DeepMind

# Quantum Monte Carlo and Solving the Many-Electron Schrödinger Equation with Deep Neural Networks

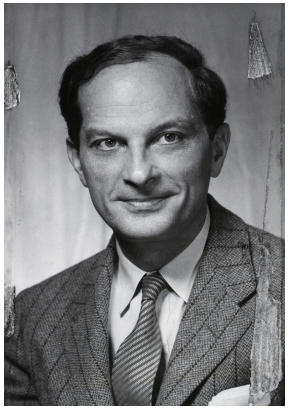
David Pfau[1*], James Spencer[1*], Alex Matthews[1], W. M. C. Foulkes[2]
[1]DeepMind
[2]Imperial College London
[*]{pfau, jamessspencer}@google.com

arXiv:1909.02487

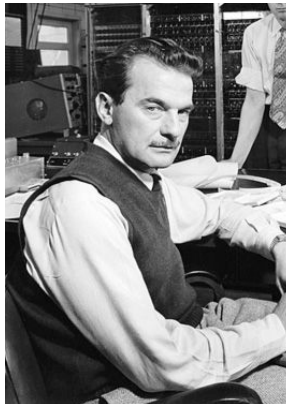
Imperial College London

MCQMC
12nd August 2020

# A Brief History of Monte Carlo Methods

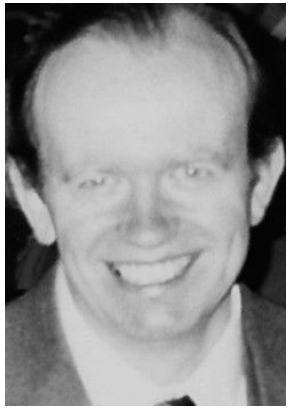


Ulam



Metropolis

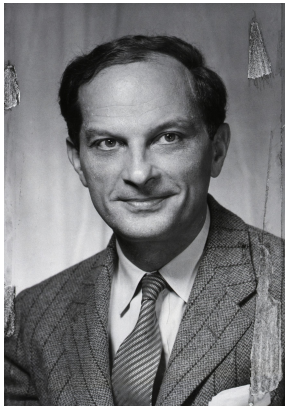
Von Neumann



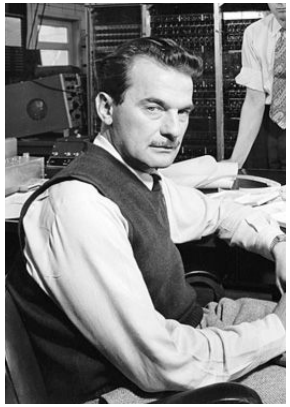
Hastings

- Developed by Ulam, Von Neumann and Metropolis at Los Alamos in the ‘40s
- Published in *"Equations of state calculations by fast computing machines"* (1953) by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller
- Major application of some of the first computing machines - ENIAC, MANIAC
- Wide variety of computational physics applications:
  - Neutron diffusion
  - Equations of state in statistical mechanics
  - Many-body quantum physics: W. L. McMillan, *“Ground state of liquid $He^4$”* (1965)
  - Lattice QCD: S. Duane et al. (1987) - Origin of HMC!
- Often adopted later in statistic and ML:
  - Metropolis-Hastings: W. K. Hastings (1970)
  - Gibbs sampling: S. Geman and D. Geman (1984)
  - HMC: R. Neal (1996)

# A Brief History of Monte Carlo Methods

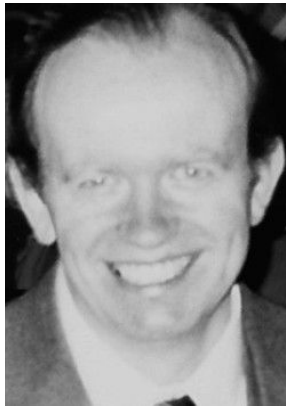


Ulam



Metropolis

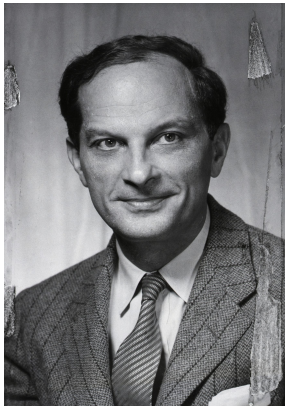
Von Neumann



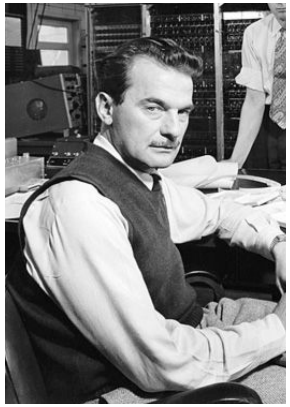
Hastings

- Developed by Ulam, Von Neumann and Metropolis at Los Alamos in the ‘40s
- Published in *"Equations of state calculations by fast computing machines"* (1953) by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller
- Major application of some of the first computing machines - ENIAC, MANIAC
- Wide variety of computational physics applications:
  - Neutron diffusion
  - Equations of state in statistical mechanics
  - **Many-body quantum physics**: W. L. McMillan, *“Ground state of liquid $He^4$”* (1965)
  - Lattice QCD: S. Duane et al. (1987) - Origin of HMC!
- Often adopted later in statistic and ML:
  - Metropolis-Hastings: W. K. Hastings (1970)
  - Gibbs sampling: S. Geman and D. Geman (1984)
  - HMC: R. Neal (1996)

# A Brief History of Monte Carlo Methods



Ulam



Metropolis

Von Neumann

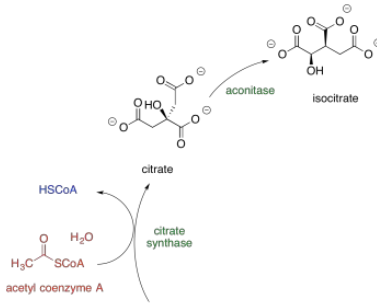
Hastings

- Developed by Ulam, Von Neumann and Metropolis at Los Alamos in the ‘40s
- Published in *"Equations of state calculations by fast computing machines"* (1953) by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller
- Major application of some of the first computing machines - ENIAC, MANIAC
- Wide variety of computational physics applications:
  - Neutron diffusion
  - Equations of state in statistical mechanics
  - **Many-body quantum physics**: W. L. McMillan, *“Ground state of liquid $He^4$”* (1965)
  - Lattice QCD: S. Duane et al. (1987) - Origin of HMC!
- Often adopted later in statistic and ML:
  - Metropolis-Hastings: W. K. Hastings (1970)
  - Gibbs sampling: S. Geman and D. Geman (1984)
  - HMC: R. Neal (1996)
- **Now is the time for ideas from ML to instead feed back into computational physics**

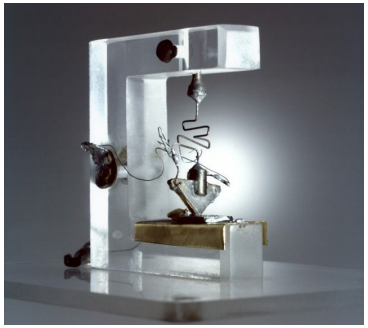
# QMC: ~~Quási~~ Monte Carlo

QÚANTUM

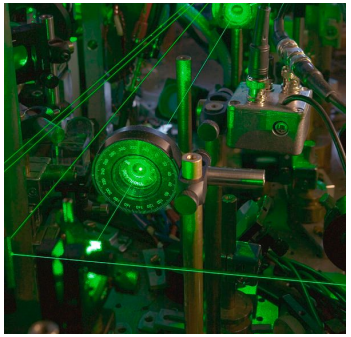
# Quantum Mechanics and the Schrödinger Equation



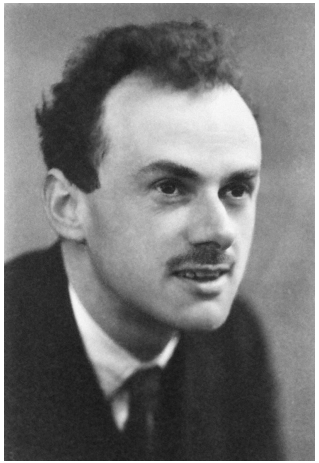

Chemical Reactions



Electronic properties of materials



Exotic states of matter



*The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.*

**Paul Dirac**

# Quantum Mechanics and the Schrödinger Equation

$$\hat{H}\Psi = E\Psi$$

$$\hat{H} = -\frac{1}{2}\sum_i \nabla_i^2 - \sum_{iI} \frac{Z_I}{|r_i - R_I|} + \sum_{ij} \frac{1}{|r_i - r_j|} + \sum_{IJ} \frac{Z_I Z_J}{|r_I - r_J|}$$

Kinetic Energy | Electron-Ion Potential | Electron-Electron Potential | Ion-Ion Potential

- H is a linear operator on 3N-dimensional functions Ψ for a system of N particles
- Ψ is a function such that $\Psi^2$ is a probability density function
- Eigenfunctions of H represent states of constant energy - the lowest energy state is the *ground state*
- Can be solved exactly for hydrogen - and nothing else!

