Seyed Mohamad Moosavi & Kevin Maik Jablonka *MolSim - 2020-Amsterdam*

Basics of machine learning for chemistry and materials science

Why ML in chemistry and materials science?

- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

Slide from Prof. Smit's lecture on Jan 6

We assume the interactions between the particles are known!

Exact= in the limit of infinitely long simulations the error bars can be made infinitely small

The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*

If one could envision an experimental system of these N particles that interact with the potential.

Pressure
Heat capacity
Heat of adsorption
Structure
….

Diffusion coefficient
Viscosity

Why ML in chemistry and materials science?

- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

Slide from Prof. Smit's lecture on Jan 6

The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*

We assume the interactions between the particles are known!

Exact= in the limit of infinitely long simulations the error bars can be made infinitely small

The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*

If one could envision an experimental system of these N particles that interact with the potential.

Pressure
Heat capacity
Heat of adsorption
Structure
....

Diffusion coefficient
Viscosity

Why ML in chemistry and materials science?

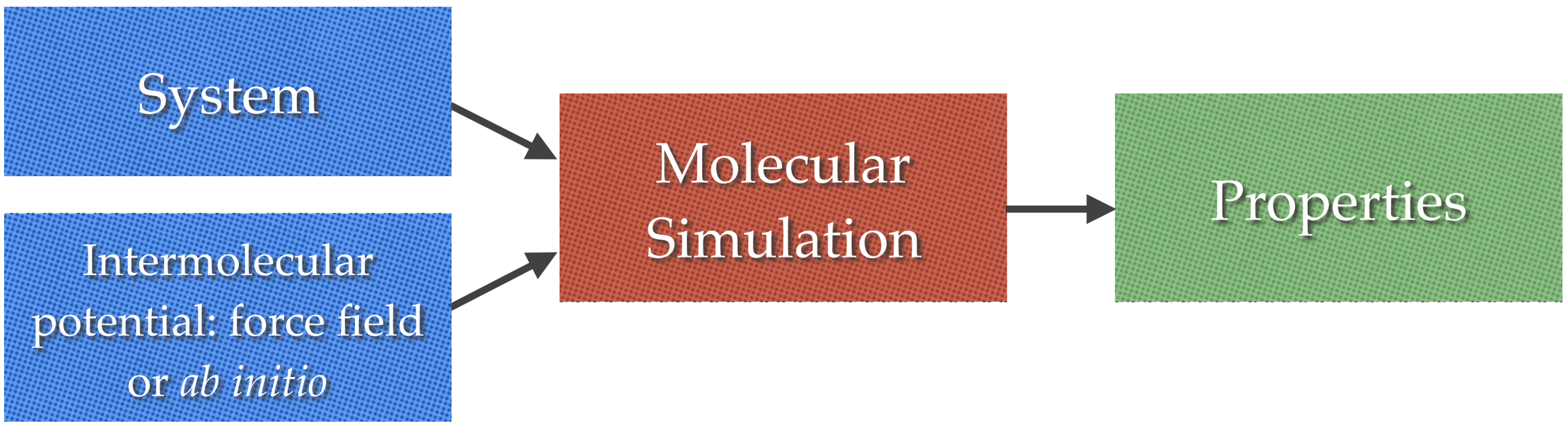
- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*

Why ML in chemistry and materials science?

- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

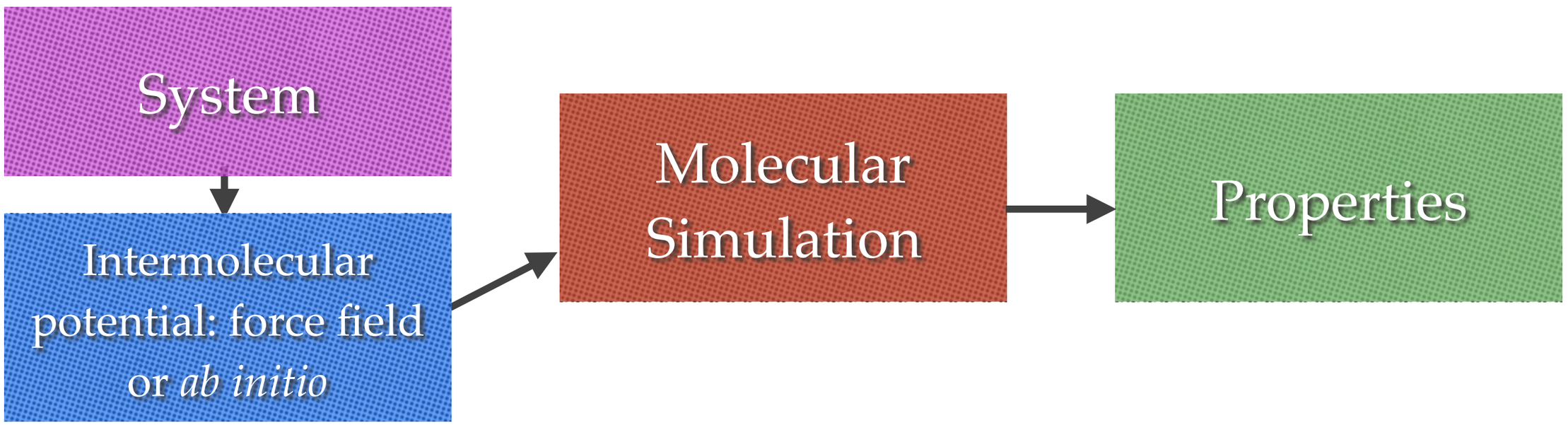
The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*



Why ML in chemistry and materials science?

- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

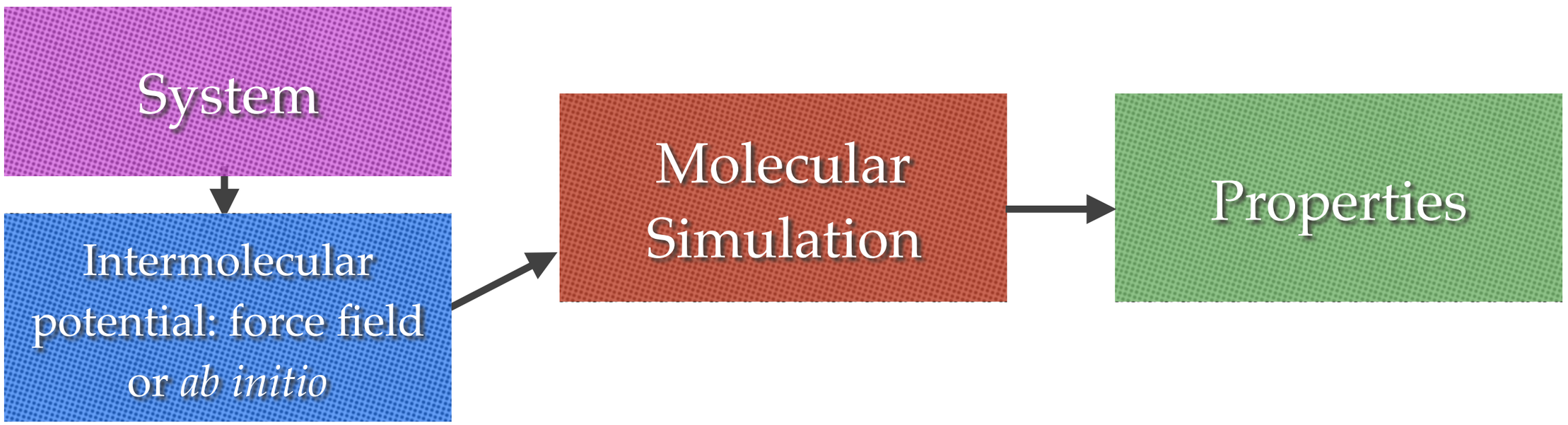
The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*



Why ML in chemistry and materials science?

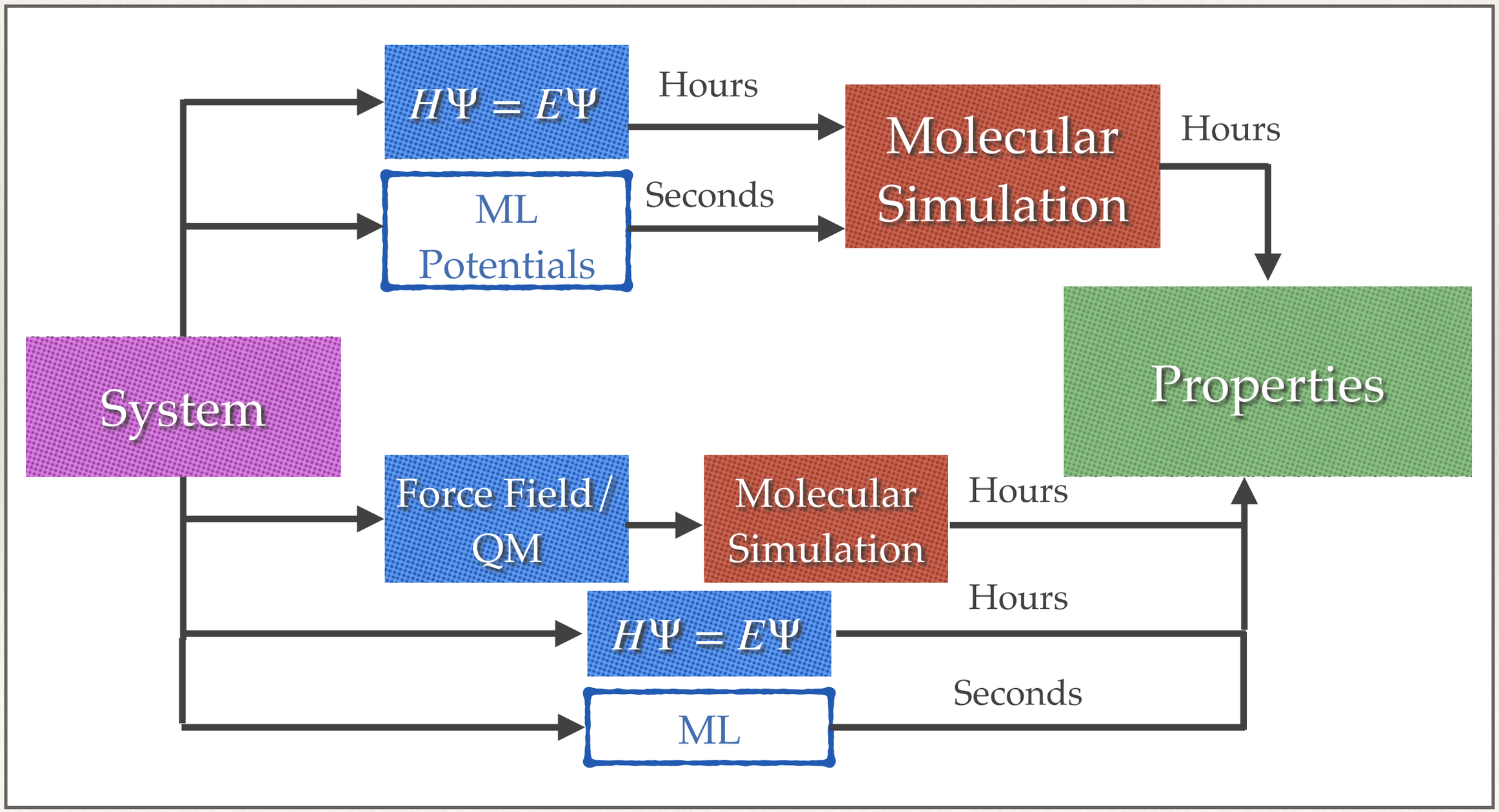
- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help

The idea for a given *intermolecular potential* "*exactly*" compute the *thermodynamic* and *transport* properties of the *system*



Why ML in chemistry and materials science?

- If we **agree** molecular simulation is useful:
  - Then: let's see where ML can help



Why ML in chemistry and materials science?

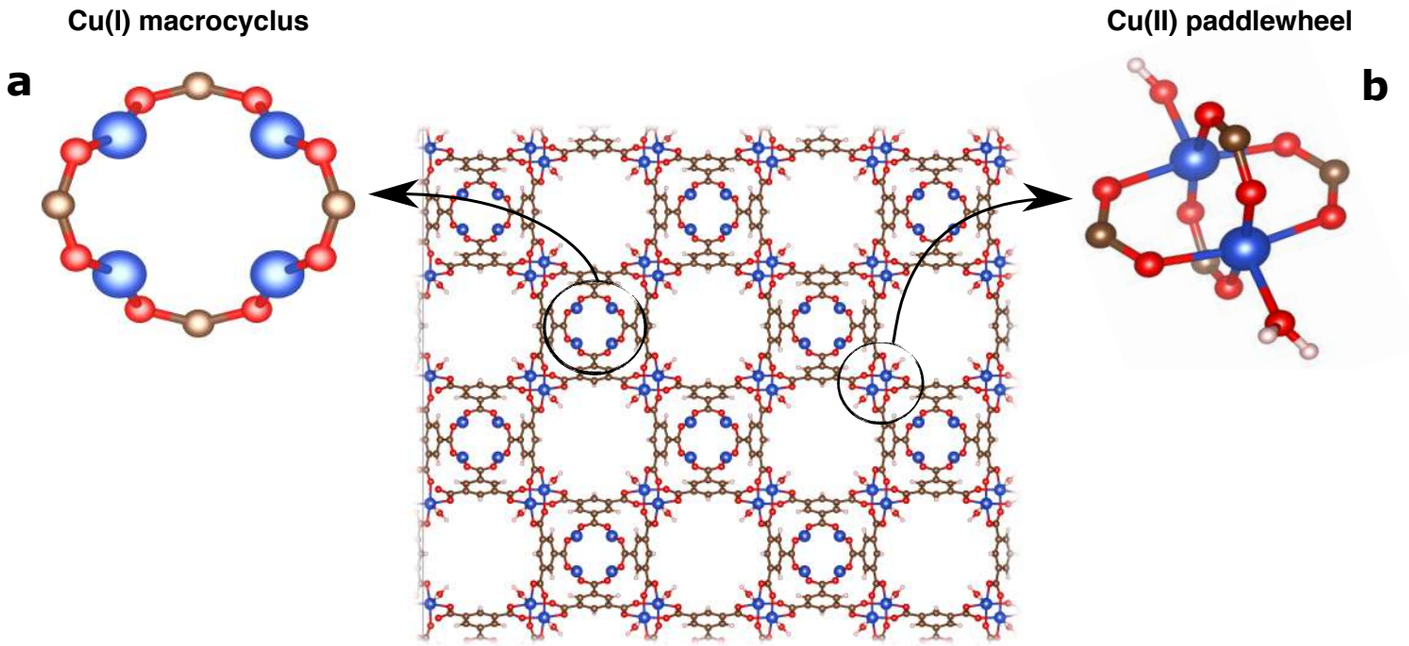
- ML enables us to do new things too!

  - We have access to enormous amount of data

Why ML in chemistry and materials science?

- ML enables us to do new things too!

Oxidation states —> metal centres of MOFs



Current approaches

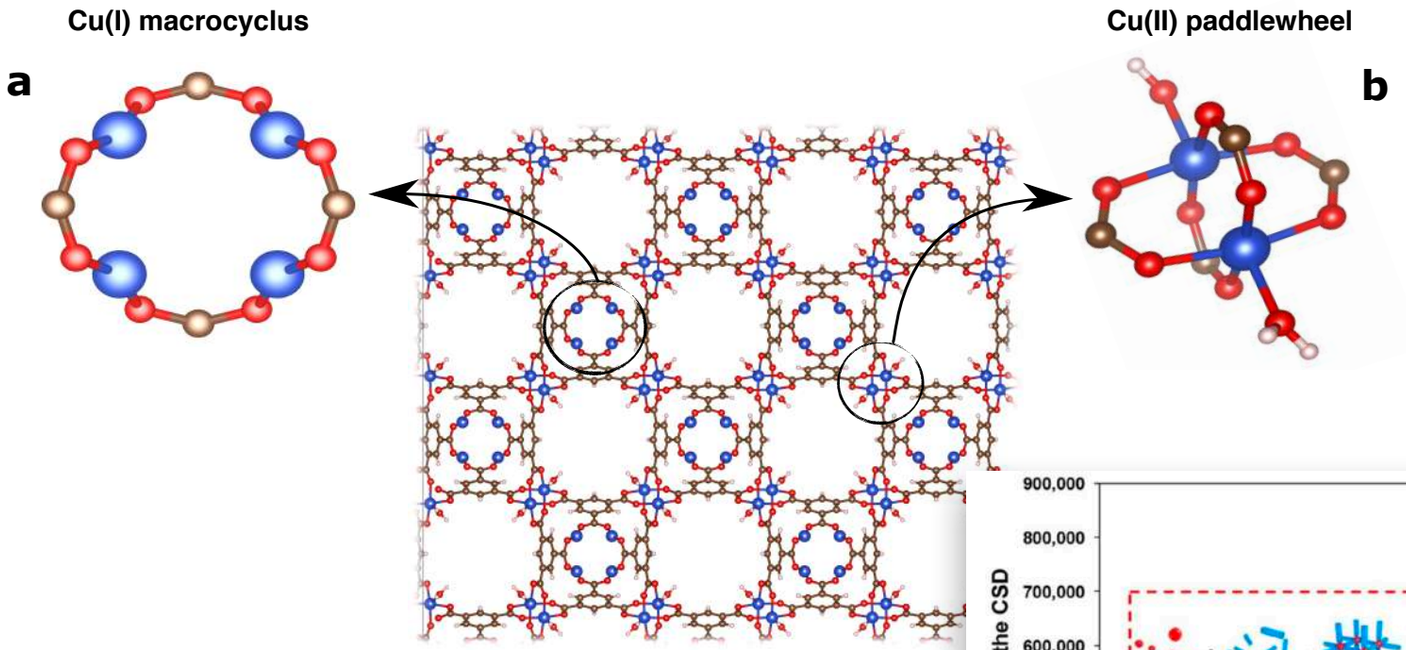
- Empirical assignment
- Theoretical models —> bond valence approach
- Quantum calculations/spectroscopy

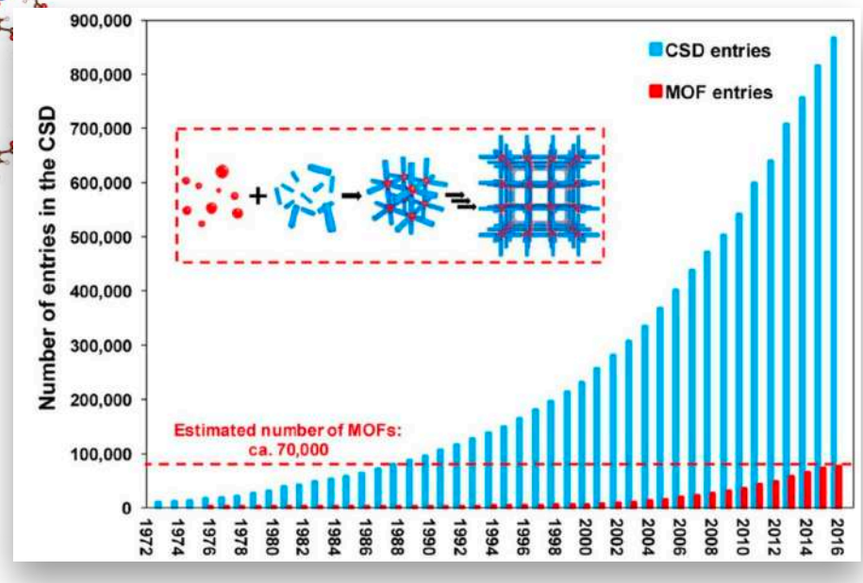
Jablonka, et al. submitted (2020)

Why ML in chemistry and materials science?

- ML enables us to do new things too!

Oxidation states —> metal centres of MOFs





Current approaches

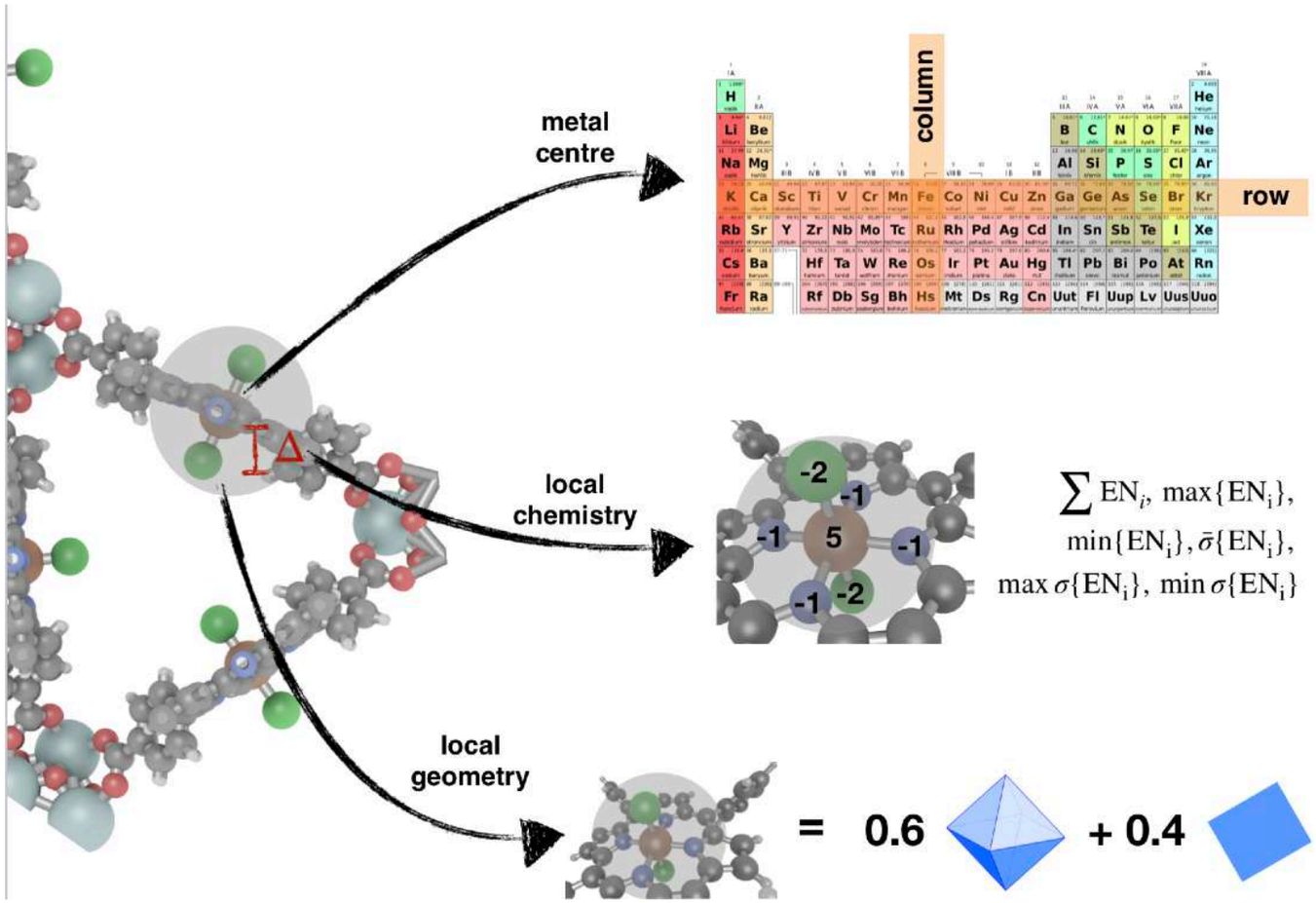
- Empirical assignment
- Theoretical models —> bond valence approach
- Quantum calculations/spectroscopy

Moghadam et al. Chemistry of Materials (2018)
Jablonka, et al. submitted (2020)

Why ML in chemistry and materials science?

- ML enables us to do new things too!

Oxidation states —> metal centres of MOFs



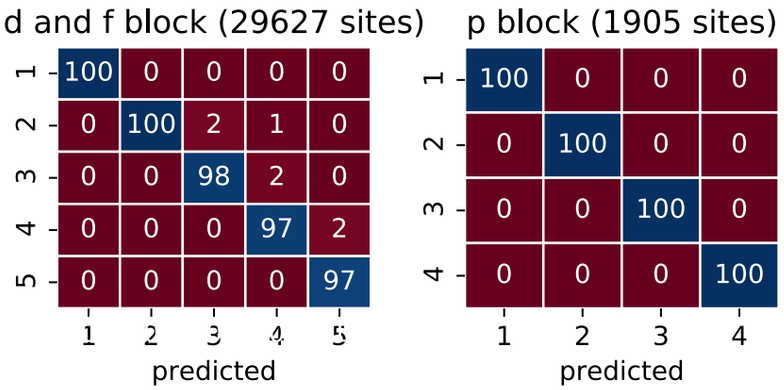
Jablonka, et al. submitted (2020)

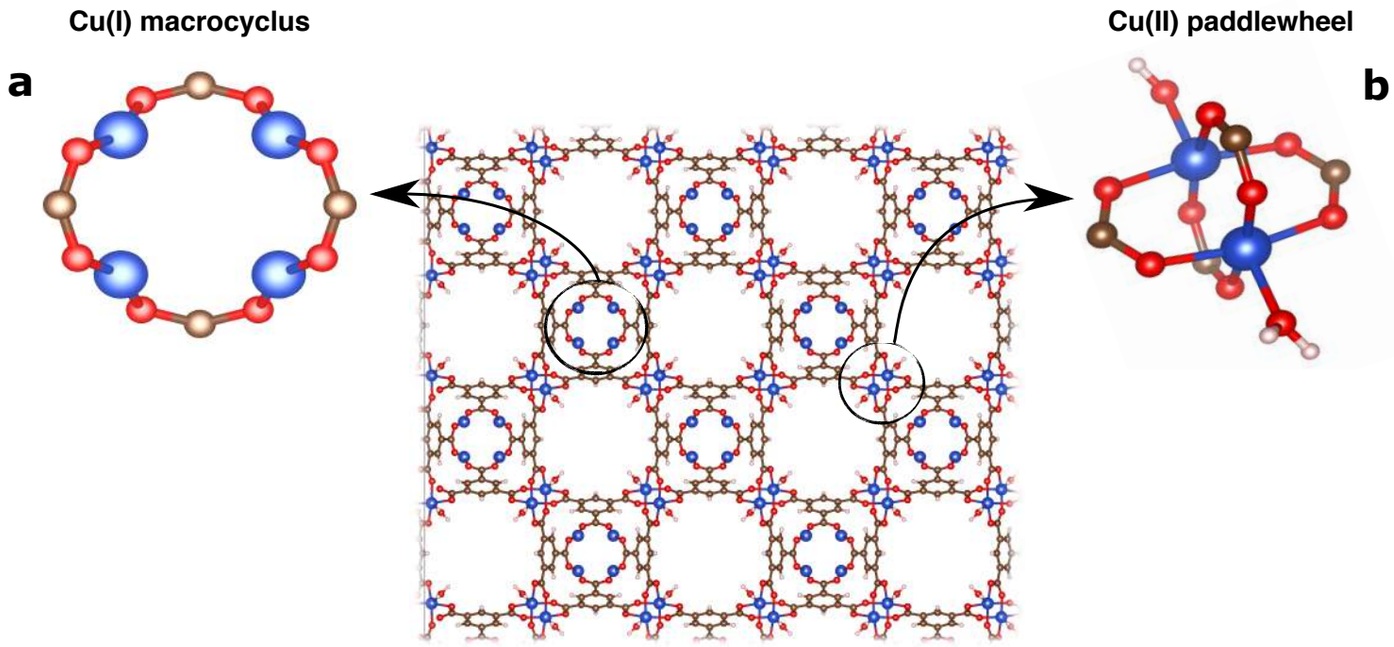
Why ML in chemistry and materials science?

- ML enables us to do new things too!

Current approaches

- Empirical assignment
- Theoretical models
  —> bond valence approach
- QM/spectroscopy





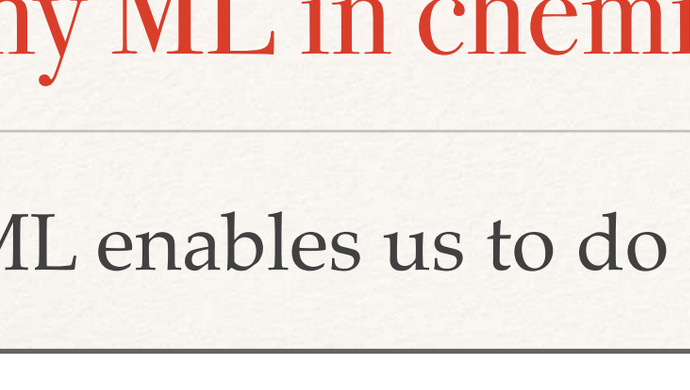
Jablonka, et al. submitted (2020)

Why ML in chemistry and materials science?

