Freie Universität Berlin

Basic principles of application of machine learning in quantum chemistry

Jan Hermann

FU Berlin, Department of Mathematics, Berlin, Germany
https://jan.hermann.name

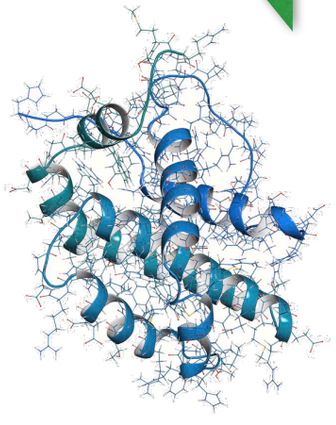
VŠCHT, 2 Jun 2022

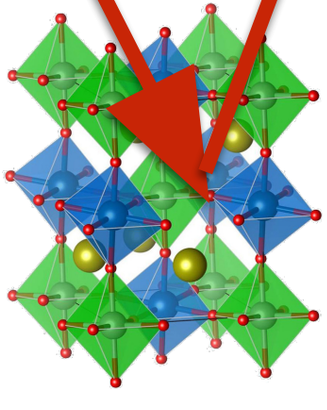
Computational modeling of molecules and materials: What and Why?

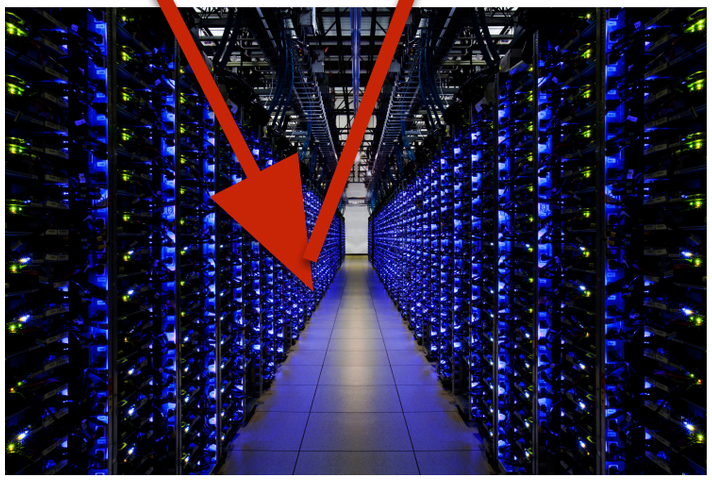
molecules and materials

lab

atom





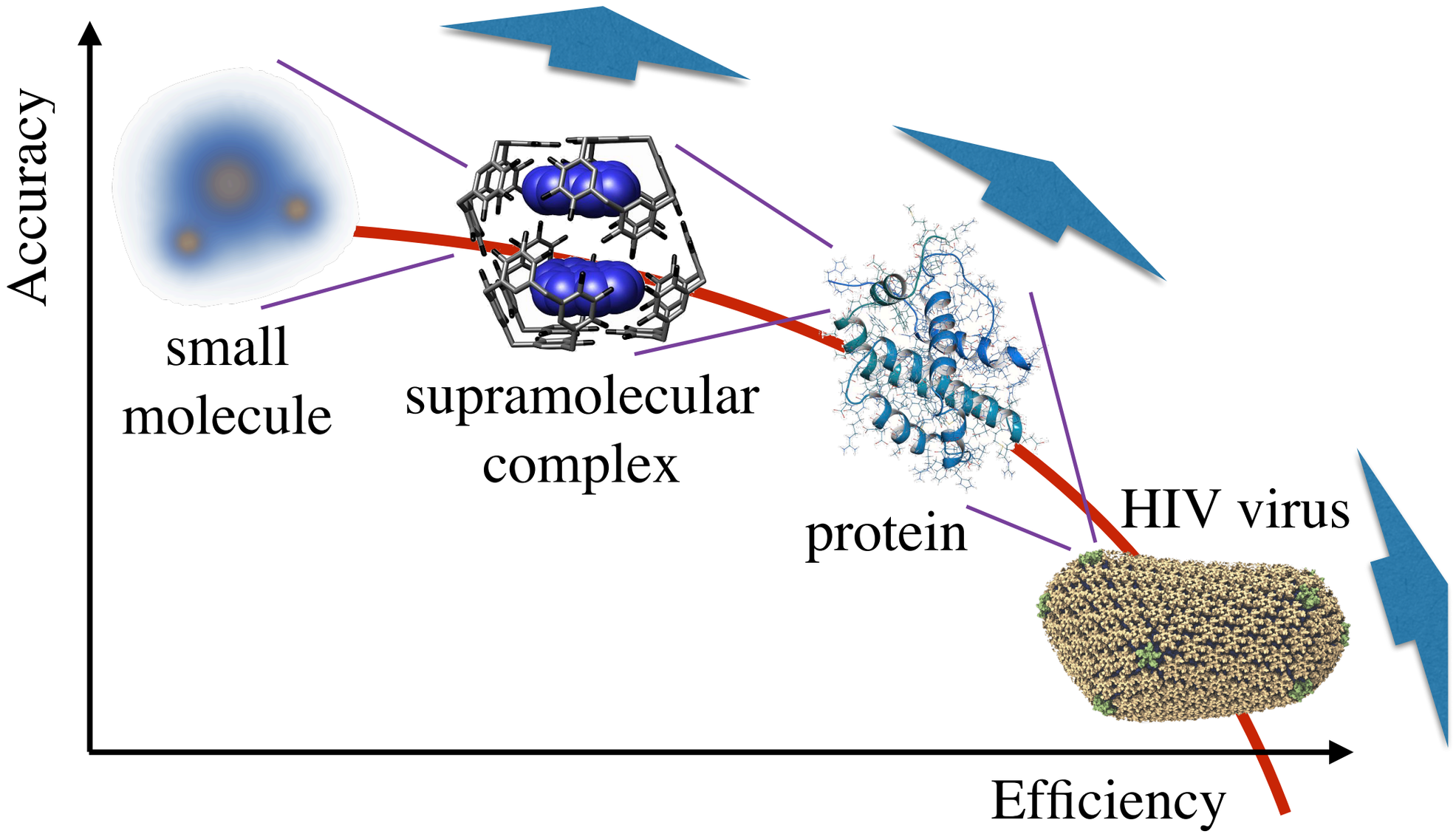


efficient solar cell?

effective cancer drug?

Computational modeling of molecules and materials: How?

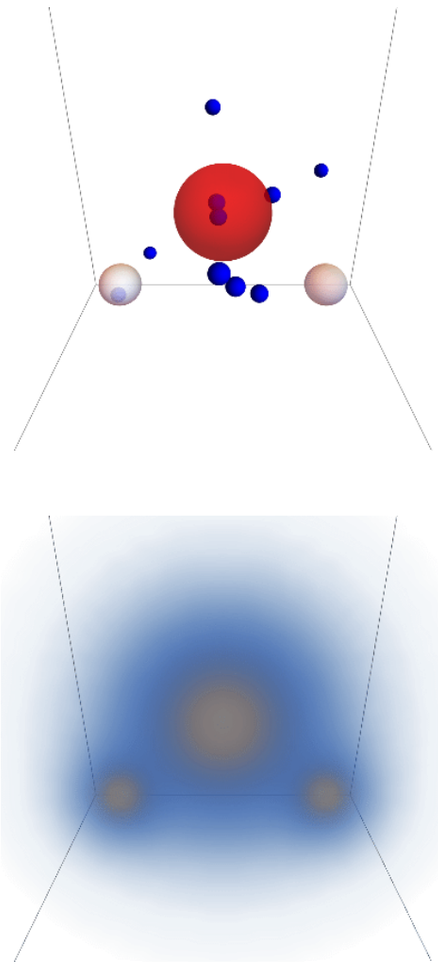
- Schrödinger equation – exact in principle, impossible to solve exactly in practice



Atoms and electrons: Thermodynamics and electronic structure

- Molecules made of nuclei ($\mathbf{R}_I$) and electrons ($\mathbf{r}_i$)
- Total molecular wave function

$$\psi(\mathbf{r}_1, \ldots, \mathbf{r}_N, \mathbf{R}_1, \ldots, \mathbf{R}_M)$$

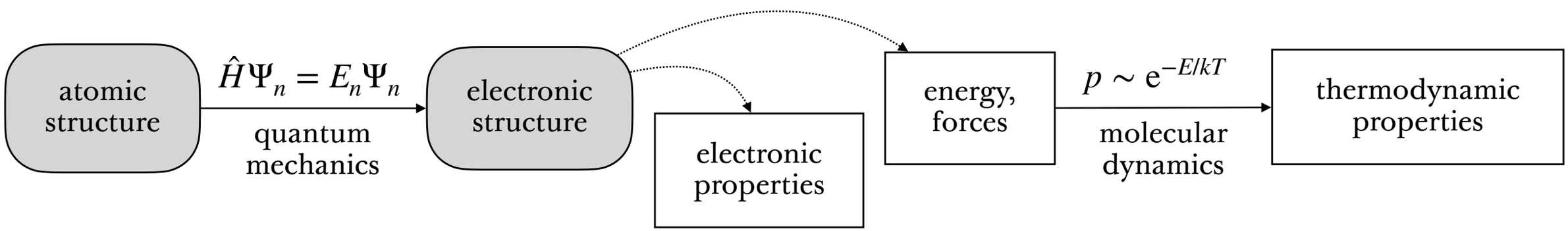


- Born–Oppenheimer approximation: separation of nuclear and electronic motion

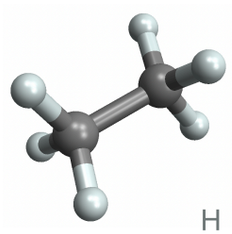
$$(\mathbf{R}_1, \ldots, \mathbf{R}_N) \rightarrow \hat{H}_{\text{elec}} \rightarrow \underset{\textit{electronic wave function}}{\psi_n(\mathbf{r}_1, \ldots, \mathbf{r}_N)}, E_n$$

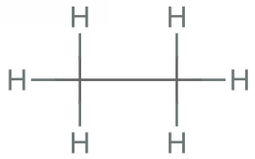
$$E_n(\mathbf{R}_1, \ldots, \mathbf{R}_M) \rightarrow \hat{H}_{\text{nucl}}$$

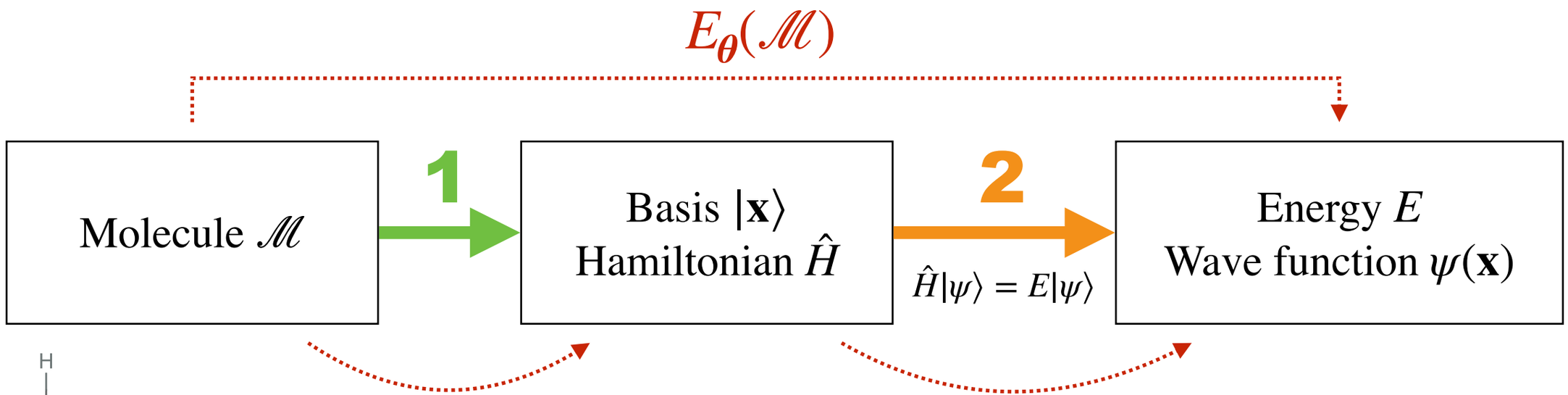
potential energy surface



Electronic structure problem





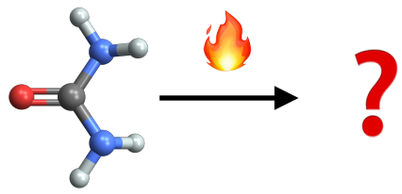
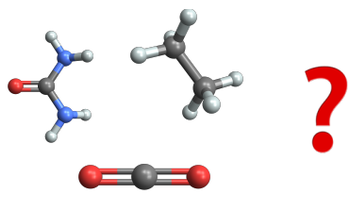
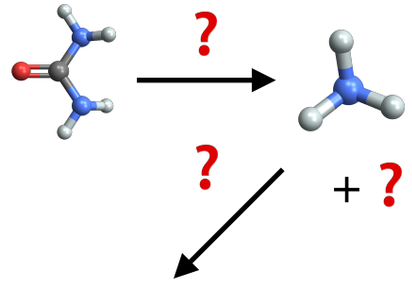


- Ab-initio: exact 1, approximate 2 (*coupled clusters*, *quantum Monte Carlo*)
- Semi-empirical: approximate 1, exact 2 (*density functional theory*, *tight-binding*, *Hubbard model*)
- Machine learning (force fields, density functionals, wave function ansatzes)

Machine learning: Data and objectives

- Human learning: World model from observation and intervention
- Machine learning: Algorithms improve from empirical evidence
- Data: Formalized empirical evidence, subset of “ground truth”
- Objective: Measure of quality of a model
- Generalization: Does the model work on new empirical evidence?

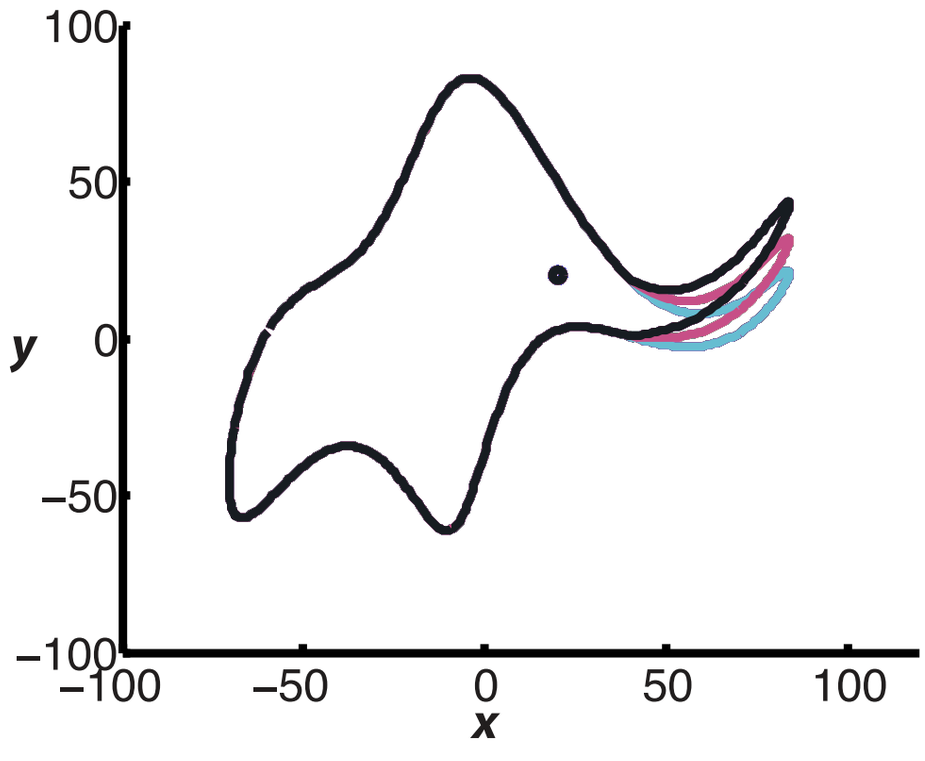
Machine learning flavors

	Data	Model	Objective/ ground truth	
Supervised	$\{\mathbf{x}_i, \mathbf{y}_i\}$	$f_{\boldsymbol{\theta}} : \mathbf{x} \rightarrow \mathbf{y}$	$\sum_i \lvert\mathbf{y}_i - f_{\boldsymbol{\theta}}(\mathbf{x}_i)\rvert^2$	
Unsupervised (generative)	$\{\mathbf{x}_i\}$	$f_{\boldsymbol{\theta}} : \mathbf{z} \rightarrow \mathbf{x}$	$\mathscr{L}(\mathbf{x})$	
Reinforcement	🔁	$f_{\boldsymbol{\theta}} : (\mathbf{x}, \mathbf{z}) \rightarrow \mathbf{y}$	$\mathscr{L}(🔁)$	

Machine learning: Glorified fitting?