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)
- Comes in two (main) flavors:

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)
- Comes in two (main) flavors:
  - **Variational** QMC - directly minimize the upper bound

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)
- Comes in two (main) flavors:
  - **Variational** QMC - directly minimize the upper bound
  - **Diffusion** QMC - simulate particles undergoing diffusion, update weights based on potential energy

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)
- Comes in two (main) flavors:
  - **Variational** QMC - directly minimize the upper bound
  - **Diffusion** QMC - simulate particles undergoing diffusion, update weights based on potential energy
- Diffusion QMC is generally more accurate, and can be interpreted as **particle filtering**, where potential energy = log likelihood

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# Quantum Monte Carlo

- Solve for ground state energy by Monte Carlo
- Originally used just to evaluate integrals for liquid helium: W. L. McMillan (1965)
- Extended to fermions = electrons = chemical and material systems: D. Ceperley, G. V. Chester and M. H. Kalos (1977)
- Comes in two (main) flavors:
  - **Variational** QMC - directly minimize the upper bound
  - **Diffusion** QMC - simulate particles undergoing diffusion, update weights based on potential energy
- Diffusion QMC is generally more accurate, and can be interpreted as **particle filtering**, where potential energy = log likelihood
- This talk is focused on **variational** QMC

$$\min_{\Psi} \frac{\langle \Psi \hat{H} \Psi \rangle}{\langle \Psi^2 \rangle} =$$

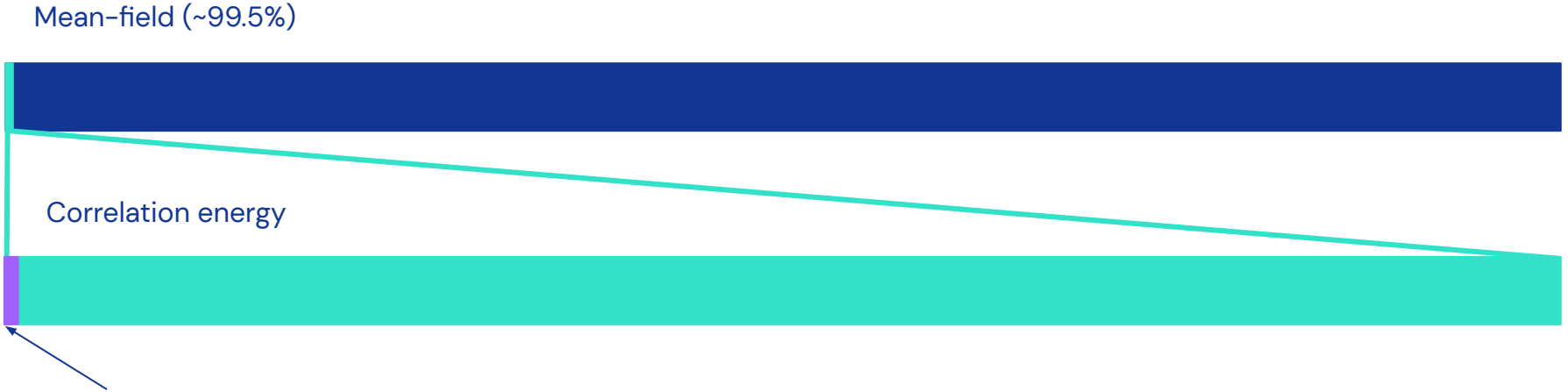
$$\min_{\Psi} \frac{\int d\mathbf{r} \Psi(\mathbf{r}) \hat{H} \Psi(\mathbf{r})}{\int d\mathbf{r} \Psi^2(\mathbf{r})} =$$

$$\min_{\Psi} \mathbb{E}_{x \sim p} [\Psi^{-1}(\mathbf{r}) \hat{H} \Psi(\mathbf{r})]$$

$$p(\mathbf{r}) \propto \Psi^2(\mathbf{r})$$

# “Chemical Accuracy”






- To be useful to chemists, the standard benchmark is “chemical accuracy” or 1 kCal/mol
- This is a tiny fraction of the total energy - something like 99.995% accuracy
- Mean field methods already capture ~99.5% of the total energy - need 99% of the remainder

# How do we approximate a wavefunction?

# Wavefunctions for electrons must be *antisymmetric*

$$\Psi(\ldots, \mathbf{r}_i, \ldots, \mathbf{r}_j, \ldots) = -\Psi(\ldots, \mathbf{r}_j, \ldots, \mathbf{r}_i, \ldots)$$

$$\Psi(\{\mathbf{r}_i\}) = \begin{vmatrix} \phi_1(\mathbf{r}_1) & \dots & \phi_1(\mathbf{r}_N) \\ \vdots & & \vdots \\ \phi_N(\mathbf{r}_1) & \dots & \phi_N(\mathbf{r}_N) \end{vmatrix}$$

Slater Determinant

- Minimum energy solution gives mean-field or “Hartree-Fock” solution for ground state

$$\Psi(\{\mathbf{r}_i\}) = \exp\left(-\sum_{i<j} U(|\mathbf{r}_i - \mathbf{r}_j|)\right) \begin{vmatrix} \phi_1(\mathbf{r}_1) & \dots & \phi_1(\mathbf{r}_N) \\ \vdots & & \vdots \\ \phi_N(\mathbf{r}_1) & \dots & \phi_N(\mathbf{r}_N) \end{vmatrix}$$

Jastrow Factor

Slater Determinant

➢ Adds correlation between electrons at the output

$$\mathbf{r}_i' = \mathbf{r}_i + \sum_{j \neq i} \eta(|\mathbf{r}_i - \mathbf{r}_j|)\mathbf{r}_j$$

Backflow Transform

$$\Psi(\{\mathbf{r}_i\}) = \exp\left(-\sum_{i<j} U(|\mathbf{r}_i - \mathbf{r}_j|)\right) \begin{vmatrix} \phi_1(\mathbf{r}_1') & \cdots & \phi_1(\mathbf{r}_N') \\ \vdots & & \vdots \\ \phi_N(\mathbf{r}_1') & \cdots & \phi_N(\mathbf{r}_N') \end{vmatrix}$$

Jastrow Factor

Slater Determinant

➢ Adds correlation between electrons at the *input*

$$\mathbf{r}_i' = \mathbf{r}_i + \sum_{j \neq i} \eta(|\mathbf{r}_i - \mathbf{r}_j|)\mathbf{r}_j$$

Backflow Transform

$$\Psi(\{\mathbf{r}_i\}) = \exp\left(-\sum_{i<j} U(|\mathbf{r}_i - \mathbf{r}_j|)\right) \sum_k \omega_k \begin{vmatrix} \phi_1^k(\mathbf{r}_1') & \cdots & \phi_1^k(\mathbf{r}_N') \\ \vdots & & \vdots \\ \phi_N^k(\mathbf{r}_1') & \cdots & \phi_N^k(\mathbf{r}_N') \end{vmatrix}$$

- Build a larger linear basis of antisymmetric functions
- Often requires thousands or millions to converge
- Classic *Slater-Jastrow-Backflow* wavefunction:
  Y. Kwon, D. Ceperley and R. M. Martin (1993)
- **Can we improve on this with modern function approximation from machine learning?**

