Data-driven approaches and machine learning as DMFT solvers

Cedric Weber

King's College London

Juelich, October 2022

School of natural science
Department of Physics

KING'S College LONDON

Personal background

ARCTIC OCEAN
Greenland (Denmark)
Svalbard (Norway)
United States
Canada
Iceland
Russia
U.K.
Ireland
United States
ATLANTIC OCEAN
Portugal
Kazakhstan
Mongolia
China
Japan
Iran
India
PACIFIC OCEAN
Mexico
Caribbean
PACIFIC OCEAN
Algeria
Libya
Egypt
Sudan
Philippines
Brazil
INDIAN OCEAN
Madagascar
ATLANTIC OCEAN
Australia
New Zealand
Argentina
Chile

KCL group members

Top-bottom, left-right:

A. Haque, C. Weber, Y. Wei, F. Jamet

Z. Zhao, E. Plekahnov, C. Lupo, E. Sheridan

E. Chachkarova, H. Lee, T. Tse, D. Banerjee

External partners and funding agencies

AIM/Quantum

UKRI

Innovate UK

rahko

Rahko Quantum Machine Learning

NPL National Physical Laboratory

Dr Ivan Rungger

DFT

www.onetep.org

www.castep.org

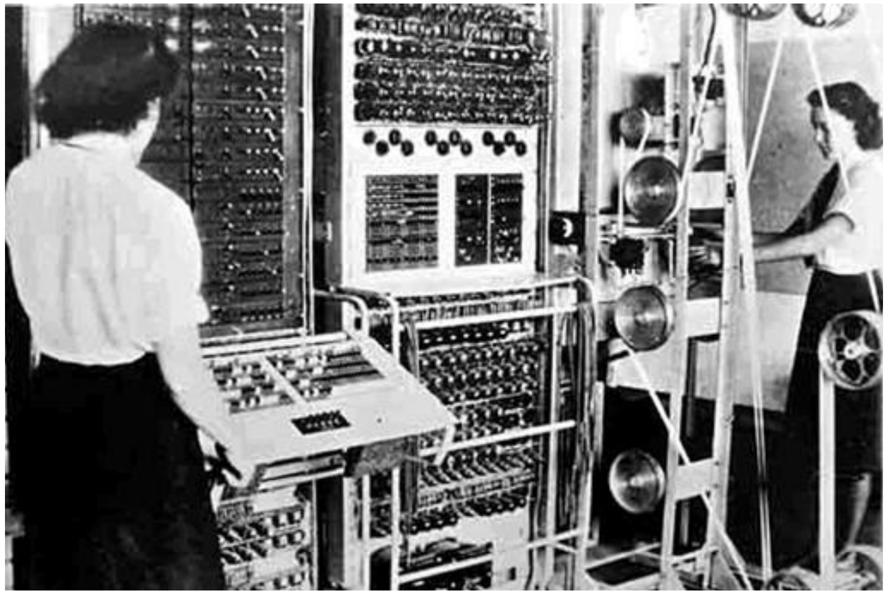
Outline

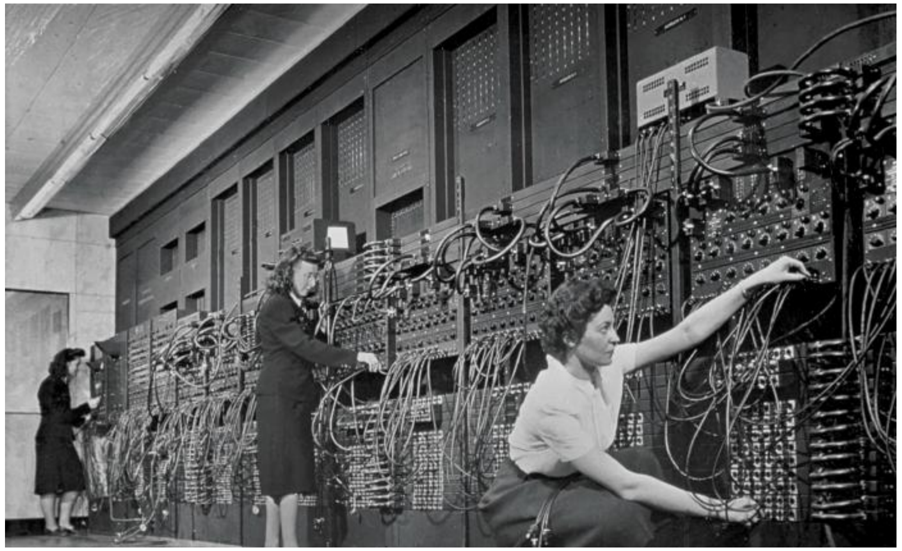
Introduction -

aims and motivations

Computational modelling

- *Eniac* – First programmable computer (US), *Electronic Numerical Integrator and Computer* (1940s)
- 30 tons and including 17,468 vacuum tubes.

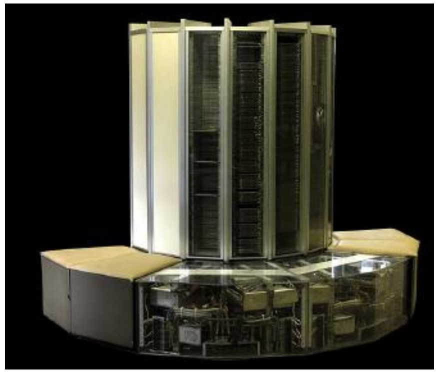




Emergence of quantum modelling

- Rapid progress with central architectures

Cray 1



Titan, Guangzhou (2014)

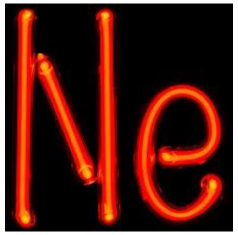
- But .. Most importantly <u>progresses in algorithmic</u>

- *Fast Fourier Transform:* N^2 to log(N)
- *Divide and conquer*

Size	DFT	FFT
10	800	166
100	80000	3321.93
1000	8e+06	49828.9
5000	4e+08	307193
10000	8e+08	664386
50000	4e+10	3.90241e+06
100000	8e+10	8.30482e+06
500000	4e+12	4.73289e+07
1000000	8e+12	9.96578e+07

Quantum wave-function

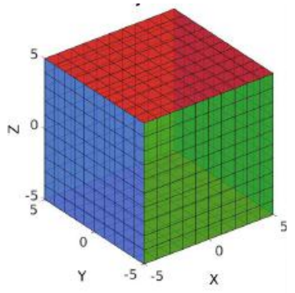
1 atom, 10 electrons



We are looking for a solution of the type of a wave function for many electrons:

$$\Psi(x_1, x_2, \ldots, x_N)$$

The problem is easy to write down …but the solution …



Storage required:

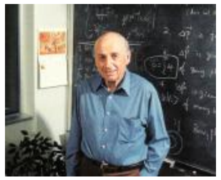
$$x \rightarrow 10 \times 10 \times 10 = 1000 \text{ data}$$

$$10 \text{ electrons} \rightarrow 1000^{10} \text{ data} \rightarrow 10^{30} \times 16 \text{ bytes}$$

$$= 16 \times 10^{21} \text{ Gb}$$

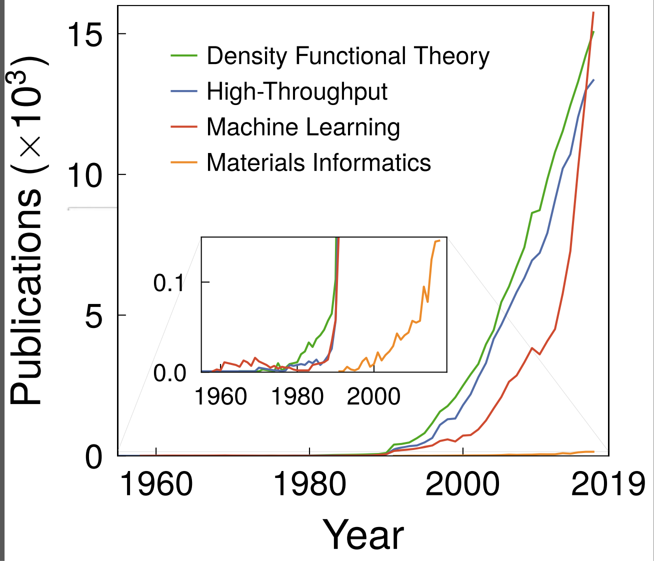
W-F to density

$$\frac{1}{\sqrt{N!}}\begin{vmatrix} \psi_1(\vec{x}_1) & \psi_2(\vec{x}_1) & \dots & \psi_N(\vec{x}_1) \\ \psi_1(\vec{x}_2) & \psi_2(\vec{x}_2) & \dots & \psi_N(\vec{x}_2) \\ \vdots & \vdots & & \vdots \\ \psi_1(\vec{x}_N) & \psi_2(\vec{x}_N) & \dots & \psi_N(\vec{x}_N) \end{vmatrix}$$

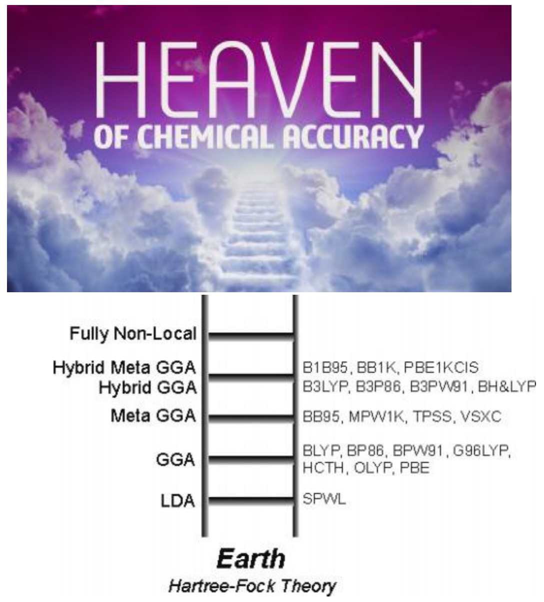


$$\rho(\vec{r}) = N\int \dots \int |\Psi(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N)|^2 ds_1 d\vec{x}_2 \dots d\vec{x}_N$$

DFT a success story

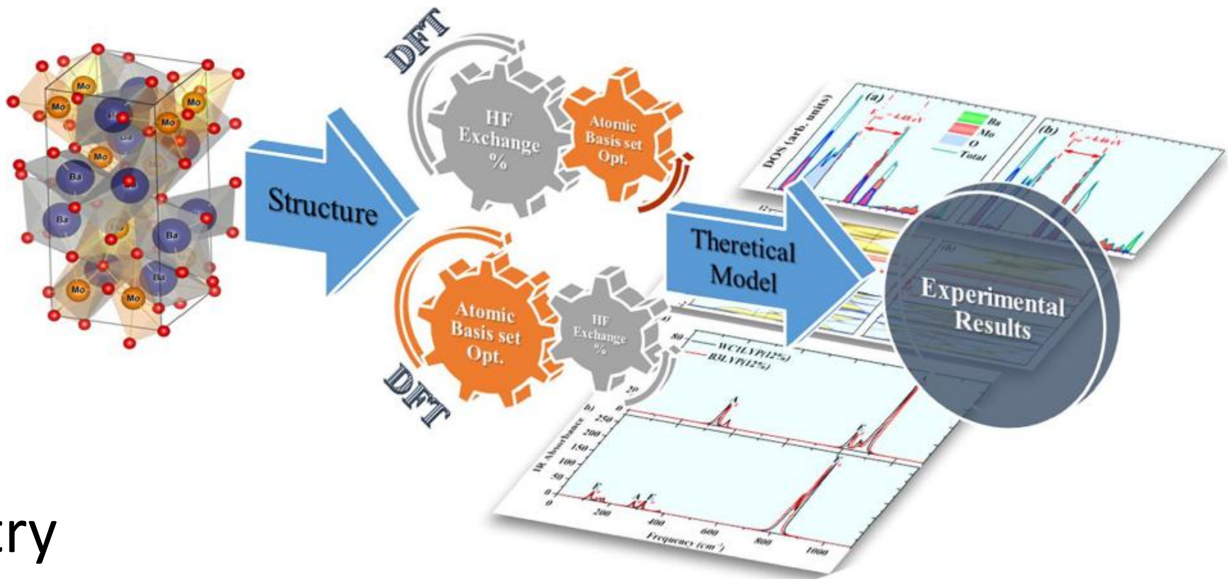


Jacob's ladder



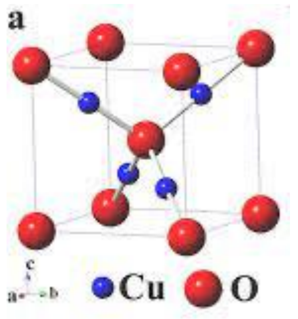
Density functional theory

Computational experiment

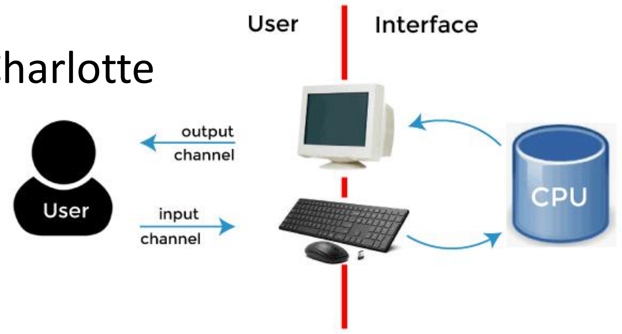


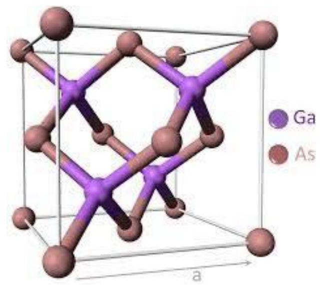
- Stoichiometry / Geometry
- Structure optimization
- Accuracy test and validation
- Properties (spectroscopy, thermal/mechanical, electronic, …)

Until recently ….