"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

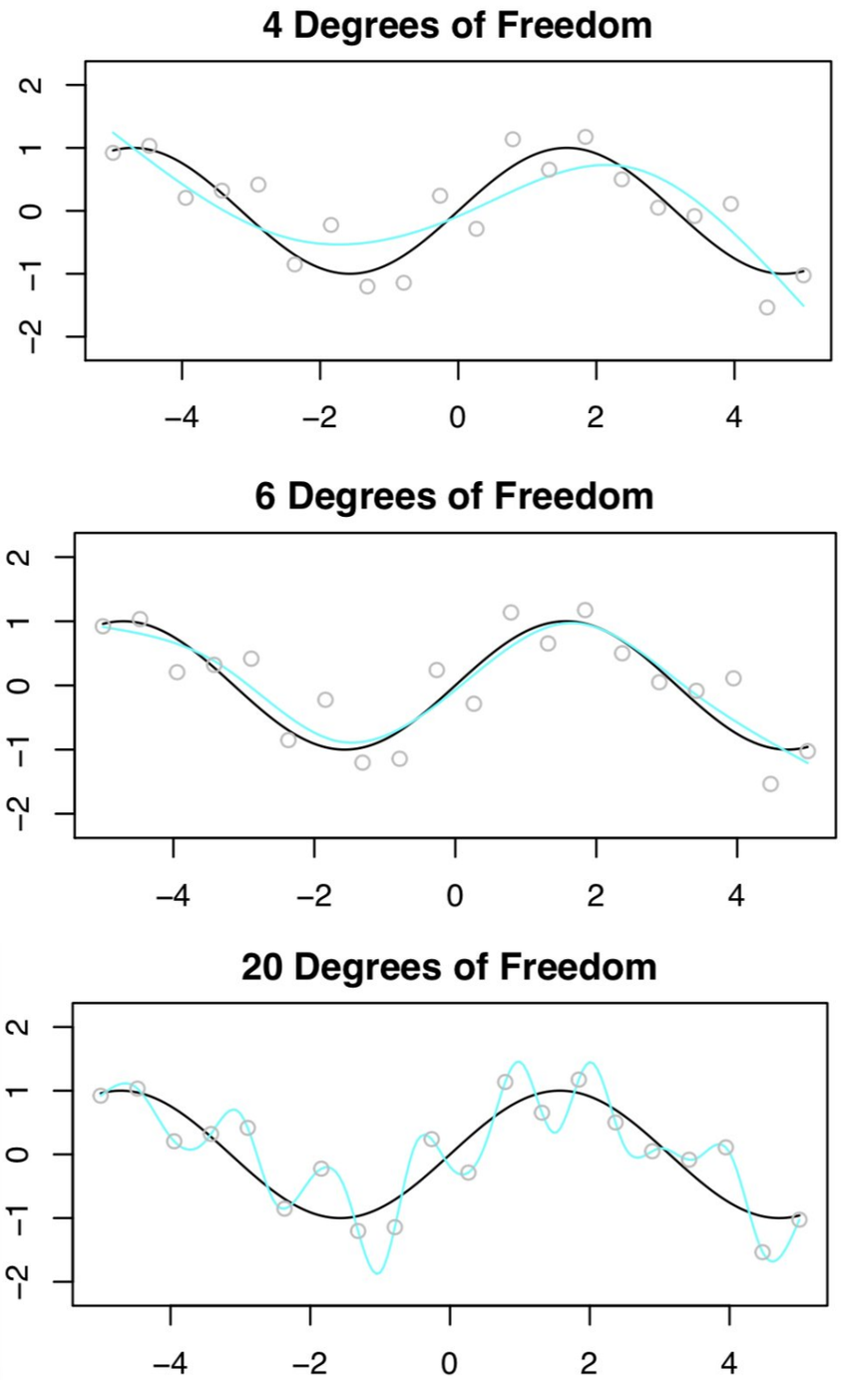
—J. von Neumann

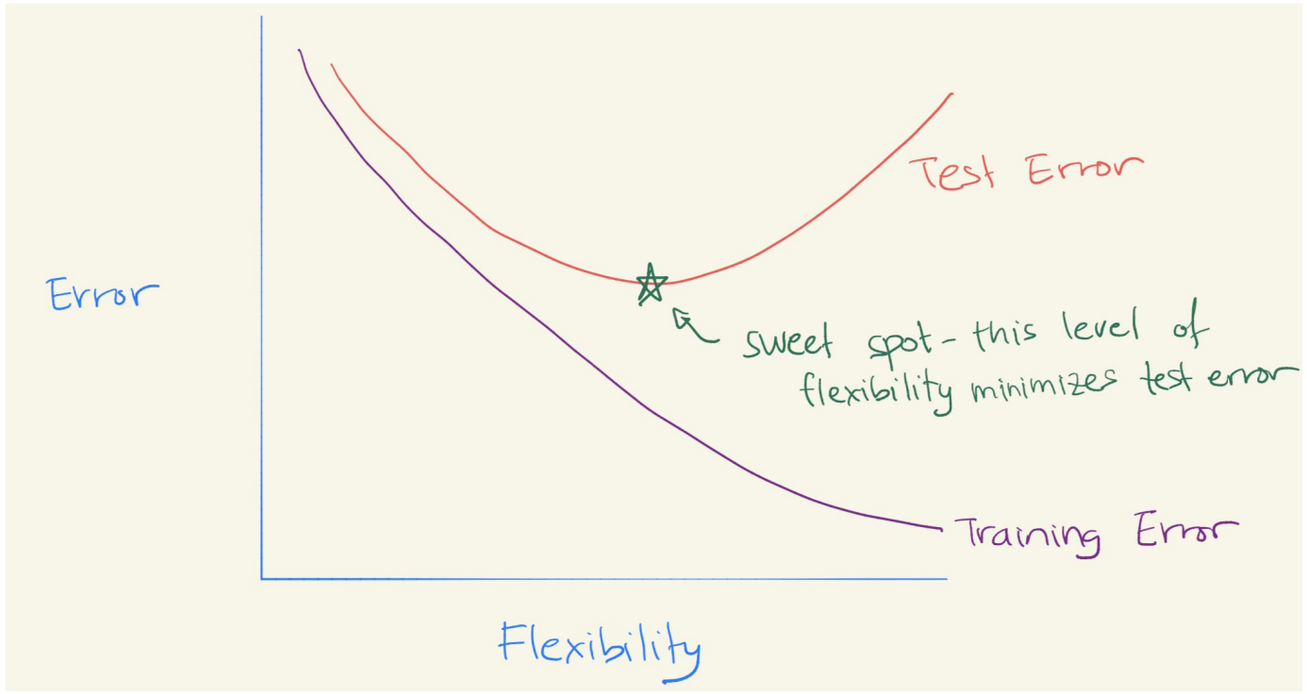


neural networks	number of parameters
toy models	~10^3
serious models	>10^5
AlphaFold 2	~10^8
GPT-3	~10^{11}

Mayer et al. Drawing an elephant with four complex parameters. *AJP* **78**, 648 (2010)

Bias–variance tradeoff and overfitting

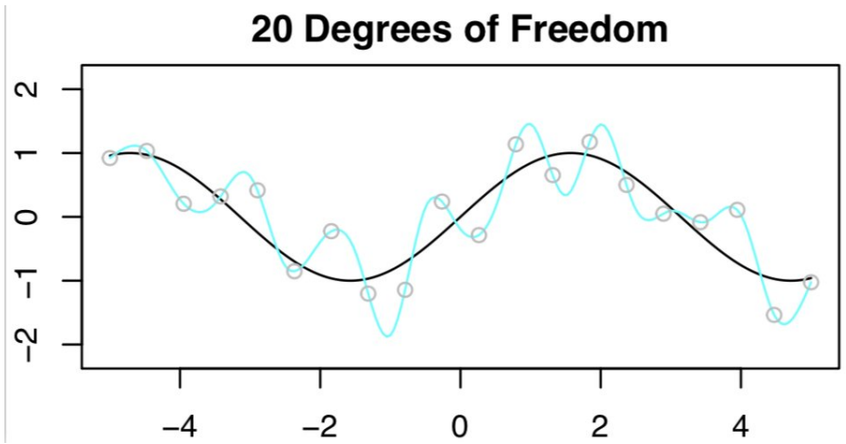


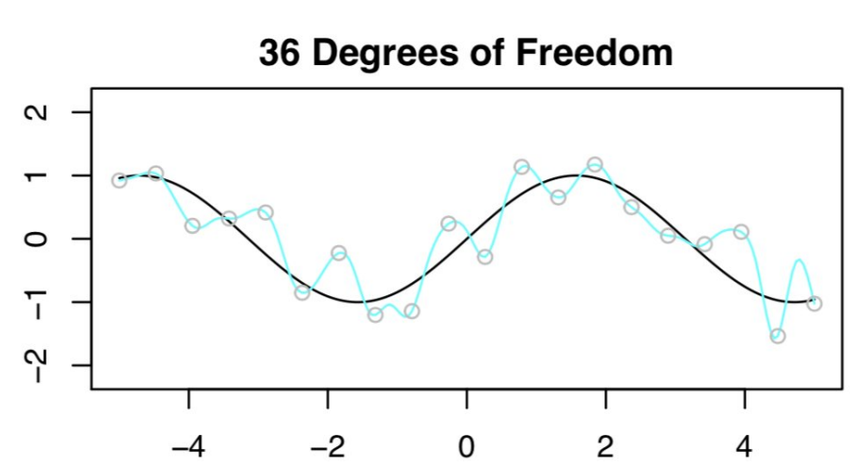


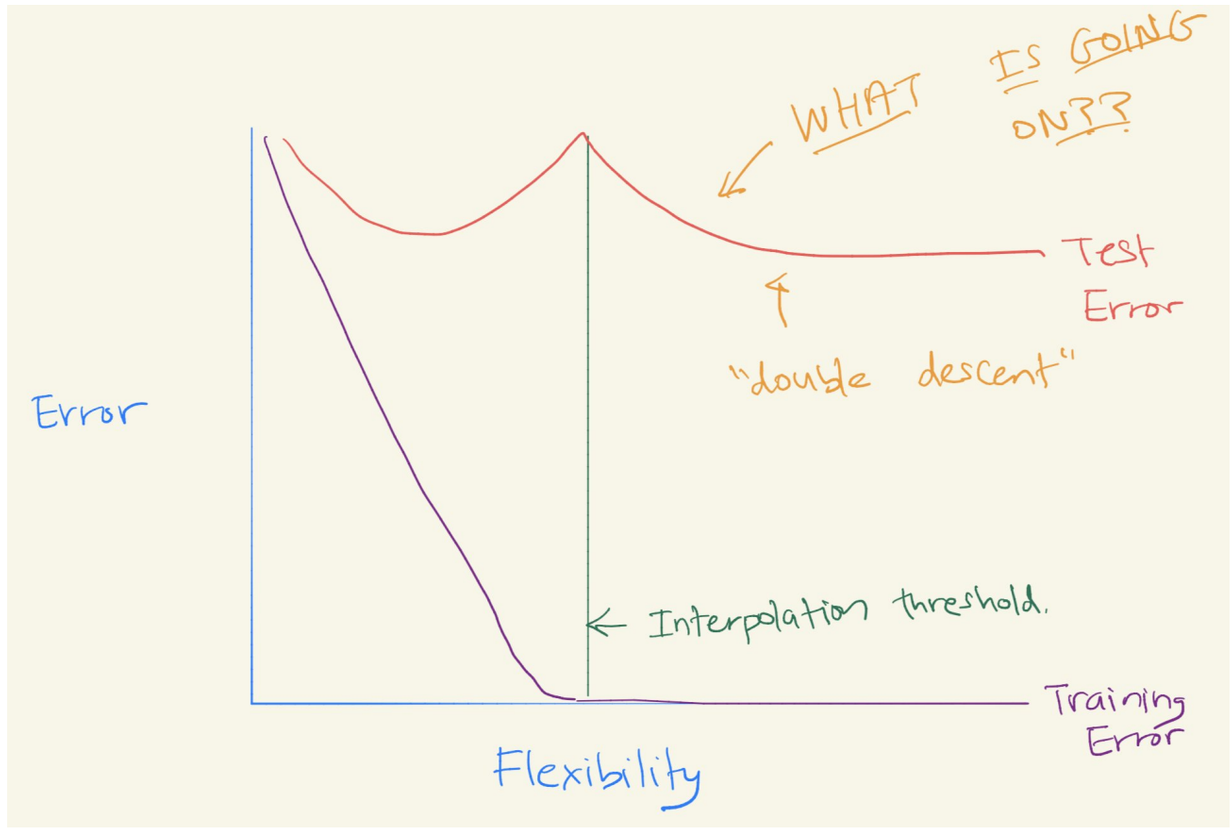
- Increasing number of parameters eventually leads to overfitting

@daniela_witten

Regularization and high dimensionality



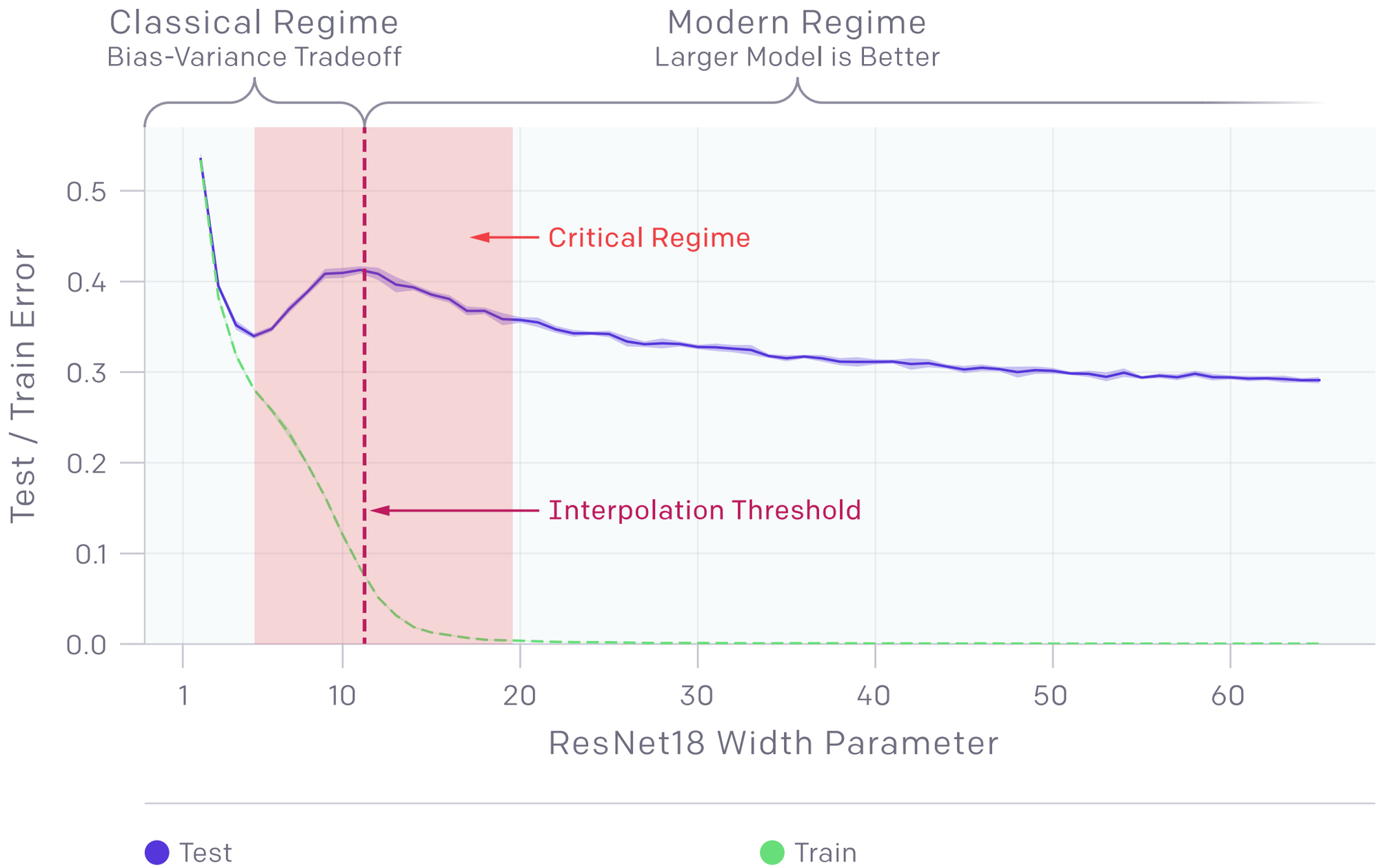




- Regularization—picking the most smooth fit
- Balestriero et al.: Learning in High Dimension Always Amounts to Extrapolation

