2024-12-04

Neural Wavefunction

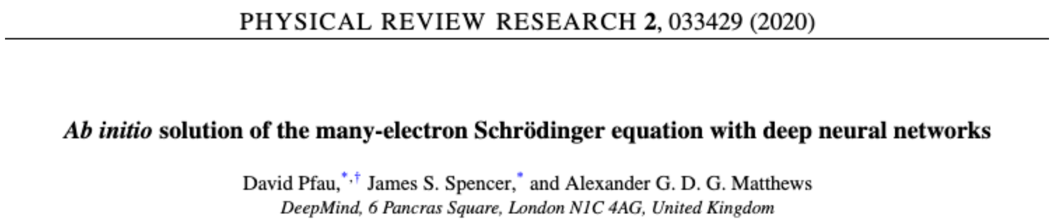
Variational Monte Carlo (VMC) Method
(with quantum mechanical research topics for ML)

Seongsu Kim

Pohang University of Science and Technology (POSTECH)

Neural Wavefunction

PRR ‘20

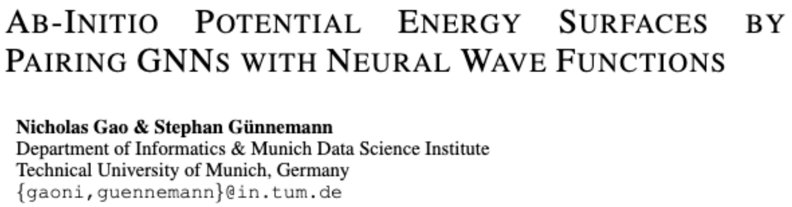


PHYSICAL REVIEW RESEARCH **2**, 033429 (2020)

Ab initio solution of the many-electron Schrödinger equation with deep neural networks

David Pfau,[*,†] James S. Spencer,[*] and Alexander G. D. G. Matthews
DeepMind, 6 Pancras Square, London N1C 4AG, United Kingdom

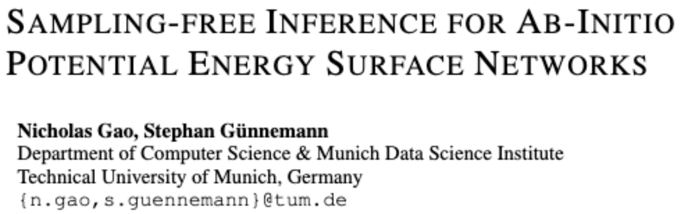
ICLR ‘22



Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions

Nicholas Gao & Stephan Günnemann
Department of Informatics & Munich Data Science Institute
Technical University of Munich, Germany
{gaoni,guennemann}@in.tum.de

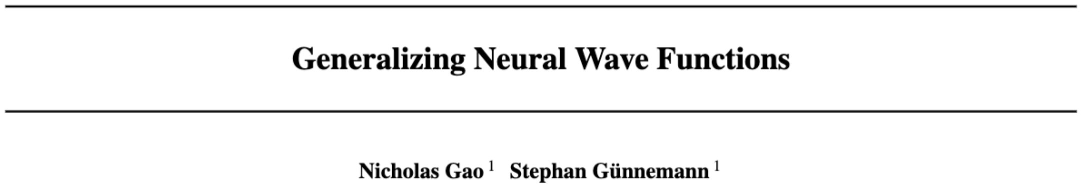
ICLR ‘23



Sampling-free Inference for Ab-Initio Potential Energy Surface Networks

Nicholas Gao, Stephan Günnemann
Department of Computer Science & Munich Data Science Institute
Technical University of Munich, Germany
{n.gao,s.guennemann}@tum.de

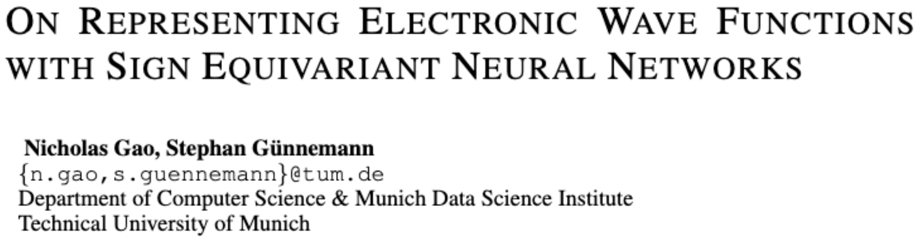
ICML ‘23



Generalizing Neural Wave Functions

Nicholas Gao [1] Stephan Günnemann [1]

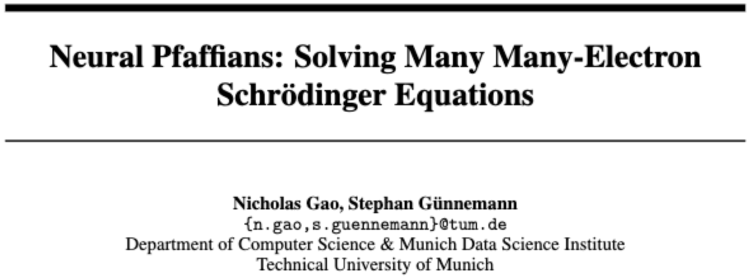
ICLR ’24 Workshop



On Representing Electronic Wave Functions with Sign Equivariant Neural Networks

Nicholas Gao, Stephan Günnemann
{n.gao,s.guennemann}@tum.de
Department of Computer Science & Munich Data Science Institute
Technical University of Munich

NIPS ’24 (ORAL)



Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations

Nicholas Gao, Stephan Günnemann
{n.gao,s.guennemann}@tum.de
Department of Computer Science & Munich Data Science Institute
Technical University of Munich

Neural Wavefunction

- ***Neural wavefunction*** is one of the task of *AI for Science*

- ***Wavefunction*** is the solution of the *Schrödinger equation*

- Finding the wavefunction of the without the ground truth (**unsupervised**)

- Achieving the *quantum precision* **chemical property** of molecule

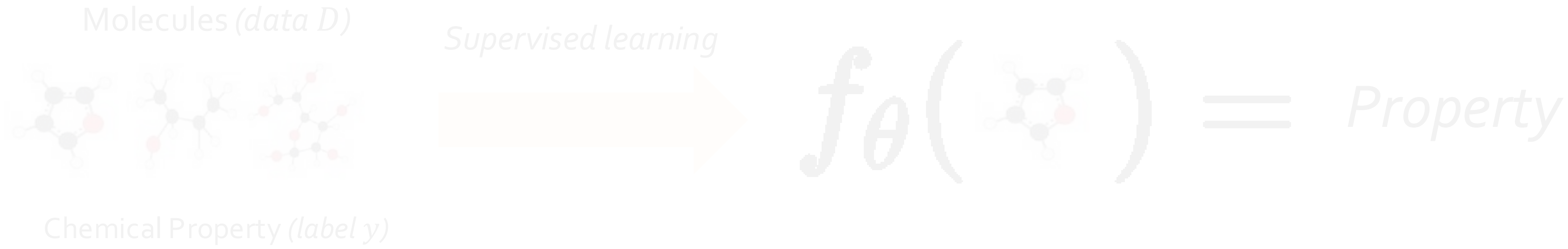
Quantum precision: Similar with quantum mechanical method (ex. DFT) - about $1kcal/mol$ (on energy)

Why should we know wavefunction?

- To calculate the quantum **chemical property**
  - Electrostatics potentials, Electro density, HOMO, LUMO, ΔEgap and etc.

Ex. SchNet, EGNN

- *Classical GNN* method learned these from the **supervised learning**
  - Cannot make "*quantum level accuracy*" due to regression error
  - Surrogate models are subject to the training domain
    - Re-train the model for each chemical property

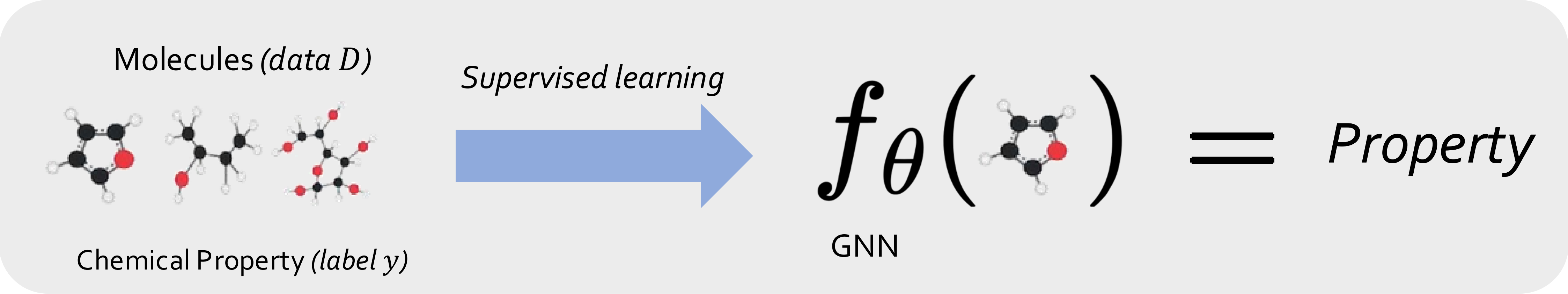


Why should we know wavefunction?

- To calculate the quantum **chemical property**
  - Electrostatics potentials, Electro density, HOMO, LUMO, ΔEgap and etc.

