(r_2), \cdots)$

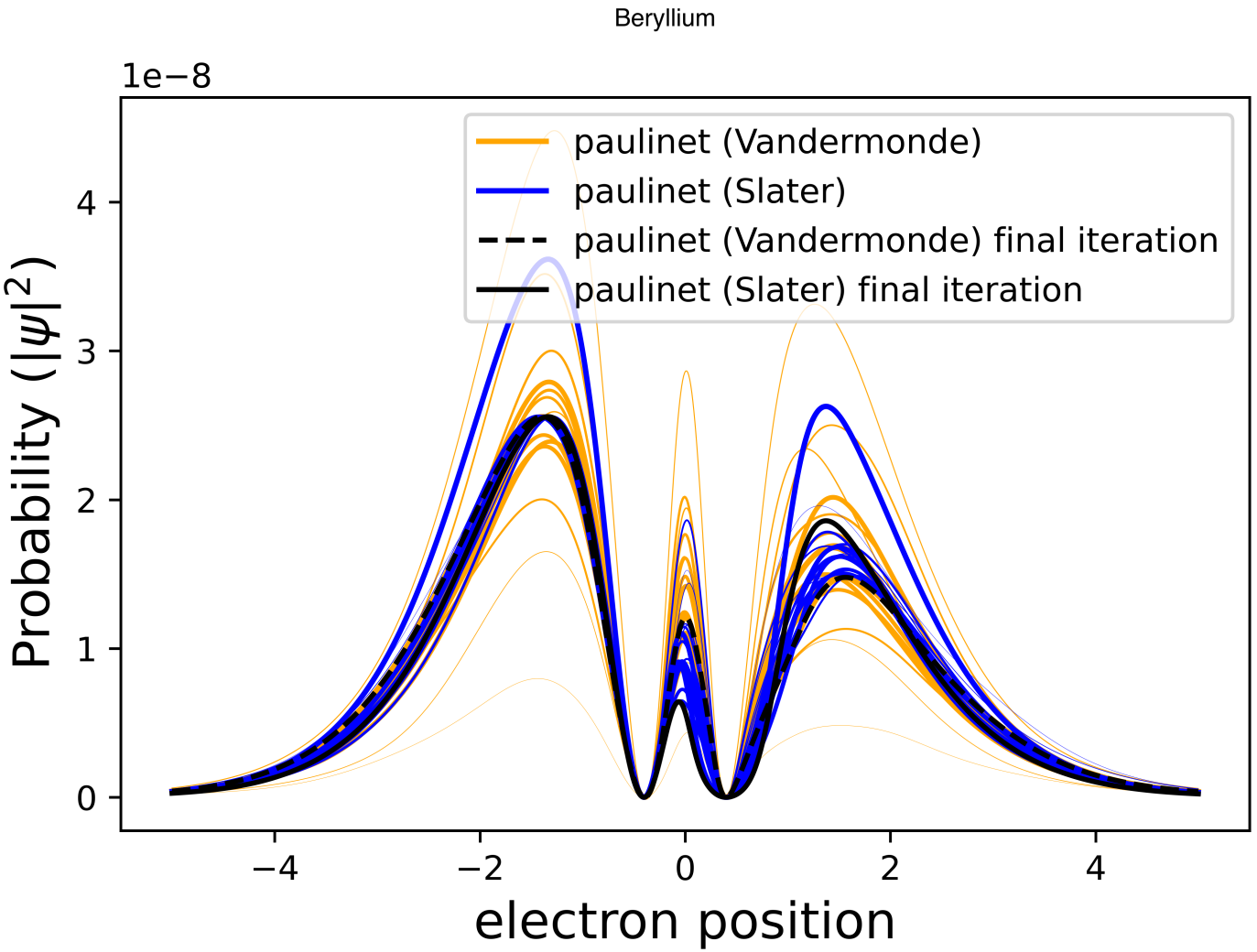
Modification to PauliNet: Vandermonde Determinants

- We replace the Slater determinants in the PauliNet ansatz with Vandermonde determinants.
- We define $\tilde{\varphi}(r_j) = \prod_{i=1}^{N} \tilde{\varphi}_i(r_j)$ and then take a Vandermonde determinant of $(\tilde{\varphi}(r_1), \tilde{\varphi}(r_2), \cdots)$
- Certain boundary conditions are preserved by Slater determinant but lost when moving to the Vandermonde determinant. We premultiply by a wavefunction applied to each electron $\varphi(r_1)\varphi(r_2)\cdots$ to preserve this property.

PauliNet: Slater v Vandermonde



PauliNet: Slater v Vandermonde



Outline for section 5

The Vandermonde determinant

- We find that the Vandermonde determinant is a viable way to enforce antisymmetry, as opposed to the classic Slater determinant.

The Vandermonde determinant

- We find that the Vandermonde determinant is a viable way to enforce antisymmetry, as opposed to the classic Slater determinant.
- On our simple systems, a Vandermonde-based ansatz can sometimes compete with a Slater determinant.

The Vandermonde determinant

- We find that the Vandermonde determinant is a viable way to enforce antisymmetry, as opposed to the classic Slater determinant.
- On our simple systems, a Vandermonde-based ansatz can sometimes compete with a Slater determinant.
- When integrated with PauliNet, the Vandermonde determinant trained successfully but did not surpass a simple baseline in converged energy.
  - One possible avenue for improvement might be a set of basis wavefunctions more suited to the Vandermonde determinant, or different methods for enforcing boundary conditions.

The Vandermonde determinant

- We find that the Vandermonde determinant is a viable way to enforce antisymmetry, as opposed to the classic Slater determinant.
- On our simple systems, a Vandermonde-based ansatz can sometimes compete with a Slater determinant.
- When integrated with PauliNet, the Vandermonde determinant trained successfully but did not surpass a simple baseline in converged energy.
  - One possible avenue for improvement might be a set of basis wavefunctions more suited to the Vandermonde determinant, or different methods for enforcing boundary conditions.
- 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.

Conclusions and new directions

- Machine learning can definitely be employed in the design of ansatz for Variational Monte Carlo. There is a real possibility these techniques will be adopted into HPC packages like QMCPACK.

Conclusions and new directions

- Machine learning can definitely be employed in the design of ansatz for Variational Monte Carlo. There is a real possibility these techniques will be adopted into HPC packages like QMCPACK.
- In principle a neural network can learn any arbitrary function. But even with the most complex and flexible deep learning ansatz, enforcement of physical properties by construction seems to be quite valuable.

Conclusions and new directions

- Machine learning can definitely be employed in the design of ansatz for Variational Monte Carlo. There is a real possibility these techniques will be adopted into HPC packages like QMCPACK.
- In principle a neural network can learn any arbitrary function. But even with the most complex and flexible deep learning ansatz, enforcement of physical properties by construction seems to be quite valuable.
- We have focused only on the ground state and the time-independent Schrödinger equation. Considering whether VMC techniques could be used to solve for higher excited states, or a time-evolving version of the equation, would be interesting.

Acknowledgments

- IPAM
  - Dr. Susana Serna (Dir of RIPS, IPAM)
  - Dr. Christian Ratsch (Deputy Director, IPAM)
  - Dr. Dimitri Shlyakhtenko (Dir of IPAM)
- IPAM staff
  - Neli Petrosyan
  - Parama Sigurdsen
  - Kayleigh Steele
  - David Medina
  - Jim Kimmick
- AMD
  - Dr. Nicholas Malaya (Industry Mentor)

Questions?