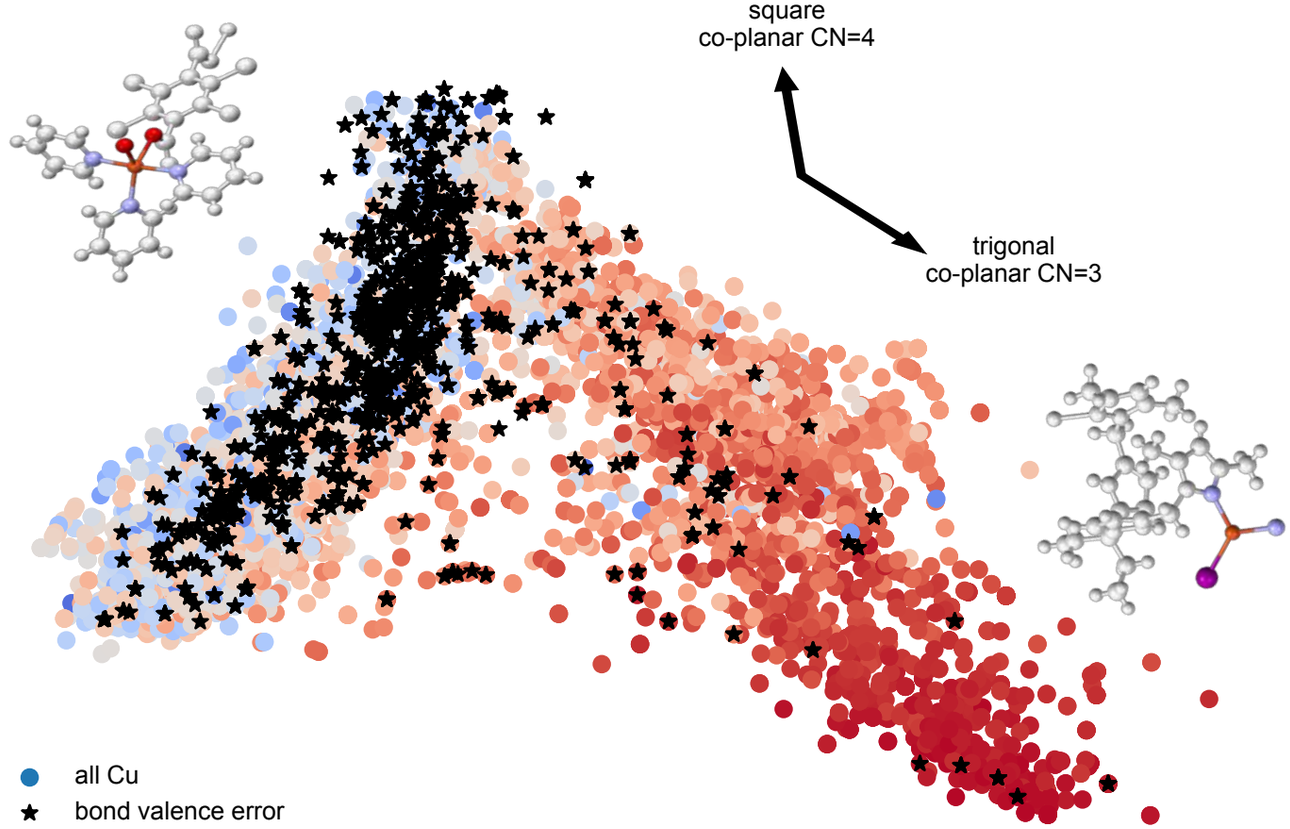
- ML enables us to do new things too!

Current approaches

- Empirical assignment
- Theoretical models
  —> bond valence approach
- QM/spectroscopy



oxiMACHINE

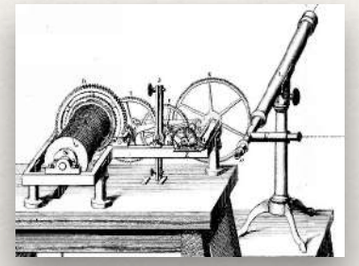


Jablonka, et al. submitted (2020)

The “fourth paradigm” of science

1600 1950 2000

Empiral Science (1st Paradigm)



Experimental Observation

Theoretical Science (2nd Paradigm)

$$F = ma$$

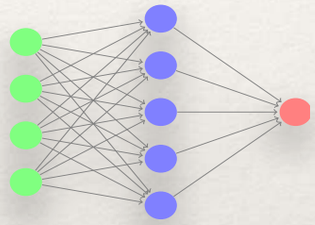
$$\hat{\mathcal{H}} |\psi\rangle = E |\psi\rangle$$

Development of Theories and Generalizations

Computational Science (3rd Paradigm)

Simulation of complex phenomena, High troughput computational screenings

Data Driven Science (4th Paradigm)



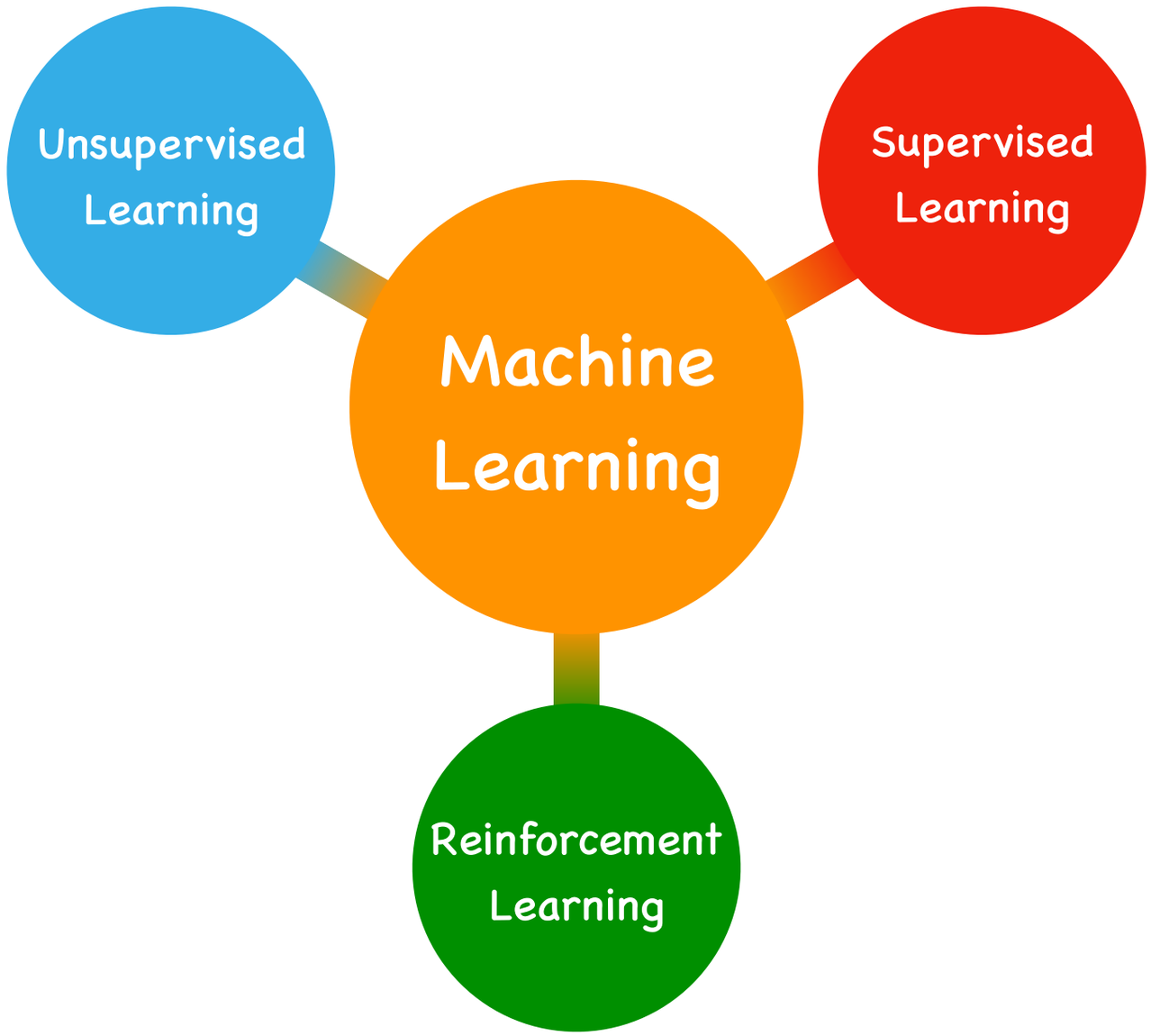
Machine Learning

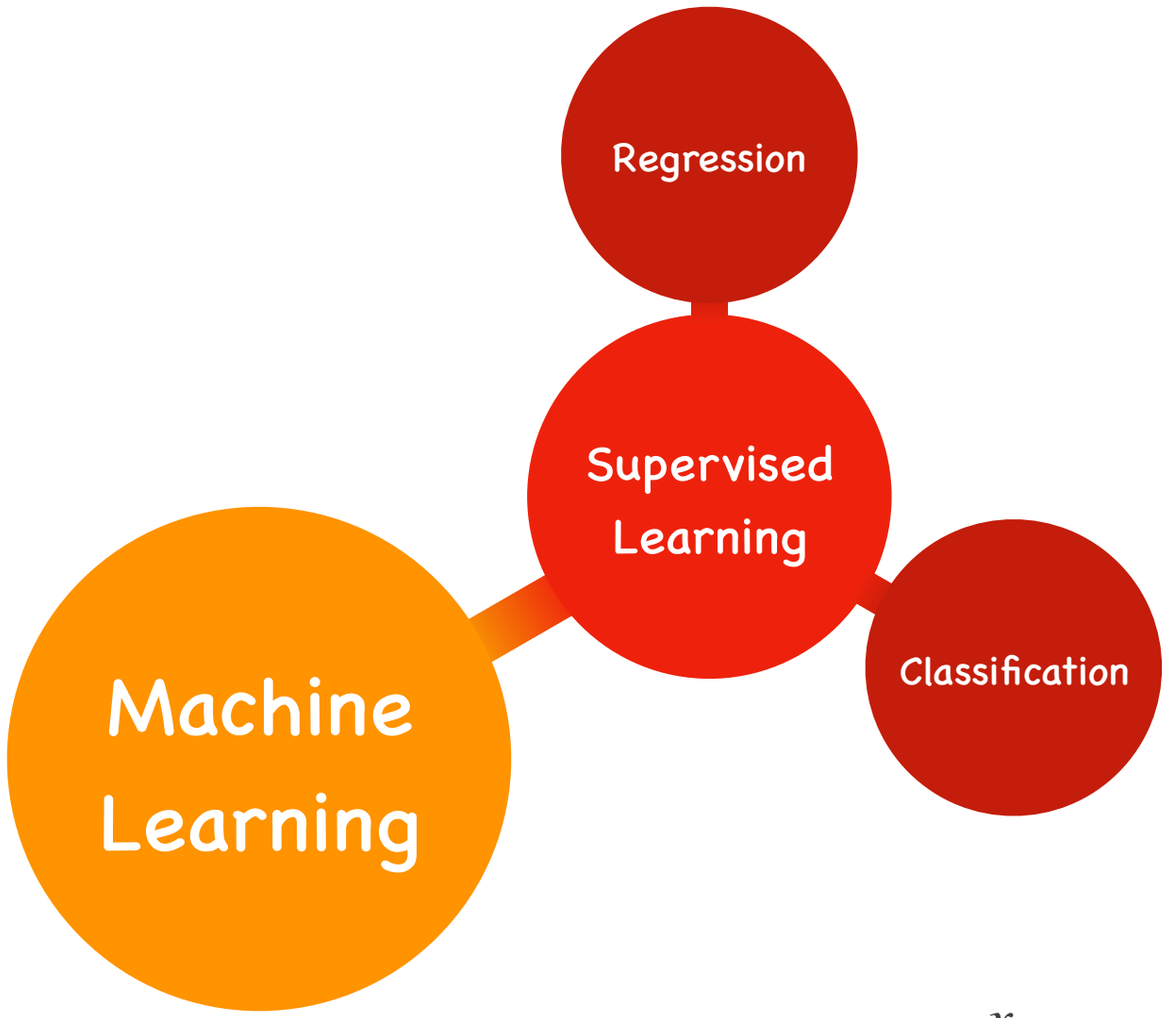
“It is not that machines are going to replace chemists. It’s that the chemists who use machines will replace those that do not.”

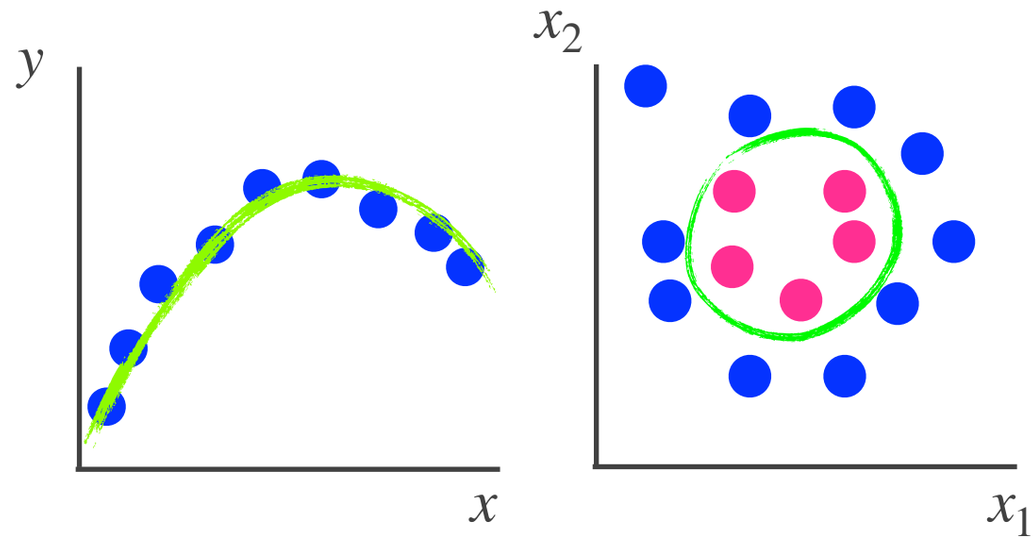
- Derek Lowe, In the pipeline, Science Mag.

“AI is good at automatic tasks, rather than jobs.”

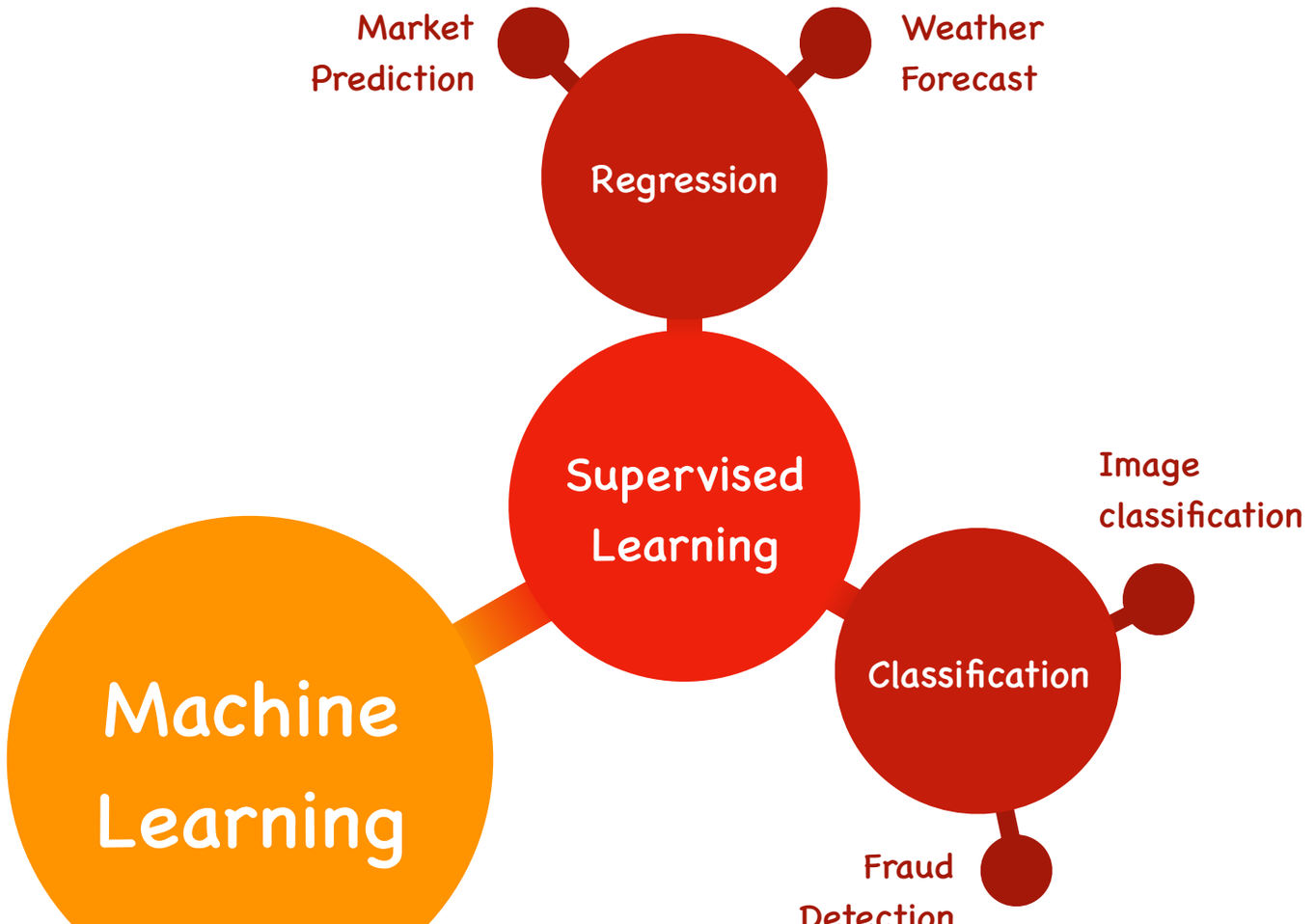
- Andrew Ng, Google Brain and Stanford

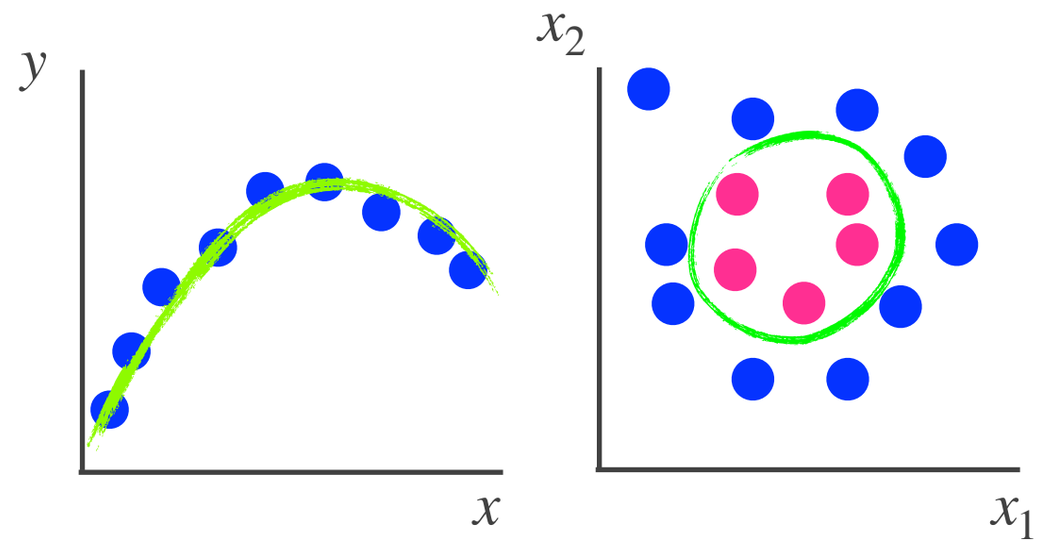

Unsupervised Learning
Supervised Learning
Machine Learning
Reinforcement Learning