# Neural networks are excellent function approximators

# Neural networks are excellent function approximators

## Object Recognition


A. Krizhevsky, I. Sutskever and G. Hinton (2012)

# Neural networks are excellent function approximators

## Object Recognition


A. Krizhevsky, I. Sutskever and G. Hinton (2012)

## Game Playing

D. Silver, A. Huang, C. Maddison *et al.* (2016)

# Neural networks are excellent function approximators

## Object Recognition


A. Krizhevsky, I. Sutskever and G. Hinton (2012)

## Game Playing

D. Silver, A. Huang, C. Maddison *et al.* (2016)

## Language

```
This is a story about Nell (short
for Penelope), a 12-year old girl
who lives in a planet called New
Texas. She and her whole family
lives with another family in a
settlement with few houses, called a
soddy. One day Nell has a weird
dream that a storm is coming. When
she wakes up she discovers that she
is not crazy, it is really a storm
coming. The storm is about to
destroy the settlement. Nell and her
younger brother Cip run away from
the house to look for shelter. They
```

T. B. Brown, B. Mann, N. Ryder *et al.* (2020)

# Why not a wavefunction?

# Why not a wavefunction?

# How do we build an *antisymmetric* neural network?

# Fermionic Neural Network

Key idea: extend function of one electron to be functions of ***all*** electrons, but not change under permutation of other electrons

$$\phi(\mathbf{r}_i) \rightarrow \phi(\mathbf{r}_i; \mathbf{r}_1, \ldots, \mathbf{r}_{i-1}, \mathbf{r}_{i+1}, \ldots, \mathbf{r}_N)$$

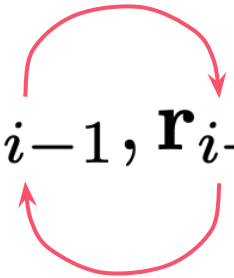
- Orbital invariant under change of order of electrons other than ***i***
- Determinant remains antisymmetric
- Can capture many-electron interactions directly in the orbital

How to represent these permutation-equivariant functions?

# Fermionic Neural Network

Key idea: extend function of one electron to be functions of ***all*** electrons, but not change under permutation of other electrons

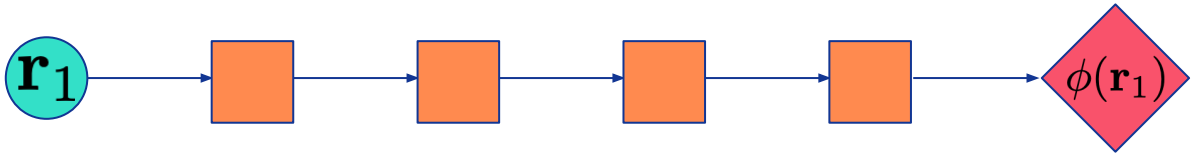
$$\phi(\mathbf{r}_i) \rightarrow \phi(\mathbf{r}_i; \mathbf{r}_1, \ldots, \mathbf{r}_{i-1}, \mathbf{r}_{i+1}, \ldots, \mathbf{r}_N)$$



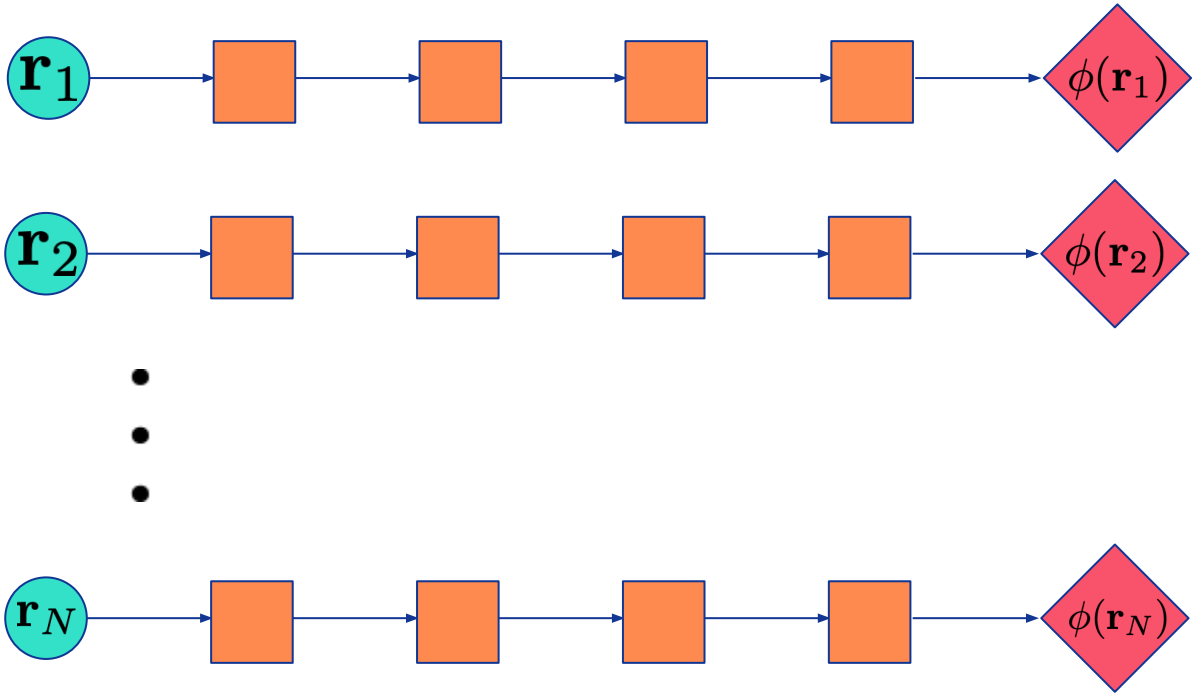
- Orbital invariant under change of order of electrons other than ***i***
- Determinant remains antisymmetric
- Can capture many-electron interactions directly in the orbital

How to represent these permutation-equivariant functions?