Ex. SchNet, EGNN

- *Classical GNN* approach learned these from the **supervised learning**
  - Cannot make "*quantum level accuracy*" due to regression error
  - Surrogate models are subject to the training domain
    - Re-train the model for each chemical property



Molecules *(data D)*

Chemical Property *(label y)*

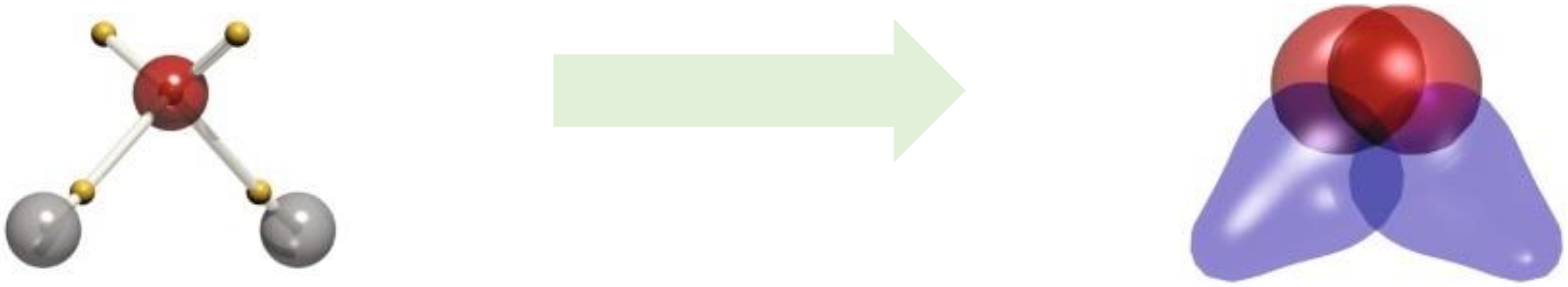
Supervised learning

$f_\theta(\;) = \textit{Property}$

GNN

Neural Wavefunction

- Neural wavefunction tries to model the wavefunction directly
  - Not using the supervised learning, but **variational Monte Carlo (VMC)** framework
  - Direct optimize network to model ground truth wavefunction



Molecule Configuration
(Pos X , Atomic num Z)

VMC + Gradient descent

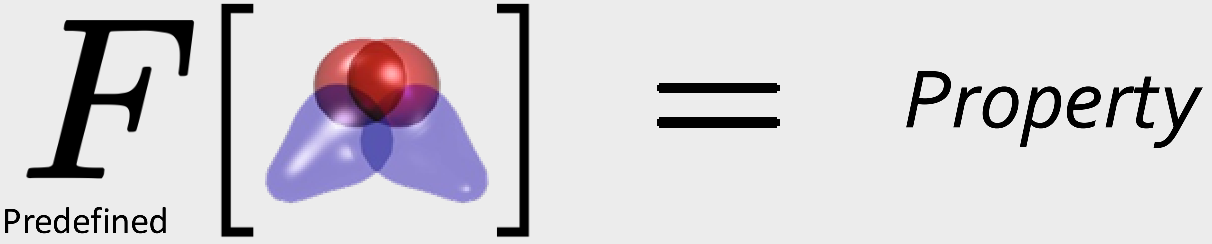
Wavefunction of molecule

No label
(No property given)

Wavefunction

- Wavefunction can be used to **measure the chemical property** in principle

Functional of wavefunction



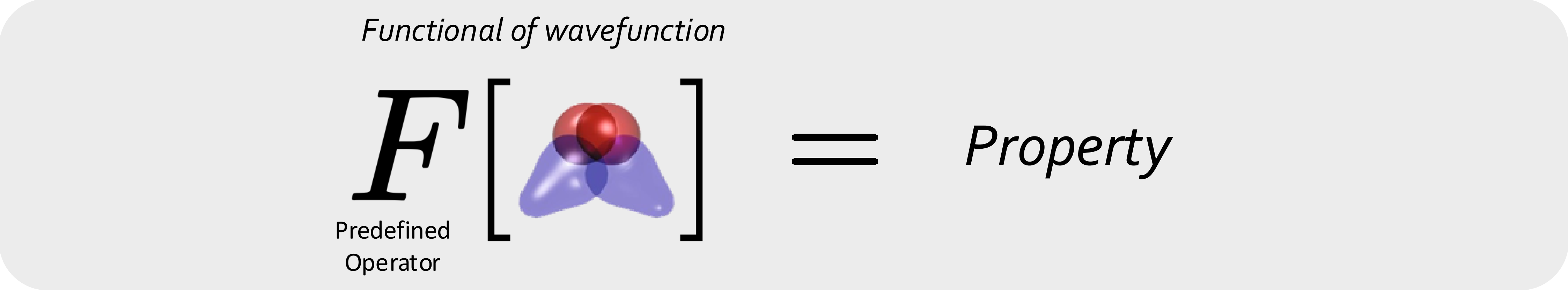
Predefined Operator

- **Quantum property** calculated in this way:
  - First, calculate wavefunction from the configuration
  - Second, calculate property from functional of wavefunction

- These methodology is called ***ab initio***
  - Do not use experimental data or observation, derivate from the principles and laws

Wavefunction

- Wavefunction can be used to **measure the chemical property** in principle



Functional of wavefunction

$F[\ \ ] = $ *Property*

Predefined Operator

- **Quantum property** calculated in this way:
  - First, calculate wavefunction from the configuration
  - Second, calculate property from functional of wavefunction

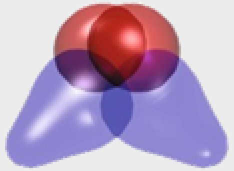
- These methodology is called ***ab initio***
  - Do not use experimental data or observation, derivate from the principles and laws

Wavefunction

- Wavefunction can be used to **measure the chemical property** in principle

Functional of wavefunction

$$F\left[\quad\right] = \textit{Property}$$



Predefined Operator

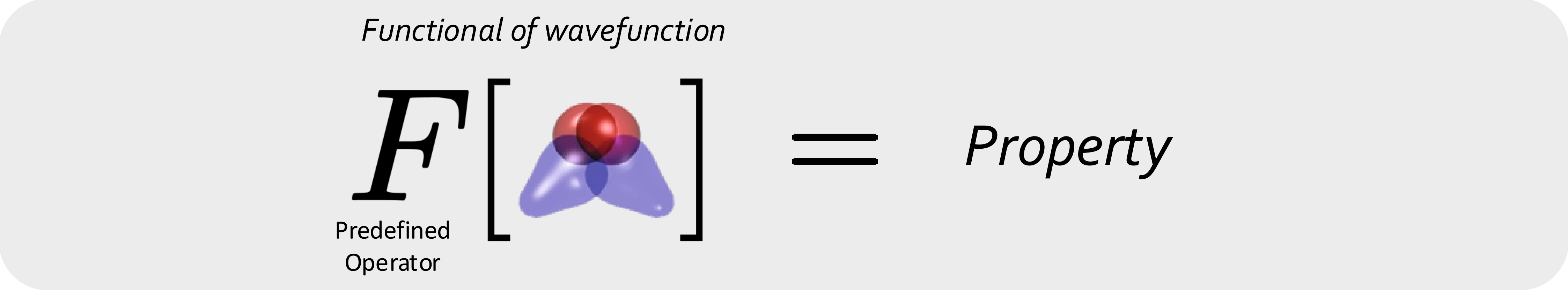
- **Quantum property** calculated in this way:
  - First, calculate wavefunction from the configuration
  - Second, calculate property from functional of wavefunction

- These methodology is called *ab initio*
  - Do not use experimental data or observation, derivate from the principles and laws

Neural wavefunction deal with this step

Wavefunction

- Wavefunction can be used to **measure the chemical property** in principle



Functional of wavefunction

$$F[\quad] = \textit{Property}$$

Predefined Operator

- **Quantum property** calculated in this way:
  - First, calculate wavefunction from the configuration
  - Second, calculate property from functional of wavefunction

- These methodology is called ***ab initio*** *(derived by first principle)*
  - Do not use ***experimental*** data or observation, derivate from the ***principles*** and ***laws***

Wavefunction

- Solving Many-Electron Stationary Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

E : Energy of the system

- It is one of the *eigenvalue and eigenstate problem*

- Objective is finding the ***ground-state*** eigenstate ψ_0

Ground-state: *State with the smallest eigenvalue*

Wavefunction

- Solving Many-Electron Stationary Schrödinger Equation

$$\boldsymbol{H}\psi = \boldsymbol{E}\psi$$

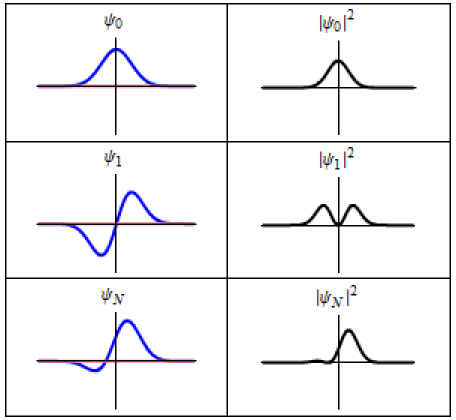
$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

E : Energy of the system

- Glossary

Stationary vs. Non-stationary



Wavefunction

- Solving Many-Electron Stationary Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

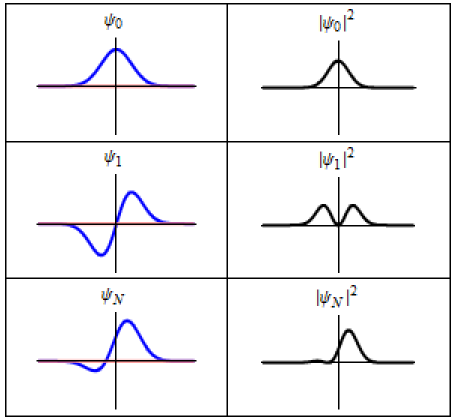
$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

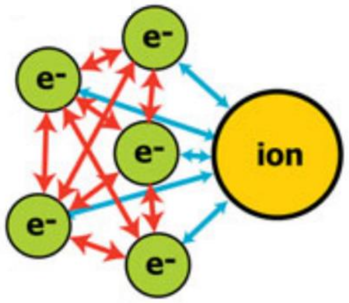
E : Energy of the system

- Glossary

Stationary vs. Non-stationary

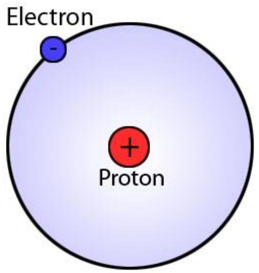


Many-electron



Most cases
(Manu atoms)

Single Electron



Restricted cases
(Hydrogen H)

Wavefunction

- Solving Many-Electron Stationary Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