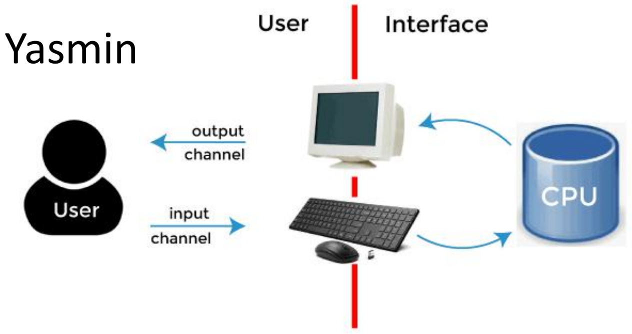


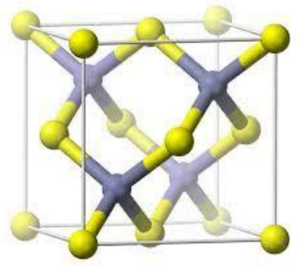
Charlotte



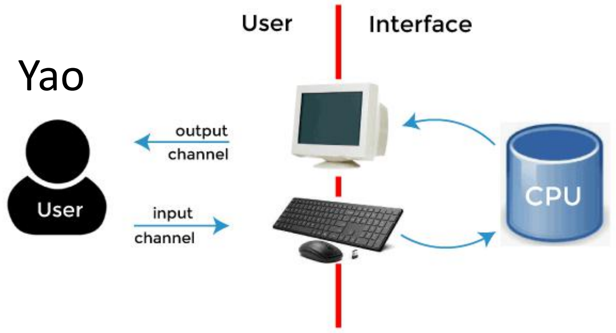


Yasmin

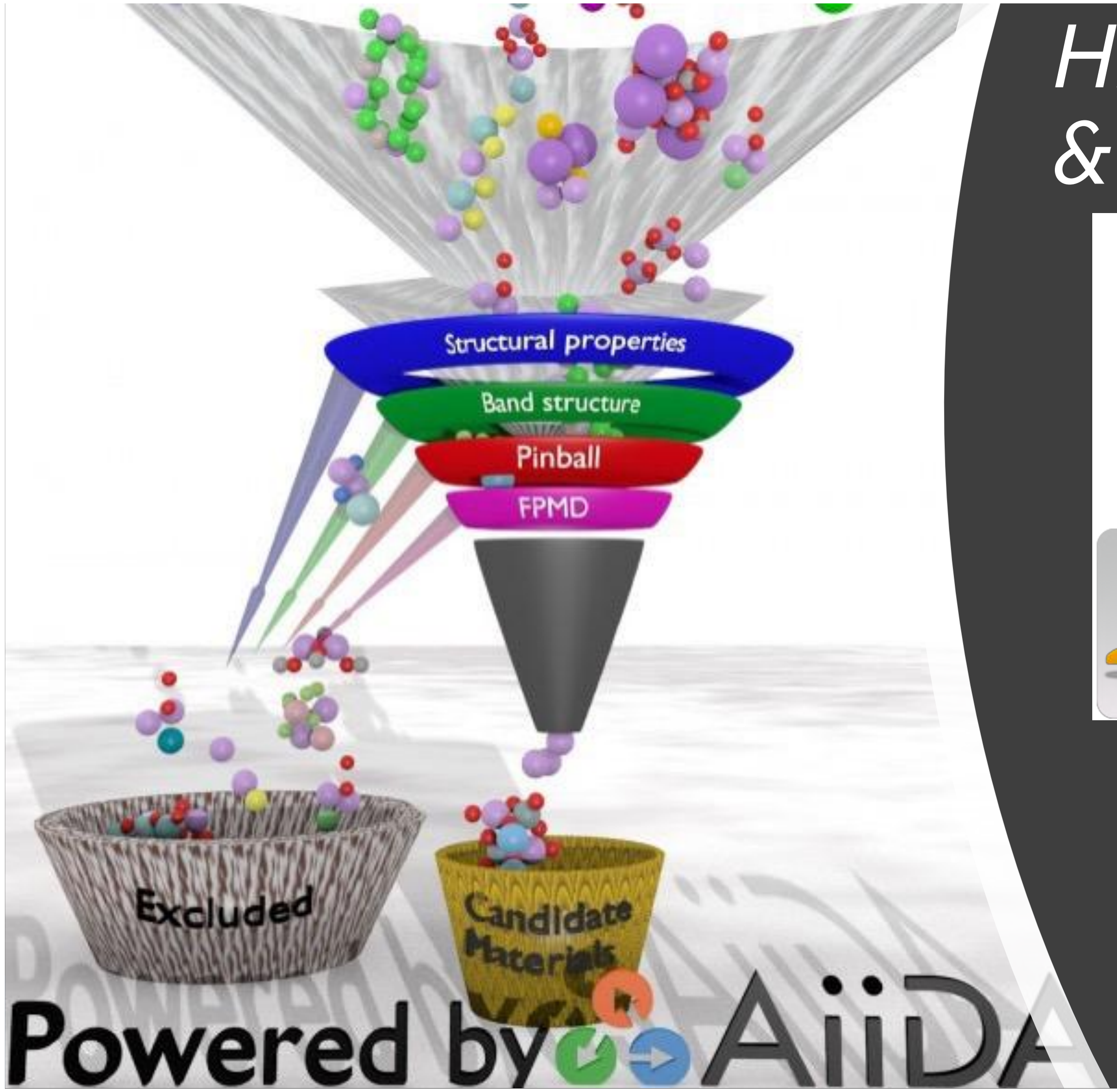




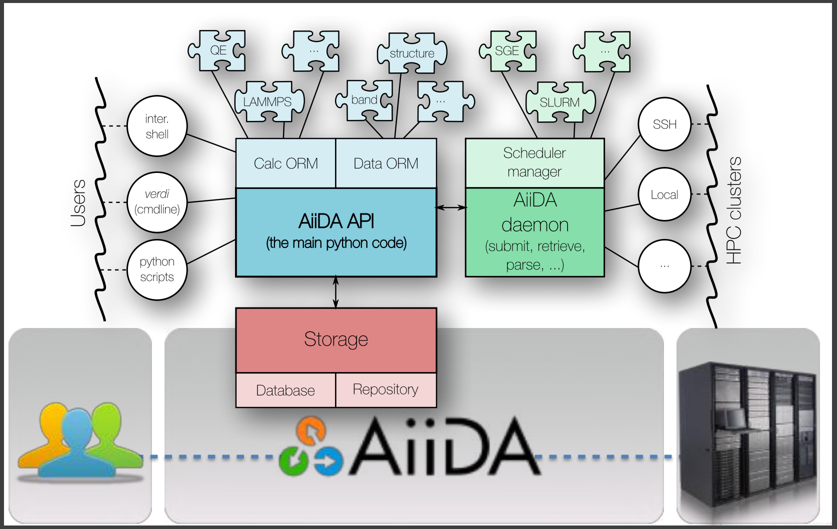
Yao



Where of course some characters are real and some are imaginary ….



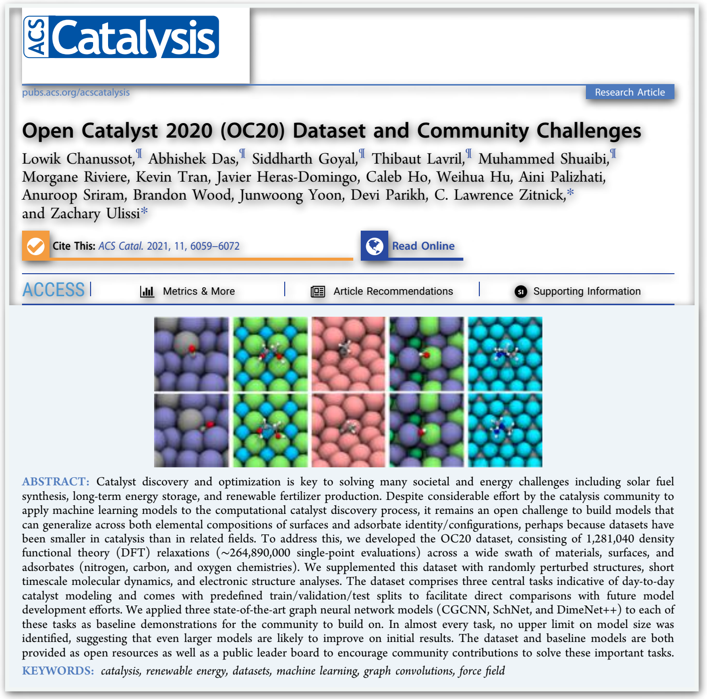
High Throughput & Automation



Accelerates and automates material screening for desired properties

DFT & machine learning, different strategies

Predicting energetics and forces from direct sampling, large compositional space for small molecules where accuracy matters



ACS Catalysis

pubs.acs.org/acscatalysis Research Article

Open Catalyst 2020 (OC20) Dataset and Community Challenges

Lowik Chanussot,¶ Abhishek Das,¶ Siddharth Goyal,¶ Thibaut Lavril,¶ Muhammed Shuaibi,¶ Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick,* and Zachary Ulissi*

Cite This: *ACS Catal.* 2021, 11, 6059−6072 Read Online

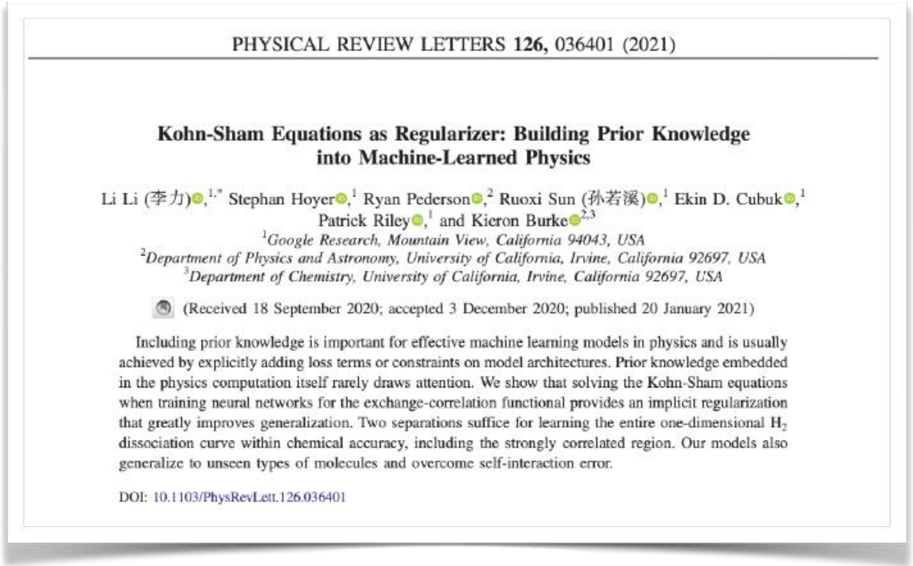
ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Catalyst discovery and optimization is key to solving many societal and energy challenges including solar fuel synthesis, long-term energy storage, and renewable fertilizer production. Despite considerable effort by the catalysis community to apply machine learning models to the computational catalyst discovery process, it remains an open challenge to build models that can generalize across both elemental compositions of surfaces and adsorbate identity/configurations, perhaps because datasets have been smaller in catalysis than in related fields. To address this, we developed the OC20 dataset, consisting of 1,281,040 density functional theory (DFT) relaxations (~264,890,000 single-point evaluations) across a wide swath of materials, surfaces, and adsorbates (nitrogen, carbon, and oxygen chemistries). We supplemented this dataset with randomly perturbed structures, short timescale molecular dynamics, and electronic structure analyses. The dataset comprises three central tasks indicative of day-to-day catalyst modeling and comes with predefined train/validation/test splits to facilitate direct comparisons with future model development efforts. We applied three state-of-the-art graph neural network models (CGCNN, SchNet, and DimeNet++) to each of these tasks as baseline demonstrations for the community to build on. In almost every task, no upper limit on model size was identified, suggesting that even larger models are likely to improve on initial results. The dataset and baseline models are both provided as open resources as well as a public leader board to encourage community contributions to solve these important tasks.

KEYWORDS: *catalysis, renewable energy, datasets, machine learning, graph convolutions, force field*

Facebook / Carnegie collaboration, OC20 database for catalysis

Finding the exchange functional with machine learning & non-local functional for correlation



PHYSICAL REVIEW LETTERS **126,** 036401 (2021)

Kohn-Sham Equations as Regularizer: Building Prior Knowledge into Machine-Learned Physics

Li Li (李力),[1,*] Stephan Hoyer,[1] Ryan Pederson,[2] Ruoxi Sun (孙若溪),[1] Ekin D. Cubuk,[1] Patrick Riley,[1] and Kieron Burke[2,3]

[1]*Google Research, Mountain View, California 94043, USA*
[2]*Department of Physics and Astronomy, University of California, Irvine, California 92697, USA*
[3]*Department of Chemistry, University of California, Irvine, California 92697, USA*

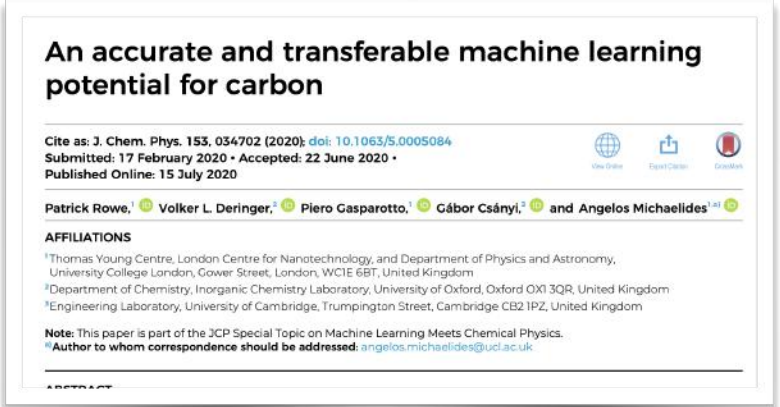
(Received 18 September 2020; accepted 3 December 2020; published 20 January 2021)

Including prior knowledge is important for effective machine learning models in physics and is usually achieved by explicitly adding loss terms or constraints on model architectures. Prior knowledge embedded in the physics computation itself rarely draws attention. We show that solving the Kohn-Sham equations when training neural networks for the exchange-correlation functional provides an implicit regularization that greatly improves generalization. Two separations suffice for learning the entire one-dimensional H_2 dissociation curve within chemical accuracy, including the strongly correlated region. Our models also generalize to unseen types of molecules and overcome self-interaction error.

DOI: 10.1103/PhysRevLett.126.036401

Kieron Burke group

Inter-atomic potential trained with DFT data-set for specific systems



An accurate and transferable machine learning potential for carbon

Cite as: J. Chem. Phys. 153, 034702 (2020); doi: 10.1063/5.0005084
Submitted: 17 February 2020 • Accepted: 22 June 2020 •
Published Online: 15 July 2020

View Online Export Citation CrossMark

Patrick Rowe,[1] Volker L. Deringer,[2] Piero Gasparotto,[1] Gábor Csányi,[3] and Angelos Michaelides[1,a)]

AFFILIATIONS

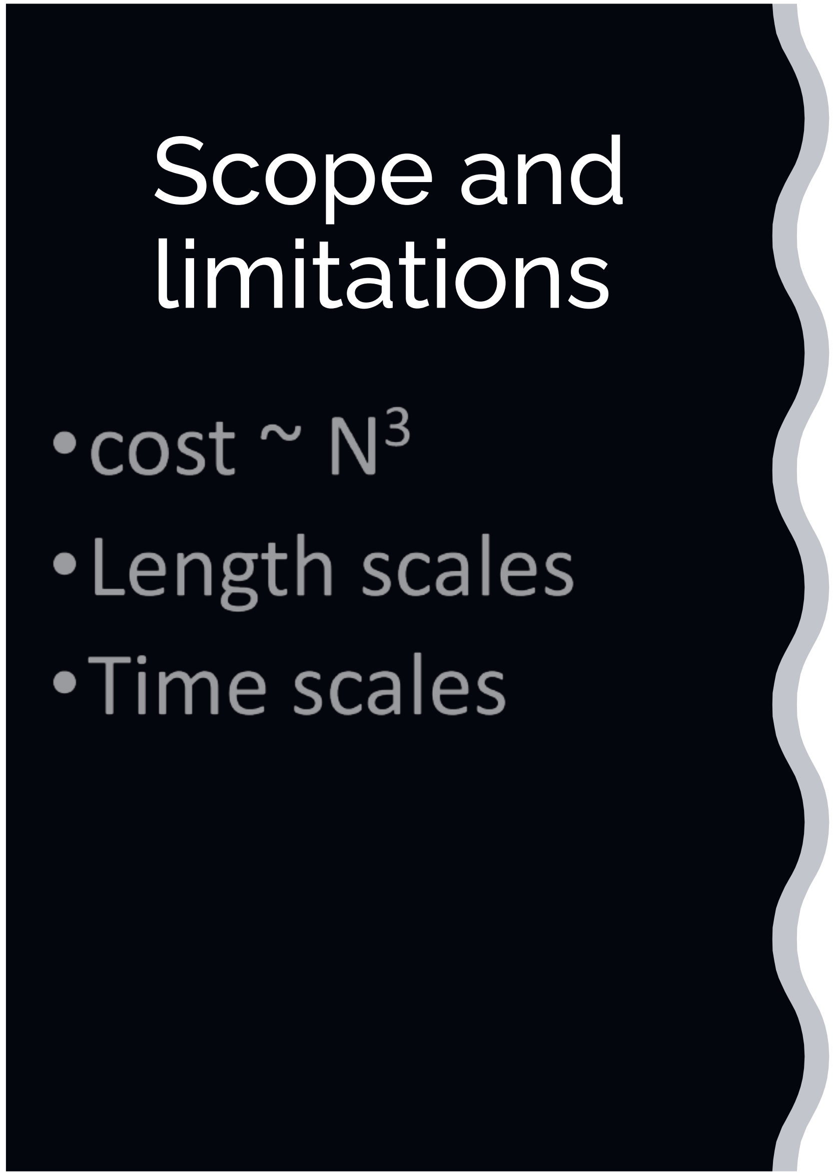
[1]Thomas Young Centre, London Centre for Nanotechnology, and Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT, United Kingdom
[2]Department of Chemistry, Inorganic Chemistry Laboratory, University of Oxford, Oxford OX1 3QR, United Kingdom
[3]Engineering Laboratory, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, United Kingdom