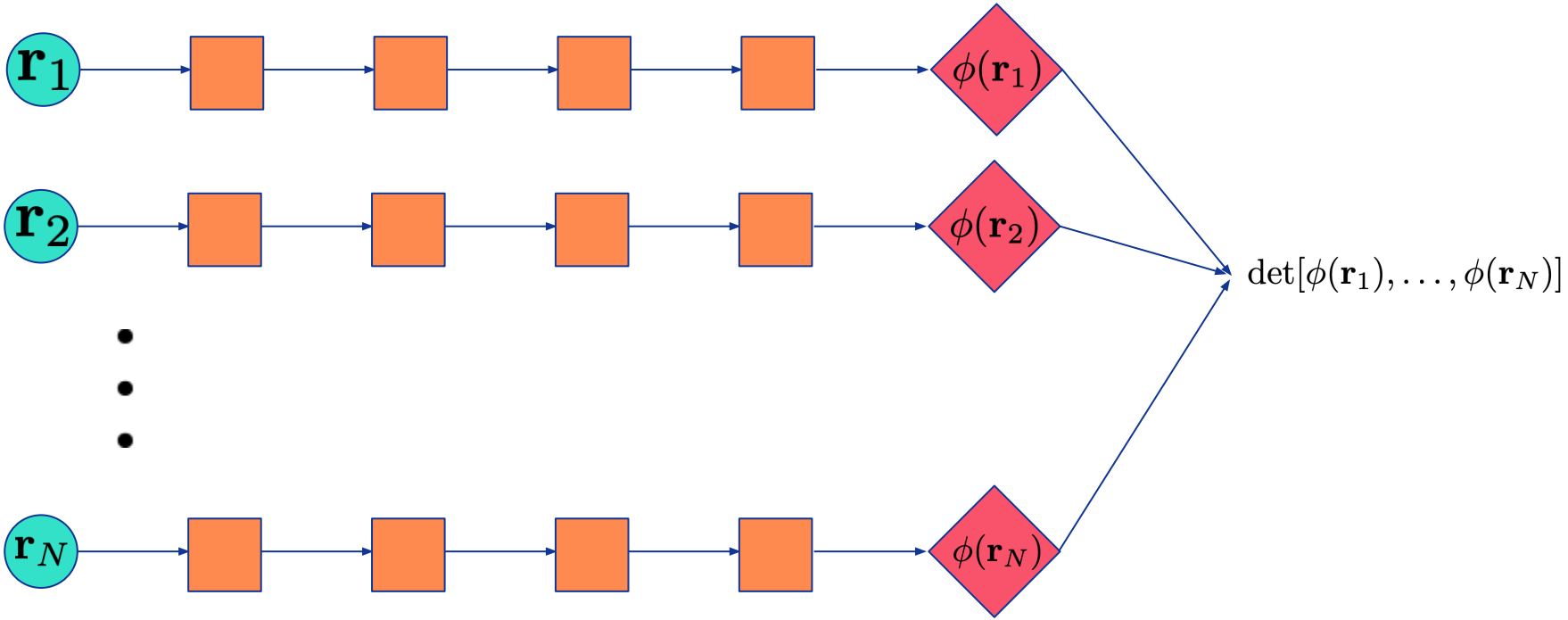
# Fermionic Neural Network



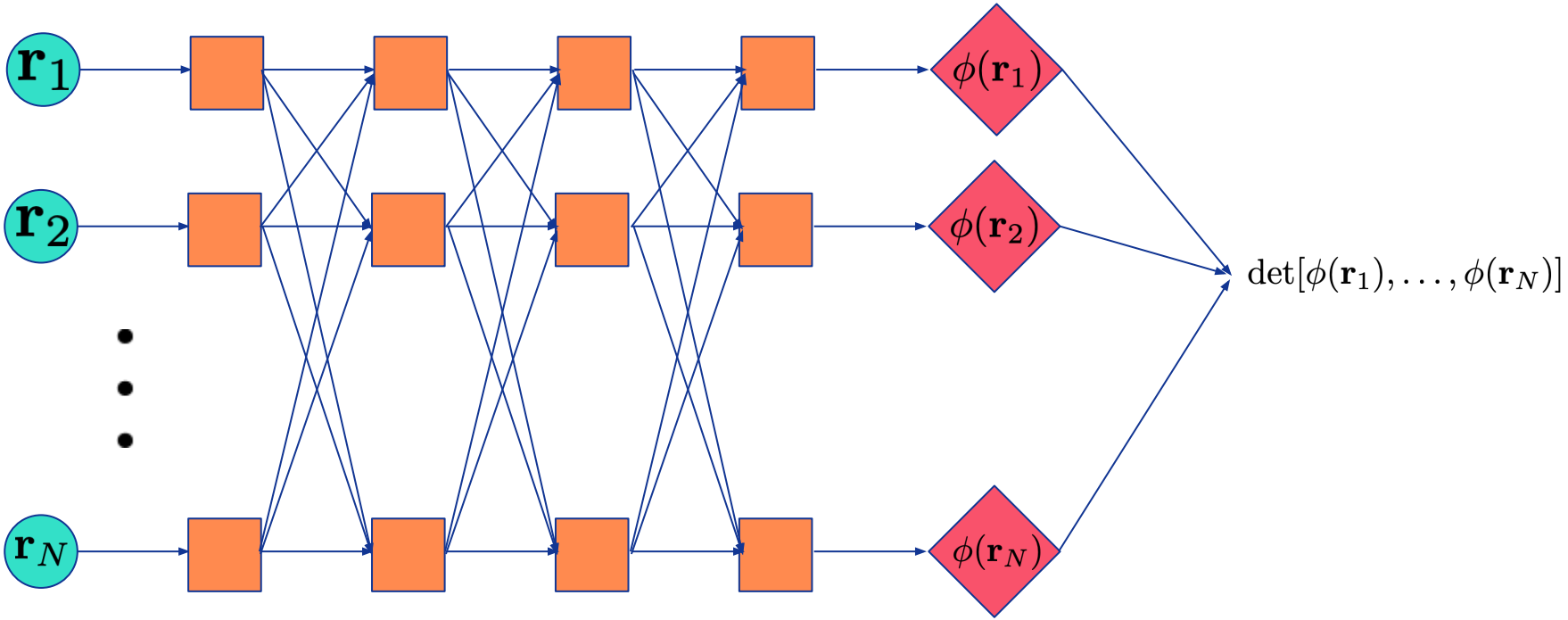
# Fermionic Neural Network



# Fermionic Neural Network

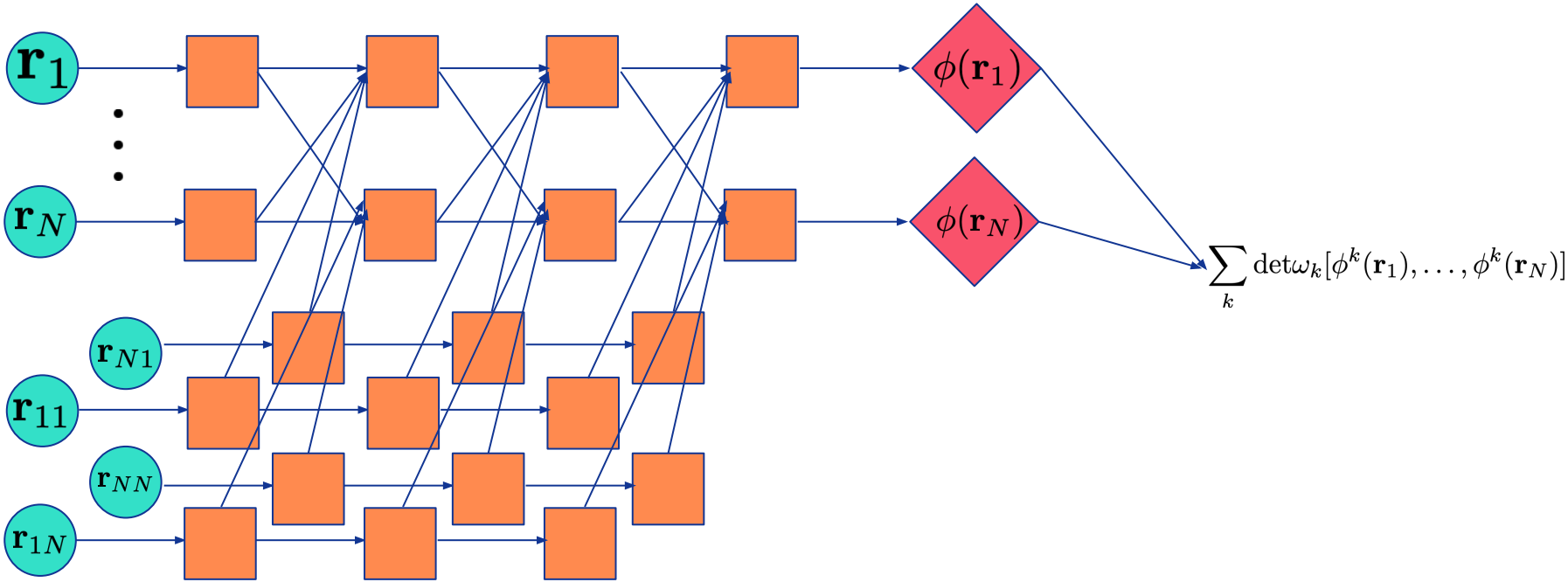


# Fermionic Neural Network




- Just sum the activation vectors from other streams together and add them as inputs to the next layer
- Simple, but universal in theory: M. Hutter *arXiv:2007.15298* (2020)

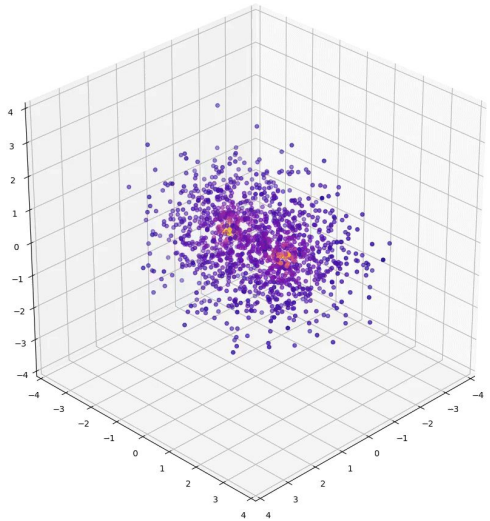
# Fermionic Neural Network




- In practice, need to add a second stream with pairs of electron features, and have a sum of a small number of determinants (a few dozen)

# Implementation: MCMC

- Nothing fancy - just conventional Metropolis-Hastings
- Gibbs sampling/One-electron moves are often used, but not needed for small systems
- Interaction between optimization and sampling is very subtle
  - Gradient-based methods (HMC, MALA) are actually *much noisier*
  - Often smaller steps, slower mixing leads to faster convergence (?!?)
  - Promising topic for deeper theoretical work