@daniela_witten

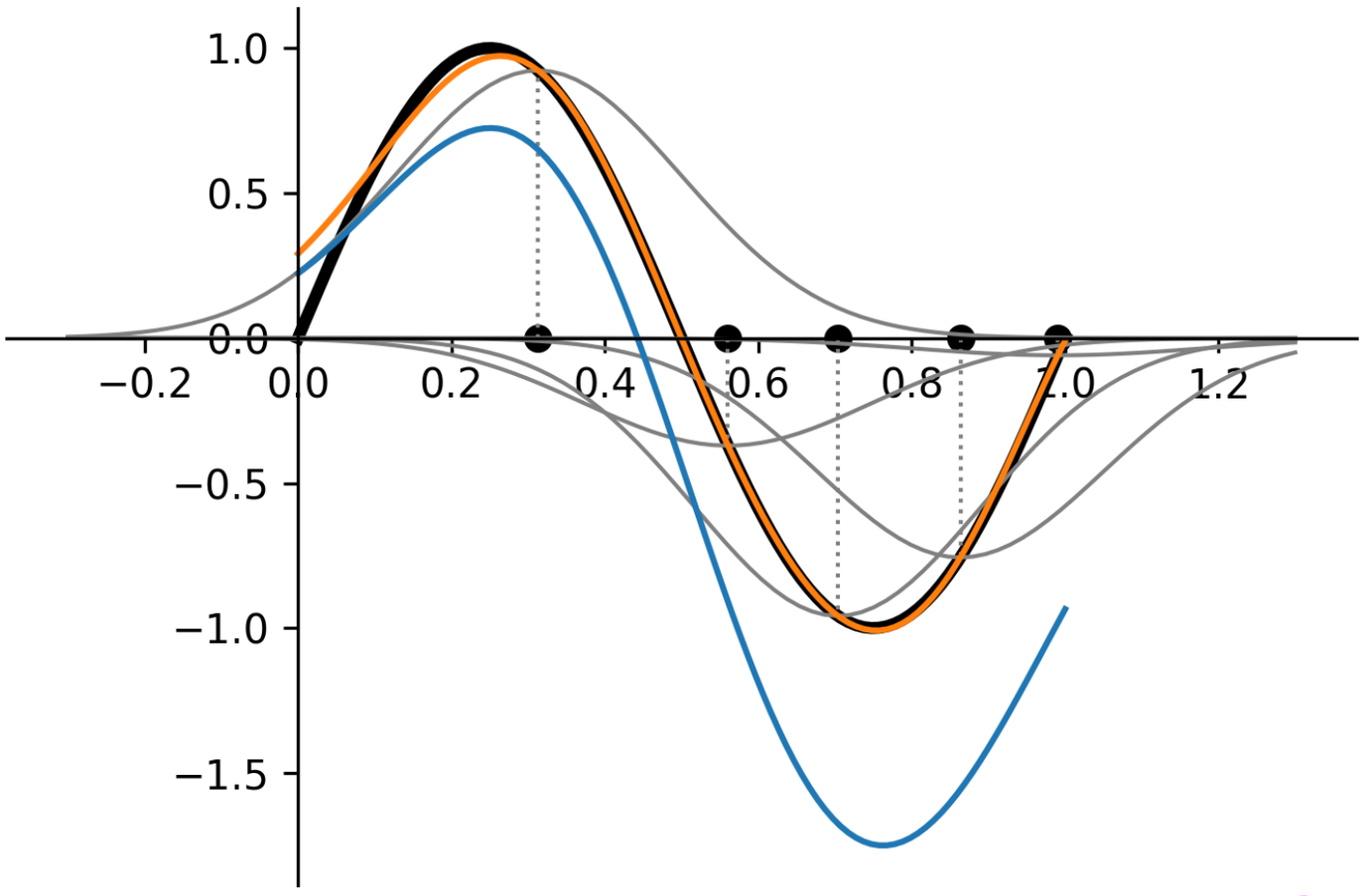
Machine learning and “double descent”



Nakkiran et al. Deep Double Descent (ICRL, 2019)

Gaussian process regression

$$f(x) = \sin(2\pi x)$$

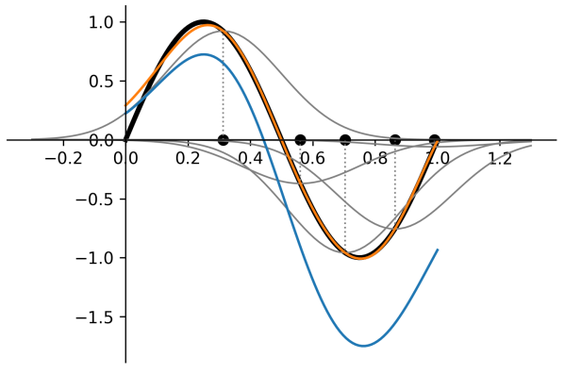


$$f(x) = \sum_i y_i \mathrm{e}^{-\frac{(x - x_i)^2}{\sigma^2}}$$

$$f(x) = \sum_k c_k \mathrm{e}^{-\frac{(x - x_i)^2}{\sigma^2}}$$

$$\mathbf{c} = \mathbf{K}^{-1}\mathbf{y}, \quad K_{ij} = \mathrm{e}^{-\frac{(x_i - x_j)^2}{\sigma^2}}$$

Gaussian process regression



$$y_j = \sum_k W_{jk} \mathrm{e}^{-\frac{|\boldsymbol{\xi}(\mathbf{x}) - \boldsymbol{\xi}(\mathbf{x}_k)|^2}{\sigma^2}}$$

$$\boldsymbol{\xi} : \quad \rightarrow \mathbb{R}^n$$

- Straightforward generalization to multiple dimensions and arbitrary *descriptors*
- Needs good descriptors
- “Just” a linear combination – limited flexibility
- Limited to small datasets
- Largely replaced with neural networks

From Gaussian processes to neural networks

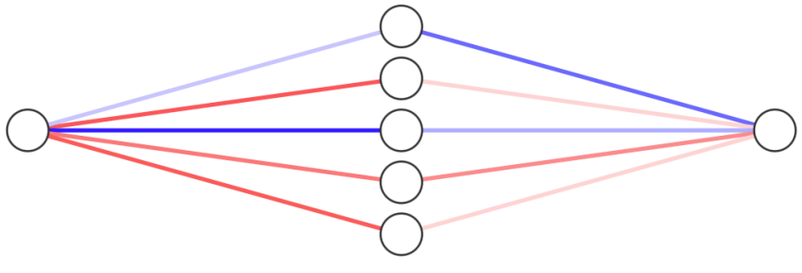
1

$$y = \sum_k W_k \mathrm{e}^{-\frac{(x - x_k)^2}{\sigma^2}}$$

$$= \sum_k W_k \mathrm{e}^{-(\sigma^{-1}x - \sigma^{-1}x_k)^2}$$

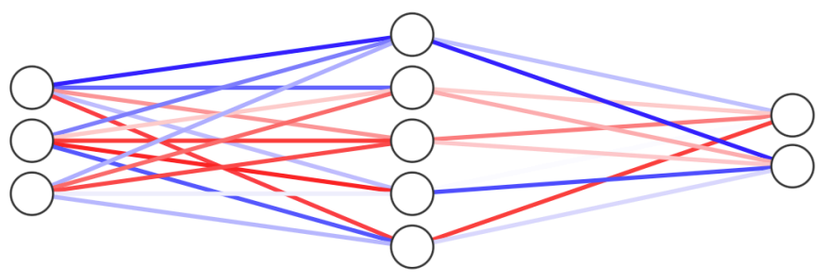
$$= \sum_k W_k \mathrm{e}^{-(W'_k x - b_k)^2}$$

$$= \sum_k W_k g(W'_k x - b_k)$$



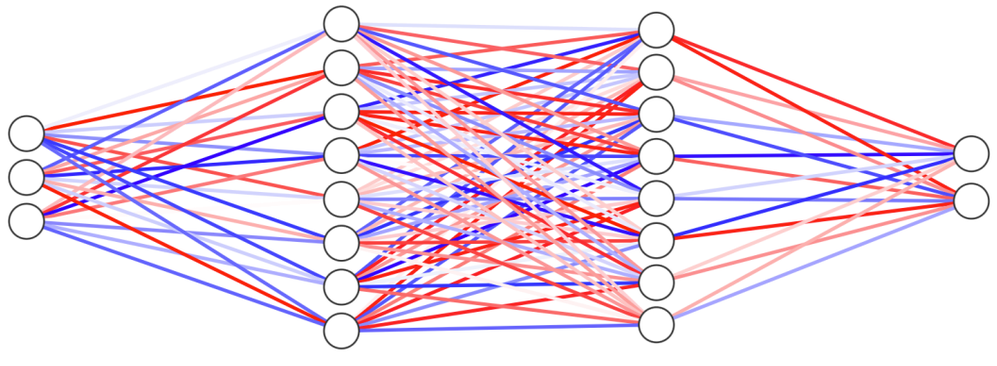
2

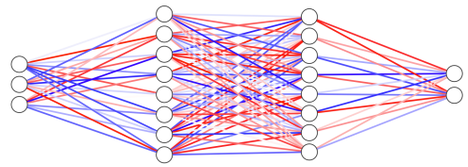
$$\mathbf{y} = \mathbf{W} g(\mathbf{W}'\mathbf{x} - \mathbf{b})$$



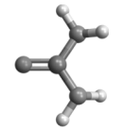
3

$$\mathbf{y} = \mathbf{W} g\Big(\mathbf{W}' g(\mathbf{W}''\mathbf{x} - \mathbf{b}') - \mathbf{b}\Big)$$

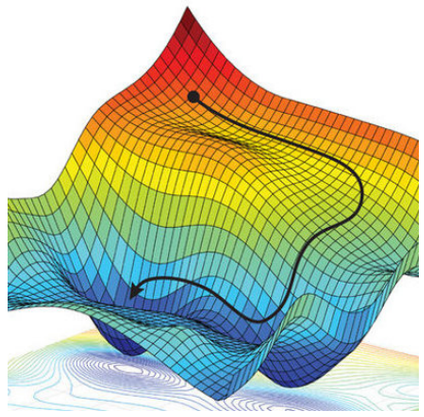




Neural networks and deep learning

$$\mathbf{y} = \mathbf{W} g\big(\mathbf{W}' g(\mathbf{W}'' \xi(\mathbf{x}) - \mathbf{b}') - \mathbf{b}\big) \qquad \xi : \text{  } \to \mathbb{R}^n$$

- Deep neural networks are “universal approximators”
  – good descriptors unnecessary
- Cannot be optimized analytically – gradient descent
- Fast evaluation of gradients via backpropagation
- Stochastic gradient descent from minibatches (implicit regularization)



$$\boldsymbol{\theta} := \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathscr{L}\big(\boldsymbol{\theta}, \{\mathbf{x}_i\}_{i \in B_n}\big)$$

Some deep learning successes

1

RESEARCH

COMPUTER SCIENCE

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver[1,2*†], Thomas Hubert[1*], Julian Schrittwieser[1*], Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Kumaran[1], Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], Demis Hassabis[1†]

The game of chess is the longest-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. By contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go by reinforcement learning from self-play. In this paper, we generalize this approach into a single AlphaZero algorithm that can achieve superhuman performance in many challenging games. Starting from random play and given no domain knowledge except the game rules, AlphaZero convincingly defeated a world champion program in the games of chess and shogi (Japanese chess), as well as Go.

programmers, combined with a high-performance alpha-beta search that expands a vast search tree by using a large number of clever heuristics and domain-specific adaptations. In (*10*) we describe these augmentations, focusing on the 2016 Top Chess Engine Championship (TCEC) season 9 world champion Stockfish (*11*); other strong chess programs, including Deep Blue, use very similar architectures (*1*, *12*).

In terms of game tree complexity, shogi is a substantially harder game than chess (*13*, *14*): It is played on a larger board with a wider variety of pieces; any captured opponent piece switches sides and may subsequently be dropped anywhere on the board. The strongest shogi programs, such as the 2017 Computer Shogi Association (CSA) world champion Elmo, have only recently defeated human champions (*15*). These programs use an algorithm similar to those used by computer chess programs, again based on a highly optimized alpha-beta search engine with many domain-specific adaptations.

AlphaZero replaces the handcrafted knowledge and domain-specific augmentations used in traditional game-playing programs with deep

2

Article

Highly accurate protein structure prediction with AlphaFold

https://doi.org/10.1038/s41586-021-03819-2

Received: 11 May 2021

Accepted: 12 July 2021

Published online: 15 July 2021

Open access

Check for updates

John Jumper[1,4]✉, Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], Rishub Jain[1,4], Jonas Adler[1], Trevor Back[1], Stig Petersen[1], David Reiman[1], Ellen Clancy[1], Michal Zielinski[1], Martin Steinegger[2,3], Michalina Pacholska[1], Tamas Berghammer[1], Sebastian Bodenstein[1], David Silver[1], Oriol Vinyals[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Pushmeet Kohli[1] & Demis Hassabis[1,4]✉

Proteins are essential to life, and understanding their structure can facilitate a mechanistic understanding of their function. Through an enormous experimental effort[1–4], the structures of around 100,000 unique proteins have been determined[5], but this represents a small fraction of the billions of known protein sequences[6,7]. Structural coverage is bottlenecked by the months to years of painstaking effort required to determine a single protein structure. Accurate computational approaches are needed to address this gap and to enable large-scale structural bioinformatics. Predicting the

3

Article

Magnetic control of tokamak plasmas through deep reinforcement learning

https://doi.org/10.1038/s41586-021-04301-9

Received: 14 July 2021

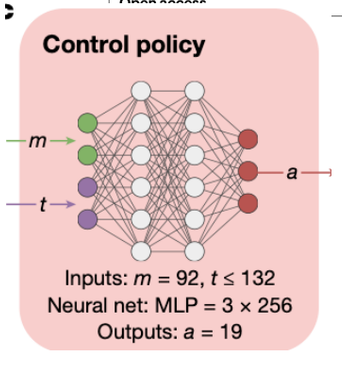
Accepted: 1 December 2021

Published online: 16 February 2022

Open access

Jonas Degrave[1,3], Federico Felici[2,3]✉, Jonas Buchli[1,3]✉, Michael Neunert[1,3], Brendan Tracey[1,3]✉, Francesco Carpanese[1,2,3], Timo Ewalds[1,3], Roland Hafner[1,3], Abbas Abdolmaleki[1], Diego de las Casas[1], Craig Donner[1], Leslie Fritz[1], Cristian Galperti[2], Andrea Huber[1], James Keeling[1], Maria Tsimpoukelli[1], Jackie Kay[1], Antoine Merle[2], Jean-Marc Moret[2], Seb Noury[1], Federico Pesamosca[2], David Pfau[1], Olivier Sauter[2], Cristian Sommariva[2], Stefano Coda[2], Basil Duval[2], Ambrogio Fasoli[2], Pushmeet Kohli[1], Koray Kavukcuoglu[1], Demis Hassabis[1] & Martin Riedmiller[1,3]

Nuclear fusion using magnetic confinement, in particular in the tokamak configuration, is a promising path towards sustainable energy. A core challenge is to shape and maintain a high-temperature plasma within the tokamak vessel. This requires high-dimensional, high-frequency, closed-loop control using magnetic actuator coils, further complicated by the diverse requirements across a wide range of plasma configurations. In this work, we introduce a previously undescribed architecture for tokamak magnetic controller design that autonomously learns to command the full set of control coils. This architecture meets control objectives specified at a high level, at the same time satisfying physical and operational constraints. This approach has unprecedented flexibility and generality in problem specification and yields a notable reduction in design effort to produce new plasma



4

RESEARCH

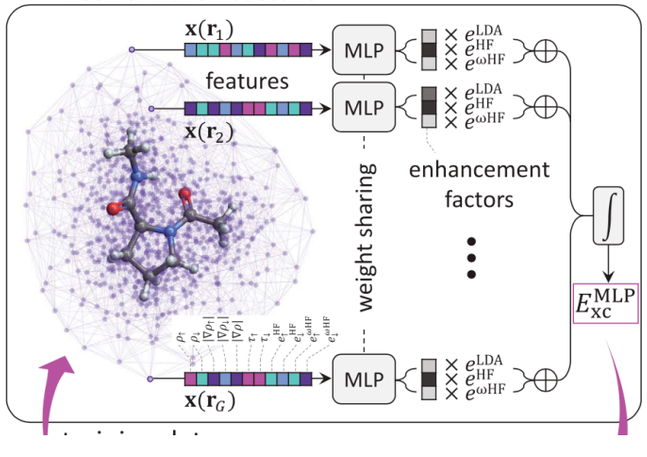
QUANTUM CHEMISTRY

Pushing the frontiers of density functionals by solving the fractional electron problem

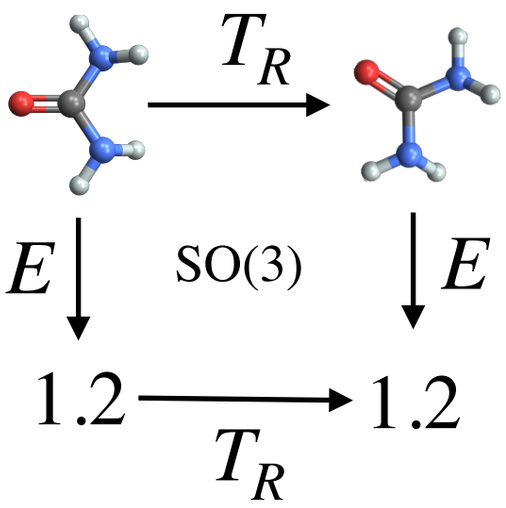
James Kirkpatrick[1*†], Brendan McMorrow[1†], David H. P. Turban[1†], Alexander L. Gaunt[1†], James S. Spencer[1], Alexander G. D. G. Matthews[1], Annette Obika[1], Louis Thiry[2], Meire Fortunato[1], David Pfau[1], Lara Román Castellanos[1], Stig Petersen[1], Alexander W. R. Nelson[1], Pushmeet Kohli[1], Paula Mori-Sánchez[3], Demis Hassabis[1], Aron J. Cohen[1,4*]

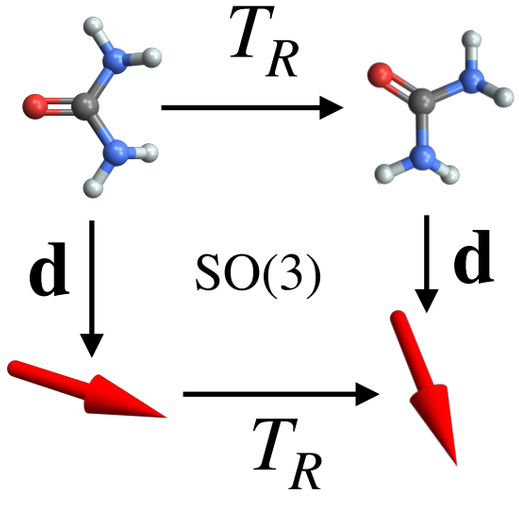
Density functional theory describes matter at the quantum level, bu
suffer from systematic errors that arise from the violation of mathe
functional. We overcame this fundamental limitation by training a n
data and on fictitious systems with fractional charge and spin. The
(DeepMind 21), correctly describes typical examples of artificial cha
correlation and performs better than traditional functionals on thor
atoms and molecules. DM21 accurately models complex systems su
charged DNA base pairs, and diradical transition states. More cruci
methodology relies on data and constraints, which are continually i
viable pathway toward the exact universal functional.