Note: This paper is part of the JCP Special Topic on Machine Learning Meets Chemical Physics.
a)**Author to whom correspondence should be addressed:** angelos.michaelides@ucl.ac.uk

G Csanyi & M Michaelides groups

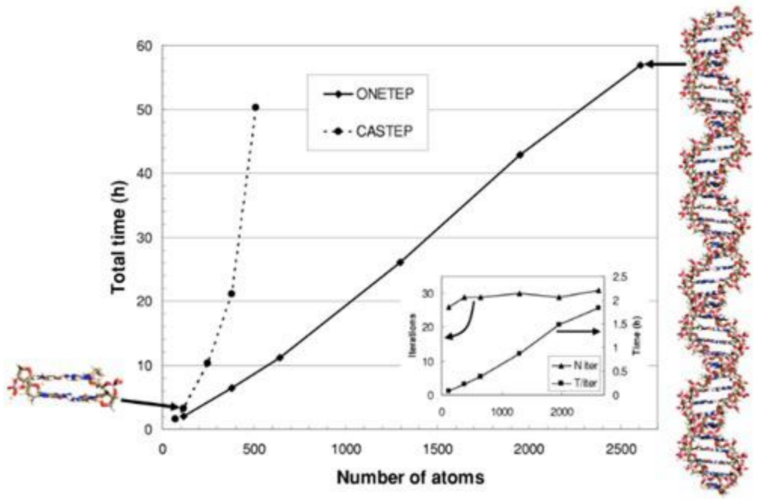
Global theme:
data sharing & community driven

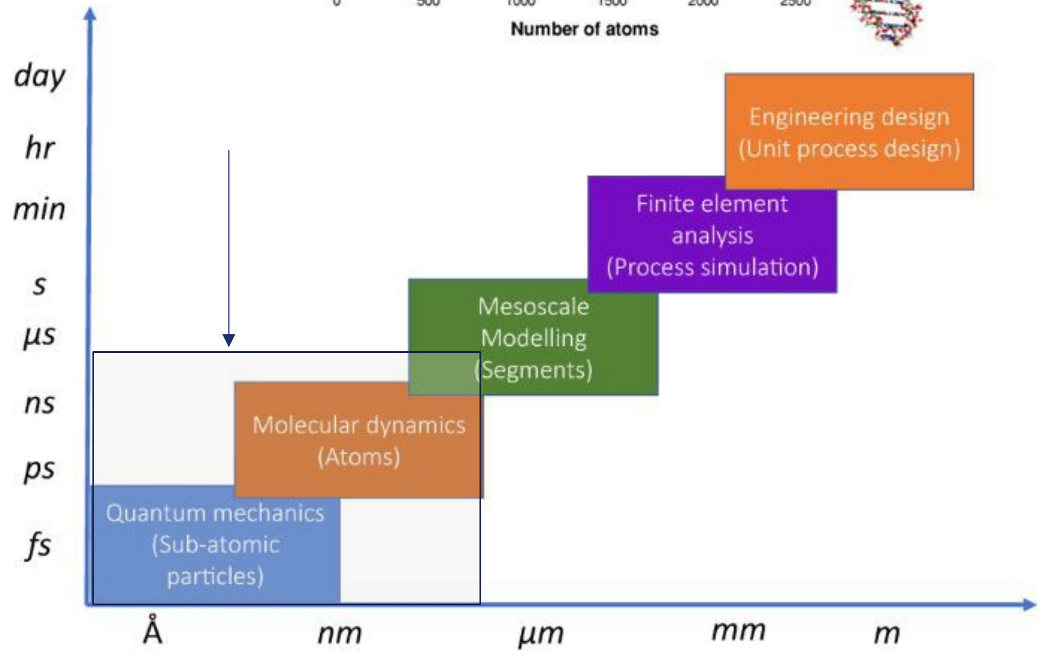
ML allows faster & larger



Scope and limitations

- cost ~ N^3
- Length scales
- Time scales

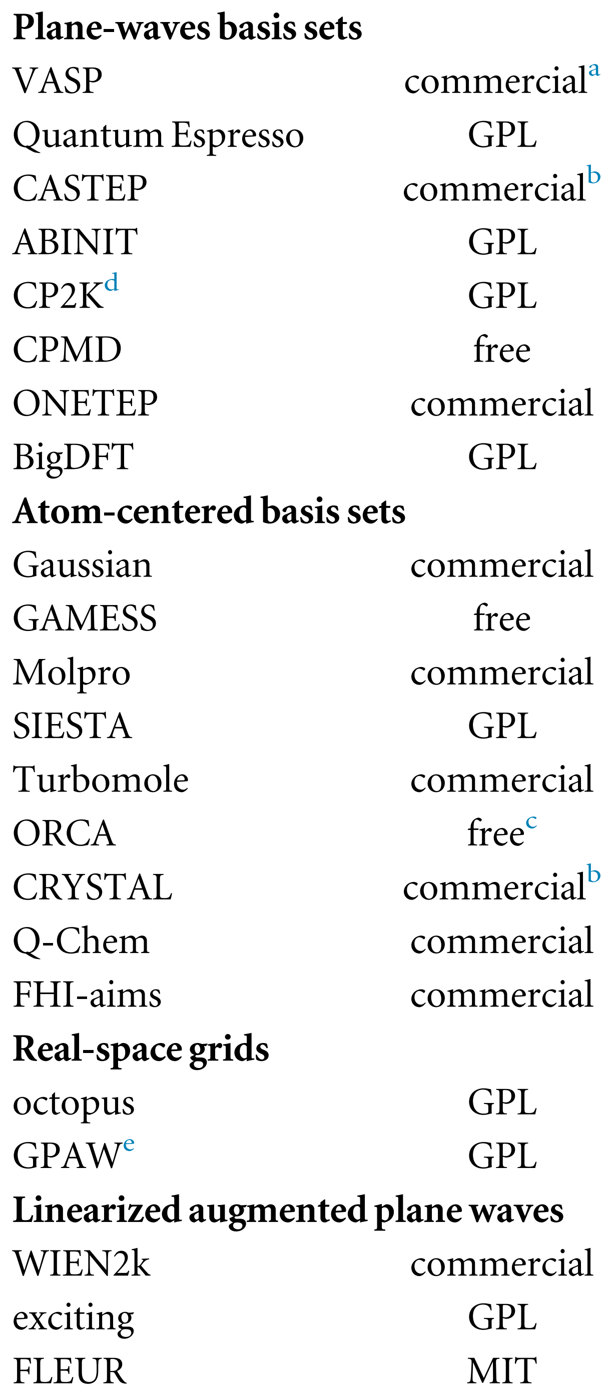




day
hr
min
s
μs
ns
ps
fs
Å
nm
μm
mm
m
Quantum mechanics (Sub-atomic particles)
Molecular dynamics (Atoms)
Mesoscale Modelling (Segments)
Finite element analysis (Process simulation)
Engineering design (Unit process design)

Blockers, bottlenecks and challenges for ML-DFT:

1. ***compositional material space*** is vast
2. Learning functionals challenging: ***complex nature of Kohn-Sham functionals***
3. In DFT total energies (or other ***traced quantities***) are meaningful
4. ***Various codes and functionals***, data-base to adapt for each implementation, inter-operability
5. ML model for DFT won't better DFT - issues for ***self-interaction and electronic interactions remain***



Plane-waves basis sets	
VASP	commercial[a]
Quantum Espresso	GPL
CASTEP	commercial[b]
ABINIT	GPL
CP2K[d]	GPL
CPMD	free
ONETEP	commercial
BigDFT	GPL
Atom-centered basis sets	
Gaussian	commercial
GAMESS	free
Molpro	commercial
SIESTA	GPL
Turbomole	commercial
ORCA	free[c]
CRYSTAL	commercial[b]
Q-Chem	commercial
FHI-aims	commercial
Real-space grids	
octopus	GPL
GPAW[e]	GPL
Linearized augmented plane waves	
WIEN2k	commercial
exciting	GPL
FLEUR	MIT

Machine learning and correlated materials

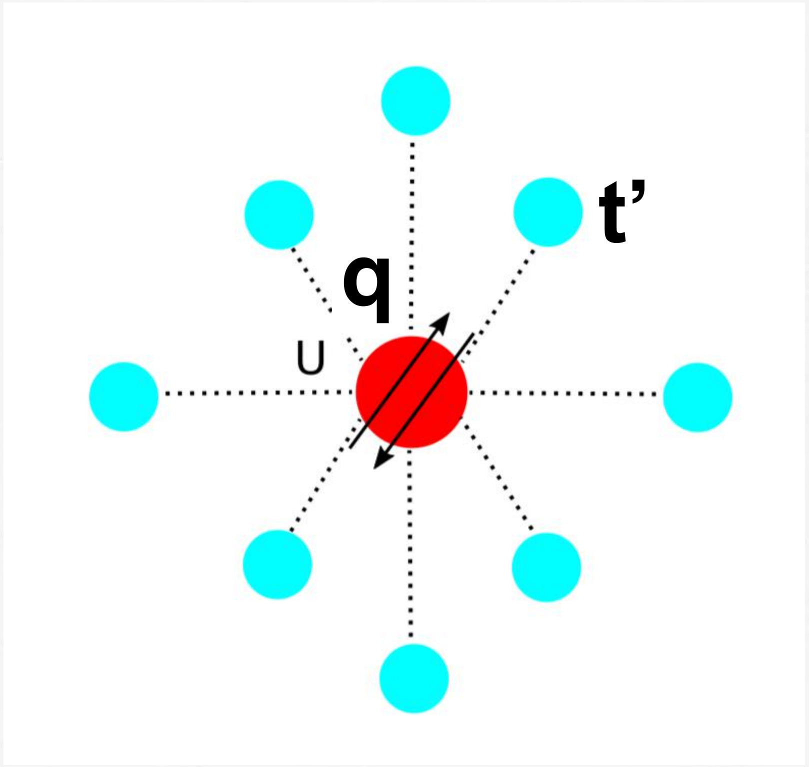
(for DMFT & DFT+DMFT see lectures Prof. ***Vollhardt***, ***Werner***, ***Held*** & ***Lichtenstein***)

Anderson impurity model: Hamiltonian representation

$$\boxed{H_{\mathrm{AIM}} = \sum_{\alpha,\beta} t'_{\alpha\beta} c^\dagger_\alpha c_\beta + \sum_{\alpha,\mu} \left(\theta_{\alpha\mu} c^\dagger_\alpha a_\mu + \mathrm{H.c.}\right) + \sum_{\mu}^{N_b} \varepsilon_\mu a^\dagger_\mu a_\mu}$$

$$\boldsymbol{G}_{\mathrm{full}}(i\omega_n) = \frac{1}{i\omega_n - \boldsymbol{T}}$$

$$\boldsymbol{T} = \begin{array}{c} \\ \\ \end{array}\begin{array}{cc} \text{bath} & \text{imp} \\ \left(\begin{array}{c:c} \boldsymbol{t}' & \boldsymbol{\theta} \\ \hdashline \boldsymbol{\theta}^\dagger & \boldsymbol{\varepsilon} \end{array}\right) & \end{array}\begin{array}{l} \\ \text{bath} \\ \text{imp} \end{array}$$



T is the full hopping matrix, bath and impurity

Weiss field

$$\boldsymbol{G}_{\text{full}}(i\omega_n) = \frac{1}{i\omega_n - \boldsymbol{T}}$$

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}^{-1}$$

$$A_{11} = \omega - \boldsymbol{t}$$

$$A_{12} = A_{21}^{\dagger} = \boldsymbol{\theta}$$

$$A_{22} = \omega - \boldsymbol{\varepsilon}$$

B_{11}=G_{imp}

Weiss field

$$\boldsymbol{G}_{\text{full}}(i\omega_n) = \frac{1}{i\omega_n - \boldsymbol{T}}$$

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}^{-1}$$

$$A_{11} = \omega - \boldsymbol{t}$$

$$A_{12} = A_{21}^{\dagger} = \boldsymbol{\theta}$$

$$A_{22} = \omega - \boldsymbol{\varepsilon}$$

Inverse condition:

$$A_{11}B_{11} + A_{12}B_{21} = \mathbf{1}$$

$$B_{21} = -A_{22}^{-1}A_{21}B_{11}$$

$$\Rightarrow \quad \left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)B_{11} = \mathbf{1}$$

Weiss field

$$\boldsymbol{G}_{\mathrm{full}}(i\omega_n) = \frac{1}{i\omega_n - \boldsymbol{T}}$$

$$\left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right) B_{11} = \mathbf{1}$$

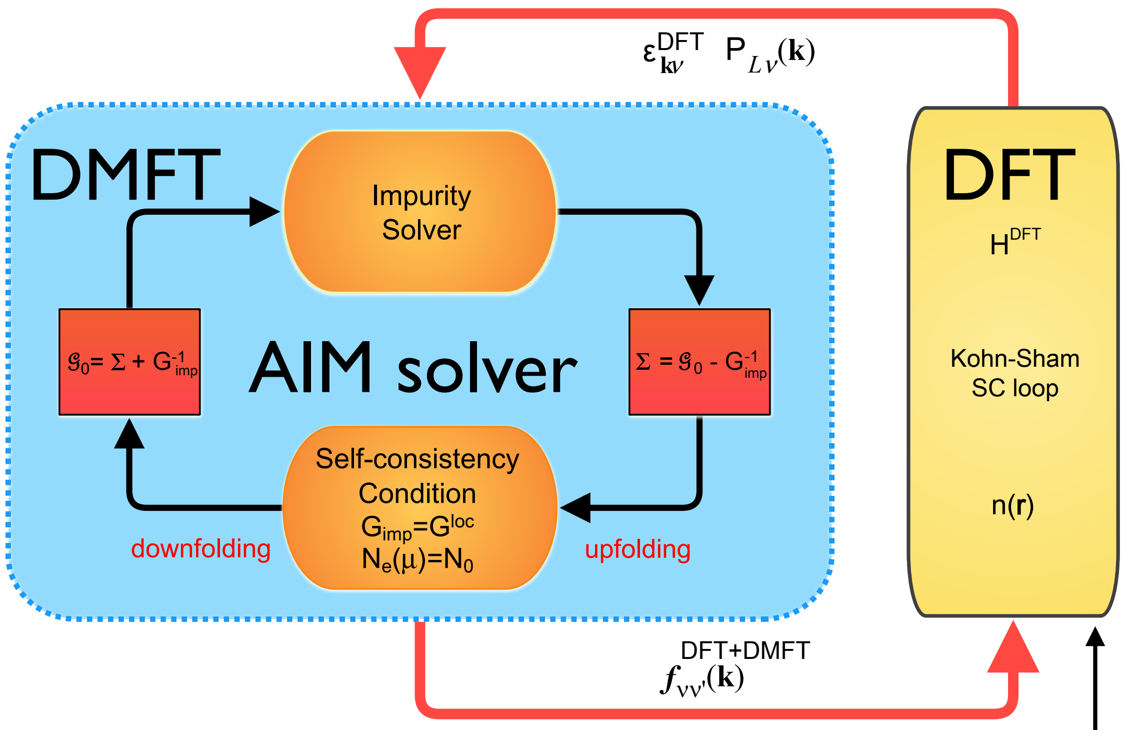
$$\Rightarrow \mathcal{G}^{-1} = \omega - \boldsymbol{t}' - \boxed{\boldsymbol{\theta}\frac{1}{\omega - \boldsymbol{\varepsilon}}\boldsymbol{\theta}^{\dagger}}$$

