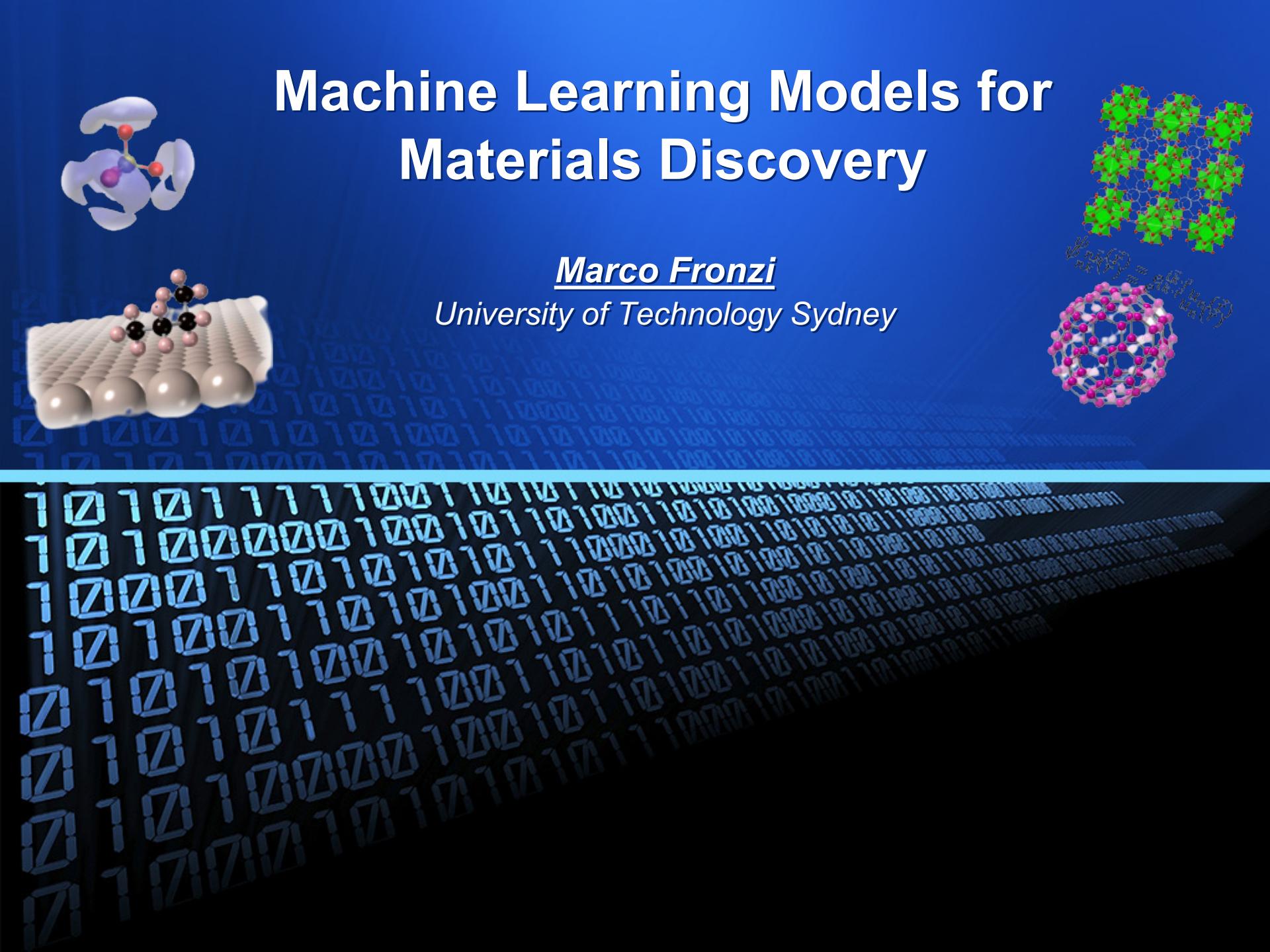


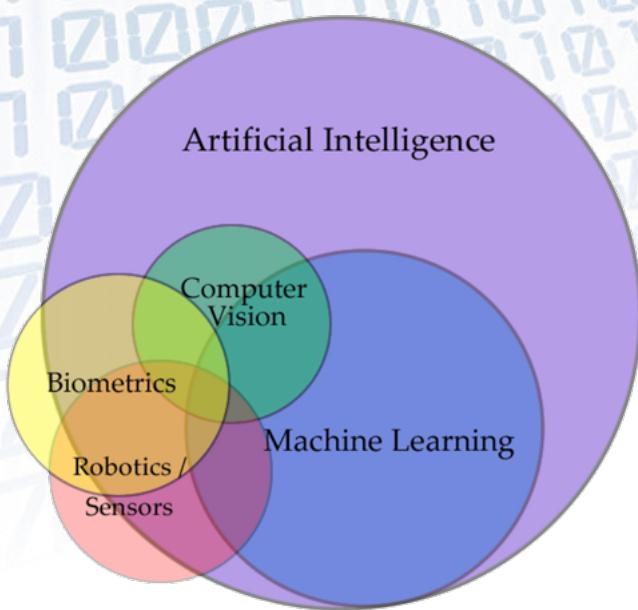
Machine Learning Models for Materials Discovery

Marco Fronzi

University of Technology Sydney

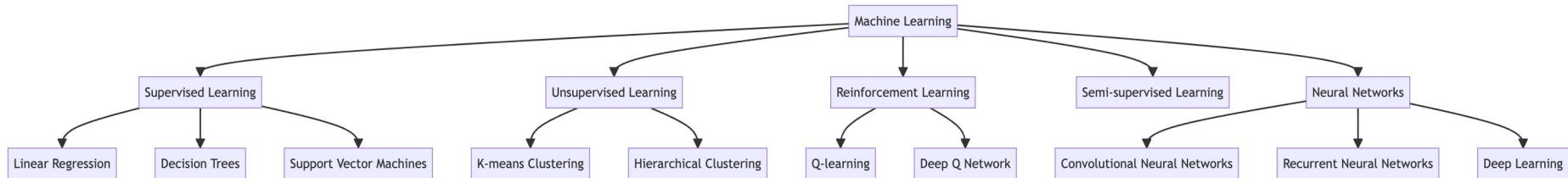


Artificial Intelligence and Machine learning



Machine Learning:

subset of artificial intelligence that focuses on **algorithms and models that enable computers to learn and make predictions or decisions without explicit programming**. It involves the development of computational systems that can automatically learn and improve from experience, without being explicitly programmed for every specific task.

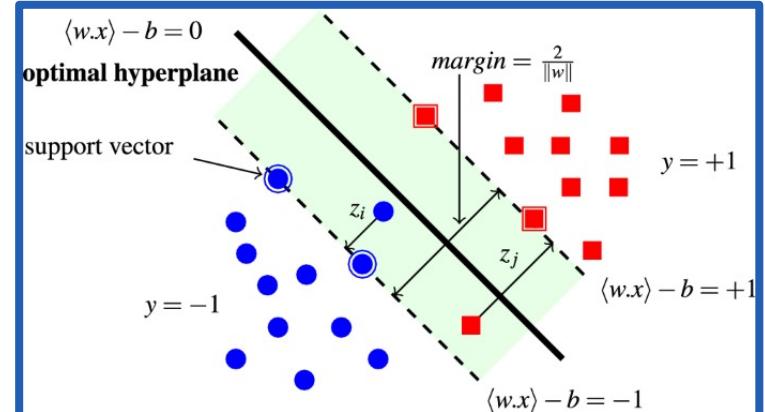
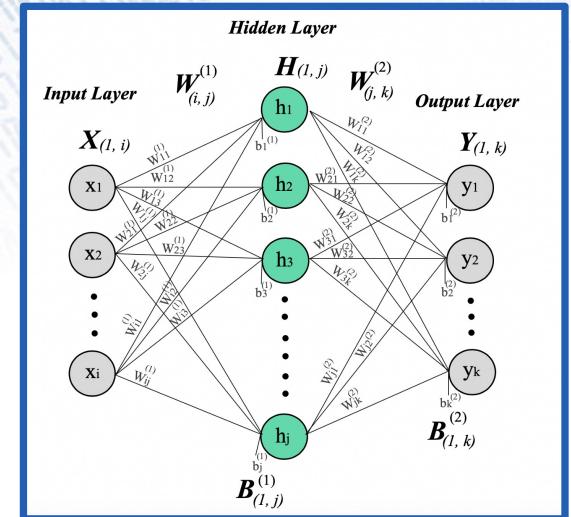
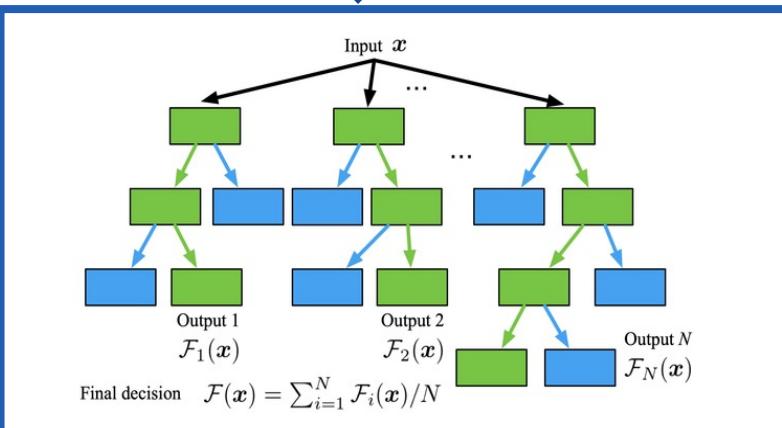


Second Slide Master

Neural Networks

Support Vector Machine

Random Forest



What is a Machine Learning Model?

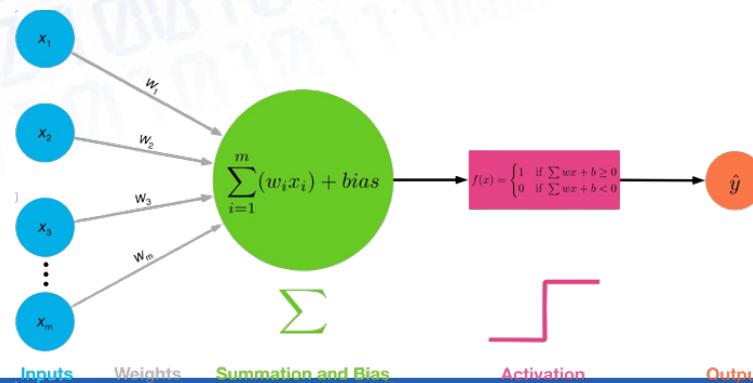
Machine learning object is to find transfer function
To map structural parameters to target property

\vec{x}
**Parameter Space
(Descriptors or Features)**

$$f_{\lambda_1, \lambda_2, \dots, \lambda_m}(x_1, x_2, \dots, x_n) = y$$

y

Target Property



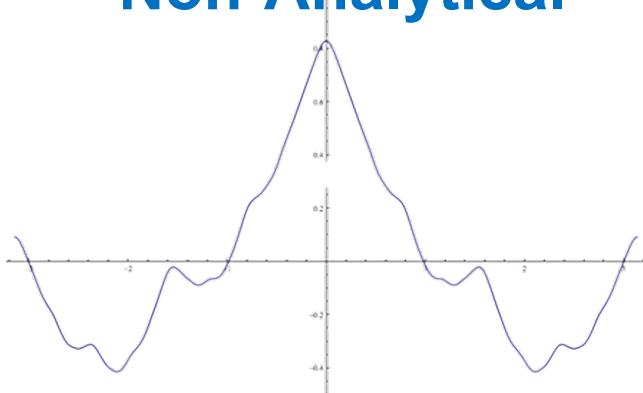
ML model is a function that maps n-dimensional space to a number
(regression = continuous distribution)
(classification = discrete distribution)

If it maps to a **discrete space (integer numbers)** = **classification**
If it maps to a **continuous space (real numbers)** = **regression**