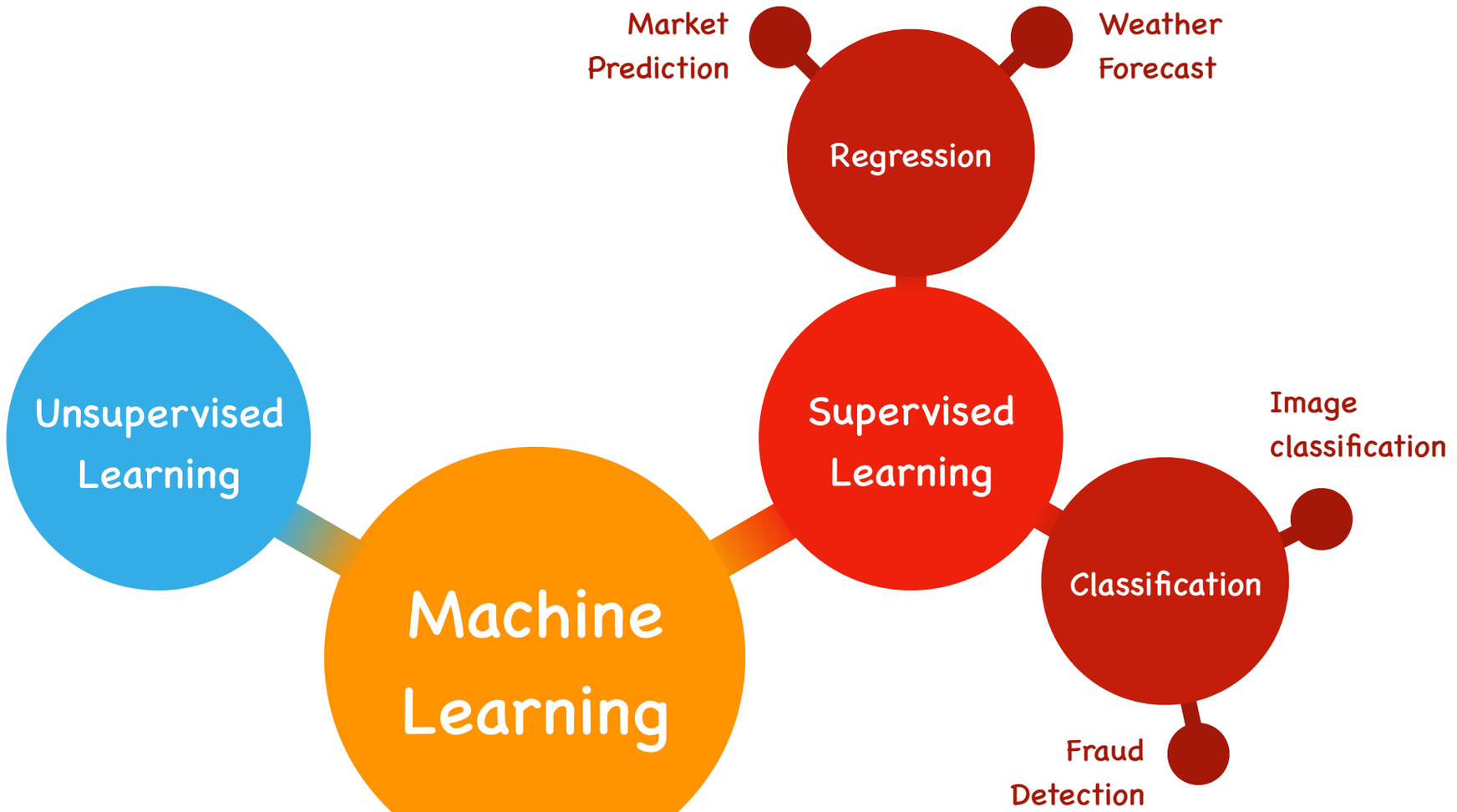


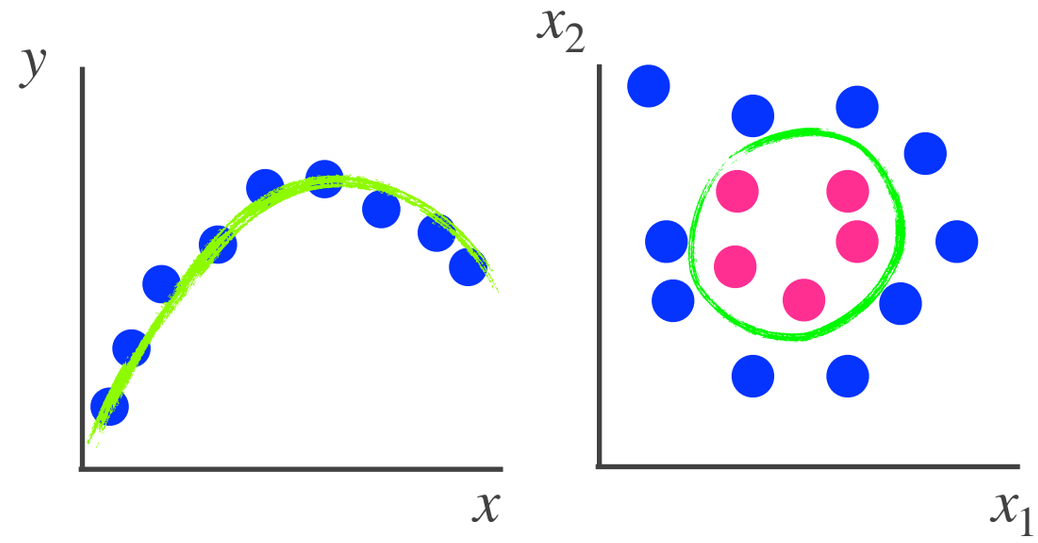
Labeled data



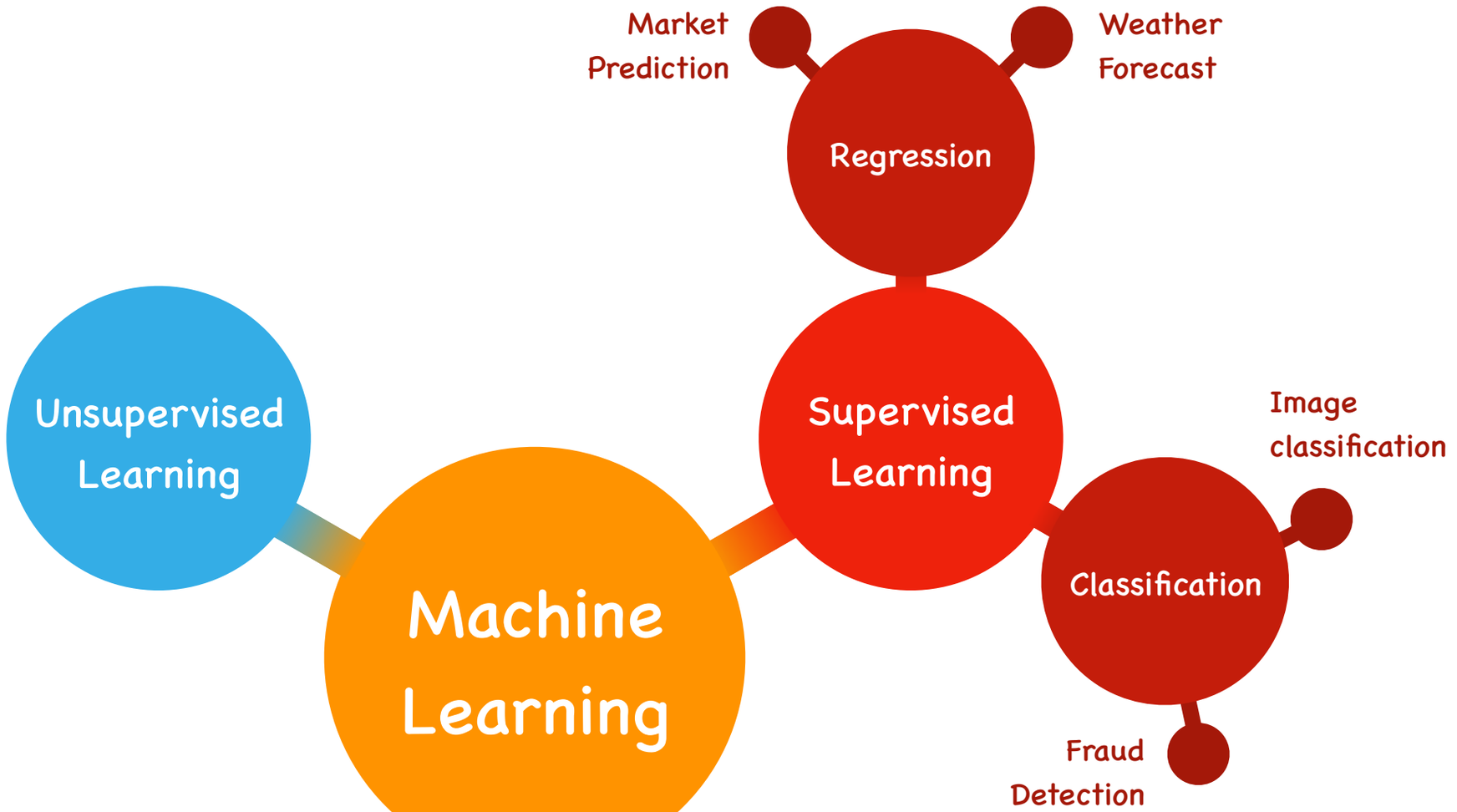


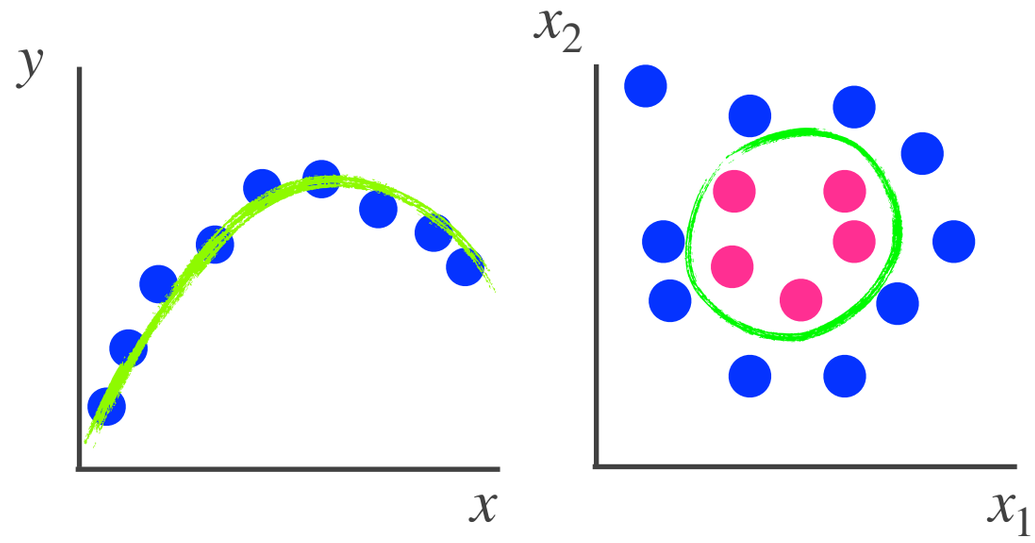
Labeled data



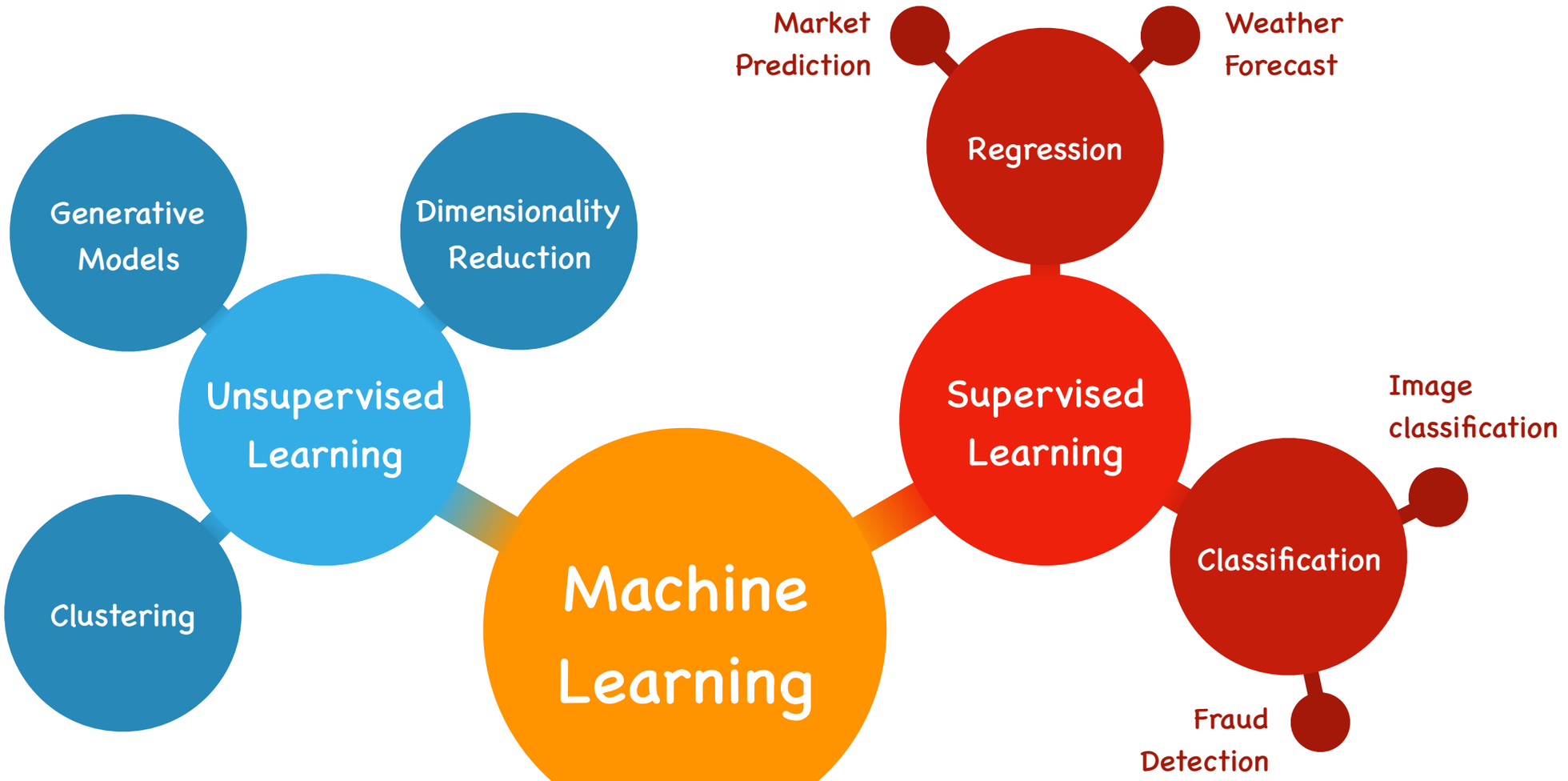


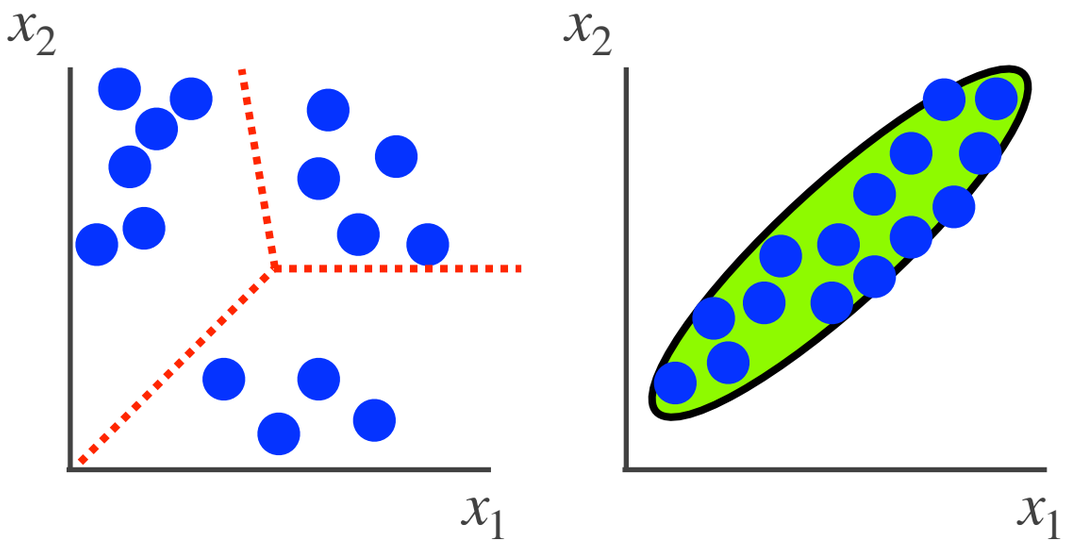
Labeled data



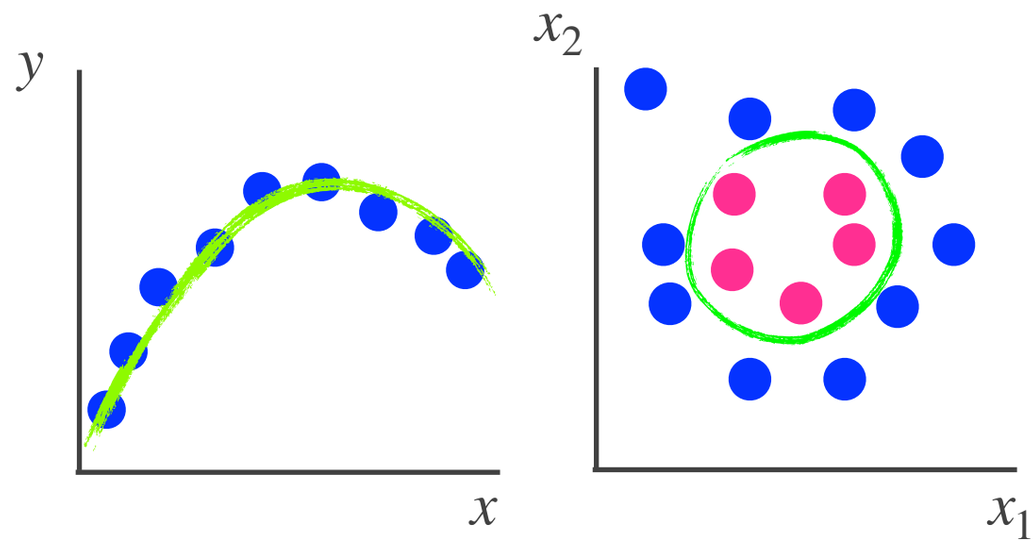


Labeled data





Unlabelled data



Labeled data

moj

Generative Models

Data Visualisation

Dimensionality Reduction

Feature Engineering

Diversity Analysis

Clustering

Outlier detection

Unsupervised Learning

Machine Learning

Market Prediction

Weather Forecast

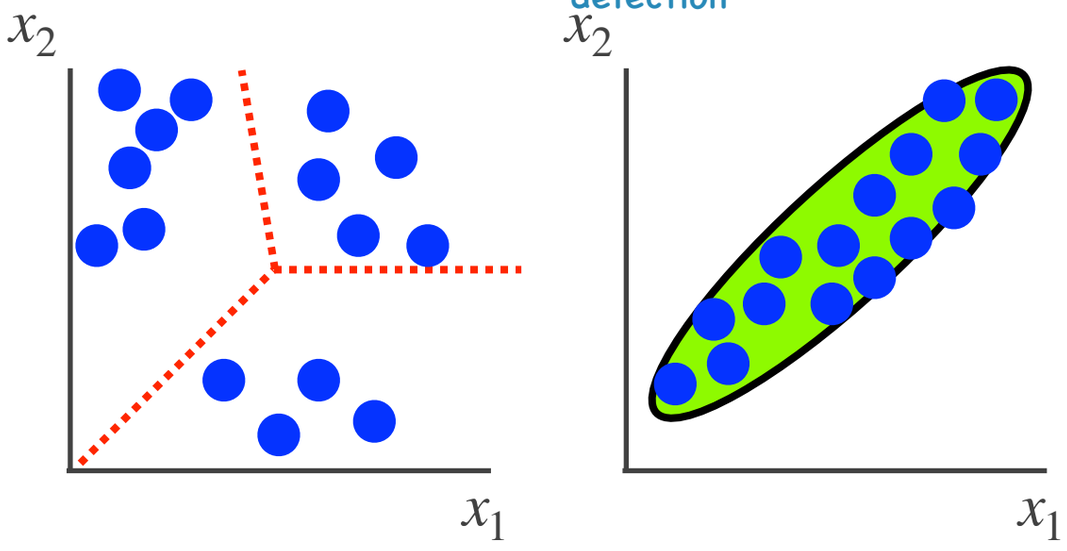
Regression

Supervised Learning

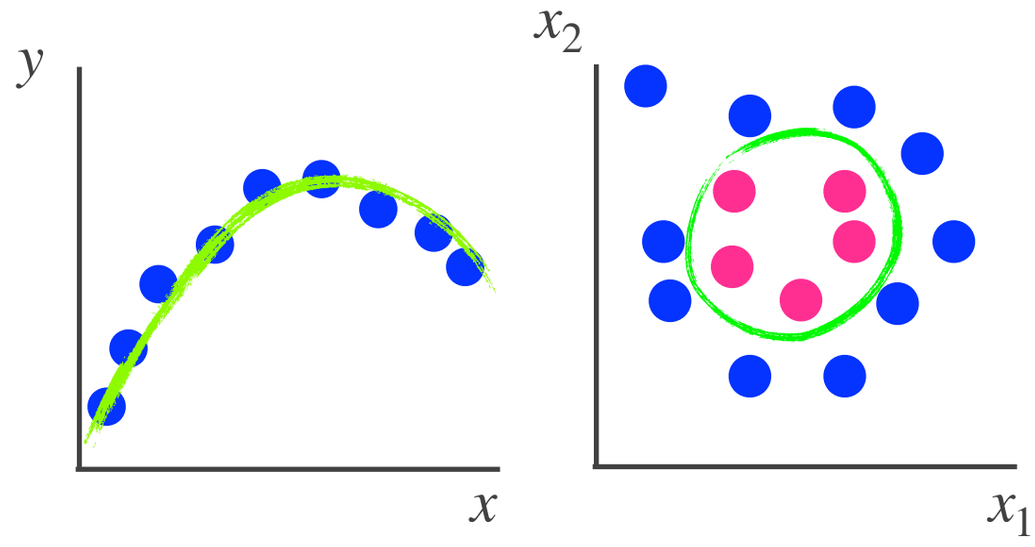
Image classification

Classification

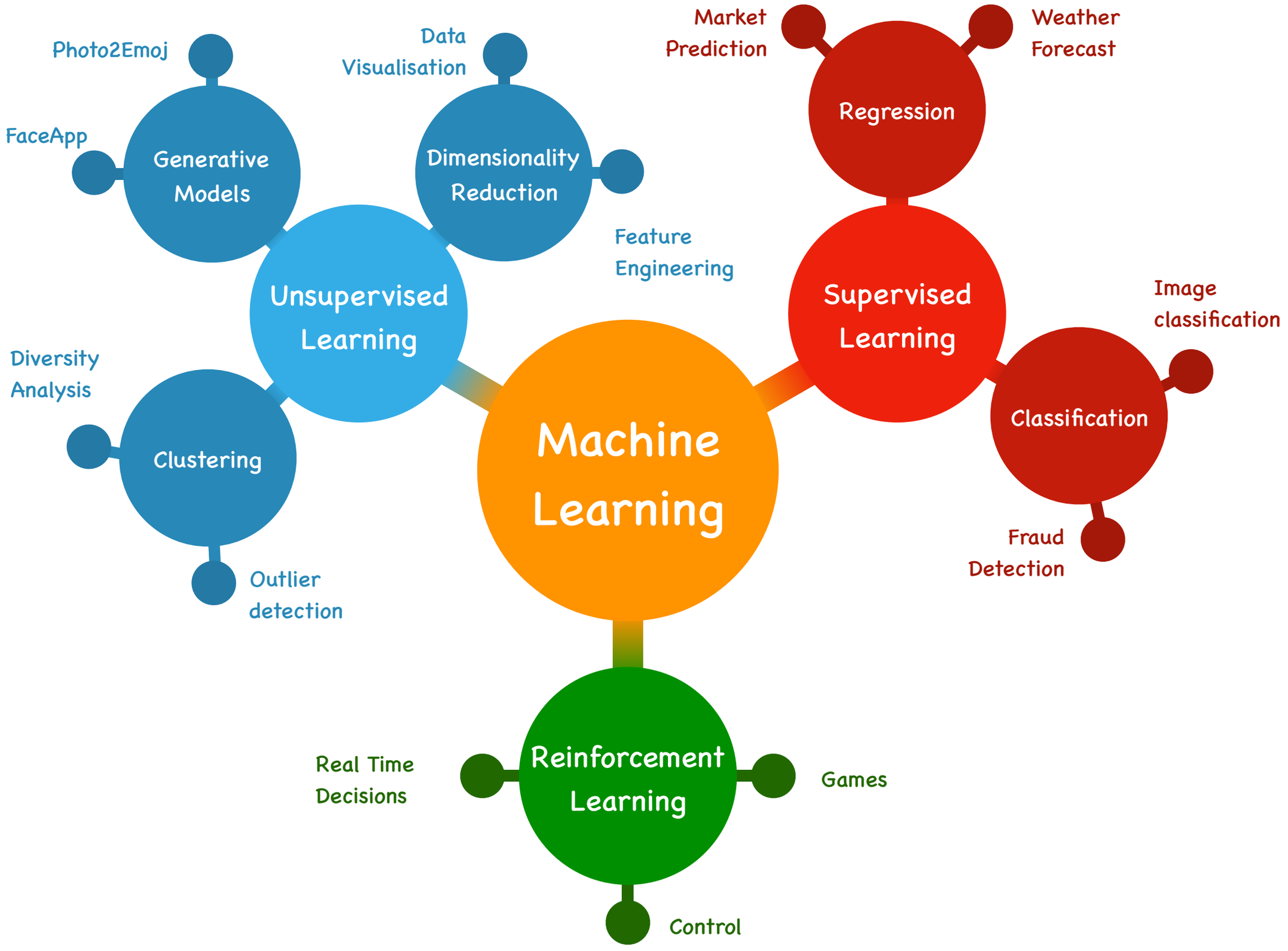
Fraud Detection



Unlabelled data

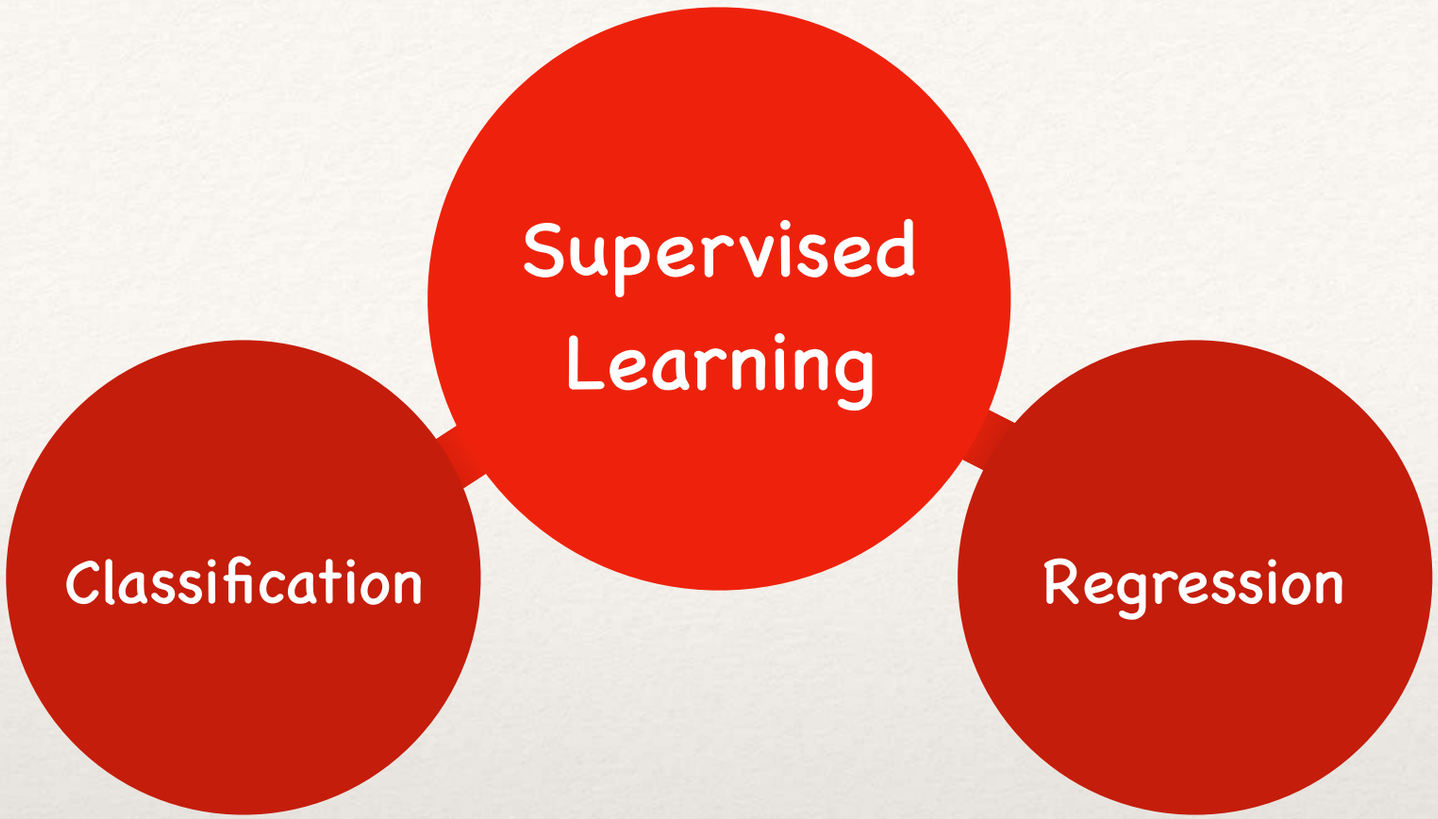


Labeled data


Machine Learning
Unsupervised Learning
Generative Models
Photo2Emoj
FaceApp
Dimensionality Reduction
Data Visualisation
Feature Engineering
Clustering
Diversity Analysis
Outlier detection
Supervised Learning
Regression
Market Prediction
Weather Forecast
Classification
Image classification
Fraud Detection
Reinforcement Learning
Real Time Decisions
Games
Control

Lecture structure

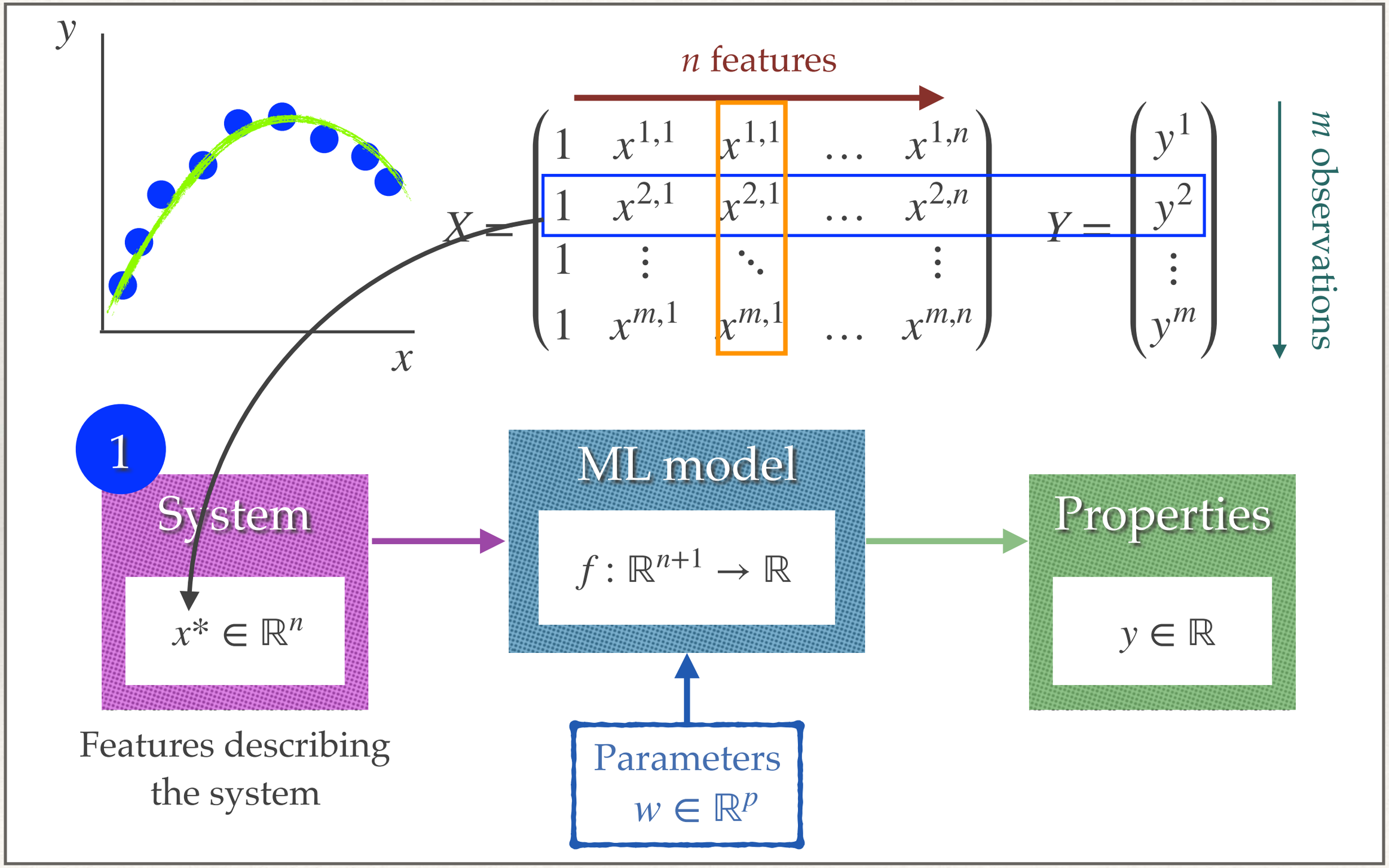
- Supervised machine learning workflow
- Model interpretation
- Feature selection
- Dimensionality reduction
- Applications



Supervised learning

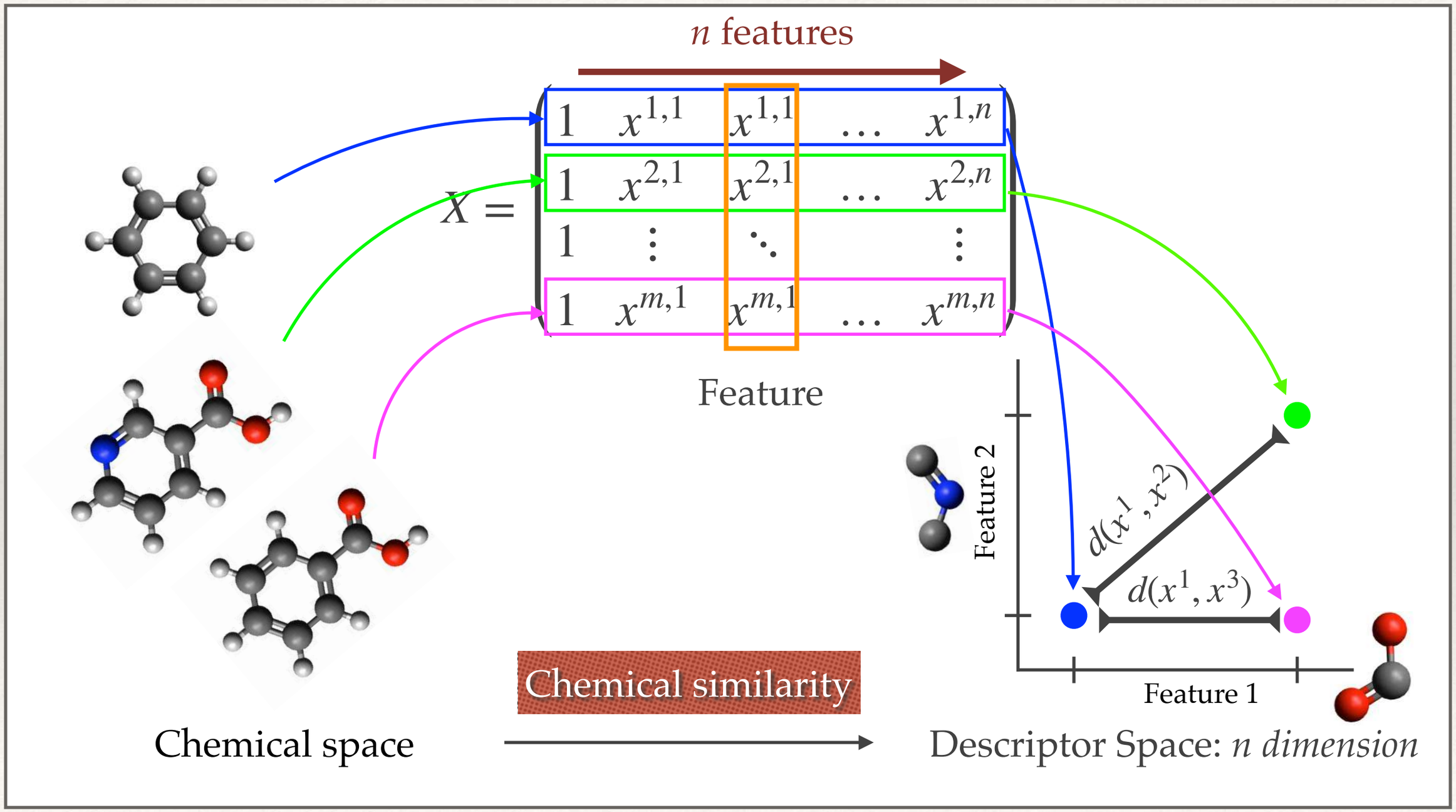
- Featurisation
- Model training
- Hyperparameter optimisation
- Linear and kernel models

Supervised machine learning



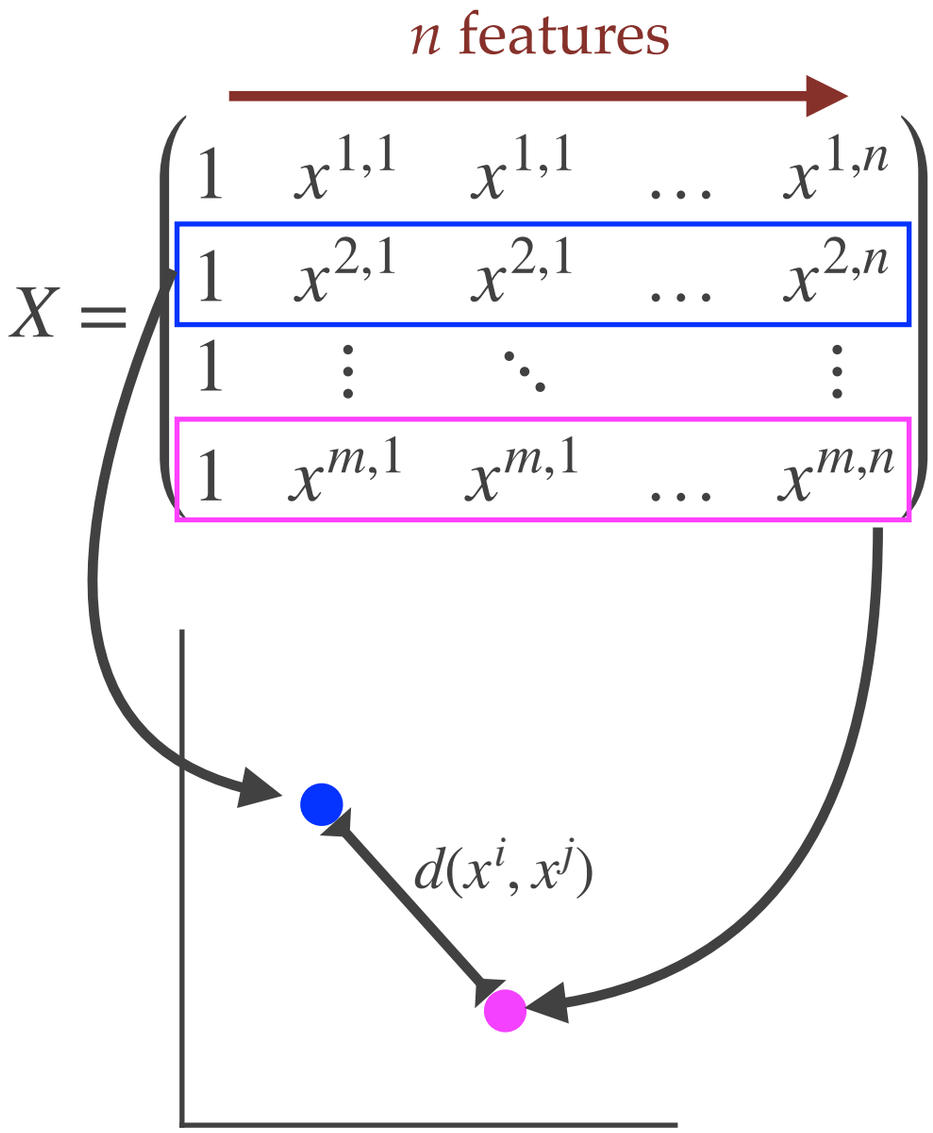
What is featurisation/a descriptor?