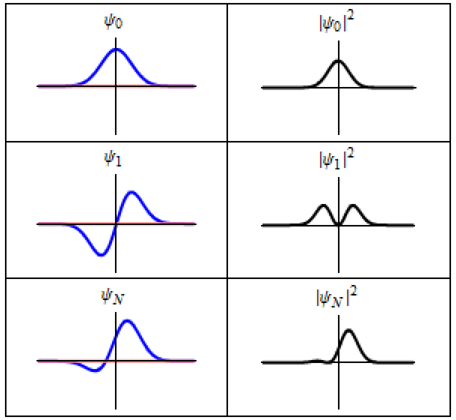
$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

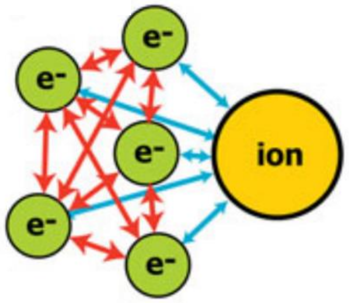
E : Energy of the system

- Glossary

Stationary vs. Non-stationary



Many-electron



Most cases

Impossible to solve analytically

Wavefunction

- Solving Many-Electron **Stationary** Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

E : Energy of the system

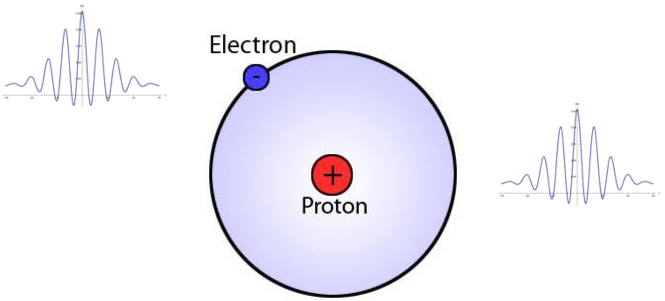
- To solve this intractable problem, need some ***approximation***

Wavefunction

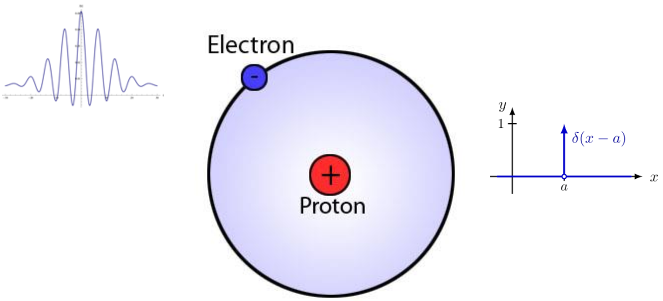
- **Born-Oppenheimer (BO) approximation**
  1. Fixing the nuclei position (= nuclei is deterministic and has no kinetic energy)
  2. Assuming ***electronic*** wave function is independent each other

Ex. $\psi(r_0, r_1) = \psi_0(r_0)\psi_1(r_1)$

W/O BO



With BO



Wavefunction

- Solving Many-Electron Stationary Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

E : Energy of the system

- With BO approximation, Hamiltonian is given by

$$\boldsymbol{H} = -\frac{1}{2}\sum_{i=1}^{N_\mathrm{e}}\sum_{k=1}^{3}\frac{\partial^2}{\partial \vec{r}_{ik}^{\,2}} + \sum_{j>i}^{N_\mathrm{e}}\frac{1}{\|\vec{r}_i - \vec{r}_j\|} - \sum_{i=1}^{N_\mathrm{e}}\sum_{m=1}^{N_\mathrm{n}}\frac{Z_m}{\left\|\vec{r}_i - \vec{R}_m\right\|} + \sum_{n>m}^{N_\mathrm{n}}\frac{Z_m Z_n}{\left\|\vec{R}_m - \vec{R}_n\right\|}$$

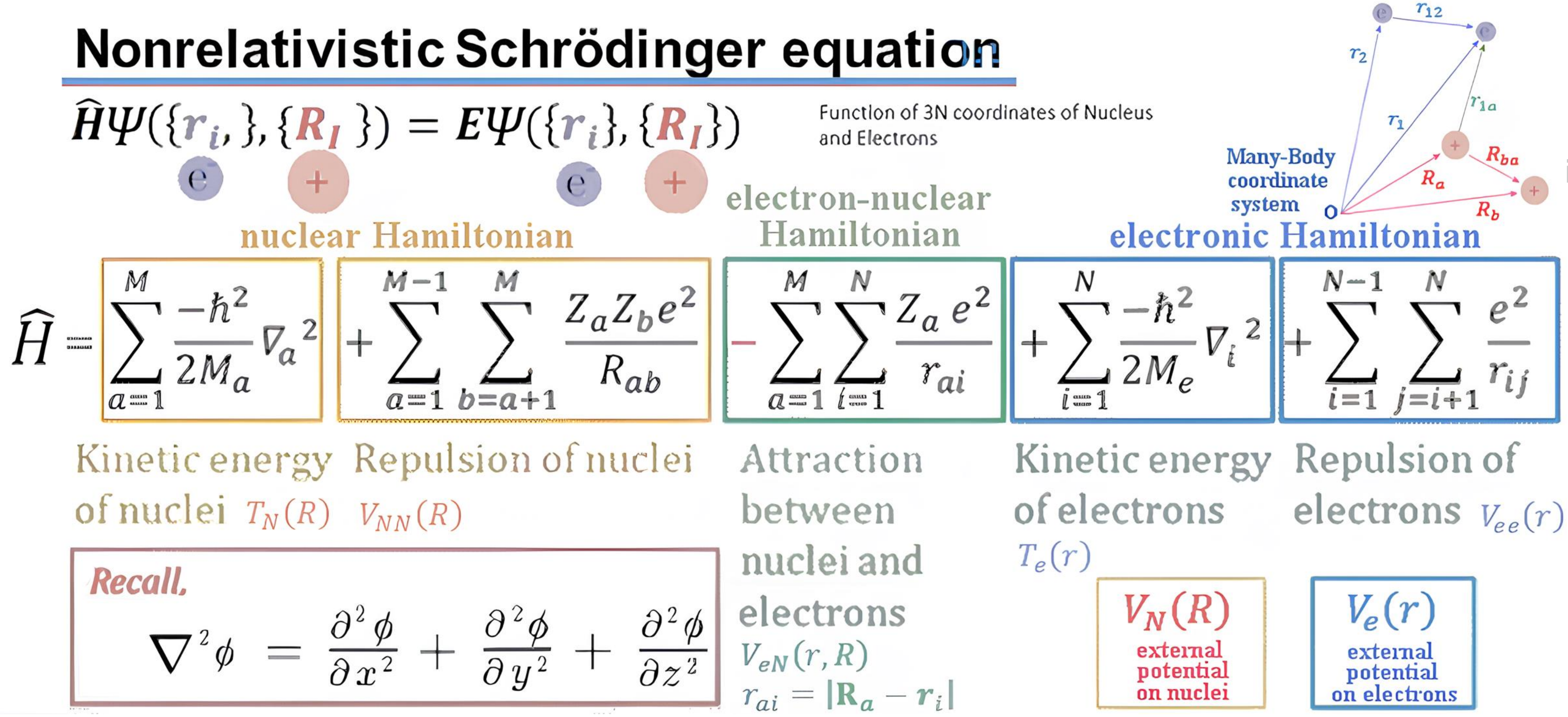
Kinetic energy Elec-Elec pot Elec-Nucl pot Nucl-Nucl pot

Wavefunction

- Sol

- Wit

ion



Nonrelativistic Schrödinger equation

$$\hat{H}\Psi(\{\boldsymbol{r}_i,\},\{\boldsymbol{R}_I\}) = E\Psi(\{\boldsymbol{r}_i\},\{\boldsymbol{R}_I\})$$

Function of 3N coordinates of Nucleus and Electrons

e + e +

nuclear Hamiltonian

electron-nuclear Hamiltonian

electronic Hamiltonian

$$\hat{H} = \boxed{\sum_{a=1}^{M}\frac{-\hbar^2}{2M_a}\nabla_a{}^2} \boxed{+\sum_{a=1}^{M-1}\sum_{b=a+1}^{M}\frac{Z_aZ_be^2}{R_{ab}}} \boxed{-\sum_{a=1}^{M}\sum_{i=1}^{N}\frac{Z_a\,e^2}{r_{ai}}} \boxed{+\sum_{i=1}^{N}\frac{-\hbar^2}{2M_e}\nabla_i{}^2} \boxed{+\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{e^2}{r_{ij}}}$$

Kinetic energy of nuclei $T_N(R)$

Repulsion of nuclei $V_{NN}(R)$

Attraction between nuclei and electrons $V_{eN}(r,R)$

$r_{ai} = |\mathbf{R}_a - \boldsymbol{r}_i|$

Kinetic energy of electrons $T_e(r)$

Repulsion of electrons $V_{ee}(r)$

$V_N(R)$ external potential on nuclei

$V_e(r)$ external potential on electrons

Recall,

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2}$$

Neural
Wavefunction

- Solving Many-Electron **Stationary** Schrödinger Equation

$$\boldsymbol{H}\psi = E\psi$$

$\boldsymbol{H}$: Hamiltonian (operator)

ψ : (Electronic) wave function

E : Energy of the system

- With **two constraints**,
  - **Fermi-Dirac statistics (*Antisymmetric*)**

Two atoms: $\psi(\boldsymbol{r}_1, \boldsymbol{r}_2) = -\psi(\boldsymbol{r}_1, \boldsymbol{r}_2)$, General: $\psi(\boldsymbol{r}) = -\operatorname{sign}(\pi)\psi(\pi(\boldsymbol{r}))$

  - **Squared integral normalized**

$$\int \psi^2(\boldsymbol{r})d\boldsymbol{r} = 1$$

Problem objective

- Given Hamiltonian $\boldsymbol{H}$, finding the lowest energy E_0 and state ψ_0
- To solve it, we **parametrize** the wave function ψ_θ
- Make the parametrize the wave function to be ground state ψ_0

$$\boldsymbol{H}\psi_\theta \approx \boldsymbol{H}\psi_0$$

- Then the eigenvalue of the ψ_θ becomes the E_0

How can we do it in the unsupervised way?

