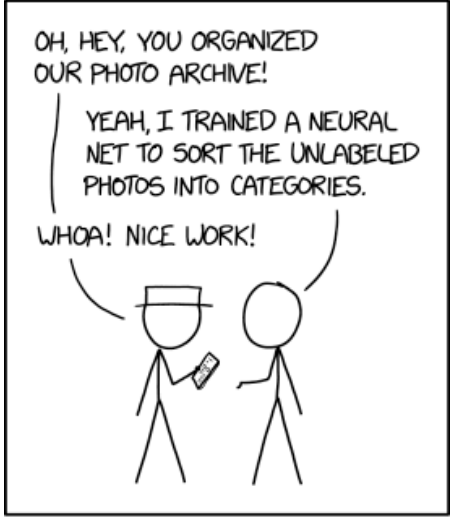


ENGINEERING TIP:
WHEN YOU DO A TASK BY HAND, YOU CAN TECHNICALLY SAY YOU TRAINED A NEURAL NET TO DO IT.

Applications of machine learning to materials science

Sai Gautam Gopalakrishnan, Reshma Devi, Dereje Bekele Tekliye and Aqshat Seth

Simulations And Informatics of MATerials (SAI-MAT) group

Materials Engineering, Indian Institute of Science

saigautamg@iisc.ac.in; https://sai-mat-group.github.io

Namma Psi-k Workshop
Jul 25, 2023

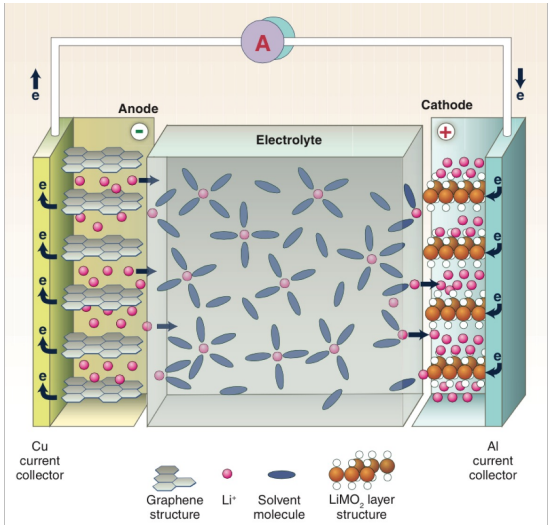
Acknowledgments

Group picture, May 2023

Dereje Aqshat Reshma

Why bother about materials science?

Key performance bottlenecks in key applications: governed by materials used



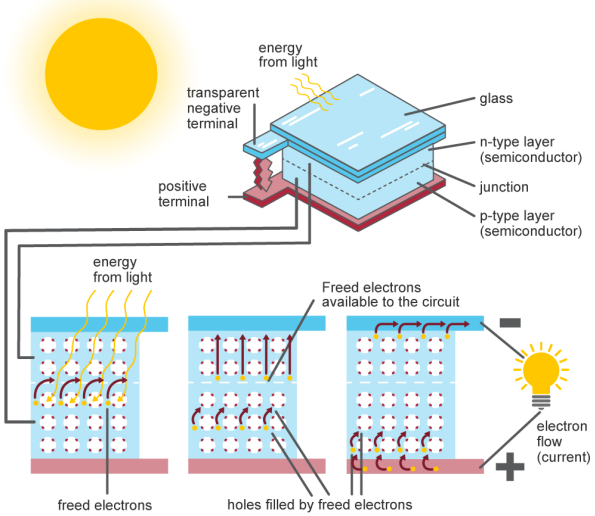
B. Dunn et al., Science 2011

Energy and power density of a battery: limited by materials used as electrodes (and at times, electrolytes)

Key material properties: stability, ionic mobility, reaction energies

Usage of better materials → better performance

Inside a photovoltaic cell



Source: U.S. Energy Information Administration

Efficiency of a photovoltaic: choice of semiconductor used as the light absorber

Key material properties: band gap, stability, resistance to point defects

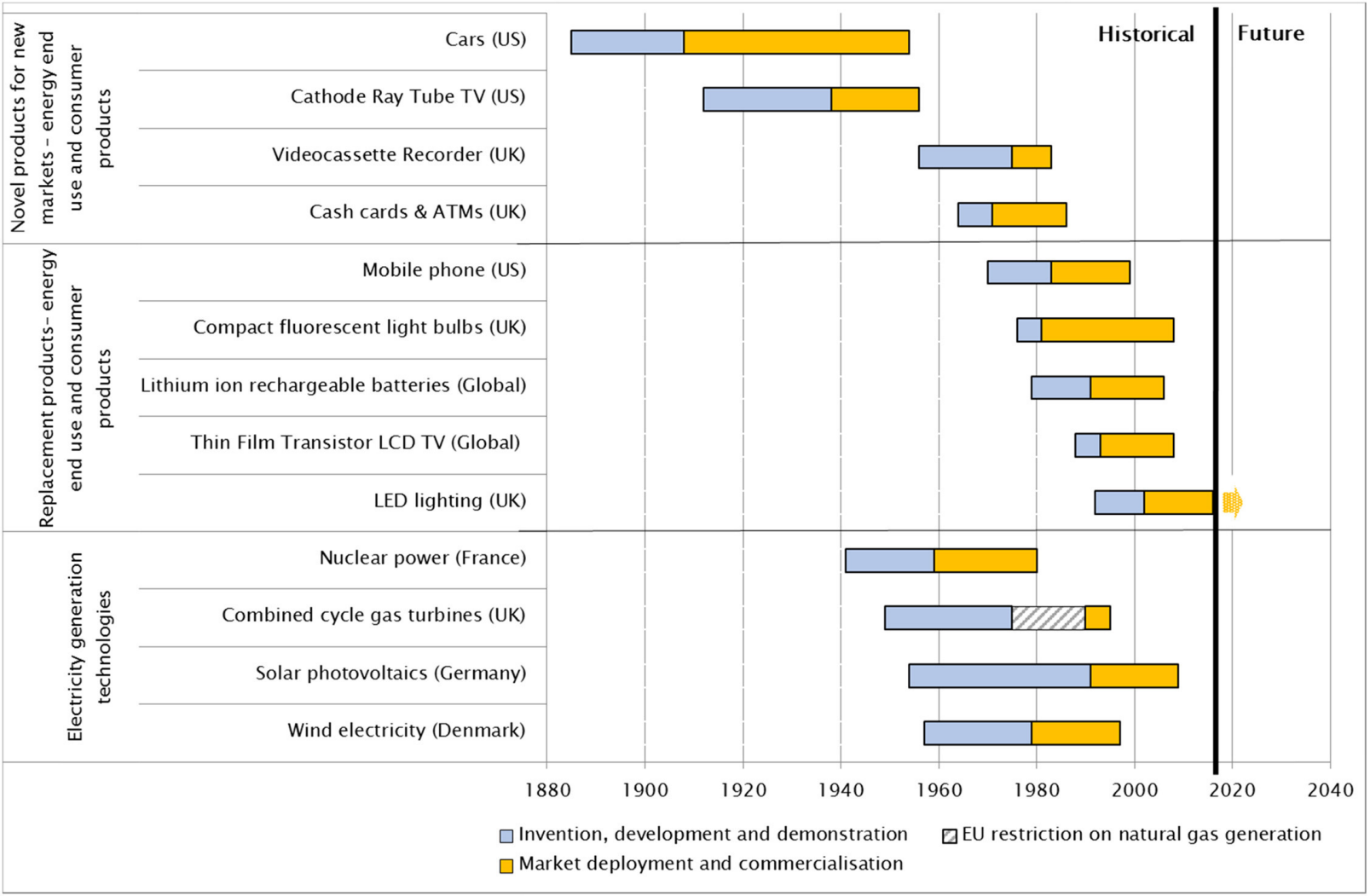
Why use machine learning (ML) in materials science?

Technological innovation and deployment is a ‘slow’ process: often limited by materials

Gross et al., Energy Policy 123, 682-699 (2018)

Why use machine learning (ML) in materials science?

Technological innovation and deployment is a ‘slow’ process: often limited by materials

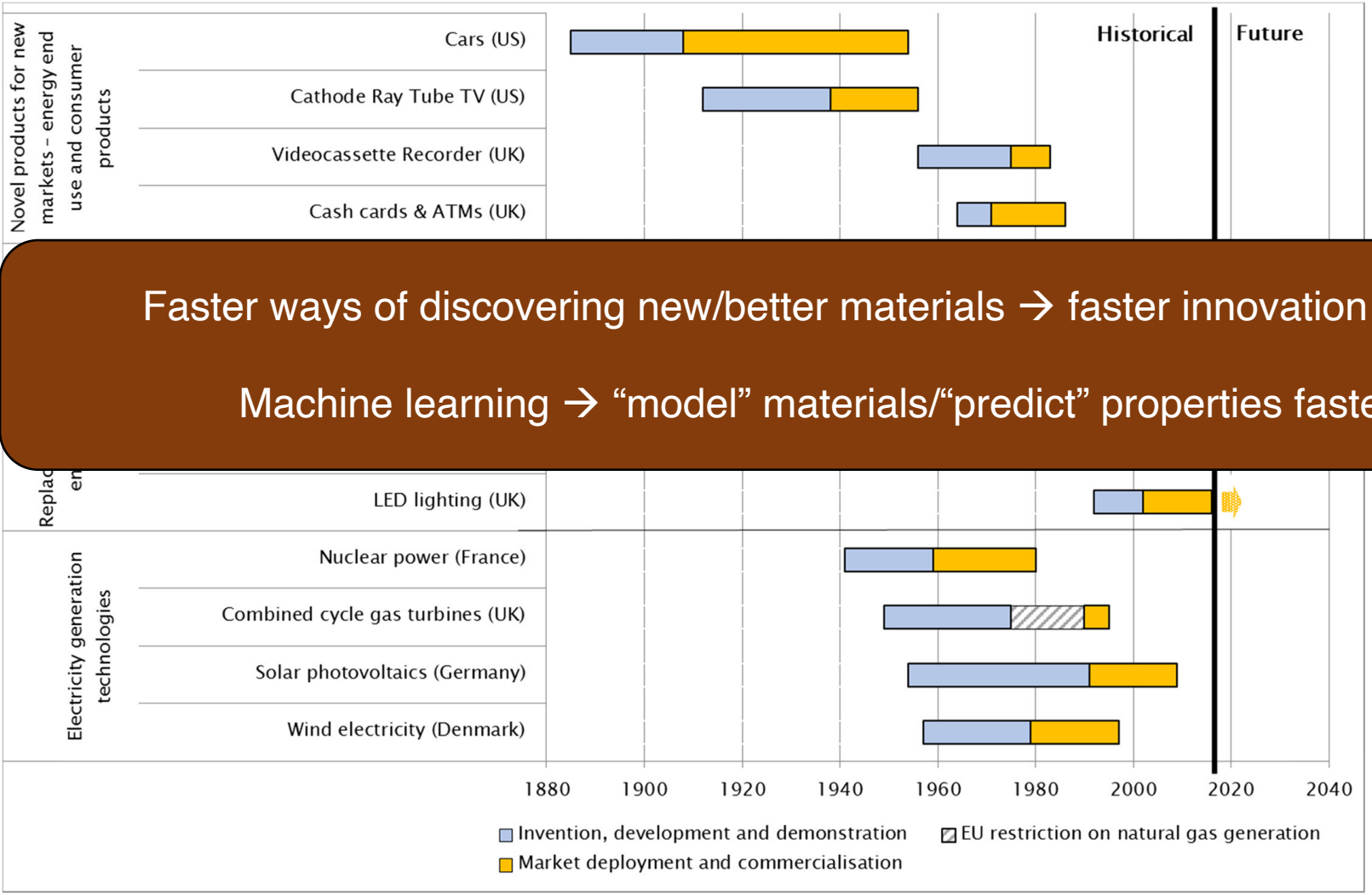


Innovation is particularly slow in energy generation sector!

Gross et al., Energy Policy 123, 682-699 (2018)

Why use machine learning (ML) in materials science?

Technological innovation and deployment is a 'slow' process: often limited by materials



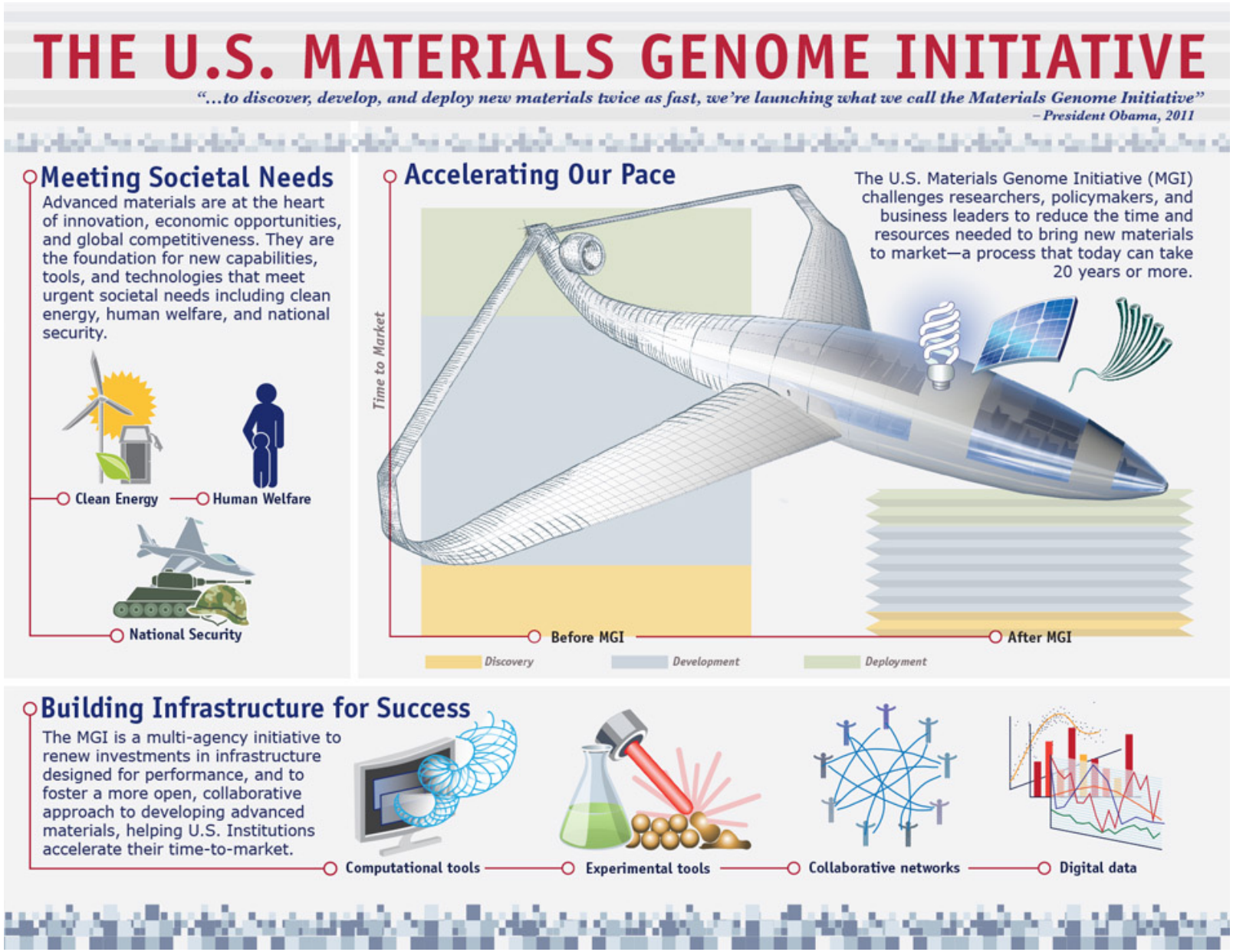
Faster ways of discovering new/better materials → faster innovation cycles

Machine learning → "model" materials/"predict" properties faster

Innovation is particularly slow in energy generation sector!

Gross et al., Energy Policy 123, 682-699 (2018)

Materials Genome (2011-present)



Evolution of ‘modelling’ in materials science

On the determination of molecular fields. —II. From the equation of state of a gas

J. E. Jones

Published: **01 October 1924** | https://doi.org/10.1098/rspa.1924.0082

1924

1957

RESEARCH ARTICLE | AUGUST 13 2004

Phase Transition for a Hard Sphere System

Special Collection: JCP 90 for 90 Anniversary Collection

B. J. Alder; T. E. Wainwright

Check for updates

J. Chem. Phys. 27, 1208–1209 (1957)

https://doi.org/10.1063/1.1743957 **Article history**

Inhomogeneous Electron Gas

P. Hohenberg and W. Kohn

Phys. Rev. **136**, B864 – Published 9 November 1964

1964

1975

Clustering and ordering in solid solutions

D. de Fontaine

Computer simulation of local order in condensed phases of silicon

Frank H. Stillinger and Thomas A. Weber

Phys. Rev. B **31**, 5262 – Published 15 April 1985; Erratum Phys. Rev. B **33**, 1451 (1986)

1986

1996

Generalized Gradient Approximation Made Simple

John P. Perdew, Kieron Burke, and Matthias Ernzerhof

Phys. Rev. Lett. **77**, 3865 – Published 28 October 1996; Erratum

From ultrasoft pseudopotentials to the projector augmented-wave method

G. Kresse and D. Joubert

Phys. Rev. B **59**, 1758 – Published 15 January 1999

1999

Shock Waves in High-Energy Materials: The Initial Chemical Events in Nitramine RDX

Alejandro Strachan, Adri C. T. van Duin, Debashis Chakraborty, Siddharth Dasgupta, and William A. Goddard, III

Phys. Rev. Lett. **91**, 098301 – Published 28 August 2003

2003

Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces

Jörg Behler and Michele Parrinello

Phys. Rev. Lett. **98**, 146401 – Published 2 April 2007

2007

Predicting Crystal Structures with Data Mining of Quantum Calculations

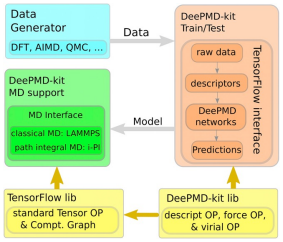
Stefano Curtarolo, Dane Morgan, Kristin Persson, John Rodgers, and Gerbrand Ceder

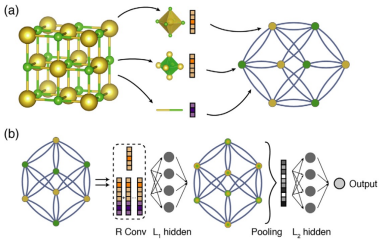
Phys. Rev. Lett. **91**, 135503 – Published 24 September 2003