Encoding chemistry into numbers: "chemical space" to descriptor space



What makes a descriptor good?

Encoding chemistry into numbers



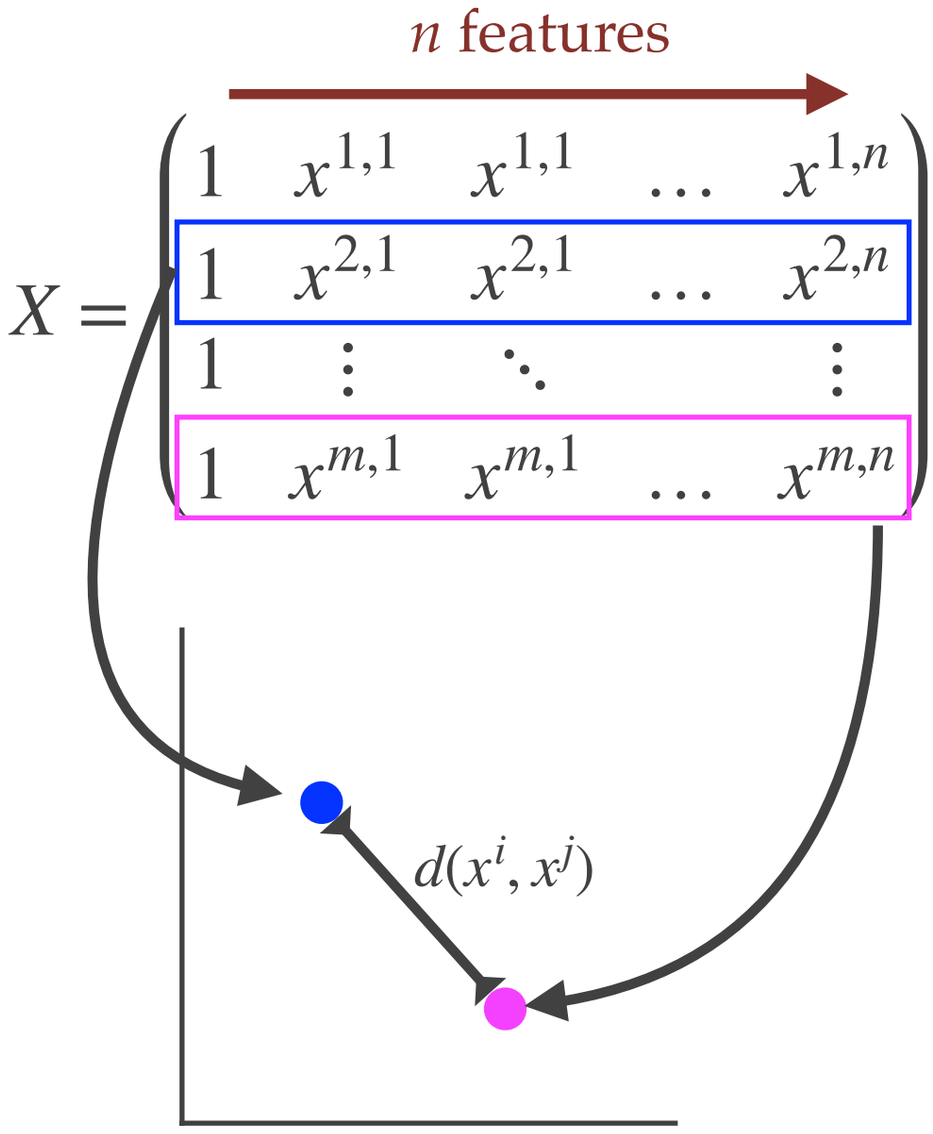
Descriptor Space: *n dimension*

Good descriptors —> obey physics

- Invariant w.r.t. symmetries
- As low dimension as possible
- Cheap to compute
- Non-degenerate
- Transferability across elements

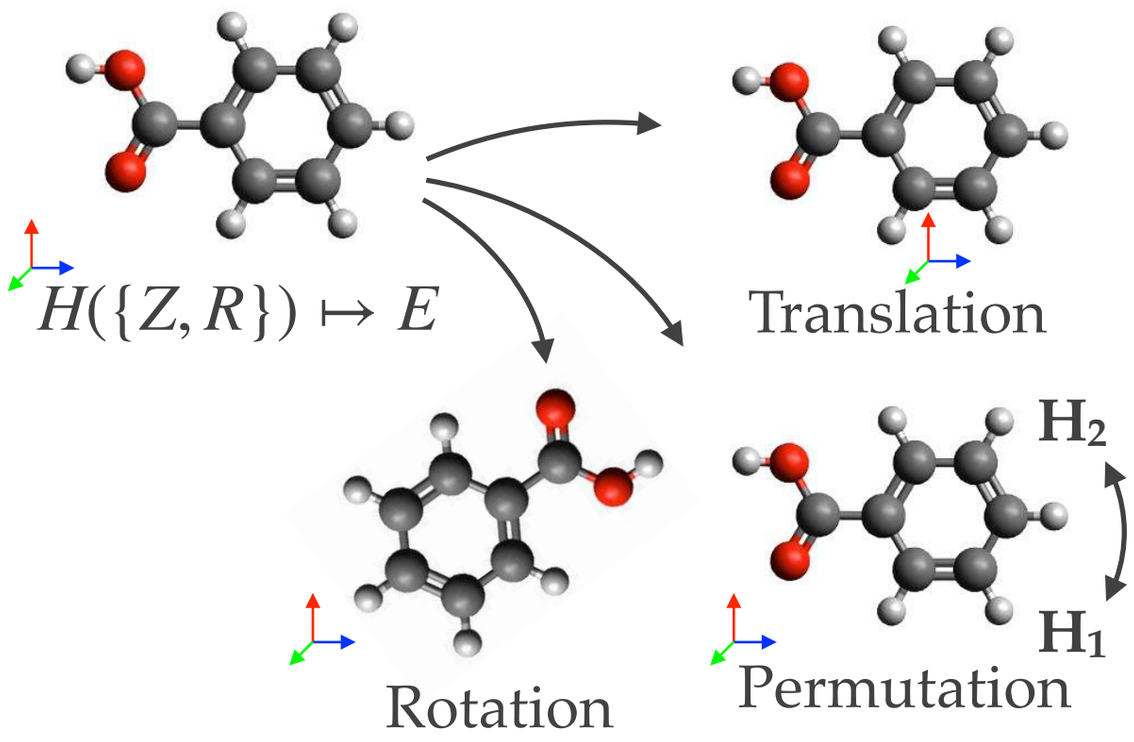
What makes a descriptor good?

Encoding chemistry into numbers



Good descriptors:

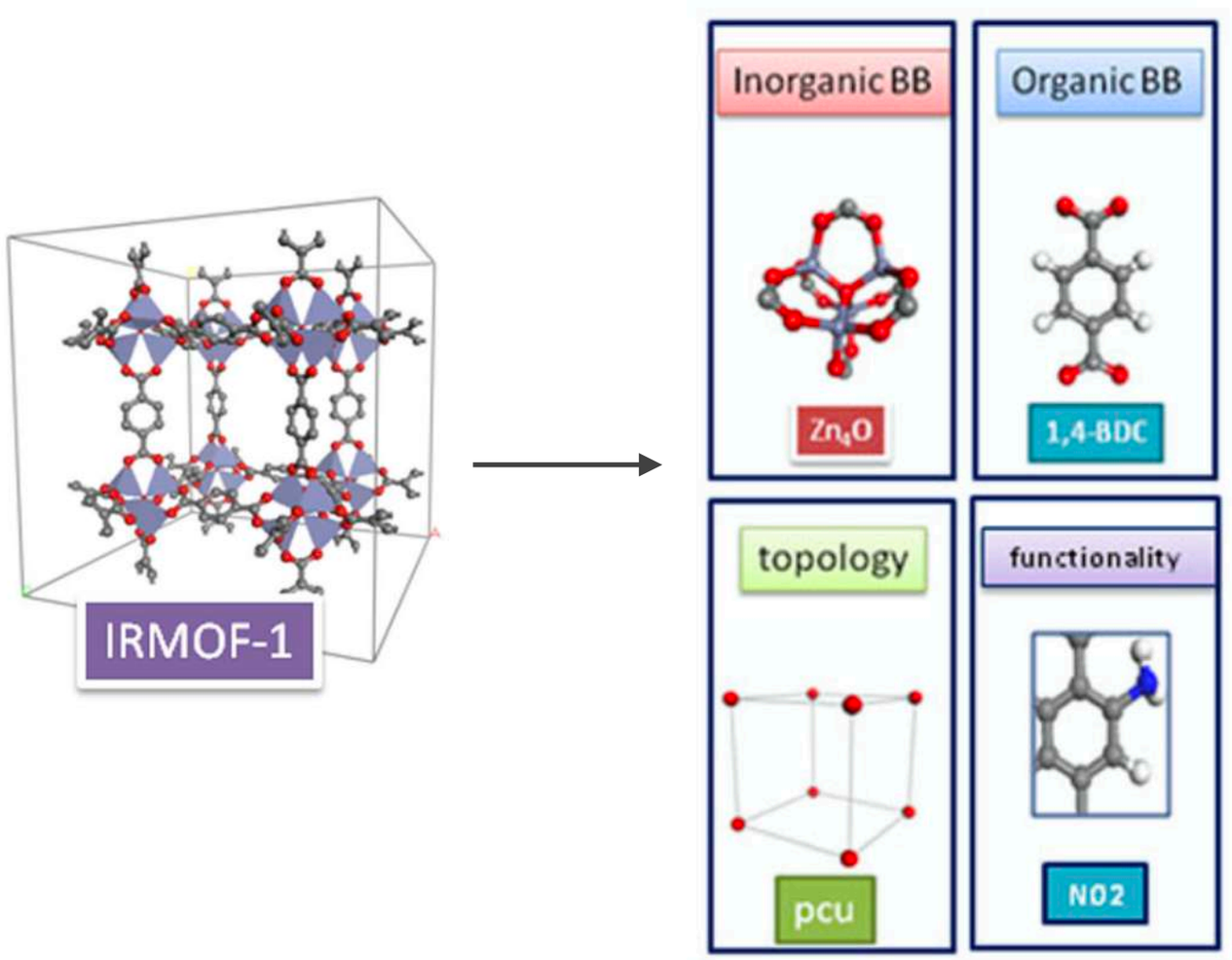
- Invariant w.r.t. symmetries
- As low dimension as possible
- Cheap to compute
- Non-degenerate
- Transferability across elements



Ad hoc descriptors or properties:

Based on chemical intuition
In principle, can work but often not generalisable

One hot featurisation



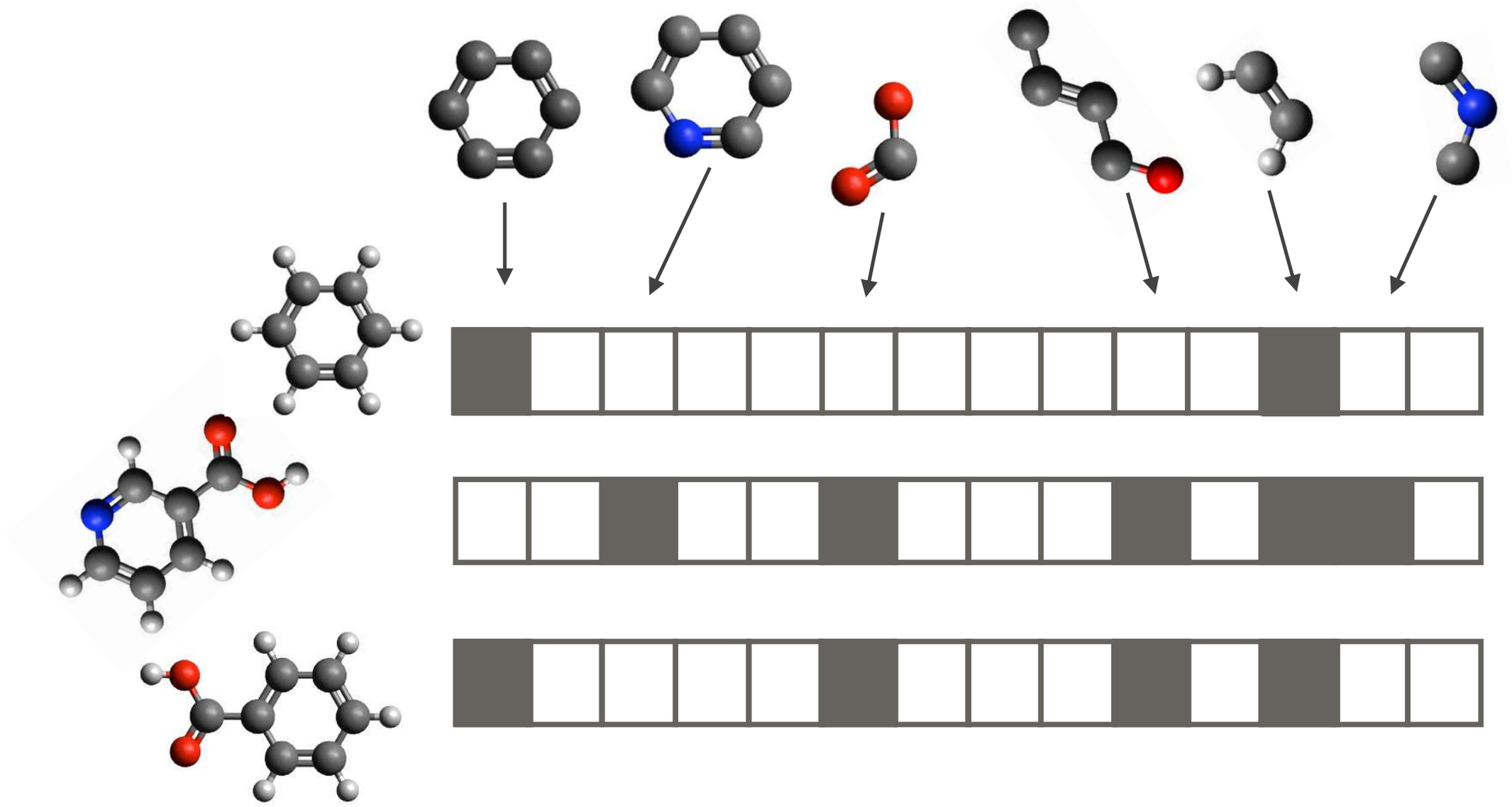
Computed properties:

- Atom identity
- Maximum positive charge
- Minimum negative charge
- etc.

Borboudakis, et al. npj computational materials (2017)
Anderson, et al. Chemistry of materials (2018)

Fragment based descriptors: fingerprints

Binary vectors for molecular similarity
Varying length, e.g., FP2 fingerprint has 1024 bits

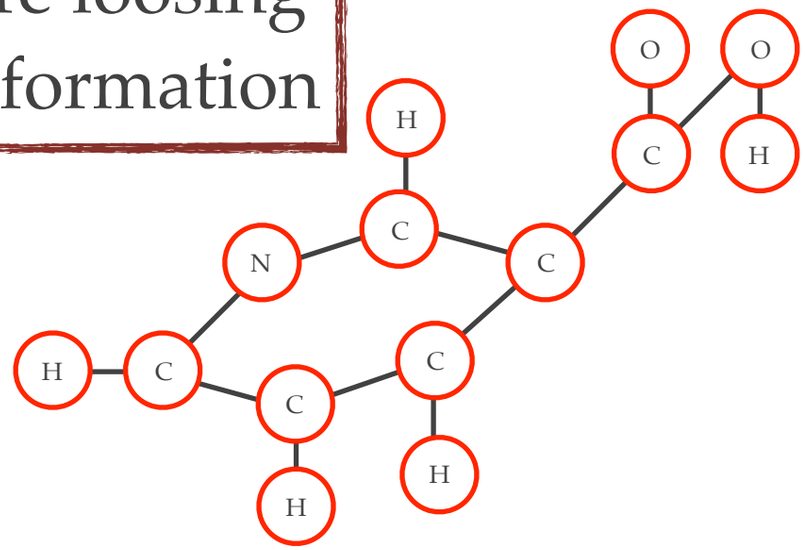


Rogers, D.; Hahn, M. “Extended-Connectivity Fingerprints.” J. Chem. Inf. and Model. **50**:742-54 (2010)

Connectivity based descriptors: RACs

Notice we are loosing geometric information

Molecular graph



Autocorrelations $\underset{scope}{\overset{start}{P}}{}_{d}^{diff} = \sum_{i}^{start} \sum_{j}^{scope} (P_i - P_j)\delta(d_{i,j}, d)$

Atomic properties χ, Z, T, S, I, α

Bond distance

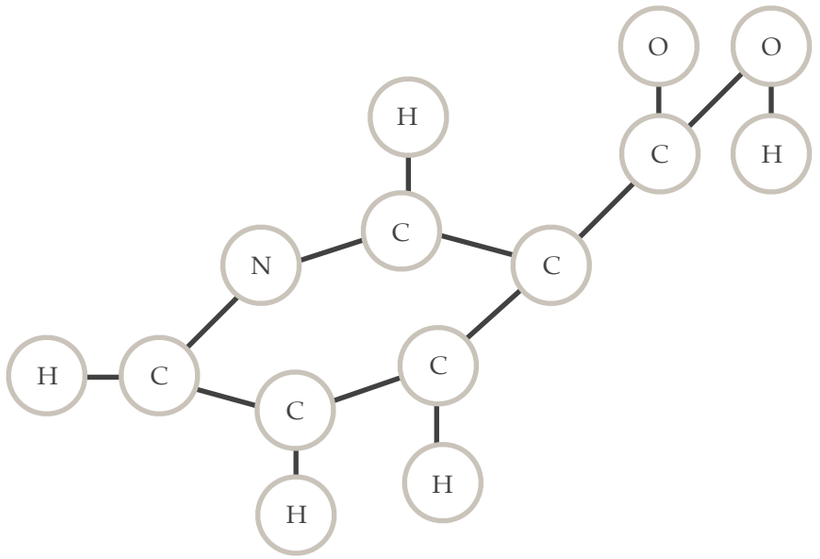
Janet, Jon Paul, and Heather J. Kulik. The Journal of Physical Chemistry (2017)

Connectivity based descriptors: RACs

Autocorrelations $${}^{start}_{scope}P^{diff}_{d} = \sum_{i}^{start} \sum_{j}^{scope} (P_i - P_j)\delta(d_{i,j}, d)$$

Examples:

$${}^{[N]}_{all}\chi^{diff}_{1} = \sum_{i}^{[N]} \sum_{j}^{all} (\chi_i - \chi_j)\delta(d_{i,j},1)$$

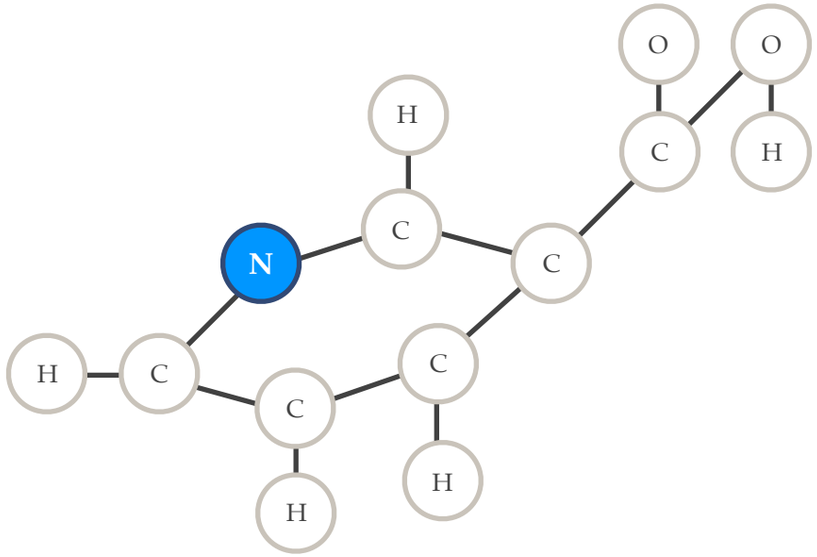


Connectivity based descriptors: RACs

Autocorrelations $${}^{start}_{scope}P^{diff}_{d} = \sum_{i}^{start} \sum_{j}^{scope} (P_i - P_j)\delta(d_{i,j}, d)$$

Examples:

$${}^{[N]}_{all}\chi^{diff}_{1} = \sum_{i}^{[N]} \sum_{j}^{all} (\chi_i - \chi_j)\delta(d_{i,j},1)$$



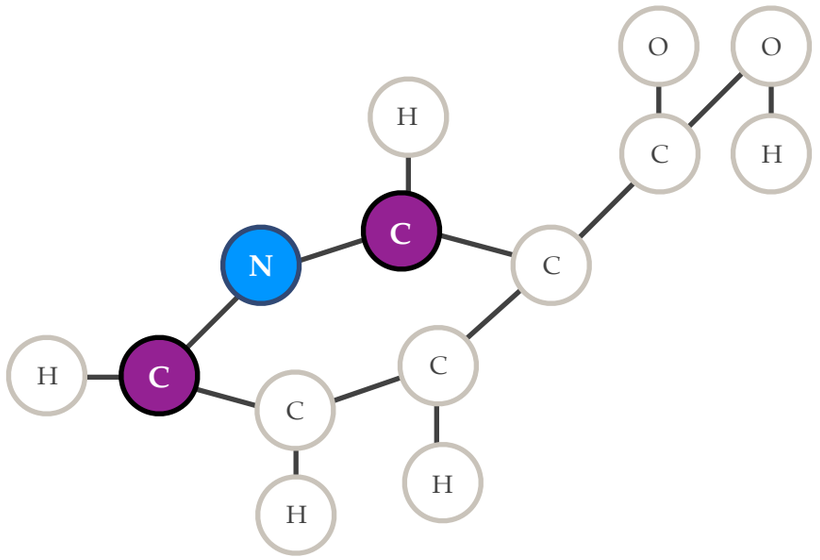
Connectivity based descriptors: RACs

Autocorrelations $${}^{start}_{scope}P^{diff}_{d} = \sum_{i}^{start} \sum_{j}^{scope} (P_i - P_j)\delta(d_{i,j}, d)$$

Examples:

1 bond distance

$${}^{[N]}_{all}\chi^{diff}_{1} = \sum_{i}^{[N]} \sum_{j}^{all} (\chi_i - \chi_j)\delta(d_{i,j},1)$$



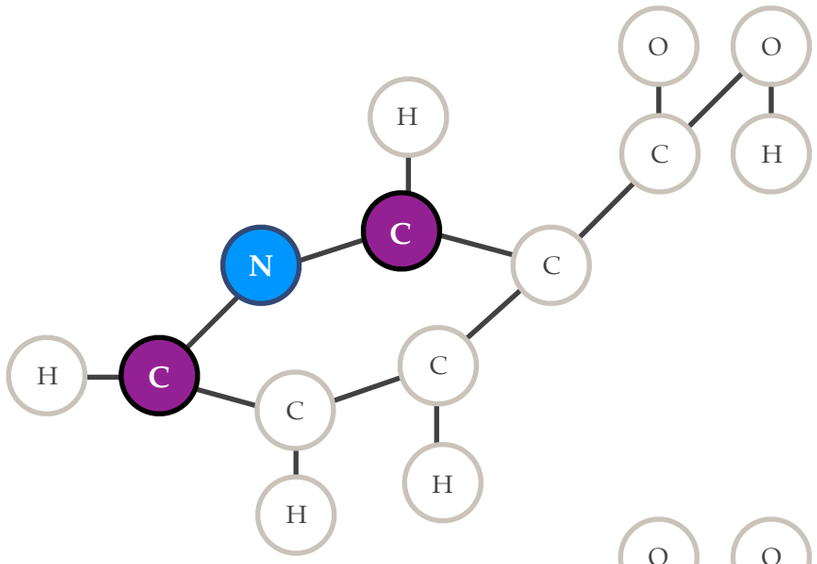
Connectivity based descriptors: RACs

Autocorrelations $${}^{start}_{scope}P^{diff}_{d} = \sum_{i}^{start} \sum_{j}^{scope} (P_i - P_j)\delta(d_{i,j}, d)$$

Examples:

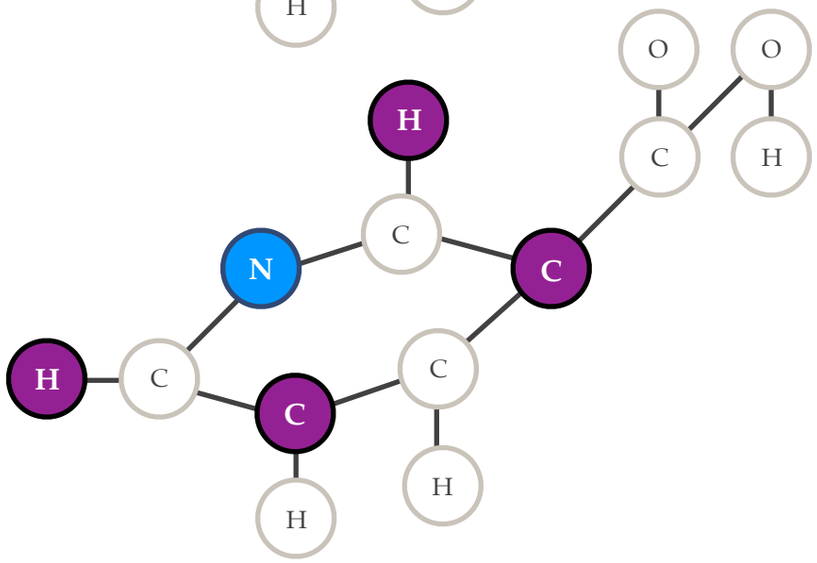
1 bond distance

$${}^{[N]}_{all}\chi^{diff}_{1} = \sum_{i}^{[N]} \sum_{j}^{all} (\chi_i - \chi_j)\delta(d_{i,j},1)$$



2 bond distance

$${}^{[N]}_{all}\chi^{diff}_{2} = \sum_{i}^{[N]} \sum_{j}^{all} (\chi_i - \chi_j)\delta(d_{i,j},2)$$