Problem objective

- Given Hamiltonian $\boldsymbol{H}$, finding the lowest energy E_0 and state ψ_0

- To solve it, we **parametrize** the wave function ψ_θ

- Make the parametrize the wave function to be ground state ψ_0

$$\boldsymbol{H}\psi_\theta \approx \boldsymbol{H}\psi_0$$

- Then the eigenvalue of the ψ_θ becomes the E_0

How can we do it in the unsupervised way?

Problem objective

- Given Hamiltonian $\boldsymbol{H}$, finding the lowest energy E_0 and state ψ_0

- To solve it, we **parametrize** the wave function ψ_θ

- Make the parametrize the wave function to be ground state ψ_0

$$\boldsymbol{H}\psi_\theta \approx \boldsymbol{H}\psi_0$$

- Then the eigenvalue of the ψ_θ becomes the E_0

How can we do it in the unsupervised way?

Variational Monte Carlo (VMC)

SE: $\boldsymbol{H}\psi = E\psi$

- ***Ground state energy*** can be calculated by the *inner product* over *SE:*

$$\frac{\langle\psi_0|\boldsymbol{H}\,|\psi_0\rangle}{\langle\psi_0\mid\psi_0\rangle} = \frac{\langle\psi_0|E_0\,|\psi_0\rangle}{\langle\psi_0\mid\psi_0\rangle} = E_0$$

- It is the same with below expression:

$$E = \frac{\int \psi(\boldsymbol{r})\boldsymbol{H}\psi(\boldsymbol{r})d\boldsymbol{r}}{\int \psi(\boldsymbol{r})^2 d\boldsymbol{r}}$$

Variational Monte Carlo (VMC)

SE: $\boldsymbol{H}\psi = E\psi$

- ***Ground state energy*** can be calculated by the *inner product* over *SE:*

$$\frac{\langle\psi_0|\boldsymbol{H}\,|\psi_0\rangle}{\langle\psi_0\mid\psi_0\rangle} = \frac{\langle\psi_0|E_0\,|\psi_0\rangle}{\langle\psi_0\mid\psi_0\rangle} = E_0$$

- Q: Is it valid for any wave-function (ex. ψ_θ)?
  - A: Actually, it is hard to guarantee since ψ_θ may not be the eigen state of $\boldsymbol{H}$

$$\boldsymbol{H}\psi_\theta \neq E\psi_\theta$$

Variational Monte Carlo (VMC)

SE: $\boldsymbol{H}\psi = E\psi$

- Assuming ***trial energy*** E_θ is defined by the

$$E_\theta = \frac{\int \psi_\theta(\boldsymbol{r}) \boldsymbol{H} \psi_\theta(\boldsymbol{r}) d\boldsymbol{r}}{\int \psi_\theta(\boldsymbol{r})^2 d\boldsymbol{r}}$$

- This is solved by the ***variational principle***
  - States that the trial energy of any trial wavefunction ψ_θ upper bounds E_0

$$E_0 \leq E_\theta = \frac{\int \psi_\theta(\mathbf{r}) \boldsymbol{H} \psi_\theta(\mathbf{r}) \mathrm{d}\mathbf{r}}{\int \psi_\theta(\mathbf{r})^2 \mathrm{d}\mathbf{r}}$$

Variational Monte Carlo (VMC)

- Simply calculate the gradient of trial energy E_θ is enough to find E_0

- To find $\nabla_\theta E_\theta$, define ***local energy*** $E_L(\boldsymbol{r})$ and ***probability density*** $p(\boldsymbol{r})$

$$E_L(\boldsymbol{r}) = \frac{\boldsymbol{H}\psi(\boldsymbol{r})}{\psi(\boldsymbol{r})} \qquad p(\boldsymbol{r}) = \frac{\psi^2(\boldsymbol{r})}{\int \psi^2(\boldsymbol{r})d\boldsymbol{r}}$$

- Then, we have

$$E_\theta = \int p(\boldsymbol{r})E_L(\boldsymbol{r})d\boldsymbol{r} = \mathbb{E}_{p(\boldsymbol{r})}[E_L(\boldsymbol{r})]$$

$$\nabla_\theta E_\theta = \mathbb{E}_{p(\boldsymbol{r})}\left[\left(E_L(\boldsymbol{r}) - \mathbb{E}_{p(\boldsymbol{r})}[E_L(\boldsymbol{r})]\right)\nabla_\theta \log \psi_\theta(\boldsymbol{r})\right]$$

Variational Monte Carlo (VMC)

- This is called Monte Carlo

$$\boldsymbol{E}_\theta = \int \underline{p(\boldsymbol{r})}\boldsymbol{E}_L(\boldsymbol{r})d\boldsymbol{r} = \mathbb{E}_{\underline{p(\boldsymbol{r})}}[\boldsymbol{E}_L(\boldsymbol{r})]$$

$$\nabla_\theta \boldsymbol{E}_\theta = \mathbb{E}_{\underline{p(\boldsymbol{r})}}\left[\left(\boldsymbol{E}_L(\boldsymbol{r}) - \mathbb{E}_{\underline{p(\boldsymbol{r})}}[\boldsymbol{E}_L(\boldsymbol{r})]\right)\nabla_\theta \log \psi_\theta(\boldsymbol{r})\right]$$

, since the gradient calculation needs to sample over $p(\boldsymbol{r})$

- In addition, the true ground energy E_0 and state ψ_0 is hard to get
  - Criterion for good wave function is the lowest energy E_θ over baseline

Variational Monte Carlo (VMC)

- This is called Monte Carlo

$$E_\theta = \int \underline{p(\boldsymbol{r})} E_L(\boldsymbol{r}) d\boldsymbol{r} = \mathbb{E}_{\underline{p(\boldsymbol{r})}}[E_L(\boldsymbol{r})]$$

$$\nabla_\theta E_\theta = \mathbb{E}_{\underline{p(\boldsymbol{r})}}\left[\left(E_L(\boldsymbol{r}) - \mathbb{E}_{\underline{p(\boldsymbol{r})}}[E_L(\boldsymbol{r})]\right)\nabla_\theta \log \psi_\theta(\boldsymbol{r})\right]$$

, since the gradient calculation needs to sample over $p(\boldsymbol{r})$

- In addition, the true ground energy E_0 and state ψ_0 is hard to get
  - Criterion for good wave function is the lowest energy E_θ over baseline

Three types of the Research direction

- **Antisymmetric** architecture/representation
  - Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations ('24 NIPS)
  - On Representing Electronic Wave Functions with Sign Equivariant Neural Networks ('24 ICLRw)
  - Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions ('21 ICLR)

- **Transferrable** architecture / parameterization over molecules
  - Generalizing Neural Wave Functions ('23 ICML)
  - Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks ('22 Nat. Comp)
  - Variational Monte Carlo on a Budget –Fine-tuning pre-trained Neural Wavefunctions ('23NIPS)

- **Tackling the VMC** or sampling process (Optimization?, Training?)
  - Sampling-free Inference for Ab-Initio Potential Energy Surface Networks ('22 ICLR)
  - Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo ('21 Arxived)
  - A Score-Based Model for Learning Neural Wavefunctions ('23 rejected)

Three types of the Research direction

- **Antisymmetric** architecture/representation
  - Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations ('24 NIPS)
  - On Representing Electronic Wave Functions with Sign Equivariant Neural Networks ('24 ICLRw)
  - Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions ('21 ICLR)

- **Transferrable** architecture / parameterization over molecules
  - Generalizing Neural Wave Functions ('23 ICML)
  - Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks ('22 Nat. Comp)
  - Variational Monte Carlo on a Budget –Fine-tuning pre-trained Neural Wavefunctions ('23NIPS)

- **Tackling the VMC** or sampling process (Optimization?, Training?)
  - Sampling-free Inference for Ab-Initio Potential Energy Surface Networks ('22 ICLR)
  - Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo ('21 Arxived)
  - A Score-Based Model for Learning Neural Wavefunctions ('23 rejected)

Three types of the Research direction

- **Antisymmetric** architecture/representation
  - Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations ('24 NIPS)
  - On Representing Electronic Wave Functions with Sign Equivariant Neural Networks ('24 ICLRw)
  - Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions ('21 ICLR)

- **Transferrable** architecture / parameterization over molecules
  - Generalizing Neural Wave Functions ('23 ICML)
  - Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks ('22 Nat. Comp)
  - Variational Monte Carlo on a Budget –Fine-tuning pre-trained Neural Wavefunctions ('23NIPS)

- **Tackling the VMC** or sampling process (Optimization?, Training?)
  - Sampling-free Inference for Ab-Initio Potential Energy Surface Networks ('22 ICLR)
  - Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo ('21 Arxived)
  - A Score-Based Model for Learning Neural Wavefunctions ('23 rejected)

FermiNet ('22 PRR)

- One of the baseline from the physics community + (DeepMind)
  - Didn't use the GNN

- Using Slater determinant for **antisymmetric Ansatz**

Ansatz: guess or assumption of a mathematical expression or function of physical model

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

Slater determinant

- Antisymmetry is the one of the property of the determinant operation

$$\begin{aligned}\Psi\left(\mathbf{x}_1, \mathbf{x}_2\right) &= \frac{1}{\sqrt{2}}\left\{\psi_1\left(\mathbf{x}_1\right)\psi_2\left(\mathbf{x}_2\right)-\psi_1\left(\mathbf{x}_2\right)\psi_2\left(\mathbf{x}_1\right)\right\} \\ &= \frac{1}{\sqrt{2}}\begin{vmatrix}\psi_1\left(\mathbf{x}_1\right) & \psi_2\left(\mathbf{x}_1\right) \\ \psi_1\left(\mathbf{x}_2\right) & \psi_2\left(\mathbf{x}_2\right)\end{vmatrix}\end{aligned}$$

- Swapping, x_1 and x_2 implying row change
- Naturally, Ψ is antisymmetric

$$\boldsymbol{\Psi}(\mathbf{x}_1, \mathbf{x}_2) = -\boldsymbol{\Psi}(\mathbf{x}_2, \mathbf{x}_1)$$

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

Slater determinant

Orbital: single paricle (or electron) wave function

- Slater determinant based on **Hatree-Fock** (or mean-field) **approximation**
  - **(Hatree Product)** a multi-electron wave function as a simple product of one-electron wave functions (orbitals).