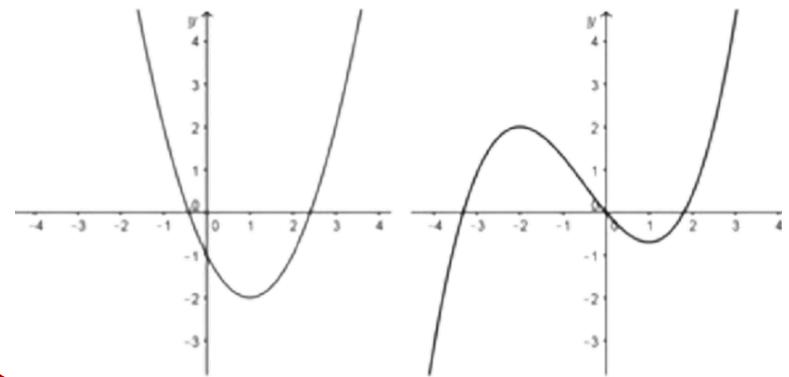
ML Models Properties

Machine learning object is to find transfer function
To map structural parameters to target property

Non-Analytical



Analytical



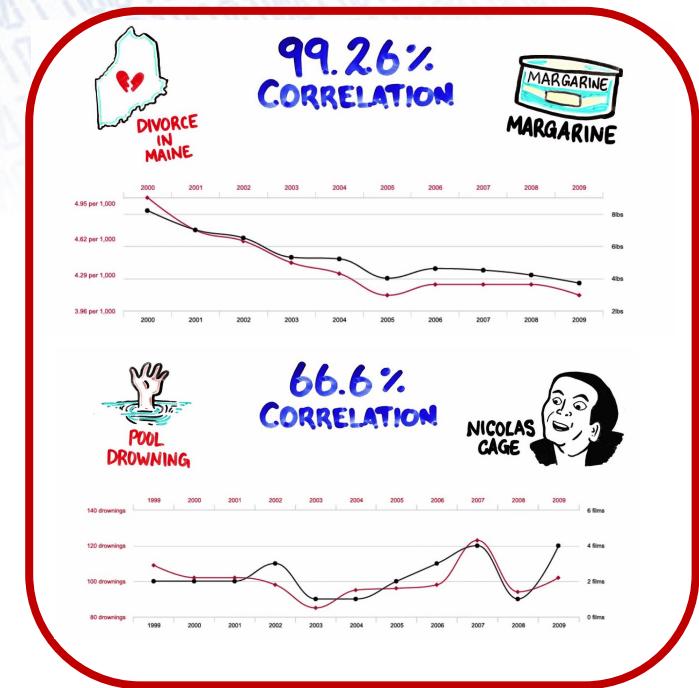
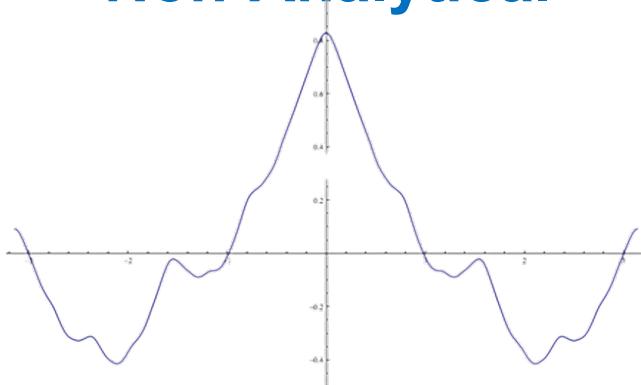
Properties= highly non linear; analytically unknown;
numerically known;

Depends on internal parameters (hyper-parameter)

ML Models Properties

Machine learning object is to find transfer function
To map structural parameters to target property

Non-Analytical



Machine learning models

Pro: very good at finding correlations

Cons: very bad at understanding causation

ML Models Properties

Pros:

- 1. Versatility:** ML models can be applied to a vast array of problems, not just those in the physical sciences but across industries, such as finance, healthcare, marketing, etc.
- 2. Adaptability:** ML models can learn from new data, improving their accuracy and effectiveness over time.
- 3. Efficiency:** be faster and more computationally efficient than complex QM simulations.

Cons:

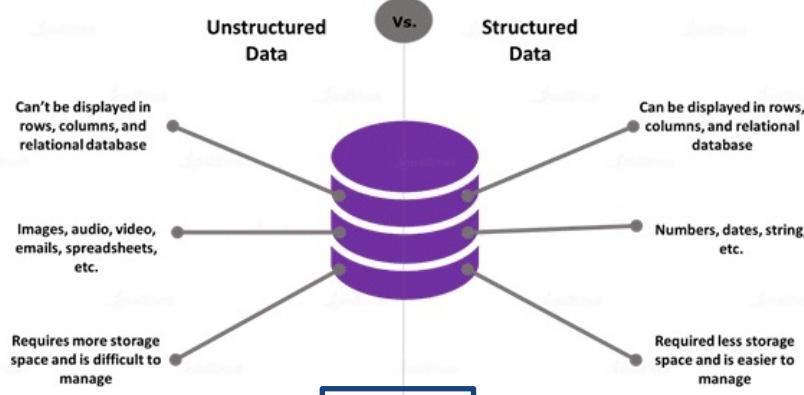
- 1. Data Dependence:** The accuracy of ML models is heavily dependent on the quality and quantity of the training data.
- 2. Black box:** ML models can act as "black boxes," making it hard to understand why they make certain predictions
Deep understanding of the science is needed

Building a ML Model

$$f_{\lambda_1, \lambda_2, \dots, \lambda_m}(x_1, x_2, \dots, x_n) = y$$

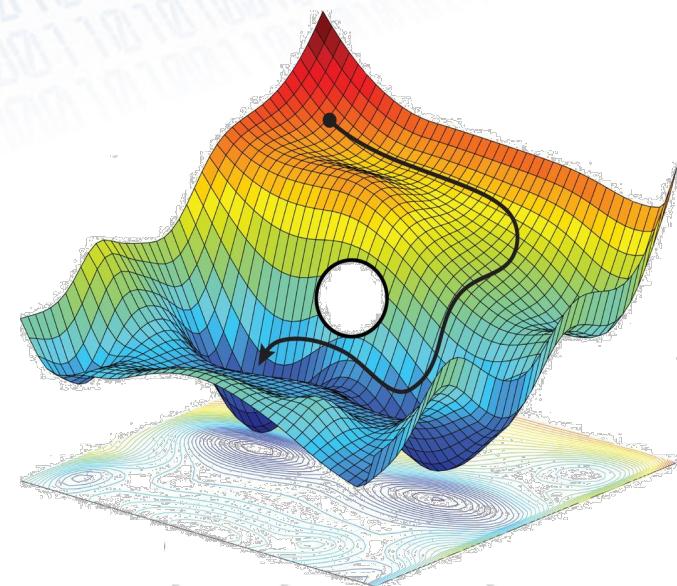
Data-set to provide as example
 (Comparable to Experience)

$$(x_1, x_2, \dots, x_n)$$



$$y$$

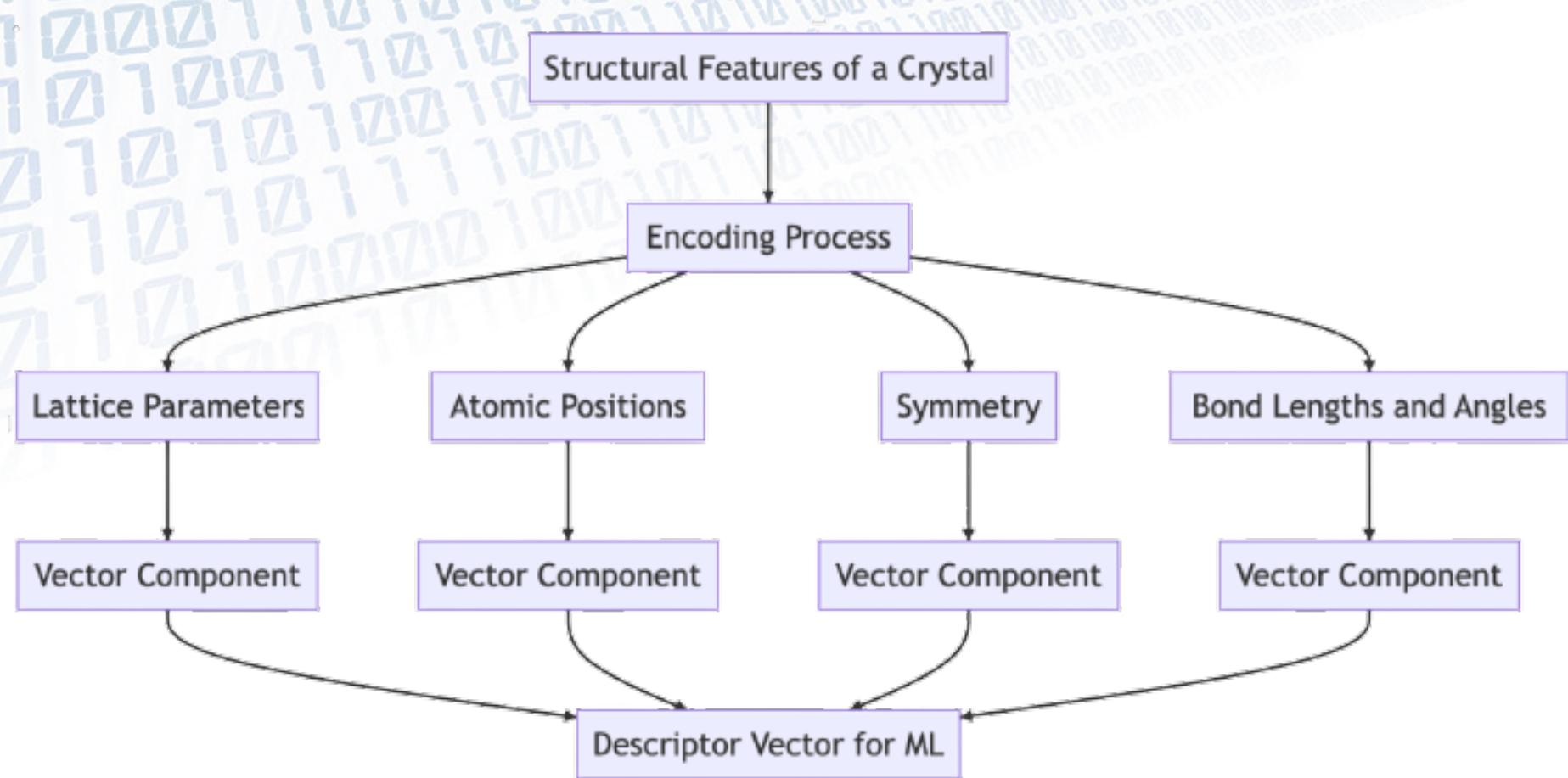
Lot of work to do !!!