THE U.S. MATERIALS GENOME INITIATIVE

2011-2018

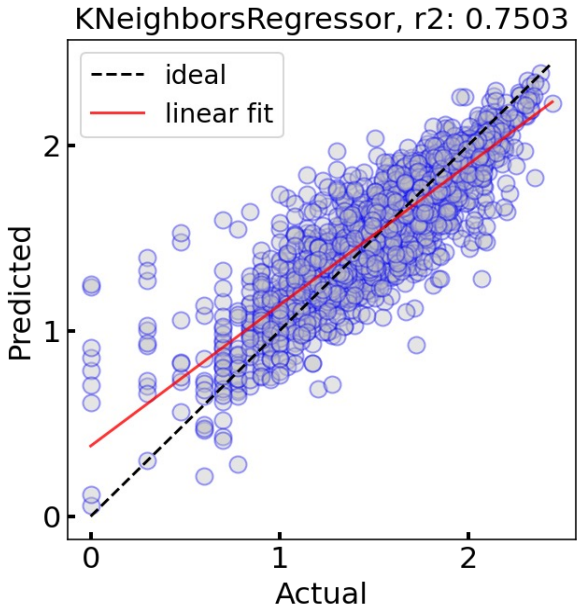
2018-present

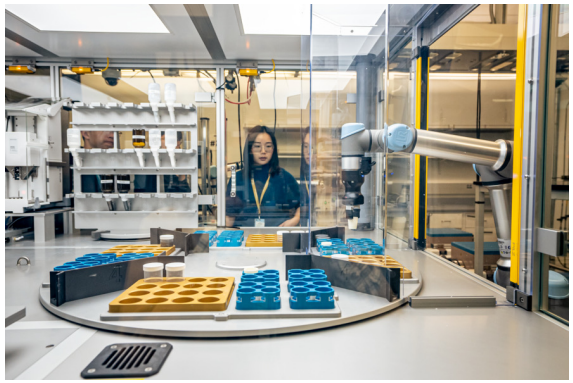




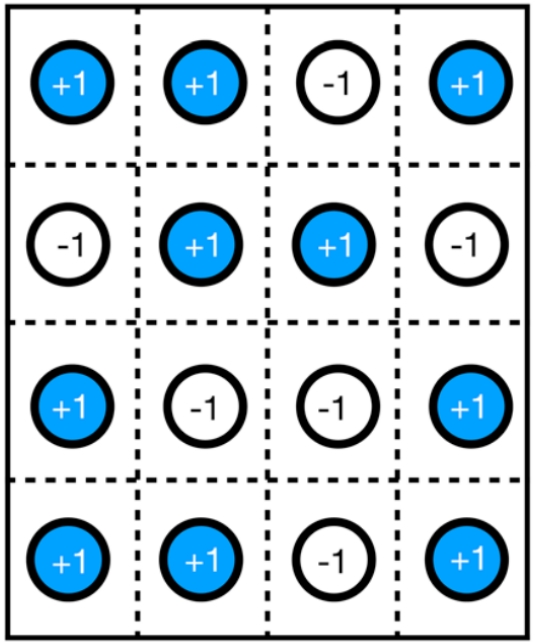
Types of ML in materials science

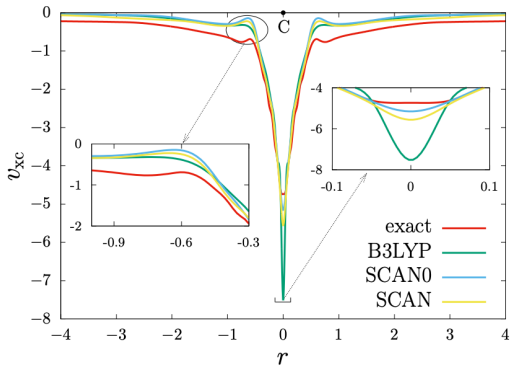
Regressions: make property predictions better with 'simple' inputs
(also classifications)



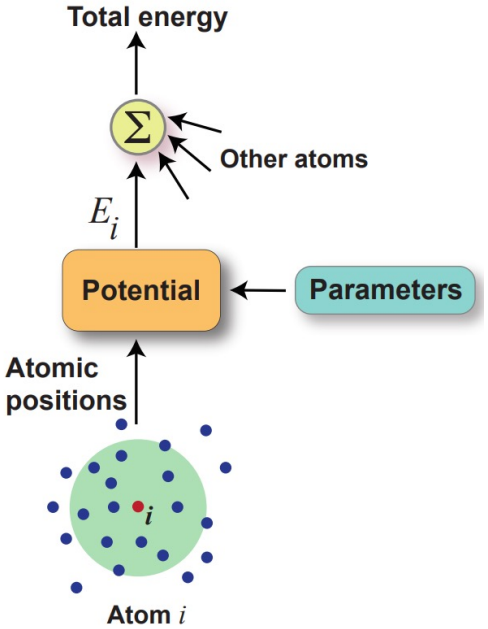


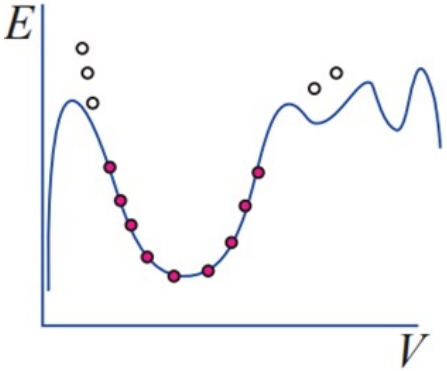
Coarse graining: create 'simple' models to mimic properties of larger lattice(s)





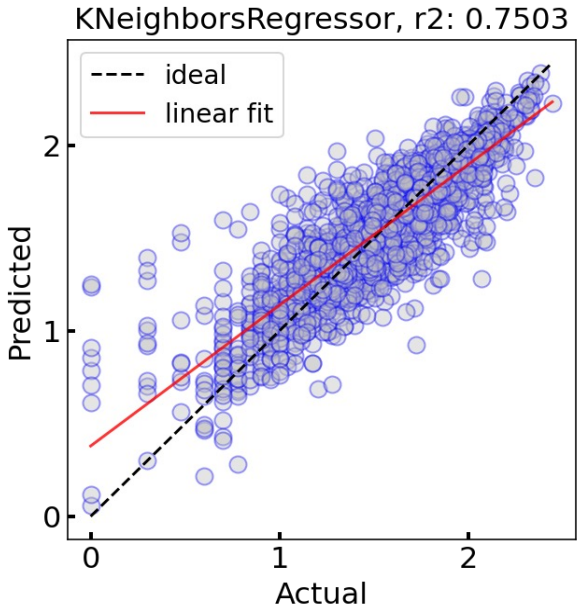
Interatomic potentials: describe potential energy surface accurately



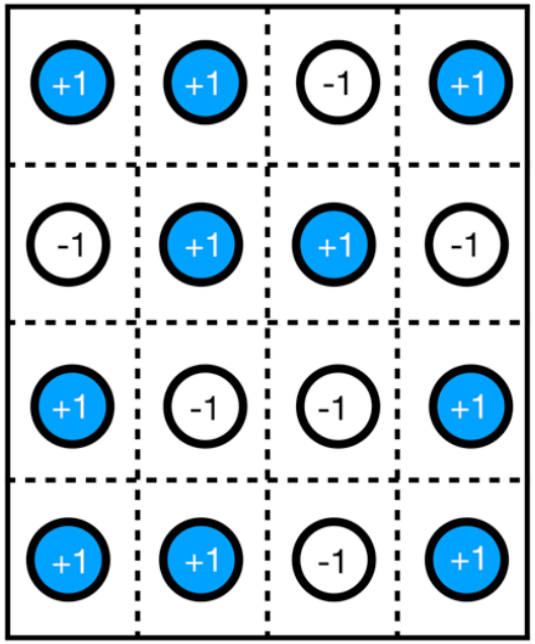


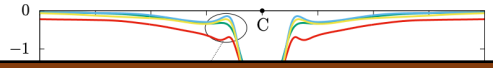
Types of ML in materials science

Regressions: make property predictions better with 'simple' inputs
(also classifications)

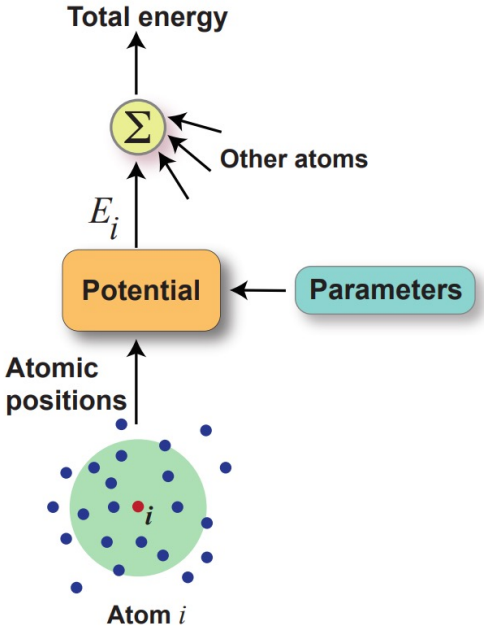


Coarse graining: create 'simple' models to mimic properties of larger lattice(s)





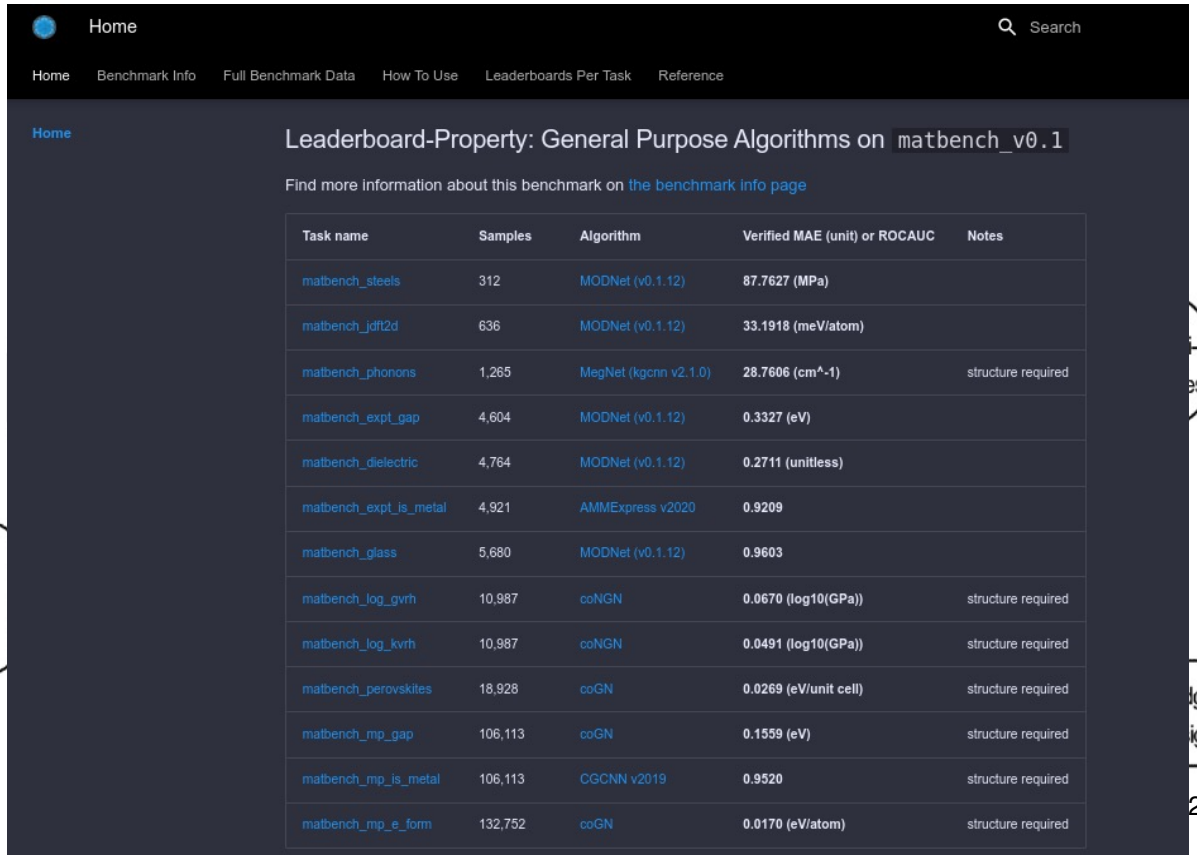
Interatomic potentials: describe potential energy surface accurately



This is not the complete classification: language models, transfer- or reinforcement-learned models, artificial intelligence (AI) models, etc.

Where does the data come from?

Optimization design and scale-up



Home

Search

Home | Benchmark Info | Full Benchmark Data | How To Use | Leaderboards Per Task | Reference

Home

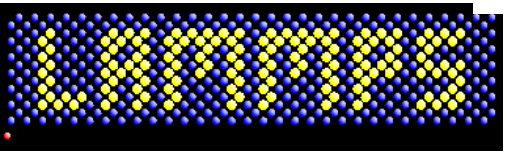
Leaderboard-Property: General Purpose Algorithms on `matbench_v0.1`

Find more information about this benchmark on the benchmark info page

Task name	Samples	Algorithm	Verified MAE (unit) or ROCAUC	Notes
matbench_steels	312	MODNet (v0.1.12)	87.7627 (MPa)	
matbench_jdft2d	636	MODNet (v0.1.12)	33.1918 (meV/atom)	
matbench_phonons	1,265	MegNet (kgcnn v2.1.0)	28.7606 (cm^-1)	structure required
matbench_expt_gap	4,604	MODNet (v0.1.12)	0.3327 (eV)	
matbench_dielectric	4,764	MODNet (v0.1.12)	0.2711 (unitless)	
matbench_expt_is_metal	4,921	AMMExpress v2020	0.9209	
matbench_glass	5,680	MODNet (v0.1.12)	0.9603	
matbench_log_gvrh	10,987	coNGN	0.0670 (log10(GPa))	structure required
matbench_log_kvrh	10,987	coNGN	0.0491 (log10(GPa))	structure required
matbench_perovskites	18,928	coGN	0.0269 (eV/unit cell)	structure required
matbench_mp_gap	106,113	coGN	0.1559 (eV)	structure required
matbench_mp_is_metal	106,113	CGCNN v2019	0.9520	structure required
matbench_mp_e_form	132,752	coGN	0.0170 (eV/atom)	structure required

2019)

https://matbench.materialsproject.org/



Data organization: python/API

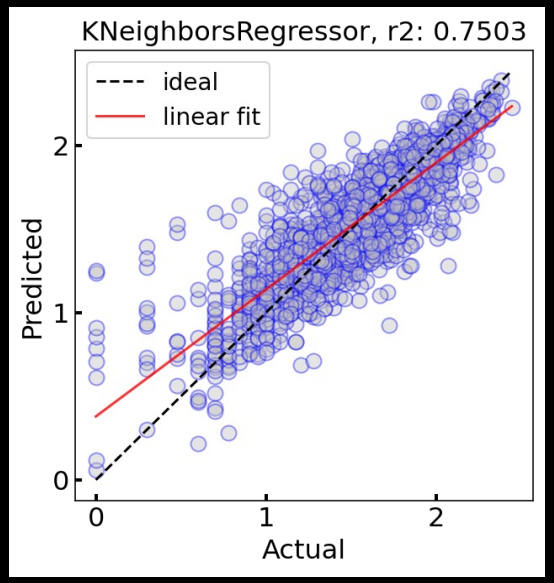
ML: python

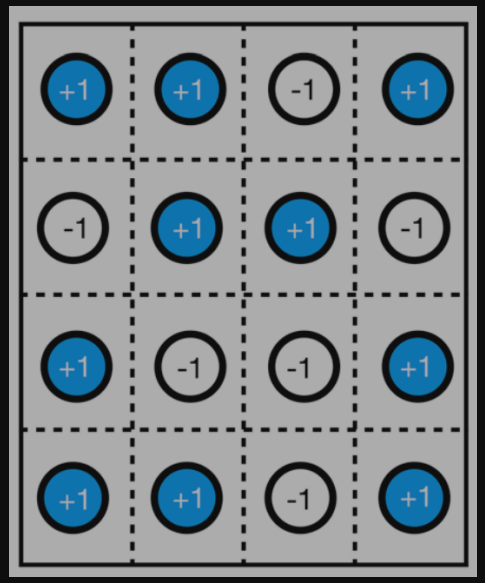
ICSD

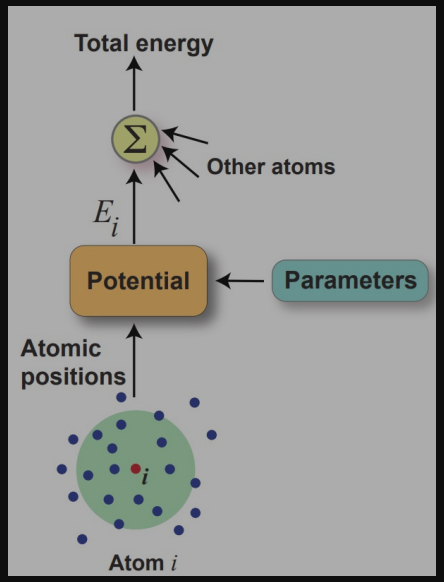
NIST National Institute of Standards and Technology

HTEM

Overview







Regression models: examples and utility

Reshma Devi

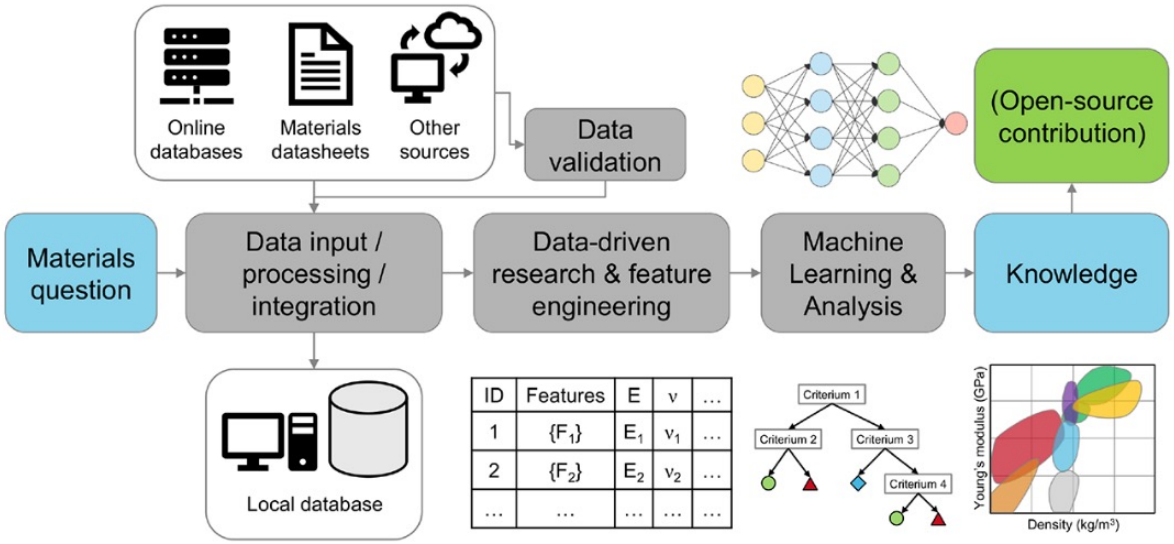
Coarse graining models: the example of cluster expansion

Dereje Bekele Tekliye

Machine learned interatomic potentials: construction and usage

Aqshat Seth

Regression models: things to note



Wang et al., Chem. Mater., 32, 4954-4965 (2020)

Important considerations

How large is your data?

How and with what ease can your model be used by the research community?

Model interpretability vs. predictive power trade off (e.g., complex neural networks vs. simple regression models)

Objectives of a ML model

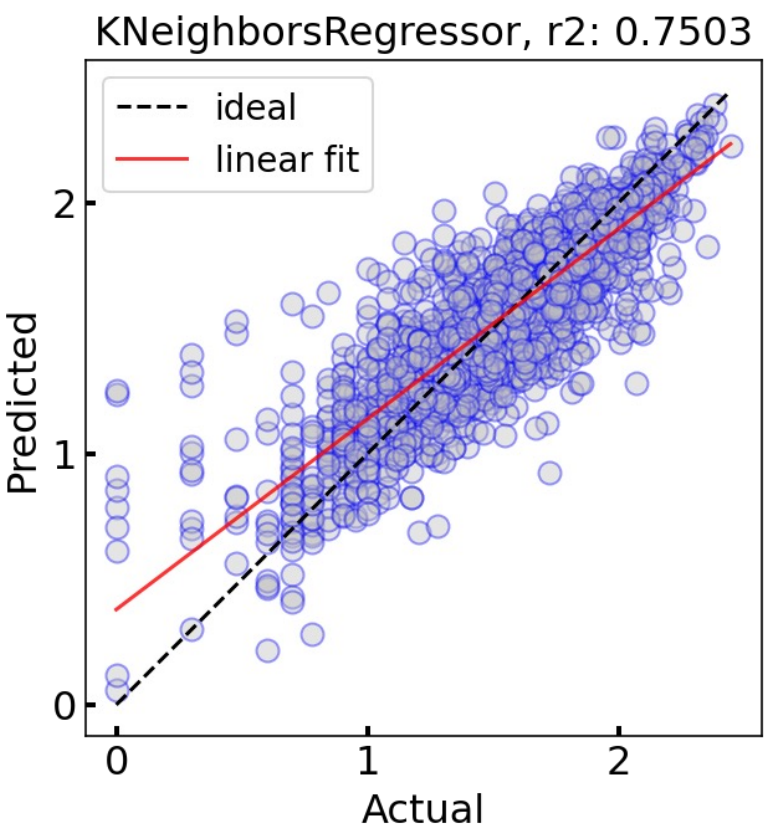
- Screen materials from a database for a given application or property
- Process data to gain new insights
- Conceptualize new modelling approaches

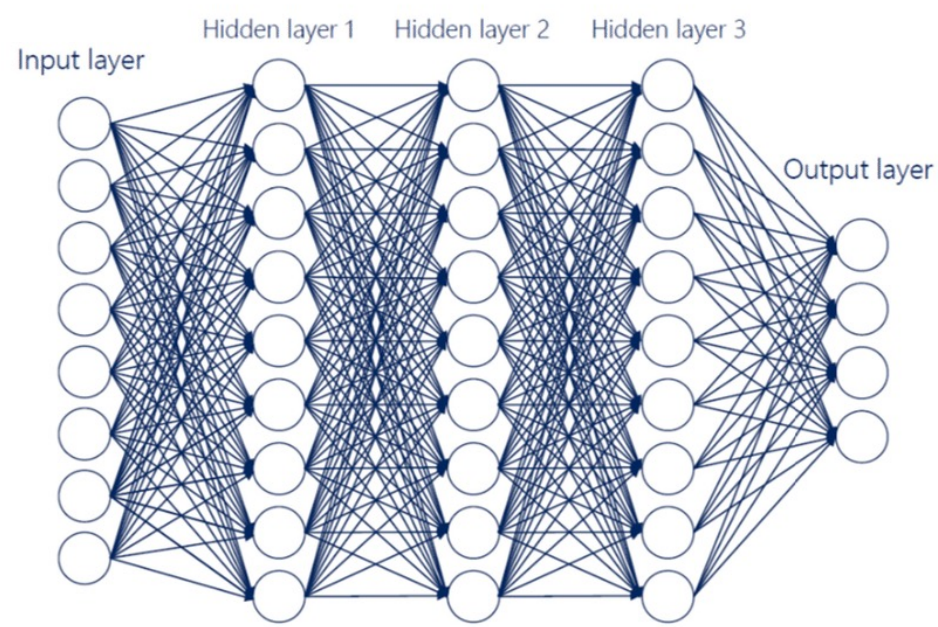
What model to choose?