Ex.$\Psi(x_1, x_2) = \chi_1(x_1)\chi_2(x_2)$

- For each slater determinant is

Electron position index x_i

$$\det[\Phi^k] = \begin{vmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_1(\mathbf{x}_n) \\ \vdots & & \vdots \\ \phi_n(\mathbf{x}_1) & \dots & \phi_n(\mathbf{x}_n) \end{vmatrix}$$

$\phi_i \in k$

Orbital position index ϕ_i

$(k \in K)$,

- K is the possible subset of the orbitals having n numbers of the electrons

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

Slater determinant

$$\det[\mathbf{\Phi}^k] = \begin{vmatrix} \phi_1(\mathbf{x}_1) & \ldots & \phi_1(\mathbf{x}_n) \\ \vdots & & \vdots \\ \phi_n(\mathbf{x}_1) & \ldots & \phi_n(\mathbf{x}_n) \end{vmatrix}$$

- ***Antisymmetrized ansatz*** of the wave function is weighted sum of the possible slater determinant:

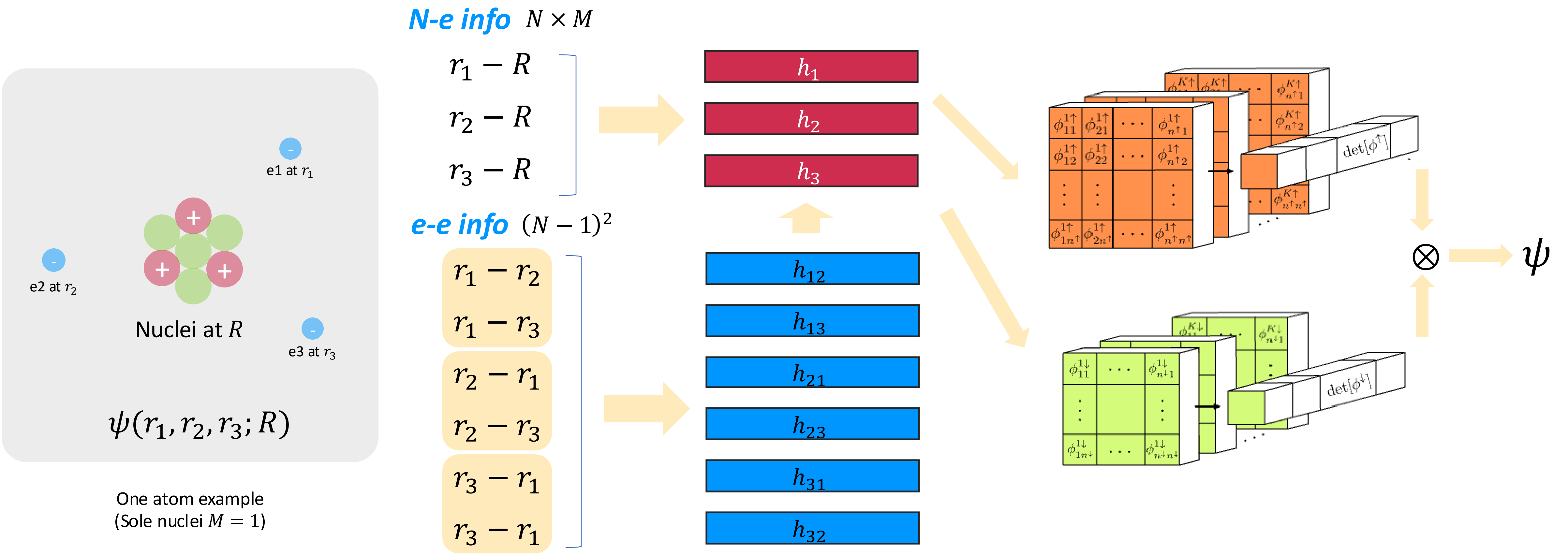
$$\psi(\mathbf{x}_1, \ldots \mathbf{x}_n) = \sum_k \omega_k \det[\mathbf{\Phi}^k]$$

- Limitation: exponential complexity along the system size n

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

FermiNet ('22 PRR)

- Encoding all possible N-e & e-e interactions and make orbital matrix
  - Make slater determinant from orbital matrix



One atom example
(Sole nuclei $M = 1$)

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

FermiNet ('22 PRR)

- Using **slater determinant** to anti-symmetry

- Define arbitrary orbital that decay to 0 infinitely far away from any nuclei

$$\phi_{ij}^k = \left(\boldsymbol{w}_i \boldsymbol{h}_j + b_i^k\right) \sum_{m}^{M} \pi_{im}^k \exp\left(-\sigma_{im}^k \left\|\vec{\boldsymbol{r}}_j - \vec{\boldsymbol{R}}_m\right\|\right)$$

$$\pi_{im}^k = \mathrm{Sigmoid}\left(p_{im}^k\right), \quad \sigma_{im}^k = \mathrm{Softplus}\left(s_{im}^k\right)$$

M, m	Nucleus num/idx
K, k	Orbital num/idx
p, s	Free parameter
i, j	Electron idx

- Q) Why not using spherical harmonics ?
  - Using spherical harmonics as orbital makes results trivial

Ex. $\psi(\vec{r}R) = \psi(\vec{r}) = c$

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

FermiNet ('22 PRR)

- Using **slater determinant** to anti-symmetry

- Define arbitrary orbital that decay to o infinitely far away from any nuclei

$$\phi_{ij}^k = \left(\boldsymbol{w}_i \boldsymbol{h}_j + b_i^k\right) \sum_m^M \pi_{im}^k \exp\left(-\sigma_{im}^k \left\|\vec{\boldsymbol{r}}_j - \vec{\boldsymbol{R}}_m\right\|\right)$$

$$\pi_{im}^k = \mathrm{Sigmoid}\left(p_{im}^k\right), \quad \sigma_{im}^k = \mathrm{Softplus}\left(s_{im}^k\right)$$

M, m	Nucleus num/idx
K, k	Orbital num/idx
p, s	Free parameter
i, j	Electron idx

- Q) Why not using spherical harmonics ?
  - Using spherical harmonics as orbital makes results trivial

Ex. $\psi(\vec{\boldsymbol{r}}R) = \psi(\vec{\boldsymbol{r}}) = c$

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

FermiNet ('22 PRR)

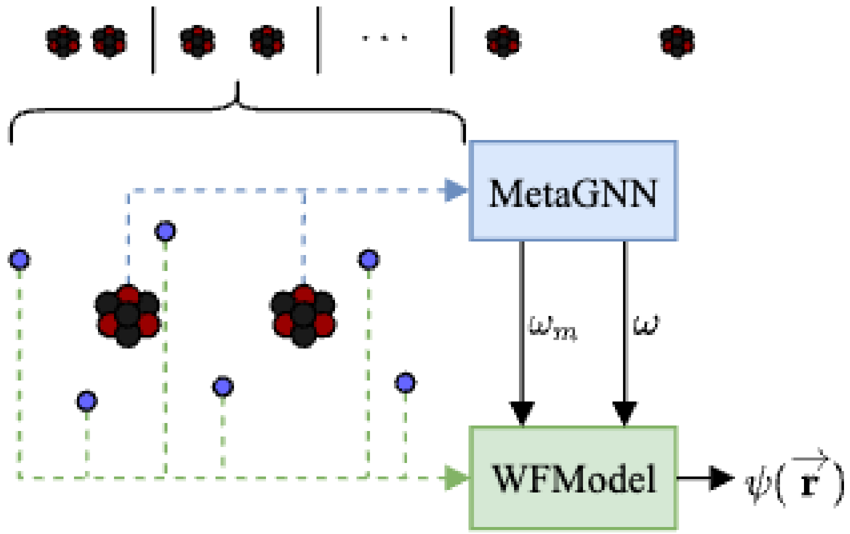
- Limitation
  - Computational demands due to all electron interaction encoding
  - Network can be trained and optimized on a single system. (single atom configuration)

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

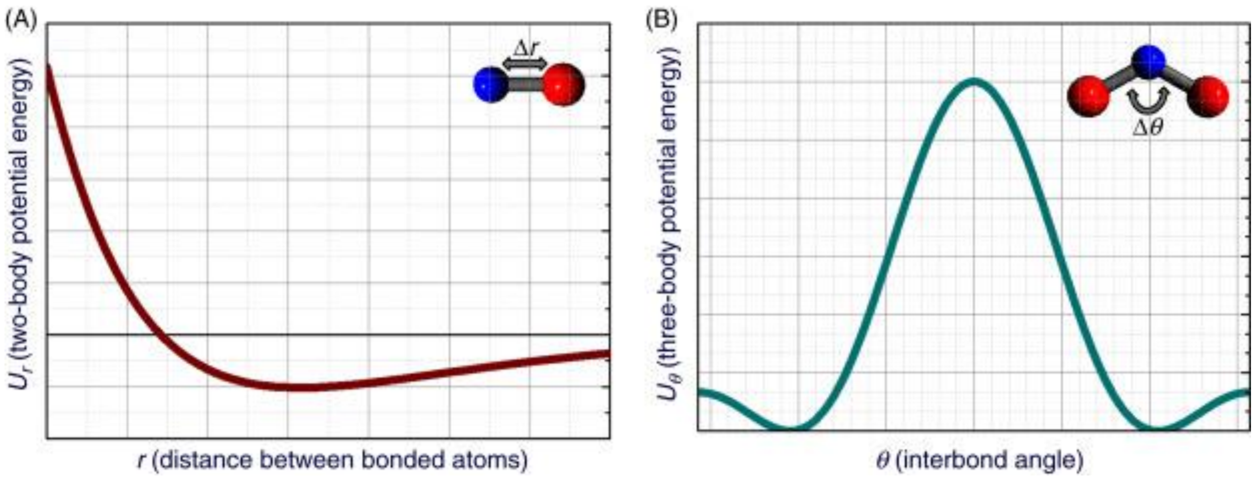
PESNet (‘22 ICLR)