RACs for MOFs, hands on session in the afternoon

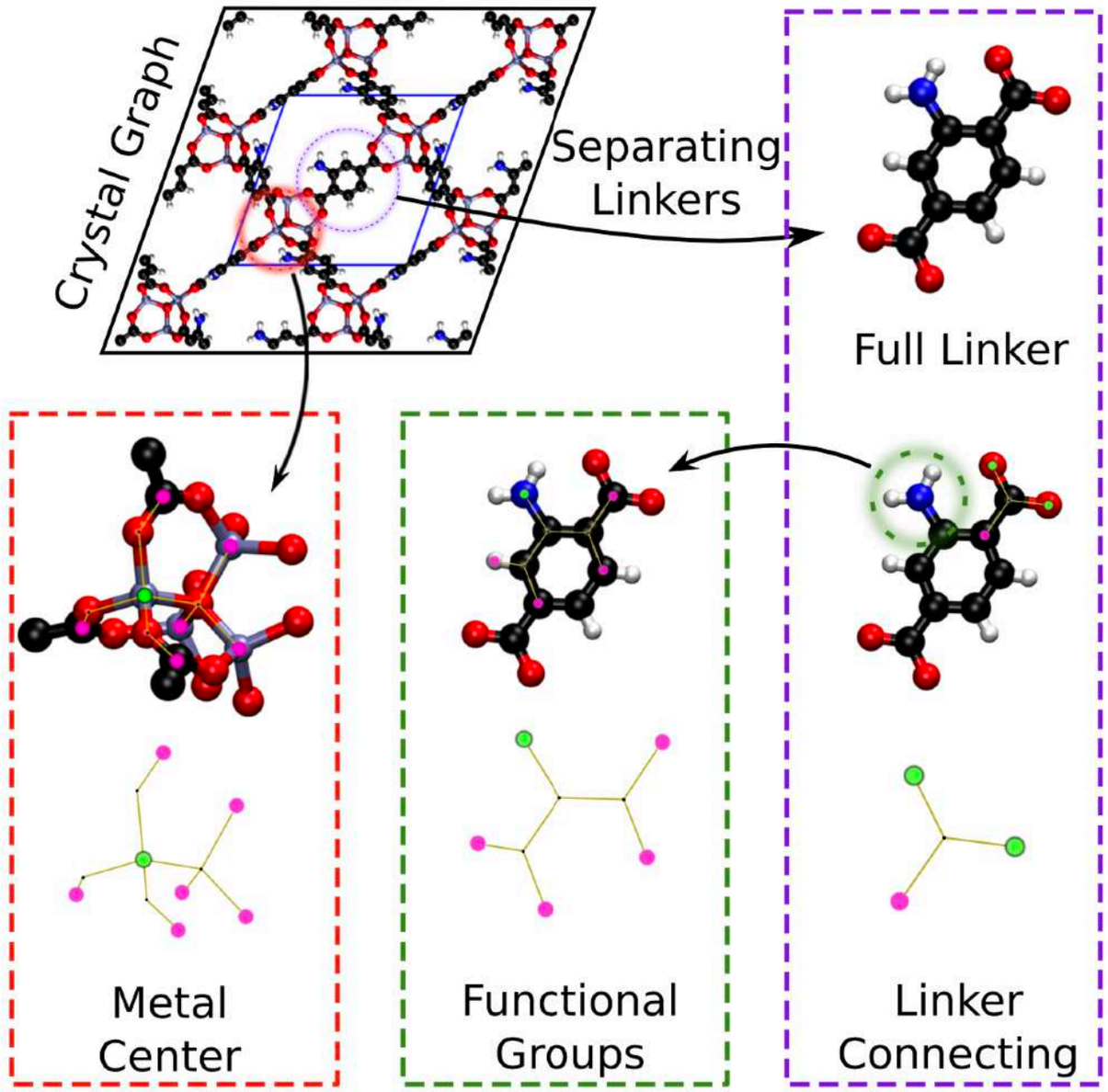
molSimplify

$$ {}^{start}_{scope}P^{diff}_{d} = \sum_{i}^{start\ scope} \sum_{j} (P_i - P_j)\delta(d_{i,j}, d) $$

$$ {}^{start}_{scope}P^{prod}_{d} = \sum_{i}^{start\ scope} \sum_{j} (P_i \times P_j)\delta(d_{i,j}, d) $$

$$ {}^{start}_{scope}R^{type}_{depth} $$

$$ \chi, Z, T, S, I, \alpha $$



https://github.com/hjkgrp/molSimplify

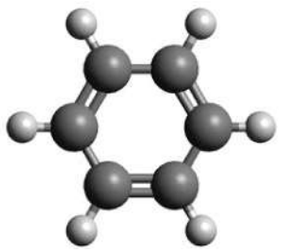
37

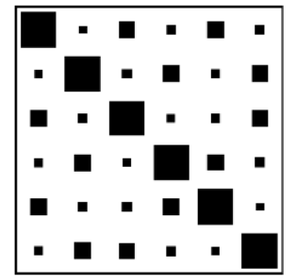
Encoding geometry: Coulomb matrix

Inspired by how quantum mechanics works:

$$H(\{Z,R\}) \xrightarrow{\Psi} E \qquad \Leftrightarrow \qquad \{Z,R\} \xrightarrow{\mathrm{ML}} E$$

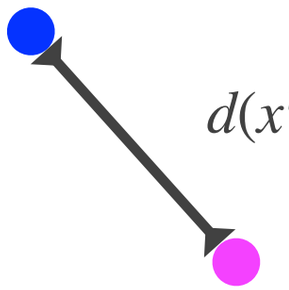
$$M_{ij} = \begin{cases} 0.5\, Z_i^{2.4} & i = j \\ \dfrac{Z_i\, Z_j}{(|\mathbf{r}_i - \mathbf{r}_j|)} & i \neq j \end{cases}$$





Similarity is defined as:

The difference in eigenvalues of **M**s between two systems

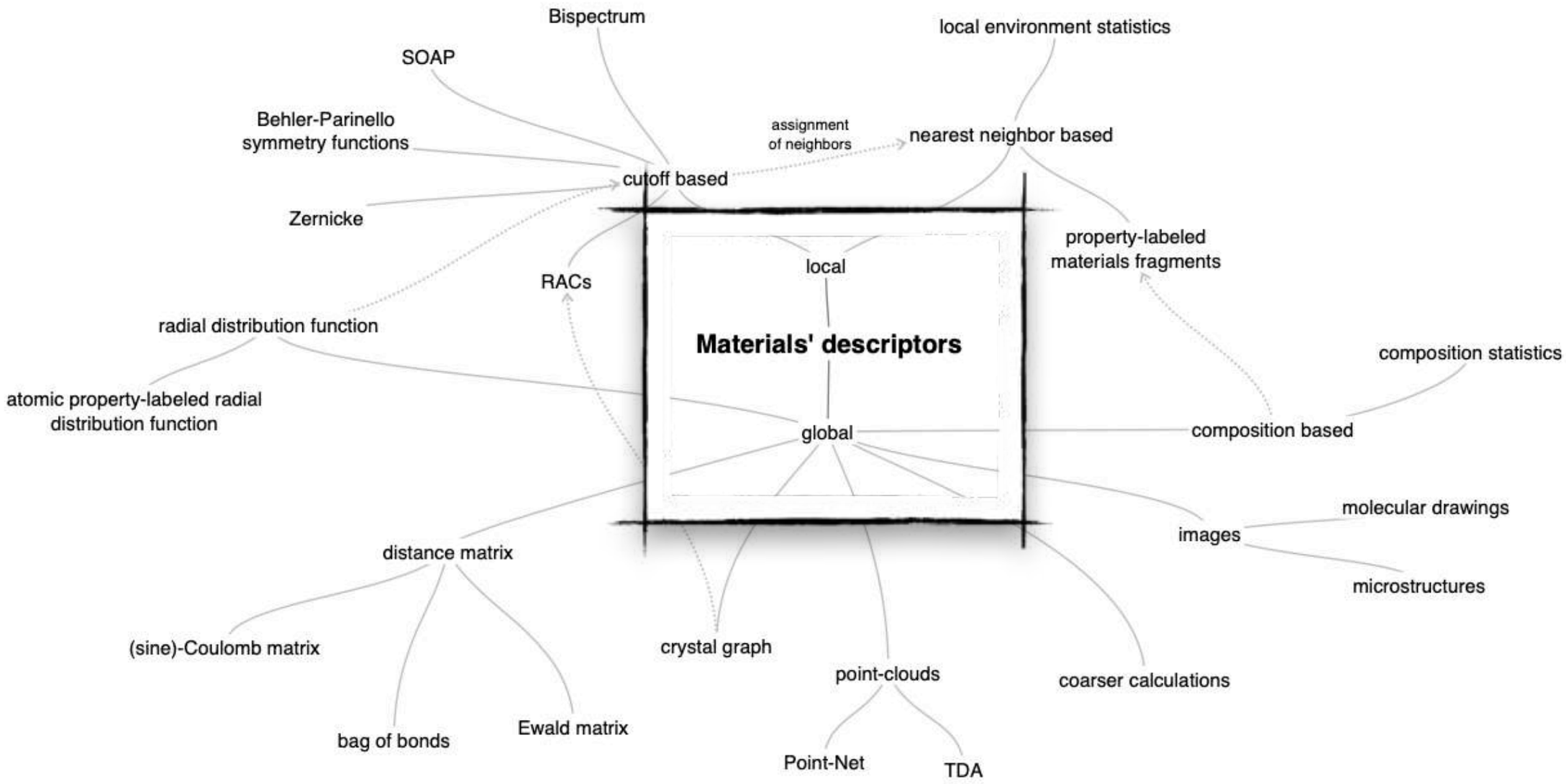


$$d(x^i, x^j) = d(\epsilon^i, \epsilon^j) = \sqrt{\sum_I |\epsilon^i - \epsilon^j|}$$

Rupp, Matthias, et al. "Fast and accurate modeling of molecular atomization energies with machine learning." Physical review letters 108.5 (2012): 058301.

And there exist many more!

The choice must be motivated with the application



Encoding local environments: symmetry functions

Chemical Locality Assumption: decomposing property into local environments

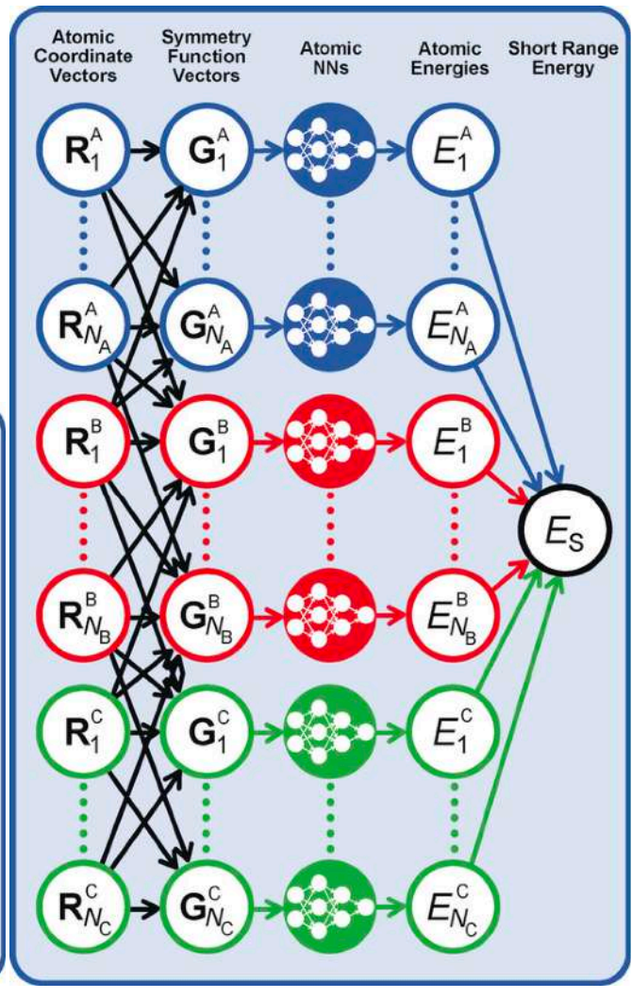
$$\text{property}(\text{descriptor}) = \sum_{i}^{\text{atoms}} \text{models}_i(\text{descriptor}_i)$$

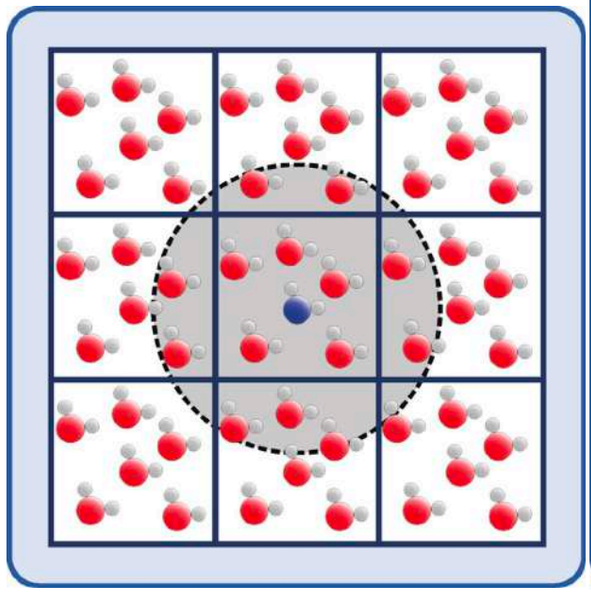
Energy can be decomposed into atomic contributions
—> this approach is used to describe PES
—> Scalable to large systems
—> differentiability of descriptors is essential

$$E_S = \sum_{\nu=1}^{N_{atoms,\nu}} \sum_{\mu=1}^{N_{elem}} E_\mu^\nu$$

$$f_{\text{cut}}(r_{ij}) = \begin{cases} \frac{1}{2}\left[\cos\left(\pi \frac{r_{ij}}{r_c}\right) + 1\right] & \text{for } r_{ij} \le r_{\text{cut}} \\ 0 & \text{for } r_{ij} > r_{\text{cut}} \end{cases}$$

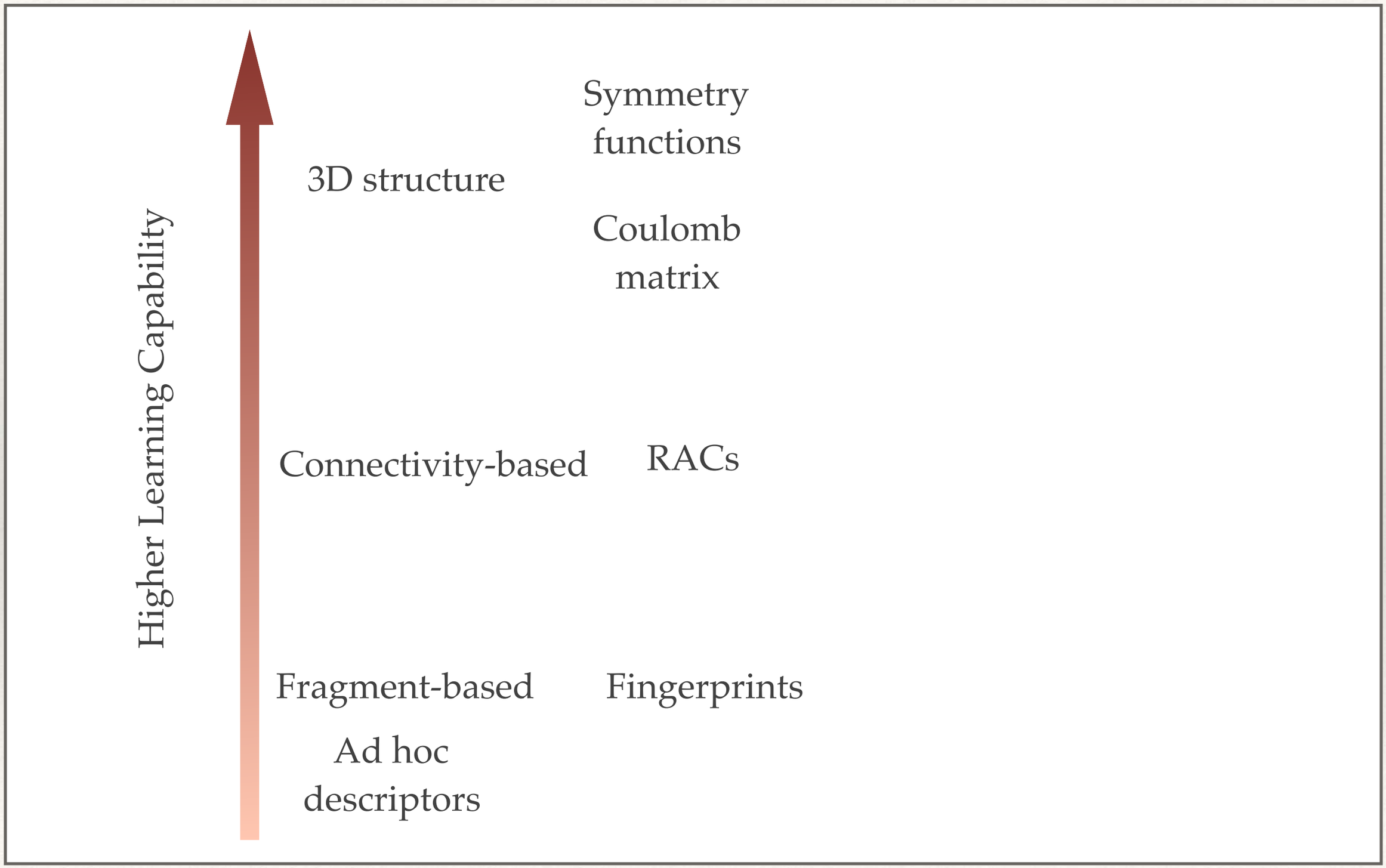
$$G_i^2 = \sum_j \exp\left[-\eta_i (r_{ij} - r_{si})^2\right] f_{cut}(r_{ij})$$





Behler, Jörg, and Michele Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces." *Physical review letters* 98.14 (2007): 146401.

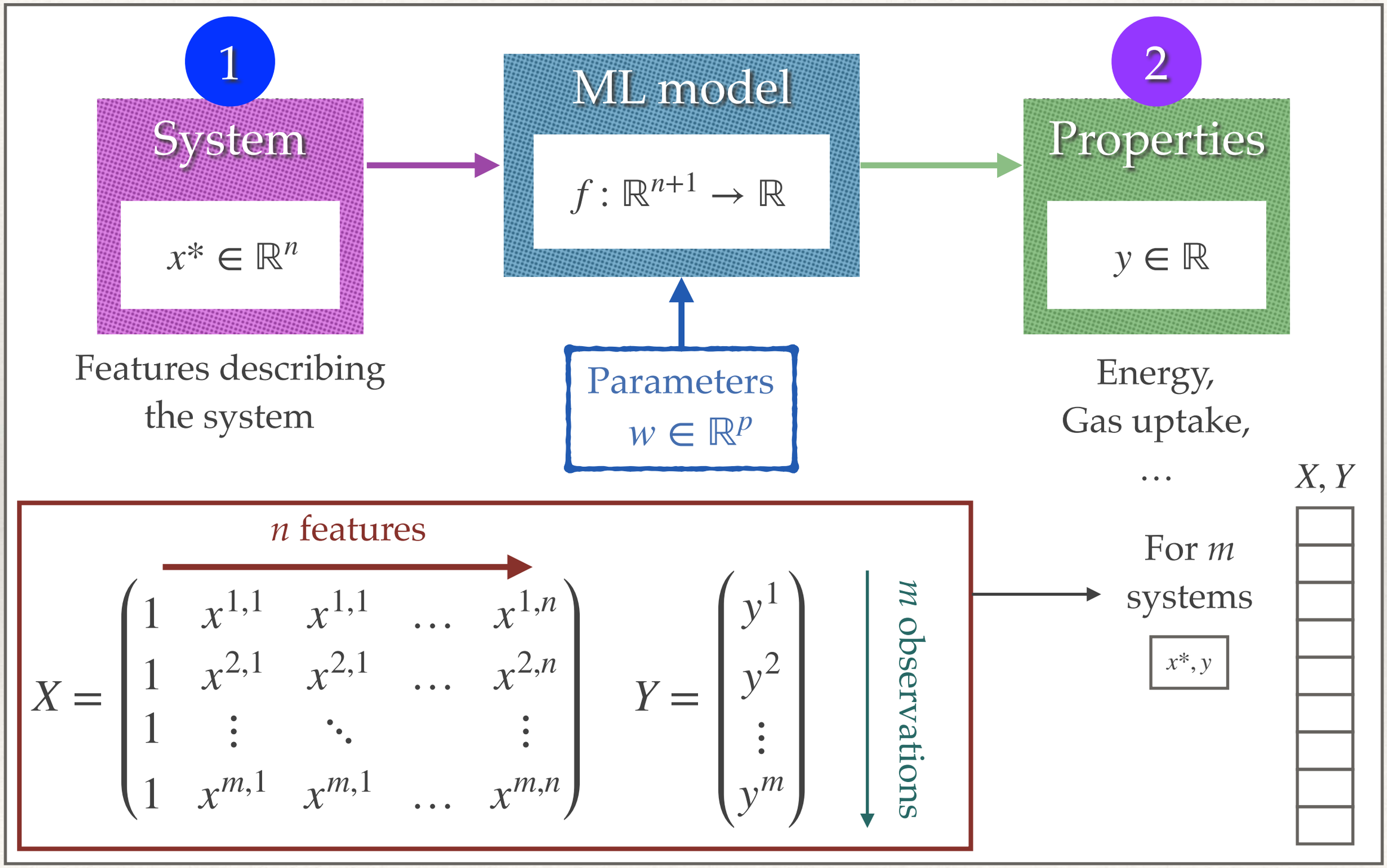
Complexity and richness



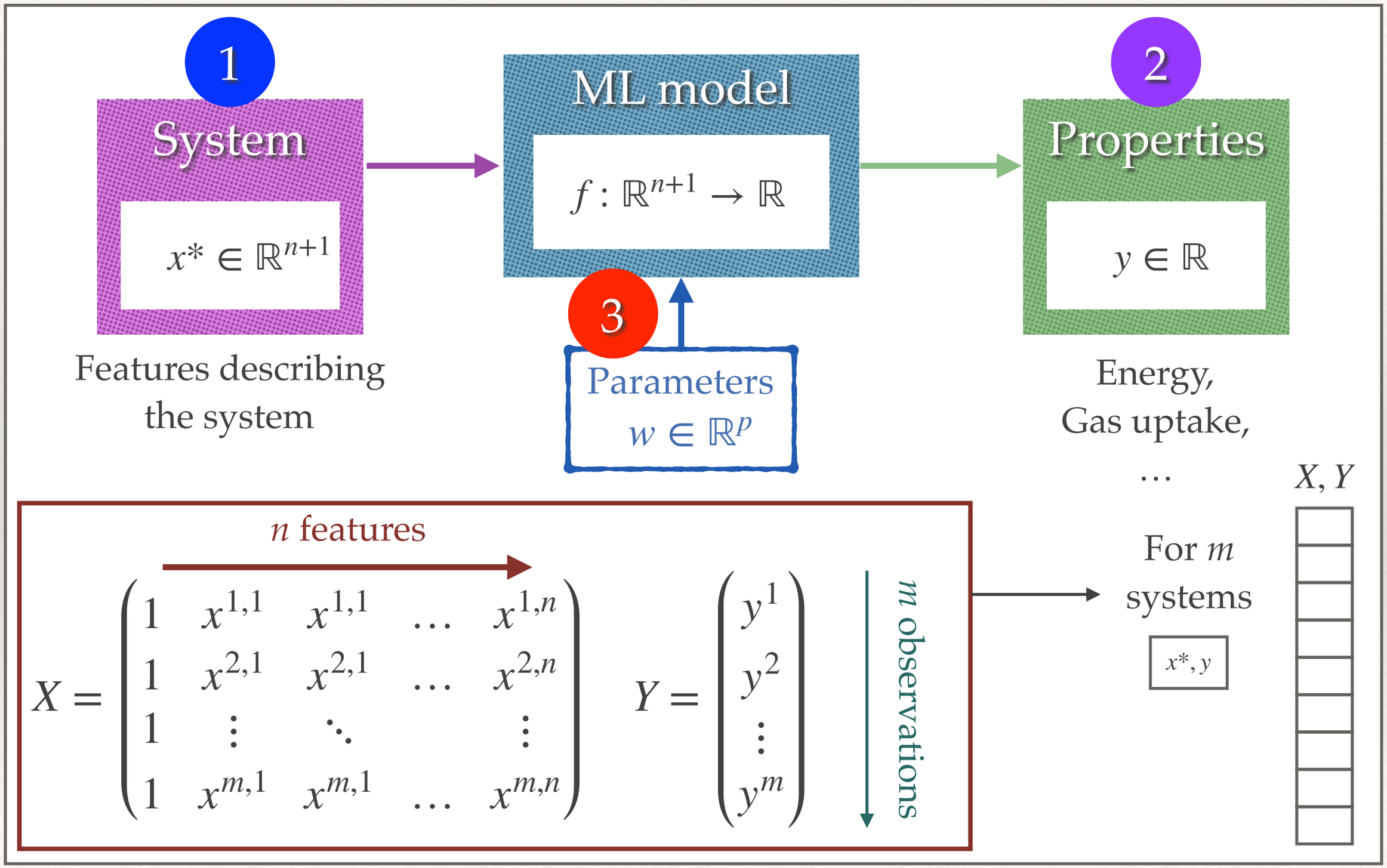
Summary of featurisation

- The aim is to map chemical space to numbers, such that:
  - Chemical similarity is preserved
  - Physics obeyed
- Many kinds of representation exist
  - Global vs. Local
  - Richness and complexity
- Should choose representation based on the application

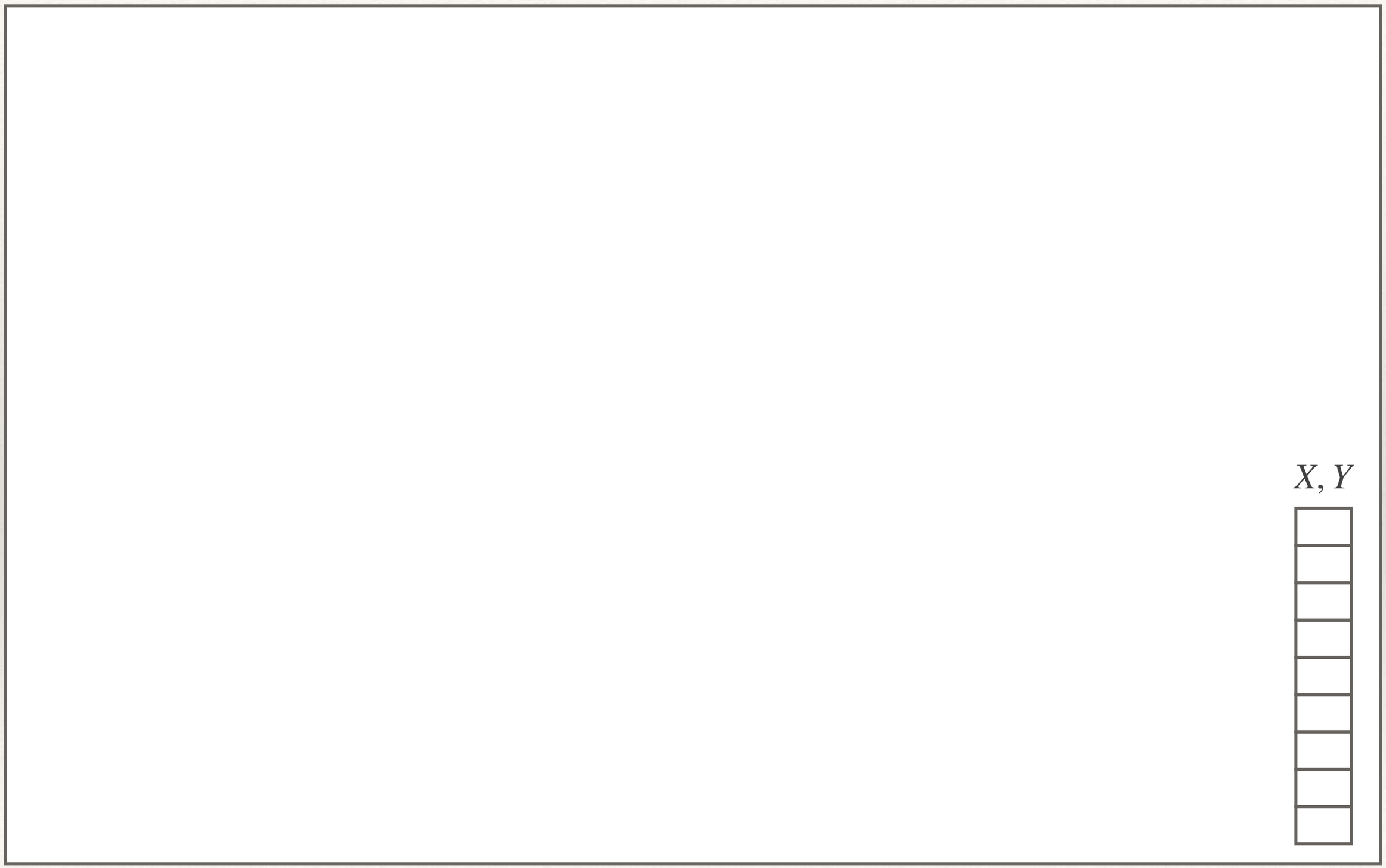
Collect data: features and labels



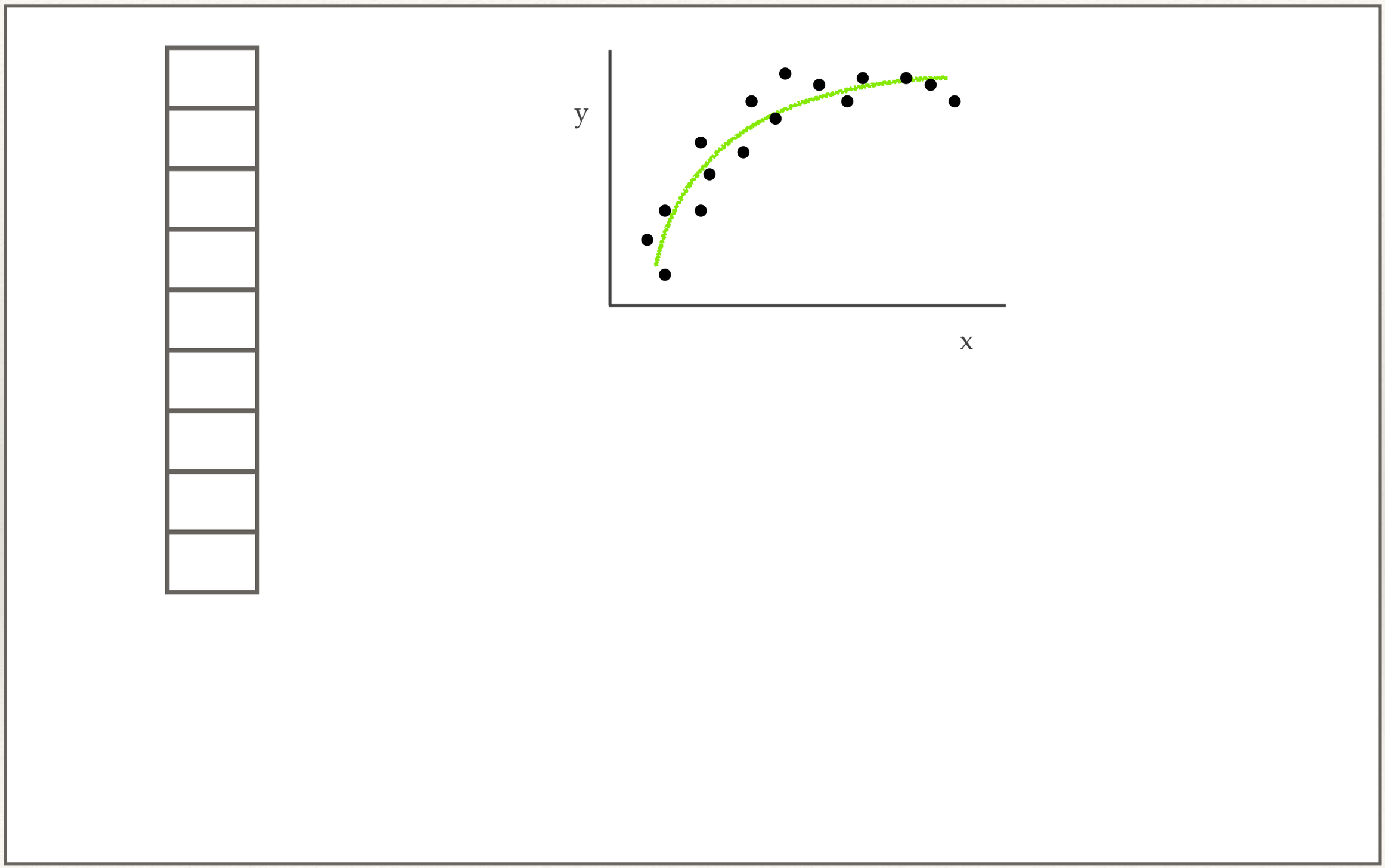
Training the model (finding parameters w)