Simpler models are interpretable but less accurate, typically

- "Smaller" data sets - simpler models
  - Ridge/Lasso regression
  - K-nearest neighbours
  - Random forest
  - Support vector machines

- "Larger" data sets - complex models
  - Neural networks (NNs)
  - Graph neural networks (GNN)
  - Crystal graph convolutional neural network (CGCNN)
  - Atomistic line graph neural network (ALIGNN)



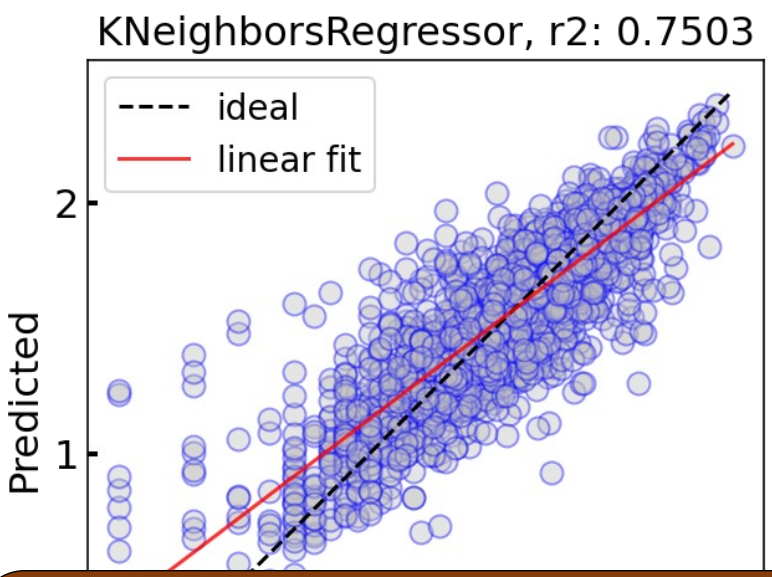


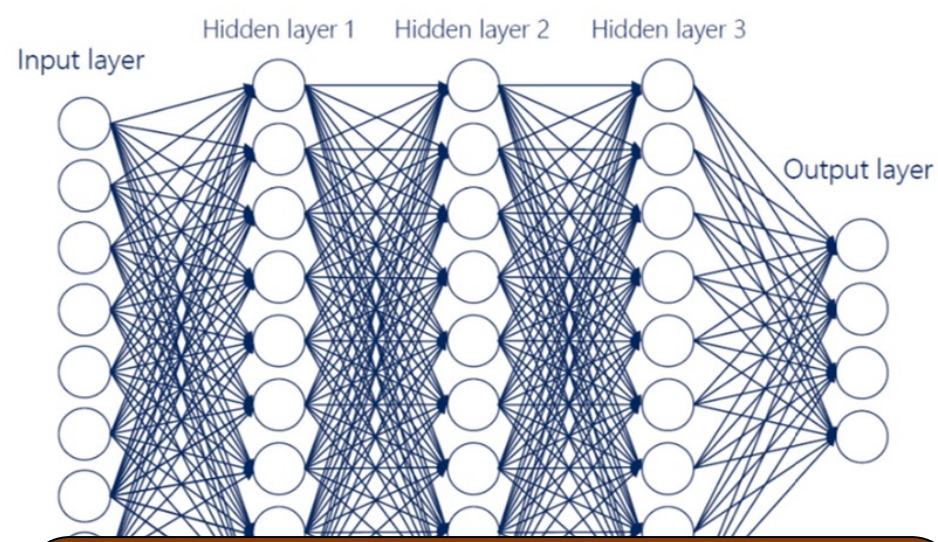
What model to choose?

Simpler models are interpretable but less accurate, typically

- "Smaller" data sets - simpler models
  - Ridge/Lasso regression
  - K-nearest neighbours
  - Random forest
  - Support vector machines

- "Larger" data sets - complex models
  - Neural networks (NNs)
  - Graph neural networks (GNN)
  - Crystal graph convolutional neural network (CGCNN)
  - Atomistic line graph neural network (ALIGNN)

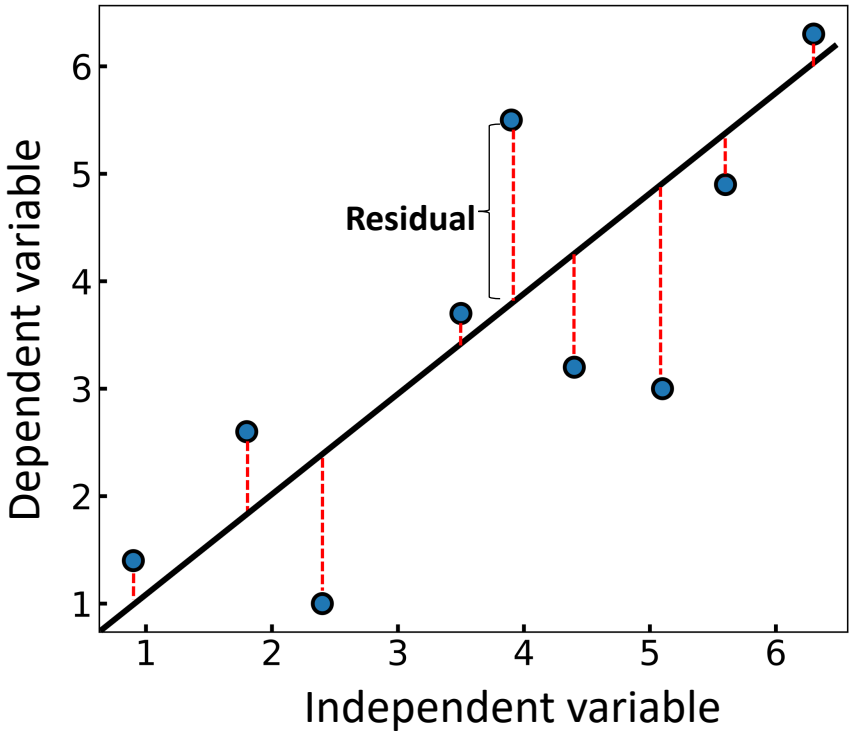




- Human interpretable
- Provides chemical and physical insights
- Low accuracy

- "Black Box"
- Does not provide chemical/physical insights
- High accuracy

How to quantify model accuracy?



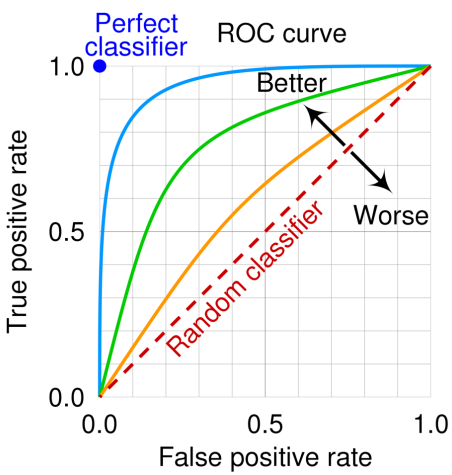
Higher accuracy → smaller squared sum of residuals (SSR)

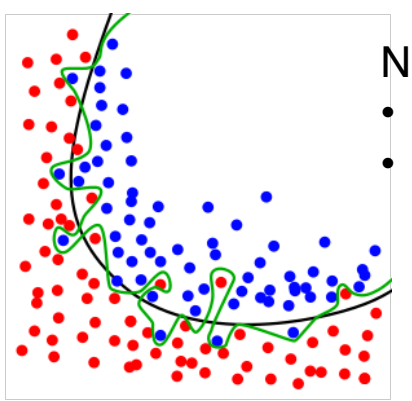
Regression models (continuous target)

- r^2
- Mean absolute error (MAE)
- Root mean square error (RMSE)

Classification models (binary target)

- Accuracy: fraction of correct predictions
- Precision: fraction of correct 'positives' among all positives
- Recall: actual fraction of correct 'positives'
- Receiver operator characteristic (ROC) curve



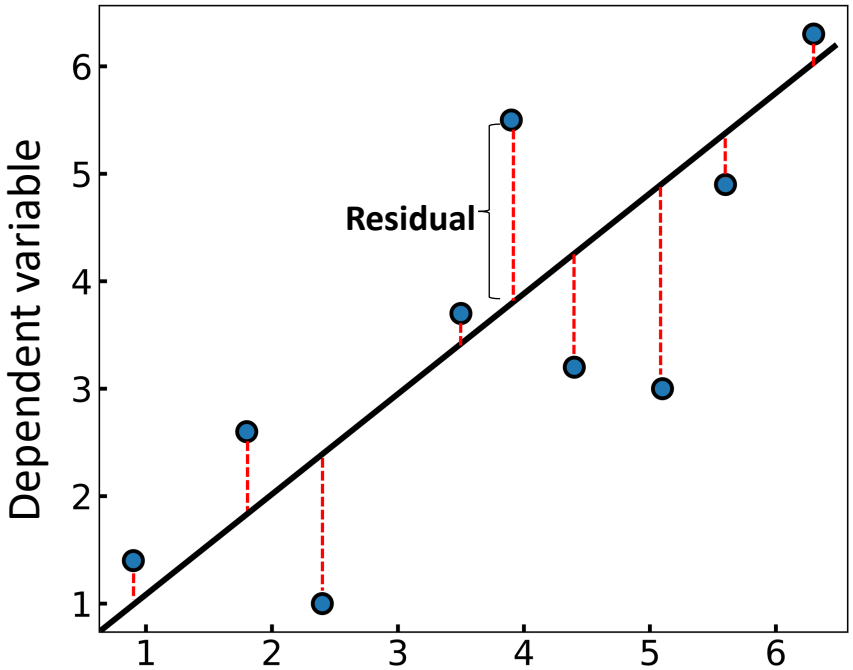


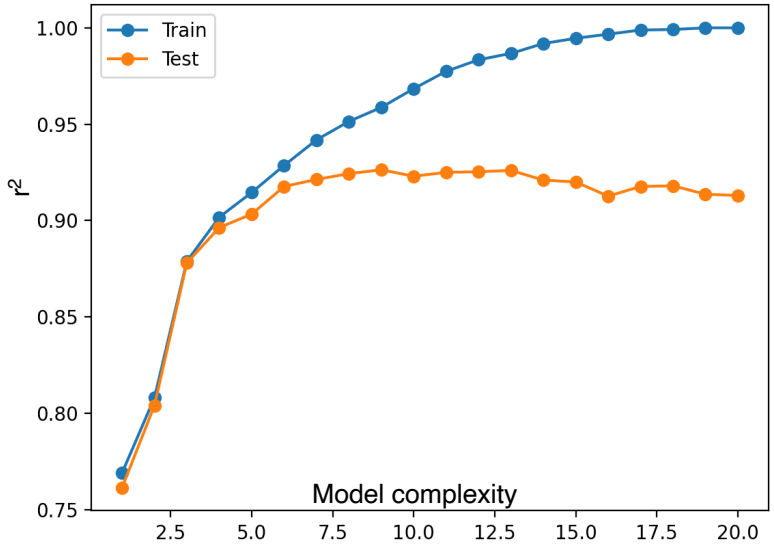
How do we know our model isn't overfit on data?

Need to test our model on 'unseen' data

- k-fold cross-validation (CV) score (simple models)
- Error on test dataset (complex models)

How to quantify model accuracy?





Higher accuracy → smaller squared sum of residuals (SSR)

Regression models (continuous target)

- r^2
- Mean absolute error (MAE)
- Root mean square error (RMSE)

Classification models (binary target)

- Accuracy: fraction of correct predictions
- Precision: fraction of correct 'positives' among all positives
- Recall: actual fraction of correct 'positives'
- Receiver operator characteristic (ROC) curve

How do we know our model isn't overfit on data?

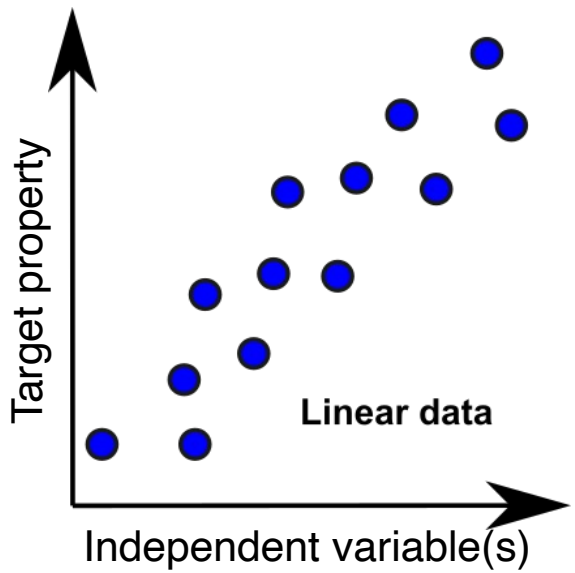
Need to test our model on 'unseen' data

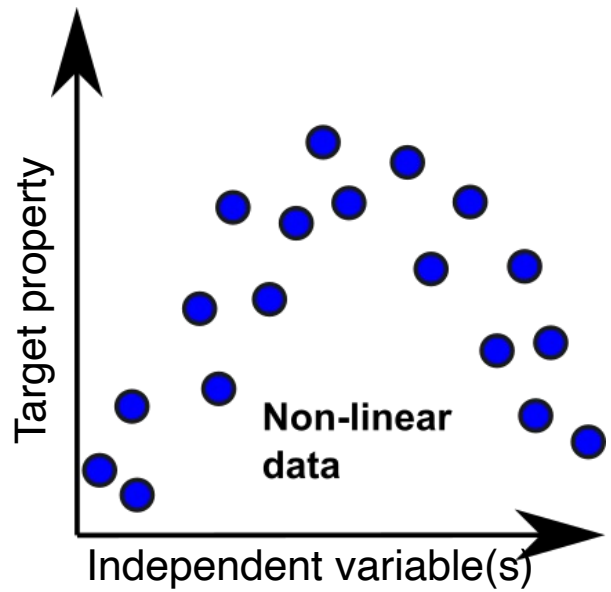
- k-fold cross-validation (CV) score (simple models)
- Error on test dataset (complex models)

Significant deviation between training and test errors→ overfit model

Linear and non-linear models

Relationship of target data can be linear/non-linear with underlying independent variables (descriptors)





Linear regression/linear model works best

$$y = b + \sum_i a_i x_i$$

Popular models:

- Linear regression (RMSE reduction)
- LASSO regression (L_1 norm)
- Ridge regression (L_2 norm)

Non-linear regression/non-linear model works best

$$y = b + \sum_i f(a_i, x_i)$$

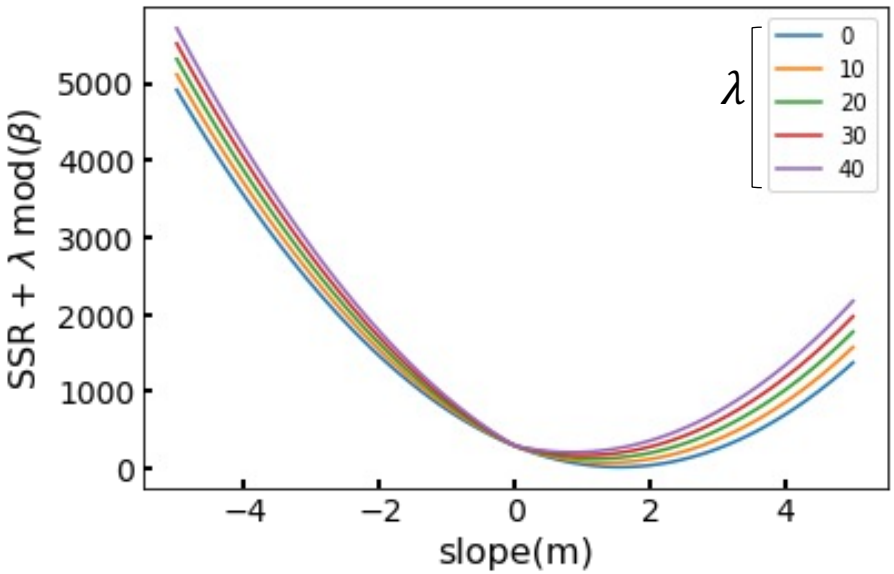
Popular models:

- Random forest
- Support vector machine (SVM)
- K-nearest neighbors (KNN)
- Neural networks

Overview of linear models

LASSO (L_1 norm)

$$L_1 = \min(SSR + \lambda \left|\left|\beta\right|\right|_1)$$

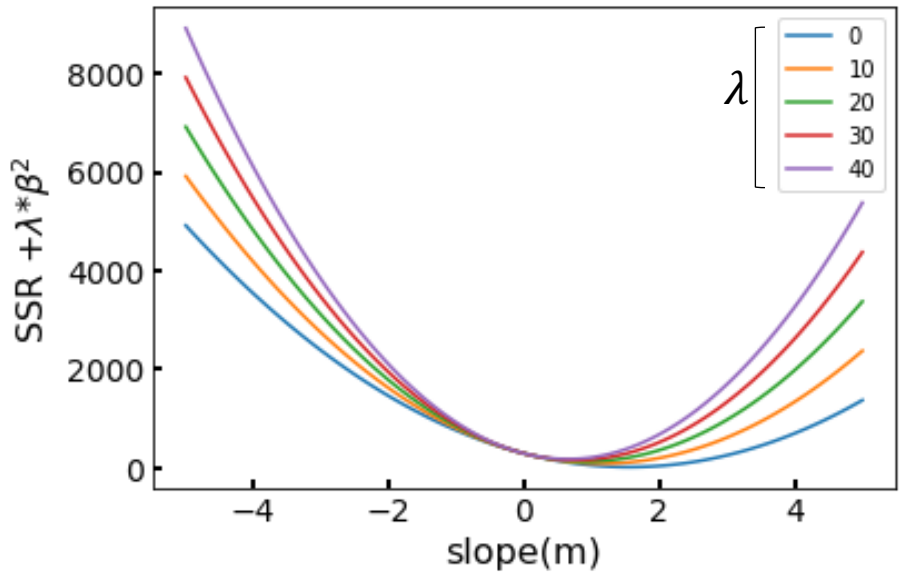


Decreases coefficients of non-important descriptors to 0

Can be difficult to get best model

Ridge (L_2 norm)

$$L_1 = \min(SSR + \lambda \left|\left|\beta\right|\right|_2^2)$$



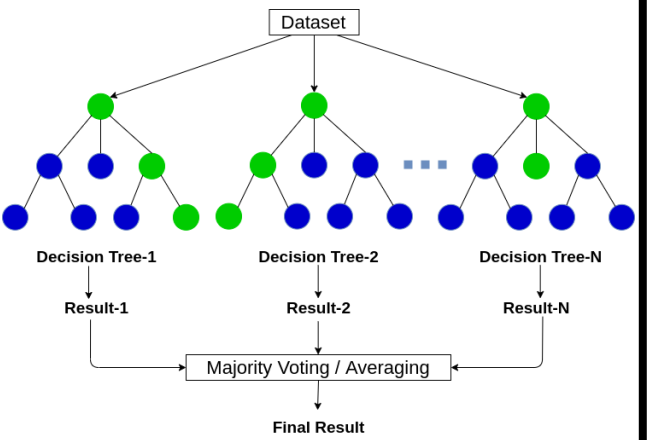
Does not necessarily decrease coefficients of non-important descriptors to 0