Easy task!
Many optimizers available

Featurization

Convert available material information (e.g. structural, elemental) into a numerical vector that uniquely describe the crystal



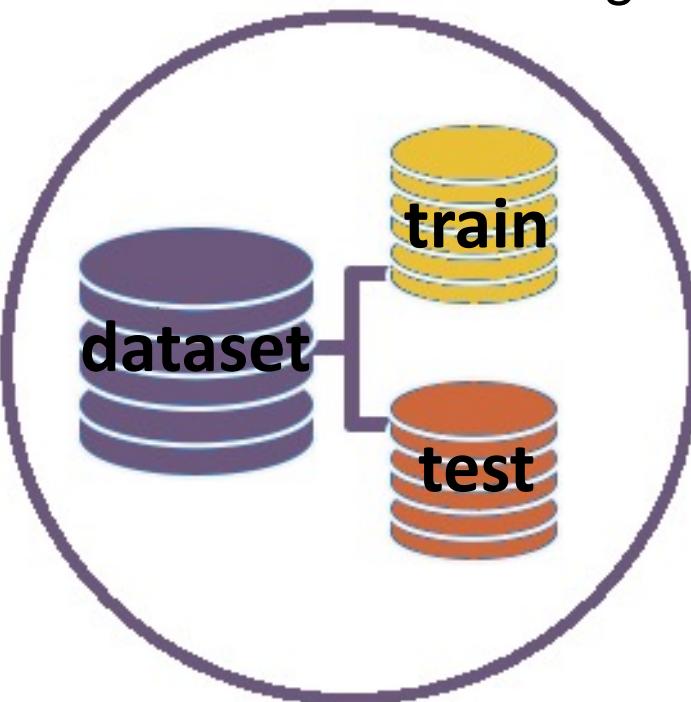
In this context, Feature and Descriptors are interchangeable words

Select Samples

Two sets are necessary to build a model:

Training set used to optimize the model (tune the hyperparameters)

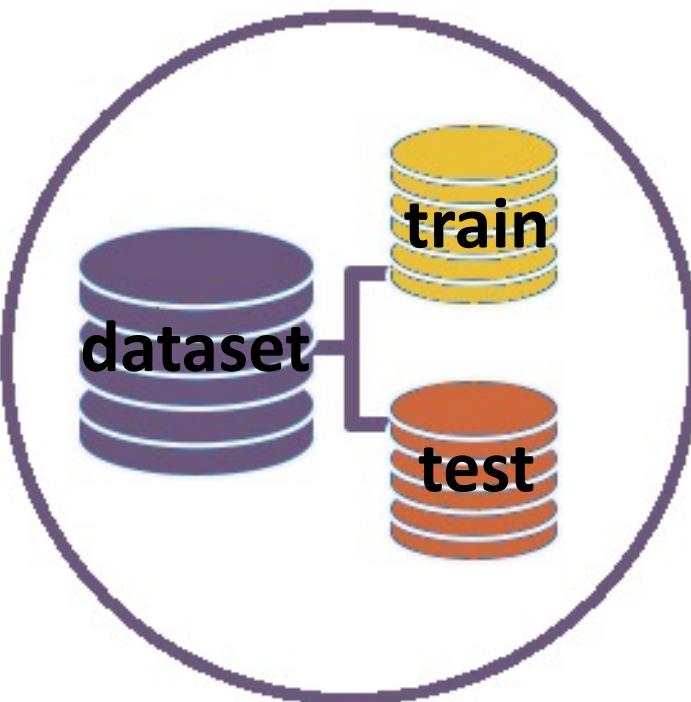
Test set: use to check quality of the model after optimization
Test if the model is able to generalize with reasonable error



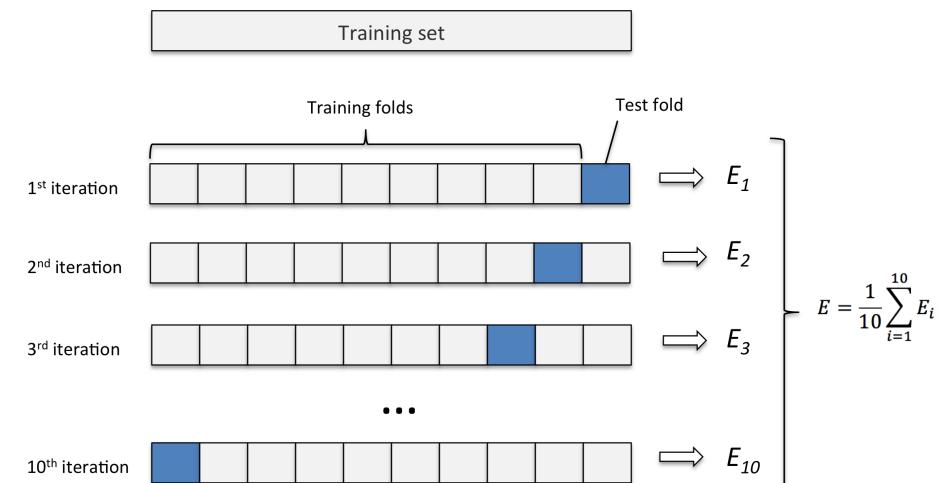
Train:test=80:20

Cross Validation

Perform multiple optimization choosing each time a different training test subset



Train:test=80:20



Optimization

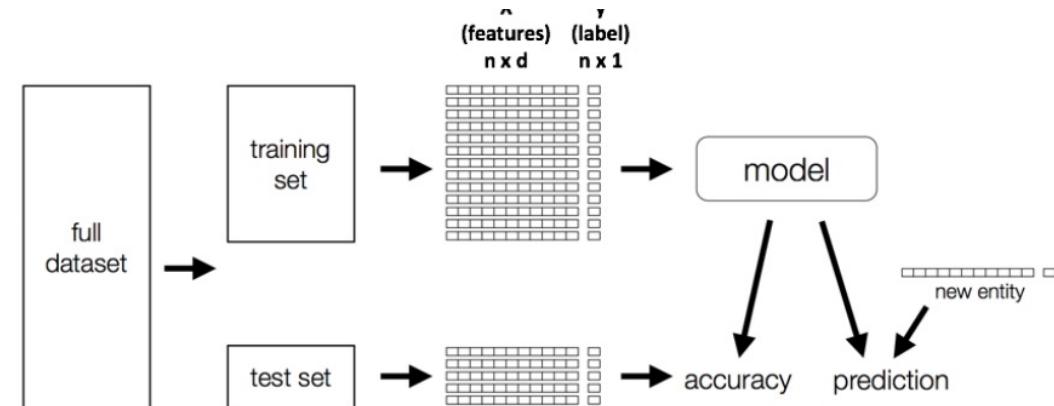
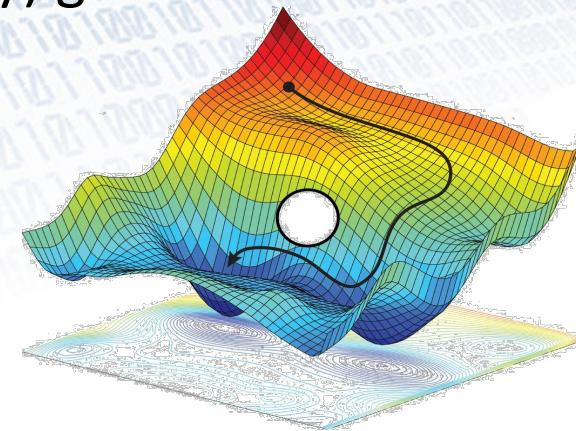
Model training is a search for (energy) global minimum

Algorithms

=

**provide M examples
and search for
global minimum**

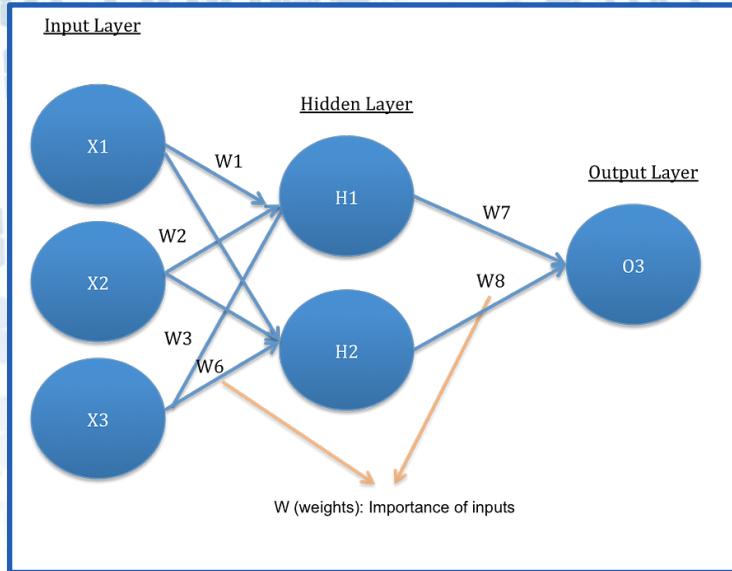
**by changing hyper-parameters
values**



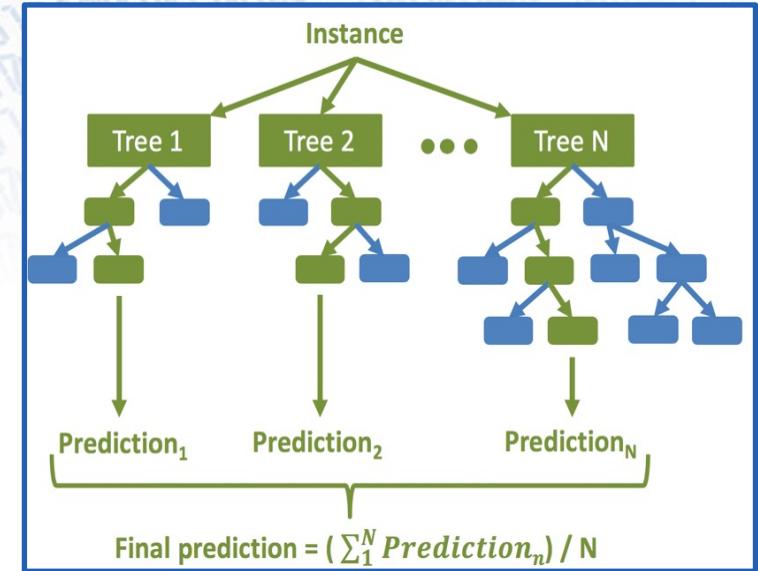
In this context, Feature and Descriptors are interchangeable words

Neural Networks and Random Forest

Neural Networks



Random Forest



Neural networks are computational models inspired by the structure and functioning of the human brain. They consist of interconnected nodes, called neurons, organized in layers. Each neuron takes inputs, performs a computation, and produces an output. Neural networks can learn complex patterns and relationships in data through a process called training.

Random forests are ensemble learning models that combine multiple decision trees to make predictions. Each tree in the forest is built independently using a random subset of the training data and features. When making predictions, each tree averages its individual output. A decision tree is a flowchart-like structure that uses rules based on features to make predictions. It starts with a root node, splits the data based on conditions, and reaches outcome predictions at the leaf nodes.

Neural Networks

$$y = f \left(\sum_{i=1}^n w_i \cdot x_i + b \right) \quad (1)$$

Where:

- y is the output of the neuron,
- f is the activation function (e.g., sigmoid, ReLU, tanh),
- w_i are the weights,
- b is the bias,
- n is the number of inputs.

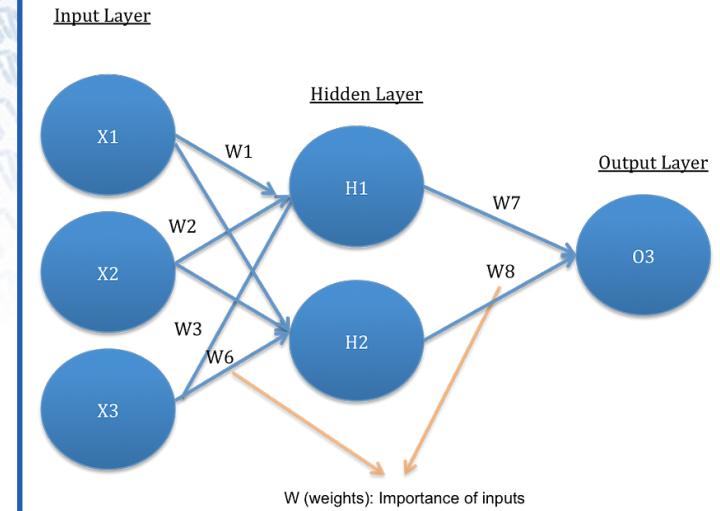
This equation represents the basic idea behind the weight update in neural networks. There are also advanced optimization algorithms like Adam, RMSprop, and others that have slightly different update rules, but they are all based on the concept of gradient descent.

During the optimization of Neural Networks, the weights w_i are adjusted using the gradient descent algorithm or its variants. The basic update rule for a weight w_i in gradient descent is:

$$w_i = w_i - \alpha \frac{\partial L}{\partial w_i} \quad (2)$$

Where:

- w_i is the weight being updated.
- α is the learning rate, a hyperparameter that determines the step size during the weight update.
- L is the loss function that measures how far the network's predictions are from the actual target values.
- $\frac{\partial L}{\partial w_i}$ is the partial derivative of the loss function with respect to the weight w_i , also known as the gradient. It indicates how the loss would change with respect to a small change in w_i .



– R^2 :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

– Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

– Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Random Forest

- Random Forests build multiple decision trees using the datasets.
- Each tree is built by recursively making binary splits in the dataset.
- The splits are made by minimizing a loss function. For regression, common loss functions are Mean Squared Error (MSE) and Mean Absolute Error (MAE). For classification, common measures are Gini Impurity and Entropy.