Geometry: Same chemical form, different position

- To handle the multiple geometry, using Meta learning
- *Potential energy surface* (PES) is



PES architecture



Example of the PES

- Using **slater determinant** to anti-symmetry / Using GNN

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

PESNet (‘22 ICLR)

- Using **slater determinant** to anti-symmetry

- Define arbitrary orbital that decay to o infinitely far away from any nuclei

$$\phi_{ij}^k = (\boldsymbol{w}_i \boldsymbol{h}_j + b_i^k) \sum_m^M \pi_{im}^k \exp\left(-\sigma_{im}^k \left\|\vec{\boldsymbol{r}}_j - \vec{\boldsymbol{R}}_m\right\|\right)$$

$$\pi_{im}^k = \mathrm{Sigmoid}\left(p_{im}^k\right), \quad \sigma_{im}^k = \mathrm{Softplus}\left(s_{im}^k\right)$$

M, m	Nucleus num/idx
K, k	Orbital num/idx
p, s	Free parameter
i, j	Electron idx

- If we use GNN, why not use spherical harmonics ?
  - Ans) Using spherical harmonics as orbital makes **results trivial**

Ex. $\psi(\vec{\boldsymbol{r}}R) = \psi(\vec{\boldsymbol{r}}) = c$

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

PESNet (‘22 ICLR)

- Limitation
  - Relaxed to the PES, but **not transferable to another molecule**

Gao, N., & Günnemann, S. (2021). Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. *2022 ICLR*. https://doi.org/10.48550/arxiv.2110.05064

(Globe/Moon)('23 ICML)

- Motivation
  - Generalization across **different molecules** & different geometries
- Idea
  - Still using Meta-learning Framework, but try to **estimate more parameters**
  - Define the interactions more rigorously and try to expand functional space
    - Ex. Not just considering E-E, but N-E, N-O each other
  - Define **size-consistency** and propose training for size-consistency
    - Ex. The energy of the molecular is very differed by the size of

Gao, N., & Günnemann, S. (2023). Generalizing Neural Wave Functions. 23*ICML*. https://doi.org/10.48550/arxiv.2302.04168

NeurPf ('24 NIPS)

- More expressive way to build **antisymmetric function**
- The ***Pfaffian*** operator is defined on skew-symmetric $2n \times 2n$ matrix

$$\mathrm{Pf}(A) = \frac{1}{2^n n!} \sum_{\tau \in S_{2n}} \mathrm{sgn}(\tau) \prod_{i=1}^{n} A_{\tau(2i-1),\tau(2i)}$$

where S_{2n} is the symmetric group of $2n$ element

- Example:

$$\mathrm{pf}\begin{bmatrix} 0 & a & b & c \\ -a & 0 & d & e \\ -b & -d & 0 & f \\ -c & -e & -f & 0 \end{bmatrix} = af - be + dc.$$

Gao, N., & Günnemann, S. (2024). Neural Pfaffians: Solving Many Many-Electron Schr\"odinger Equations. *24NIPS*. https://doi.org/10.48550/arxiv.2405.14762

NeurPf ('24 NIPS)

- The *Pffafian* has two property

$$\mathrm{Pf}(A)^2 = \det(A)^2$$

and

$$\mathrm{Pf}(BAB^\top) = \det(B)\,\mathrm{Pf}(A)$$

B is invertible

- For permutation equivariant function $A(\tau(\boldsymbol{r})) = P_\tau A(\boldsymbol{r}) P_\tau^\top$

$$\begin{aligned}\Psi(\tau(\mathbf{r})) = \mathrm{Pf}(A(\tau(\mathbf{r}))) &= \mathrm{Pf}\left(P_\tau A(\mathbf{r}) P_\tau^T\right) \\ &= \det\left(P_\tau\right)\mathrm{Pf}(A(\mathbf{r})) \\ &= \mathrm{sign}(\tau)\Psi(\mathbf{r})\end{aligned}$$

Gao, N., & Günnemann, S. (2024). Neural Pfaffians: Solving Many Many-Electron Schr\"odinger Equations. *24NIPS*. https://doi.org/10.48550/arxiv.2405.14762

NeurPf ('24 NIPS)

- Construct wave function with Pffafian

$$\Psi_{\text{Pfaffian}}(\mathbf{r}) = \frac{1}{\text{Pf}\left(A_{\text{Pf}}\right)}\text{Pf}\left(\hat{\Phi}_{\text{Pf}}(\mathbf{r})A_{\text{Pf}}\hat{\Phi}_{\text{Pf}}(\mathbf{r})^{T}\right)$$

- It covers more functions than Slater wave functions

$$\Psi_{\text{Pfaffian}}\left(\overrightarrow{\mathbf{r}}\right) = \sum_{k=1}^{N_{\text{k}}} c_k \Psi_{\text{Pfaffian}\,,k}(\overrightarrow{\mathbf{r}}).$$

Gao, N., & Günnemann, S. (2024). Neural Pfaffians: Solving Many Many-Electron Schr\"odinger Equations. *24NIPS*. https://doi.org/10.48550/arxiv.2405.14762

A Score-Based Model for Neural Wavefunctions (23' rejected)

- Motivation:
  - Can we just model the Wavefunction and energy by the score function?

$$E_L(\boldsymbol{r}) = \frac{\boldsymbol{H}\psi(\boldsymbol{r})}{\psi(\boldsymbol{r})} = -\frac{1}{2}\sum_i \left(\nabla_i^2 \log\psi(\boldsymbol{r}) + (\nabla_i \log\psi(\boldsymbol{r}))^2\right) + V(x)$$

- Try to model the score of the gradient'

$$\boldsymbol{s}(\boldsymbol{r}) = \nabla_{\boldsymbol{r}} \log\psi(\boldsymbol{r})$$

- Then local energy become

$$E_L(\boldsymbol{r}) = -\frac{1}{2}\left(\mathrm{tr}(\nabla_{\boldsymbol{r}}\boldsymbol{s}(\boldsymbol{r}) + \|\boldsymbol{s}(\boldsymbol{r})\|^2\right) + V(\boldsymbol{r})$$

https://openreview.net/forum?id=rMQ1Wme3S0c

Zhang, X., Xu, S., & Ji, S. (2023). A Score-Based Model for Learning Neural Wavefunctions. arXiv. https://doi.org/10.48550/arxiv.2305.16540

A Score-Based Model for Neural Wavefunctions (23' rejected)

- Try to model the score of the gradient'

$$\boldsymbol{s}(\boldsymbol{r}) = \nabla_{\boldsymbol{r}} \log \psi(\boldsymbol{r})$$

- Model the $|\psi(\boldsymbol{r})|^2$ distribution through Langevin dynamics

$$\boldsymbol{r}_{t+1} = \boldsymbol{r}_t + \sqrt{\alpha}\boldsymbol{\epsilon} + \alpha \boldsymbol{s}(\boldsymbol{r}_t)$$

- Limitation
  - Score is computed from the FermiNet..
    - One of the rejected reason is it looks like surrogate model of the FermiNet
  - Hard to apply operator to wavefunction, i.e., $\mathrm{d}\Psi/\mathrm{dx}$

https://openreview.net/forum?id=rMQ1Wme3S0c

Zhang, X., Xu, S., & Ji, S. (2023). A Score-Based Model for Learning Neural Wavefunctions. arXiv. https://doi.org/10.48550/arxiv.2305.16540

Thank you

Diffusion Monte Carlo Method

- Conceptually difficult to me and I still try to understanding…
- For any trial wave function Ψ and exact wave function Ψ_0

$$\Psi(\boldsymbol{r},t) = -e^{-(\hat{H}-E_T)t}\Psi(\boldsymbol{r})$$

- When time t go to infinity, this function proportional to Ψ_0

$$\lim_{t\to\infty}\Psi(\boldsymbol{r},t) = -e^{-(E_0-E_T)t}\Psi_0(\boldsymbol{r})\int\Psi_0(\boldsymbol{r})\Psi(\boldsymbol{r})d\boldsymbol{r}$$

$$\lim_{t\to\infty}\Psi(\boldsymbol{r},t) \propto \Psi_0(\boldsymbol{r})$$

https://en.wikipedia.org/wiki/Diffusion_Monte_Carlo

Diffusion Monte Carlo Method

- Conceptually difficult to me and I still try to understanding…
- For any trial wave function Ψ and exact wave function Ψ_0

$$\Psi(\boldsymbol{r}, t) = -e^{-(\hat{H}-E_T)t}\Psi(\boldsymbol{r})$$

- When time t go to infinity, this function proportional to Ψ_0

$$\lim_{t\to\infty}\Psi(\boldsymbol{r}, t) = -e^{-(E_0-E_T)t}\Psi_0(\boldsymbol{r})\int\Psi_0(\boldsymbol{r})\Psi(\boldsymbol{r})d\boldsymbol{r}$$

$$\lim_{t\to\infty}\Psi(\boldsymbol{r}, t) \propto \Psi_0(\boldsymbol{r})$$

https://en.wikipedia.org/wiki/Diffusion_Monte_Carlo
https://arxiv.org/pdf/1508.02989