Usually easier to get best model compared to LASSO

Overview of non-linear (simple) models

Most non-linear models can be used both for regression and classification

Random forest



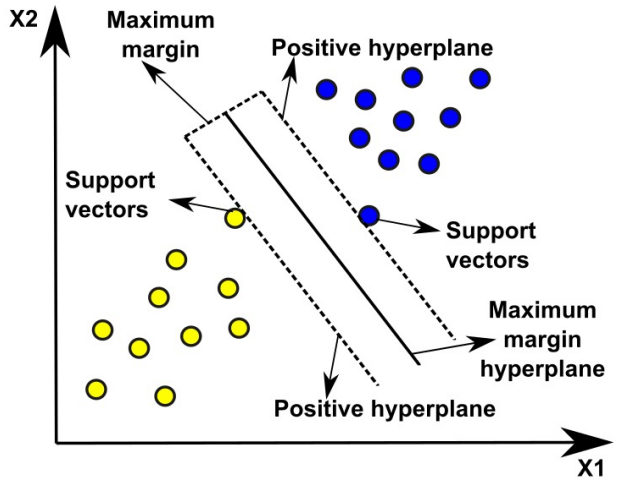
Ensemble model: final decision is an average of several trees

Each tree: if-else decisions

- Handle noisy and 'large' data
- Resistant to overfitting
- Less sensitive to training data

- May not be interpretable
- Computationally slow for 'large' datasets

Support vector machine

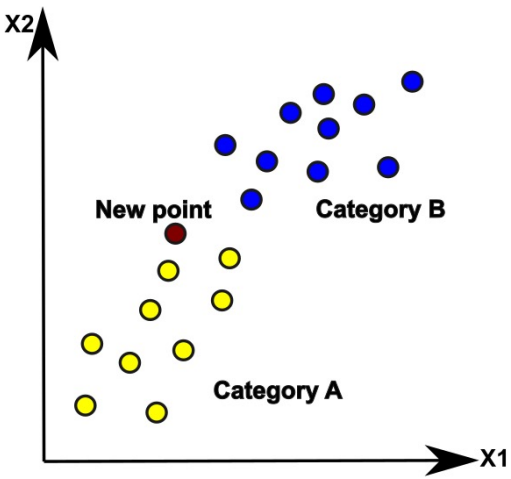


Identify hyperplanes that separate data into clusters

- Efficient at identifying key descriptors in high-dimensional space
- Memory efficient

- Sensitive to noise in data

K-nearest neighbors



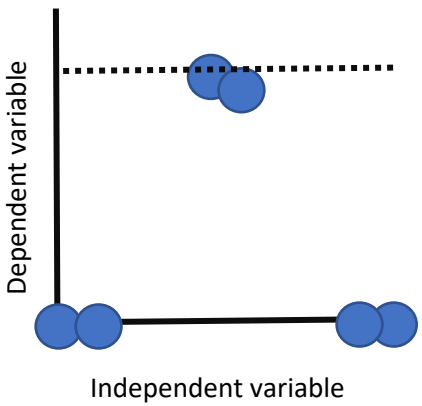
Uses feature similarity (i.e., 'distance' from other points) to make predictions about unseen data

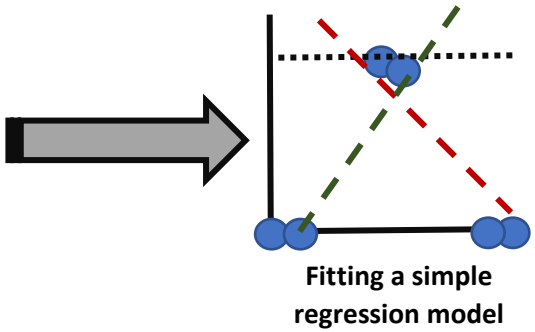
- Easy to implement
- Resistant to noisy data

- Memory inefficient (needs to store entire training data)

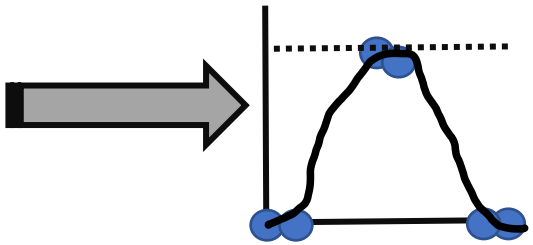
Non-linear complex model: neural network

Suppose we want to fit the following data





Fitting a simple regression model

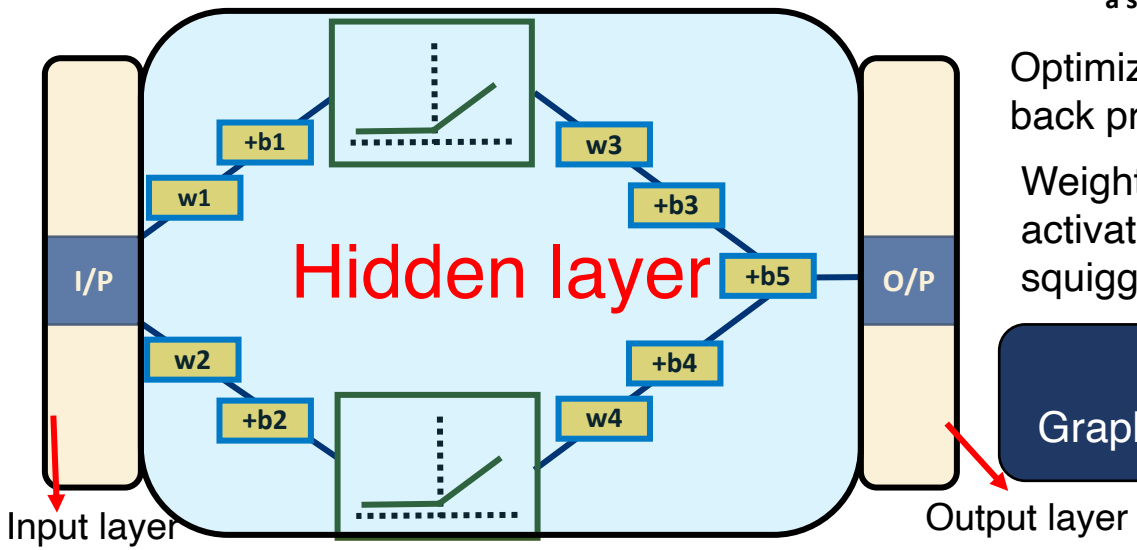


Neural networks can fit a squiggle

Pros
- Robust and accurate
- Parallel processing

Cons
- Minimal interpretability
- Tendency to overfit

Single layer neural network



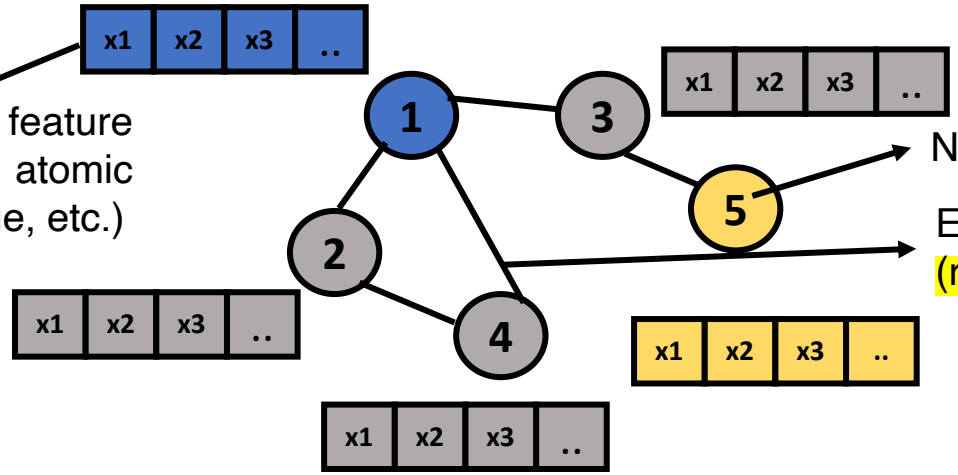
Optimized biases and weights are obtained via back propagation

Weights and biases determine the part of the activation function that will contribute to the squiggle

Several types of NNs exist
Graph NNs particularly relevant for materials

Graphs are an intuitive way to model atoms and bonds

Each node has a feature vector: collection of atomic properties (mass, charge, etc.)



Nodes are atoms (features)

Edges represent bonds (relationships)

Graphical representation of the data (structure) → Suitable convolutional scheme → Embeddings (pass information across bonds) → Feed forward to a NN (fit weights and biases) → Regression or classification

Graph neural networks can make predictions at three levels

- Graph level (overall structure)
- Edge level (for a given bond)
- Node level (for a given atom)

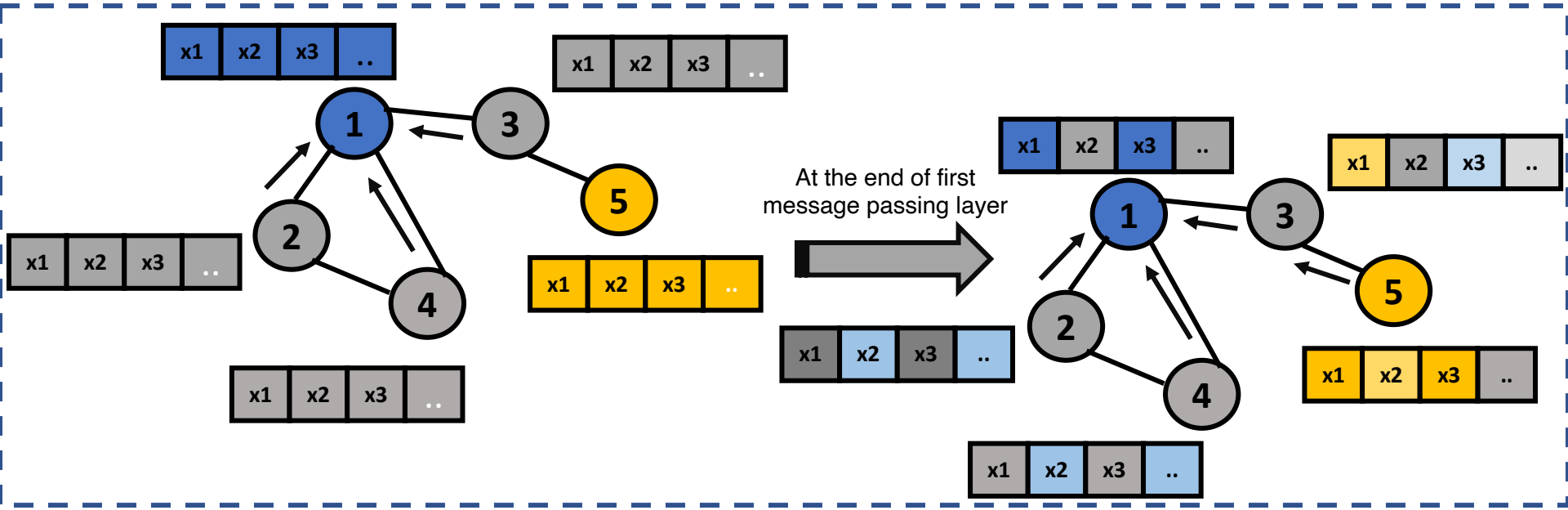
Pros

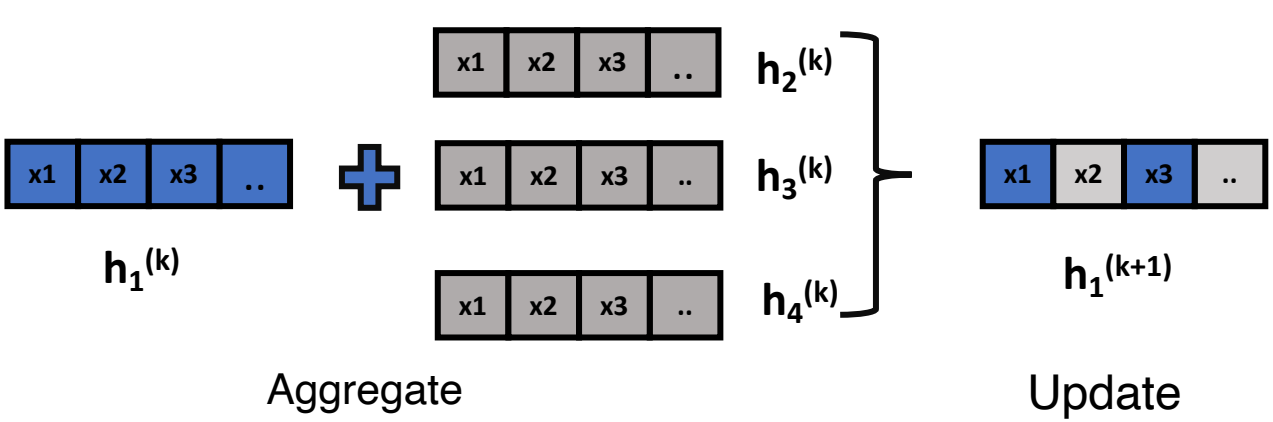
- Highly accurate
- Message passing: use information from neighbors
- Can take into account underlying symmetry

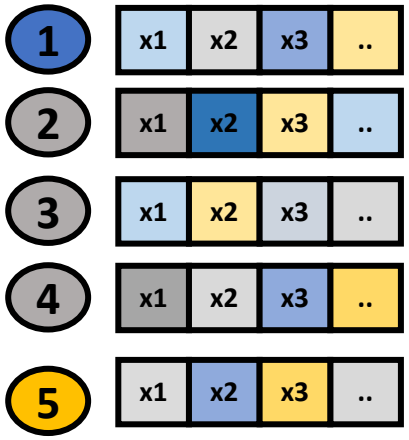
Cons

- Storage/input graph size
- Inability to distinguish multiple types of bonds
- Need to ensure permutational invariance and equivariance

Message passing: learn from neighbors







After multiple message passing layers

Examples of regressions in action

Predicting material properties:
Oxygen vacancy formation energy in ABO_3 perovskites

JACS
JOURNAL OF THE AMERICAN CHEMICAL SOCIETY
pubs.acs.org/JACS Article

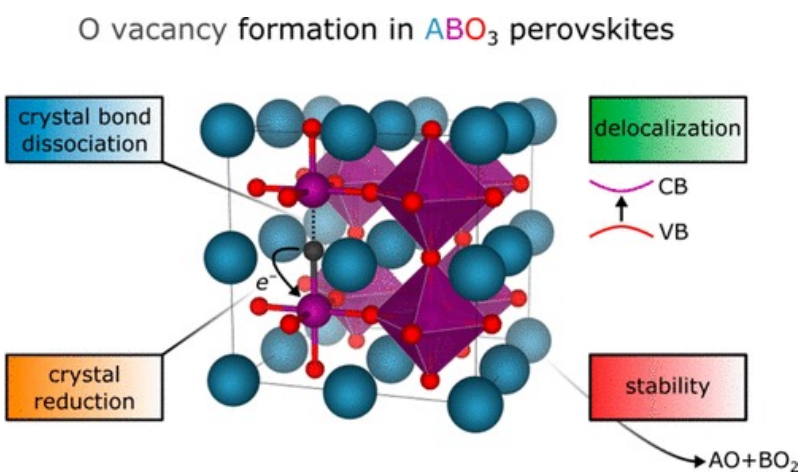
Factors Governing Oxygen Vacancy Formation in Oxide Perovskites

Robert B. Wexler, Gopalakrishnan Sai Gautam, Ellen B. Stechel, and Emily A. Carter*

Cite This: *J. Am. Chem. Soc.* 2021, 143, 13212–13227 Read Online

- ABO_3 perovskites
  - A= Ca, Sr, Ba, La, or Ce
  - B= Ti, V, Cr, Mn, Fe, Co, or Ni
- **Database**: 341 Datapoints obtained from density functional theory (DFT) calculations

- **Model**: A simple linear model with physically intuitive descriptors
  - Crystal bond dissociation energy
  - Crystal reduction potential
  - Band gaps
  - Energy above hull
- **Performance**:
  - Mean absolute error (MAE) - 0.45 eV
  - $BiFeO_3$ and $BiCoO_3$ identified as viable candidates for solar thermochemical water splitting



Predicting material properties:
Elastic moduli of inorganic compounds

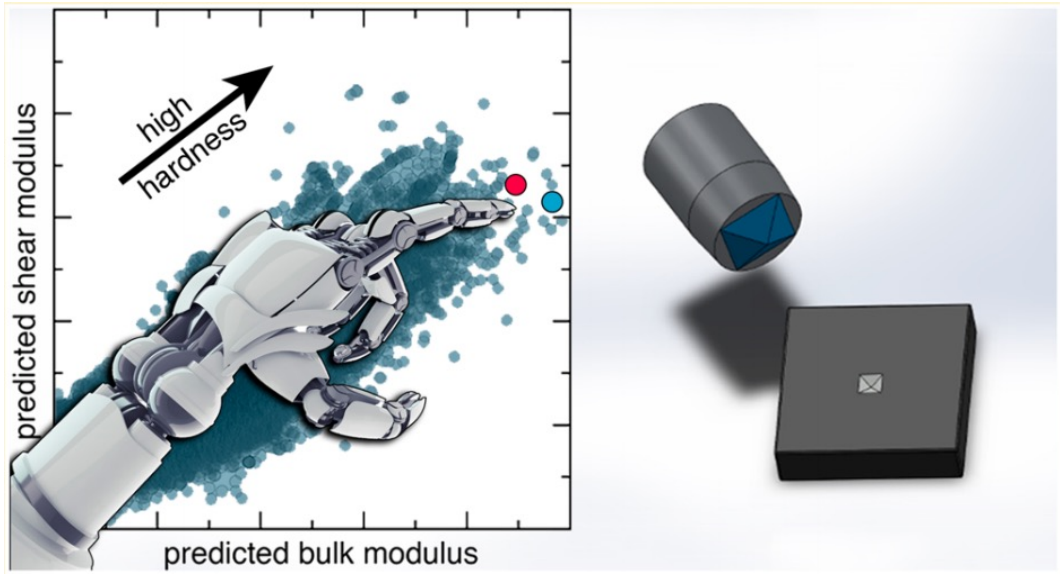
JACS
JOURNAL OF THE AMERICAN CHEMICAL SOCIETY
Article
Cite This: J. Am. Chem. Soc. 2018, 140, 9844–9853
pubs.acs.org/JACS

Machine Learning Directed Search for Ultraincompressible, Superhard Materials

Aria Mansouri Tehrani,[†,⊥] Anton O. Oliynyk,[†,⊥] Marcus Parry,[‡] Zeshan Rizvi,[†] Samantha Couper,[§] Feng Lin,[§] Lowell Miyagi,[§] Taylor D. Sparks,[‡] and Jakoah Brgoch*[,†]

Database: 3248 Bulk (*B*) and shear modulus (*G*) data obtained from the Materials Project (MP) database

- **Model**: Support vector machine regression using 150 composition and structural descriptors
- **Performance**:
  - r^2 score = 0.94
  - Identified incompressible – high hardness metal $ReWC_{0.8}$ and $Mo_{0.9}W_{1.1}BC$ with *B* = 380 and 370 GPa, respectively
  - Experimentally verified