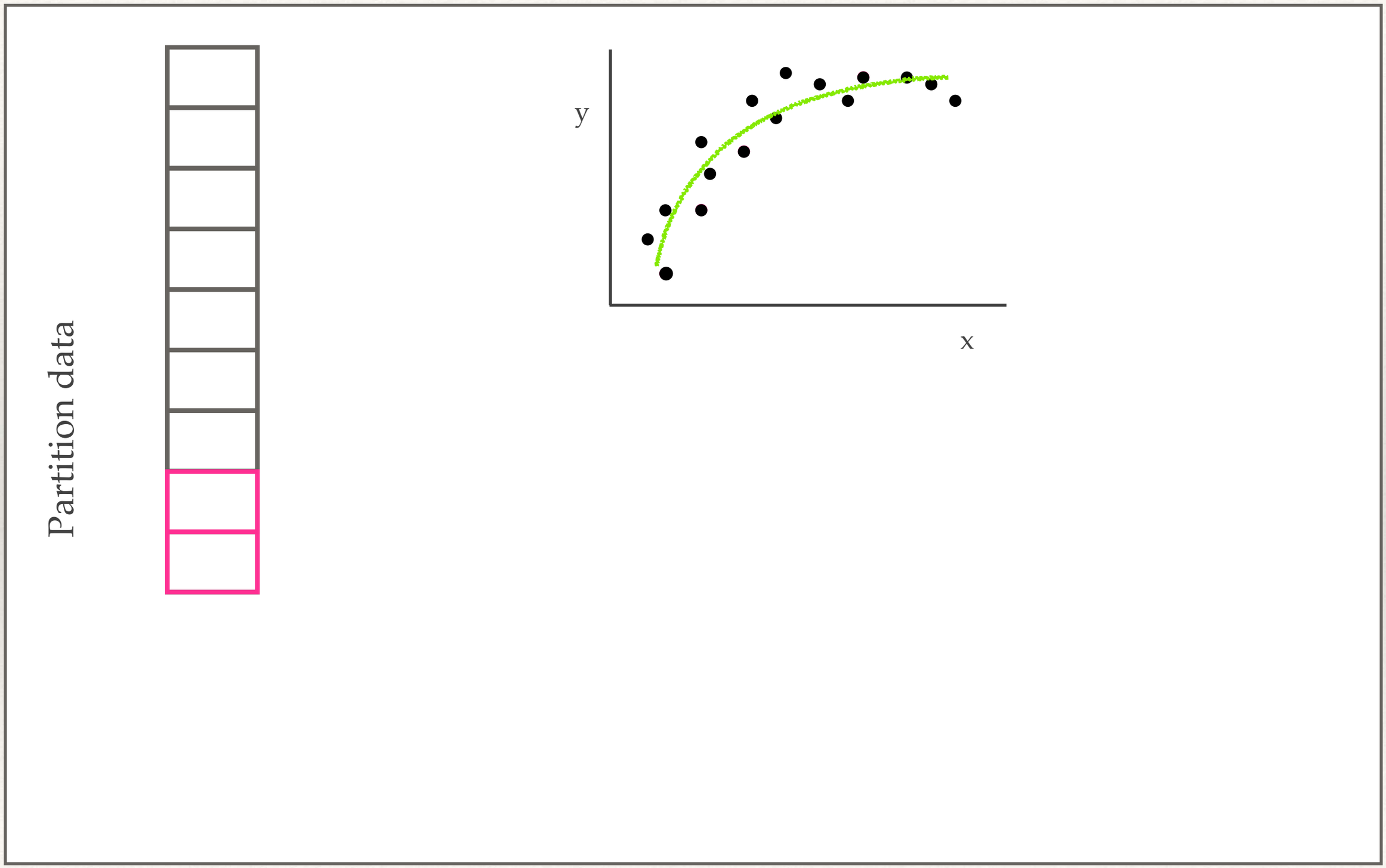
Training the model (finding parameters w)



Training the model (finding parameters *w*)



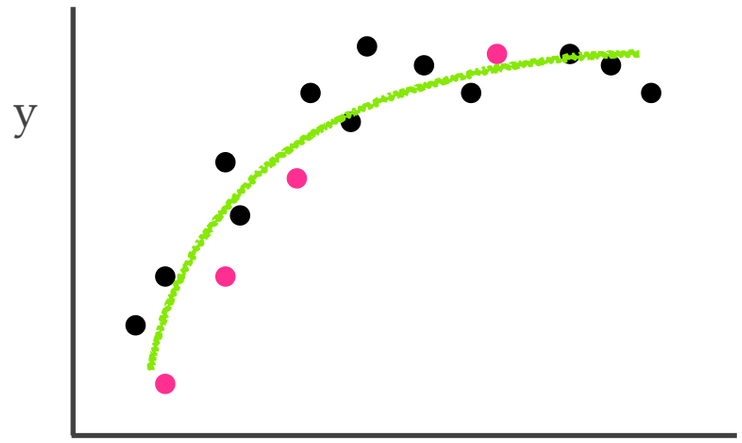
Training the model: data partitioning → keep some data for evaluation



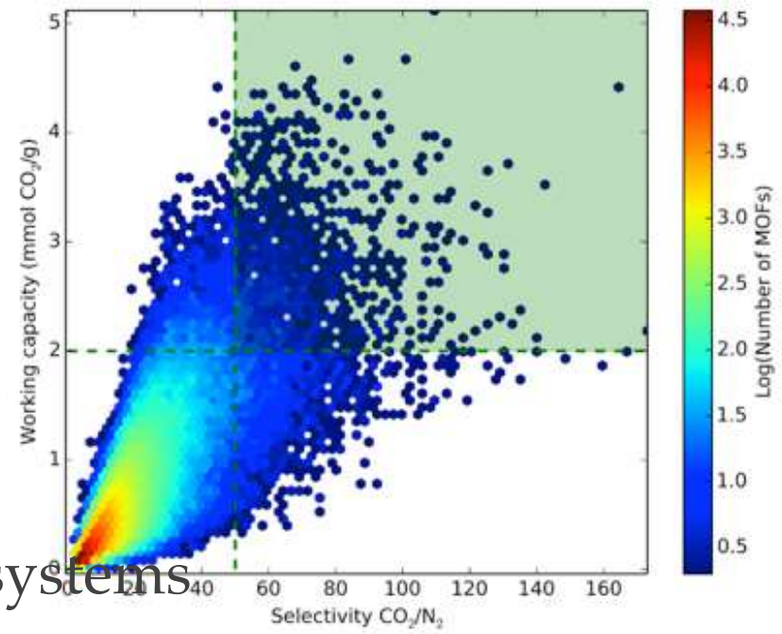
Training the model: data partitioning

Partition data

Train set



Boyd et al. Nature 2019
~300,000 hypothetical MOFs
Search for ~9,000 top performing



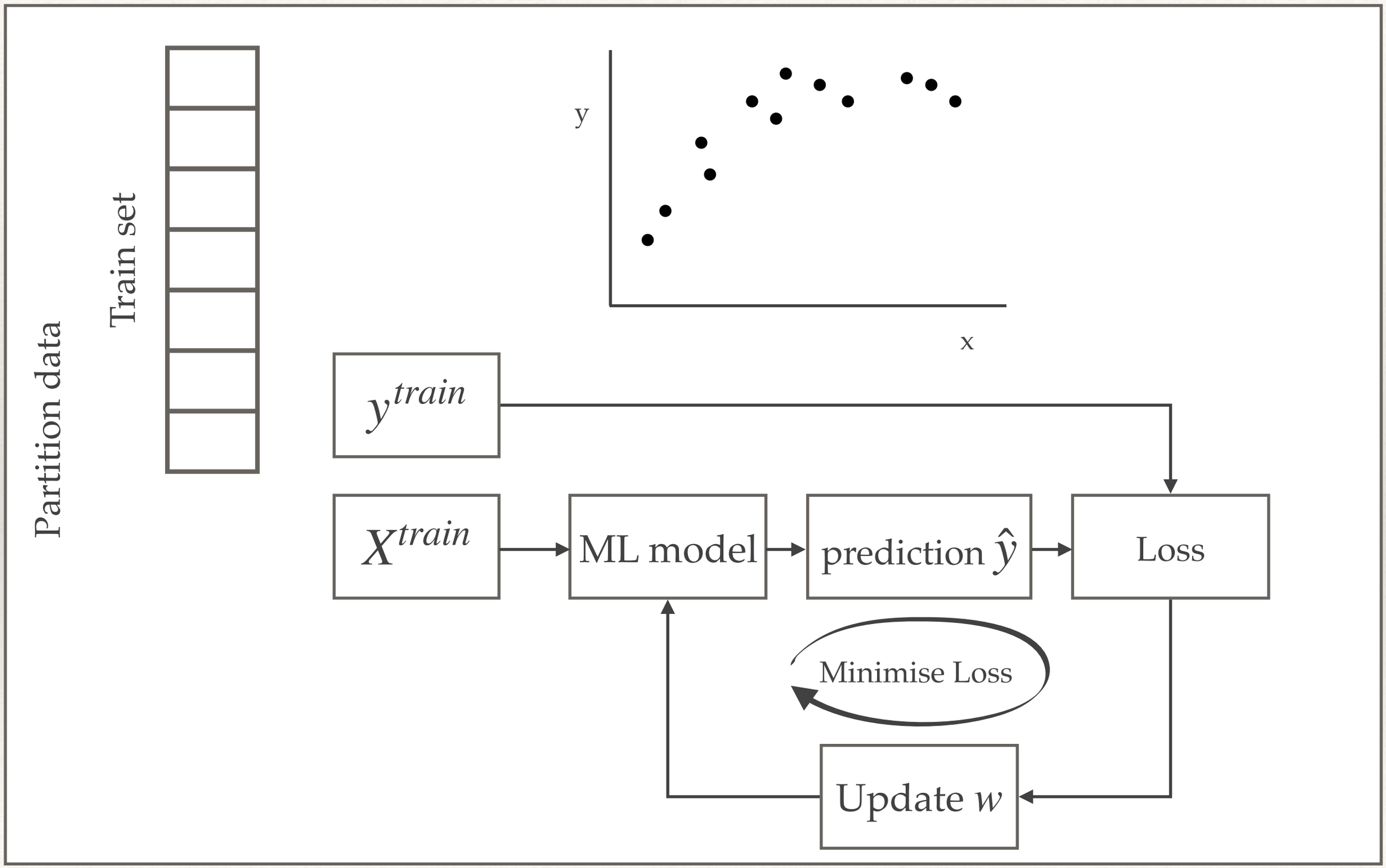
The train-test split is not trivial!

Test set

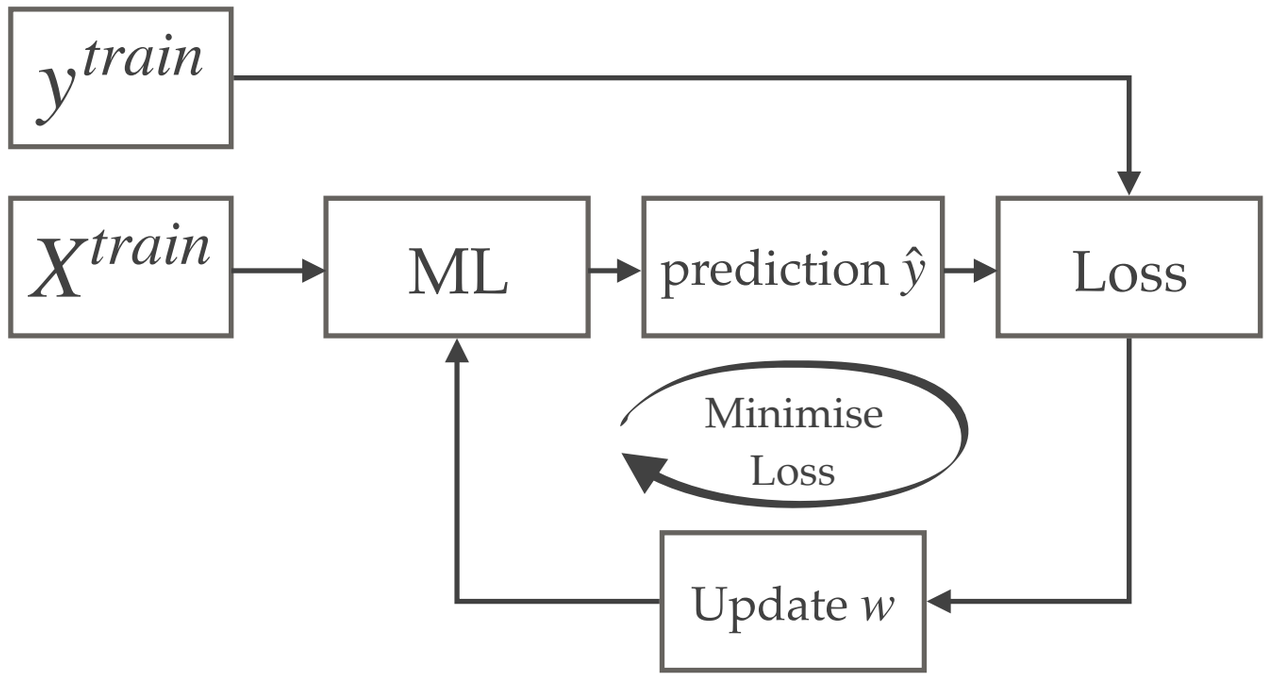
Only for model evaluation!

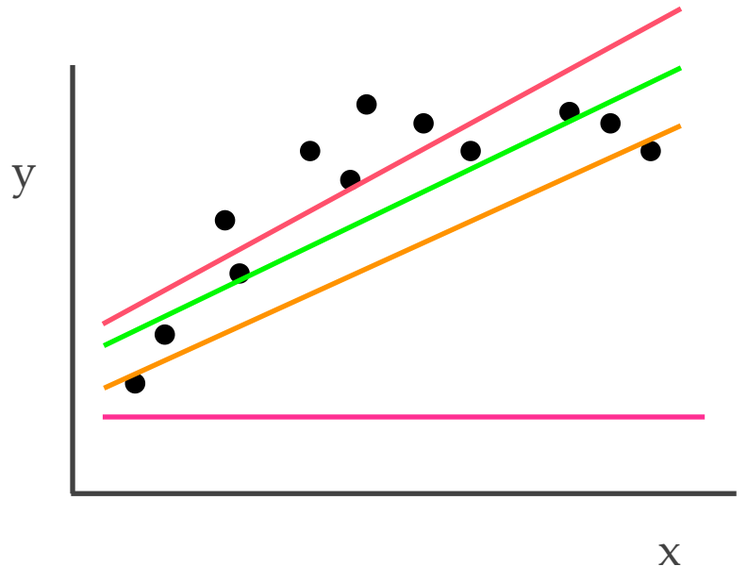
Note: the aim is to eventually use the model on the systems that the model has never seen

Training the model (finding parameters *w*)



Linear regression





A very simple case of 1 dimensional feature:

$$\hat{y}_{ML}(x^*) = x^* w = [1 \quad x^*] \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

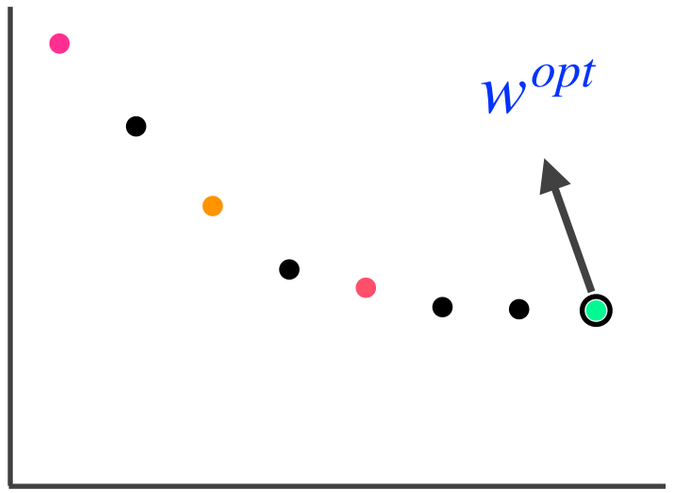
Matrix form:

$$\hat{y}_{ML}(X) = Xw = \begin{bmatrix} 1 & x^1 \\ \vdots & \vdots \\ 1 & x^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Loss function: $\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 = \|Xw - y\|_2^2$

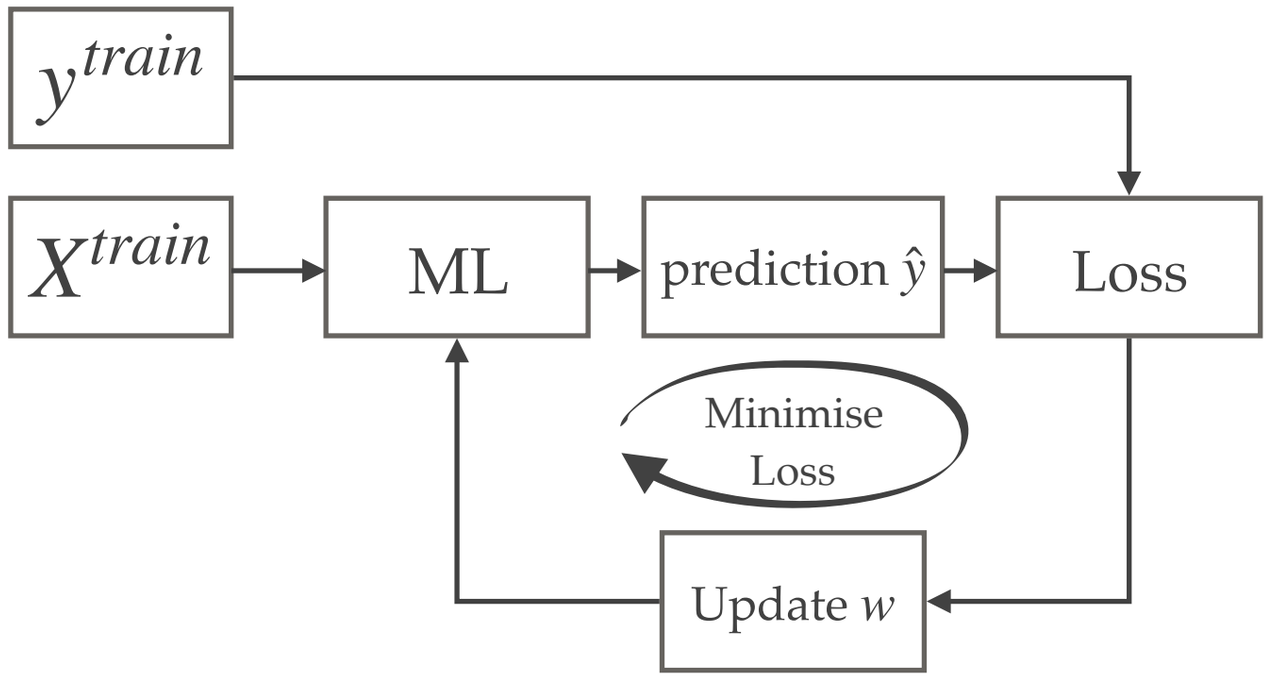
New system: $\hat{y}_{ML}(x^*) = x^* w^{opt}$

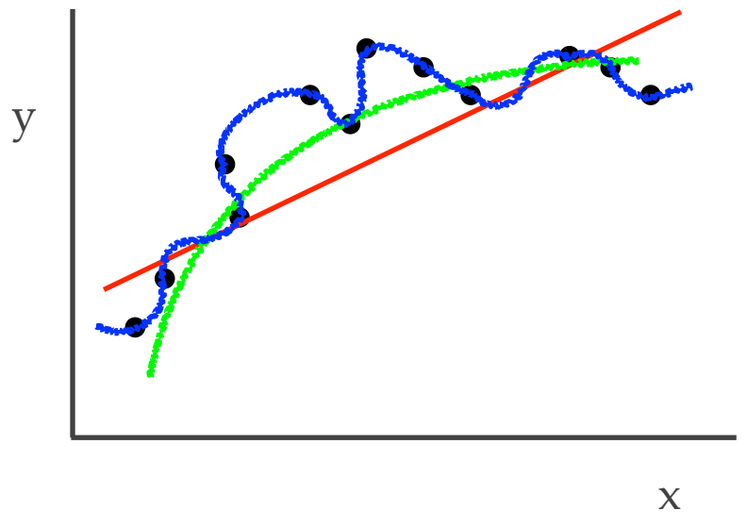
Loss



Optimisation steps

Nonlinear regression





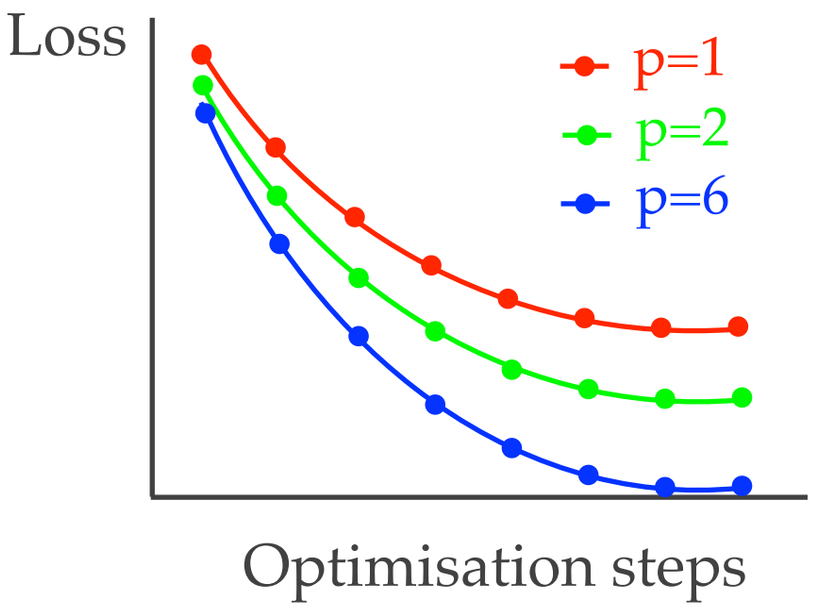
Adding nonlinear terms: polynomial order p

$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} 1 & x^* & (x^*)^2 & \dots & (x^*)^p \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$$

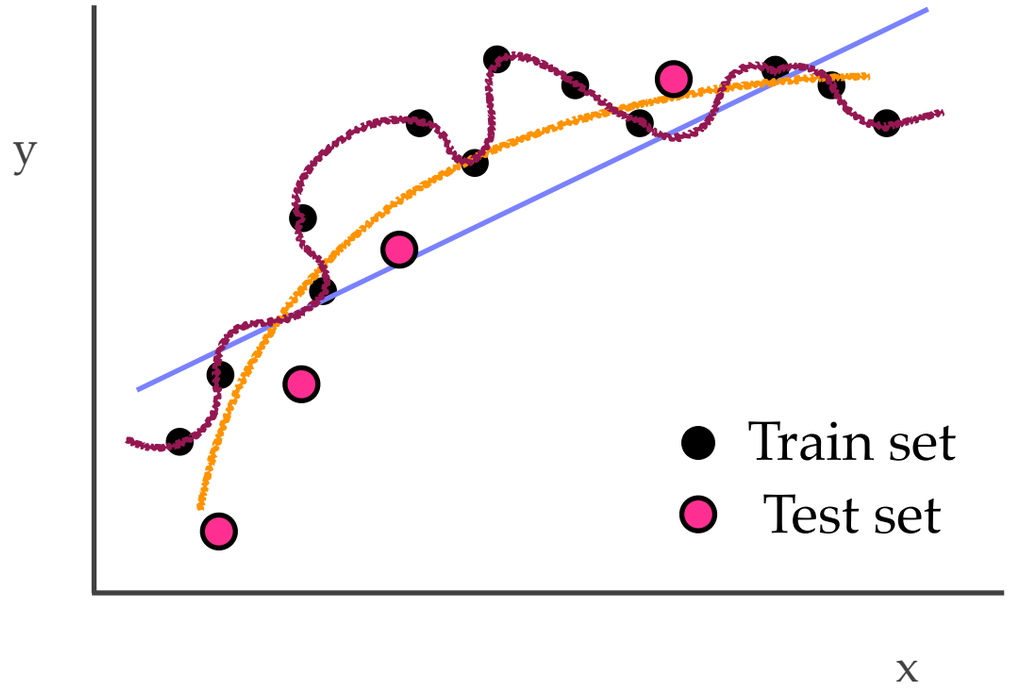
Matrix form:

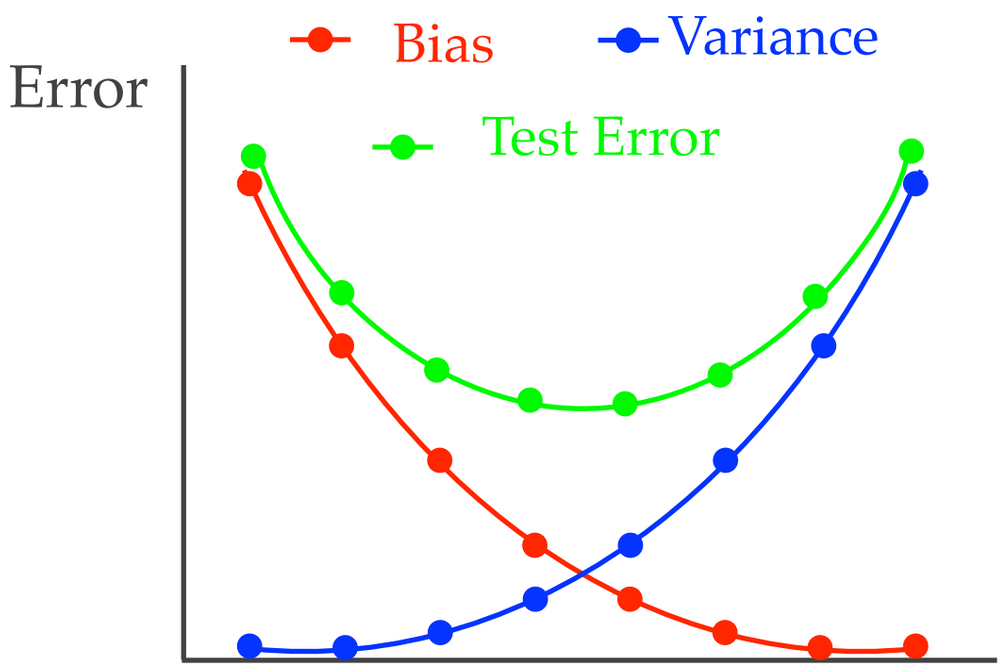
$$\hat{y}_{ML}(X) = Xw = \begin{bmatrix} 1 & x^1 & (x^1)^2 & \dots (x^1)^p \\ \vdots & \vdots & & \ddots \quad \vdots \\ 1 & x^m & (x^m)^2 & \dots (x^m)^p \end{bmatrix} \begin{bmatrix} w_0 \\ \vdots \\ w_p \end{bmatrix}$$