Diffusion Monte Carlo Method

- We know infinite time evolved wavefunction is

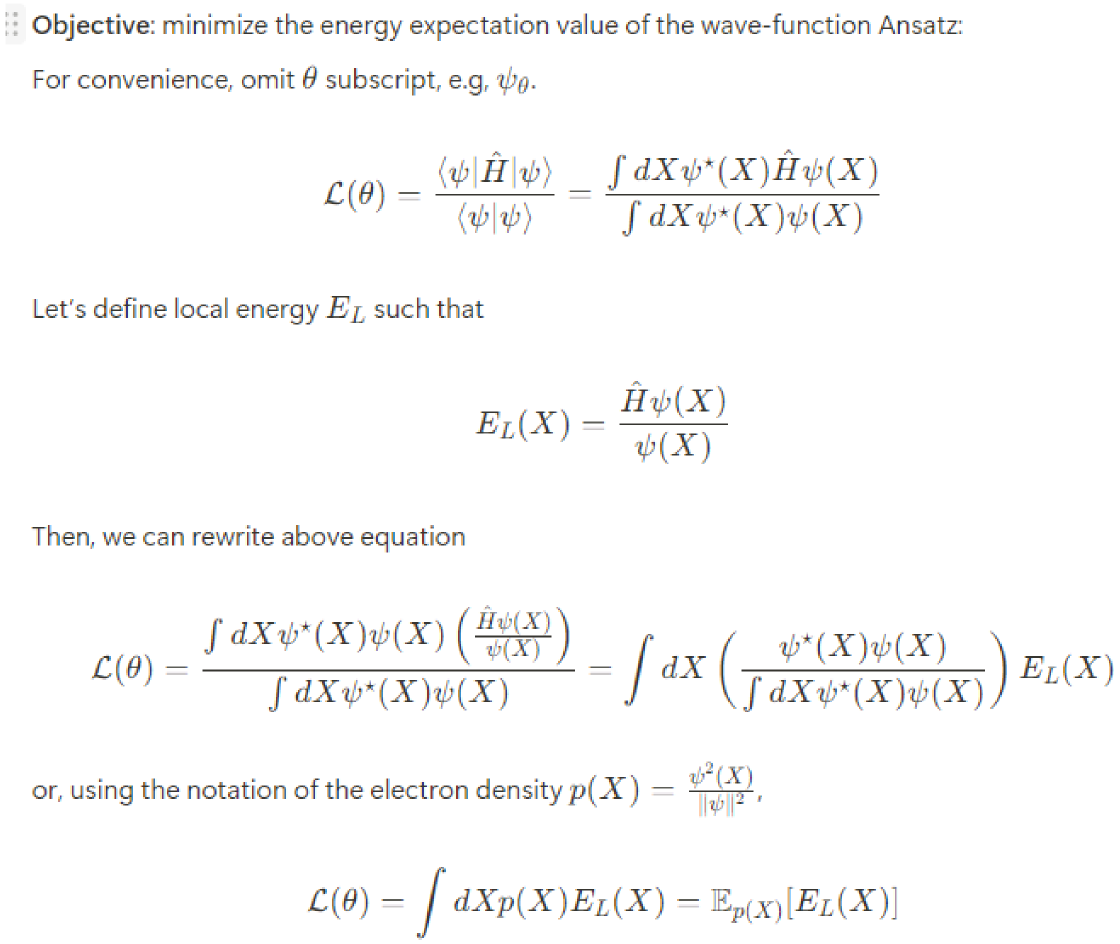
$$\lim_{t\to\infty} \Psi(\boldsymbol{r}, t) \propto \Psi_0(\boldsymbol{r})$$

- However, we only know the time derivative from Schrödinger eq.

$$-\frac{\partial \psi(\boldsymbol{r}, \tau)}{\partial \tau} = \left[\sum_{i=1}^{N} -\frac{1}{2}\nabla_i^2 \psi(\boldsymbol{r}, \tau)\right] + (V(\boldsymbol{r}) - E_T)\psi(\boldsymbol{r}, \tau)$$

- We can define the diffusion equation from above relation…
  - It is so-called "diffusion" Monte Carlo Method
  - Skipped the Monte Carlo part

Appendix 1



Objective: minimize the energy expectation value of the wave-function Ansatz:

For convenience, omit θ subscript, e.g, ψ_θ.

$$\mathcal{L}(\theta) = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\int dX \psi^\star(X)\hat{H}\psi(X)}{\int dX \psi^\star(X)\psi(X)}$$

Let's define local energy E_L such that

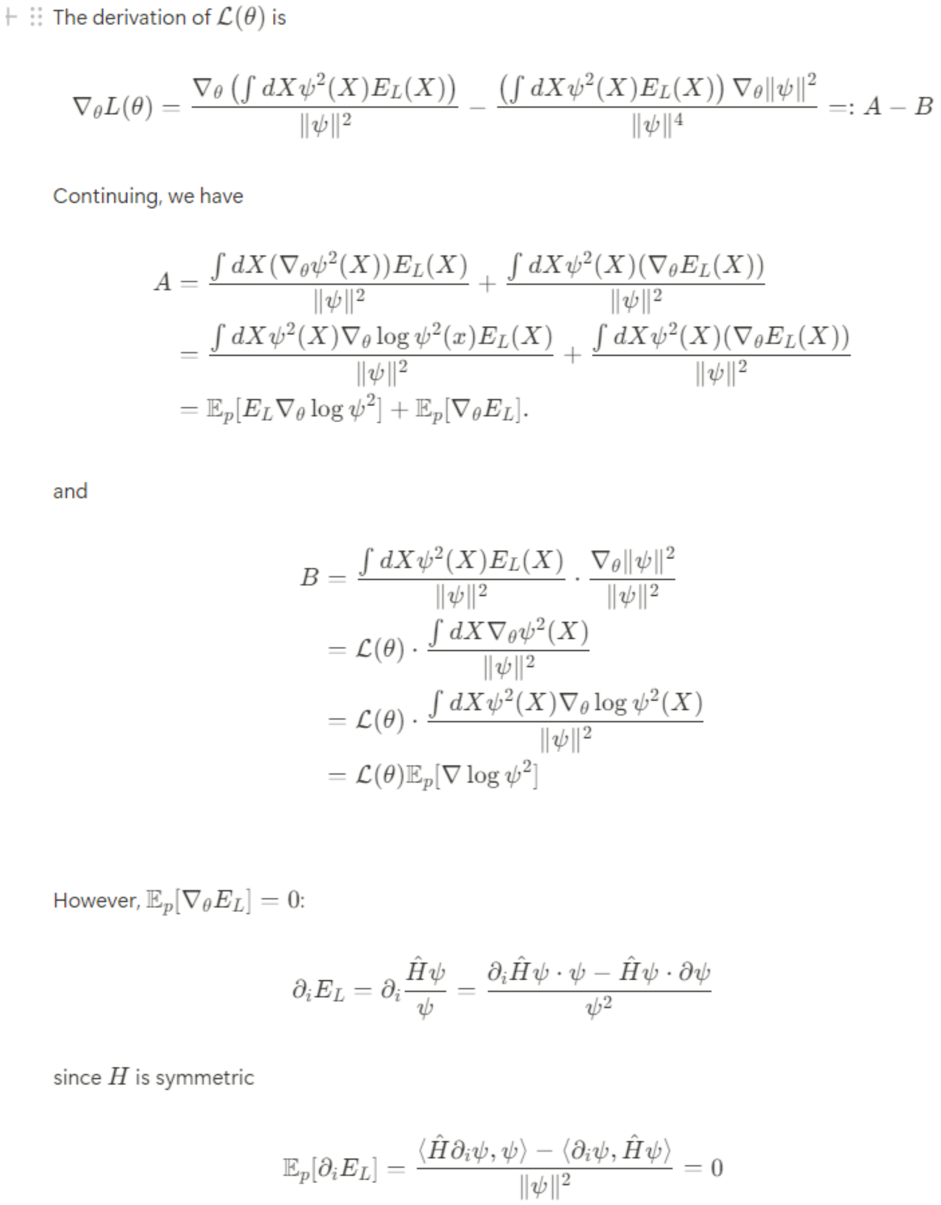
$$E_L(X) = \frac{\hat{H}\psi(X)}{\psi(X)}$$

Then, we can rewrite above equation

$$\mathcal{L}(\theta) = \frac{\int dX \psi^\star(X)\psi(X)\left(\frac{\hat{H}\psi(X)}{\psi(X)}\right)}{\int dX \psi^\star(X)\psi(X)} = \int dX \left(\frac{\psi^\star(X)\psi(X)}{\int dX \psi^\star(X)\psi(X)}\right) E_L(X)$$

or, using the notation of the electron density $p(X) = \frac{\psi^2(X)}{\|\psi\|^2}$,

$$\mathcal{L}(\theta) = \int dX p(X) E_L(X) = \mathbb{E}_{p(X)}[E_L(X)]$$



The derivation of $\mathcal{L}(\theta)$ is

$$\nabla_\theta L(\theta) = \frac{\nabla_\theta\left(\int dX \psi^2(X)E_L(X)\right)}{\|\psi\|^2} - \frac{\left(\int dX \psi^2(X)E_L(X)\right)\nabla_\theta\|\psi\|^2}{\|\psi\|^4} =: A - B$$

Continuing, we have

$$\begin{aligned} A &= \frac{\int dX (\nabla_\theta\psi^2(X))E_L(X)}{\|\psi\|^2} + \frac{\int dX \psi^2(X)(\nabla_\theta E_L(X))}{\|\psi\|^2} \\ &= \frac{\int dX \psi^2(X)\nabla_\theta \log\psi^2(x)E_L(X)}{\|\psi\|^2} + \frac{\int dX \psi^2(X)(\nabla_\theta E_L(X))}{\|\psi\|^2} \\ &= \mathbb{E}_p[E_L\nabla_\theta\log\psi^2] + \mathbb{E}_p[\nabla_\theta E_L]. \end{aligned}$$

and

$$\begin{aligned} B &= \frac{\int dX \psi^2(X)E_L(X)}{\|\psi\|^2}\cdot\frac{\nabla_\theta\|\psi\|^2}{\|\psi\|^2} \\ &= \mathcal{L}(\theta)\cdot\frac{\int dX \nabla_\theta\psi^2(X)}{\|\psi\|^2} \\ &= \mathcal{L}(\theta)\cdot\frac{\int dX \psi^2(X)\nabla_\theta\log\psi^2(X)}{\|\psi\|^2} \\ &= \mathcal{L}(\theta)\mathbb{E}_p[\nabla\log\psi^2] \end{aligned}$$