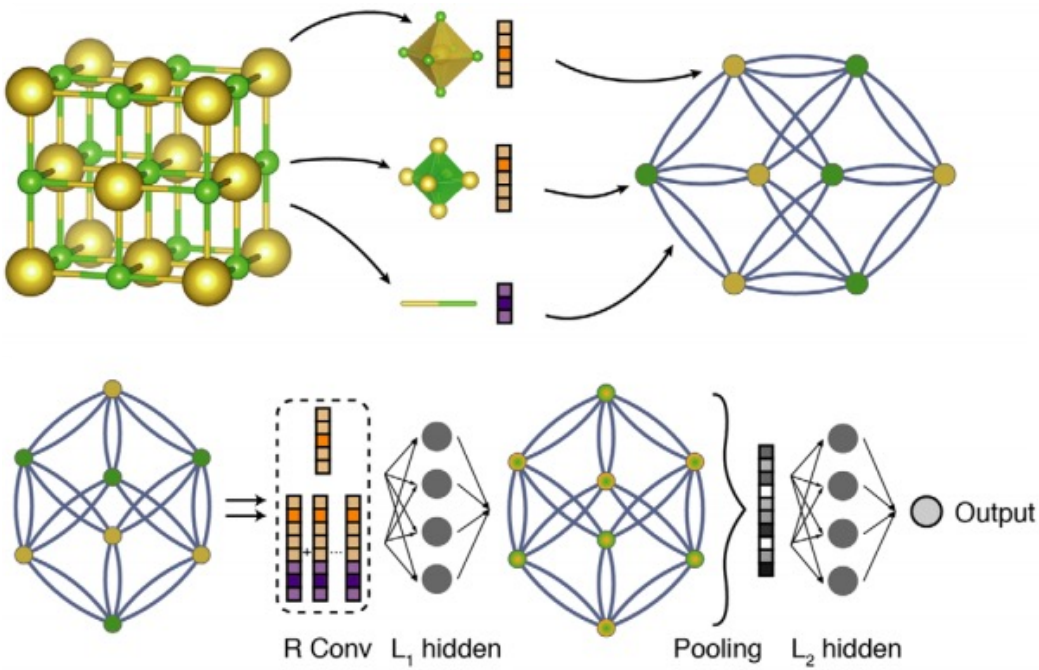


Predicting material properties:
Diverse material properties with graph neural network

PHYSICAL REVIEW LETTERS **120,** 145301 (2018)

Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties

Tian Xie and Jeffrey C. Grossman
Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA



Properties : Formation energy, band gap, Fermi energy, bulk and shear moduli, and Poisson's ratio

Database: 10^4 DFT-calculated datapoints from MP

Model: Crystal Graph convolutional neural network (CGCNN)

Performance:

- Formation energy: 0.039 eV/atom
- Band gap: 0.388 eV
- Fermi energy: 0.363 eV
- Elastic moduli: ~1-2 GPa
- Poisson's ratio: 0.03
- Identified 228 'synthesizable' perovskites out of 18928 in the training database

Predicting material properties:
Mechanical properties for energy storage

ACS central science

ACS AuthorChoice

Research Article

Cite This: ACS Cent. Sci. 2018, 4, 996–1006

http://pubs.acs.org/journal/acscii

Machine Learning Enabled Computational Screening of Inorganic Solid Electrolytes for Suppression of Dendrite Formation in Lithium Metal Anodes

Zeeshan Ahmad,[†] Tian Xie,[‡] Chinmay Maheshwari,[†] Jeffrey C. Grossman,[‡] and Venkatasubramanian Viswanathan*[,‡,§]

Mechanically anisotropic interfaces suppress dendrite growth

- Dependent on *G*, *B,* and elastic constants.

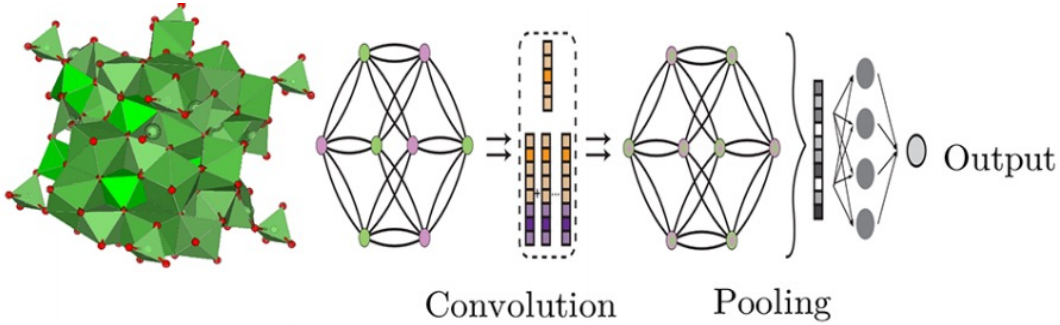
Database: Subset of MP containing 12,000 compounds with Li

Model:

- Graph neural network for *G* and *B* prediction
- Gradient boost and Kernel-ridge regression for elastic constant predictions

Performance:

- RMSE in log(GPa): 0.1268 (*G*) and 0.1013 (*B*)
- 20 interfaces with six solid electrolytes predicted to be stable against dendrite initiation



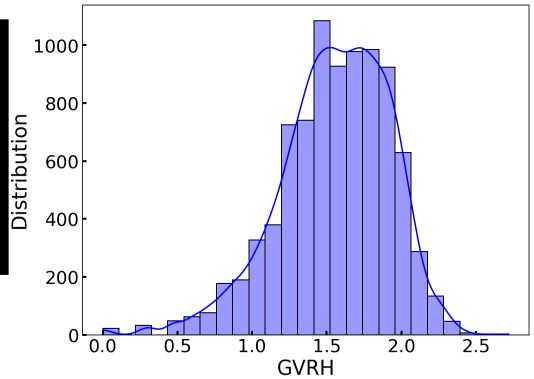
Material Database → Convolution/ Regression → Stability criteria → Dendrite suppressing candidates

Hands—on session?

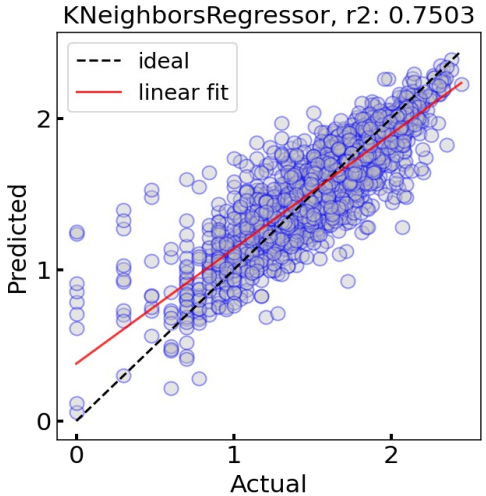
Perform ‘simple’ regressions

Data: Shear modulus, band gap, and formation energy from matbench database

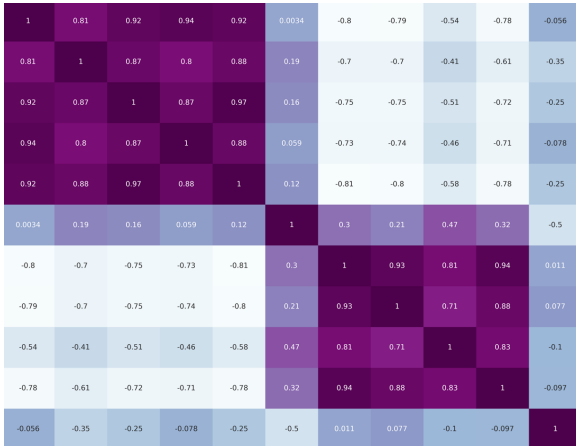
1 Extract and Clean-up the downloaded datasets



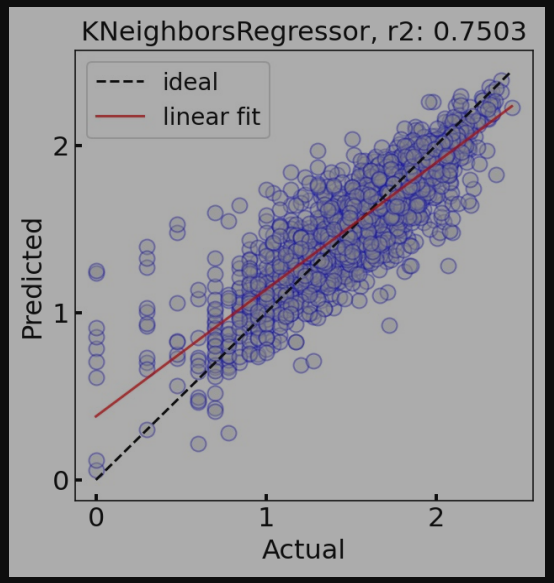
2 Train classical ML models and optimize the hyperparameters



3 Observe the correlation among the features

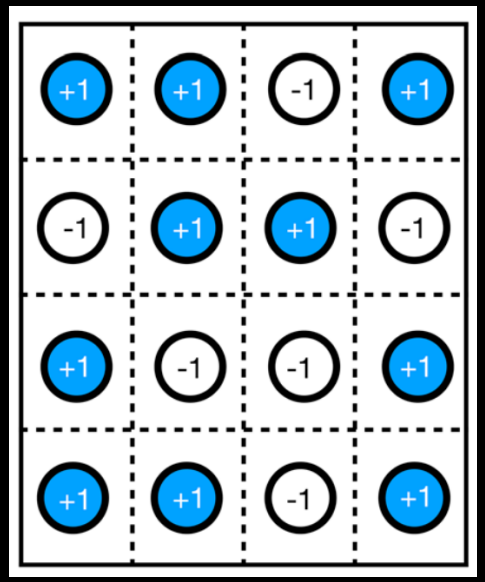


Overview



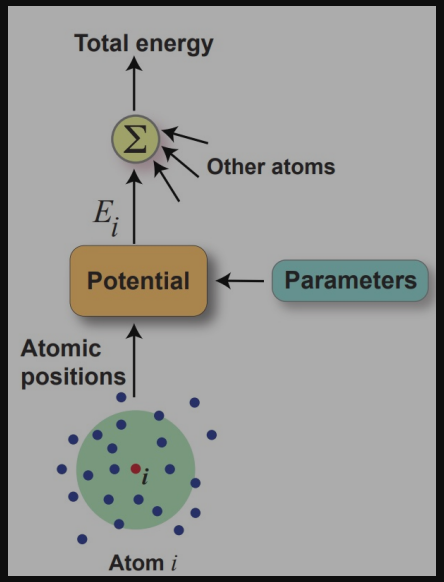
Regression models: examples and utility

Reshma Devi



Coarse graining models: the example of cluster expansion

Dereje Bekele Tekliye



Machine learned interatomic potentials: construction and usage

Aqshat Seth

Why lattice models?

- Quantum mechanics (e.g., DFT) provides accurate predictions at 0 K
  - High temperature properties?

- DFT calculations become prohibitively expensive beyond ~1000 atoms
  - Simple binary system has 2^N possible configurations (N = number of sites)
    - 16 sites → 65,536 configurations!
  - DFT is not practical for estimating configurational entropy through sampling

- Predicting phase transitions using molecular dynamics is difficult
  - Requires 'long' timescales and 'large' supercells
  - Using principles of statistical mechanics may be better

- Lattice models approximate (or abstract) the energetic interactions within a given structure to 'smaller' entities
  - Helps capture entropic contributions → high temperature properties
  - Predicts order-disorder transition temperatures
  - Calculate phase diagrams

Why lattice models?

- Quantum mechanics (e.g., DFT) provides accurate predictions at 0 K
  - High temperature properties?

- DFT calculations become prohibitively expensive beyond ~1000 atoms
  - Simple binary system has 2^N possible configurations (N = number of sites)
    - 16 sites → 65,536 configurations!
  - DFT is not practical for estimating configurational entropy through sampling

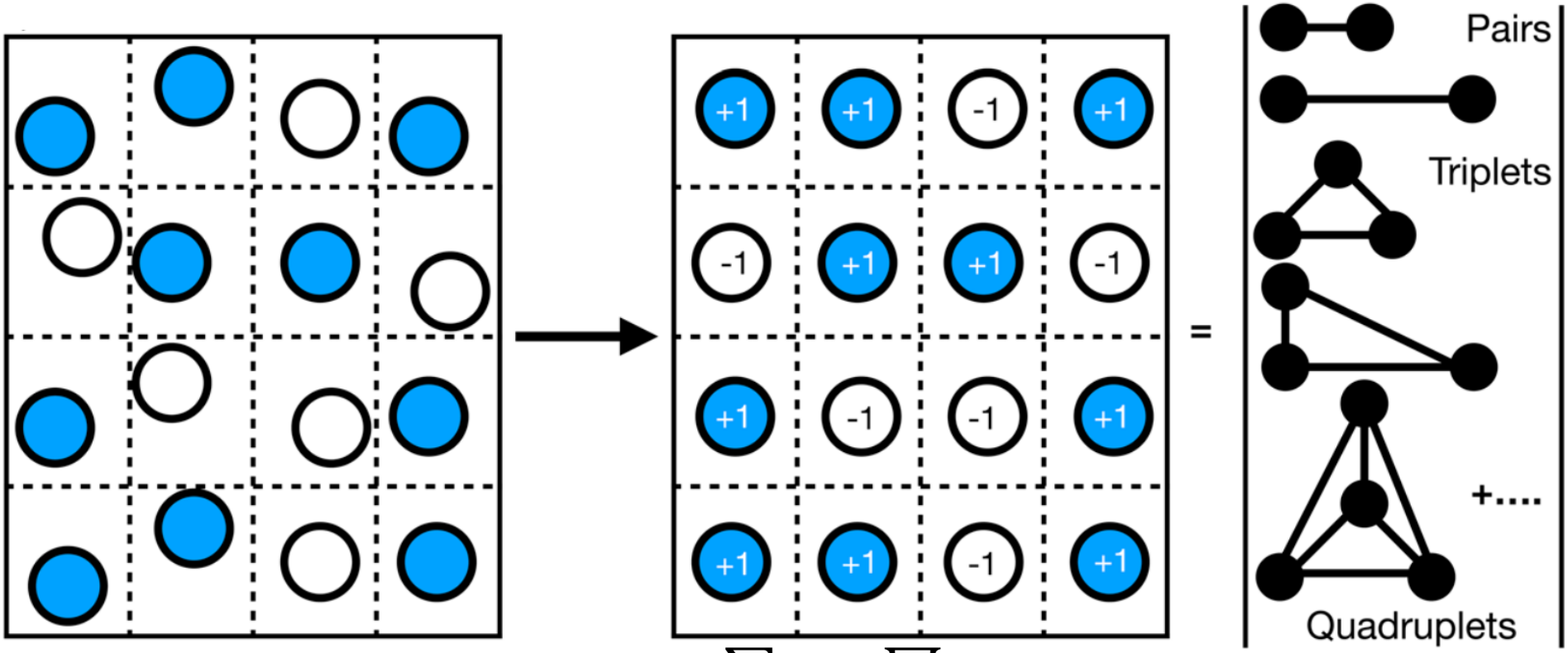
Why bother about lattice models when considering ML?
- Lattice models: simple ML models
- Provide physical intuition
- Do NOT require large datasets!

- Lattice models approximate (or abstract) the energetic interactions within a given structure to 'smaller' entities
  - Helps capture entropic contributions → high temperature properties
  - Predicts order-disorder transition temperatures
  - Calculate phase diagrams

What is a cluster expansion?

Lattice model, specifically a generalized Ising model, to abstract energies of a given structure based on the underlying atomic configuration

- Energy decomposed to clusters, each cluster expanded on a cluster basis (orthonormal)
- Coarse-grains any ‘small’ atomic displacements from ‘ideal’ sites
- Each lattice site obtains an integer value based on atom occupying it (e.g., -1 and +1)

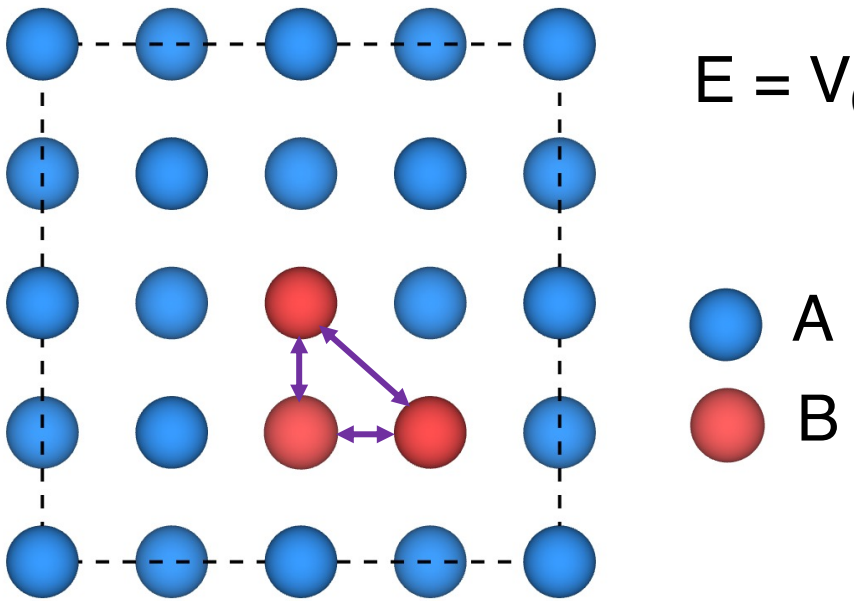


$$E(\vec{\sigma}) = \sum_{\alpha} m_{\alpha} V_{\alpha} \prod_{i \epsilon \beta} \sigma_i$$

G. Sai Gautam, and P. Canepa; Magnesium Batteries, Chap. 4, (2019)

Simplistic exercise for binary alloy A-B

Defining energy as a function of configuration



$$E = V_0 + 3V_1 + 2V_2 + V'_2 + V_3 \ \ldots$$

$A_{1-x}B_x$

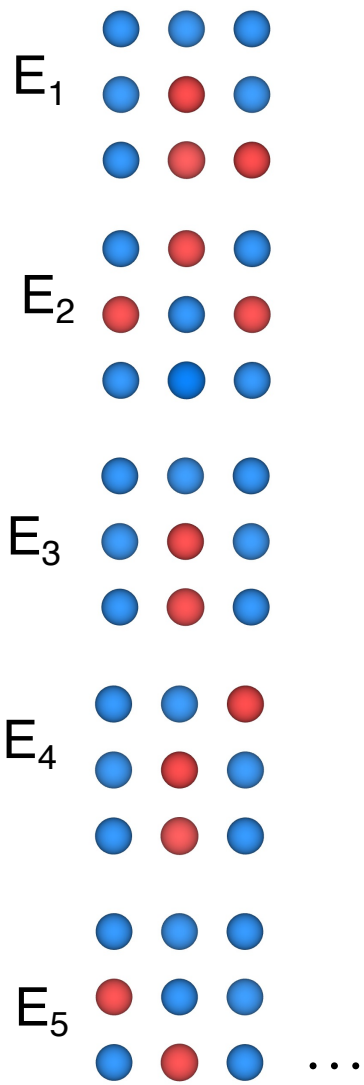
But can be done in a more systematic way: cluster expansion formalism

$$E(\vec{\sigma}) = V_o + \sum_i V_i \sigma_i + \sum_{i,j} V_{i,j} \sigma_i \sigma_j + \sum_{i,j,k} V_{i,j,k} \sigma_i \sigma_j \sigma_k + \sum_{i,j,k,l} V_{i,j,k,l} \sigma_i \sigma_j \sigma_k \sigma_l$$