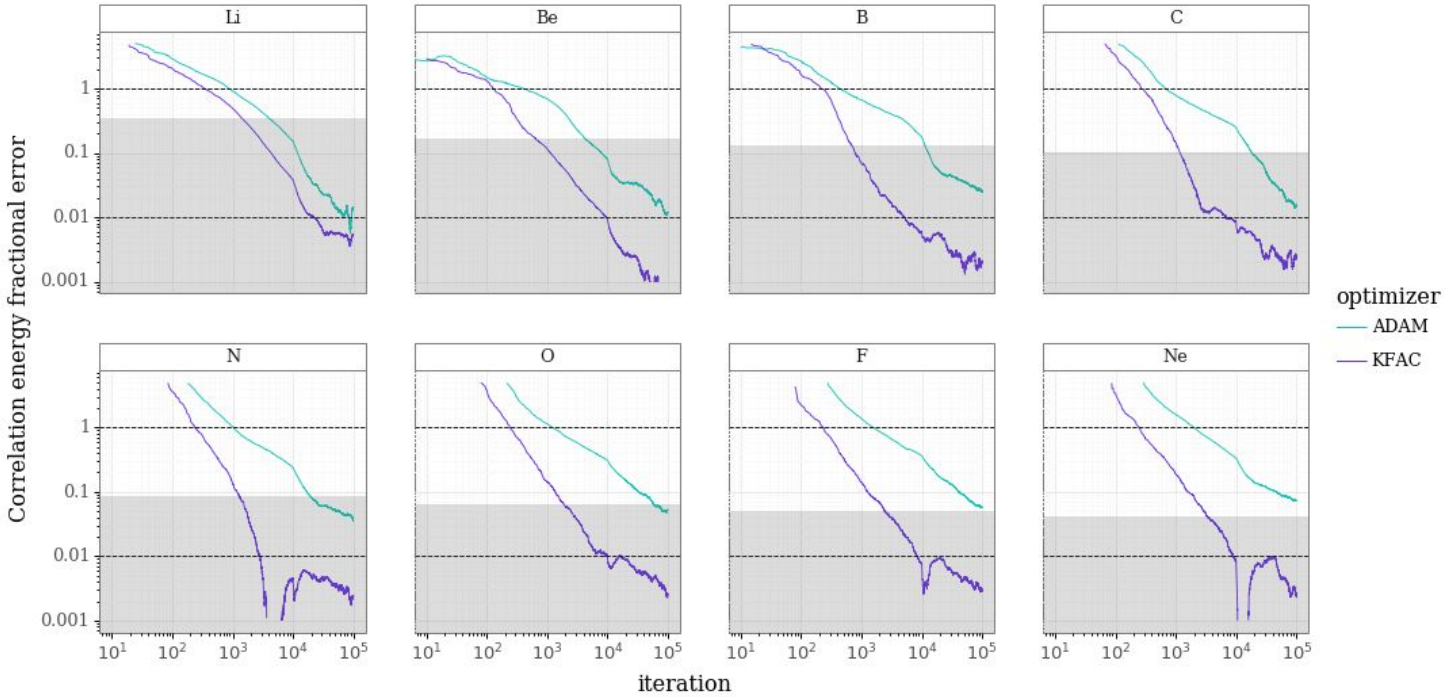


# Implementation: Custom Derivatives

$$\frac{\partial \det[\mathbf{X}]}{\partial X_{ij}} = \det[\mathbf{X}] X^{-1}_{ij}$$

- Have to compute lots of derivatives *and second derivatives* of determinants
- Singular matrices are a problem - conventional AD packages blow up
- Can implement custom derivatives by taking matrix inverse and determinant from SVD and canceling terms
- Couldn't have done it without *An extended collection of matrix derivative results…*

# Implementation: Kronecker-Factored Approximate Curvature



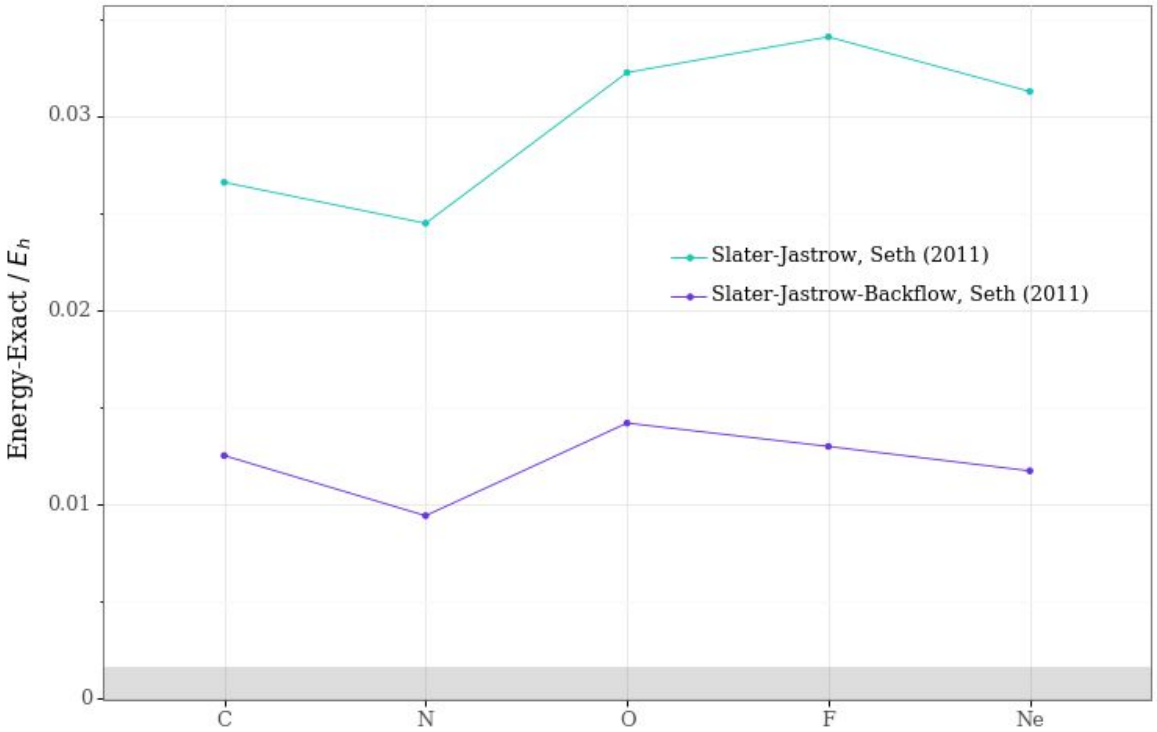

**Natural gradient descent** S.-I. Amari (1998) $d\Theta \propto \mathcal{F}^{-1}\nabla_\Theta \mathcal{L}(\Theta)$

Equivalent to stochastic reconfiguration (S. Sorella (1998)) with unnormalised probability densities

**KFAC** Approximates Fisher information matrix: J. Martens and R. Grosse (2015)

# Results

# Single determinant

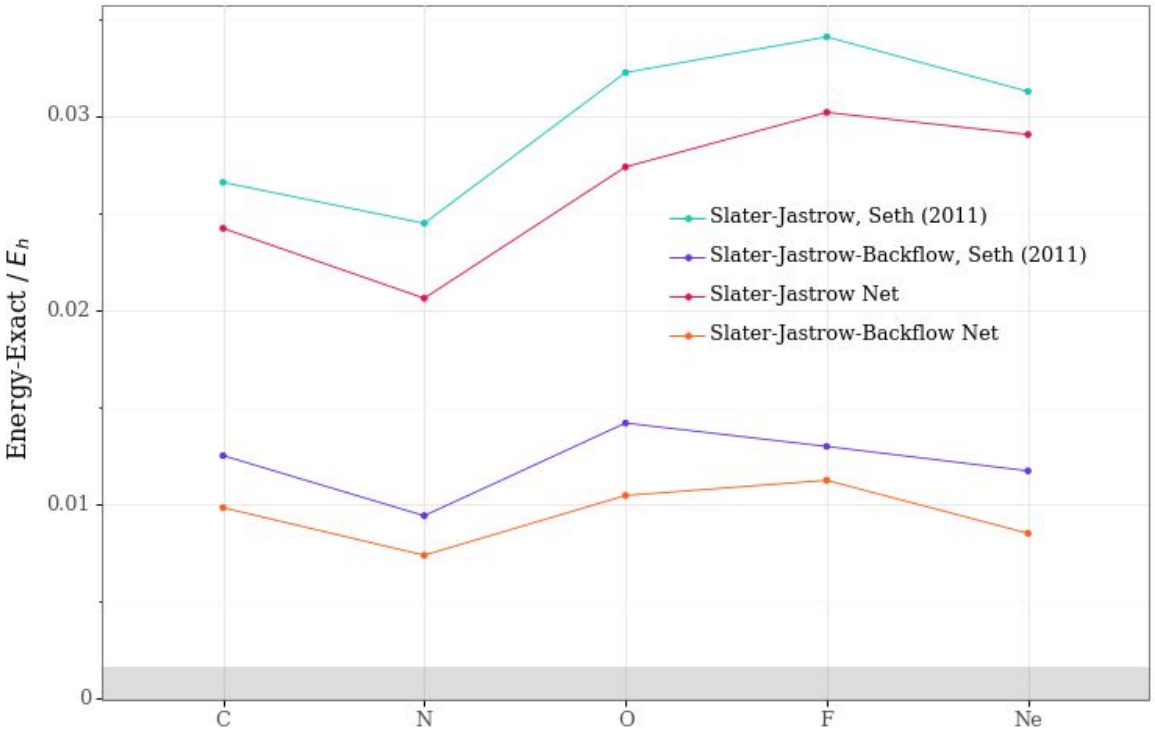



Backflow captures large fraction of remaining correlation energy.