- R^2 :

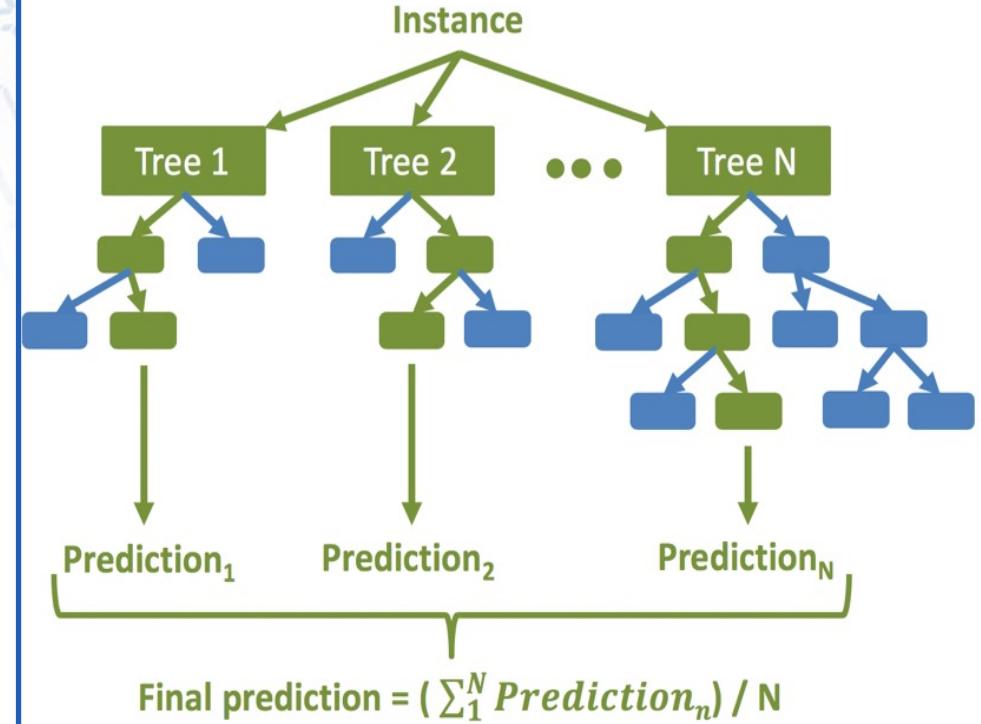
$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

- Mean Squared Error (MSE):

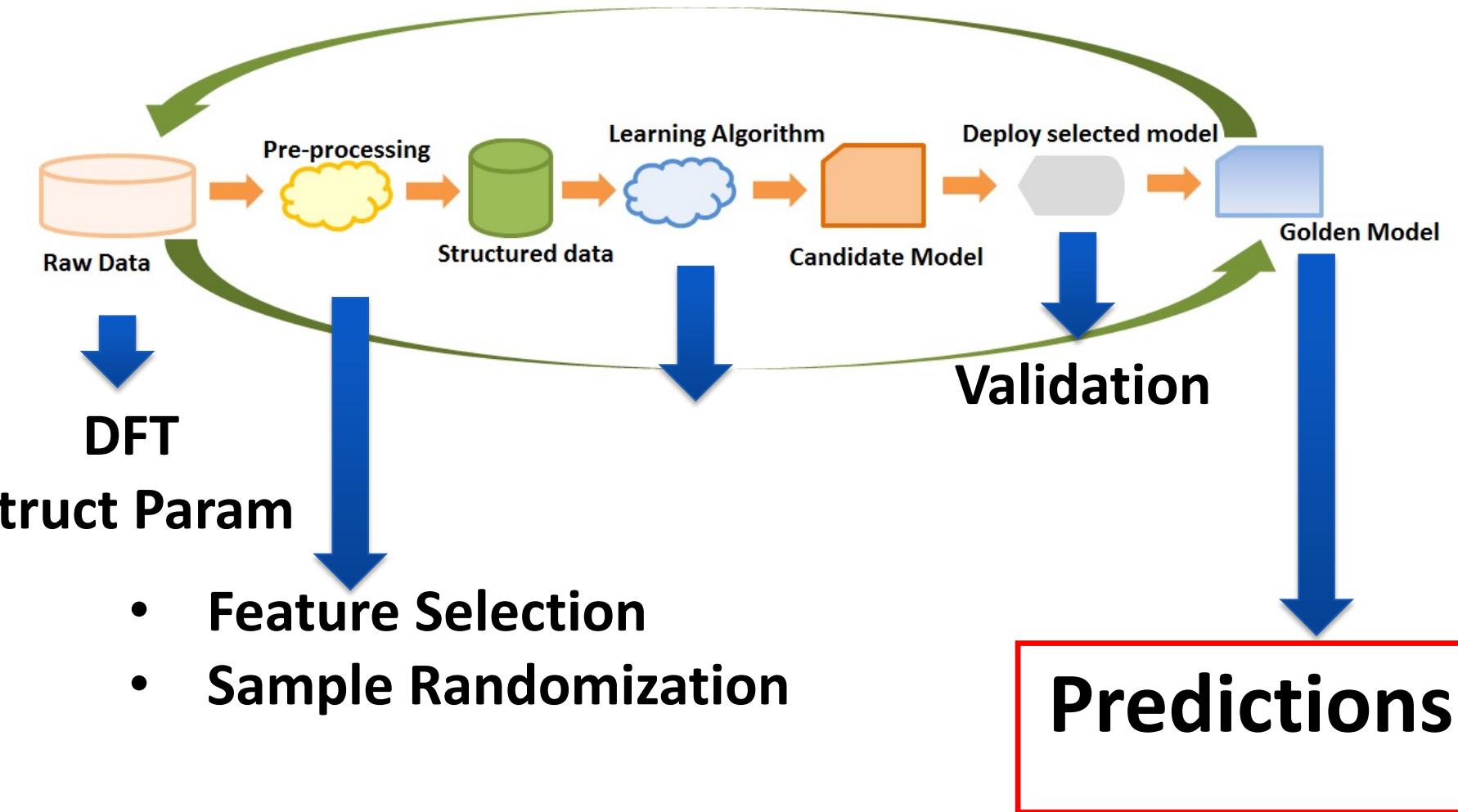
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

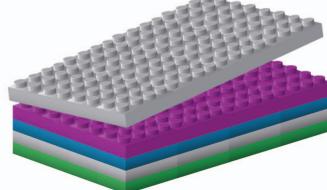
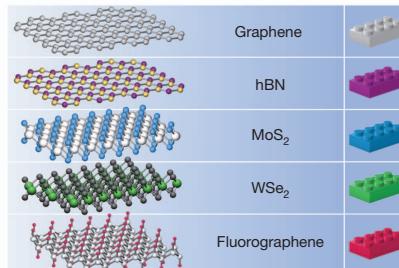
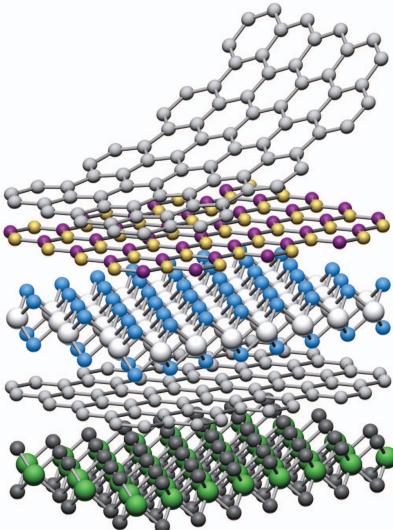


Machine Learning Workflow



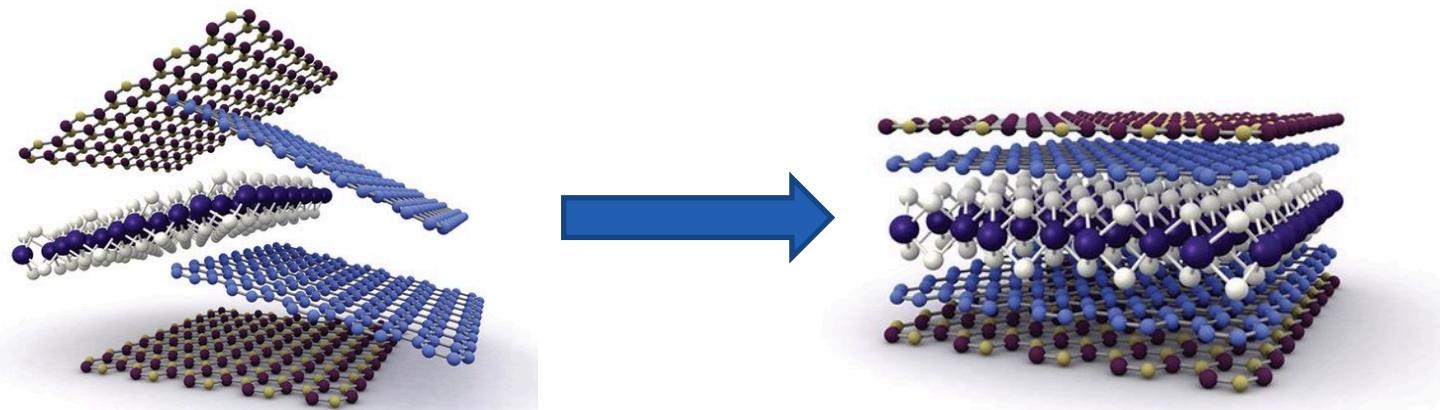
Novel 2D structures as lego blocks

Graphene family	Graphene	hBN 'white graphene'	BCN	Fluorographene	Graphene oxide
2D chalcogenides	MoS ₂ , WS ₂ , MoSe ₂ , WSe ₂		Semiconducting dichalcogenides: MoTe ₂ , WTe ₂ , ZrS ₂ , ZrSe ₂ and so on	Metallic dichalcogenides: NbSe ₂ , NbS ₂ , TaS ₂ , TiS ₂ , NiSe ₂ and so on	
				Layered semiconductors: GaSe, GaTe, InSe, Bi ₂ Se ₃ and so on	
2D oxides	Micas, BSCCO	MoO ₃ , WO ₃	Perovskite-type: LaNb ₂ O ₇ , (Ca,Sr) ₂ Nb ₂ O ₁₀ , Bi ₄ Ti ₃ O ₁₂ , Ca ₂ Ta ₂ TiO ₁₀ and so on	Hydroxides: Ni(OH) ₂ , Eu(OH) ₂ and so on	
	Layered Cu oxides	TiO ₂ , MnO ₂ , V ₂ O ₅ , TaO ₃ , RuO ₂ and so on			Others



- Few structures have been manufactured (a few tens) since 2004
- Two-dimensional crystals can be assembled into hetero-structures where the monolayers are held together by van der Waals forces.
- Self cleaning mechanism (ideal interface)
- Currently predicted 6,138 can generate a very large set of bilayers

Issues with hetero-structures



- Currently predicted 6,138 can generate a very large set of bilayers
- Only a small percentage of the hetero-structures have commensurate cells

$$m = n(n + 1)/2$$
$$\approx 18,000,000$$

Possible bilayers!!!

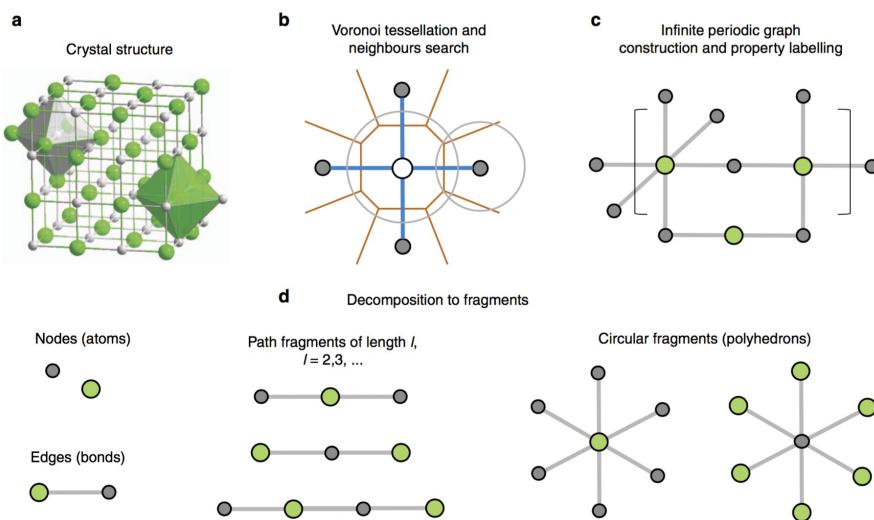
~8.2%

Commensurate cells!!