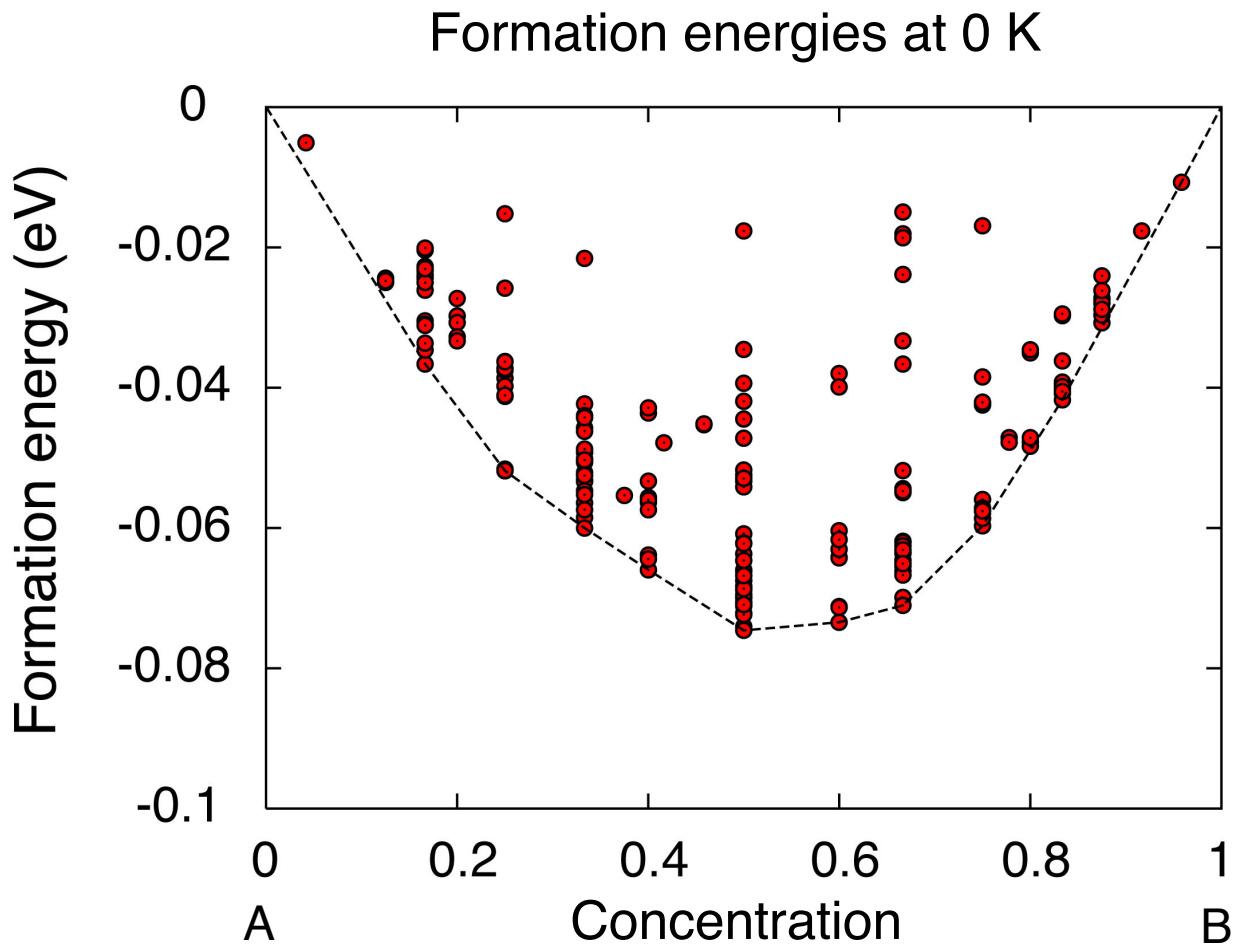
Adapted from https://www.youtube.com/watch?v=VjBUCZMZ1qI

Inputs for building a cluster expansion

DFT training data



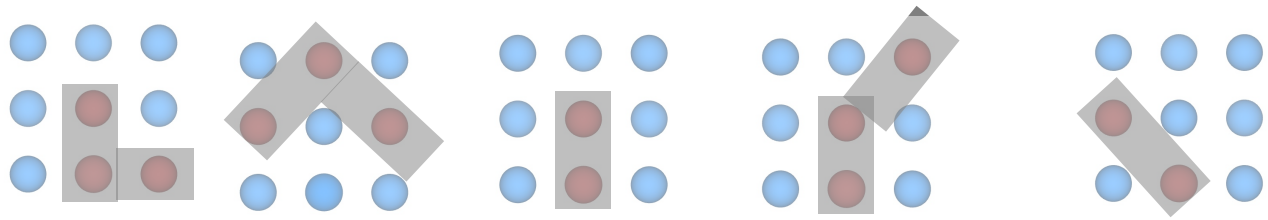
The possible configuration of a crystal is obtained by enumerating over symmetrically distinct configuration(s) across the composition(s) of interest



Adapted from https://www.youtube.com/watch?v=VjBUCZMZ1qI

Building a cluster expansion

Sets of clusters (within a given 'radius')



Calculated DFT data (Target)

Correlation matrix (Set of features)

Effective cluster interactions (Weights)

$$\begin{pmatrix} e(\vec{\sigma}_1) \\ \vdots \\ e(\vec{\sigma}_L) \\ \vdots \\ e(\vec{\sigma}_M) \end{pmatrix} = \begin{pmatrix} 1 & \Gamma_\alpha(\vec{\sigma}_1) & \Gamma_\beta(\vec{\sigma}_1) & \Gamma_\gamma(\vec{\sigma}_1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_L) & \Gamma_\beta(\vec{\sigma}_L) & \Gamma_\gamma(\vec{\sigma}_L) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Gamma_\alpha(\vec{\sigma}_M) & \Gamma_\beta(\vec{\sigma}_M) & \Gamma_\gamma(\vec{\sigma}_M) & \cdots \end{pmatrix} \begin{pmatrix} v_o \\ v_\alpha \\ v_\beta \\ v_\gamma \\ \vdots \end{pmatrix}$$

Cluster expansions are usually an under-determined system: fewer energies than ECIs available

- Both linear and non-linear optimization/regression techniques can work
  - Popular: LASSO and Genetic Algorithm
- Accuracy of fit: RMSE
- Transferability of fit: CV (Leave one-out or k-fold)

Adapted from https://github.com/prisms-center/CASMcode_demo/

Cluster expansion+Statistical mechanics

First-Principles Calculation: DFT

$$H = \sum_{i=1}^{N_e}\left(-\nabla_i^2 + V_{nuc}(r_i)\right) + \frac{1}{2}\sum_{i\neq j}\frac{1}{|r_i - r_j|} + E_{nuc}(\{R\})$$

Energies of a "few" configurations (ground + excited states)

Cluster expansion Hamiltonian

$$E(\vec{\sigma}) = \sum_{\alpha} m_\alpha V_\alpha \prod_{i\epsilon\beta}\sigma_i$$

V_α: effective cluster interactions (ECIs) fitted to DFT energies

Statistical Mechanics Approach

$$\mathrm{F} = -k_B T \ \ln Z\,, \qquad \mathrm{Z} = \sum_{\vec{\sigma}} exp\left(-\frac{E(\vec{\sigma})}{k_B T}\right)$$

Monte Carlo Simulation

Sample configurations over larger length scales to get statistical averages

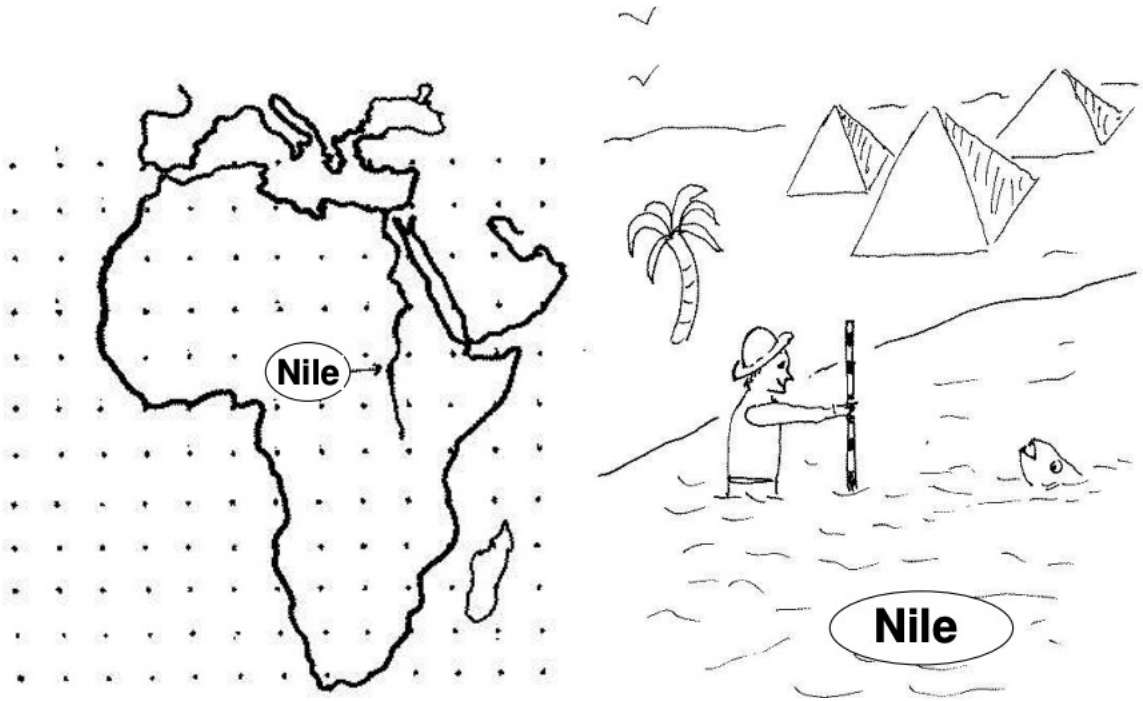
Thermodynamic quantities → **Phase diagram, diffusivity …**

Adapted from https://github.com/prisms-center/CASMcode_demo/

Monte Carlo: Metropolis or kinetic

Monte-Carlo is a general, random sampling algorithm→ can be modified to do importance sampling

Low energy configurations → important samples in materials



1. Select a particle at random, and calculate its energy $\mathcal{U}(\mathbf{r}^N)$.
2. Give the particle a random displacement; $r' = r + \Delta$, and calculate its new energy $\mathcal{U}(\mathbf{r'}^N)$.
3. Accept the move from $\mathbf{r}^N$ to $\mathbf{r'}^N$ with probability

$$\mathrm{acc}(o \to n) = \min\left(1, \exp\{-\beta[\mathcal{U}(\mathbf{r'}^N) - \mathcal{U}(\mathbf{r}^N)]\}\right).$$

One implementation of Metropolis, satisfying 'detailed balance'

Provides statistical averages of equilibrium quantities → phase diagrams, transitions

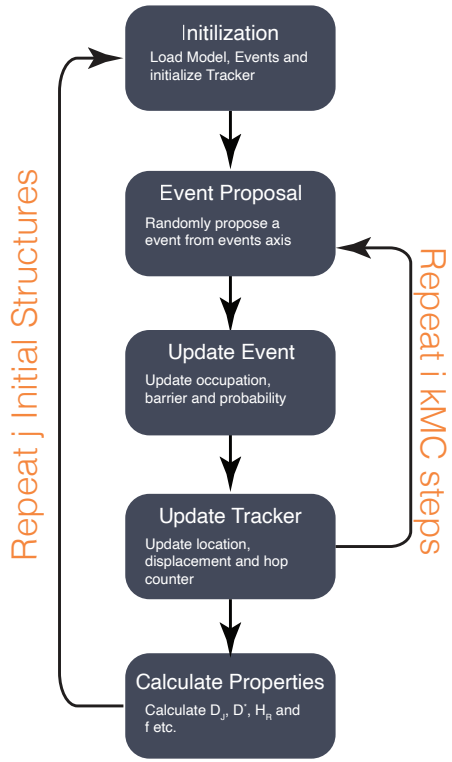
"Introduction to Monte Carlo methods" by Daan Frenkel

Monte Carlo: Metropolis or kinetic

Monte-Carlo is a general, random sampling algorithm→ can be modified to do importance sampling

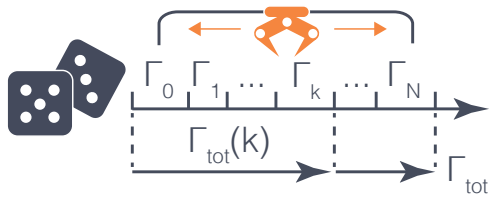
Low energy configurations → important samples in materials

Rejection-free Kinetic Monte Carlo



Kinetic Monte Carlo: dynamic properties (e.g., diffusivities)

Event Proposal



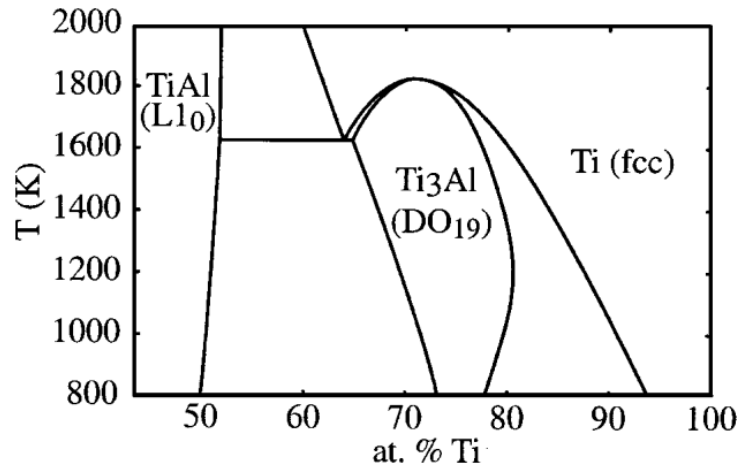
Deng et al., Comput. Mater. Sci., Accepted (2023)

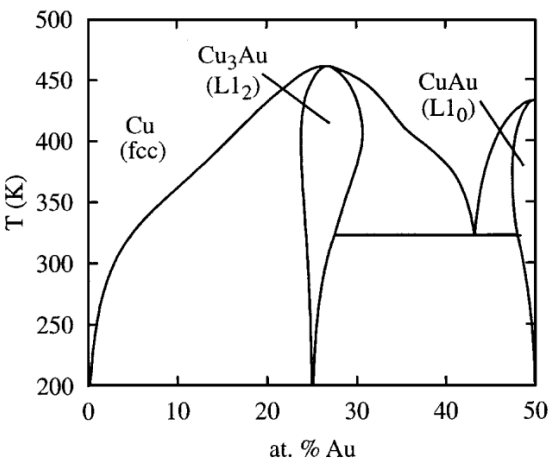
Examples of cluster expansions in action

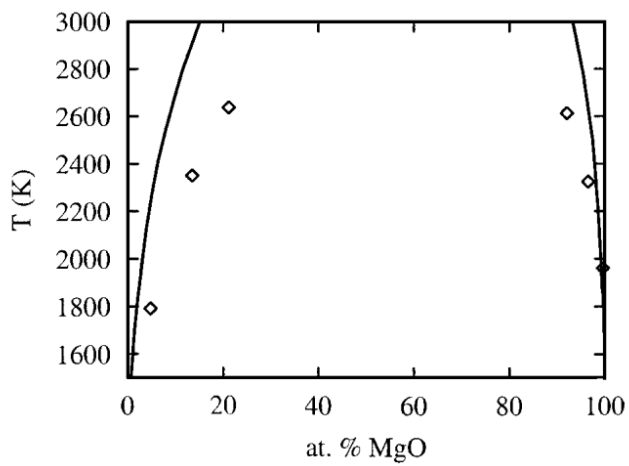
Examples of cluster expansions

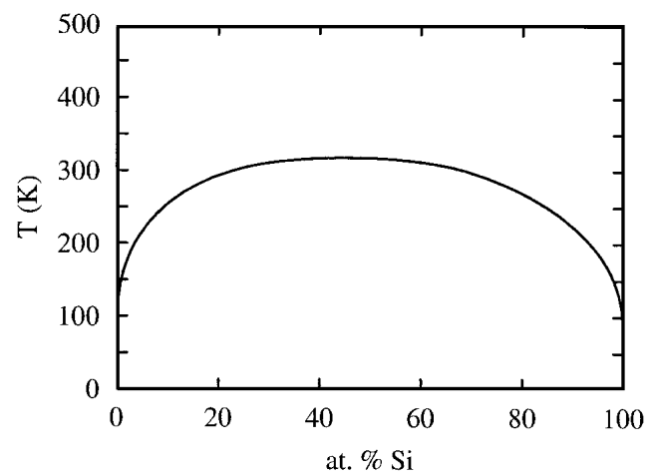
Characteristic	Si-Ge	CaO-MgO	Ti-Al hcp	Ti-Al fcc	CuAu
Number of structures	27	20	55	23	33
Number of clusters	2 + 8 + 3	2 + 3 + 7 + 1	2 + 11 + 6	2 + 3 + 2	2 + 6
CV score, meV/atom	1	18	35	49	23

The number of clusters is given as the number of each type of multiplet: empty and point clusters + pairs + triplets + quadruplets





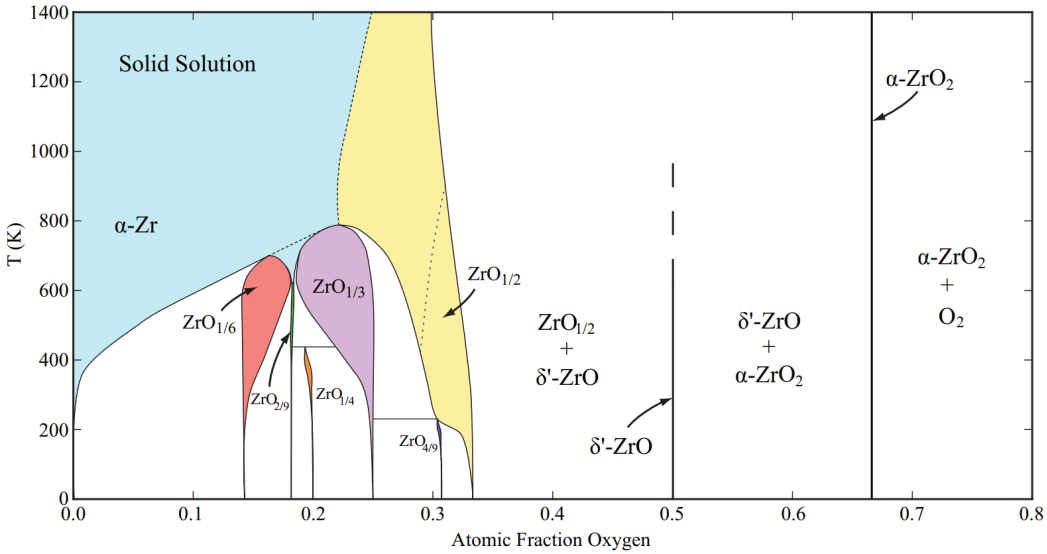




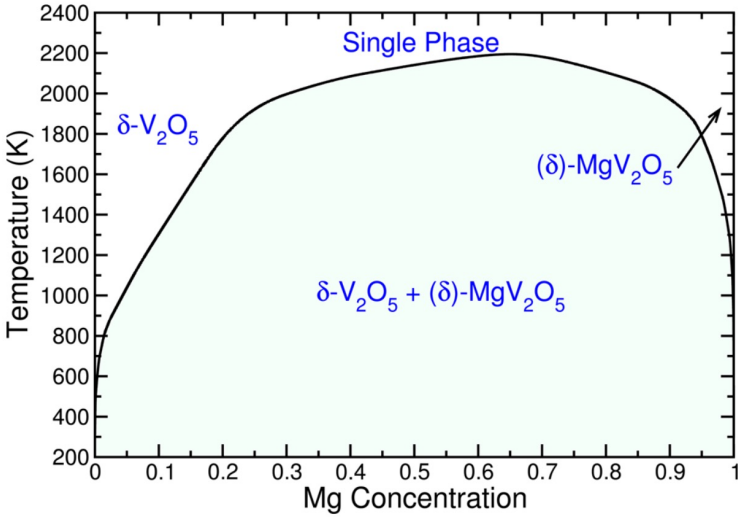
A. van de Walle and G. Ceder, J Phase Equilib. 23, 348-359 (2002)

Examples of cluster expansions

Phase diagram construction



B. Puchala and A. Van der Ven, Phys. Rev. B, 094108 (2013)

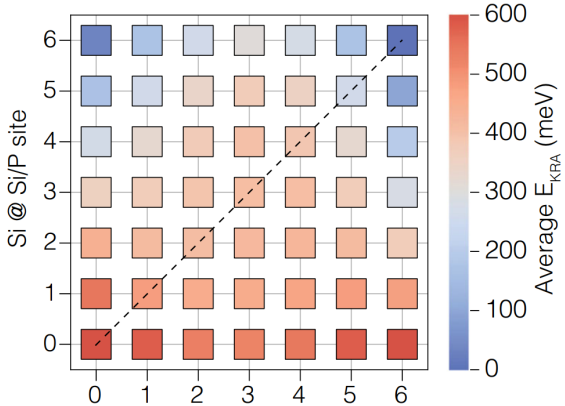


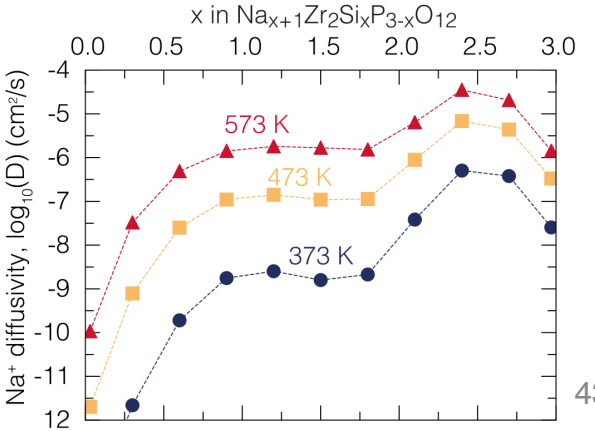
G. Sai Gautam et al., Chem. Mater. 27, 3733-3742 (2015)

Diffusivity calculations

Local cluster expansion coupled with kinetic Monte Carlo simulation

Z. Deng and G. Sai Gautam et al., Nat. Commun. 13, 4470 (2022)





Hands—on session?