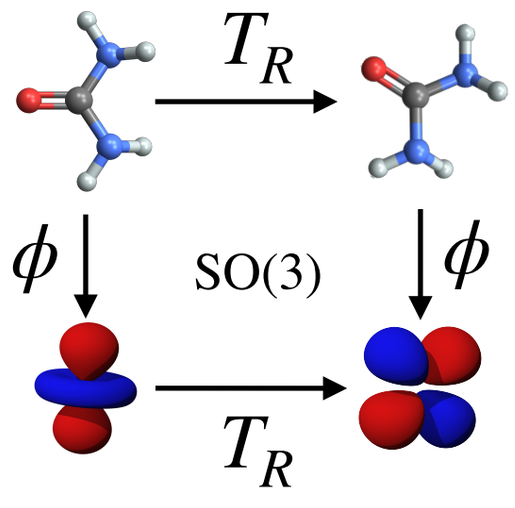
code [PySCF (*19*)]. The functional was evaluated by integrating local energies computed by a multilayer perceptron (MLP), which took as input both local and nonlocal features of the occupied Kohn-Sham (KS) orbitals, and can be described as a local range-separated hybrid. To train the functional, the sum of two objective functions was used: a regression loss for learning the exchange-correlation energy itself and a gradient regularization term that ensured that the functional derivatives

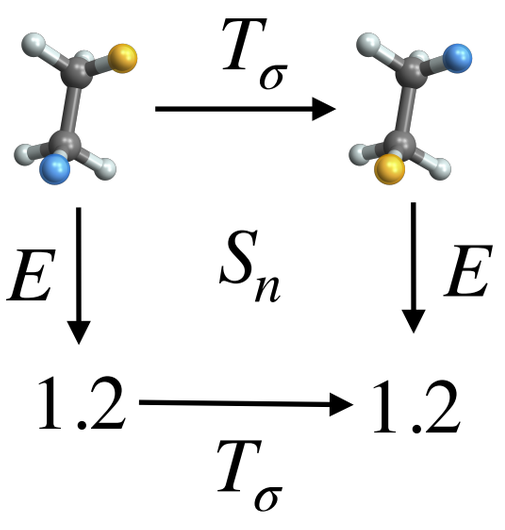


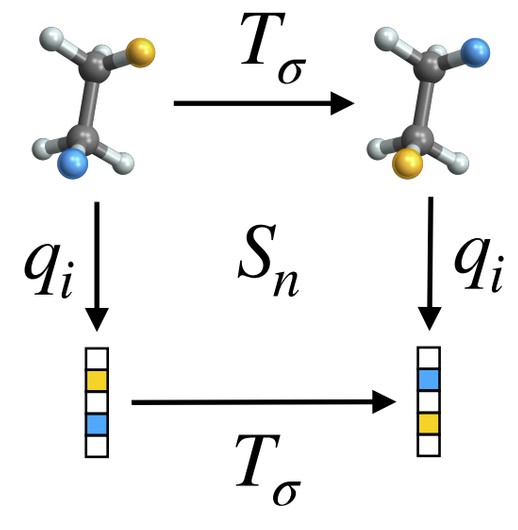
Symmetry and equivariance

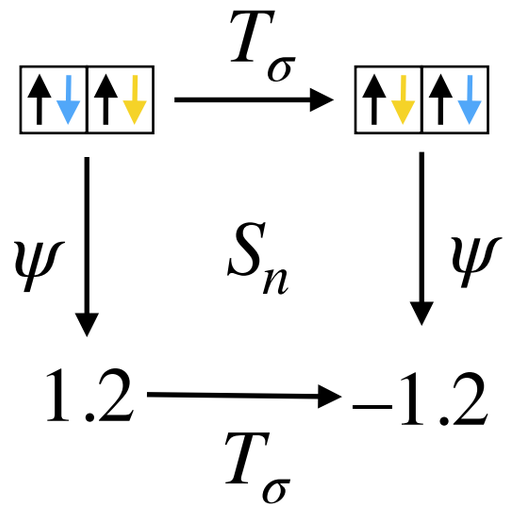












Symmetry and equivariance

- Symmetry group – set (group) of transformations g leaving an object x unchanged (invariant)
- Equivariant map (function) – map f from a symmetric object to another object that preserves symmetry

$$f(T_g x) = T_g f(x)$$

- If objects are (lists of) numbers (scalars, vectors, tensors), the transformations are encoded in irreducible representations

$$\mathbf{f}\left(\mathbf{D}_\mu(g)\mathbf{x}\right) = \mathbf{D}_\nu(g)\mathbf{f}(\mathbf{x})$$

- Invariance – special case of equivariance for the trivial representation

$$\mathbf{f}(\mathbf{D}_\mu(g)\mathbf{x}) = \mathbf{f}(\mathbf{x})$$

Symmetries and equivariance in machine learning

- Without explicit treatment, ML models are not equivariant
- Poor man's equivariance – data augmentation
- Invariance easy through invariant descriptors

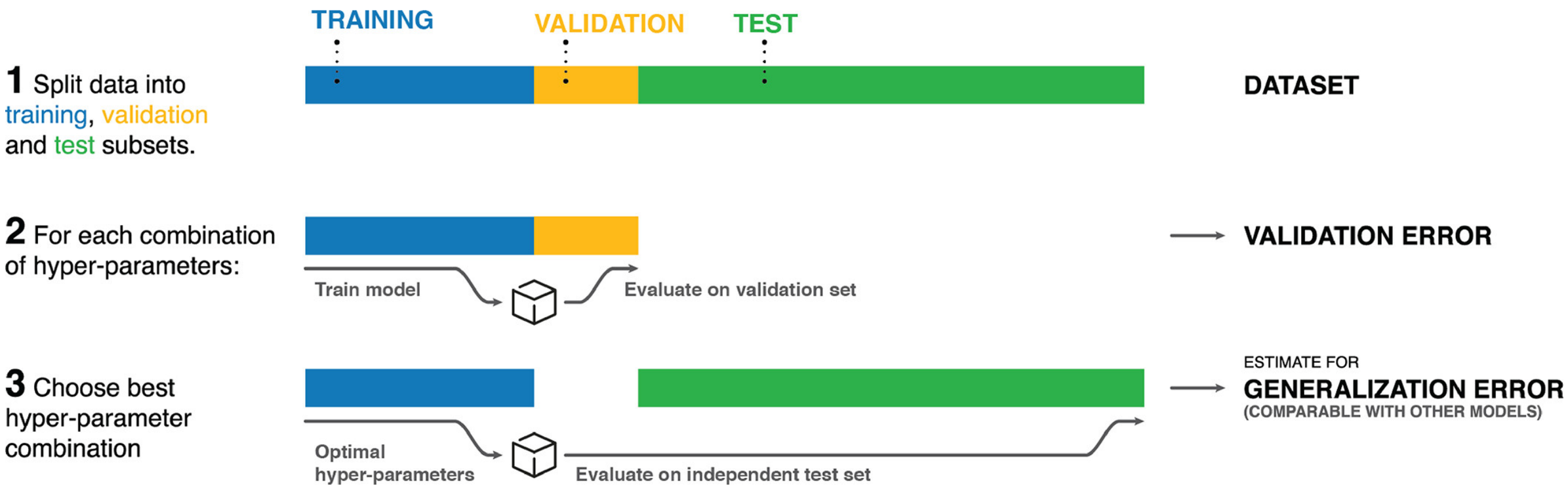
$$f_{\boldsymbol{\theta}}\left(\boldsymbol{\xi}(\mathbf{D}_{\mu}(g)\mathbf{x})\right) = f_{\boldsymbol{\theta}}\left(\boldsymbol{\xi}(\mathbf{x})\right)$$

- General equivariance must be built into the ML model
- Examples of symmetry preserving operations

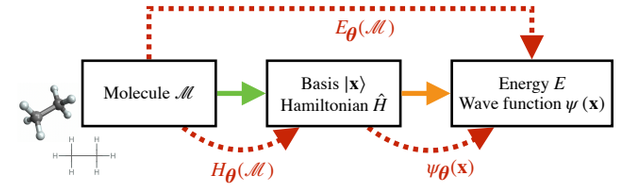
$$\mathbf{v} \to f(|\mathbf{v}|)\hat{\mathbf{v}} \qquad x_i \to g(x_i)\sum_j h(x_j)$$

Machine learning in practice

- Parameters vs. hyperparameters
- Dataset splitting

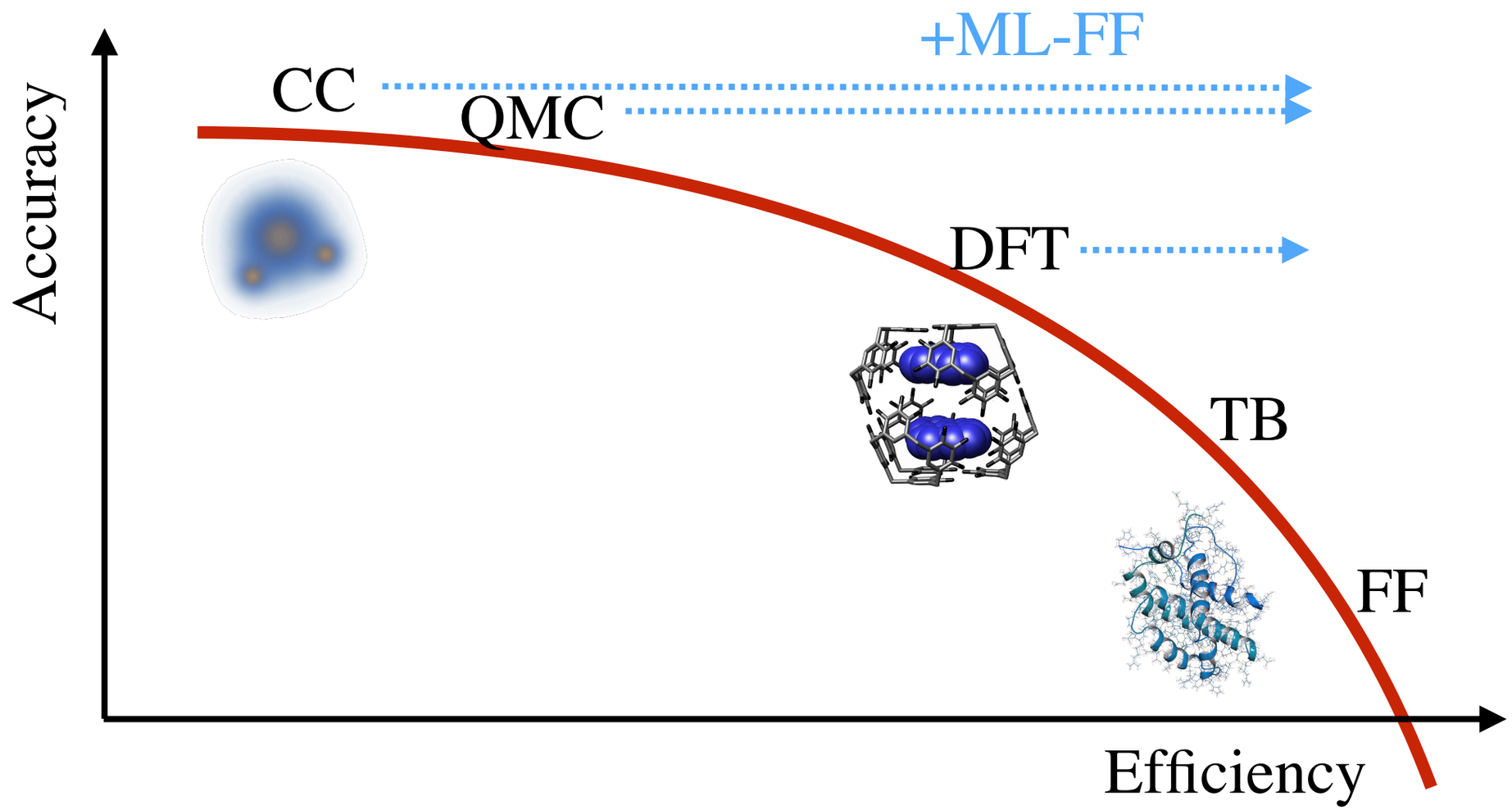


Unke et al. Machine learning force fields. *Chem. Rev.* **121**, 10142 (2021)



Machine-learning force fields

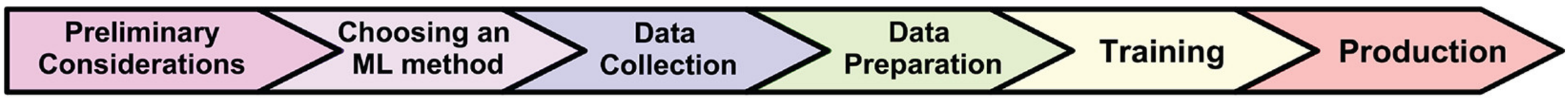
- Circumvent high cost of electronic-structure methods via a surrogate force field model for molecular dynamics



Unke et al. Machine learning force fields. *Chem. Rev.* **121**, 10142 (2021)

Machine-learning force fields

- Given a few thousand reference structures, energies, and forces, train $E_{\boldsymbol{\theta}}(\{Z_i, \mathbf{R}_i\})$
- Training set needs to cover what's sampled during MD
  - interpolation vs. extrapolation?
    - active learning, uncertainty, ensembles
- Cost of ab-initio MD vs dataset preparation, training, and ML MD



Unke et al. Machine learning force fields. *Chem. Rev.* **121**, 10142 (2021)

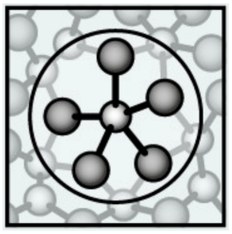
$$\mathbf{x}_i = (Z_i, \mathbf{R}_i)$$

Machine-learning force-field models

- Traditional many-body expansion

$$E\big(\{\mathbf{x}_i\}\big) = \sum_i E^{(1)}(\mathbf{x}_i) + \frac{1}{2}\sum_{ij} E^{(2)}(\{\mathbf{x}_i, \mathbf{x}_j\}) + \frac{1}{6}\sum_{ijk} E^{(3)}(\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}) + \ldots$$

- Semi-local atomic energy expansion



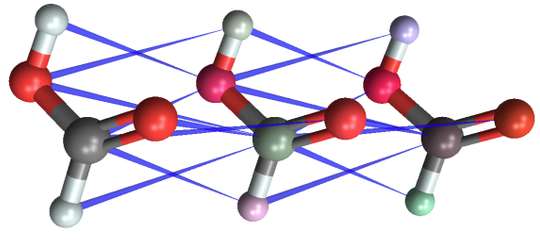
$$E\big(\{\mathbf{x}_i\}\big) = \sum_i \varepsilon\big(\mathbf{x}_i, \{\mathbf{x}_j\}_{j\in\mathcal{N}_i}\big) = \sum_i f_{\boldsymbol{\theta}}\big(\boldsymbol{\xi}(\mathbf{x}_i, \{\mathbf{x}_j\}_{j\in\mathcal{N}_i})\big)$$

- Many-body expansion of local environment

$$\boldsymbol{\xi}(\mathbf{x}_i, \{\mathbf{x}_j\}_{j\in\mathcal{N}_i}) = \big[\boldsymbol{\xi}^{(1)}(\mathbf{x}_i), \{\boldsymbol{\xi}^{(2)}(\mathbf{x}_i, \mathbf{x}_j)\}_{j\in\mathcal{N}_i}, \{\boldsymbol{\xi}^{(3)}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)\}_{j,k\in\mathcal{N}_i}, \cdots\big]$$

- Local environment descriptors from message passing

$$\boldsymbol{\xi}_i^{(n+1)} = g_{\boldsymbol{\theta}}\big(\boldsymbol{\xi}_i^{(n)}, \{\boldsymbol{\xi}_j^{(n)}\}_{j\in\mathcal{N}_i}\big)$$



$$\mathbf{x}_i = (Z_i, \mathbf{R}_i)$$

Radial and angular basis for atomic environments

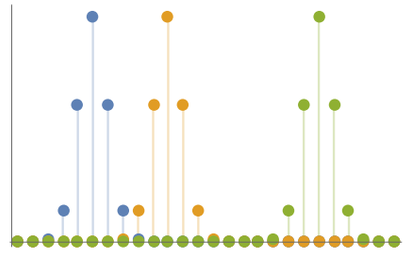
the only rotationally invariant one-atom descriptor