VMC: P. Seth, P. L. Ríos, and R. J. Needs, J. Chem. Phys. 134, 084105 (2011).

Exact: Chakravorty *et al.* Phys. Rev. A 47, 3649 (1993).

# Single determinant



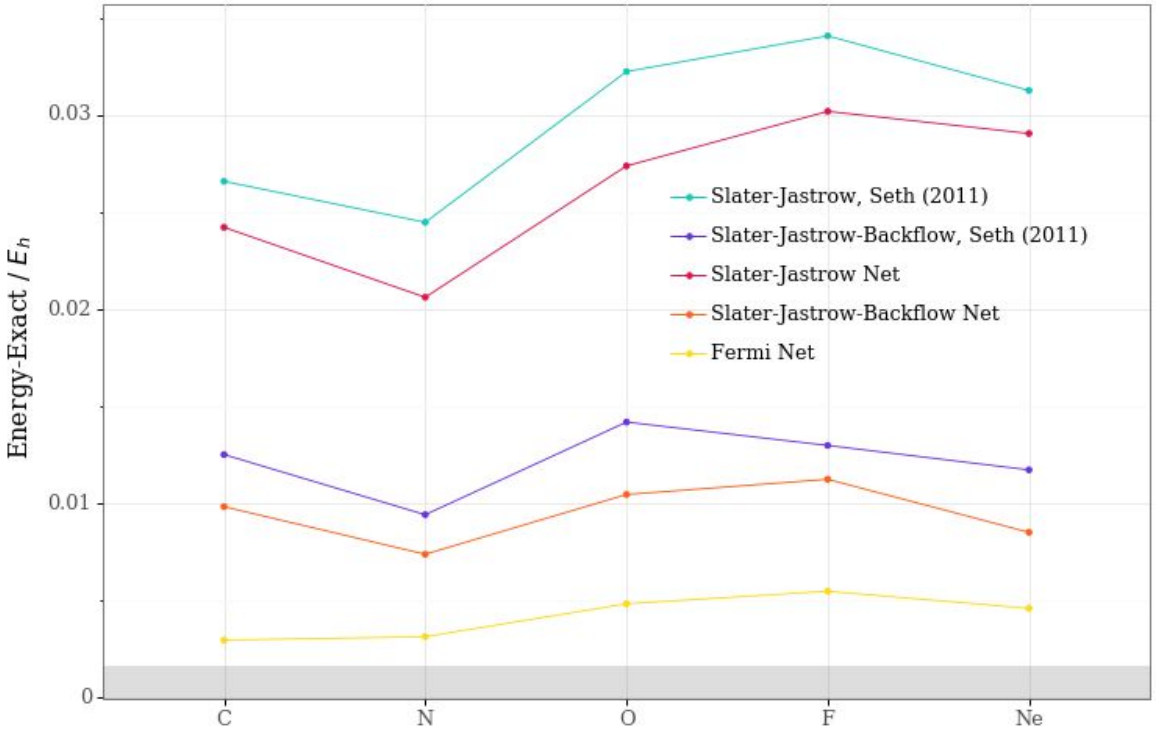

Backflow captures large fraction of remaining correlation energy.

Neural networks offer more flexible functional forms.

VMC, DMC: P. Seth, P. L. Ríos, and R. J. Needs, J. Chem. Phys. 134, 084105 (2011).

Exact: Chakravorty *et al.* Phys. Rev. A 47, 3649 (1993).

# Single determinant




Backflow captures large fraction of remaining correlation energy.

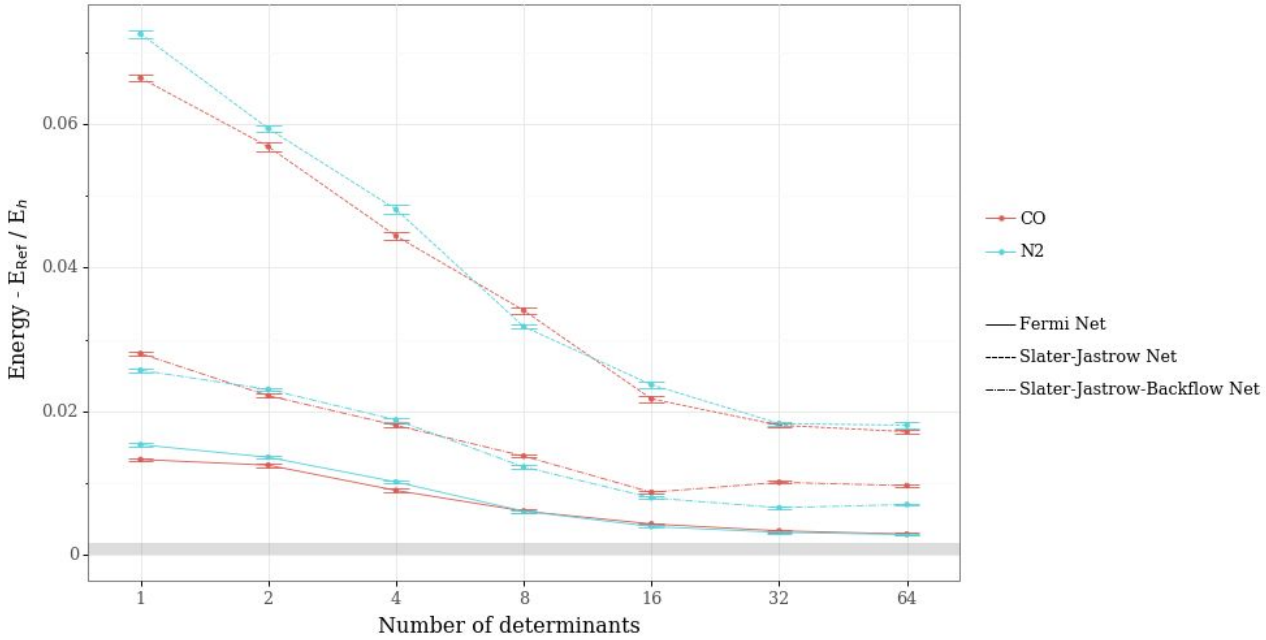
Neural networks offer more flexible functional forms.

Fermi Net substantially reduces the error.

VMC, DMC: P. Seth, P. L. Ríos, and R. J. Needs, J. Chem. Phys. 134, 084105 (2011).

Exact: Chakravorty *et al.* Phys. Rev. A 47, 3649 (1993).