Weiss field

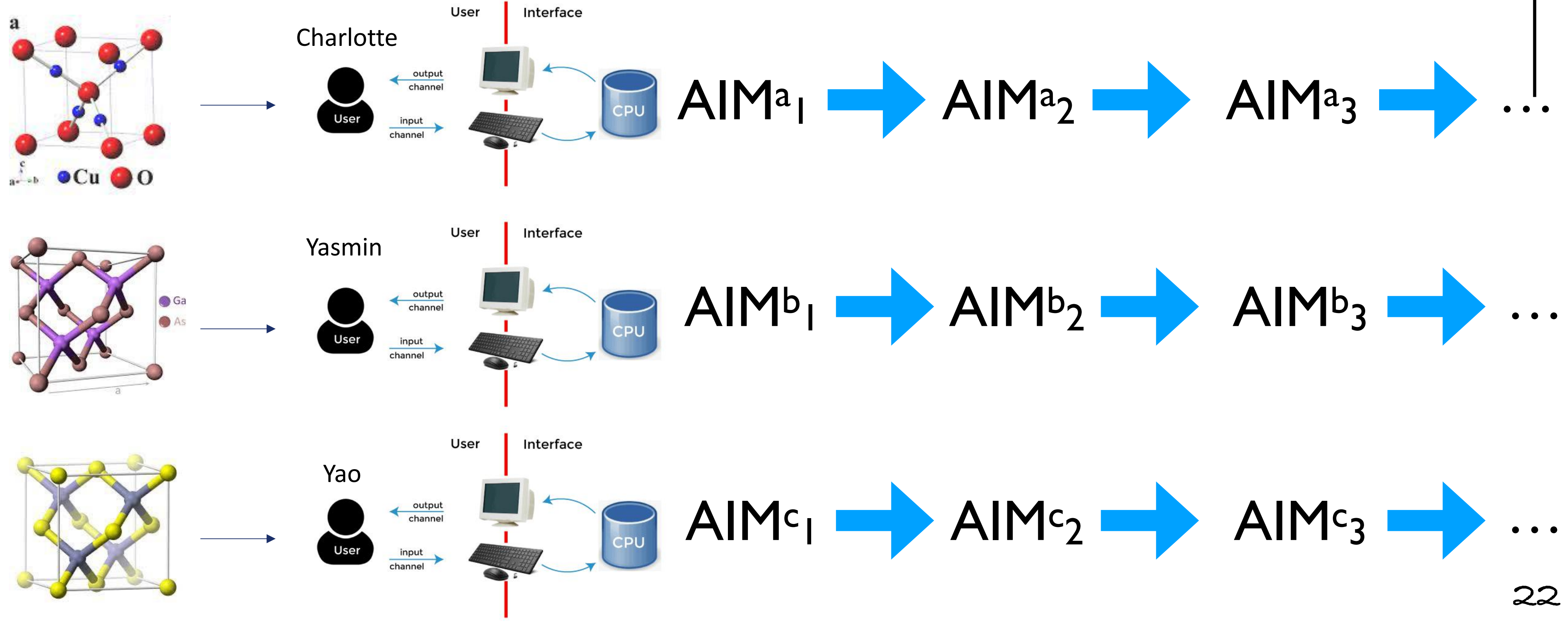
Δ(ω)

$$\boxed{\begin{aligned} A_{11} &= \omega - \boldsymbol{t} \\ A_{12} &= A_{21}^{\dagger} = \boldsymbol{\theta} \\ A_{22} &= \omega - \boldsymbol{\varepsilon} \end{aligned}}$$

hybridisation matrix Δ: “dynamic transfer between impurity and bath”


$\varepsilon^{DFT}_{\mathbf{k}\nu}$ $P_{L\nu}(\mathbf{k})$
DMFT
Impurity Solver
$\mathcal{G}_0 = \Sigma + G^{-1}_{imp}$
AIM solver
$\Sigma = \mathcal{G}_0 - G^{-1}_{imp}$
Self-consistency Condition
$G_{imp} = G^{loc}$
$N_e(\mu) = N_0$
downfolding
upfolding
DFT
H^{DFT}
Kohn-Sham SC loop
$n(\mathbf{r})$
DFT+DMFT
$f_{\nu\nu'}(\mathbf{k})$

$$H_{\mathrm{AIM}} = \sum_{\alpha,\beta} t'_{\alpha\beta} c^{\dagger}_{\alpha} c_{\beta} + \sum_{\alpha,\mu} \left(\theta_{\alpha\mu} c^{\dagger}_{\alpha} a_{\mu} + \mathrm{H.c.} \right) + \sum_{\mu}^{N_b} \varepsilon_{\mu} a^{\dagger}_{\mu} a_{\mu}$$


a
Cu
O
Ga
As
Charlotte
Yasmin
Yao
User
Interface
output channel
input channel
CPU
AIM^{a}_{1} AIM^{a}_{2} AIM^{a}_{3} ...
AIM^{b}_{1} AIM^{b}_{2} AIM^{b}_{3} ...
AIM^{c}_{1} AIM^{c}_{2} AIM^{c}_{3} ...

Range of AIM parameters and energy scales - example of Bethe lattice

- Bethe lattice, semi-circular DOS with half bandwidth D (DOS from -D to +D)
- local impurity GF:

$$G(i\omega_n) = \frac{2}{(\pi D)^2}\int_{-\infty}^{\infty} d\epsilon \frac{\sqrt{D^2-\epsilon^2}}{i\omega_n - \epsilon}\Theta(D-|\epsilon|)$$

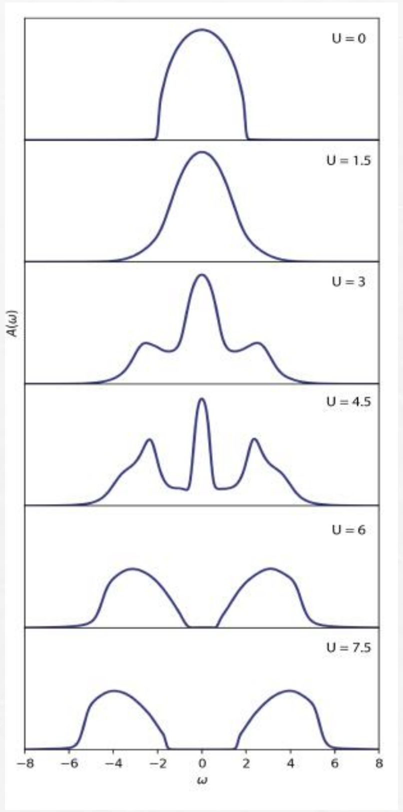
- discretized GF approximation:

$$\mathcal{G}_0^{-1} = i\omega_n + \mu - \sum_{i=1}^{N_b} \frac{V_i^2}{i\omega_n - \epsilon_i},$$

- Bounds for database: $V \in [V_{\min}, V_{\max}]$ $\epsilon \in [\epsilon_{\min}, \epsilon_{\max}]$
- Range of parameters (V^2 and ϵ scale with bandwith):

$$\sum_i V_i^2 = D^2/4$$

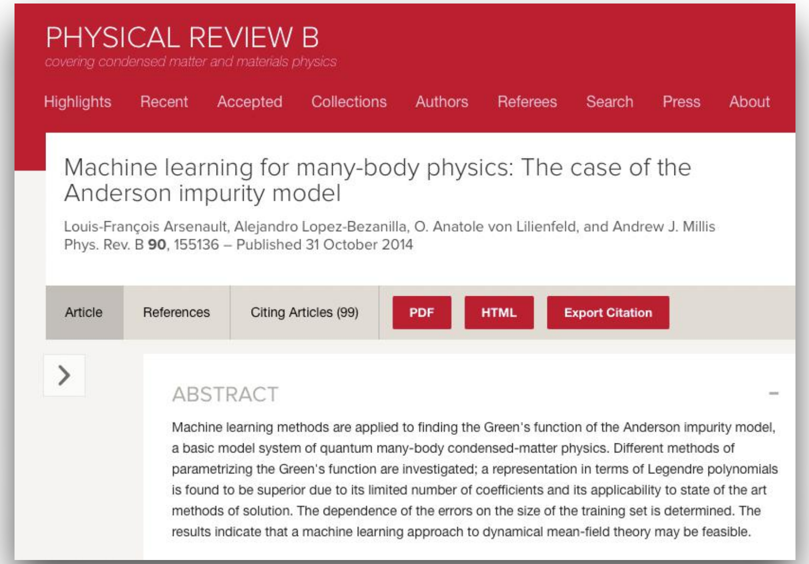
$$\frac{\max[\{\epsilon_1,\ldots,\epsilon_N\}] - \min[\{\epsilon_1,\ldots,\epsilon_N\}]}{2D} = 1$$



ML for DMFT, advantages:

1. ***compositional AIM space*** is moderate ~ 20-100 parameters
2. ***Various codes and implementations of DMFT***, but low entry-cost to adapt-change solvers, inter-operability
3. ML model for DMFT will provide ***improvements beyond-DFT***
4. Learning Green's functions facilitated in some limits, e.g. ***high temperature, weak-coupling or atomic limits***
5. We have fast solvers for generating Green's functions, we only need to provide good models for ***corrections to known approximations***
6. AIM exponential wall - ***large benefit and speed-up***
7. DMFT iterations are ***resilient with respect to errors, high accuracy not always critical***
8. AIM solutions might be applicable to several close combinations of ***structure and stoichiometry (structural relaxation, doping & pressure phase diagrams, phonons, …)***

ML for DMFT - learning solutions of DMFT with regression kernels for the Hubbard model



PHYSICAL REVIEW B
covering condensed matter and materials physics

Highlights Recent Accepted Collections Authors Referees Search Press About

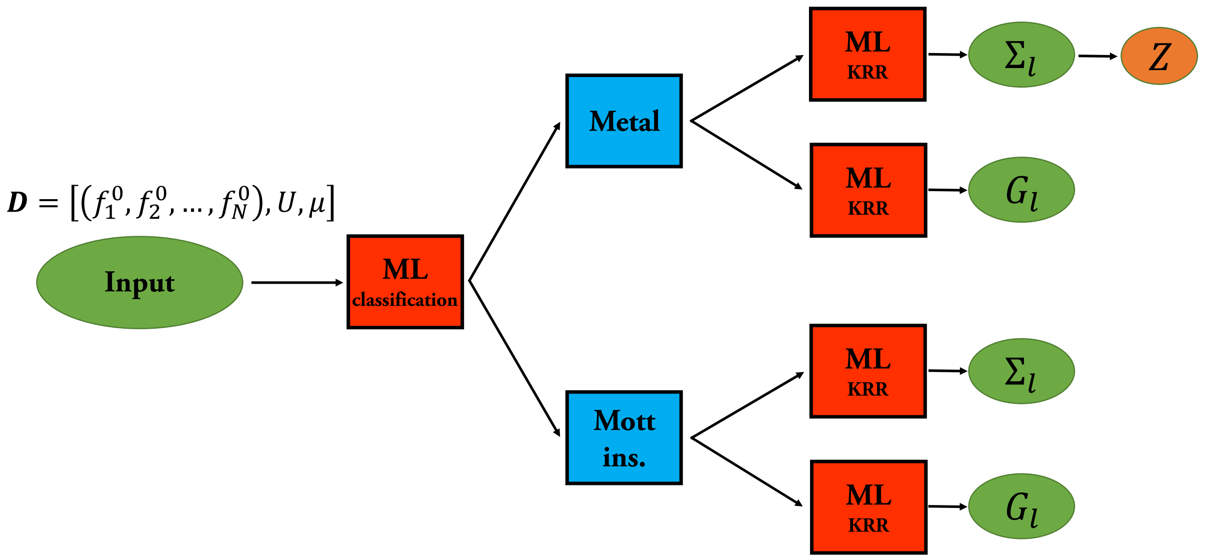
Machine learning for many-body physics: The case of the Anderson impurity model

Louis-François Arsenault, Alejandro Lopez-Bezanilla, O. Anatole von Lilienfeld, and Andrew J. Millis
Phys. Rev. B **90**, 155136 – Published 31 October 2014

Article References Citing Articles (99) PDF HTML Export Citation

ABSTRACT

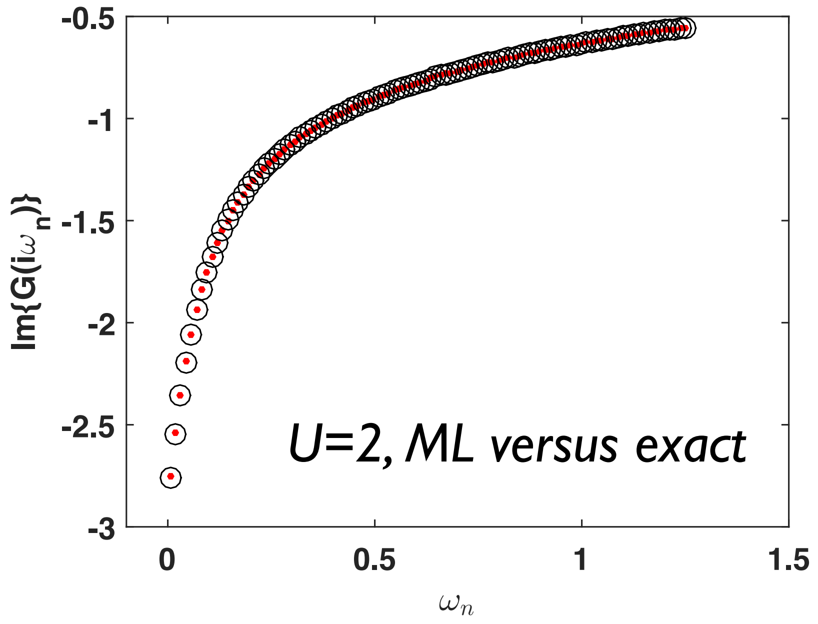
Machine learning methods are applied to finding the Green's function of the Anderson impurity model, a basic model system of quantum many-body condensed-matter physics. Different methods of parametrizing the Green's function are investigated; a representation in terms of Legendre polynomials is found to be superior due to its limited number of coefficients and its applicability to state of the art methods of solution. The dependence of the errors on the size of the training set is determined. The results indicate that a machine learning approach to dynamical mean-field theory may be feasible.