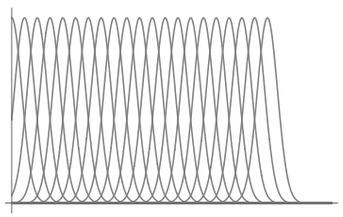
$$\boldsymbol{\xi}^{(1)}(\mathbf{x}_i) = Z_i$$

$$\boldsymbol{\xi}^{(1)}(\mathbf{x}_i) = \left(f_z(Z_i)\right)$$

$Z_i = 4$ 

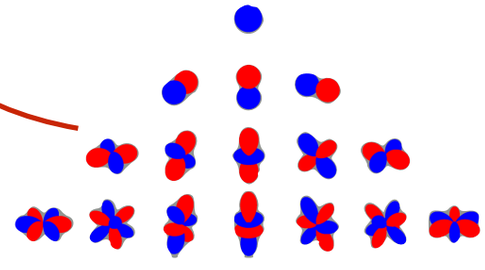
"one-hot encoding"





$$\boldsymbol{\xi}^{(2)}(\mathbf{x}_i, \mathbf{x}_j) = [Z_i, Z_j, \mathbf{R}_i - \mathbf{R}_j]$$

$$\boldsymbol{\xi}^{(2)}(\mathbf{x}_i, \mathbf{x}_j) = \left[\left(f_z(Z_i)\right), \left(f_z(Z_j)\right), \left(\varphi_n(R_{ij})\right), \left(Y_{lm}(\hat{\mathbf{R}}_{ij})\right]\right]$$

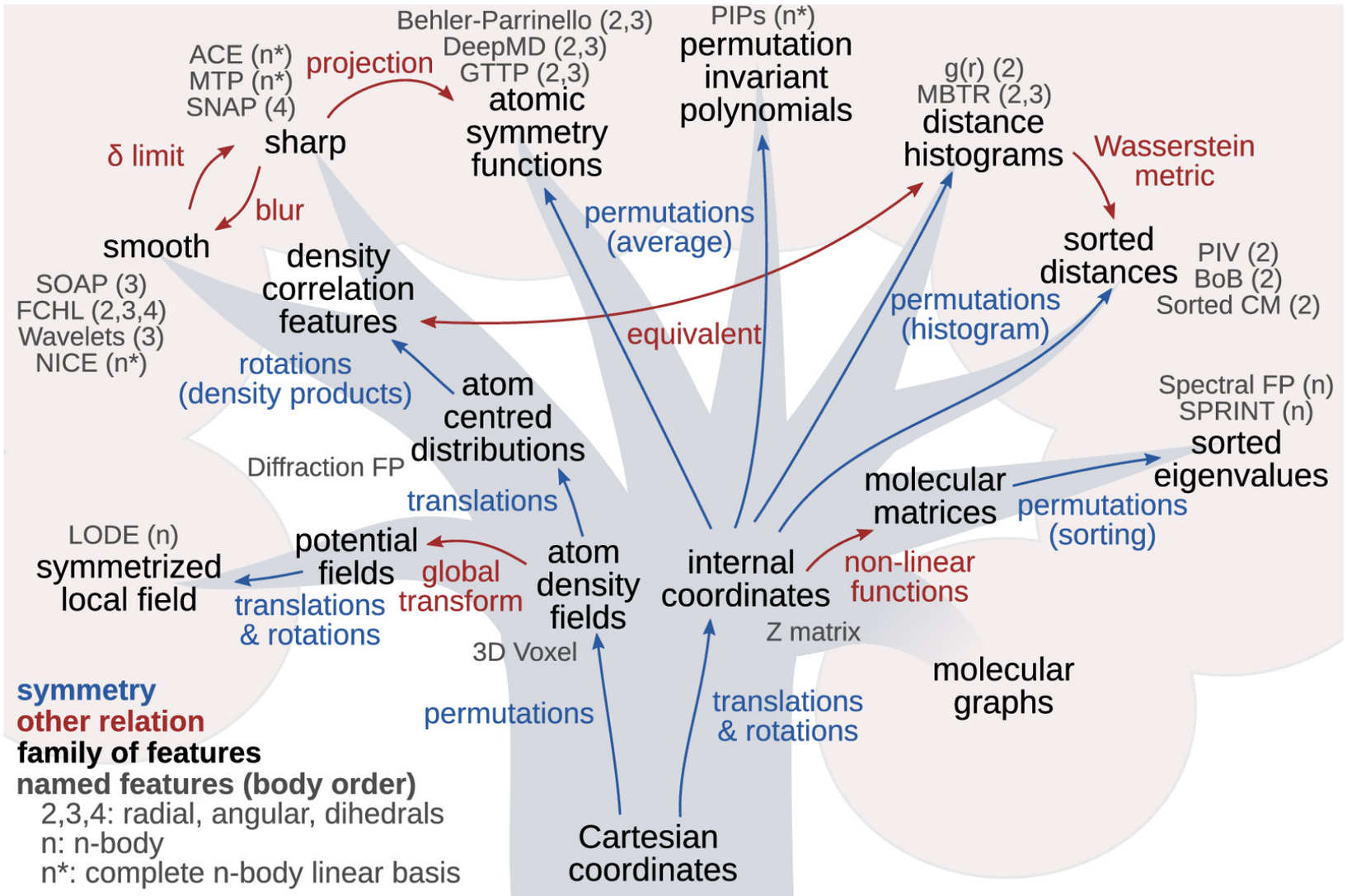


$$\boldsymbol{\xi}^{(3)}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = [Z_i, Z_j, Z_k, R_{ij}, R_{ik}, \theta_{jik}]$$

$$\boldsymbol{\xi}^{(3)}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = \left[\ldots, \left(\varphi_n(R_{ij})\varphi_{n'}(R_{ik})\right), \left(Y_{lm}(\hat{\mathbf{R}}_{ij})Y_{l'm'}(\hat{\mathbf{R}}_{ik})\right)\right]$$

Zoo of atomic descriptors

- Good news – convergence towards a single framework



Musil et al. Physics-Inspired Structural Representations. *Chem. Rev.* **121**, 9759 (2021)

Simplified classification of ML force fields

	Fixed descriptor	Message-passing descriptor
Gaussian process	**SOAP**, FCHL,…	
Neural network	**Behler–Parrinello**, DeepMD, ANI,…	**SchNet**, PhysNet, NequIP, PaiNN, DimeNet, NewtonNet,…

Example 1: SOAP–GAP

Article

Origins of structural and electronic transitions in disordered silicon

https://doi.org/10.1038/s41586-020-03072-z

Received: 14 December 2019

Accepted: 12 November 2020

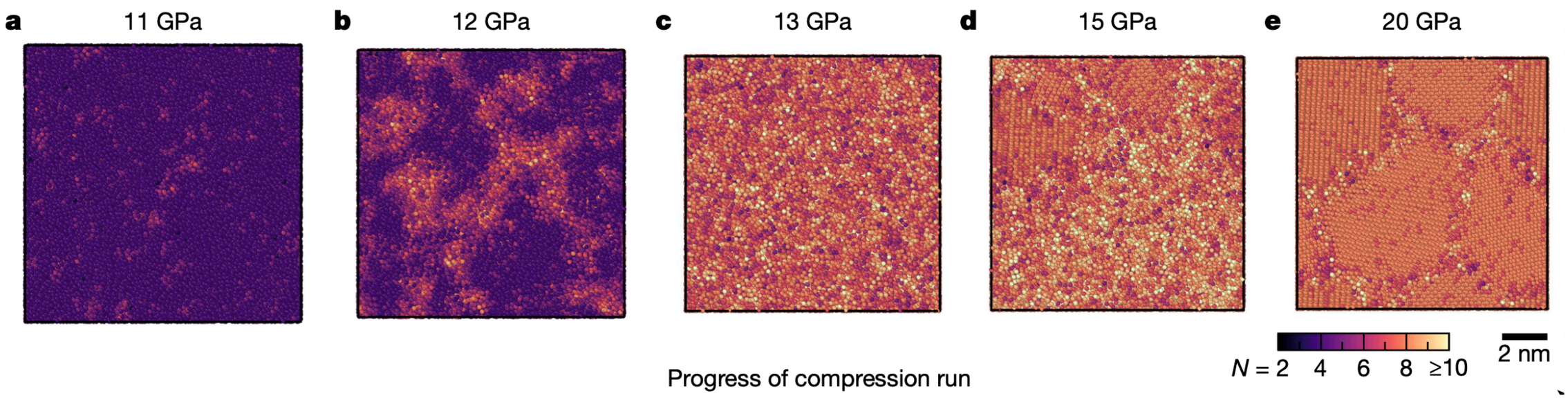
Published online: 6 January 2021

Check for updates

Volker L. Deringer[1]✉, Noam Bernstein[2], Gábor Csányi[3], Chiheb Ben Mahmoud[4,5], Michele Ceriotti[4,5], Mark Wilson[6], David A. Drabold[7] & Stephen R. Elliott[8,9]

Structurally disordered materials pose fundamental questions[1–4], including how different disordered phases ('polyamorphs') can coexist and transform from one phase to another[5–9]. Amorphous silicon has been extensively studied; it forms a

- Molecular dynamics: 100,000 atoms, 10^3-nm cell, 10 ps
- Training: PW91, <200 atoms per uc, 2,500 structures, 170,000 atomic environments



Example 2: Behler–Parrinello

PNAS

Ab initio thermodynamics of liquid and solid water

Bingqing Cheng[a,1], Edgar A. Engel[a], Jörg Behler[b,c], Christoph Dellago[d], and Michele Ceriotti[a]

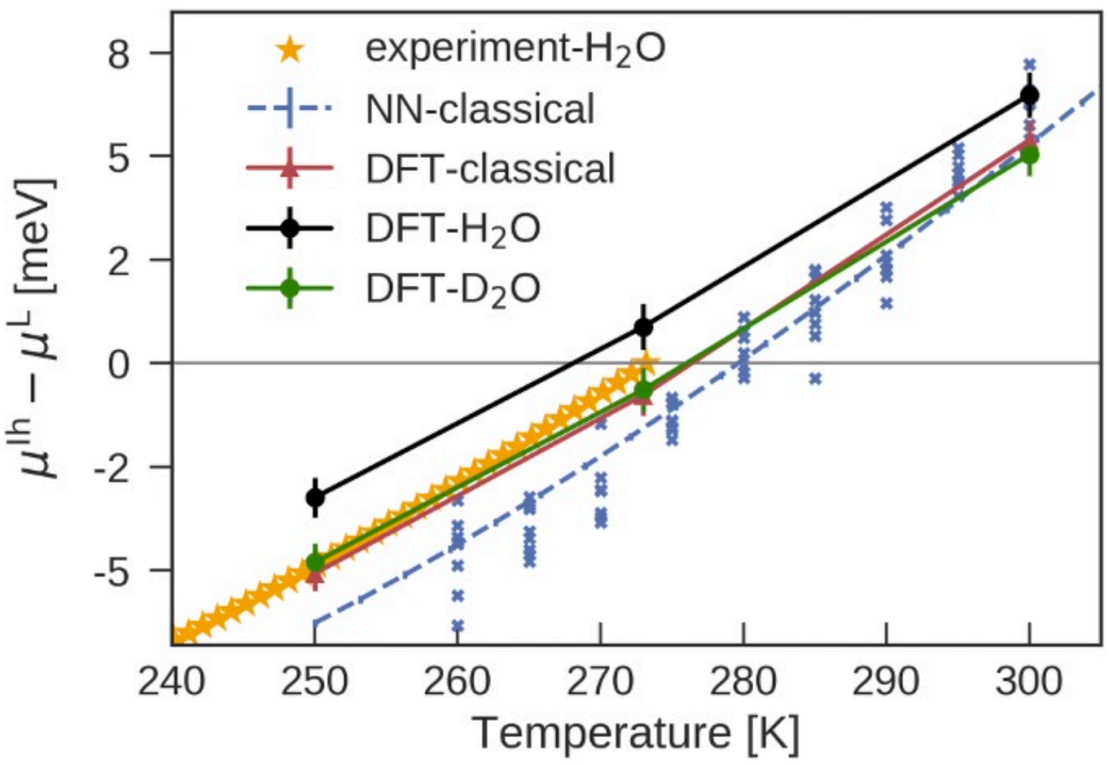
[a]Laboratory of Computational Science and Modeling, Institute of Materials, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland; [b]Universität Göttingen, Institut für Physikalische Chemie, Theoretische Chemie, 37077 Göttingen, Germany; [c]International Center for Advanced Studies of Energy Conversion, Universität Göttingen, 37073 Göttingen, Germany; and [d]Faculty of Physics, University of Vienna, 1090 Vienna, Austria

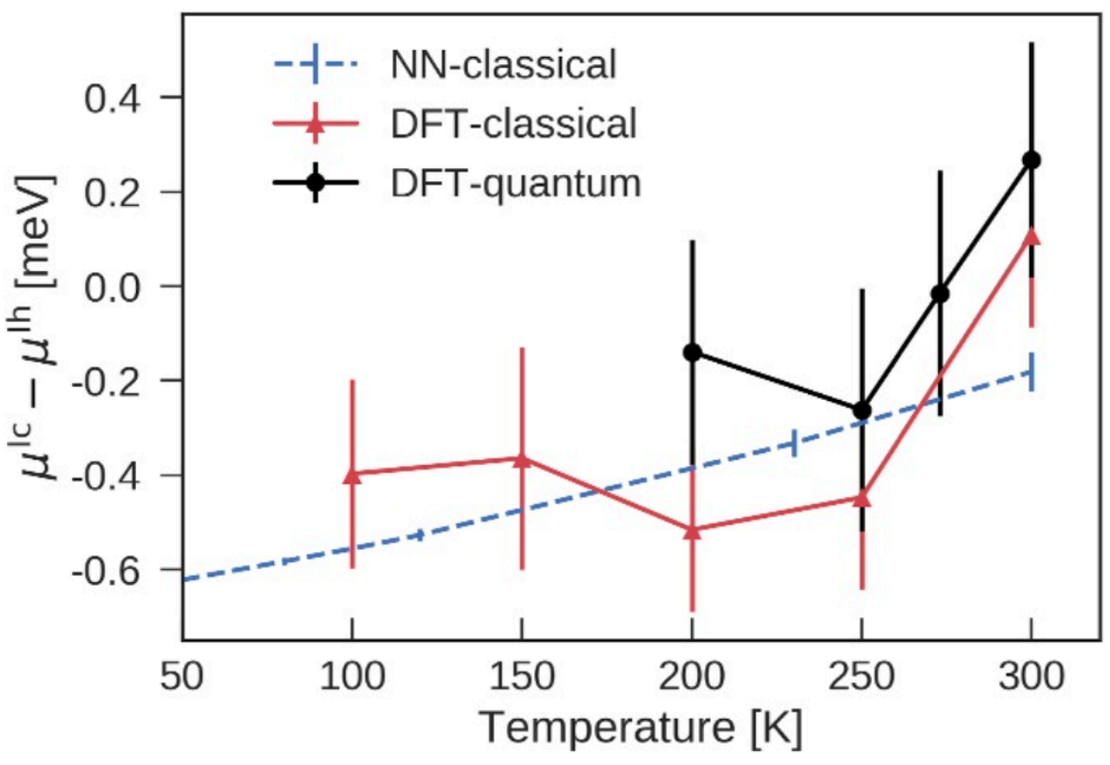
Edited by Pablo G. Debenedetti, Princeton University, Princeton, NJ, and approved December 3, 2018 (received for review September 4, 2018)

Thermodynamic properties of liquid water as well as hexagonal (Ih) and cubic (Ic) ice are predicted based on density functional theory at the hybrid-functional level, rigorously taking into account quantum nuclear motion, anharmonic fluctuations, and

machine-learning (ML) techniques to avoid the prohibitively large computational expenses otherwise incurred by extensively sampling phase space by using first-principles methods. In particular, we use sophisticated thermodynamic integration (TI)

- Molecular dynamics: 800 molecules, nuclear quantum effects
- Training: revPBE0-D3, 64 molecules, 1,600 structures





Example 3: SchNet

Roaming leads to amino acid photodamage: A deep learning study of tyrosine

Julia Westermayr,[1] Michael Gastegger,[2] Dora Vörös,[3] Lisa Panzenboeck,[3,4] Florian Joerg,[3,5] Leticia González,[3,6] and Philipp Marquetand[3,6,7,a)]

[1)] *University of Warwick, Department of Chemistry, Gibbet Hill Rd, Coventry, CV4 7AL, UK*

[2)] *Technical University of Berlin, Machine Learning Group, 10587 Berlin, Germany*