Build a ‘simple’ cluster expansion

smol

Statistical Mechanics on Lattices

CASM

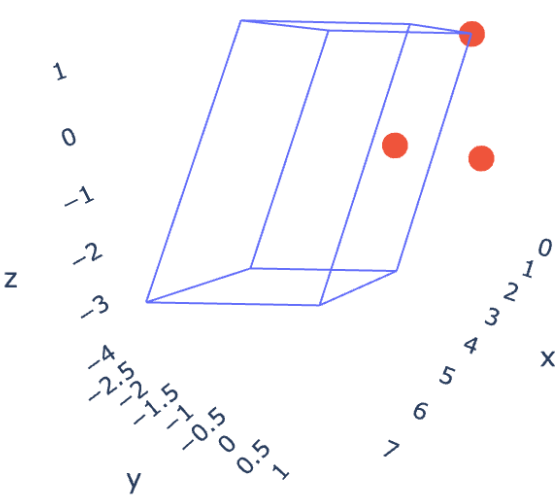
A Clusters Approach to Statistical Mechanics

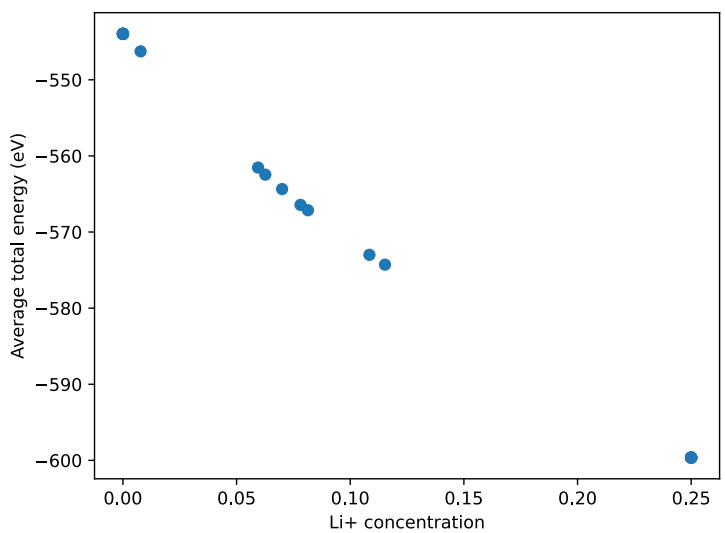
The Alloy-Theoretic Automated Toolkit (ATAT): A User Guide

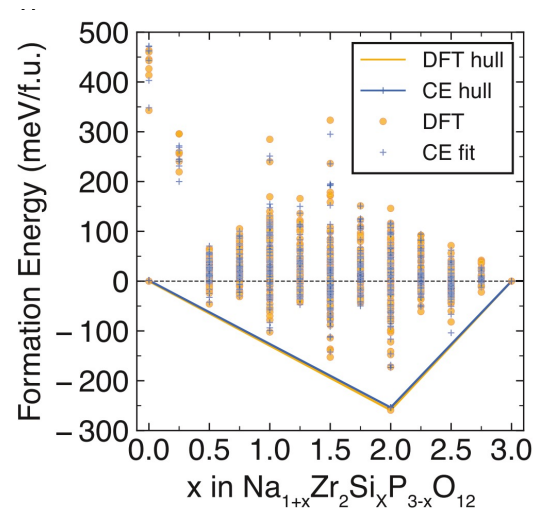
Axel van de Walle

icet

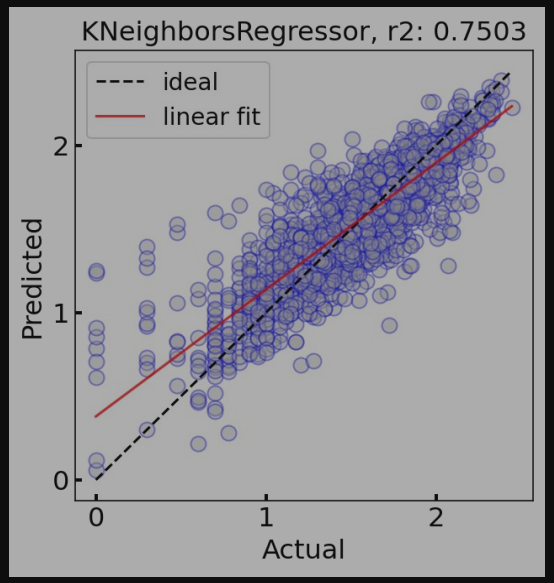
And run a sample Monte-Carlo!

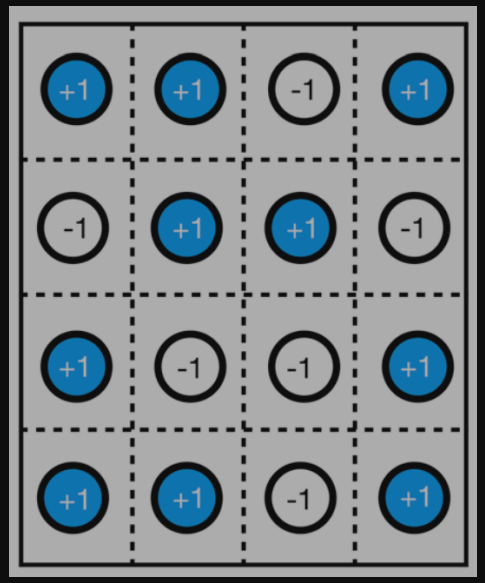


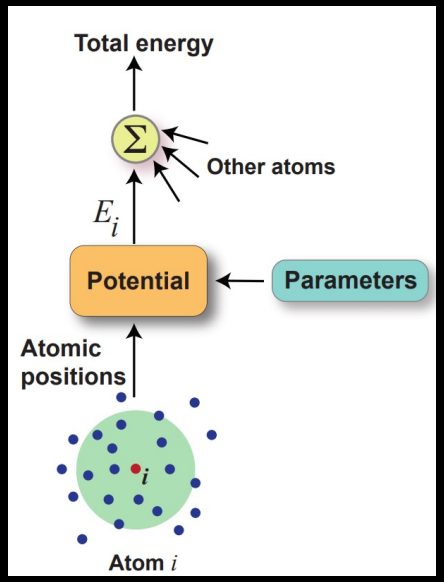




Overview







Regression models: examples and utility

Reshma Devi

Coarse graining models: the example of cluster expansion

Dereje Bekele Tekliye

Machine learned interatomic potentials: construction and usage

Aqshat Seth

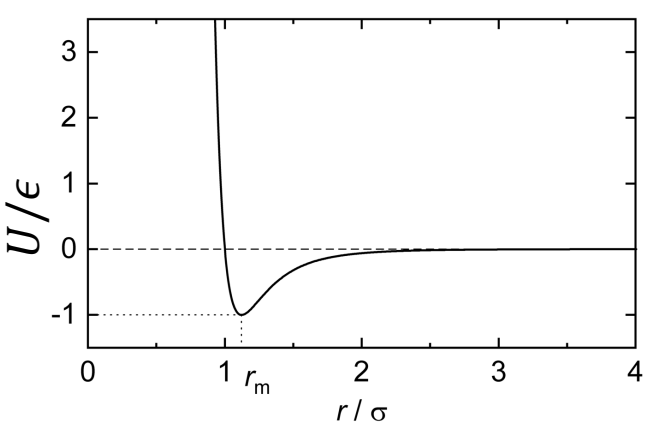
Why interatomic potentials?

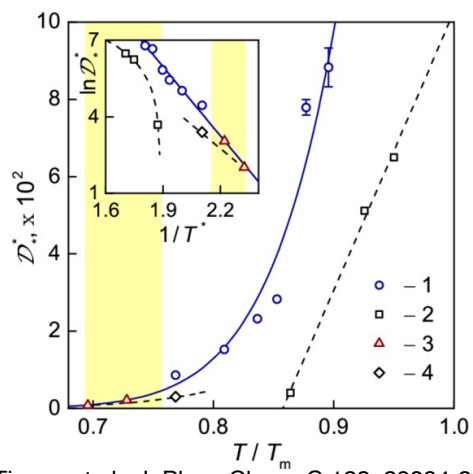
Interatomic potentials: simulate 'large' length-scale or 'long' time-scale phenomena

- Classical force-fields
- Length: ~nm, Time: ~ns (with molecular dynamics)
- Interfaces, diffusivities, rapid phase transitions (→phase diagrams)
- Underlying structure can change (vs. lattice models)
- Computational cost-accuracy trade-off

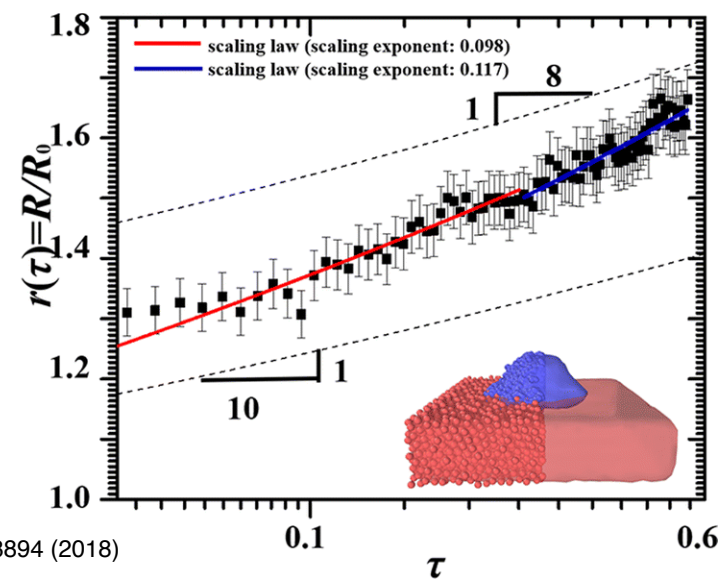
Interatomic potentials model the potential energy surface of a given material

Lennard-Jones: $$U(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$





Tipeev et al., J. Phys. Chem. C 122, 28884-28894 (2018)

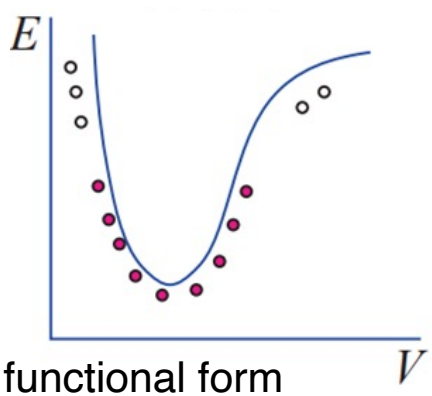


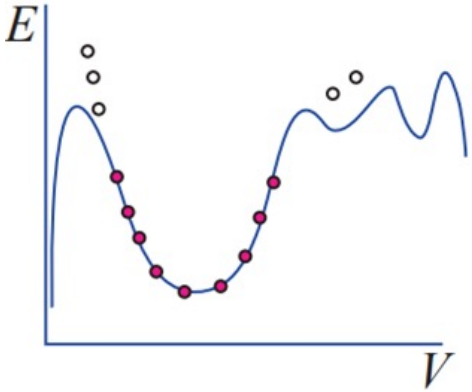
Miao and Yuan, Phys. Chem. Chem. Phys. 25, 7487-7495 (2023)

Why machine learned interatomic potentials (MLIPs)?

Classical force-fields have difficulties in modelling ‘complex’ potential energy surfaces

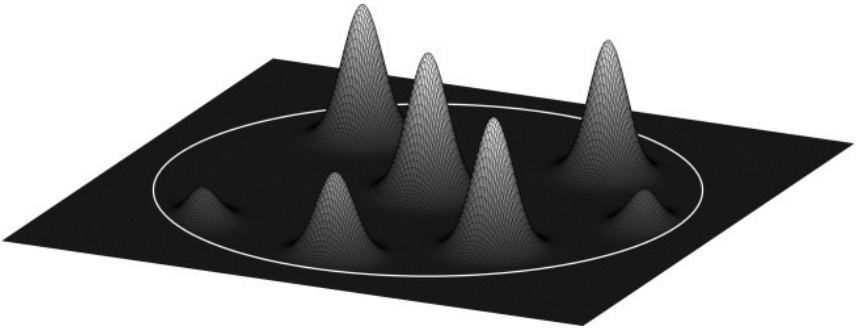
- Diversity of species and bonding environments
- Limited accuracy vs. DFT





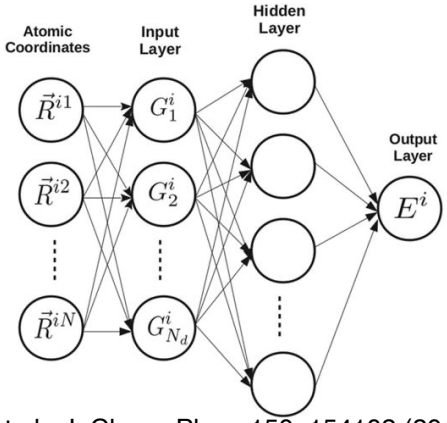
MLIPs: Flexible functional form

- Can handle diversity of species and bonding environments
- Introduce permutation, rotation invariance
- Improved accuracy vs. DFT compared to classical force-fields



Bartók and Csányi, Int. J. Quantum Chem. 116, 1049 (2016)

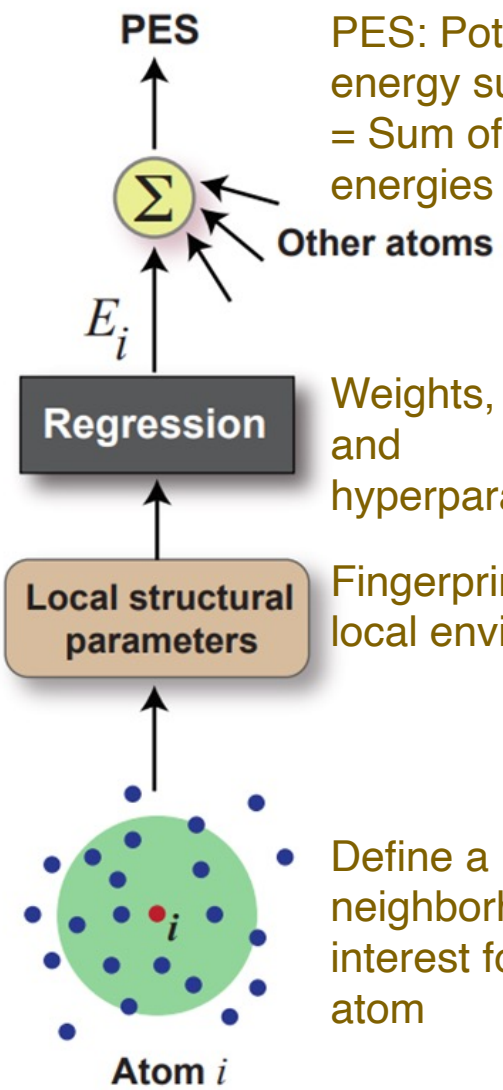
Mishin, Acta Mater. 214, 116980 (2014)



Kocer et al., J. Chem. Phys. 150, 154102 (2019)

Fingerprint a local environment around a reference atom + machine-learning model = MLIP

How do MLIPs work?

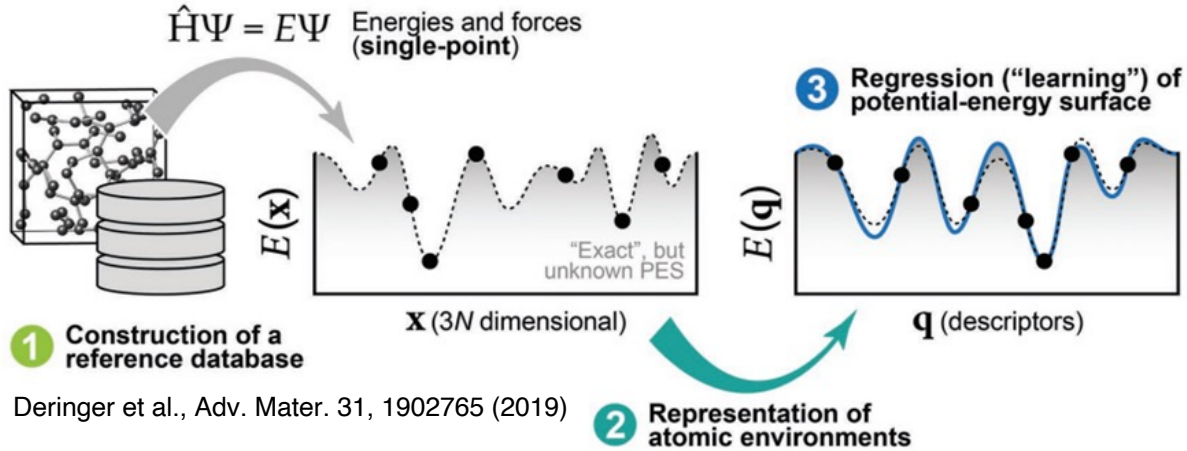


PES: Potential energy surface = Sum of atomic energies

Weights, biases, and hyperparameters

Fingerprint the local environment

Define a neighborhood of interest for each atom



Deringer et al., Adv. Mater. 31, 1902765 (2019)

Typically MLIPs are trained on total energies, atomic forces, and lattice stresses of several different structures in a chemical space

Popular MLIPs:

- Artificial neural network potential (ANNP)
- Gaussian approximation potential (GAP)
- **Moment tensor potential (MTP)**
- Spectral neighbor analysis potential (SNAP)
- **Neural equivariant interatomic potential (NequIP)**

Mishin, Acta Mater. 214, 116980 (2014)

Moment tensor potential: 'classic'

$$E^{\mathrm{mtp}}(\mathrm{cfg}) = \sum_{i=1}^{n} V(\mathfrak{n}_i)$$

n_i - atomic environment (within a cut-off radius) comprising of a reference atom, its neighbours, and their relative positions

V: function invariant to permutations, rotations, and reflections

- Smooth with respect to exchange of atoms from neighborhood

$$V(\mathfrak{n}_i) = \sum_{\alpha} \xi_\alpha B_\alpha(\mathfrak{n}_i)$$

Weights to be fit

Basis functions: written up to a maximum 'level' of 'contracted' moment tensors

Moment tensor:

$$M_{\mu,\nu}(\mathfrak{n}_i) = \sum_j f_\mu(|r_{ij}|, z_i, z_j) \underbrace{\mathbf{r}_{ij} \otimes \ldots \otimes \mathbf{r}_{ij}}_{\nu \text{ times}}$$

$$\mathrm{lev} M_{\mu,\nu} = 2 + 4\mu + \nu$$

$$\mathrm{lev}(\mathrm{M}_{1,2}{:}M_{0,2}) = (2+4+2) + (2+0+2) = 12$$

Radial component

Angular component

Expanded via radial basis functions: pair-wise

$$f_\mu(|r_{ij}|, z_i, z_j) = \sum_{\beta=1}^{N_Q} c^{(\beta)}_{\mu,z_i,z_j} Q^{(\beta)}(|r_{ij}|)$$