Loss function: $\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 = \|Xw - y\|_2^2$

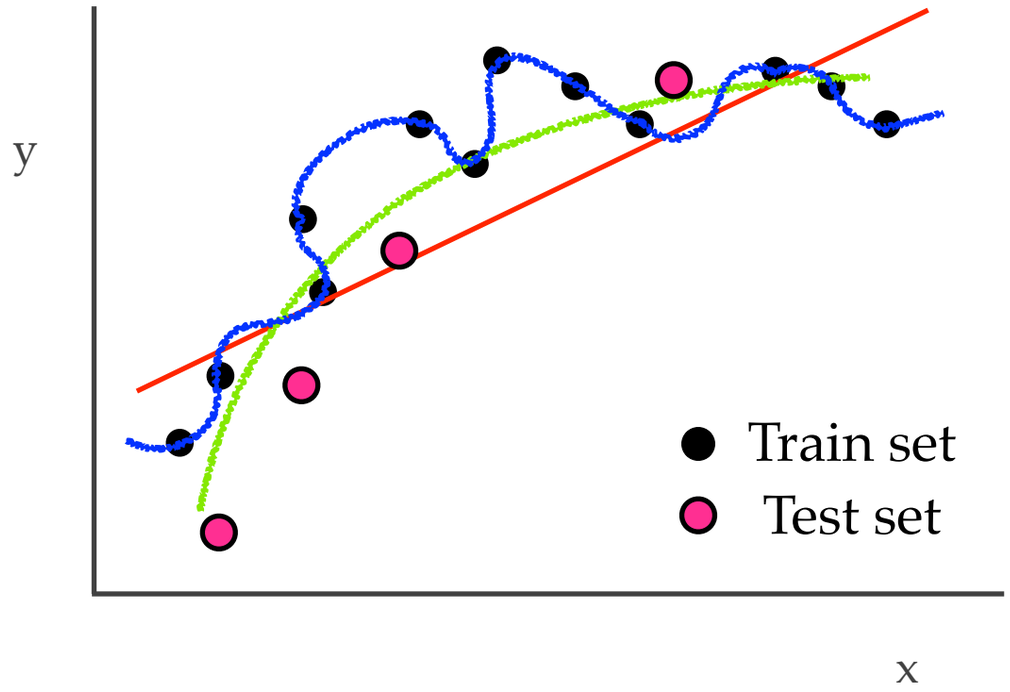


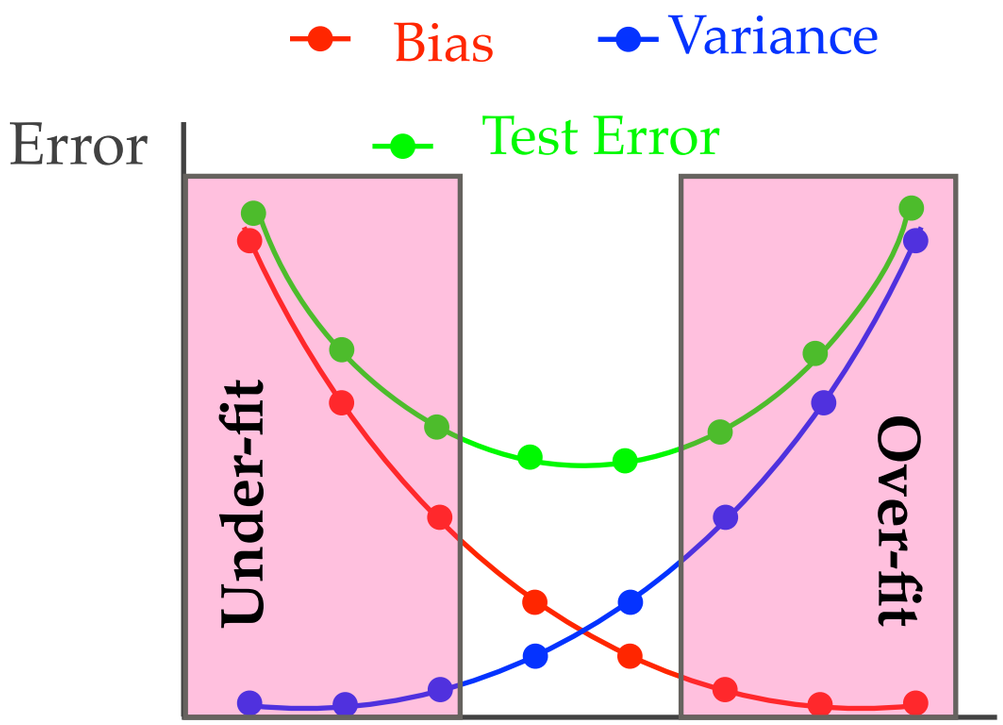
Bias –variance tradeoff





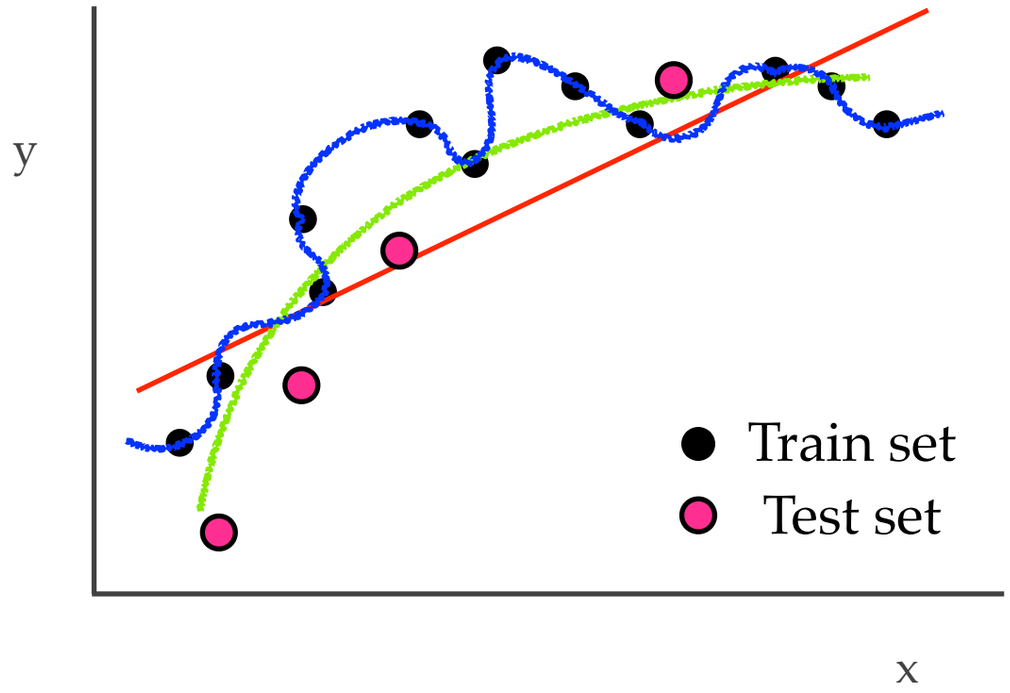
Bias –variance tradeoff

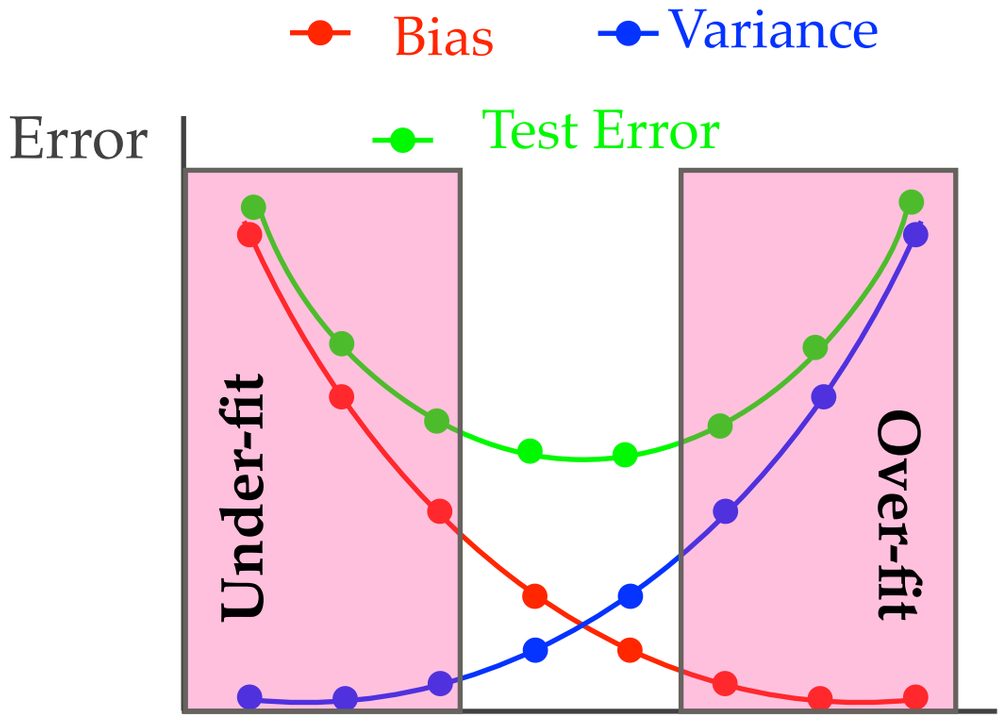




How to include this in the model?

Bias –variance tradeoff: regularisation



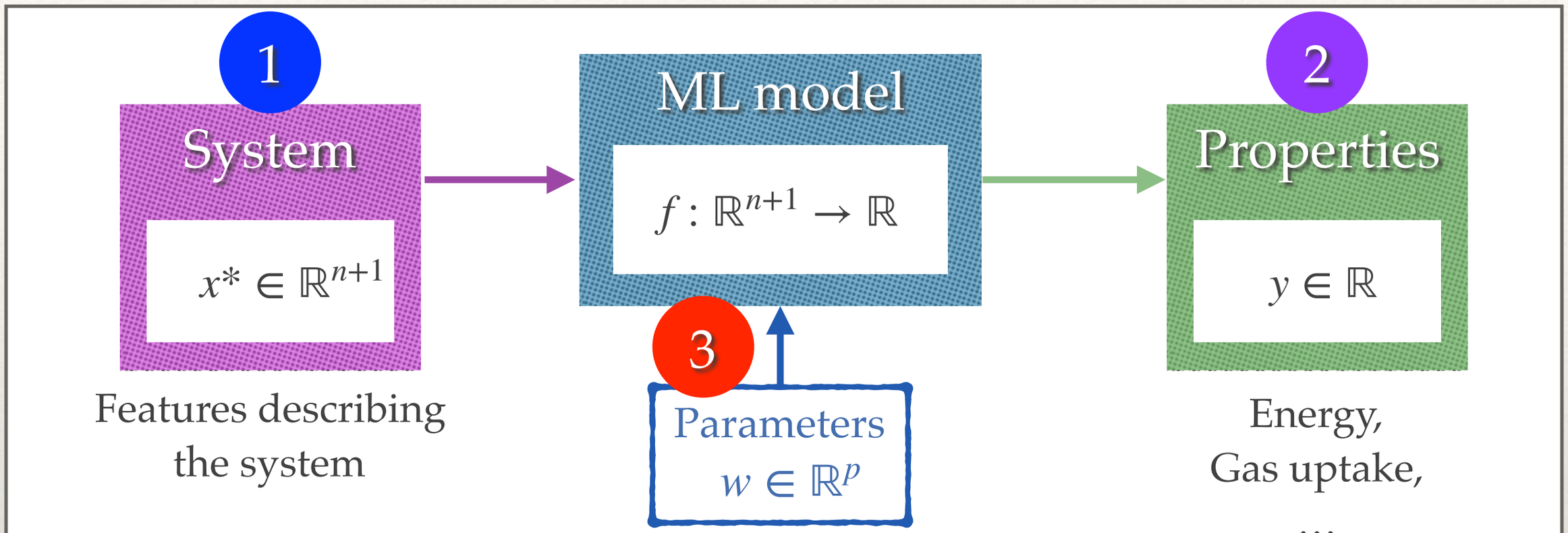


How to include this in the model?

Regularise the loss function: $\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2$

Penalising the terms with large weight

Hyperparameters



So far, we learned how to:

- Represent chemical systems
- Collect data and split it to train-test sets
- Find model parameters w during training by minimising loss

--> What about the hyperparameters, i.e. the parameters we fix before training, e.g., λ

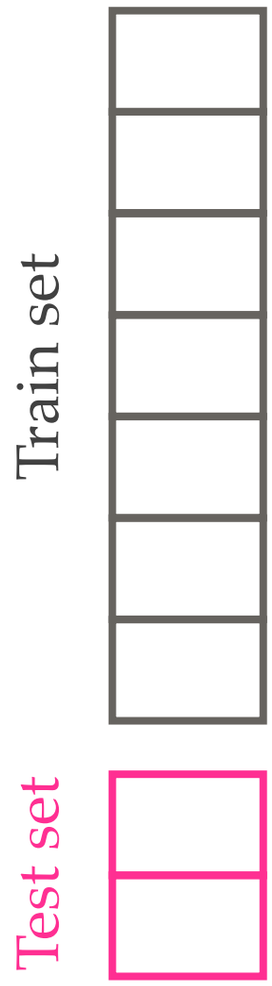
$$\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2$$

Hyperparameter optimisation

We want a model that generalise —> low test error

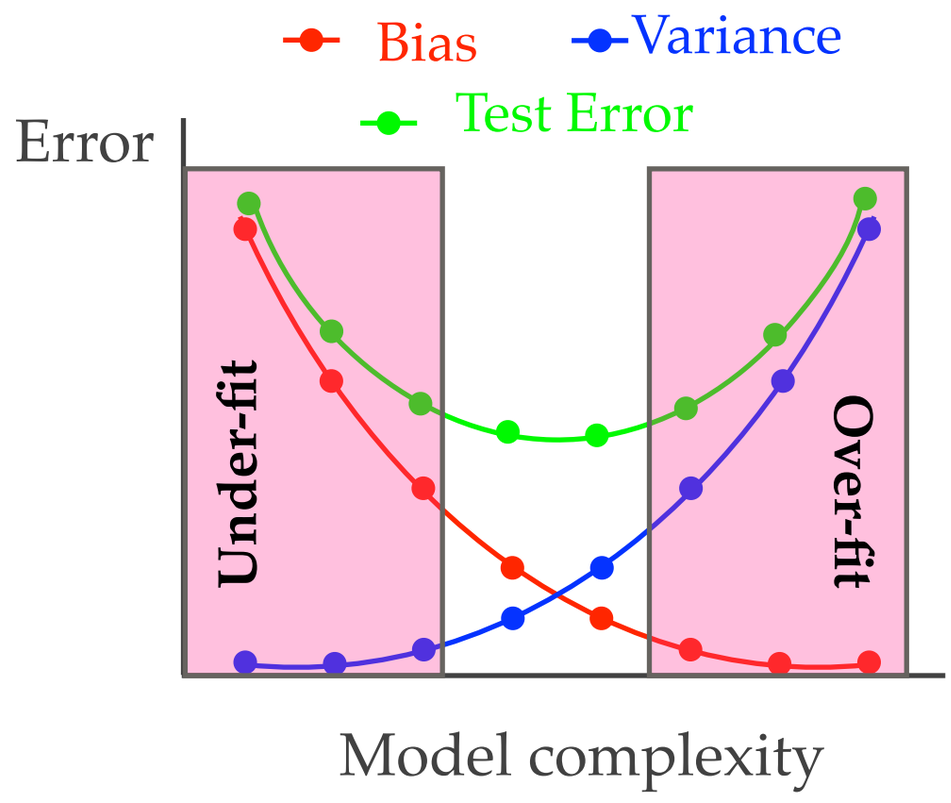
—> but we don't have access to the test error!

—> we need a way to assess model performance only from train data



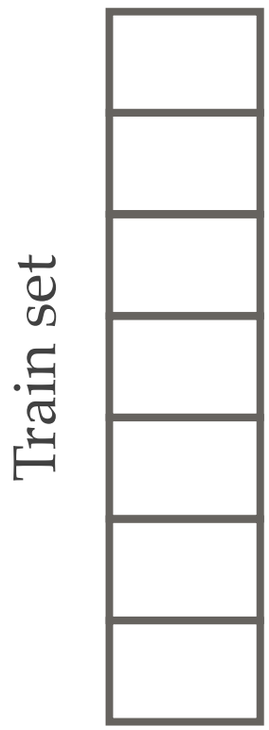
—> **Cross-validation**

Model must not see
this before / while training



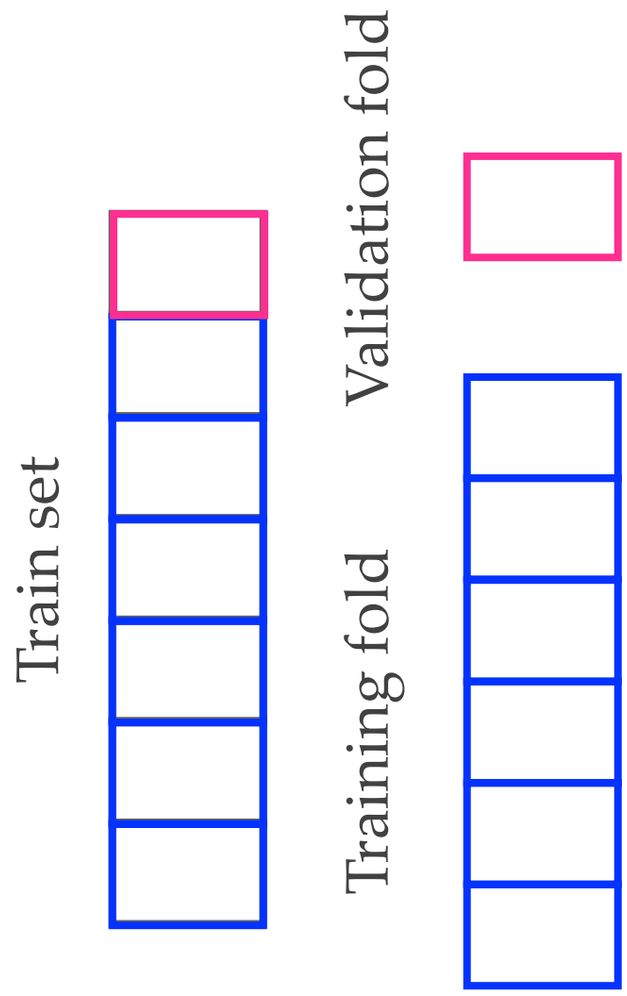
Hyperparameter optimisation: cross-validation

Split the train set into K folds —> use one fold for testing



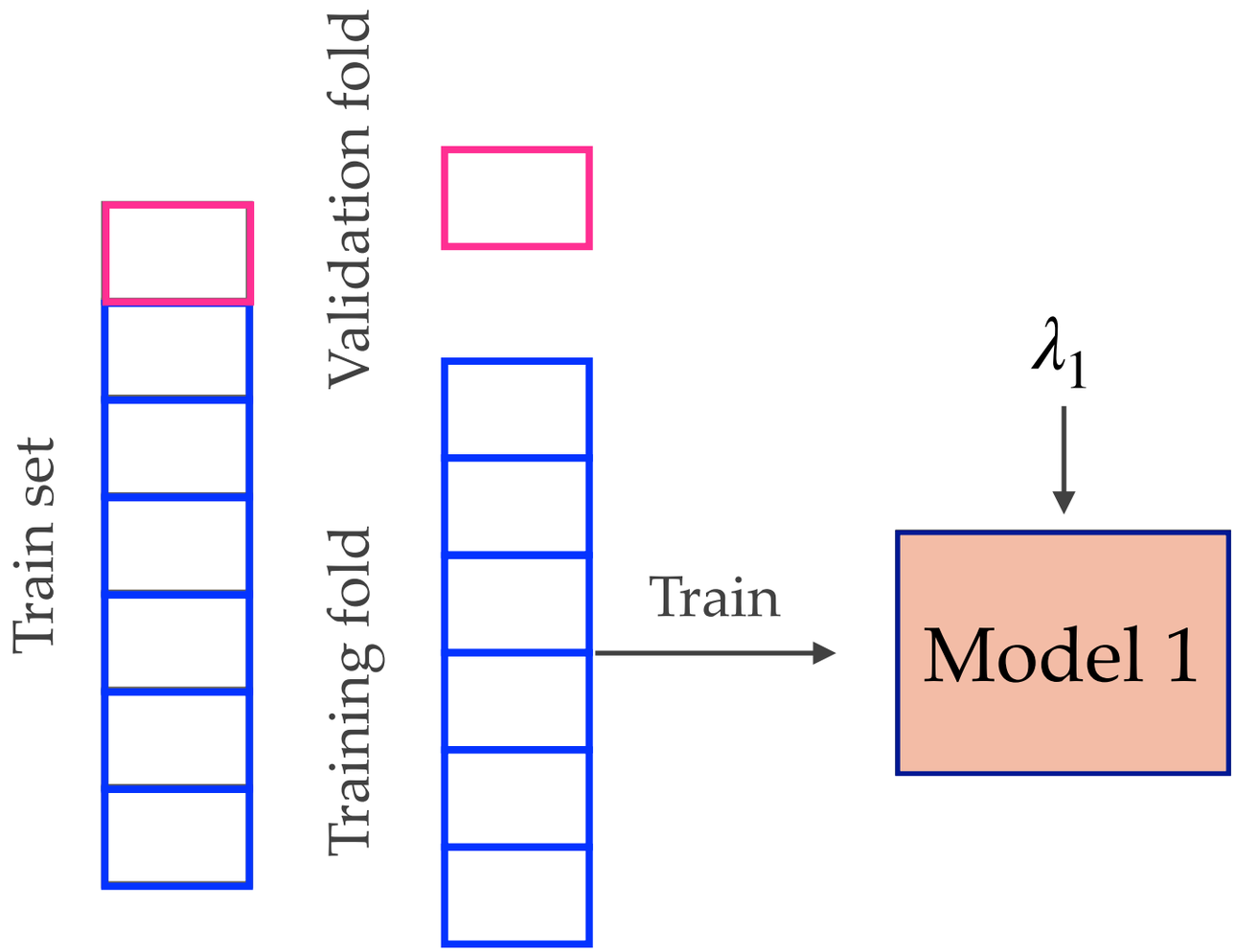
Hyperparameter optimisation: cross-validation

Split the train set into K folds —> use one fold for validation



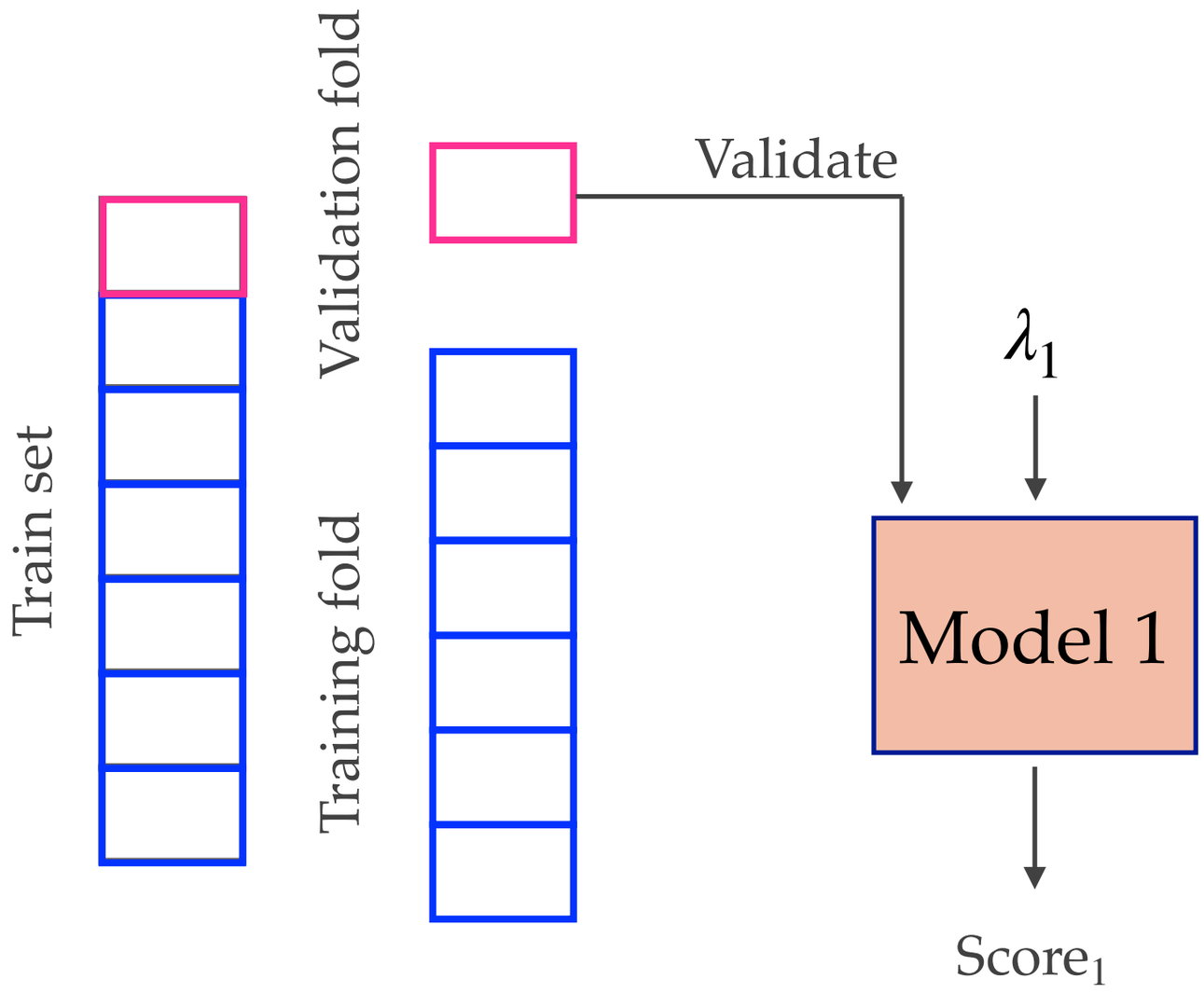
Hyperparameter optimisation: cross-validation

Split the train set into K folds —> use one fold for testing



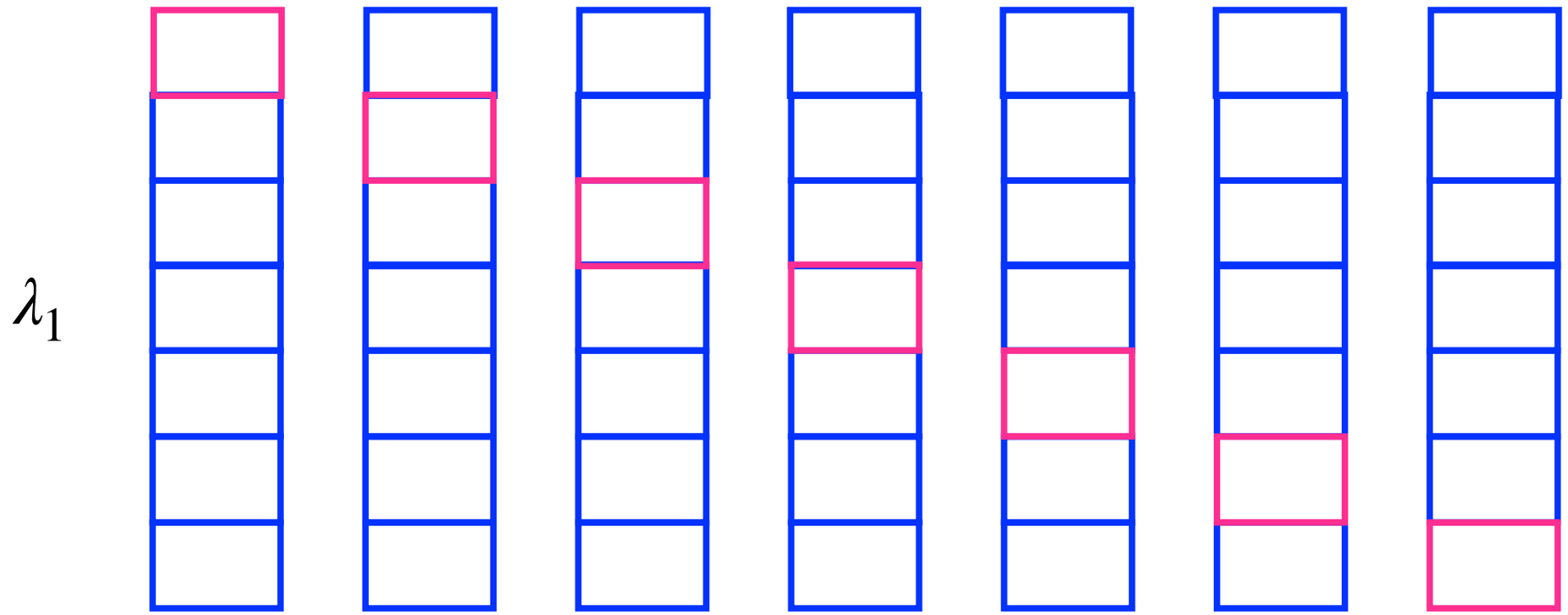
Hyperparameter optimisation: cross-validation