Inputs: information to be learned, vectors: hybridisation function (tau or Legendre)
Outputs: ML prediction, vectors: DMFT iterations are
Descriptor D (Problem to be solved): input function + few scalar parameters (U & chem.pot.)

$$f(z) \rightarrow \boldsymbol{f} = (f_1, f_2, \dots, f_N)_{output}$$

: *Interpolate solutions using Kernel Ridge Regression*



Limitation : <u>***one database per model***</u> (… and material)

Brief introduction to neural networks

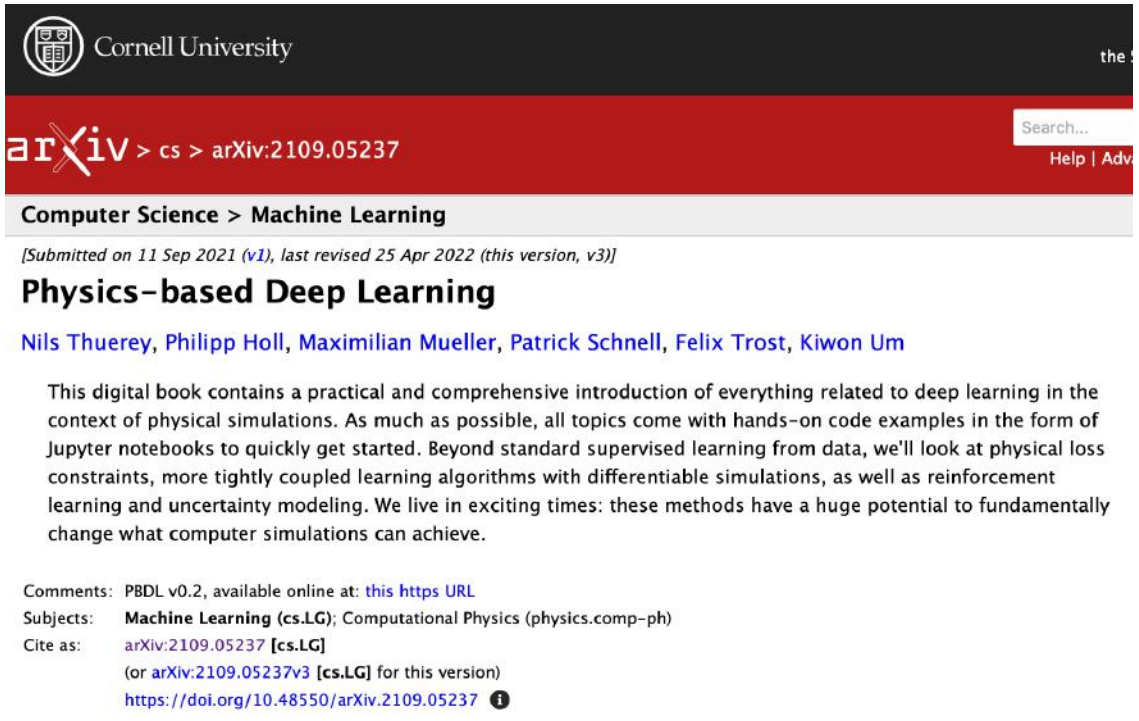
Literature

1. *Andrew Ng* class on machine learning (open access course *https://www.andrewng.org/courses/*) stepping stone to dive into the field

2. *Physics-based Deep Learning* (arxiv.org/abs/2109.05237) Focus on deep learning.

3. *Kieron Burke: Machine Learning in Materials Science and Electronic Structure Theory* (https://www.youtube.com/watch?v=vceNTbOGU-4&t=282s) - covers regression, classification, outliers …

4. .. many more!



Cornell University

arXiv > cs > arXiv:2109.05237

Computer Science > Machine Learning

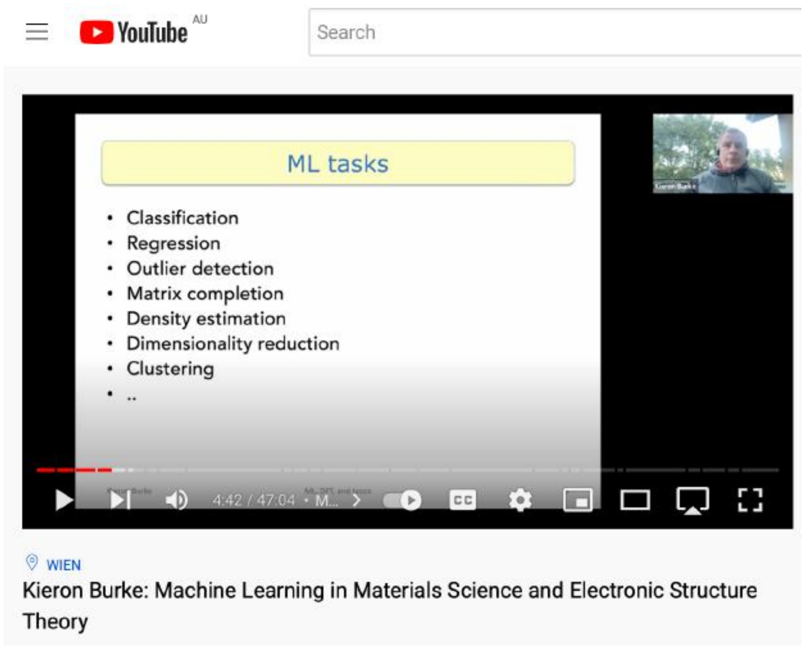
[Submitted on 11 Sep 2021 (v1), last revised 25 Apr 2022 (this version, v3)]

Physics-based Deep Learning

Nils Thuerey, Philipp Holl, Maximilian Mueller, Patrick Schnell, Felix Trost, Kiwon Um

This digital book contains a practical and comprehensive introduction of everything related to deep learning in the context of physical simulations. As much as possible, all topics come with hands-on code examples in the form of Jupyter notebooks to quickly get started. Beyond standard supervised learning from data, we'll look at physical loss constraints, more tightly coupled learning algorithms with differentiable simulations, as well as reinforcement learning and uncertainty modeling. We live in exciting times: these methods have a huge potential to fundamentally change what computer simulations can achieve.

Comments: PBDL v0.2, available online at: this https URL
Subjects: Machine Learning (cs.LG); Computational Physics (physics.comp-ph)
Cite as: arXiv:2109.05237 [cs.LG]
(or arXiv:2109.05237v3 [cs.LG] for this version)
https://doi.org/10.48550/arXiv.2109.05237



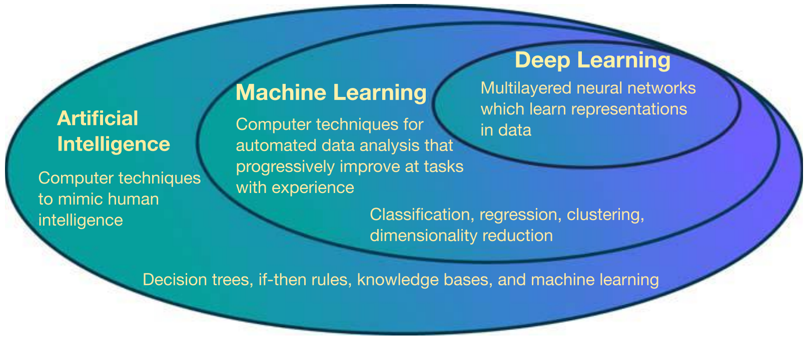
WIEN
Kieron Burke: Machine Learning in Materials Science and Electronic Structure Theory

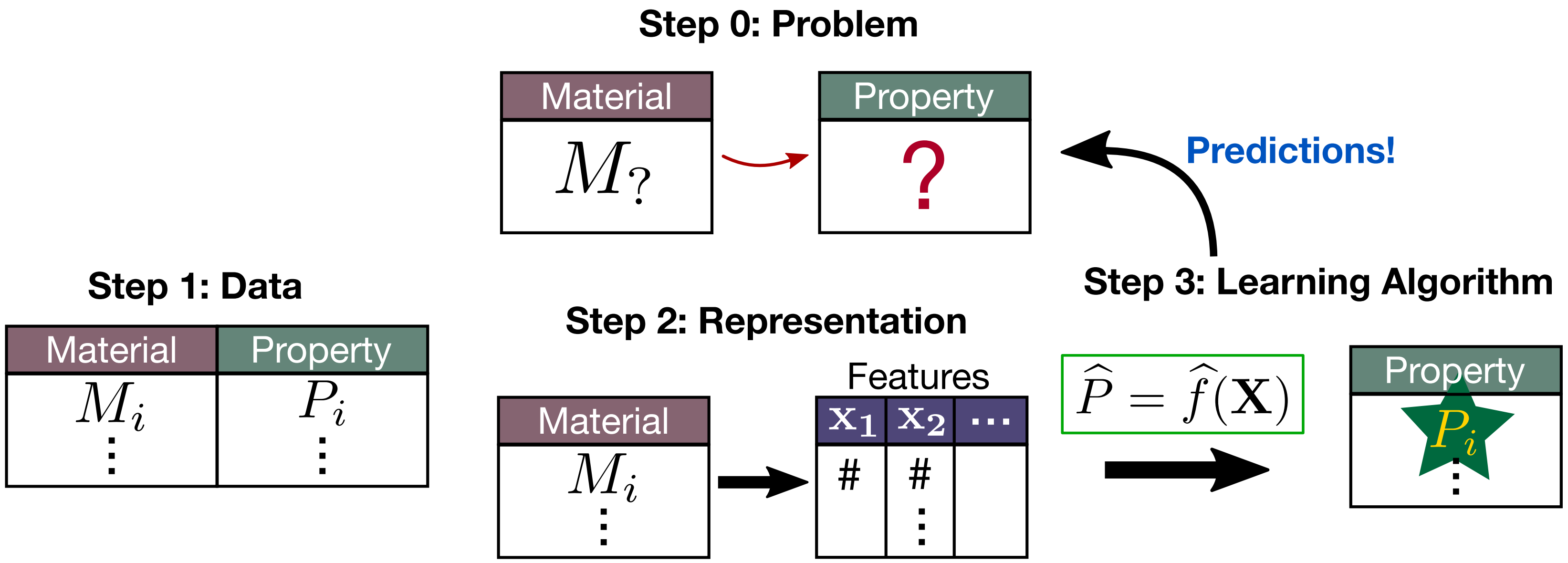
ML and neural network

Neural networks: a subset of machine learning techniques, itself part of the larger scope of AI

What is machine learning: multi-step process to provide predictions based on previous observations

1. Dataset
2. Representation of datas (possibly classification into features)
3. Problem to solve (materials property)
4. Learning algorithm (compare the model with the dataset)
5. An inference process to make predictions


Deep Learning
Multilayered neural networks which learn representations in data
Machine Learning
Computer techniques for automated data analysis that progressively improve at tasks with experience
Artificial Intelligence
Computer techniques to mimic human intelligence
Classification, regression, clustering, dimensionality reduction
Decision trees, if-then rules, knowledge bases, and machine learning



Supervised learning - linear regression

Model with *two* variables

x_1: weight
x_2: battery capacity

—> Predict : mileage

Vehicle List		
Vehicle weight (Kg)	Battery Capacity (kWh)	Mileage (MPGe)
1000	54	108
1500	81	103
2000	108	98
2500	135	93
3000	162	88
3500	189	83
4000	217	78

We want a good ***model*** for the dataset, we choose two parameters (***weights***) and a constant (***bias***):

$$h(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \quad \longrightarrow \quad h(\mathbf{x}) = \sum_{i=0}^{d} \theta_i x_i$$

(sake of notations, we add the variable x_0=1)

How can we find the parameters θ ? We minimise a “distance” between model and dataset (or ***cost function***):

$$\boxed{J(\theta) = \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}$$

Minimising the cost functions (steepest descent)

Minimum of cost function will provide a model to predict mileages for *unknown* battery capacity and vehicle weights (***inference***)

Initial guess for θ_j and iterate by steps in directions that decrease the cost function (following derivatives or steepest gradients)

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \longleftarrow \boxed{\frac{\partial}{\partial \theta_j} J(\theta) = (h_\theta(x) - y)\, x_j}$$

α : arbitrary parameter (***learning rate***)
α >>1 : optimisation 'jumpy'
α <<1 : large number of steps required

Iterative process:

$$\theta_j := \theta_j + \alpha \boxed{\left(y^{(i)} - h_\theta(x^{(i)})\right)} x_j^{(i)}$$

Changes in parameters according to the "***error***"

No changes / update when model is accurate.

Neural networks - perceptron model, two neurons model

Classification task - class 1 or 0

Two parameters dataset (same as battery, but now instead of mileage we predict whether commercially viable or not):

Training set		
x_1	x_2	outcome
0.8	0.3	1
0.4	0.1	0

Find coefficients ω to obtain model for the outcome:

$$\begin{aligned} z &= 0, \qquad \text{if} \qquad \sum_{i=1}^{n} x_i \omega_i \le \theta \\ z &= 1, \qquad \text{if} \qquad \sum_{i=1}^{n} x_i \omega_i > \theta \end{aligned}$$

θ is ***threshold*** value

Neural networks - perceptron model, two neurons model

Training set		
x_1	x_2	outcome
0.8	0.3	1
0.4	0.1	0

θ = 0.1
Random set of initial weights
Iteration through database - cost function

$$\boxed{\sum_{i=1}^{n} x_i \omega_i > \theta}$$

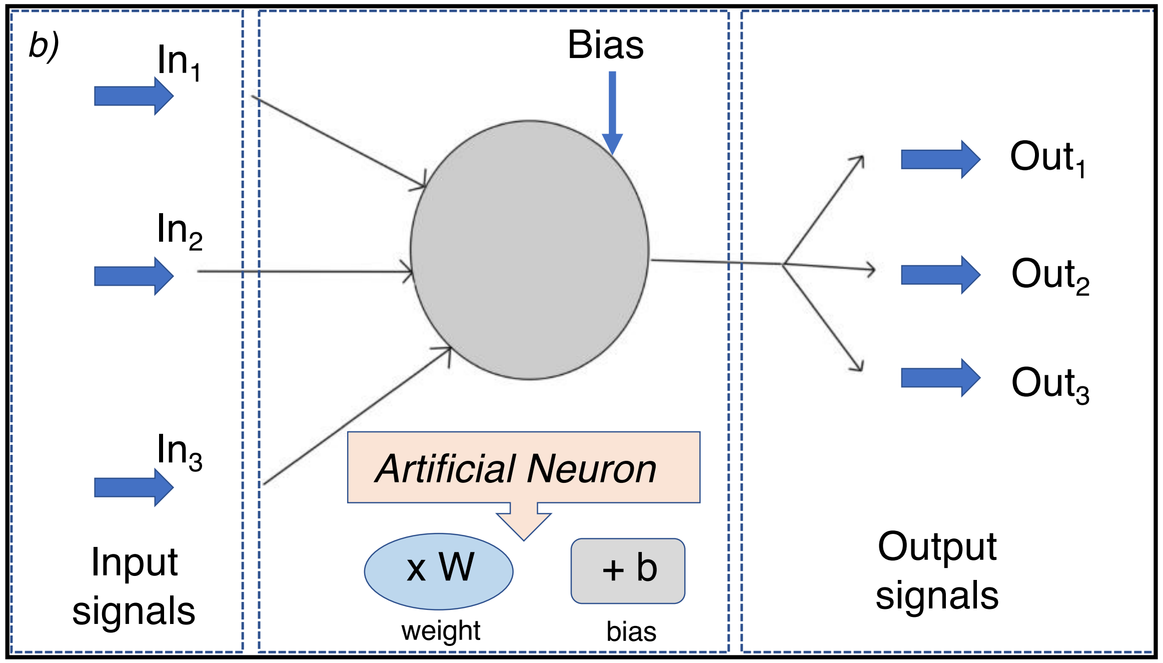
logical threshold unit or ***activation function***

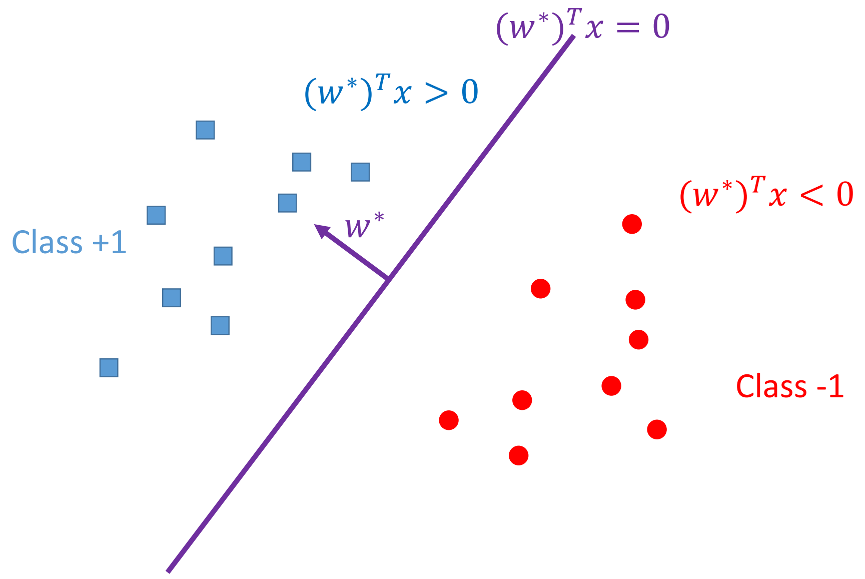
Training set						
x_1	x_2	ω_1	ω_2	Prediction $\mathcal{P}$	Dataset $\mathcal{D}$	
0.8	0.3	0.4	-0.2	1	1	$\rightarrow$ Correct
0.4	0.1	0.4	-0.2	1	0	$\rightarrow$ Wrong!