Structural Parameters

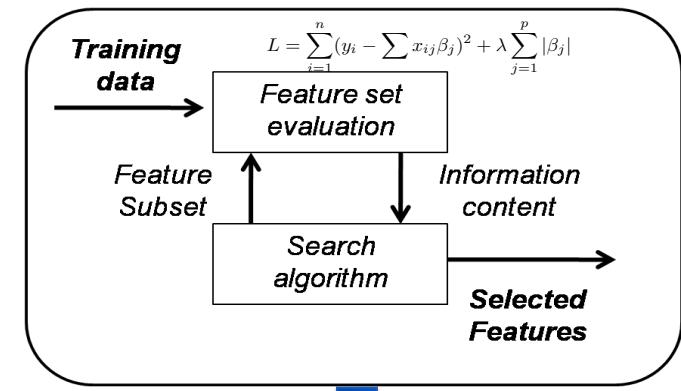
Universal fragment descriptors¹

Each bilayer is uniquely described by a set of ~2700 descriptors

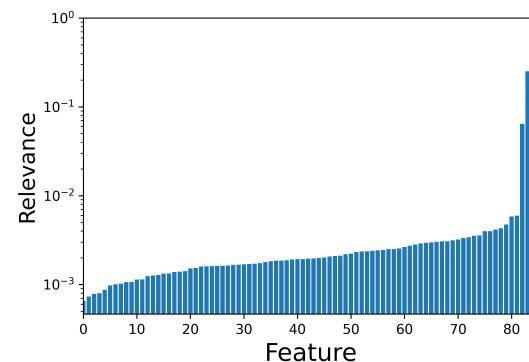


Each crystal structure is represented as a graph, with vertices decorated according to the reference properties of the atoms they represent, and each node is connected to its neighbor according to the Voronoi tessellation.

The final descriptor vector is obtained by partitioning a full graph into sub-graphs called fragments.



GA and LASSO reduced
descriptors to less than 100

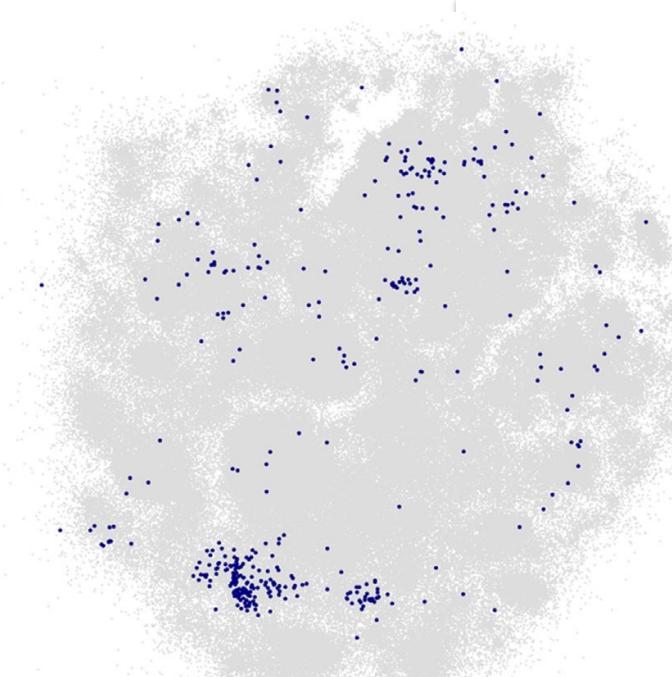


Select examples

Training and test set must be representative of the problem you want to solve

Must be uniformly distributed in the descriptors space

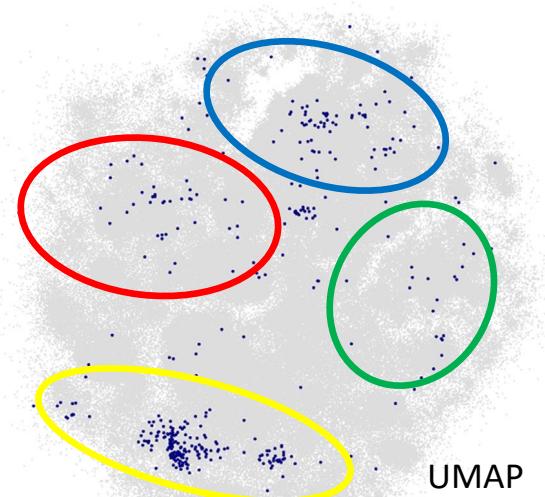
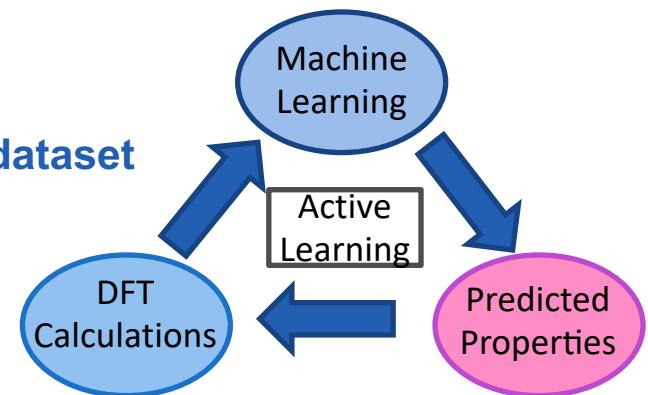
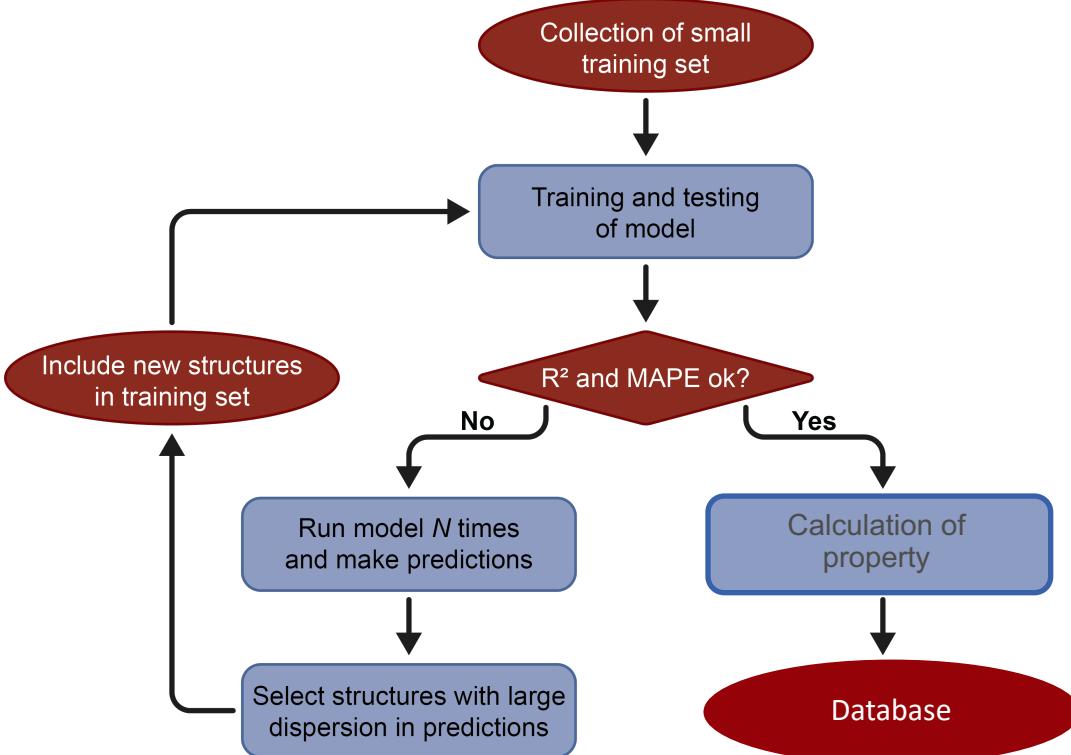
UMAP



Active Machine Learning

- 1) Quality of DFT dataset
- 2) Size of DFT dataset
- 3) Distribution of DFT set in descriptor hyperspace

Active learning is fundamental to select elements of DFT dataset



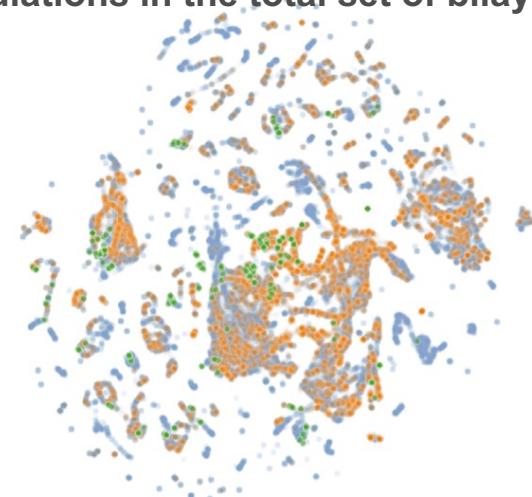
DFT calculated band gap

Total 473 DFT-HSE06 band gaps calculations

Calculated Band Gap vs Literature

Bilayer	Bandgap [eV] This work	Bandgap [eV] Other works
O ₂ Pt–NiO ₂	1.72	1.39 ^[58]
OTl ₂ –GeI ₂	1.83	1.45 ^[58]
Br ₂ Mg–Cl ₂ Zn	5.50	5.49 ^[58]
Cl ₂ Zn–CdCl ₂	5.10	5.29 ^[58]
OTl ₂ –O ₂ Pt	0.62	0.86 ^[58]
I ₂ Yb–Br ₂ Ge	1.30	1.04 ^[58]
InSe/AsP	1.26	1.07 ^[59]
HfS ₂ /MoTe ₂	0.59	0.35 ^[60]

UMAP distribution of DFT calculations in the total set of bilayers



[58] R. Dong, A. Jacob, S. Bourdais, S. Sanvito, *npj 2D Mater. Appl.* **2021**, 5.

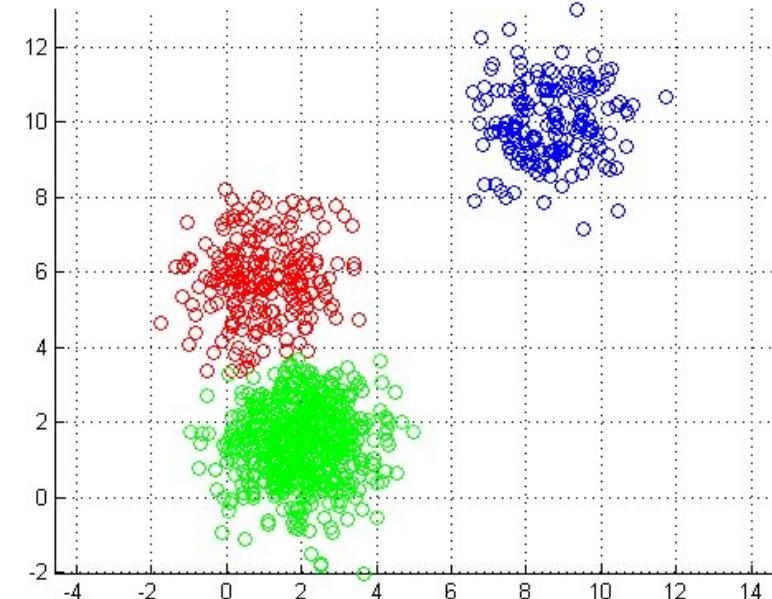
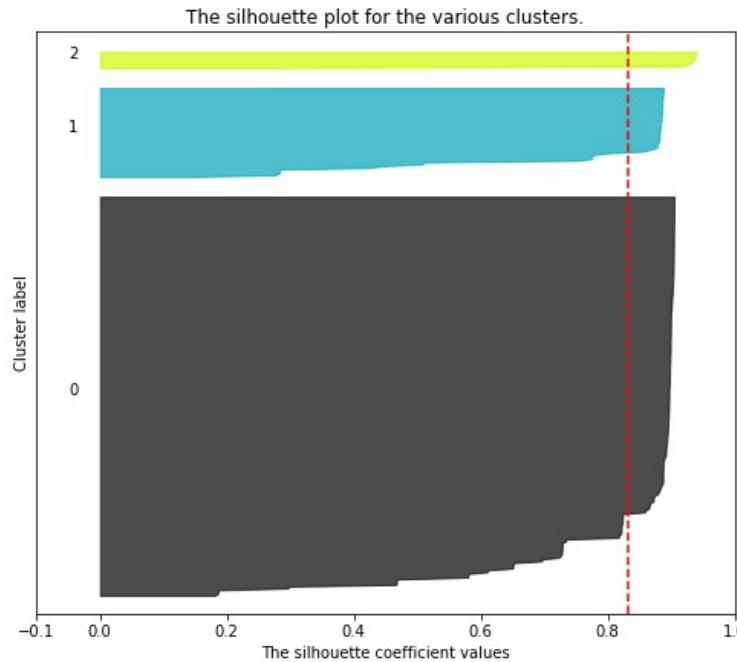
[59] S. Wang, Y. Hu, Y. Wei, W. Li, N. T. Kaner, Y. Jiang, J. Yang, X. Li, *Phys. E* **2021**, 130, 114674.