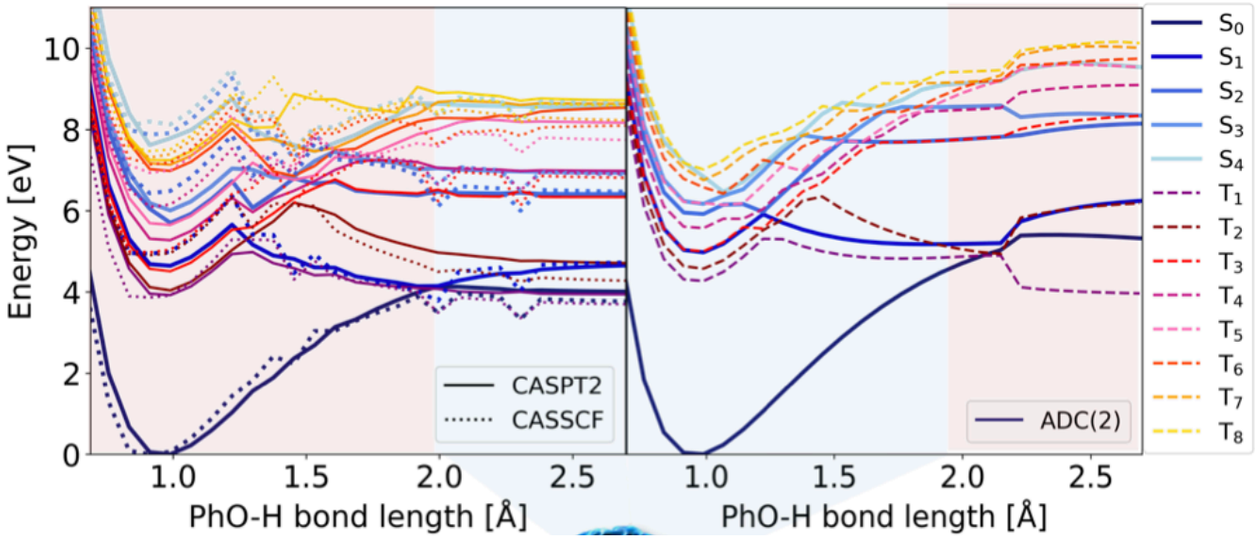
[3)] *University of Vienna, Faculty of Chemistry, Institute of Theoretical Chemistry, Währinger Str. 17, 1090 Vienna, Austria.*

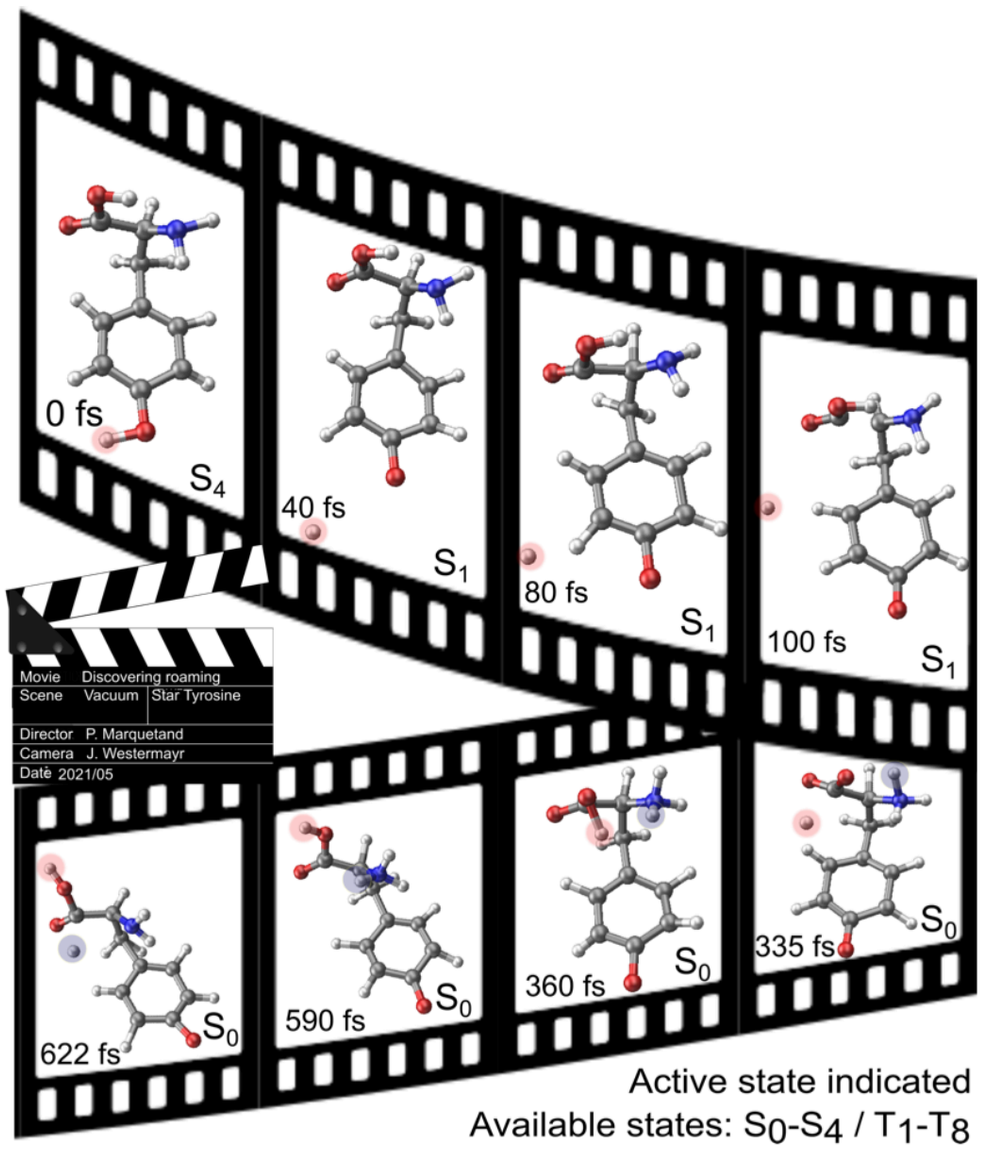
[4)] *Present address: University of Vienna, Faculty of Chemistry, Department of ... 1090 Vienna, Austria.*

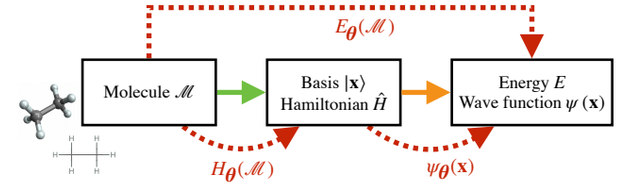
[5)] *Present address: University of Vienna, Faculty of Chemistry, Institute of Co... 1090 Vienna, Austria*

nature chemistry

- Molecular dynamics: 1000 1ps trajectories, 29 electronic states
- Training: CASPT2(12e,11o), 15,000 structures





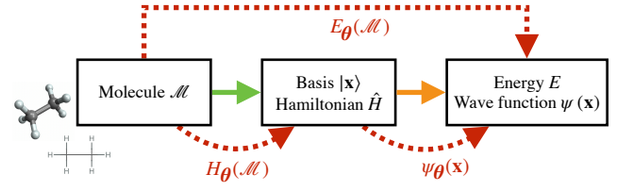


Electronic Schrödinger equation

$$\hat{H}\psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) = E\psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$$

$$\hat{H} := \sum_{i=1}^{N} \underbrace{\left(-\frac{1}{2}\nabla_{\mathbf{r}_i}^2 - \sum_I \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \right)}_{\hat{h}_i} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

- Electron coordinates: $(\mathbf{r}_1, \ldots, \mathbf{r}_N) \equiv \mathfrak{r}$
- Molecule specified by atom charges Z_I and coordinates $\mathbf{R}_I$
- Solution: eigenstates ψ_n and their energies E_n, including the ground state ψ_0, E_0
- Variational principle: $E_0 = \min_\psi \int \mathrm{d}\mathfrak{r}\, \psi(\mathfrak{r}) \hat{H} \psi(\mathfrak{r})$
- Analytically solvable only for hydrogen atom, but many approximate numerical methods from quantum chemistry



Machine-learned density functionals

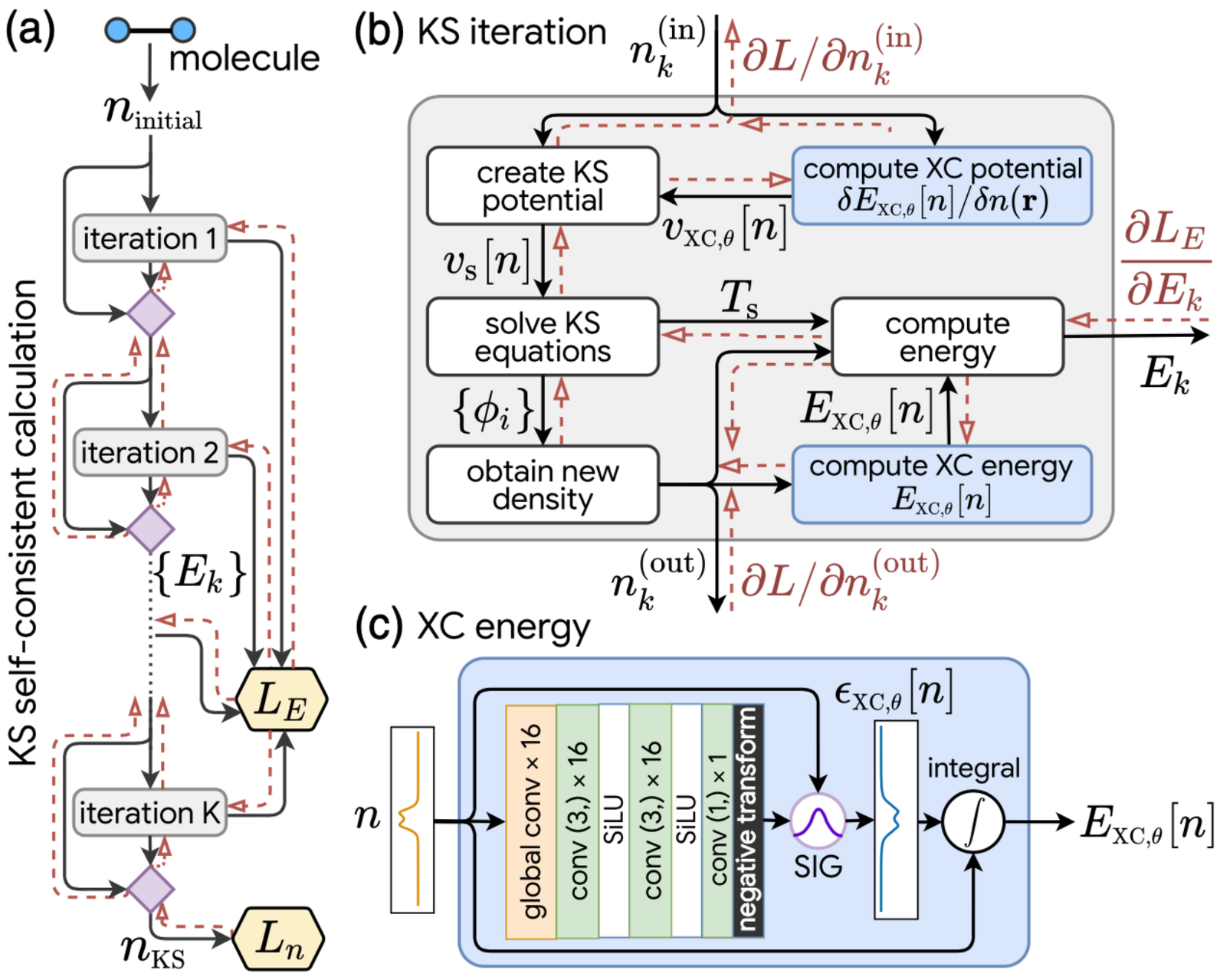
- Density-functional theory (DFT) exact in principle, but only approximate functionals known in practice

$$\hat{H} := \sum_i \left(\hat{h}_i + v_{\text{eff}}[n](\mathbf{r}_i) \right), \quad n(\mathbf{r}) = \sum_i \phi_i(\mathbf{r})^2$$

$$v_{\text{eff}}[n](\mathbf{r}) = \int \mathrm{d}\mathbf{r}' \left(\frac{n(\mathbf{r})}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{\text{xc}}[n]}{\delta n(\mathbf{r})} \right), \quad E_{\text{xc}}[n] \approx \int \mathrm{d}\mathbf{r}\, \epsilon_{\text{xc}}\big(n(r)\big)$$

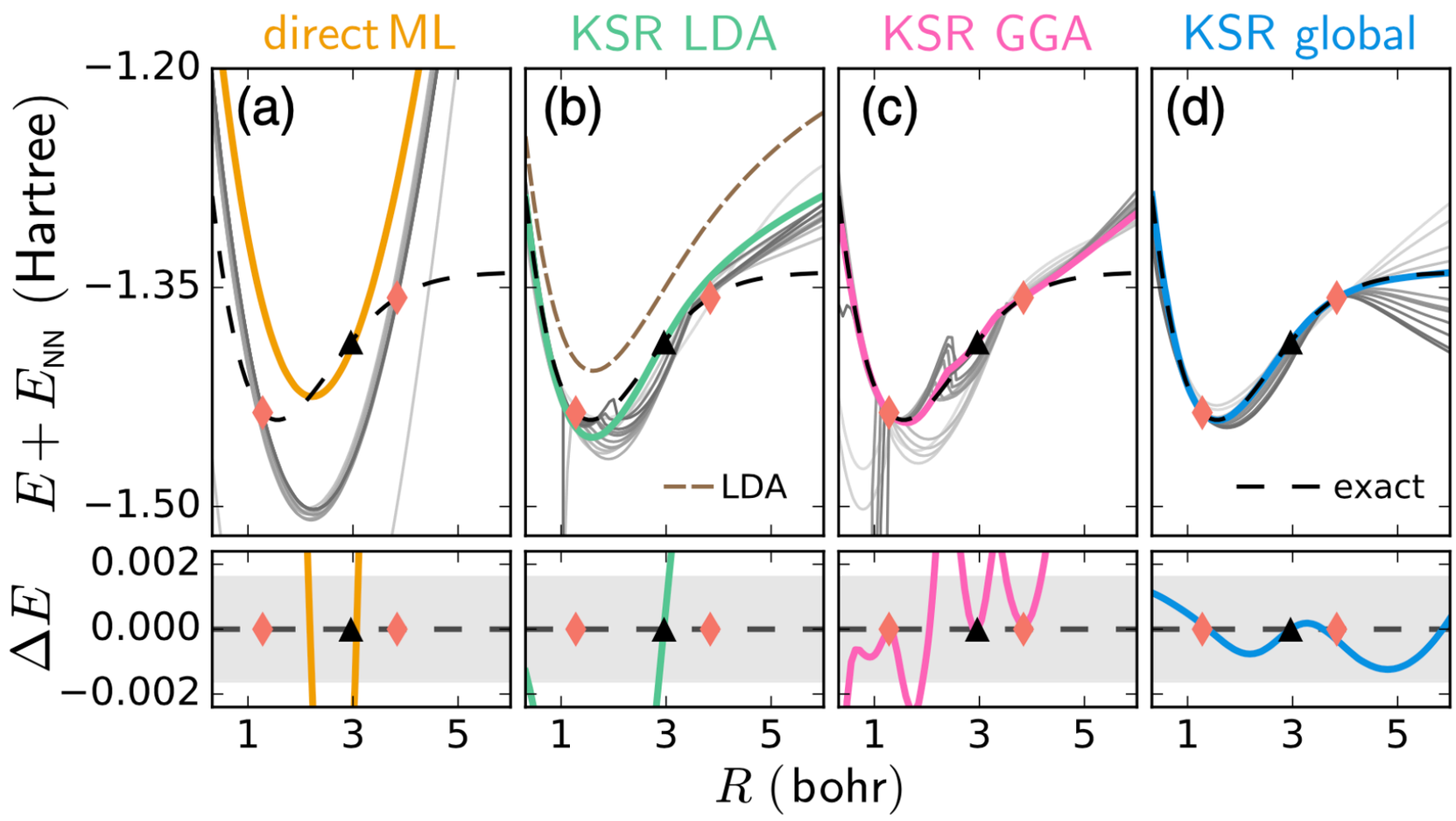
  - Given reference energies, learn functional? ❌
  - DFT calculation involves minimization w.r.t. energy
  - Given reference energies and densities, learn functional ✅

End-to-end learning with autodifferentiation



Li et al. Kohn-Sham equations as regularizer. *PRL* **126**, 036401 (2021)

Incorporating physics improves generalization



- Only two points needed to fit dissociation of H_2

Li et al. Kohn-Sham equations as regularizer. *PRL* **126**, 036401 (2021)