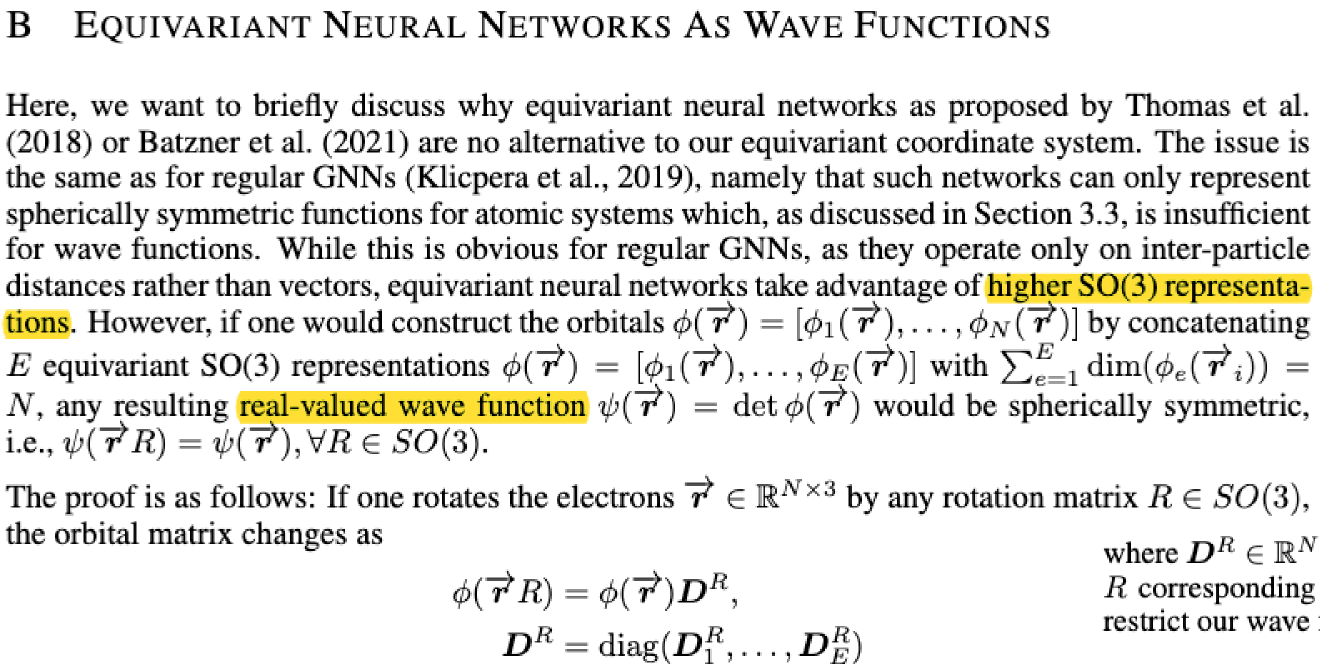
However, $\mathbb{E}_p[\nabla_\theta E_L] = 0$:

$$\partial_i E_L = \partial_i\frac{\hat{H}\psi}{\psi} = \frac{\partial_i\hat{H}\psi\cdot\psi - \hat{H}\psi\cdot\partial\psi}{\psi^2}$$

since H is symmetric

$$\mathbb{E}_p[\partial_i E_L] = \frac{\langle\hat{H}\partial_i\psi,\psi\rangle - \langle\partial_i\psi,\hat{H}\psi\rangle}{\|\psi\|^2} = 0$$

Appendix 2

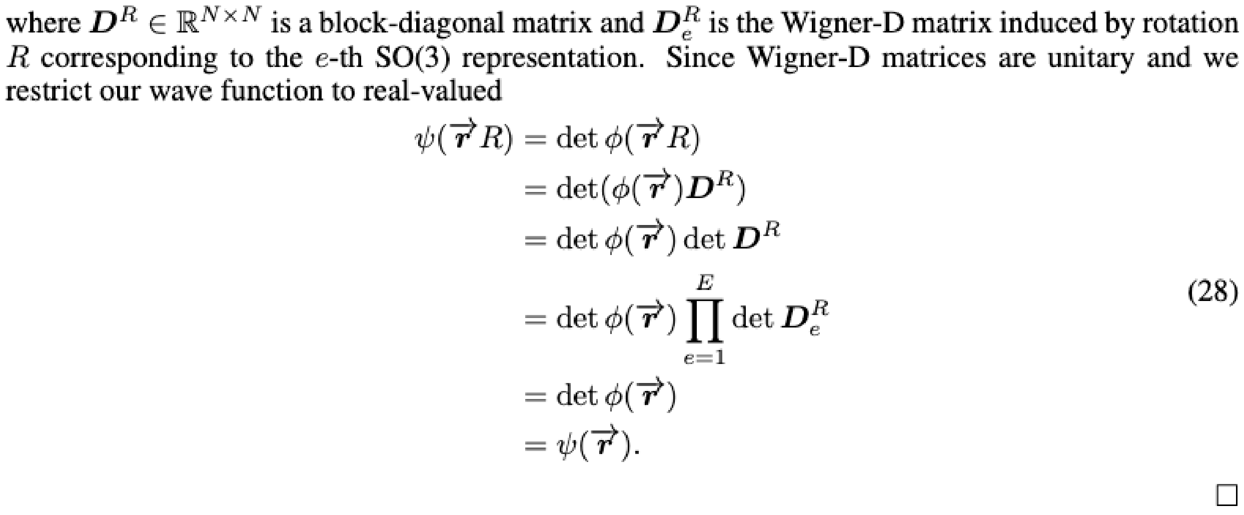


B Equivariant Neural Networks As Wave Functions

Here, we want to briefly discuss why equivariant neural networks as proposed by Thomas et al. (2018) or Batzner et al. (2021) are no alternative to our equivariant coordinate system. The issue is the same as for regular GNNs (Klicpera et al., 2019), namely that such networks can only represent spherically symmetric functions for atomic systems which, as discussed in Section 3.3, is insufficient for wave functions. While this is obvious for regular GNNs, as they operate only on inter-particle distances rather than vectors, equivariant neural networks take advantage of higher SO(3) representations. However, if one would construct the orbitals $\phi(\vec{\boldsymbol{r}}) = [\phi_1(\vec{\boldsymbol{r}}), \dots, \phi_N(\vec{\boldsymbol{r}})]$ by concatenating E equivariant SO(3) representations $\phi(\vec{\boldsymbol{r}}) = [\phi_1(\vec{\boldsymbol{r}}), \dots, \phi_E(\vec{\boldsymbol{r}})]$ with $\sum_{e=1}^{E} \dim(\phi_e(\vec{\boldsymbol{r}}_i)) = N$, any resulting real-valued wave function $\psi(\vec{\boldsymbol{r}}) = \det \phi(\vec{\boldsymbol{r}})$ would be spherically symmetric, i.e., $\psi(\vec{\boldsymbol{r}}R) = \psi(\vec{\boldsymbol{r}}), \forall R \in SO(3)$.

The proof is as follows: If one rotates the electrons $\vec{\boldsymbol{r}} \in \mathbb{R}^{N\times 3}$ by any rotation matrix $R \in SO(3)$, the orbital matrix changes as

$$\begin{aligned}\phi(\vec{\boldsymbol{r}}R) &= \phi(\vec{\boldsymbol{r}})\boldsymbol{D}^R,\\ \boldsymbol{D}^R &= \operatorname{diag}(\boldsymbol{D}_1^R, \dots, \boldsymbol{D}_E^R)\end{aligned}$$



where $\boldsymbol{D}^R \in \mathbb{R}^{N\times N}$ is a block-diagonal matrix and $\boldsymbol{D}_e^R$ is the Wigner-D matrix induced by rotation R corresponding to the e-th SO(3) representation. Since Wigner-D matrices are unitary and we restrict our wave function to real-valued

$$\begin{aligned}\psi(\vec{\boldsymbol{r}}R) &= \det \phi(\vec{\boldsymbol{r}}R)\\ &= \det(\phi(\vec{\boldsymbol{r}})\boldsymbol{D}^R)\\ &= \det \phi(\vec{\boldsymbol{r}}) \det \boldsymbol{D}^R\\ &= \det \phi(\vec{\boldsymbol{r}}) \prod_{e=1}^{E} \det \boldsymbol{D}_e^R\\ &= \det \phi(\vec{\boldsymbol{r}})\\ &= \psi(\vec{\boldsymbol{r}}).\end{aligned} \tag{28}$$

□

Relation with REINFORCE

- VMC

$$E_\theta = \mathbb{E}_{p(r)}[E_L(r)] \qquad p(r) = \frac{\psi^2(r)}{\int \psi^2(r)dr}$$

$$\nabla_\theta E_\theta = \mathbb{E}_{p(r)}[(E_L(r) - \mathbb{E}_{p(r)}[E_L(r)])\nabla_\theta \log p(r)]$$

- Policy Gradient

$$J(\theta) = \mathbb{E}_{\pi_\theta(a|s)}[r(s,a)]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta(a|s)}[r(s,a)\nabla_\theta \log \pi_\theta(a|s)]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta(a|s)}[(r(s,a) - b(s))\nabla_\theta \log \pi_\theta(a|s)]$$

Relation with EBM

- VMC

$$E_\theta = \mathbb{E}_{p(r)}[E_L(r)] \qquad p(r) = \frac{\psi^2(r)}{\int \psi^2(r)dr}$$

$$\nabla_\theta E_\theta = \mathbb{E}_{p(r)}[(E_L(r) - \mathbb{E}_{p(r)}[E_L(r)])\nabla_\theta \log p(r)]$$

- EBM

$$p(r) = \frac{\exp(-f_\theta(r))}{Z(\theta)} \quad Z(\theta) = \int \psi^2(r)dr, \quad \log \psi(r) = -\frac{f_\theta(r)}{2}$$

$$\nabla_\theta E_\theta = \mathbb{E}_{p_\theta(x)}\left[\left(f_\theta(x) - \mathbb{E}_{p_\theta(x)}\left[f_\theta(x)\right]\right)\nabla_\theta \log p_\theta(x)\right].$$