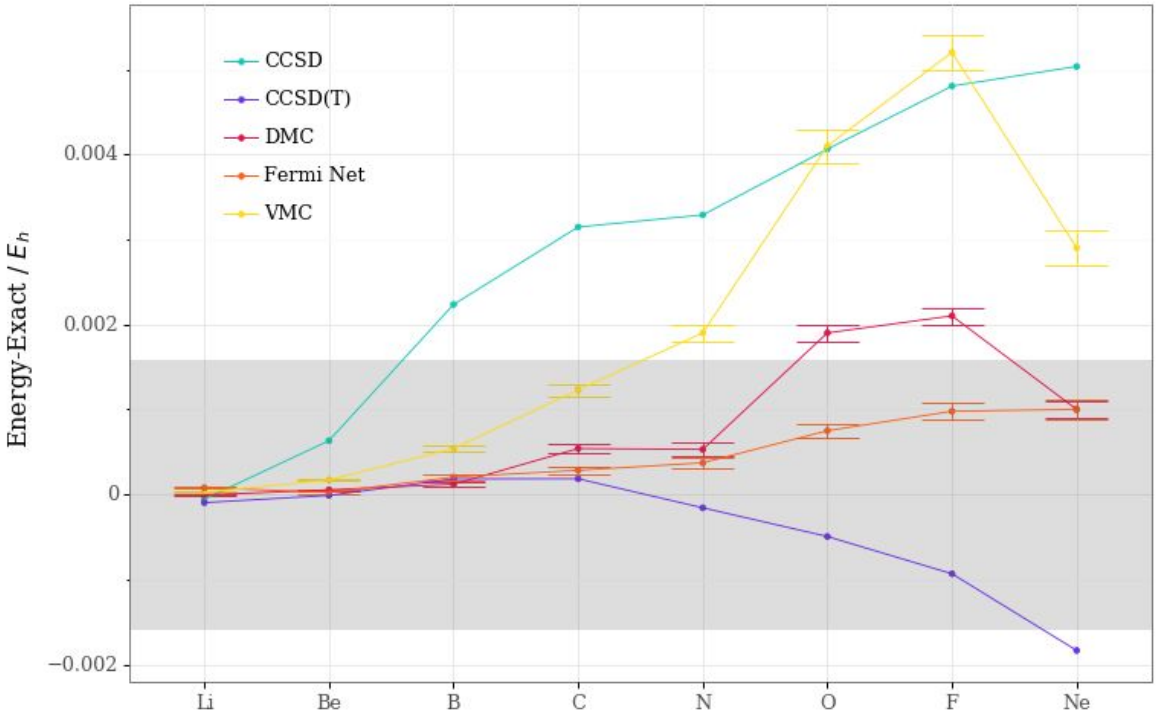
# Multiple determinants




Fermi Net converges rapidly with number of determinants

Substantially easier to optimize than Slater-Jastrow and Slater-Jastrow-Backflow networks!

# Ground state atoms




Fermi Net: 16 determinants.

VMC, DMC: 50-100 configuration state functions. SJB wavefunction.

Fermi Net outperforms conventional VMC **and** DMC.

Consistently captures 99.7% correlation energy.

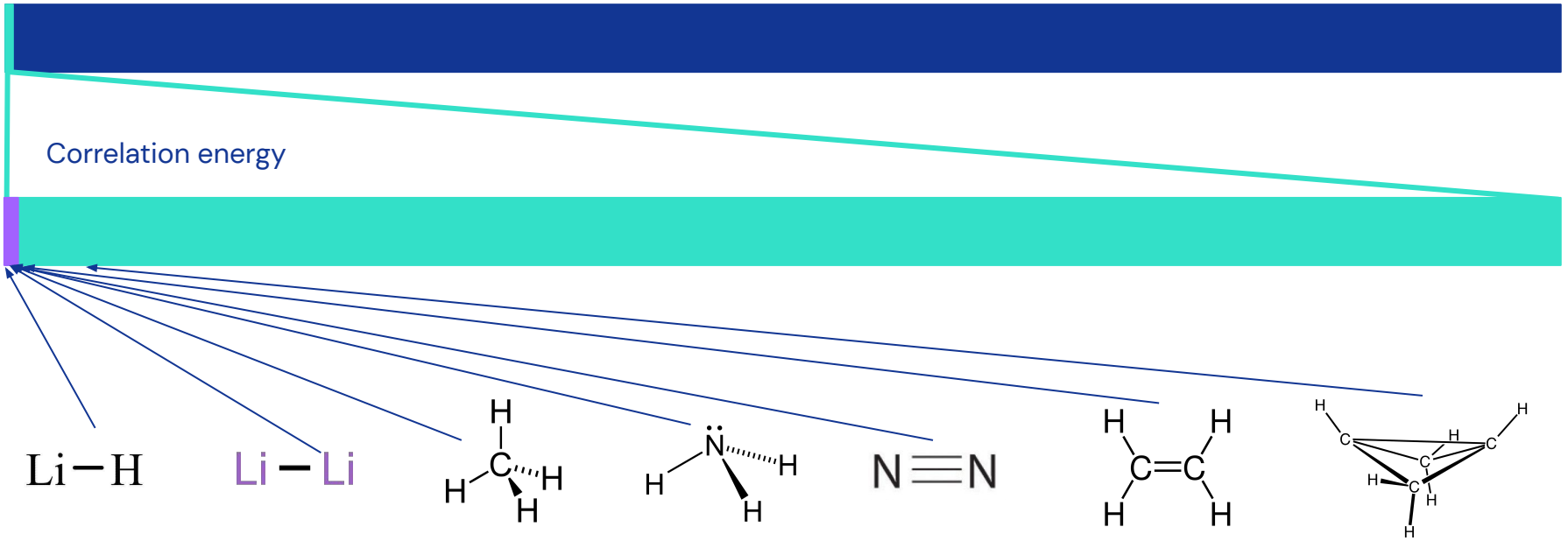
VMC, DMC: P. Seth, P. L. Ríos, and R. J. Needs, J. Chem. Phys. 134, 084105 (2011).

Exact: Chakravorty *et al.* Phys. Rev. A 47, 3649 (1993).

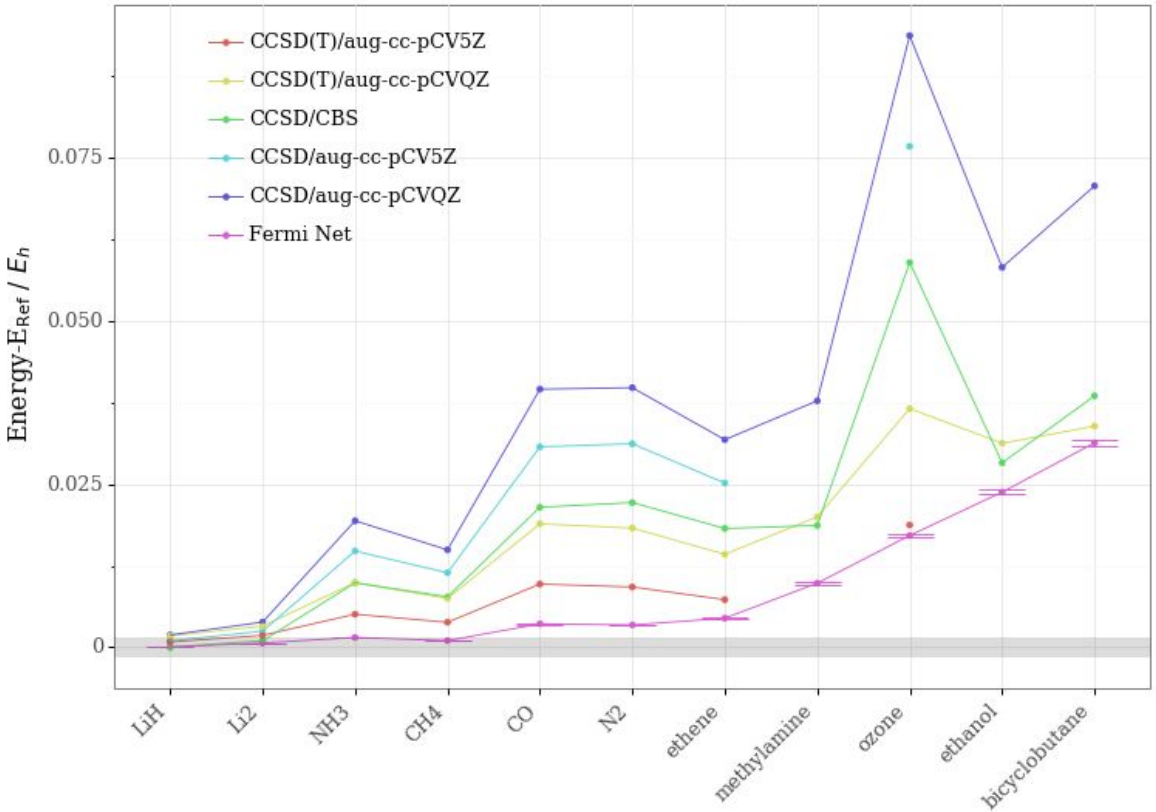
# Molecules



Mean-field (~99.5%)