[60] X. Yang, X. Qin, J. Luo, N. Abbas, J. Tang, Y. Li, K. Gu, *RSC Adv.* **2020**, 10, 2615.

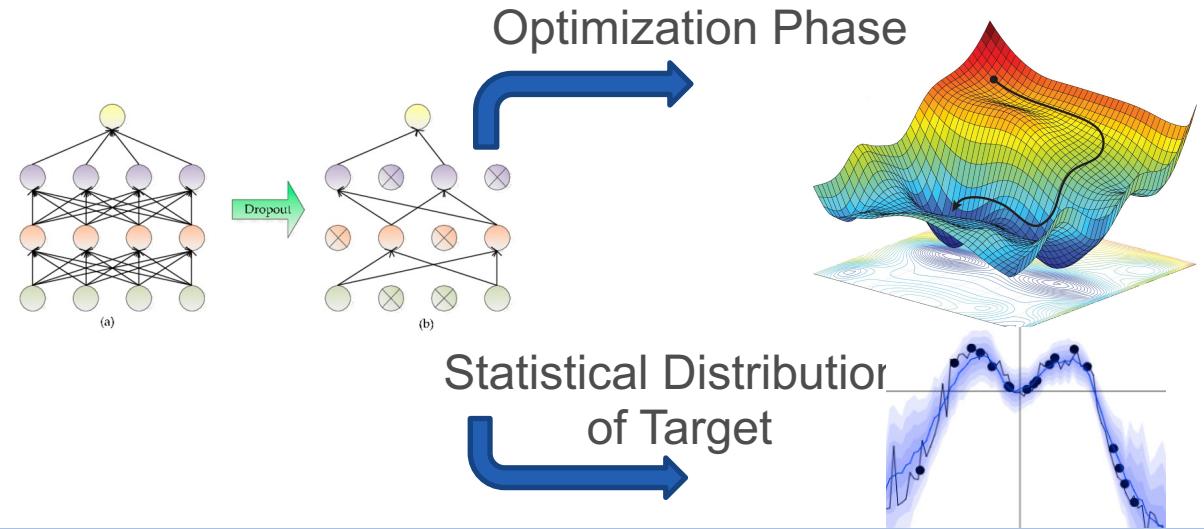
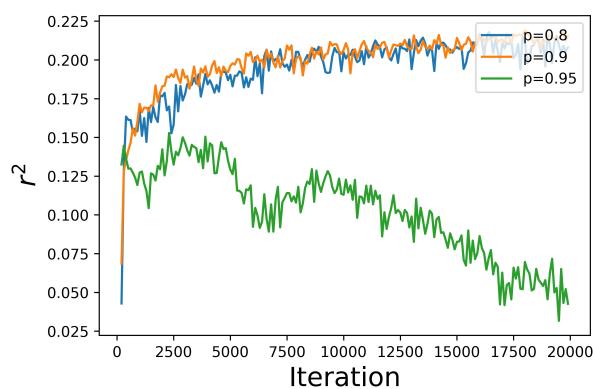
Cluster Analysis for Training Set Randomization

K-Means clustering has been carried out to ensure training samples are representative of the whole set

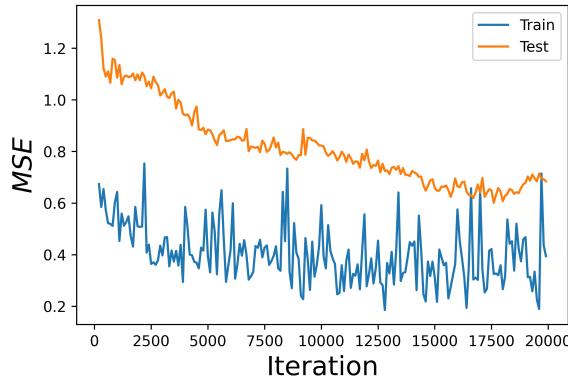
$$J = \sum_{j=1}^k \sum_{i=1}^n ||x_j^i - c_j||^2$$



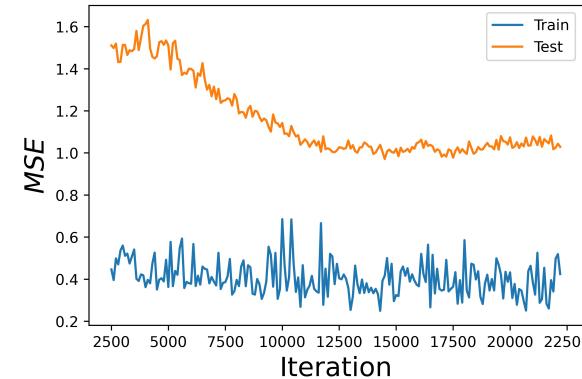
Setup ML dropout and loss function



$$\text{Loss} = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$$



$$\text{Loss} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i|$$



Metrics

Measures of prediction accuracy

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \quad MSE = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{y_i} \right|$$

Measure of the prediction of future outcomes, it measures how well observed outcomes are replicated by the model

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

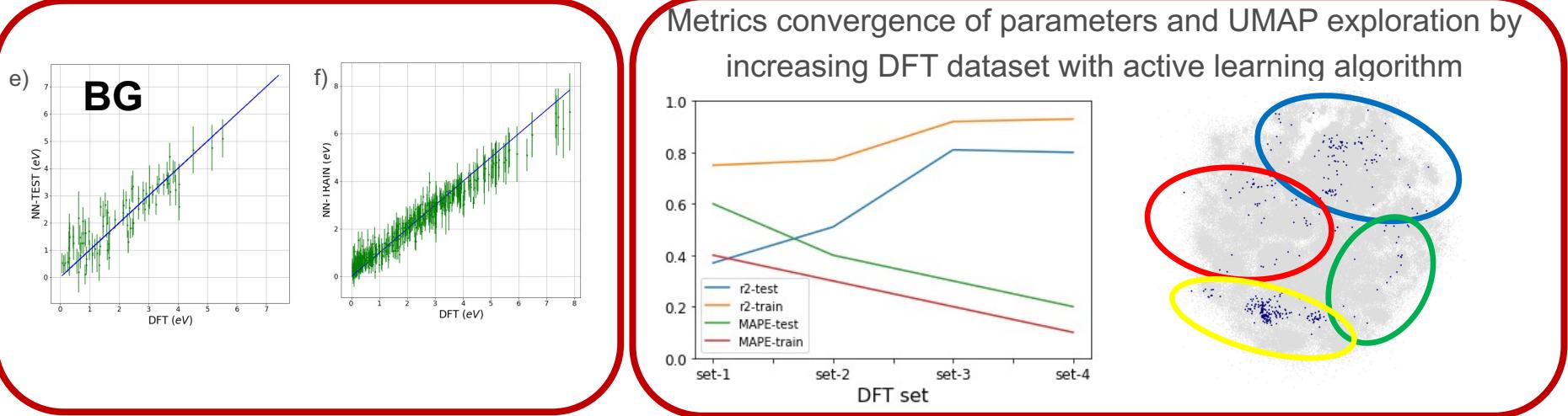
y_i is the predicted data and x_i is the observed data

Loss Function

$$\left. \begin{array}{l} \text{Loss} = \sum_{i=1}^n x_i \log y_i \\ \text{Loss} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \\ \text{Loss} = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \end{array} \right\} + \text{Dropout regularization}$$

y_i is the predicted data and x_i is the observed data

Softmax
32 nodes

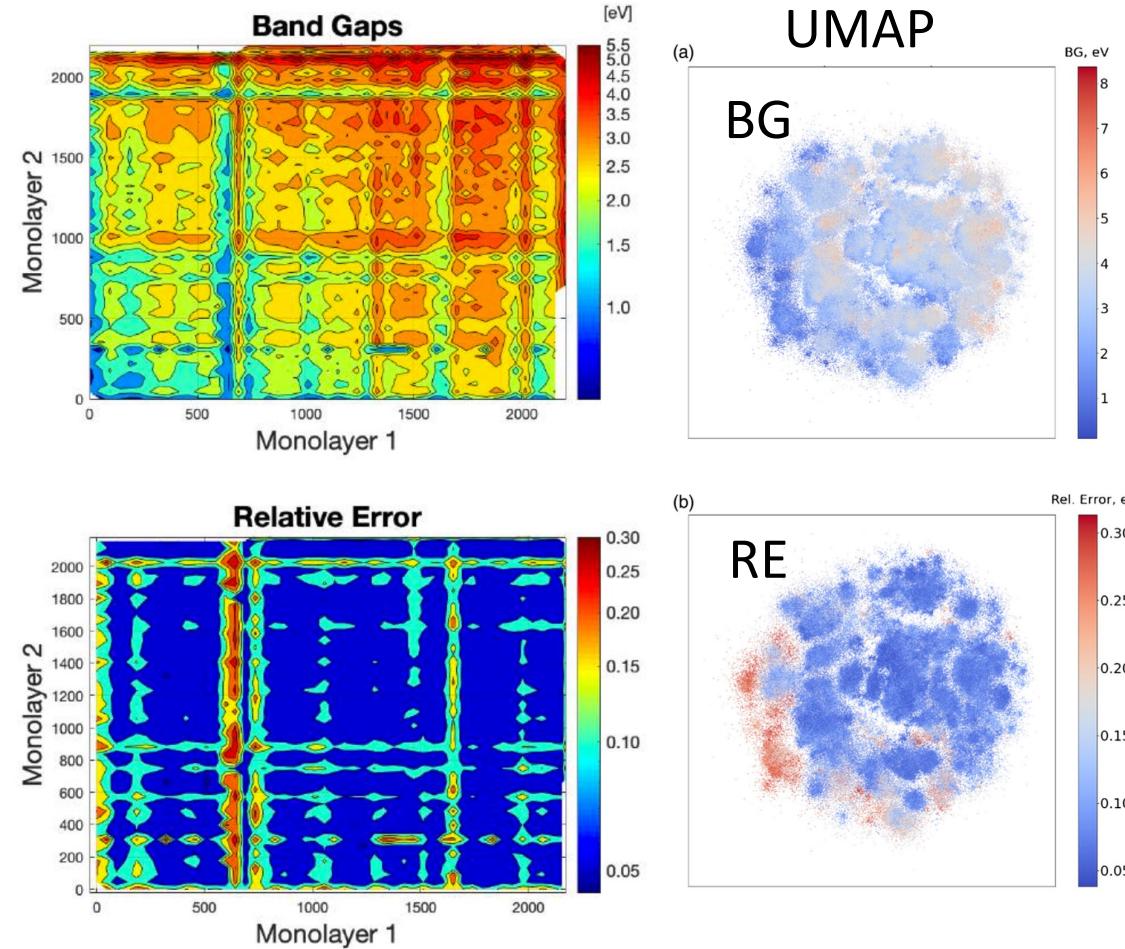


Set	R ²	RMSE [eV]	MAE [eV]	MAPE
BNN-test	0.80	0.44	0.30	0.2
BNN-train	0.93	0.40	0.28	0.1