Correction to account for error, weights update:

Iterate until convergence

$$\Delta\omega_i = \alpha \left(\underset{\uparrow \text{ Training set}}{t} - \underset{\uparrow \text{ Model}}{z} \right) x_i$$





$$\boxed{\sum_{i=1}^{n} x_i \omega_i > \theta}$$

logical threshold unit or ***activation function***

Training set					
x_1	x_2	ω_1	ω_2	Prediction $\mathcal{P}$	Dataset $\mathcal{D}$
0.8	0.3	0.4	-0.2	1	1
0.4	0.1	0.4	-0.2	1	0

$\longrightarrow$ Correct

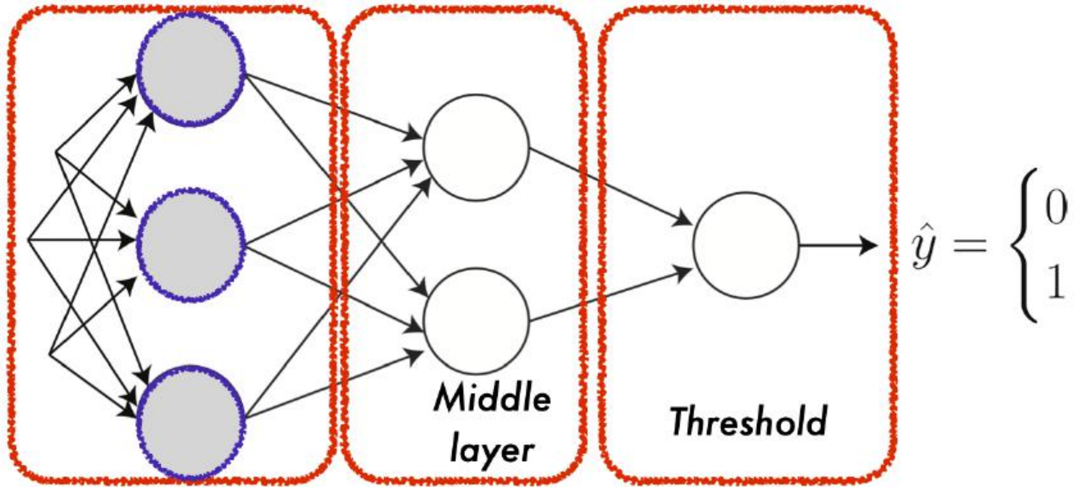
$\longrightarrow$ Wrong!

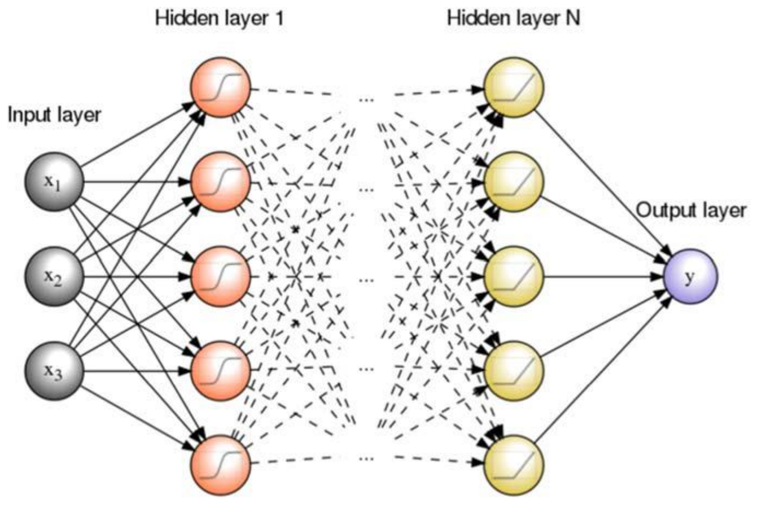
$$\Delta\omega_i = \alpha\,(\underset{\uparrow}{t} - \underset{\uparrow}{z})\,x_i$$

Model

Training set

Neural networks - general





Input & ***fully connected layer*** (=perceptron model)

Addition : intermediate neuron middle layer (***hidden layer***)

Generalisation of learning formula - for each sample in training set calculate contribution to ***cost function,*** sum over entire training set (N samples):

$$\mathcal{C} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i))$$

High confidence y=0 (cat), weak contribution

High confidence y=1 (dog), weak contribution

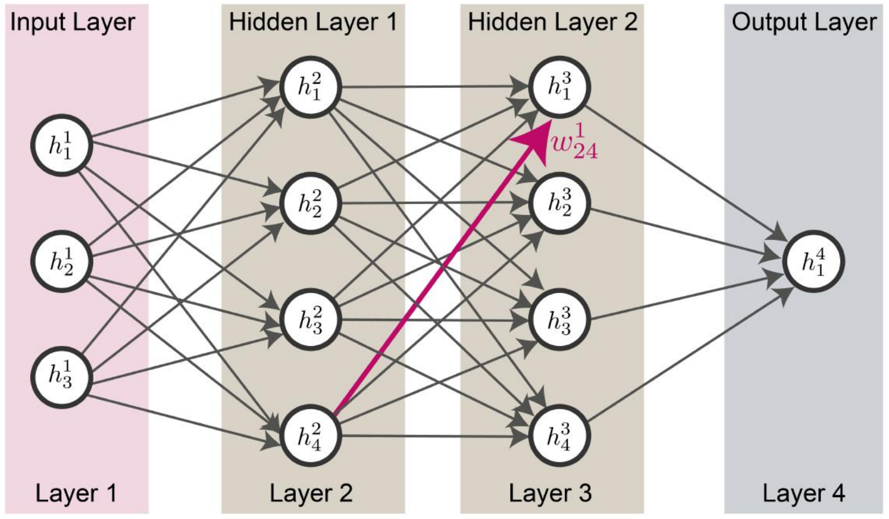
Regression cost function

$$\boxed{\mathcal{C} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Neural networks - BP/back-prop(-agation)

All is well - but now that there are multiple connections between inputs and outputs, ***how can we update weights after the cost function is evaluated***?

perceptron model: simple relation between inputs&weights, hidden layer complications



Evaluate ***activation function in each layers***:

$$h^l_j = \sigma\left[\sum_k w^l_{jk} h^{l-1}_k + b^l_j\right]$$

Final ***cost function***:

$$C = \frac{1}{2N}\sum_i^N |y_i - h^L_i|^2$$

h is a nested function:

$$h^L_i = \sigma^L\left(\sum_k w^L_{ik}\sigma^{L-1}\left(\sum_k w^L_{ik}\sigma^{L-2}[\ldots x_i \ldots] + b^{L-1}_i\right) + b^L_i\right)$$

Steepest gradient, minimise cost function with respect to weights and biases:

$$w^l_{jk} \rightarrow w^l_{jk} - \eta\frac{\partial C}{\partial w^l_{jk}}, \qquad b^l_j \rightarrow b^l_j - \eta\frac{\partial C}{\partial b^l_j},$$

Deep learning for the Anderson impurity model

Data representation

Green’s function (G$_{\text{imp}}$ or Weiss field) represented in imaginary time

Absorbing temperature dependence: $x(\tau) = \frac{2\tau}{\beta} - 1$ $[-1,+1]$

Compact representation of Green’s function, polynomial support basis (Legendre or Chebyshev):

$$G^{(k)}(\tau) = \sum_{i\geq 0} P_i^{(k)}[x(\tau)]G_i^{(k)}$$

Data preparation:

$$\begin{tikzcd} & G(\tau) = G^{\mathrm{S}}(\tau) + G^{\mathrm{AS}}(\tau) & \\ G^{\mathrm{S}}(\tau) = \sum_{\substack{l\geq 0\\ \text{even}}} \frac{\sqrt{2l+1}}{\beta} P_l[x(\tau)]G_l \arrow[ur] & & G^{\mathrm{AS}}(\tau) = \sum_{\substack{l\geq 0\\ \text{odd}}} \frac{\sqrt{2l+1}}{\beta} P_l[x(\tau)]G_l \arrow[ul] \end{tikzcd}$$

Each contribution ***can be learned separately***, asymmetric is naught away from half-filling

Particle-hole symmetry - allows ***augmenting database*** at no cost: $G^e(\tau) = G^h(\beta - \tau)$

Activation functions for DMFT

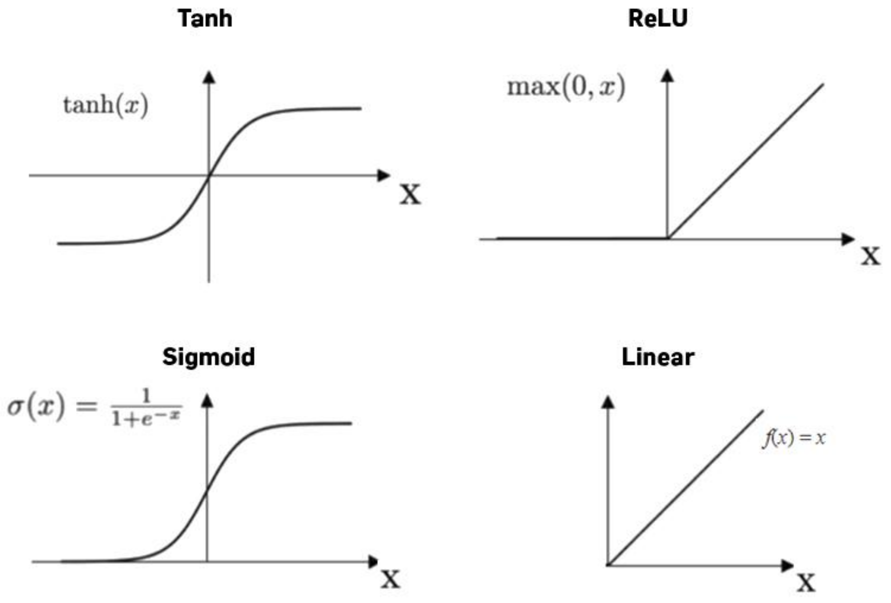
Mapping/transforming input data, example Legendre coefficients:

	$\mathcal{T}_l$	$\mathcal{T}_l^{-1}$
$\mathcal{T}_0$	G_l	G_l
$\mathcal{T}_1$	$\tanh(G_l)$	$\tanh^{-1}(G_l)$
$\mathcal{T}_2$	$-\tanh(G_l)$	$-\tanh^{-1}(G_l)$

Negative and positive entries not treated on same footing - activation functions not symmetric in general

From physical coefficients to intermediate representation layer (reversible transformation)

Helpful to map inputs to a scale suitable for ***activation functions*** (and avoids network being dominated by large weights)



Learning corrections to known approximations

Instead of predicting Green's functions of the AIM for wide range of parameters, we learn the error or corrections of known approximations to the exact result

less ambitious - but requires *fast solvers*

Library of solvers for ML : Hubbard-I (**H1**), Iterative perturbation theory (**IPT**), Exact diagonalisation (Nb~2,3,4) (**ED-Nb**)

ML : model for corrections to known approximation

Solvers used individually, or ***collectively*** (input and output vector x m_{solver})

Motivation: combining approximations obtained from different limits, interpolation

Cost function:

Set of exact solutions of AIM

$$C(\mathbf{X}, \mathbf{Y}, \boldsymbol{\alpha}) = \frac{1}{N_s} \sum_j^{N_s} [\mathbf{y}_j - g_{\boldsymbol{\alpha}}(\mathbf{x}_j)]^2$$

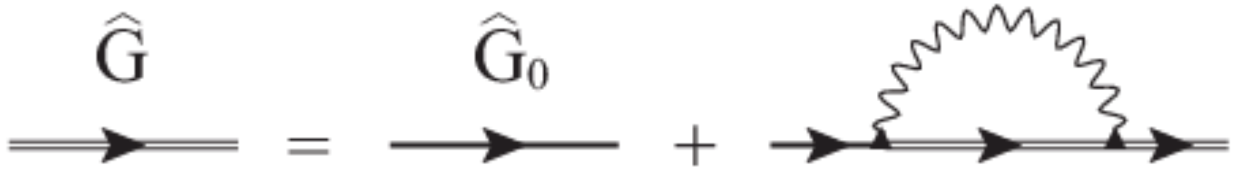
Set of approximations of AIM

Training set: ED with Nb=8 (or CTQMC)

Model that provides corrections to known approximated solutions

Learning corrections to known approximations

Dyson equation : $G_0^{-1}(i\omega_n) = \Sigma(i\omega_n) + G^{-1}(i\omega_n)$



Learning self-energy = learning corrections from free GF

Physically insightful : we know the free GF analytically

But not meaningful for the network : offers a generalisation, learn corrections to an effective theory as a starting point, any theory valid.