Example: DM21

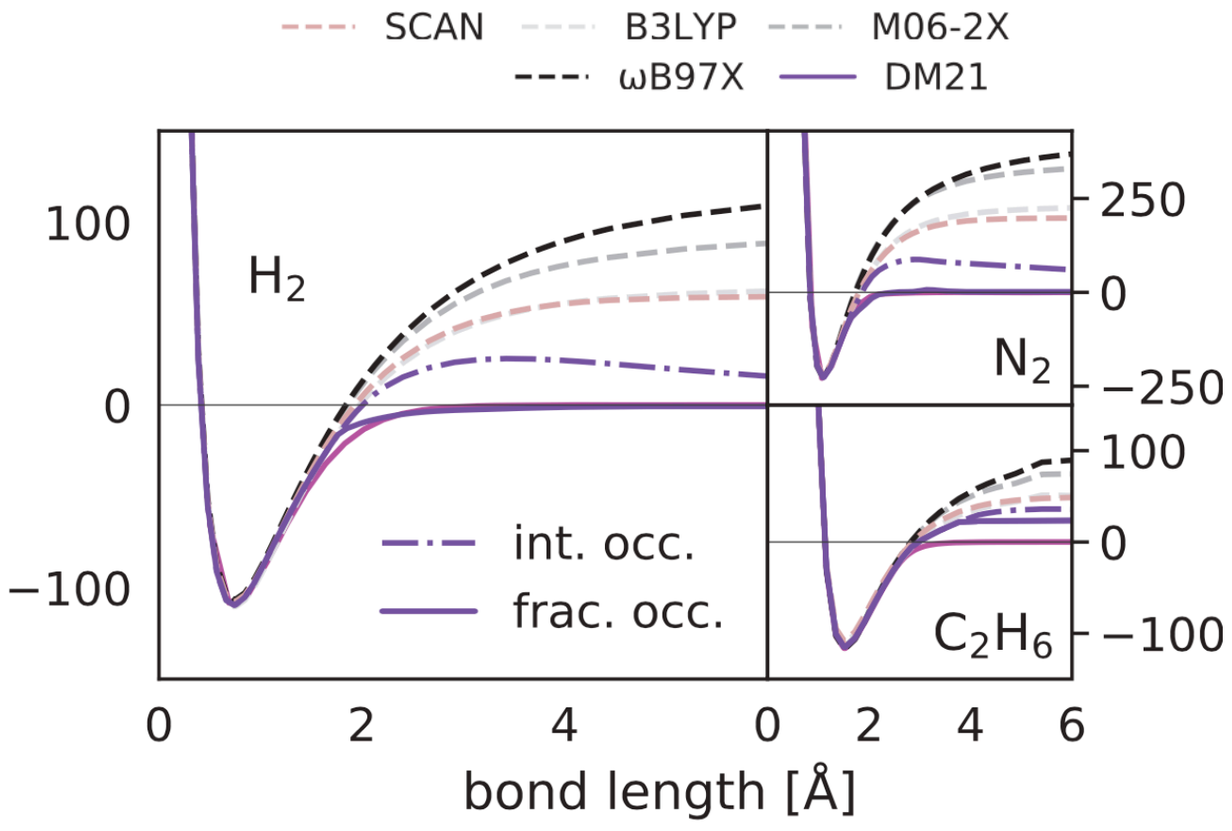
QUANTUM CHEMISTRY

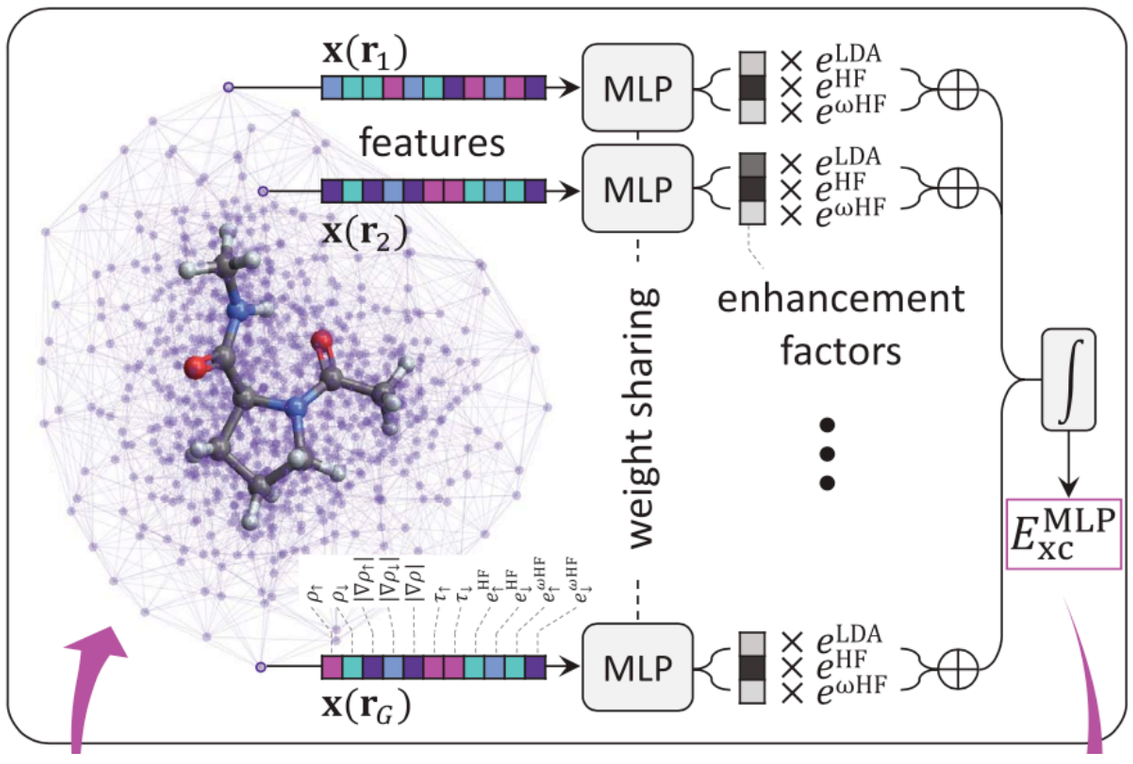
Pushing the frontiers of density functionals by solving the fractional electron problem

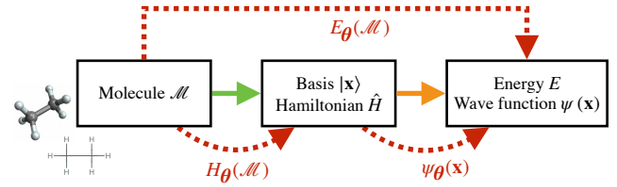
James Kirkpatrick[1]*†, Brendan McMorrow[1]†, David H. P. Turban[1]†, Alexander L. Gaunt[1]†, James S. Spencer[1], Alexander G. D. G. Matthews[1], Annette Obika[1], Louis Thiry[2], Meire Fortunato[1], David Pfau[1], Lara Román Castellanos[1], Stig Petersen[1], Alexander W. R. Nelson[1], Pushmeet Kohli[1], Paula Mori-Sánchez[3], Demis Hassabis[1], Aron J. Cohen[1,4]*

code [PySCF (*19*)]. The functional was evaluated by integrating local energies computed by a multilayer perceptron (MLP), which took as input both local and nonlocal features of the occupied Kohn-Sham (KS) orbitals, and can be described as a local range-separated hybrid. To train the functional, the sum of two objective functions was used: a regression loss for learning the exchange-correlation energy itself and a gradient regularization term that ensured that the functional derivatives

- DM21: Range-separated meta-GGA hybrid, comparable in accuracy and training to state-of-the-art functionals
- Exact constraints through data augmentation







Machine-learned wave functions

Chem. Listy *112*, 640–647 (2018)

JE STROJOVÉ UČENÍ BUDOUCNOSTÍ TEORETICKÉ CHEMIE?

KAREL BERKA[a], ŠTĚPÁN SRŠEŇ[b] a PETR SLAVÍČEK[b,c]

Numerické algoritmy strojového učení ale mohou změnit přímo i *ab initio* kvantovou chemii. Přesné výpočty jsou zde komplikovány korelacemi mezi pohyby jednotlivých částic, což vede k exponenciálně se zvyšujícím nárokům s rostoucí velikostí systému. Techniky strojového učení aplikované na samotnou vlnovou funkci mohou být v některých případech schopny výrazně výpočty urychlit[48].

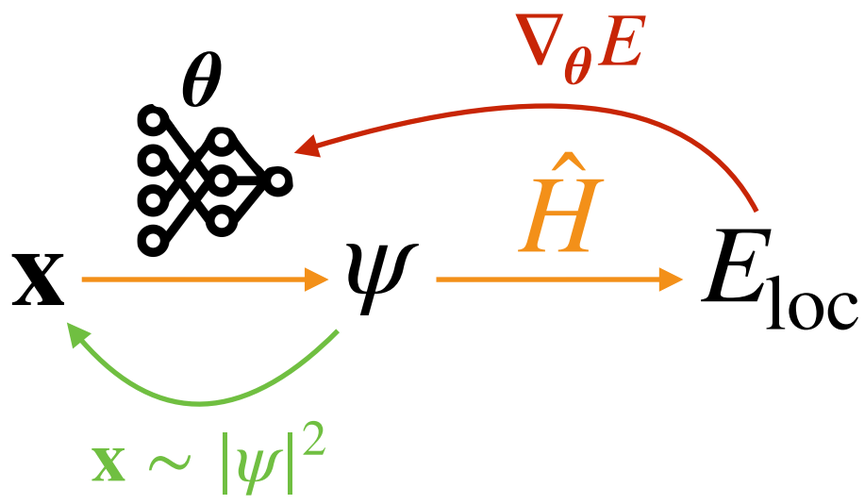
[48]Carleo & Troyer, *Science* **355**, 602 (2017)

Variational quantum Monte Carlo

Schrödinger equation	$\hat{H}\,\lvert\psi\rangle = E\,\lvert\psi\rangle$
Basis states, wave function	$\lvert\psi\rangle = \sum_{\mathbf{x}} \lvert\mathbf{x}\rangle\langle\mathbf{x}\rvert\psi\rangle = \sum_{\mathbf{x}} \psi(\mathbf{x})\,\lvert\psi\rangle$
Monte Carlo integration	$E[\psi] = \dfrac{\langle\psi\rvert\hat{H}\lvert\psi\rangle}{\langle\psi\rvert\psi\rangle} = \mathbb{E}_{\mathbf{x}\sim\lvert\psi(\mathbf{x})\rvert^2}[E_{\mathrm{loc}}(\mathbf{x})]$
Local energy	$E_{\mathrm{loc}}(\mathbf{x}) = \sum_{\mathbf{x}'} \langle\mathbf{x}\rvert\hat{H}\rvert\mathbf{x}'\rangle \dfrac{\psi(\mathbf{x}')}{\psi(\mathbf{x})}$ (zero variance)
Variational principle	$E = \min_{\psi} E[\psi] \le \min_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}]$
Gradient descent	$\nabla_{\boldsymbol{\theta}} E[\psi_{\boldsymbol{\theta}}] = \mathbb{E}_{\mathbf{x}}\left[\left(E_{\mathrm{loc}}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}'}[E_{\mathrm{loc}}(\mathbf{x}')]\right)\nabla_{\boldsymbol{\theta}} \ln\lvert\psi_{\boldsymbol{\theta}}(\mathbf{x})\rvert\right]$
Markov-chain sampling	$\mathbf{x}' \sim \exp(-\lVert\mathbf{x}-\mathbf{x}'\rVert/\tau), \quad P(\mathbf{x} := \mathbf{x}') = \lvert\psi(\mathbf{x}')\rvert^2/\lvert\psi(\mathbf{x})\rvert^2$

Deep variational quantum Monte Carlo

```
def fit_wf(
    hamil: Hamiltonian[X],
    wf: Callable[[X], Psi],
    sampler: Iterator[X],
    opt: torch.optim.Optimizer,
    steps: int,
):
    for _ in zip(steps):
        x = next(sampler)
        psi = wf(x)
        E_loc = hamil.local_energy(x, wf)
        loss = (
            (E_loc - E_loc.mean()).detach() * psi.log()
        ).mean()
        loss.backward()
        opt.step()
        opt.zero_grad()
```



First and second quantization

- Quantum states of many-particle systems must be (anti)symmetric for bosons (fermions) with respect to exchange of particles
- Two ways to treat the (anti)symmetry

$N = 3,\ K = 5$

First quantization

$$|\psi\rangle = \sum_{\mathbf{x}_1 \cdots \mathbf{x}_N} \psi(\mathbf{x}_1, \ldots, \mathbf{x}_N)\, |\mathbf{x}_1 \cdots \mathbf{x}_N\rangle$$

coordinates $\mathbf{x}_i$

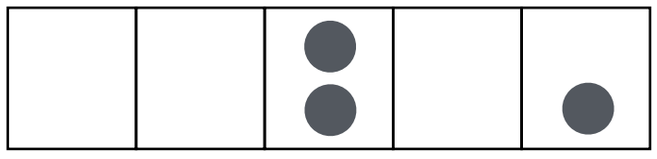
$\mathbf{x} = (2.7, 3.1, 4.9)$

Second quantization (one-particle discrete basis: $\mathbf{x} \to k$)

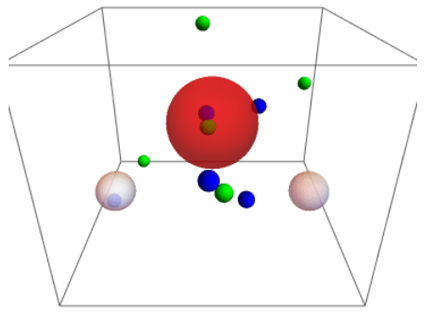
$$|\psi\rangle = \sum_{n_1 \cdots n_K} \psi(n_1, \ldots, n_K)\, |n_1 \cdots n_K\rangle$$

occupation numbers n_i

$\mathbf{n} = (0, 0, 0, 2, 0, 1)$



Electronic Schrödinger equation
(first quantization)

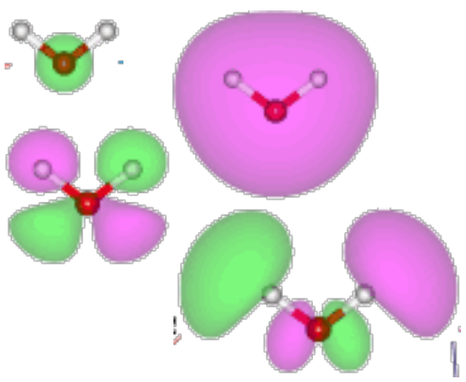


- Electron coordinates, $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$, $3N$-dimensional space
- (Asymmetric) basis states are $|\mathbf{r}\rangle$, antisymmetric wave function $\psi(\mathbf{r})$

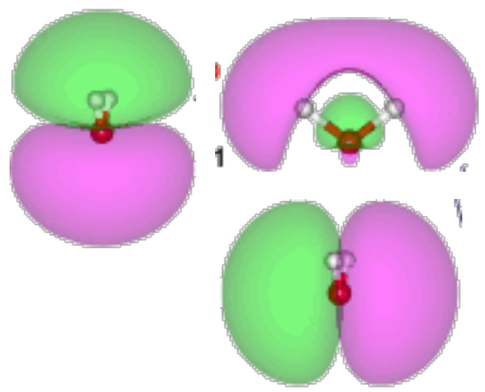
$$\hat{H} := \underbrace{-\frac{1}{2}\Delta}_{\text{kinetic}} + \underbrace{\frac{1}{2}\sum_{ij}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{electron–electron}} - \underbrace{\sum_{iI}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}}_{\text{electron–nucleus}}$$

- Molecule specified by atom charges Z_I and coordinates $\mathbf{R}_I$
- Local energy needs evaluation of Laplacian

$$E_{\text{loc}}(\mathbf{r}) = -\frac{1}{2}\frac{\Delta\psi(\mathbf{r})}{\psi(\mathbf{r})} + \frac{1}{2}\sum_{ij}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{iI}\frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}$$



Electronic Schrödinger equation
(second quantization)



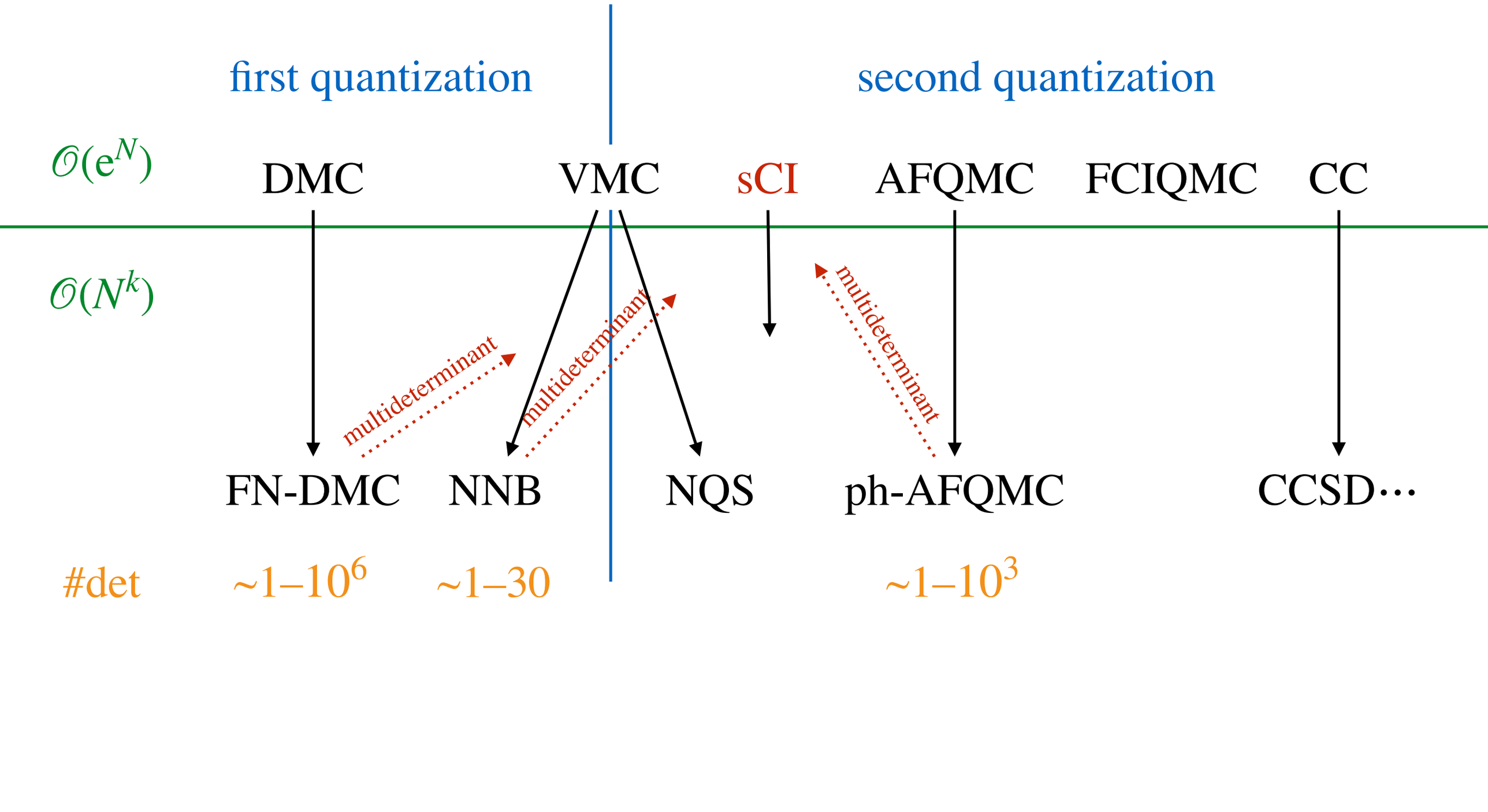
- Discrete basis: orthogonal orbitals $\varphi_k(\mathbf{r})$, $k = 1,\ldots, K$
- Occupation numbers $\mathbf{n}$, $n_k = 0, 1$, $\sum_k n_k = N$, $\mathbf{n} \equiv \{k_1, \ldots, k_N\}$
- Antisymmetric basis states are *Slater determinants*, $|\mathbf{n}\rangle = \det \varphi_{k_j}(\mathbf{r}_i)$, (asymmetric) wave function $\psi(\mathbf{n}) \equiv c_{\mathbf{n}}$
- Local energy as a sum over all Slater determinants that differ in occupation of up to two orbitals

$$E_{\mathrm{loc}}(\mathbf{n}) = \sum_{\mathbf{n}':k\to l} h_{kl}\frac{\psi(\mathbf{n}')}{\psi(\mathbf{n})} + \sum_{\mathbf{n}'':k,l\to m,n} V_{klmn}\frac{\psi(\mathbf{n}'')}{\psi(\mathbf{n})}$$

$$h_{kl} = \int \mathrm{d}\mathbf{r}\, \varphi_k(\mathbf{r})\left(-\frac{1}{2}\Delta - \sum_I \frac{Z_I}{|\mathbf{r}-\mathbf{R}_I|}\right)\varphi_l(\mathbf{r})$$

$$V_{klmn} = \frac{1}{2}\int \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'\frac{\varphi_k(\mathbf{r})\varphi_l(\mathbf{r}')\varphi_m(\mathbf{r})\varphi_n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}$$