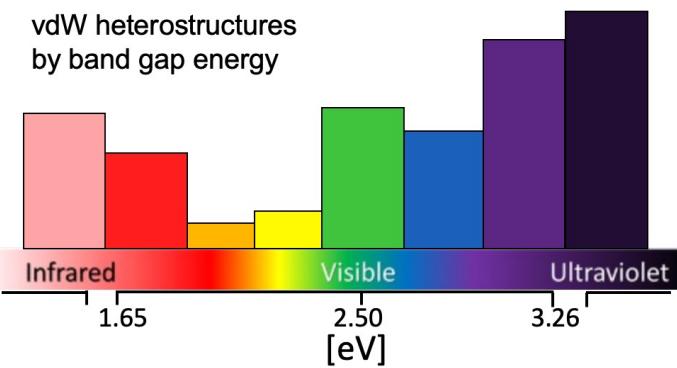
Tests indicate that the model predicts DFT results with accuracy with MAPE<20% and R²>0.8 for the test set

band gap ML prediction for ~2.2M novel structures

$$f(x_1, x_2, \dots x_n)$$

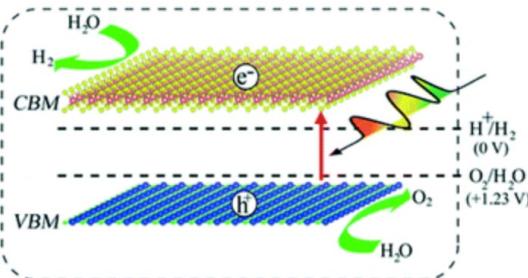


- 2M structures
- Large number of potential semiconductors suitable for energy conversion
- Band gap distribution seems to be clustered



Monolayer frequency by band gap energy

Optical spectrum	IR	R	O	Y	G	B	V	UV
Number of bilayers in band	307 100	217 180	86 089	59 342	320 450	267 739	474 231	538 011
<hr/>								
IR	R	O	Y	G	B	V	UV	
Monolayer	count	Monolayer	count	Monolayer	count	Monolayer	count	
F ₆ Li ₂ O ₃ Ta ₂	2,003	B ₂ O ₆ U	713	Cl ₄ P	343	ClP ₄	197	
C ₆ Li ₂ O ₁₈ V ₃	1,863	F ₃ Zr	694	F ₃ Ti	298	F ₃ Ti	179	
LiO ₁₂ Te ₂ V ₃	1,860	O ₈ Pb ₃ V ₂	683	H ₂ Mg ₃ O ₁₂ Si ₄	227	HfI ₃	154	
Li ₄ O ₁₂ Te ₃ V	1,843	Ni ₂ O ₈ Te ₃	673	Nb ₂ O ₃	223	Br ₃ MoTe ₆	152	
C ₃ Li ₂ O ₉ V	1,805	Ge ₂ Se ₅ Tl ₂	640	Br ₃ MoTe ₆	216	Cl ₈ Ge ₃	143	
G	B	V	UV					
Monolayer	count	Monolayer	count	Monolayer	count	Monolayer	count	
Cl ₃ Hf	737	Cl ₃ Zr	596	Fa ₃ Si	997	CuNb ₂ O ₈ Zn ₂	2,127	
S ₃ Y ₂	696	I ₃ Ti	531	Br ₃ In	996	F ₇ RbSb ₂	2,126	
O ₂ P	674	Br ₂ Mo ₂ S	516	F ₃ Hf	995	B ₂ CuO ₆ Pb ₂	2,120	
HfI ₄	655	Br ₃ Sn	508	Cl ₈ CuGa ₂	969	F ₅ VZn	2,111	
HfI ₃	654	Br ₂ Mo	489	Cl ₂ Mo	950	Cu ₃ F ₈ Li ₂	2,098	



~20k structures with a band gap that is direct and suitable for water splitting.

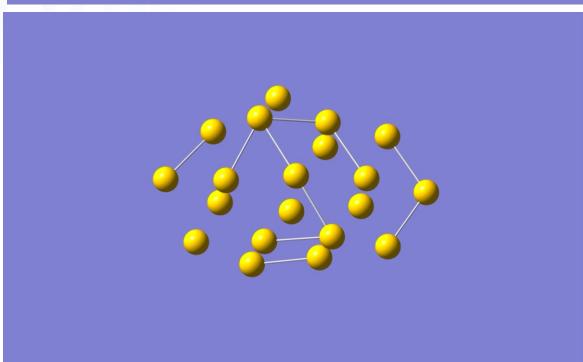
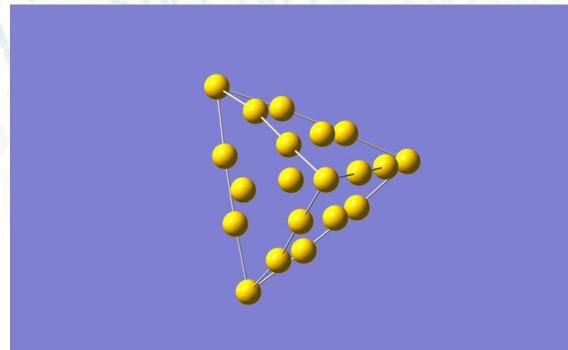
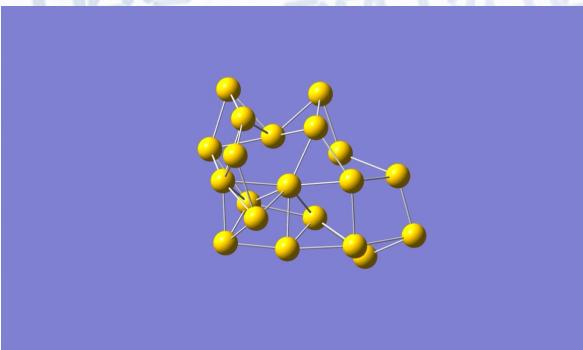
ML Interatomic potentials

Problems with classical MD

- Polynomial approximations will not work
- Bond breaking implementation requires explicit declaration of which atoms should interact via the stretching potential
- *Why then is the Morse potential not used?* The restoring force for large r is very low in case of the Morse potential, hence it takes longer for the bond length to return to the equilibrium position. The quadratic potential describes the potential well for displacements close to equilibrium and for moderate temperatures, this is the part of the potential you care about.

Au nano-particles

- Considered three Au isomers (large surface/bulk ratio)



ab-initio molecular dynamics

- VASP-MD for a total of 8000 steps at 2fs at T=800K

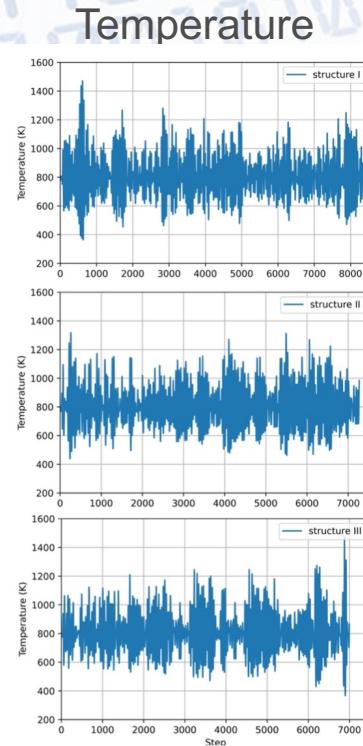
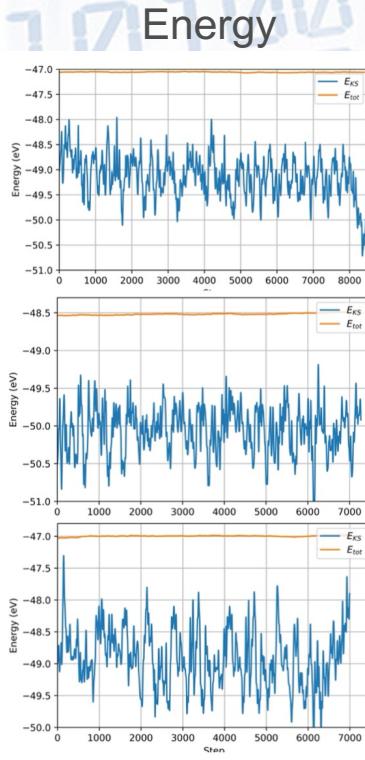


Table 1. Statistical analysis of thermal fluctuations in VASP MD simulations.

	Structure I	Structure II	Structure III
mean	799.50	799.73	799.42
std	143.91	146.47	144.87
min	362.59	438.78	367.98
25%	696.80	691.54	699.67
50%	792.57	790.09	793.40
75%	890.36	898.34	888.90
max	1470.42	1318.70	1471.67

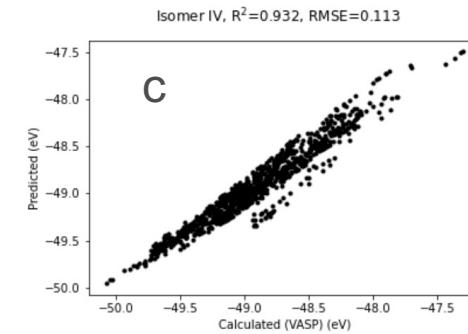
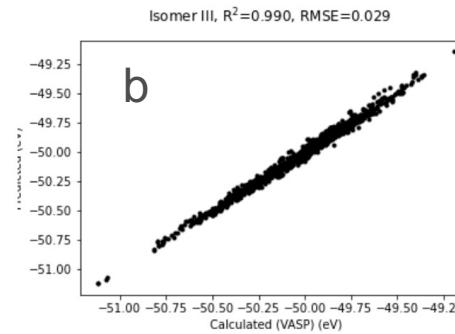
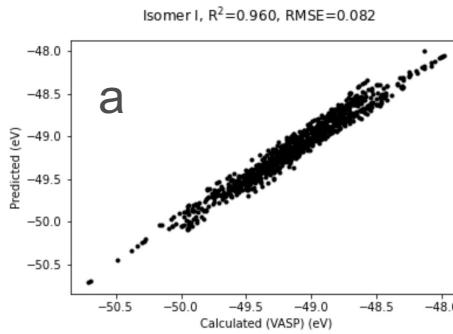
Table 2. Statistical analysis of energy fluctuations in VASP MD simulations.

	Structure I	Structure II	Structure III
mean	-49.13	-50.05	-48.91
std	0.40	0.30	0.44
min	-50.71	-51.10	-50.25
25%	-49.39	-50.35	-49.23
50%	-49.12	-50.98	-48.94
75%	48.85	-49.81	-48.63
max	-47.95	-49.25	-47.34

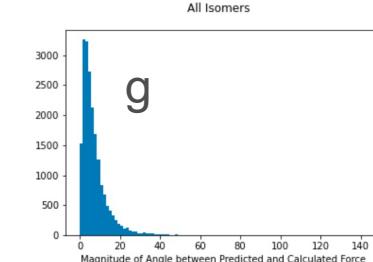
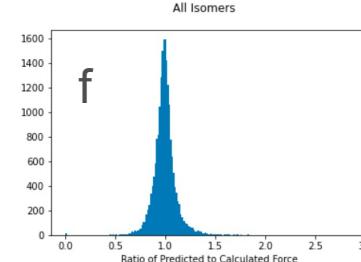
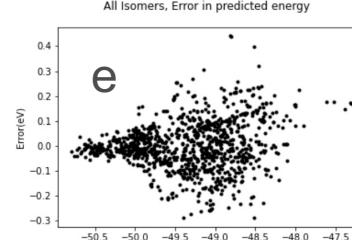
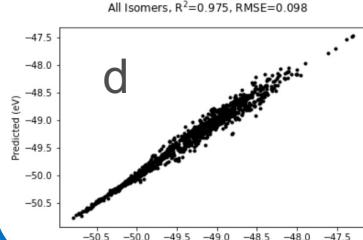
Machine Learning for inter-atomic potentials

```
#1000 points randomly chosen from the validation set
#DEEPMDF INFO      # number of test data : 1000
#DEEPMDF INFO      Energy RMSE          : 8.219273e-02 eV
#DEEPMDF INFO      Energy RMSE/Natoms   : 4.109637e-03 eV
#DEEPMDF INFO      Force RMSE           : 6.739765e-02 eV/A
#DEEPMDF INFO      Virial RMSE          : 3.285809e+00 eV
#DEEPMDF INFO      Virial RMSE/Natoms   : 1.642904e-01 eV
```