Absorbs steep behaviour in self energies in the reference theory. Example, IPT:

Reference approximate model

$$\Sigma^{IPT}(i\omega_n) = \Sigma^1(i\omega_n) + \Sigma^2(i\omega_n)$$
$$= \frac{U}{2} + U^2 \int_0^\beta d\tau e^{i\omega_n \tau} G_0^2(\tau) G_0(-\tau)$$

correction ML: Learn Σ^3 , not Σ

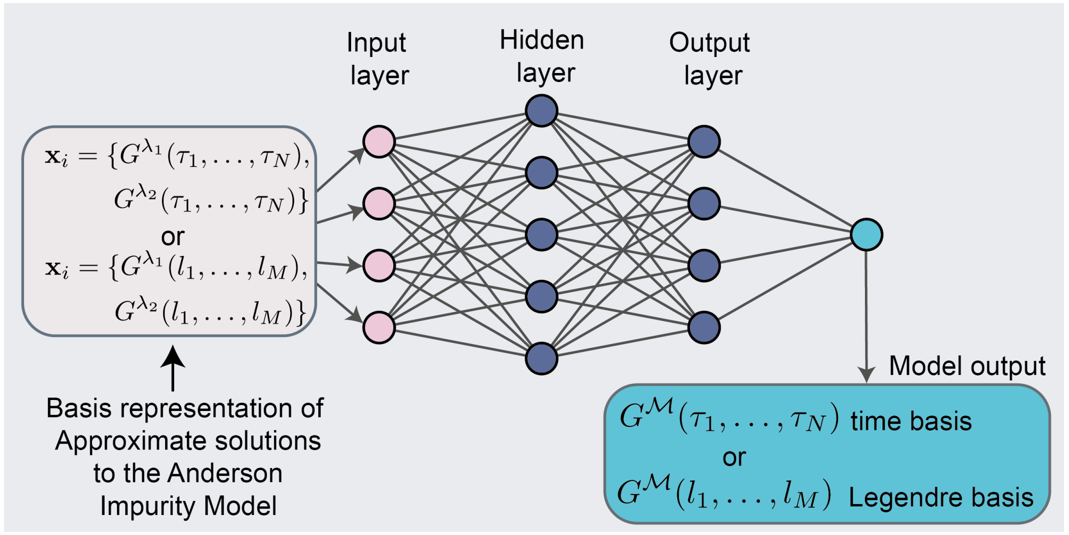
$$\Sigma(i\omega_n) = \Sigma^1(i\omega_n) + \Sigma^2(i\omega_n) \boxed{+ \Sigma^3(i\omega_n)}$$

Neural networks for AIM

Database ~ 10'000

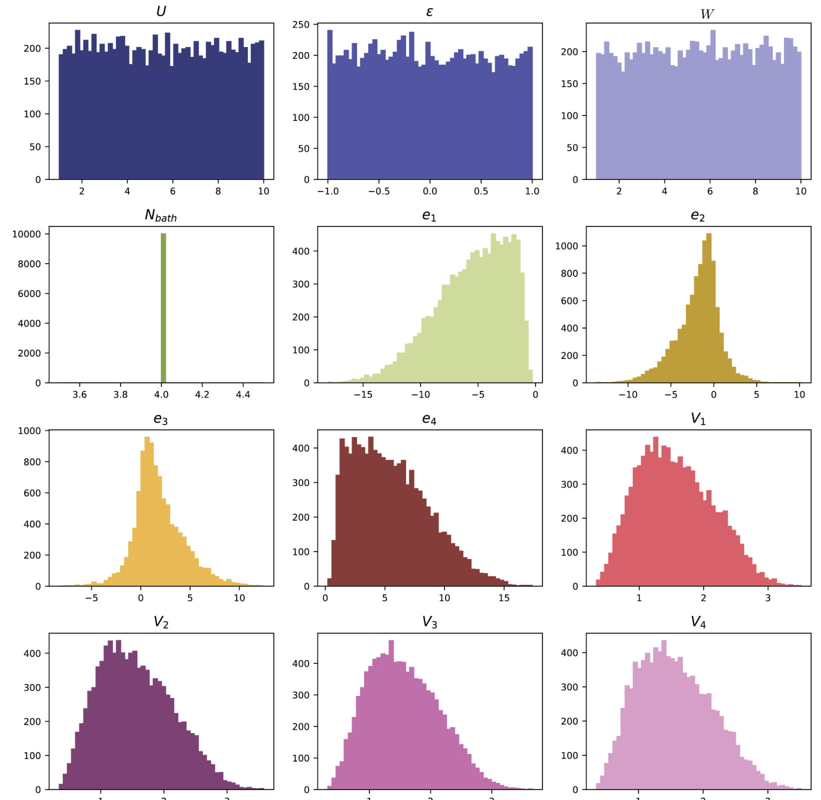
Parameter range:

U (eV)	$\{1, \ldots, 10\}$
N_{bath}, ϵ_i, V_i	4
W (eV)	$\{1, \ldots, 10\}$
ε	$\{-1, \ldots, 1\}$
β (eV^{-1})	$\{1, 2, 5, 10, 20, 50\}$
N_{samples}	10,000
$\mathcal{S}$	Hubbard-I, IPT, NCA, ED-[1,2,3]


Input layer
Hidden layer
Output layer
$\mathbf{x}_i = \{G^{\lambda_1}(\tau_1, \ldots, \tau_N), G^{\lambda_2}(\tau_1, \ldots, \tau_N)\}$
or
$\mathbf{x}_i = \{G^{\lambda_1}(l_1, \ldots, l_M), G^{\lambda_2}(l_1, \ldots, l_M)\}$
Basis representation of Approximate solutions to the Anderson Impurity Model
Model output
$G^{\mathcal{M}}(\tau_1, \ldots, \tau_N)$ time basis
or
$G^{\mathcal{M}}(l_1, \ldots, l_M)$ Legendre basis

$$N_\alpha = \underbrace{20(200+1)}_{\text{input layer}} + \underbrace{20(20+1)}_{\text{hidden layer}} + \underbrace{100(20+1)}_{\text{output layer}} = \underbrace{6540}_{\text{total}}$$

Generate random parameters for training set, here with ED and Nb=4.



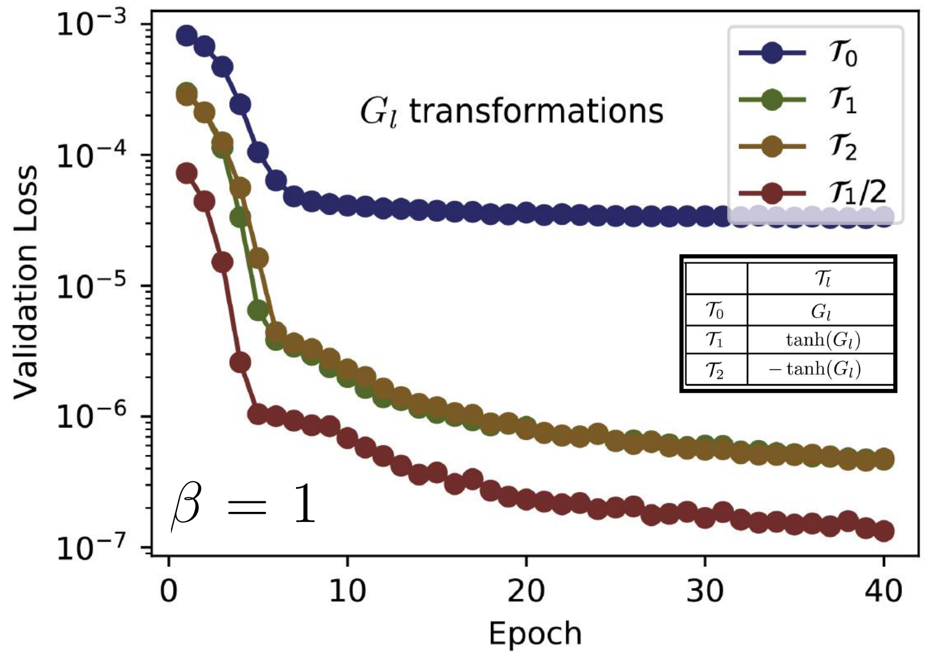
Training : 80% of database

Validation : 20% of database

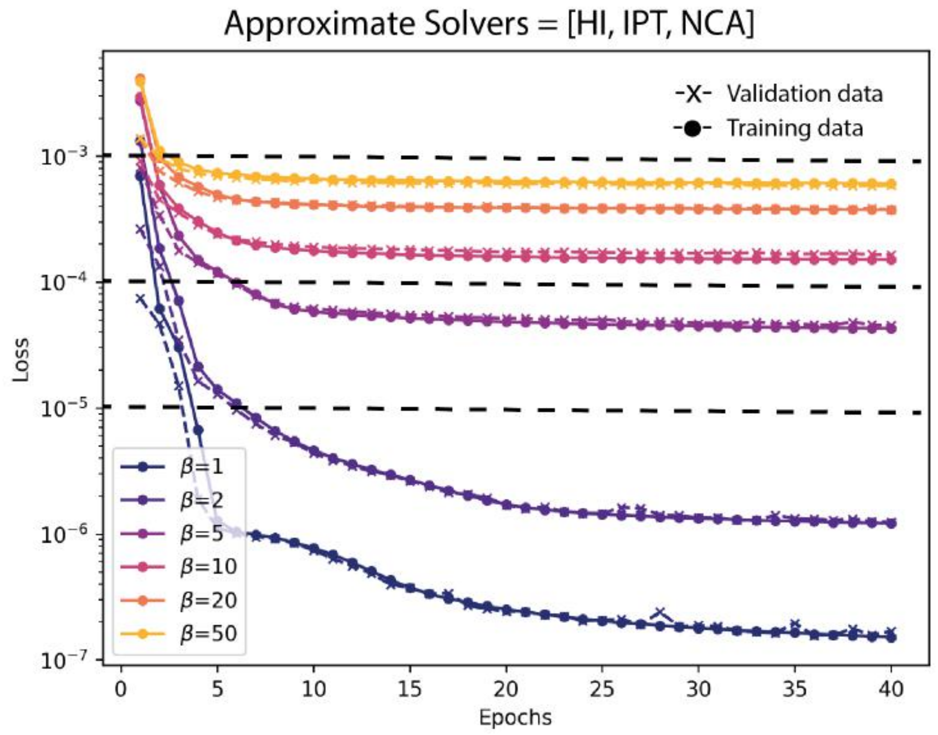
Provide mean to test the network on reference data that aren't include into the training set (inference)

Hyperparameter gridsearch - optimising the network

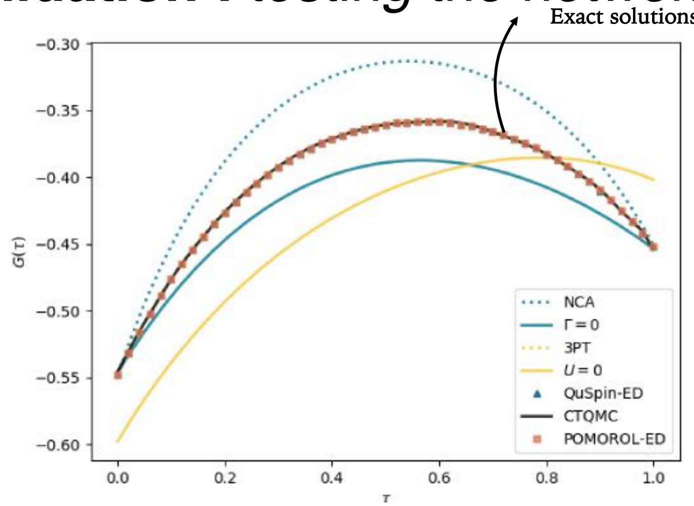
data transformation : *transforming inputs in adequate format for activation*

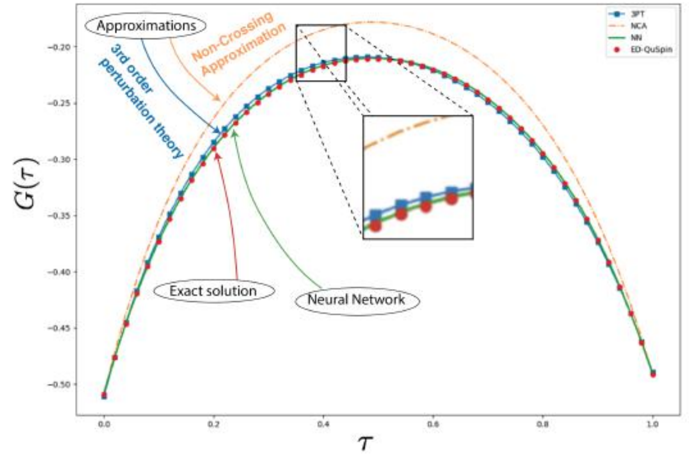


data augmentation : *combining approximate solvers*



validation : *testing the network*





Results Adaptive tau mesh learning loss functions

Target solution = ED-4 (4 bath sites)

Database size: 10,000 samples

Training data: 9,000 samples

Validation data: 1,000 samples

Beta (inverse temperature)

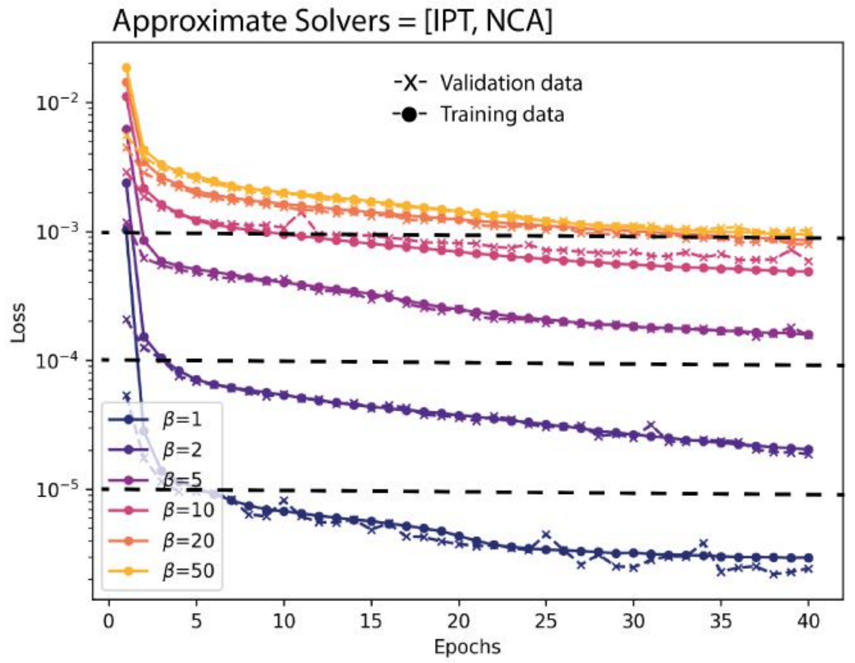
1, 2, 5, 10, 20, 50

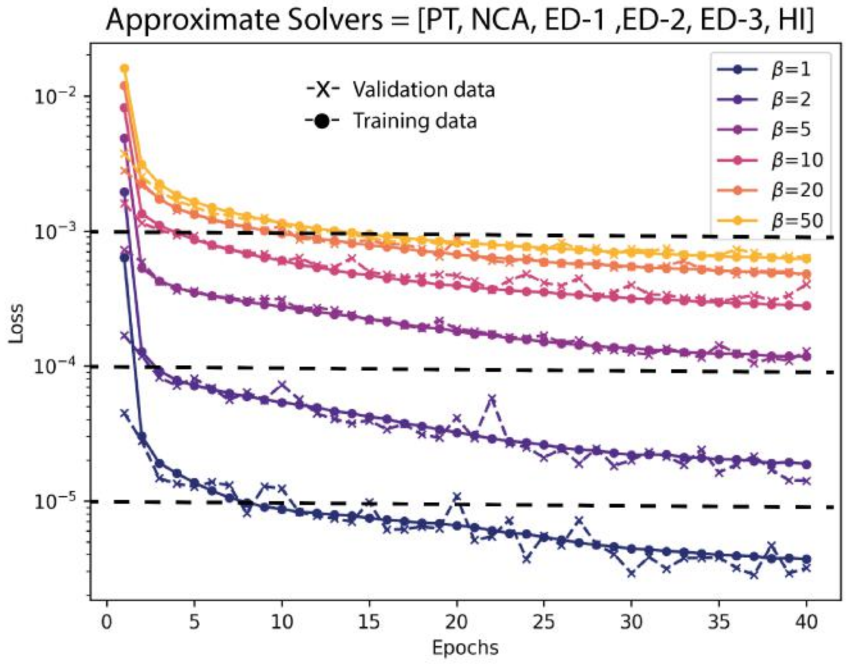
Impurity solvers:

ED-1, ED-2, ED-3, IPT, NCA, HI

Neural Network

- Fully connected
- 2 layers
- 51 neurons per layer
- Tanh activations
- Learning rate = 0.0002
- Batch size = 8
- Adaptive tau mesh = 51
- 90/10 data split



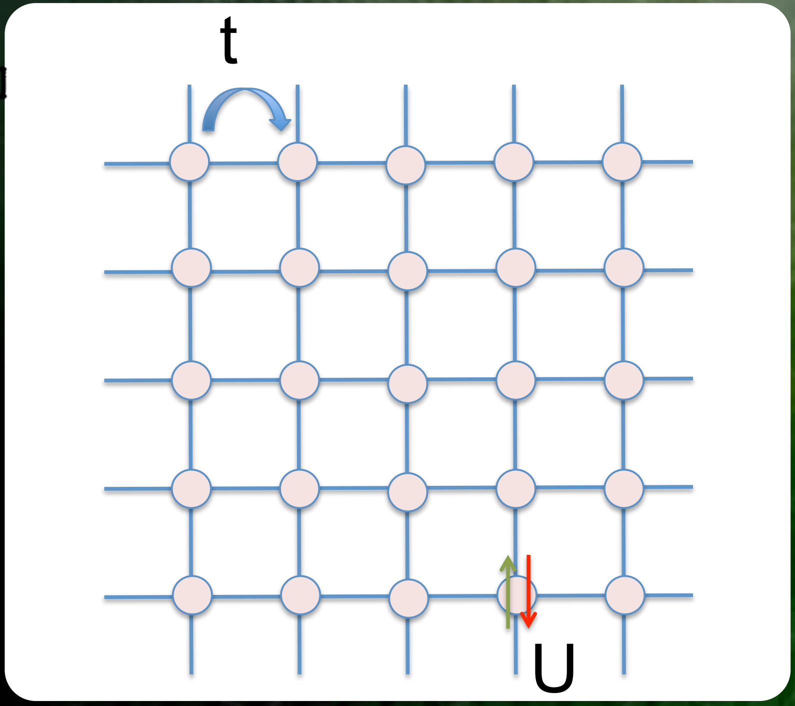


Data-driven approach to the Mott transition

Hubbard model : Coulomb repulsion U

$$H_{Hubbard} = -t \sum_{\langle i,j\rangle,\sigma} \left(c^{\dagger}_{i\sigma} c_{j\sigma} + h.c. \right) + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$

- One band crossing the Fermi level
- tunneling/transfer integral “t”
- Hilbert space 4^N, simple theory, but hard to solve.