# Molecules

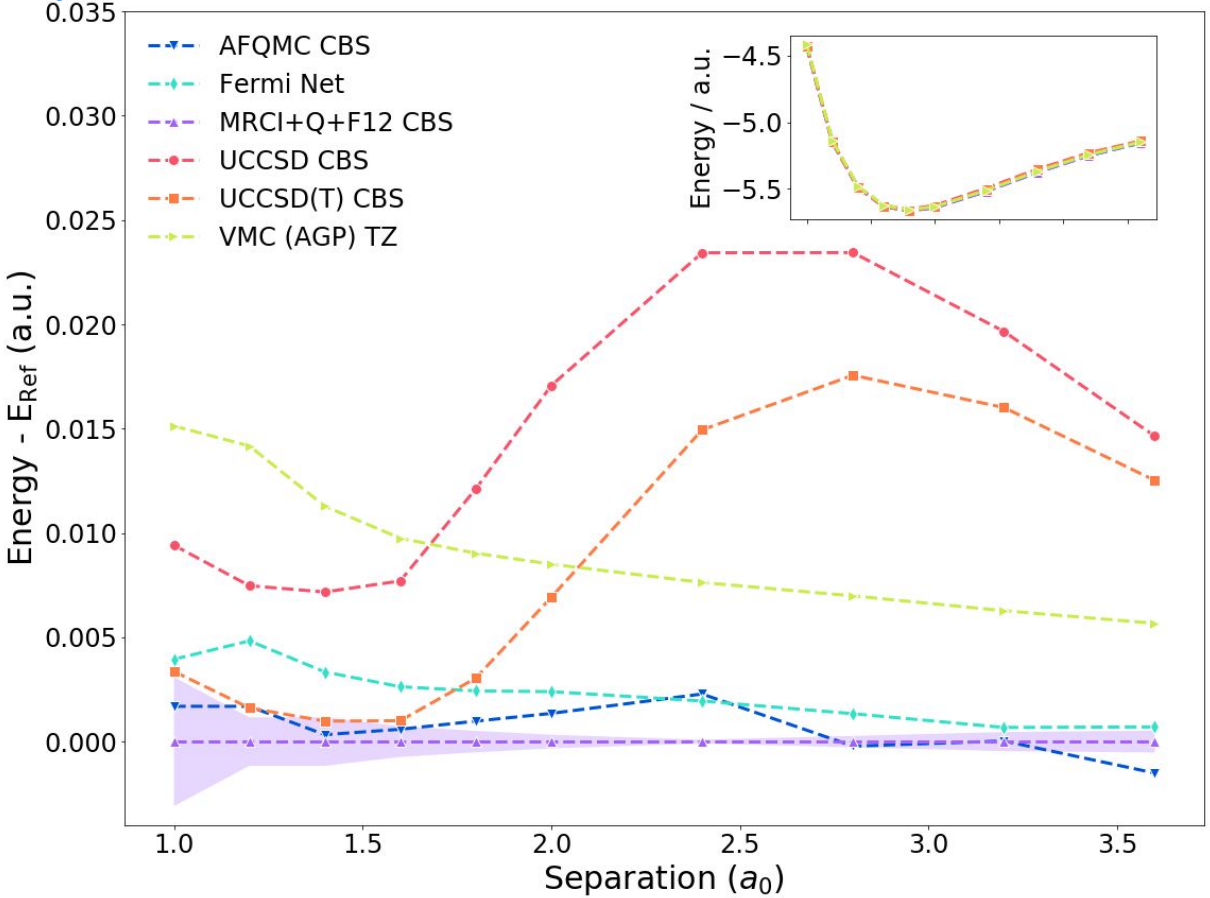



Fermi Net outperforms CCSD(T) in QZ, 5Z basis sets.

Accuracy degradation with # of electrons.

- Fixed network configuration
- Parameter optimization?
- Monte Carlo improvements?

$H_{10}$

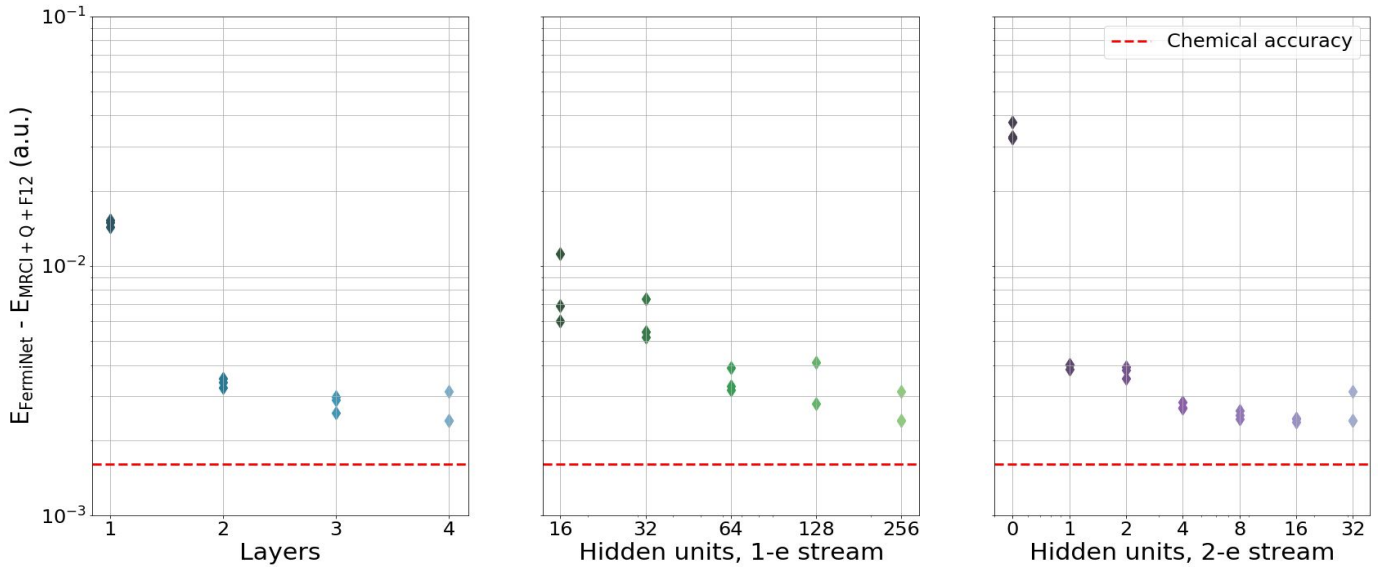



Lower energies than conventional VMC and coupled cluster.

Competitive with AFQMC.

Data:
M. Motta *et al.*, PRX, 031059 (2017).

# Network ablation: $H_{10}$




MRCI+Q+F12: Motta *et al.*, PRX, 031059 (2017).

Suggests that *wider* network may be the key to improving accuracy

# Range of ideas and approaches…



| | Systems | Parameterization | Optimization | Accuracy |
|---|---|---|---|---|
| NNB: Luo, Clark PRL 122 226401 (2019) | Hubbard model | Backflow | first-order | Surpasses conventional backflow |
| PauliNet: Hermann, Schätzle, Noé, arXiv:1909:08423 | Molecules, continuum | Jastrow and backflow | ADAM | Boron: 97.3% c.e.<br>$H_{10}$: 90-98% c.e. |
| NQS: Choo, Mezzacapo, Carleo, Nat Commun. 11 2360 (2020) | Molecules, second quantization | Map to spin problem + restricted Boltzmann machine | Stochastic reconfiguration | <1mH error compared to FCI in STO-3G. |
| Fermi Net | Molecules, continuum | Everything | KFAC | Boron: 99.8% c.e.<br>$H_{10}$: 98.5-99.3% c.e. |

c.e. - correlation energy
M. Ruggeri, S. Moroni, M. Holzmann, PRL 120, 205302 (2018): Iterative neural network backflow for $^3$He.
J. Han, L. Zhang and W. E, JCP 399, 108929 (2019): Simpler architecture, worse than mean-field results

# Conclusions

## Fermionic Neural Network

- Structured neural network for quantum wavefunctions
- Powerful scalable second-order optimization
- Optimization and sampling interact in counterintuitive ways
- Accurate ground state energies with only physically motivated inputs
- Neural networks provide flexible and compact representations of complex high-dimensional functions

DeepMind

# Thank you

James Spencer

Alex Matthews

Matthew Foulkes

{pfau, jamessspencer}@google.com
@pfau, @exp_n

https://arxiv.org/abs/1909.02487

Imperial College London