- R^2 and RMSE of MD using IP build with each isomer

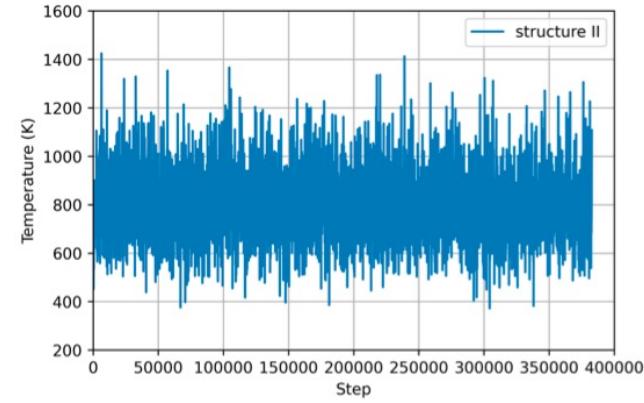
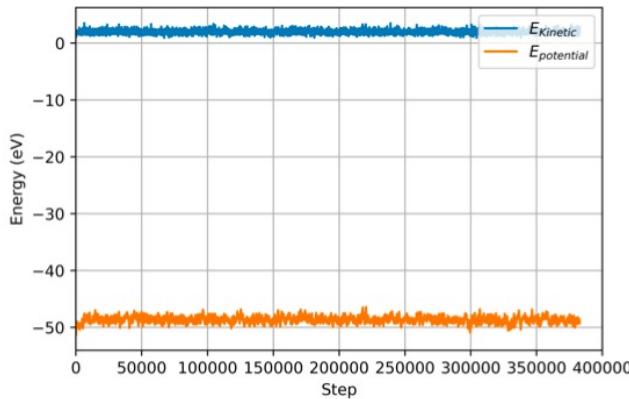


- R^2 and RMSE using IP build with all isomers



ML Simulations (LAMMPS)

Machine Learning molecular dynamics of structure-I



	Temperature (K)	Total Energy (eV)
mean	797.08	-46.70
std	152.98	0.67
min	371.95	-49.78
25%	688.94	-47.16
50%	786.95	-46.72
75%	894.24	-46.25
max	1424.30	-43.92

Simulation parameters are in good agreement with DFT structure-I

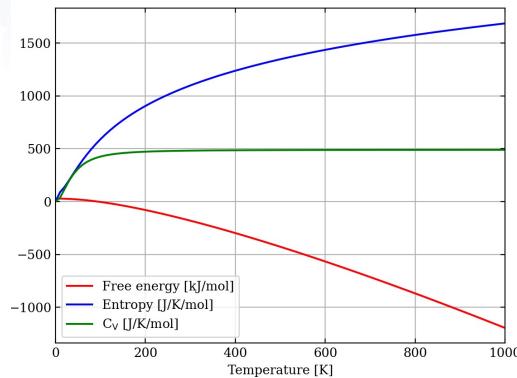
Specific heat from energy fluctuations vs frozen phonons

$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

Energy fluctuation model

Table 3. Specific heat capacity (C_v) at 800 K calculated using equation Eq.1 (values are given in J/K mol).

MD Energy-Fluctuations (VASP)	MD Energy-Fluctuations (SIESTA)	MD Energy-Fluctuations (LAMMPS)
35.12	21.50	20.63



Frozen phonons model

$$C_V = \sum_{\mathbf{q},j} C_{\mathbf{q},j} = \sum_{\mathbf{q},j} k_B \left(\frac{\hbar\omega_{\mathbf{q},j}}{k_B T} \right)^2 \frac{\exp(\hbar\omega_{\mathbf{q},j}/k_B T)}{[\exp(\hbar\omega_{\mathbf{q},j}/k_B T) - 1]^2},$$

$$F = \frac{1}{2} \sum_{\alpha i} \hbar\omega_{\mathbf{q},j} + k_B T \sum_{\alpha i} \ln[1 - \exp(-\hbar\omega_{\mathbf{q},j}/k_B T)],$$

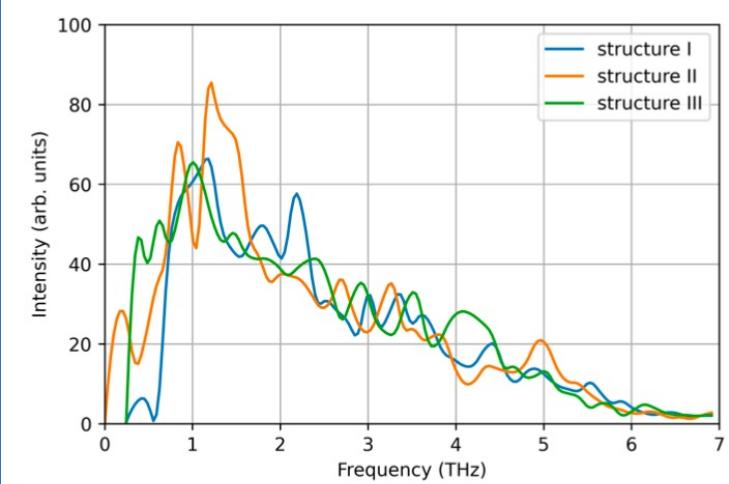
$$S = \frac{1}{2T} \sum_{\mathbf{q},j} \hbar\omega_{\mathbf{q},j} \coth[\hbar\omega_{\mathbf{q},j}/2k_B T] - k_B \sum_{\mathbf{q},j} \ln[2 \sinh(\hbar\omega_{\mathbf{q},j}/2k_B T)].$$

$$\mathbf{C_v=498J/kmol !!!}$$

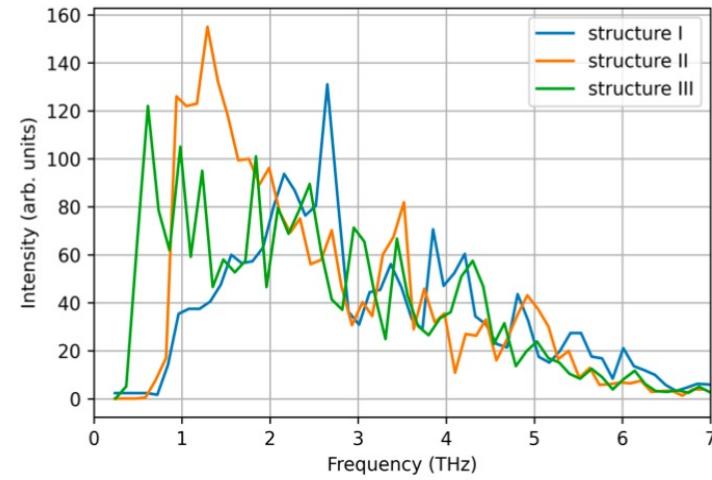
Energy fluctuation get anharmonic terms and gives a much better accurate description of the system (Experimental value 25 J/Kmol)

Vibrational DOS

DFT



Machine Learning

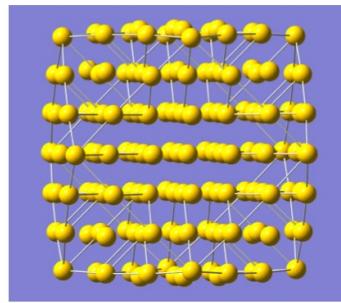


High Frequencies peak, typical of the large Au nanoparticle, is missing due to the small size of the nanoparticle in this simulations.

However, ML can accurately reproduce the VDOS calculated by DFT

Comparing Energies

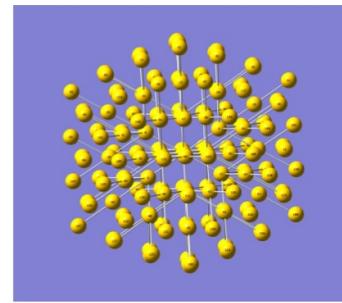
DFT



Cube

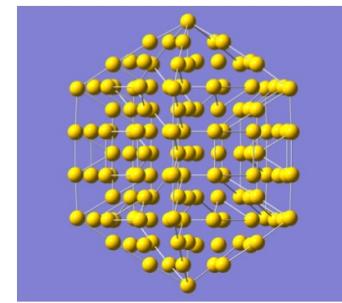
VASP E=-417.731 eV
LAMMPS E=-418.787 eV

Machine Learning



Icosahedron

VASP E=-420.674 eV
LAMMPS E=-419.309 eV



Decahedron

VASP E=-418.248 eV
LAMMPS E=-419.082 eV

Figure 7. Structures of Au₁₄₇ computed with VASP

Increasing the size of Au-NP

Specific Heat Calculations

Machine Learning

Specific Heat	
Timesteps	C_v (J/K/mol)
700	16.67
1300	17.79
2700	17.17
4000	16.98
5300	19.05
7000	18.95

Table 1. Specific heat of Au_{20} global minimum calculated by energy fluctuation of LAMMPS simulations. Five different ML-IPs that were generated using different VASP MD simulations timelength. The Timesteps column indicates the length of the *ab initio* simulations used to generate the ML-IP

Specific Heat	
ML-IP	Relative Error (%)
IP-Au20	19.65
IP-Au20+bulk	3.76
IP-Au147-amorph	2.34
IP-Au147-ico	0.00

Table 2. Specific heat relative errors calculated using four different ML-IPs in LAMMPS MD simulations. The errors are relative to the Au147-ico MLIP $C_v=20.01$ J/K/mol.



Featured Apps



JupyterLab

Start a JupyterLab instance



Virtual Desktop

Start a regular Virtual Desktop (VDI) instance



Virtual Desktop (GPU)

Start a GPU-enabled Virtual Desktop (VDI) instance



Gadi Terminal

Start a SSH connection to Gadi

All Apps

Search ...



JupyterLab

Start a JupyterLab instance



Virtual Desktop

Start a regular Virtual Desktop (VDI) instance



Virtual Desktop (GPU)

Start a GPU-enabled Virtual Desktop (VDI) instance



Gadi Terminal

Start a SSH connection to Gadi

Tutorial on Gadi

Login:

<https://are.nci.org.au/> ztswh19ca4

Home / My Interactive Sessions / JupyterLab

Interactive Apps

- Desktops
- VDI Desktop - GPU-enabled
- VDI Desktop
- Servers
- JupyterLab**
- RStudio

JupyterLab

Launch a JupyterLab session

Walltime (hours)

Number of hours your jupyter session can run (maximum). e.g. 1.5, 8, 24, 48

Queue

Compute Size

Amount of CPU/Memory resources available to your jupyter session

Project

Project to submit gadi job under; requires an SU allocation

Storage

- gdata/c03
- scratch/c03
- gdata/jm11
- gdata/g46
- gdata/x89
- scratch/dh4

Software

- abaqus
- abaqus_rmit
- adf
- ansys_monash
- ansys_mq
- ansys_nci
- ansys_rmit

I would like to receive an email when the session starts

[Advanced options ...](#)

A the end of the session, remember to delete it

Tutorial on Gadi

Login:

<https://are.nci.org.au/> ztswh19ca4

Advanced options ...

Extra arguments

Space-separated list of additional arguments to pass on the jupyterlab commandline

Module directories

/g/data/c03/modulefiles

Include module directories, eg /g/data/hr22/modulefiles (the equivalent of 'module use /g/data/hr22/modulefiles' on the command line). Make sure you add any **storage** option (above) required to access the directory (eg gdata/hr22 in this example)

Modules

python-SCCM

Includes modules eg python3/3.10.4 (the equivalent of 'module load python3/3.10.4' on the command line)

A the end of the session, remember to delete it

Tutorial on Gadi

Login:

<https://are.nci.org.au/> ztswh19ca4

The screenshot shows the Australian Research Environment (ARE) login interface. At the top, there's a blue header with a repeating binary code pattern. Below it, the ARE logo (a stylized orange and blue circular pattern) and the NCI Australia logo are displayed. The main area is titled "Featured Apps" and contains four cards:

- JupyterLab** (circled in blue): Start a JupyterLab instance.
- Virtual Desktop**: Start a regular Virtual Desktop (VDI) instance.
- Virtual Desktop (GPU)**: Start a GPU-enabled Virtual Desktop (VDI) instance.
- Gadi Terminal**: Start a SSH connection to Gadi.

Below this, there's a section titled "All Apps" with a search bar. It lists the same four options again:

- JupyterLab**: Start a JupyterLab instance.
- Virtual Desktop**: Start a regular Virtual Desktop (VDI) instance.
- Virtual Desktop (GPU)**: Start a GPU-enabled Virtual Desktop (VDI) instance.
- Gadi Terminal**: Start a SSH connection to Gadi.

At the end of the session, remember to delete it