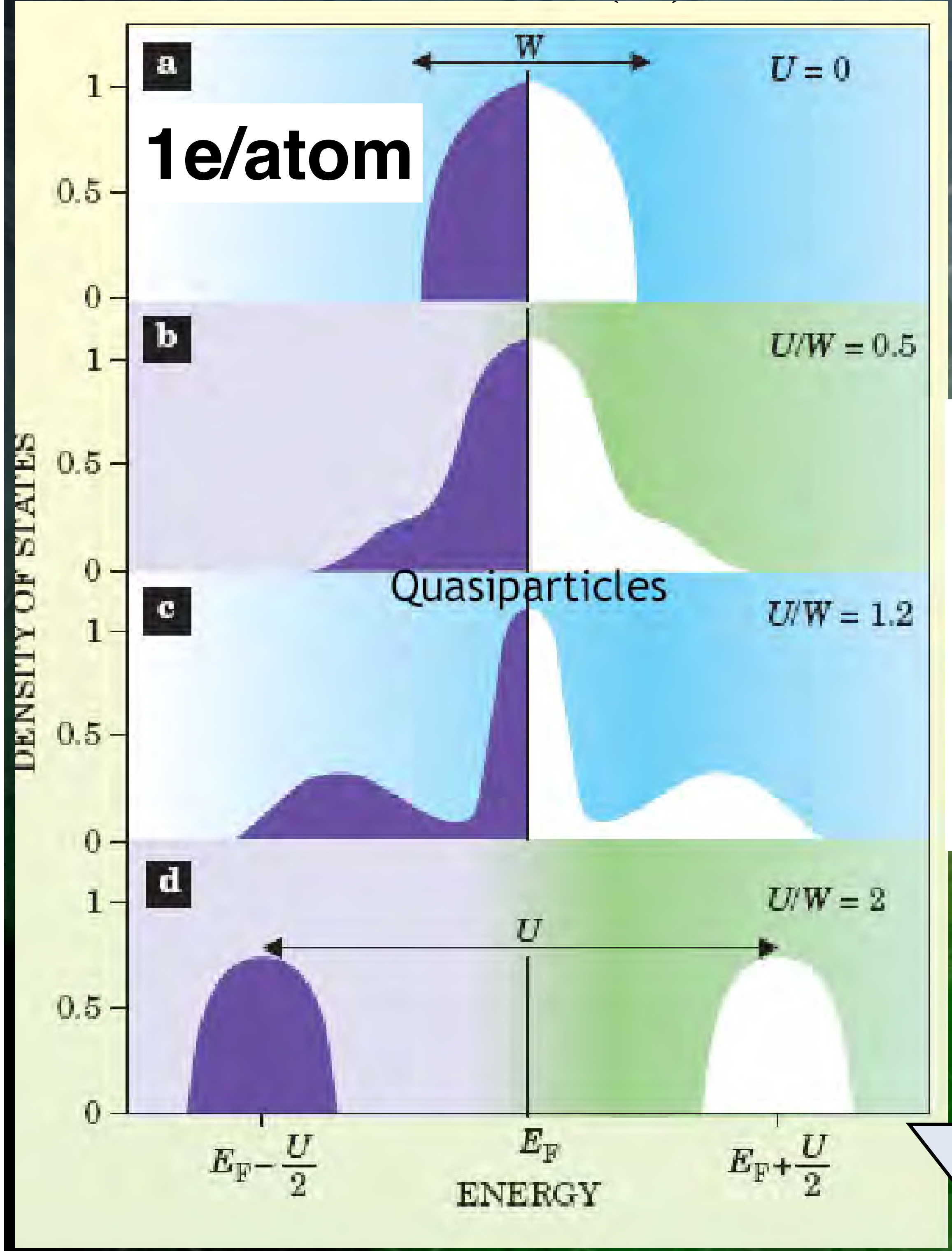


- **Metal to insulator transition**:

U<<1: paramagnetic Metal

U>>1 : Mott insulator

too simple but contains most of the physics



increasing U

Metal

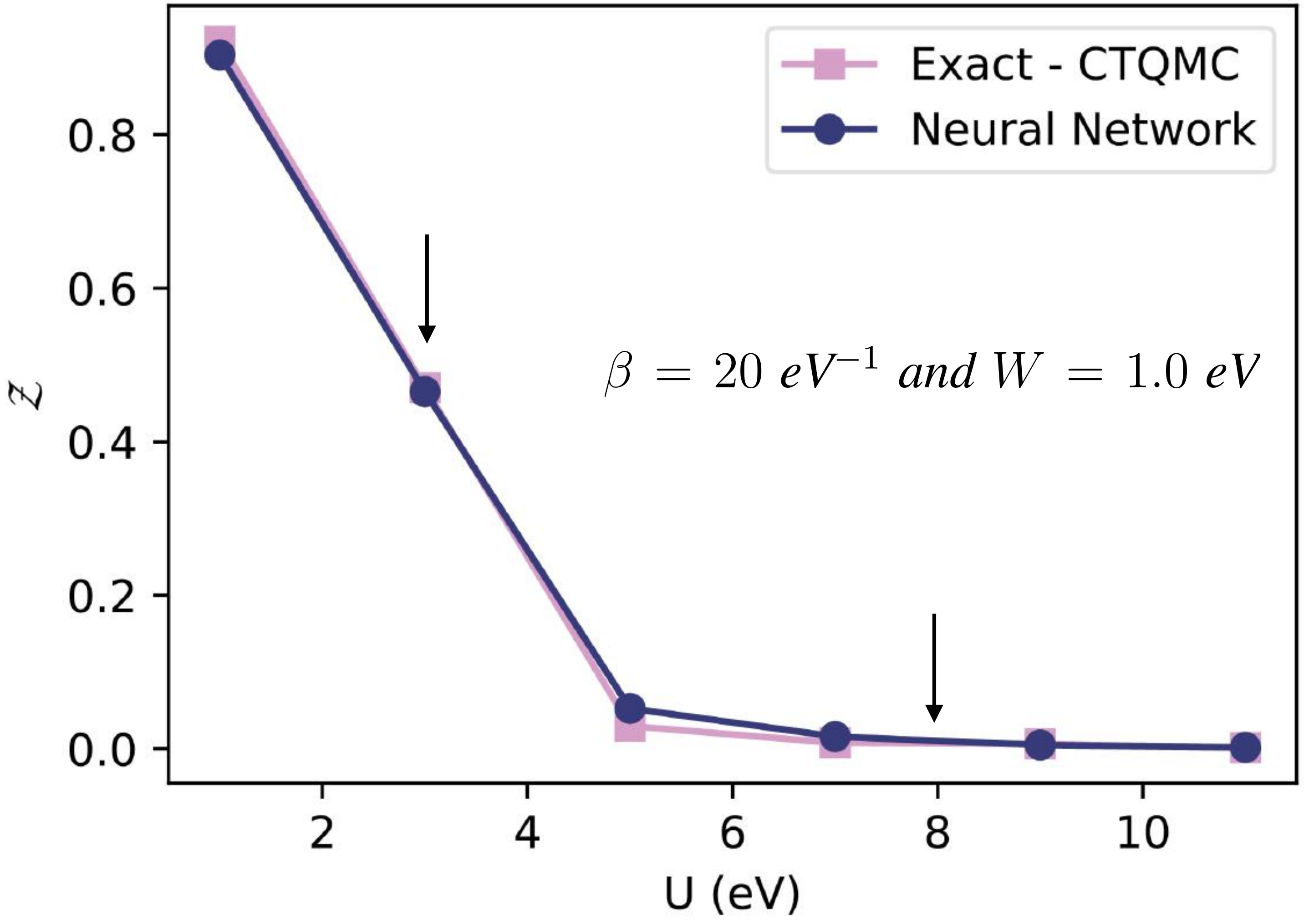
IF THESE IDIOTS WOULD JUST TAKE THE BUS, I COULD BE HOME BY NOW
IF THESE IDIOTS WOULD JUST TAKE THE BUS, I COULD BE HOME BY NOW
KIRK

insulator

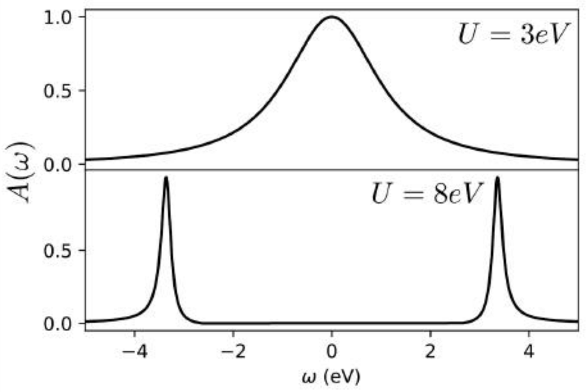
A. Georges and G. Kotliar, PRB (1992)
A. Georges et al., RMP (1996)

Hallmark of the Mott transition, quasi-particle weight

Test of NeuraNet on Bethe lattice at half-filling ***: Full DMFT iteration until convergence***



NeuraNet + Pade



Software download and testing

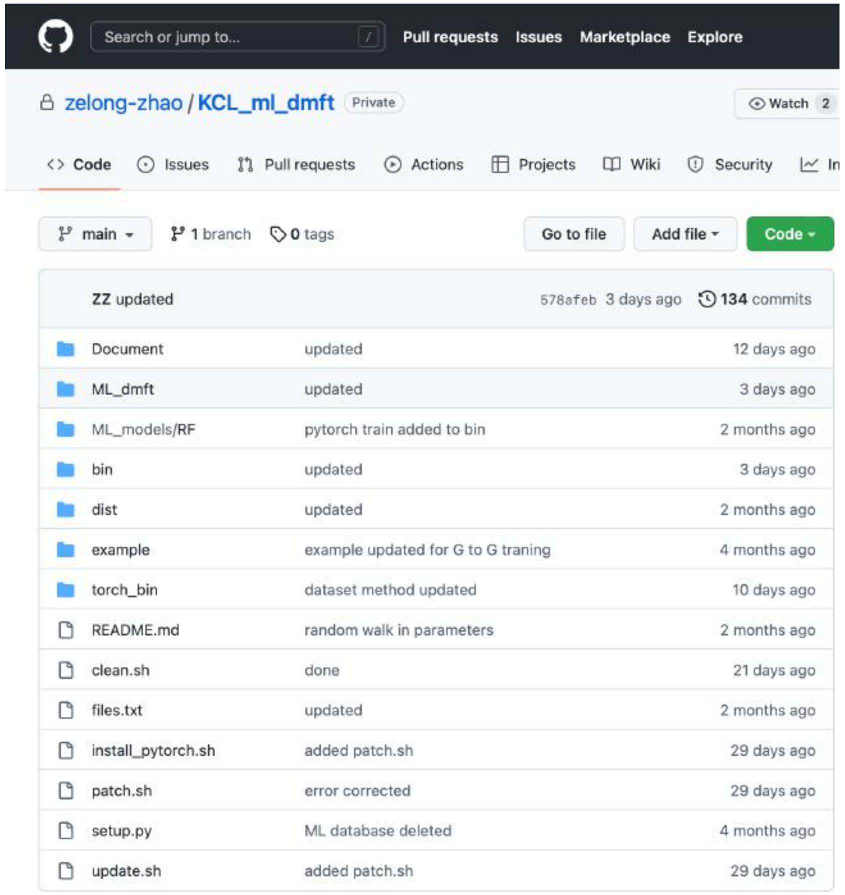
Github link : https://github.com/zelong-zhao/KCL_ml_dmft

Code development: Evan Sheridan (@phasecraft) and currently maintained/ developed further by Zelong Zhao (@KCL)

Linux : installation via *conda*

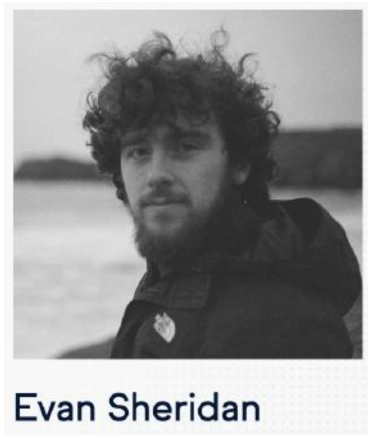
questions, pull request or contribute ->

zelong.zhao@kcl.ac.uk
cedric.weber@kcl.ac.uk



ZZ updated 578afeb 3 days ago 134 commits

Document	updated	12 days ago
ML_dmft	updated	3 days ago
ML_models/RF	pytorch train added to bin	2 months ago
bin	updated	3 days ago
dist	updated	2 months ago
example	example updated for G to G traning	4 months ago
torch_bin	dataset method updated	10 days ago
README.md	random walk in parameters	2 months ago
clean.sh	done	21 days ago
files.txt	updated	2 months ago
install_pytorch.sh	added patch.sh	29 days ago
patch.sh	error corrected	29 days ago
setup.py	ML database deleted	4 months ago
update.sh	added patch.sh	29 days ago



Zelong Zhao

What’s next ?

1. ***Feature layers, variational encoders*** : Compress information by using diminishing hidden layers (alternative to Legendre representation)

2. **Geometrical conformation** : use geometrical constraints on Green’s function, e.g. convex, smooth, angles etc… Inspired from image classification

3. ***Dynamic database*** : super-perturbation theory, adapt automatically approximate solver entries with corrections provided by DMFT hybridisation (database adapt dynamically)

4. ***Beyond deep learning*** : Generative adversarial network (GAN), use another network to arbitrate the learning of Green’s functions - "indirect" training through another neural network that can tell how "realistic" the input seems, for instance to discriminate between the choice of approximate solvers

Conclusions

- Digital design - a need for accelerated many body calculations / engines
- Data driven approaches - error correction techniques
- Scope for very large speed-up and opens up new possibilities (material screening, MD, …)
- Work in progress - feature layers, VAE, …

Thank you!

Q&A