Weights to be fit

Chebychev polynomials
× smooth cut-off function

Expanded via tensors: many-body

$\nu = 0$ → Scalar

$\nu = 1$ → Vector; $\mathbf{r}_{ij} = (x_{ij}, y_{ij}, z_{ij})$

$\nu = 2$ → Tensor; $\mathbf{r}_{ij} \otimes \mathbf{r}_{ij} = \begin{pmatrix} x_{ij}^2 & x_{ij}y_{ij} & x_{ij}z_{ij} \\ y_{ij}x_{ij} & y_{ij}^2 & y_{ij}z_{ij} \\ z_{ij}x_{ij} & z_{ij}y_{ij} & z_{ij}^2 \end{pmatrix}$

Novikov et al. Mach. Learn.: Sci. Technol. 2, 025002 (2021)

Moment tensor potential: fitting

$$\sum_{k=1}^{K}\Big[w_e\big(E^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-E^{\mathrm{qm}}(\mathrm{cfg}_k)\big)^2+w_f\sum_{i=1}^{N_k}\left|\mathbf{f}_i^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\mathbf{f}_i^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2$$
$$+w_s\left|\sigma^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\sigma^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2\Big]\to\min_{\theta},$$

Set of k configurations in the training set

θ: parameters to be fit (ξ, c)

qm: DFT or other quantum mechanical tools

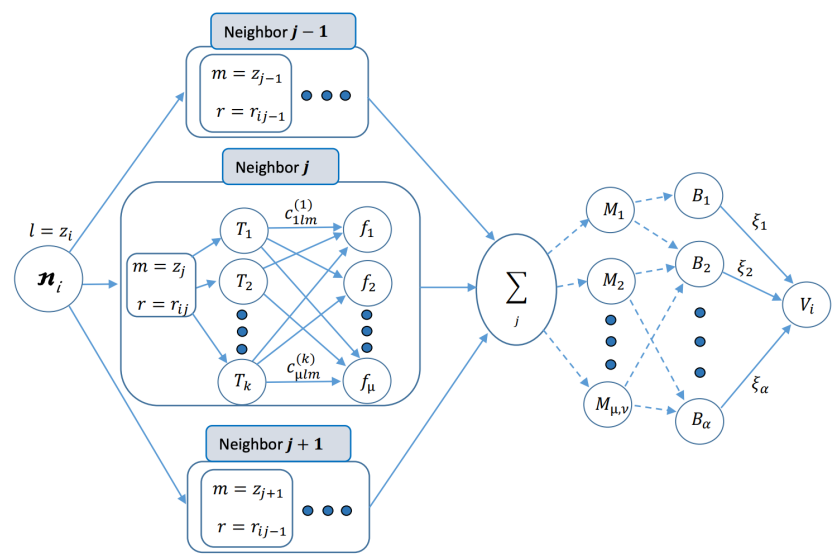
Energies, forces, and stresses considered within loss function

Hyperparameters

$$\mathrm{RMSE}(E)^2=\frac{1}{K}\sum_{k=1}^{K}\left(\frac{E^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)}{N^{(k)}}-\frac{E^{\mathrm{qm}}(\mathrm{cfg}_k)}{N^{(k)}}\right)^2,$$

$$\mathrm{RMSE}(\mathbf{f})^2=\frac{1}{K}\sum_{k=1}^{K}\frac{1}{3\,\mathrm{N}^{(k)}}\sum_{i=1}^{N_k}\left|\mathbf{f}_i^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\mathbf{f}_i^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2$$

$$\mathrm{RMSE}(\sigma)^2=\frac{1}{K}\sum_{k=1}^{K}\frac{1}{9}\left|\sigma^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\sigma^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2.$$



Novikov et al. Mach. Learn.: Sci. Technol. 2, 025002 (2021)
https://www.skoltech.ru/app/data/uploads/2019/09/THESIS_FINAL.pdf

Moment tensor potential: fitting

$$\sum_{k=1}^{K}\Big[w_e\big(E^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-E^{\mathrm{qm}}(\mathrm{cfg}_k)\big)^2+w_f\sum_{i=1}^{N_k}\left|\mathbf{f}_i^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\mathbf{f}_i^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2$$
$$+w_s\left|\sigma^{\mathrm{mtp}}(\mathrm{cfg}_k;\theta)-\sigma^{\mathrm{qm}}(\mathrm{cfg}_k)\right|^2\Big]\to\min_{\theta},$$

Set of k configurations in the training set

θ: parameters to be fit (ξ, c)

qm: DFT or other quantum mechanical tools

Energies, forces, and stresses considered within loss function

Hyperparameters

Once MTP is fit, can be used for both static and dynamic runs

- Using 'LAMMPS' for example

Also has ability to perform active learning during predictions

- Using an 'extrapolation grade'
- Structures outside a confidence interval can be calculated with DFT and the potential retrained

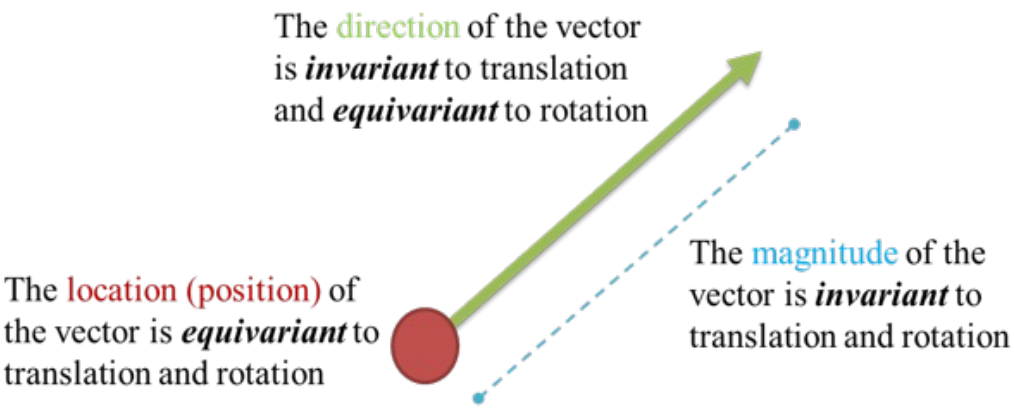
Novikov et al. Mach. Learn.: Sci. Technol. 2, 025002 (2021)
https://www.skoltech.ru/app/data/uploads/2019/09/THESIS_FINAL.pdf

Neural equivariant interatomic potential: ‘recent’

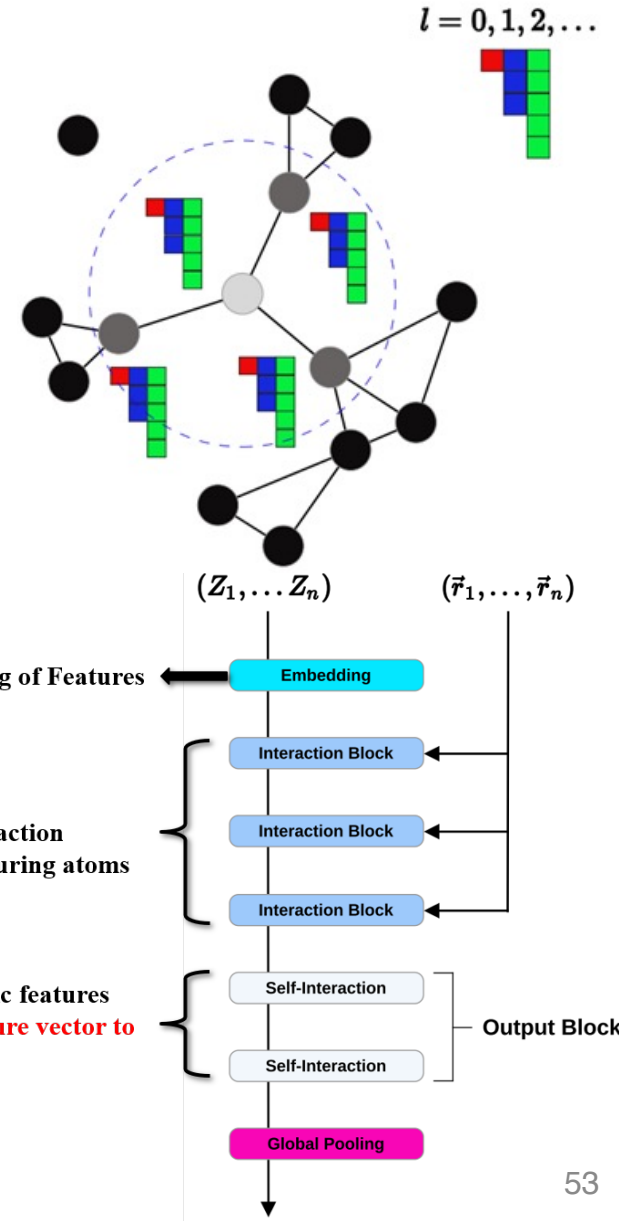
Based on using deep, graph neural networks to construct interatomic potentials
Every atom has a feature vector of different orders (scalars, vectors, and tensors)

$$E_{pot} = \sum_{i \in N_{atoms}} E_{i,atomic}$$

$$\vec{F}_i = -\nabla_i E_{pot}$$

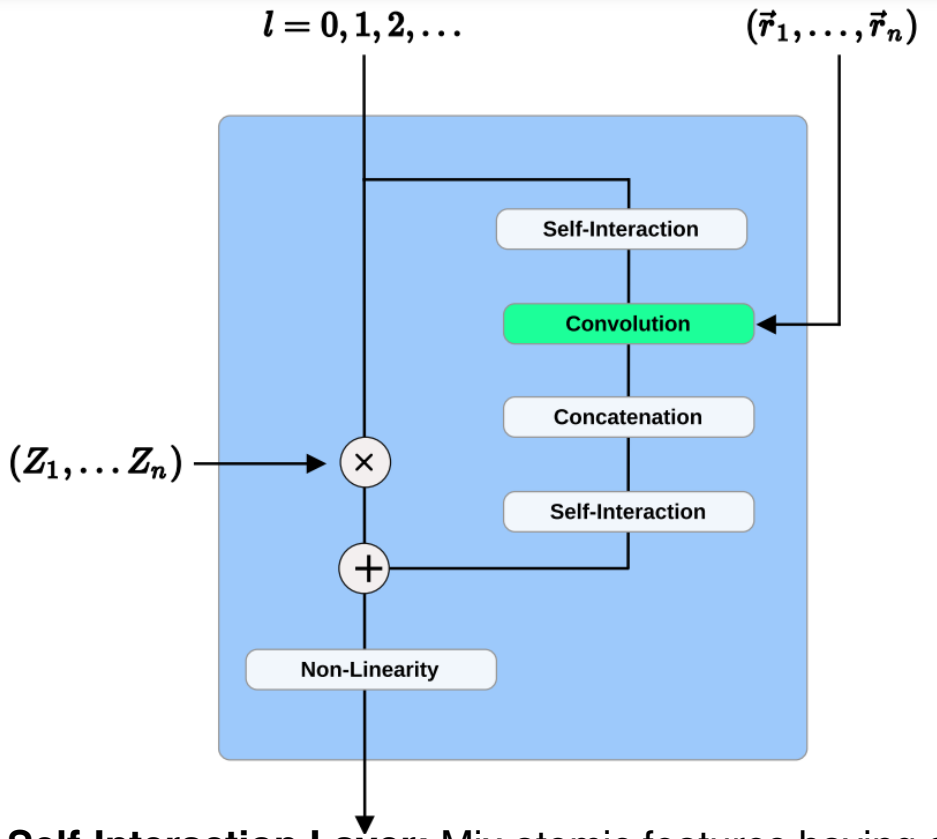


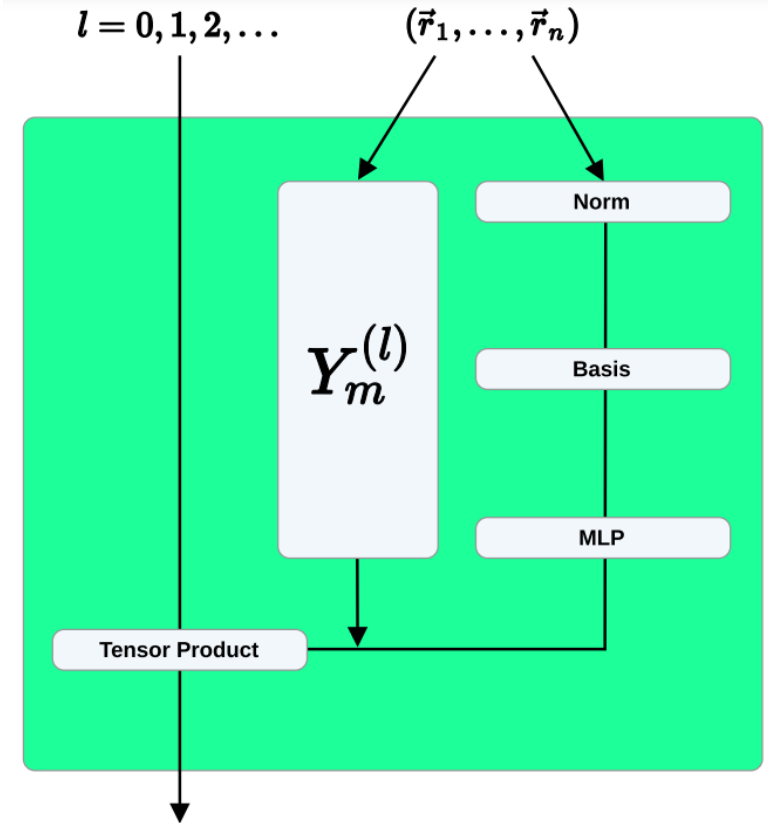
$$\mathcal{L} = \lambda_E ||\hat{E} - E||^2 + \lambda_F \frac{1}{3N} \sum_{i=1}^{N} \sum_{\alpha=1}^{3} \left|\left| -\frac{\partial \hat{E}}{\partial r_{i,\alpha}} - F_{i,\alpha} \right|\right|^2$$



Batzner et al., Nat. Commun. 13, 2453 (2022)

NequIP: code blocks





Self-Interaction Layer: Mix atomic features having same order and mirror parity, reduces dimensionality

Convolution Layer: Rotational equivariance

Concatenation: Recombines feature vectors to form new feature vectors

$$B(r_{ij}) = \frac{2}{r_c} \frac{\sin(\frac{b\pi}{r_c} r_{ij})}{r_{ij}} f_{env}(r_{ij}, r_c)$$

$$S_m^{(l)}(\vec{r}_{ij}) = R(r_{ij}) Y_m^{(l)}(\hat{r}_{ij})$$

Angular component: spherical harmonics

Batzner et al., Nat. Commun. 13, 2453 (2022)

Examples of MLIPs in action

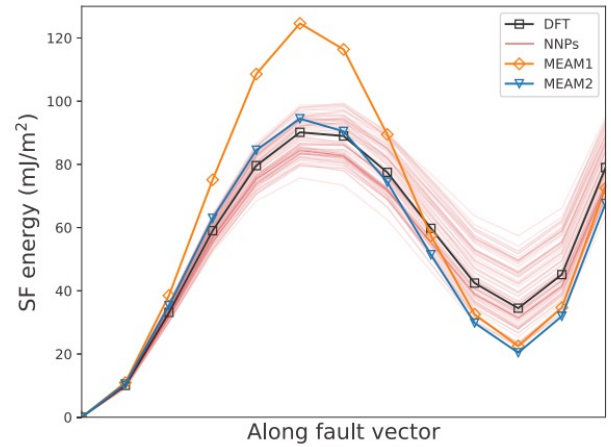
Sample usage of MLIPs so far

Predicting Li migration energies for cathode coating materials **(MTP)**

Composition	MTP E_a (eV)	Experimental E_a (eV)
$Li_3Sc_2(PO_4)_3$	0.62 ± 0.04	0.65
$Li_2B_6O_9F_2$	0.79 ± 0.10	0.92
LiCl	1.11 ± 0.13	0.83

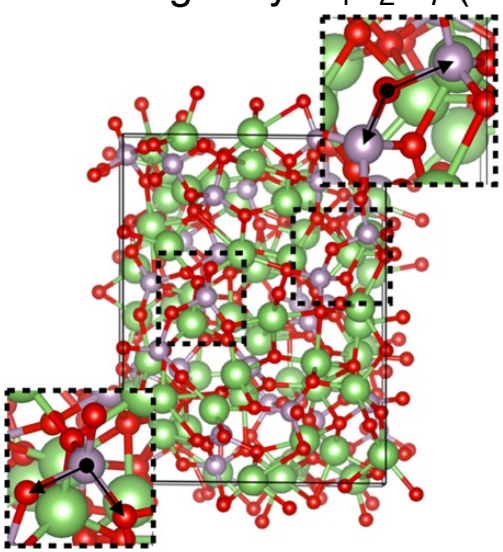
Wang et al., Chem. Mater. 32, 3741–52 (2020)

Effect of defects on deformation and failure in Mg (**ANNP**)



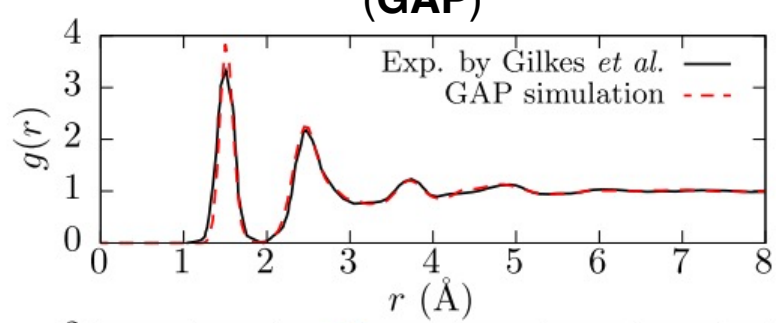
Stricker et al., Phys Rev Mater 4, 103602 (2020)

Simulations of glassy $Li_4P_2O_7$ (**NequIP**)



Batzner et al., Nat. Commun. 13, 2453 (2022)

Growth mechanism in amorphous carbon (**GAP**)



Caro et al., Phys. Rev. Lett. 120, 166101 (2018)

Hands—on session?

Build MTP and NequIP

https://gitlab.com/ashapeev/mlip-2

```
_________________Errors report_________________
Energy:
        Errors checked for 83 configurations
        Maximal absolute difference = 0.937032
        Average absolute difference = 0.0810972
        RMS     absolute difference = 0.140761

Energy per atom:
        Errors checked for 83 configurations
        Maximal absolute difference = 0.0669308
        Average absolute difference = 0.00579266
        RMS     absolute difference = 0.0100543

Forces:
        Errors checked for 1162 atoms
        Maximal absolute difference = 1.06028
        Average absolute difference = 0.0441032
        RMS     absolute difference = 0.0917132
        Max(ForceDiff) / Max(Force) = 0.189541
        RMS(ForceDiff) / RMS(Force) = 0.321722
```

https://github.com/mir-group/nequip.git

```
--- Final result: ---
              f_mae =  2.358706
             f_rmse =  3.224275
            H_f_mae =  1.773994
            C_f_mae =  3.026947
         psavg_f_mae =  2.400471
           H_f_rmse =  2.496266
           C_f_rmse =  3.893006
        psavg_f_rmse =  3.194636
              e_mae =  1.038272
            e/N_mae =  0.069218
              f_mae =  2.358706
             f_rmse =  3.224275
            H_f_mae =  1.773994
            C_f_mae =  3.026947
         psavg_f_mae =  2.400471
           H_f_rmse =  2.496266
           C_f_rmse =  3.893006
        psavg_f_rmse =  3.194636
              e_mae =  1.038272
```

Conclusions and some thoughts to chew

- Designing better materials critical for performance improvement in several applications
  - Computations + ML can significantly accelerate materials design

- Different ways to use ML (or precursors to ML)
  - Regressions (or classifications): predict properties using experimental/calculated properties
  - Coarse graining: model larger/longer phenomena on a fixed lattice
  - Interatomic potentials: model larger/longer phenomena on a dynamic lattice

- Materials science is a data-limited domain
  - Garbage in = Garbage out; data normalization
  - What model to choose? Simple models are usually better
  - Choose features carefully: physically intuitive?
  - Don't do ML just because you can (hammer doesn't beget a nail)
  - Construct models with care: overfitting, lack of transferability