Determinant expansion curse



Example 1: PauliNet

nature chemistry

ARTICLES

https://doi.org/10.1038/s41557-020-0544-y

Check for updates

Deep-neural-network solution of the electronic Schrödinger equation

Jan Hermann[1,2]✉, Zeno Schätzle[1] and Frank Noé[1,3,4]✉

1. Slater determinant

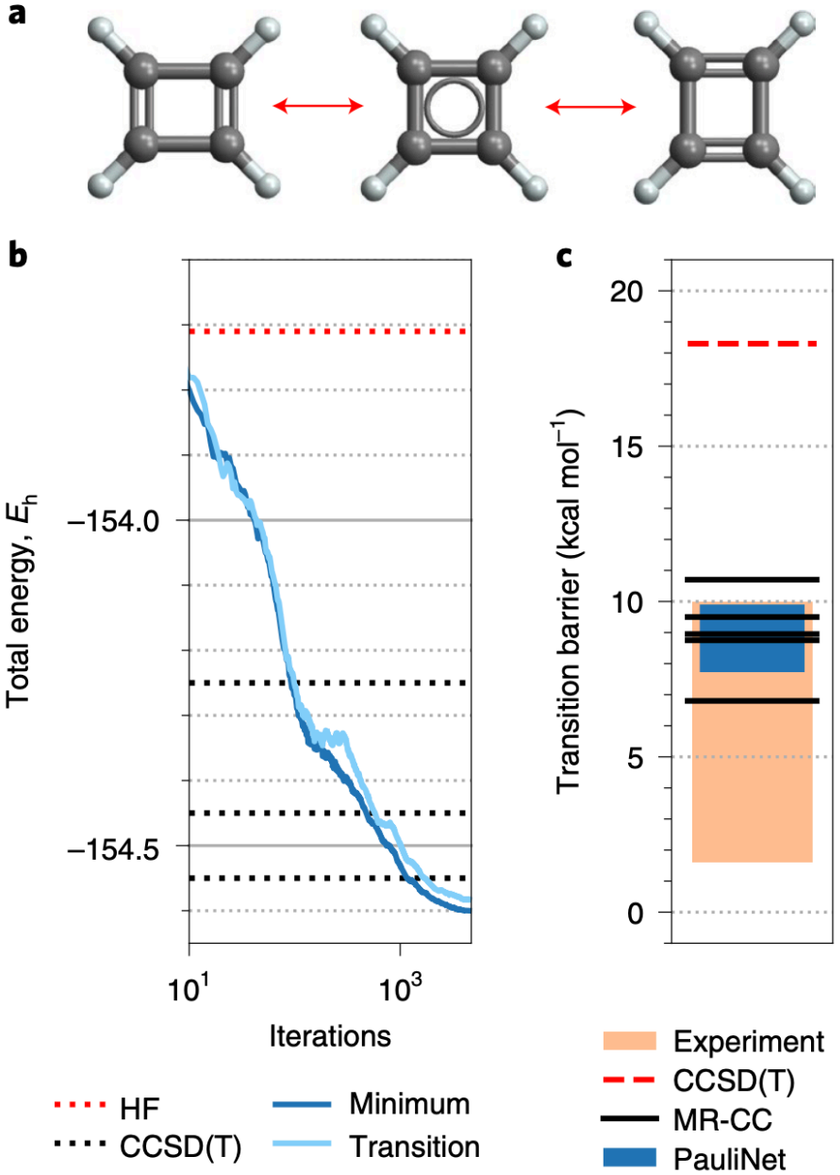
$$\psi_{\theta}(\mathbf{r}) := \det \varphi_k(\mathbf{r}_i), \qquad k = 1,\ldots,N$$

2. Generalized Slater determinant

$$\psi_{\theta}(\mathbf{r}) := \det f_{ki,\theta}(\mathbf{r}), \qquad \mathbf{f}_{\theta}(\mathscr{P}_{ij}\mathbf{r}) = \mathscr{P}_{ij}\mathbf{f}_{\theta}(\mathbf{r})$$

3. Real-space baseline/envelope

$$\psi_{\theta}(\mathbf{r}) := \det f_{ki,\theta}(\mathbf{r})\varphi_k(\mathbf{r}_i)$$



Example 2: Neural quantum states

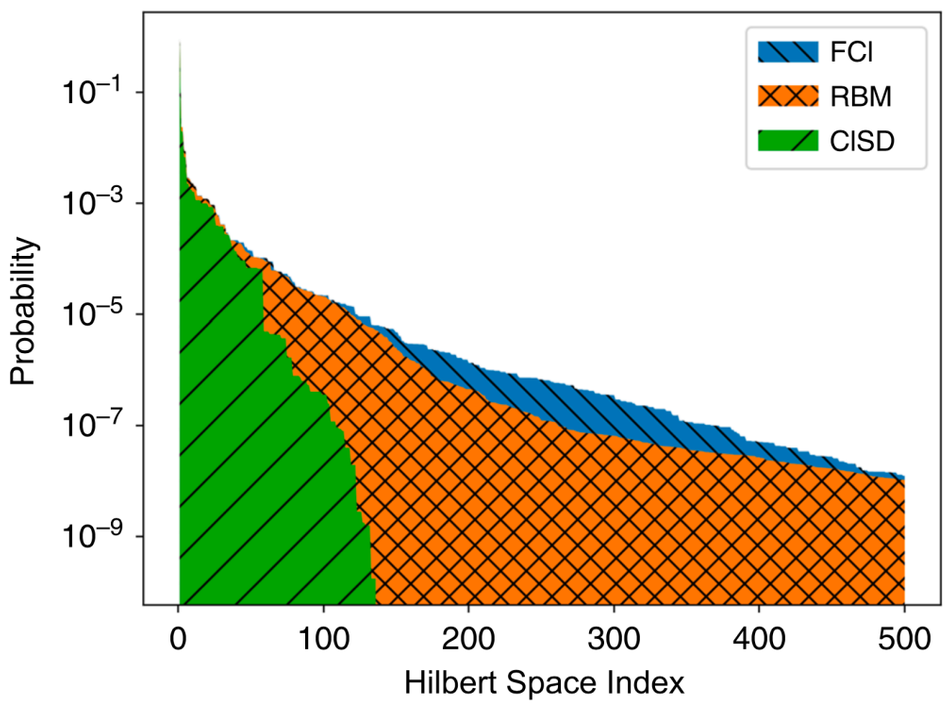
ARTICLE

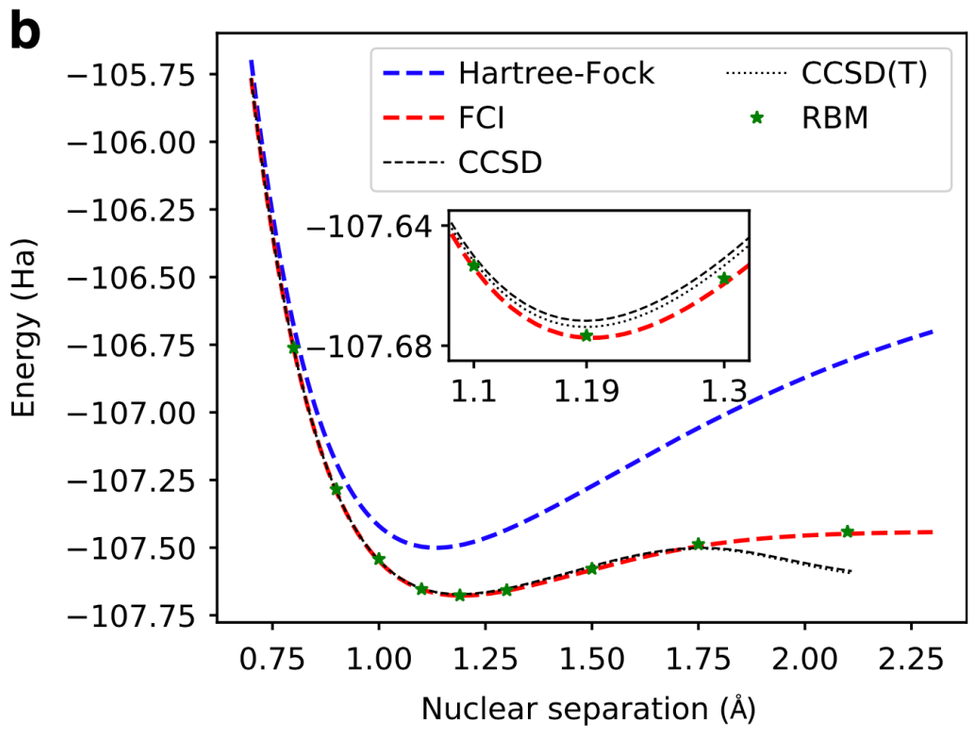
https://doi.org/10.1038/s41467-020-15724-9 OPEN

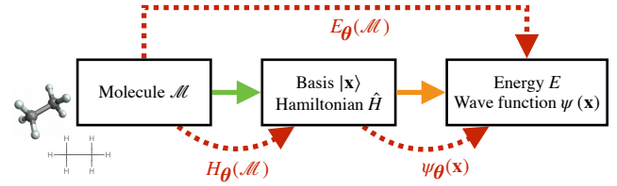
Fermionic neural-network states for ab-initio electronic structure

Kenny Choo[1]✉, Antonio Mezzacapo[2]✉ & Giuseppe Carleo[3]✉

- Instead of storing a CI vector $c_{\mathbf{n}}$ (FCI), sampling it stochastically (FCIQMC), compressing it (CC), represent it with a NN, $\psi_{\boldsymbol{\theta}}(\mathbf{n})$







Summary

- Machine learning can help avoiding as well as solving electronic structure
- Two leading ML techniques: Neural networks and Gaussian processes
- Machine-learning force fields are a mature tool ready for production calculations
- Machine-learned Hamiltonians and wave functions are a promising tool, but still in the making

SchNet for molecules

- Instance of a graph neural network
- Original SchNet is a force field

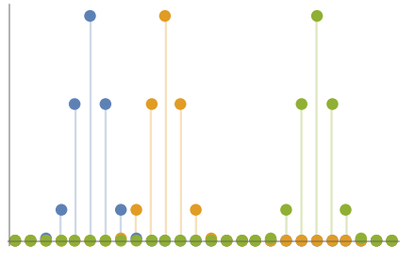
$$E\left(\{Z_i, \mathbf{R}_i\}\right)$$

$$\mathbf{x}_i^{(0)} := \mathbf{X}_{\boldsymbol{\theta}, Z_i}$$

$$\mathbf{z}_i^{(n)} := \sum_j \mathbf{w}_{\boldsymbol{\theta}}^{(n)}\left(\mathbf{e}(|\mathbf{R}_i - \mathbf{R}_j|)\right) \odot \mathbf{h}_{\boldsymbol{\theta}}^{(n)}\left(\mathbf{x}_j^{(n)}\right)$$

$$\mathbf{x}_i^{(n+1)} := \mathbf{x}_i^{(n)} + \mathbf{g}_{\boldsymbol{\theta}}^{(n)}\left(\mathbf{z}_i^{(n)}\right)$$

$$E := \sum_i \varepsilon_{\boldsymbol{\theta}}\left(\mathbf{x}_i^{(N)}\right)$$



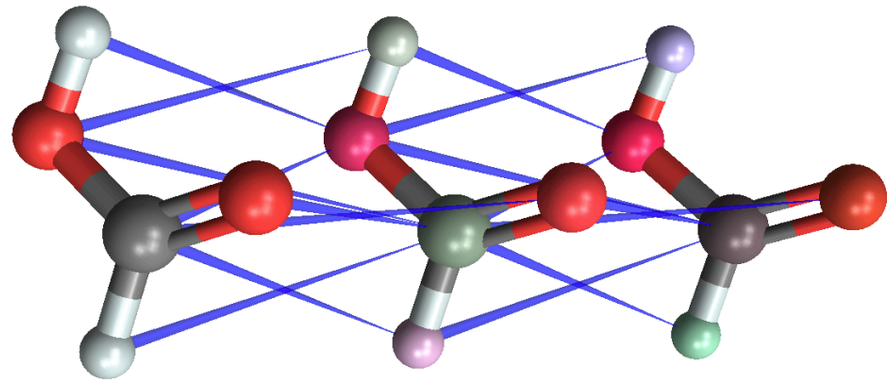
Graph neural networks

initialize $\mathbf{x}_i^{(0)}$

$$\mathbf{z}_i^{(n)} := \sum_j \mathbf{p}_{\boldsymbol{\theta}}^{(n)}(\mathbf{x}_i^{(n)}, \mathbf{x}_j^{(n)}, \mathbf{e}_{ij})$$

$$\mathbf{x}_i^{(n+1)} := \mathbf{q}_{\boldsymbol{\theta}}^{(n)}\left(\mathbf{x}_i^{(n)}, \mathbf{z}_i^{(n)}\right)$$

$$\mathcal{O}(N_{\text{edge}}^2)$$



Schütt et al. *J. Chem. Phys.* **148**, 241722 (2018)

SchNet for electrons in molecules

- Nodes represent electrons
- Separate same-spin, opposite-spin, and nuclear messages
- Single SchNet instance for both the Jastrow factor $J_{\boldsymbol{\theta}}(\mathbf{r})$ and backflow $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{r})$

$$\mathbf{x}_i^{(0)} := \mathbf{X}_{\boldsymbol{\theta},s_i}$$

$$\mathbf{z}_i^{(n,\pm)} := \sum\nolimits_{j\neq i}^{\pm} \mathbf{w}_{\boldsymbol{\theta}}^{(n,\pm)}\left(\mathbf{e}(|\mathbf{r}_i-\mathbf{r}_j|)\right) \odot \mathbf{h}_{\boldsymbol{\theta}}^{(n)}\left(\mathbf{x}_j^{(n)}\right)$$

$$\mathbf{z}_i^{(n,\mathrm{n})} := \sum\nolimits_{J} \mathbf{w}_{\boldsymbol{\theta}}^{(n,\mathrm{n})}\left(\mathbf{e}(|\mathbf{r}_i-\mathbf{R}_J|)\right) \odot \mathbf{Y}_{\boldsymbol{\theta},J}$$

$$\mathbf{x}_i^{(n+1)} := \mathbf{x}_i^{(n)} + \sum\nolimits_{\pm} \mathbf{g}_{\boldsymbol{\theta}}^{(n,\pm)}\left(\mathbf{z}_i^{(n,\pm)}\right) + \mathbf{g}_{\boldsymbol{\theta}}^{(n,\mathrm{n})}\left(\mathbf{z}_i^{(n,\mathrm{n})}\right)$$

$$J := \eta_{\boldsymbol{\theta}}\left(\sum\nolimits_i \mathbf{x}_i^{(N)}\right)$$

$$\mathbf{f}_i := \boldsymbol{\kappa}_{\boldsymbol{\theta}}\left(\mathbf{x}_i^{(N)}\right)$$

Graph neural network