Split the train set into K folds —> use one fold for testing



Hyperparameter optimisation: cross-validation

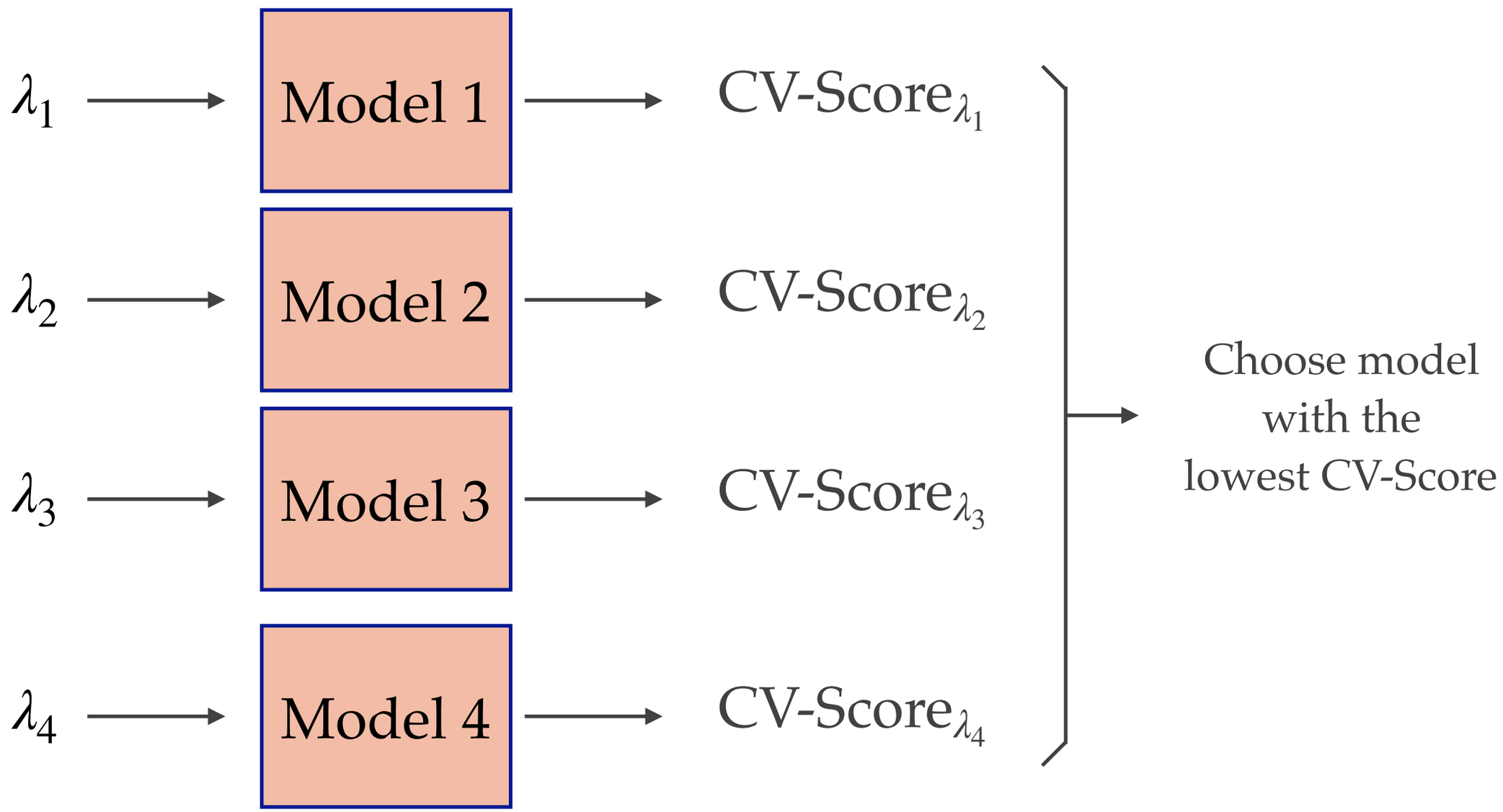
For each set of hyper parameters, we determine cross-validation score



$$\text{CV-Score}_{\lambda_1} = \frac{1}{K}\sum_{i=1}^{K} \text{Score}_{\lambda_1,i}$$

Hyperparameter optimisation: cross-validation

For each set of hyper parameters, we determine cross-validation score
This allows us to compare models



Summary of model training

- Split the data at first place to train-test sets, keep some for model evaluation
  - ☑ The model must not see test data before/during training
- Optimise loss function to find model parameters *w*
- Error on train set does not show anything about model generalisability
- Use cross validation to assign hyperparameters

Ridge regression:

Linear least squares with L2 regularisation

$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Matrix form:

$$\hat{y}_{ML}(X) = Xw = \begin{bmatrix} 1 & x_1^1 & \dots & x_d^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^m & \dots & x_d^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Loss function: $\mathscr{L} = \|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2 = \|Xw - y\|_2^2 + \lambda \|w\|_2^2$

Ridge regression: closed-form and linear kernel

Closed-form solution for ridge regression:

$$w = \arg\min_w \mathscr{L} = \arg\min_w \left(\|\hat{y}_{ML} - y\|_2^2 + \lambda \|w\|_2^2 \right)$$

$$\frac{\partial \mathscr{L}}{\partial w} = 0 \implies w = (X^T X + \lambda I)^{-1} X^T y$$

also, you can write it as (see exercise notes): $w = X^T (X X^T + \lambda I)^{-1} y$

$$a := (X X^T + \lambda I)^{-1} y \implies w = X^T a$$

$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} X^T a = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} \begin{bmatrix} x_1^1 & \dots & x_1^m \\ \vdots & \ddots & \vdots \\ x_d^m & \dots & x_d^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}$$

$$= \sum_{i=1}^{m} x^* x_i^T a_i = \sum_{i=1}^{m} K(x^*, x_i)\ a_i$$

Linear kernel

Kernel Ridge Regression (KRR):

It'e equivalent to previous solution, except we changed the basis:

$$\hat{y}_{ML}(X) = Ka\,, a = (K + \lambda I)^{-1} y$$

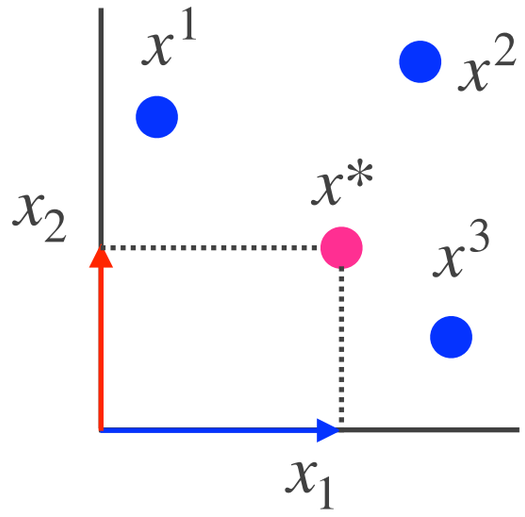
$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} K(x^*, x_i)\ \ a_i$$

What does change of basis mean in linear algebra:

For simplicity, let's look at a case with 2 variables and 3 training data

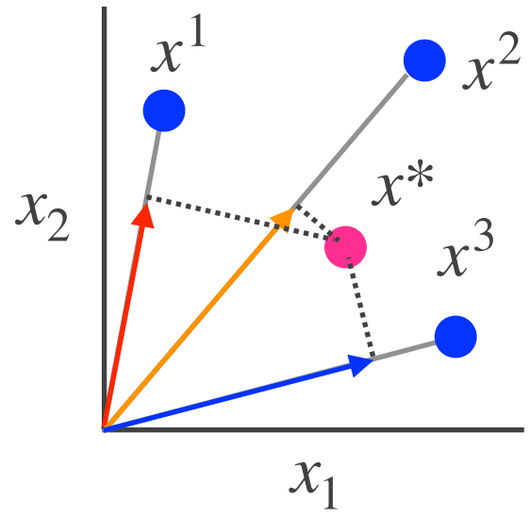
$$\hat{y}_{ML}(x^*) = x^* w = \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = x_1^* w_1 + x_1^* w_2$$

$$= \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$



$$\hat{y}_{ML}(x^*) = \begin{bmatrix} 1 & x_1^* & \dots & x_d^* \end{bmatrix} X^T a$$

$$= \begin{bmatrix} x_1^* & x_2^* \end{bmatrix} \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_1^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_3 \end{bmatrix}$$



Kernel Ridge Regression (KRR): nonlinearity

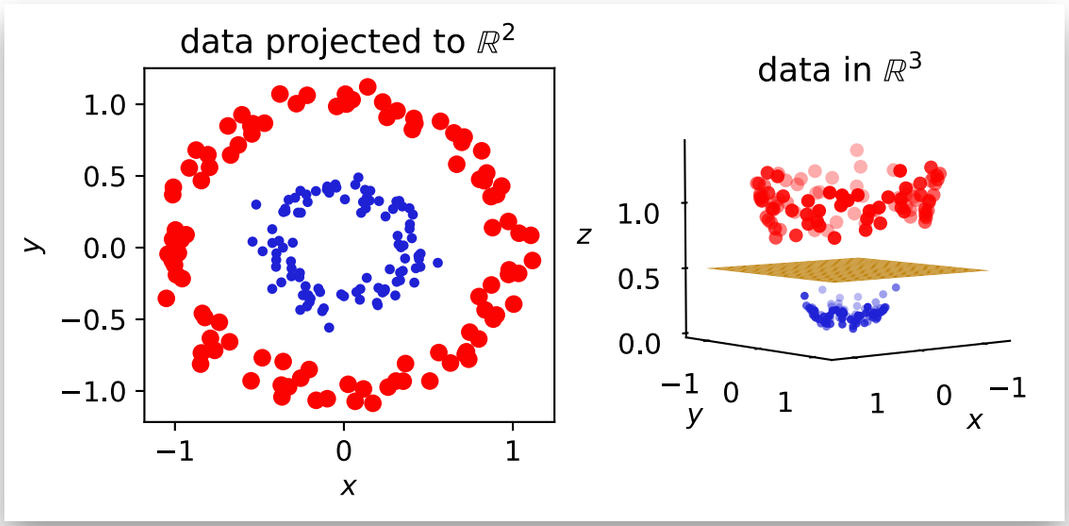
Include nonlinearity, for example a quadratic model:

$$\hat{y}_{ML}(X) = \phi(X)w$$

$$= \begin{bmatrix} 1 & x_1^1 & x_2^1 & (x_1^1)^2 & (x_2^1)^2 & x_1^1 x_2^1 \\ \vdots & \vdots & & & & \\ 1 & x_1^m & x_2^m & (x_1^m)^2 & (x_2^m)^2 & x_1^m x_2^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}$$

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle = ((x^*)^T x + 1)^2$$

This makes the data that are not linearly separable in $\mathbb{R}^2$, linearly separable in higher dimensions, still computed efficiently



Kernel Ridge Regression (KRR): kernel trick

This allows us to use non-linear kernels efficiently:

$$\hat{y}_{ML}(X) = \phi(X)w$$

$$w = (\phi(X)^T\phi(X) + \lambda I)^{-1}\phi(X)^T y$$

$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} K(x^*, x_i) \quad a_i$$

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle$$

In the case of linear kernel:

$$K(x^*, x_i) = \langle x^*, x_i \rangle$$

In the case of quadratic kernel:

$$K(x^*, x_i) = \langle \phi(x^*), \phi(x_i) \rangle = ((x^*)^T x + 1)^2$$

Kernel Ridge Regression (KRR): Gaussian kernel

The Gaussian kernel is very popular in chemistry

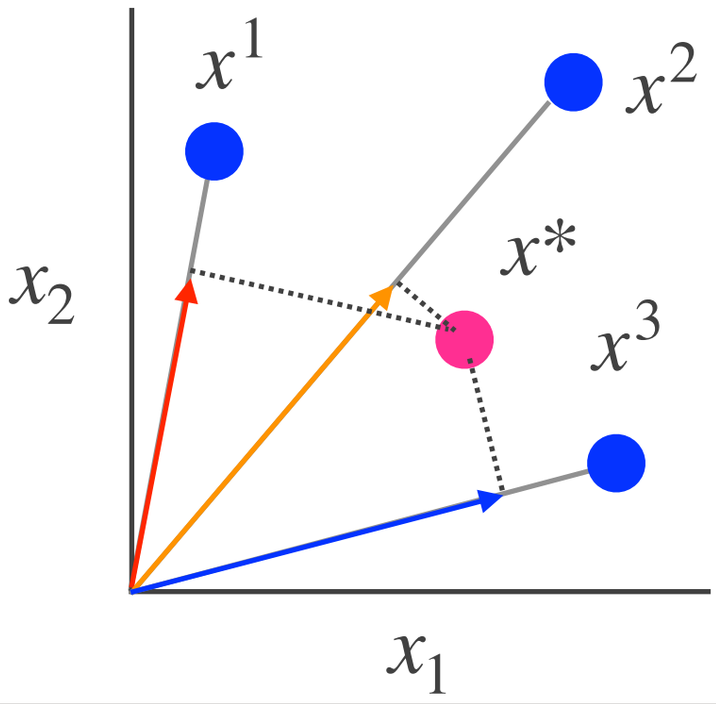
$$K(x^*, x) = \exp\left(-\gamma \|x^* - x\|_2^2\right)$$

This kernel provides an intuitive notion of similarity, i.e. distance in feature space:

$$\hat{y}_{ML}(x^*) = \sum_{i=1}^{m} a_i K(x^*, x^i)$$

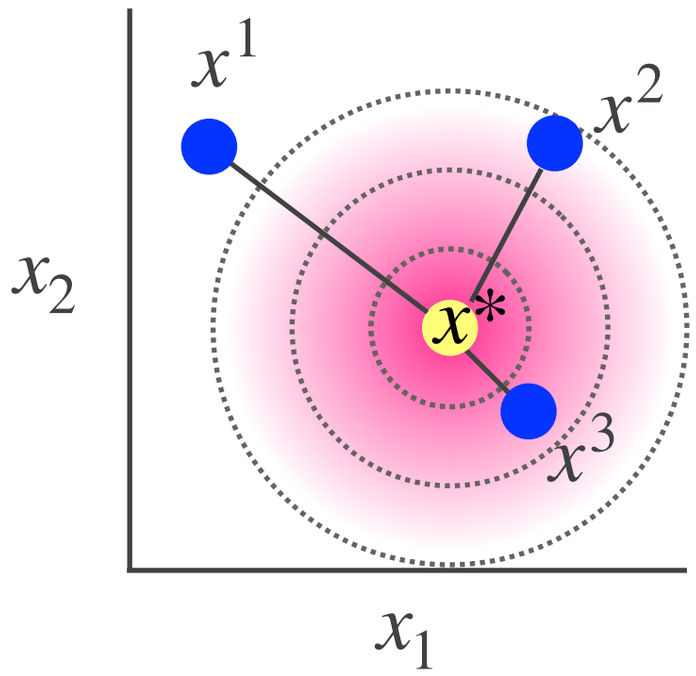
Linear basis

$K(x^*, x_i) = \langle x^*, x_i \rangle$



Radial basis

$K(x^*, x_i) = \exp\left(-\gamma \|x^* - x_i\|_2^2\right)$



Summary

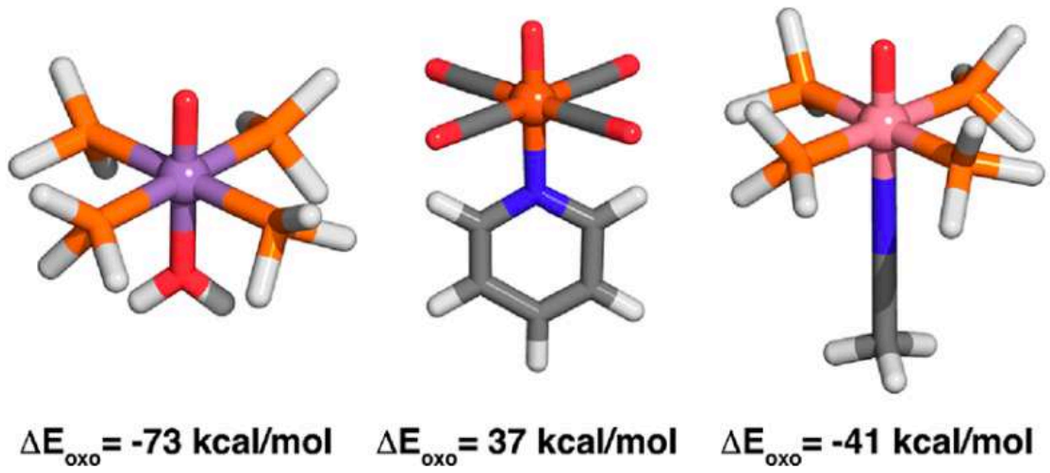
- Kernel and linear models
- Kernel trick
- Similarity and Gaussian kernel
- One should do CV to obtain the hyperparameters of kernel

Interpreting the model

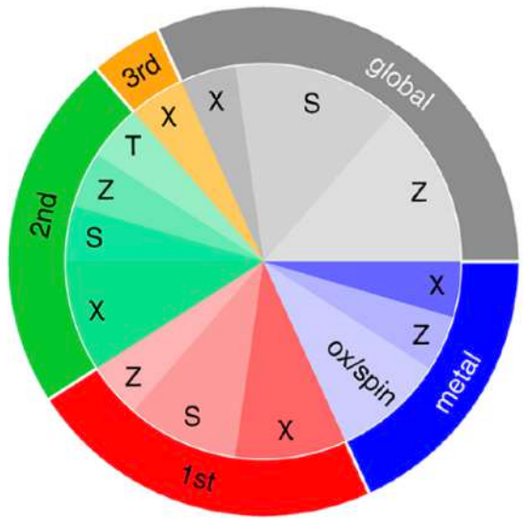
We would like to know what drives a phenomena (curiosity)

More practical:
Material design rules

Single metal catalyst:
variation in metal and ligands



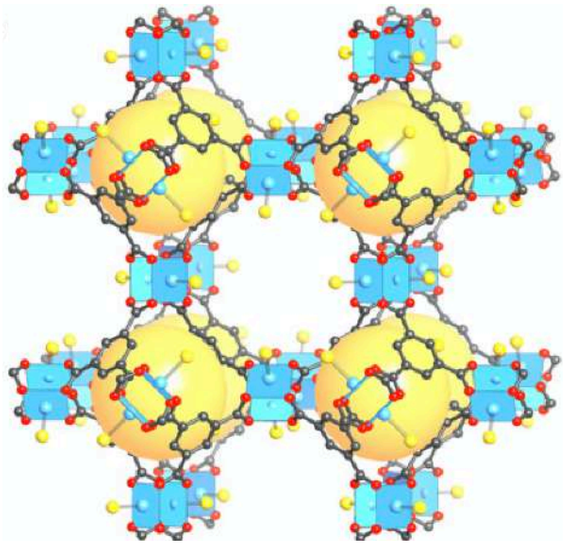
Importance of variables can help to understand which factors are important, e.g., type of atomic properties and global vs local



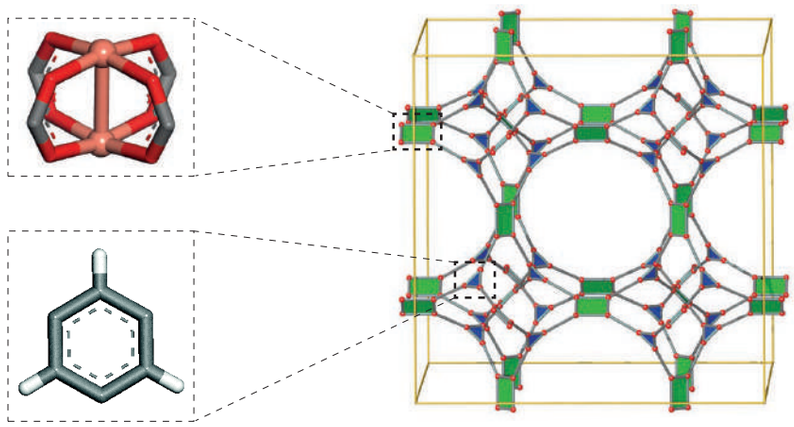
Nandy, Aditya, et al. ACS Catalysis 9.9 (2019): 8243-8255.

Interpreting the model (||)

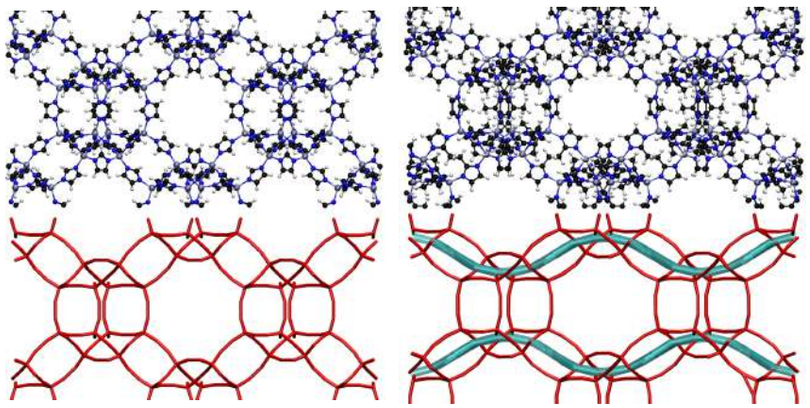
Make better models / representations
example: the case of mechanical properties of MOFs



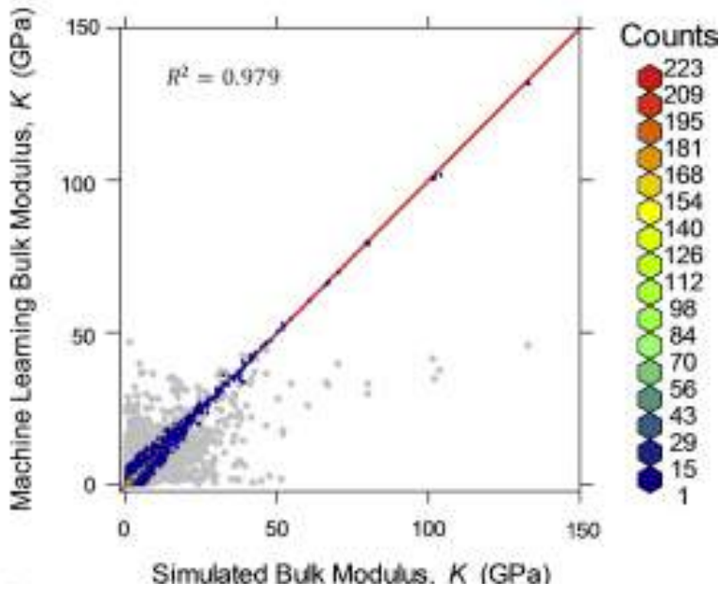
Simplified representation



It was shown that the underlying net is the most important factor for mechanical stability



Include net in representation



Moosavi, et al. ACS Central Science (2018)
Moghadam et al. Matter (2019)

Permutation importance

0. Build the model

$$X = \begin{pmatrix} 1 & x^{1,1} & x^{1,1} & \dots & x^{1,n} \\ 1 & x^{2,1} & x^{2,1} & \dots & x^{2,n} \\ 1 & \vdots & \ddots & & \vdots \\ 1 & x^{m,1} & x^{m,1} & \dots & x^{m,n} \end{pmatrix} \quad y = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{pmatrix} \qquad \hat{y}_{ML} = f(X), \mathscr{L}(y, \hat{y}_{ML})$$

1. Estimate the model error: $e^{\text{orig}} = \mathscr{L}(y, \hat{y}_{ML})$

2. For each feature j=1,..,n:
   - Generate permuted feature matrix for feature (j):

Shuffle

$$X_j^{\text{perm}} = \begin{pmatrix} 1 & x^{1,1} & \dots & x^{1,j} & \dots & x^{1,n} \\ 1 & x^{2,1} & \dots & x^{2,j} & \dots & x^{2,n} \\ 1 & \vdots & \dots & \vdots & \ddots & \vdots \\ 1 & x^{m,1} & \dots & x^{m,j} & \dots & x^{m,n} \end{pmatrix}$$

   - Estimate the error for the permuted feature matrix: $e_j^{\text{perm}} = \mathscr{L}(y, f(x_j^{\text{perm}}))$

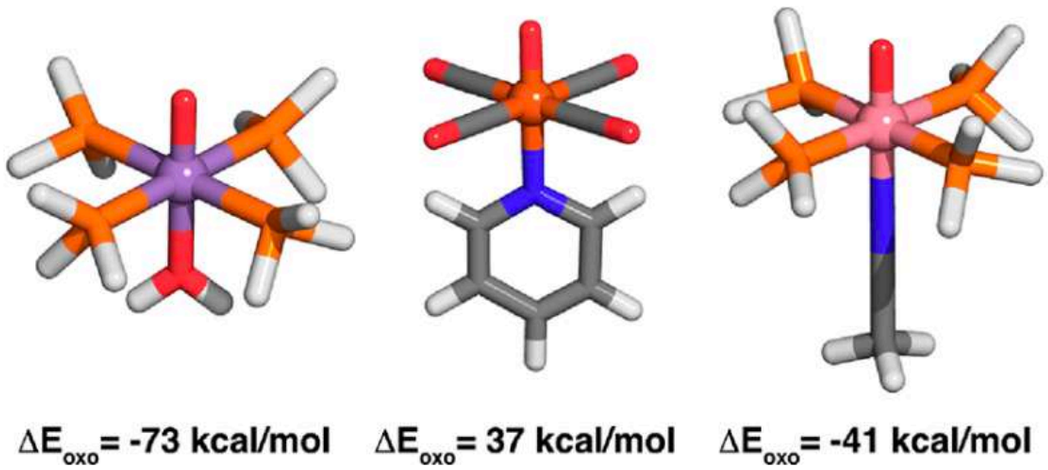
   - Estimate the error for the permuted feature matrix

$$\text{FI}_j = e_j^{\text{perm}} / e_j^{\text{orig}} \text{ or } e_j^{\text{perm}} - e_j^{\text{orig}}$$

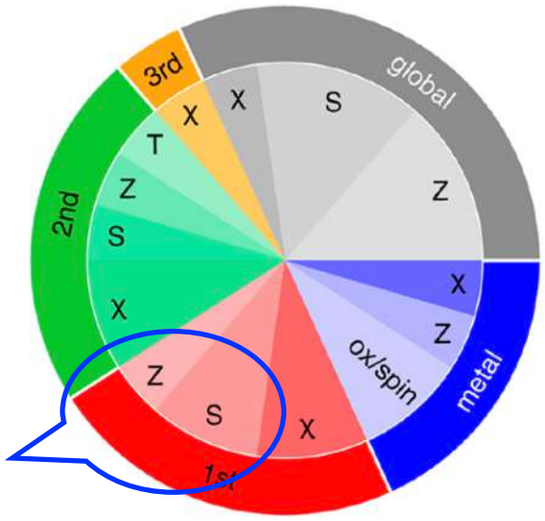
Permutation importance

- What happens if we have correlated features?

Permutation might make unphysical test cases



Split the importance between the correlated features



S = covalent radii
Z = nuclear charge

Summary of model interpretation

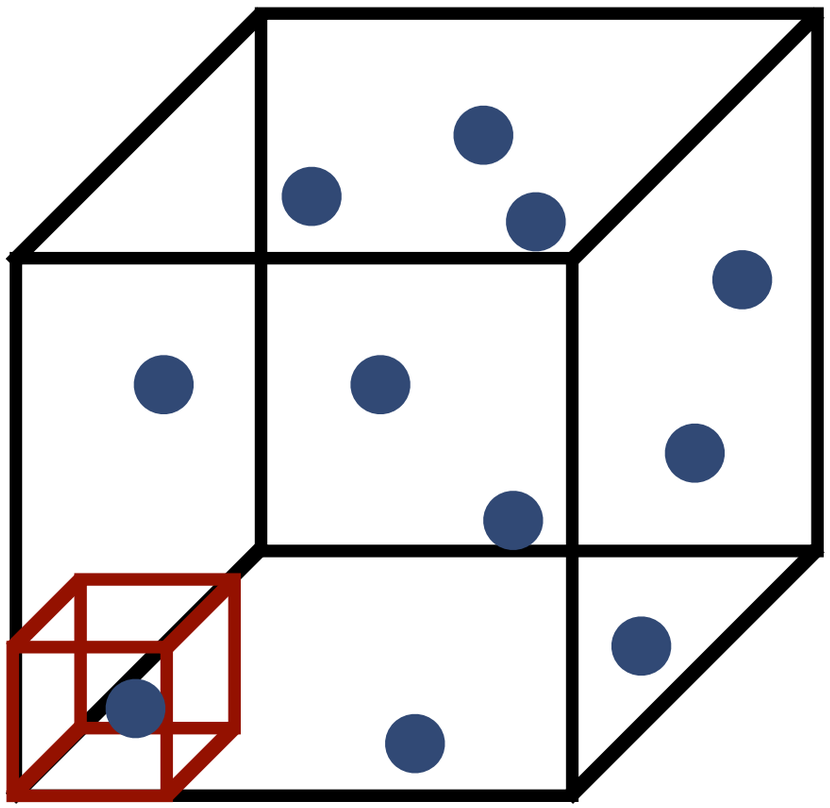
Feature importance is a way to interpret the model

- Get chemical insight
- Make better models

Permutation importance is a way to get the value of features in model predictions but one needs to be cautious to not over-interpret these numbers, e.g., when the features are correlated

The Curse of Dimensionality - No Locality in High Dimensions

$$\text{edgelength} = \text{fraction of space}^{1/\text{dimensionality}}$$



unit cube

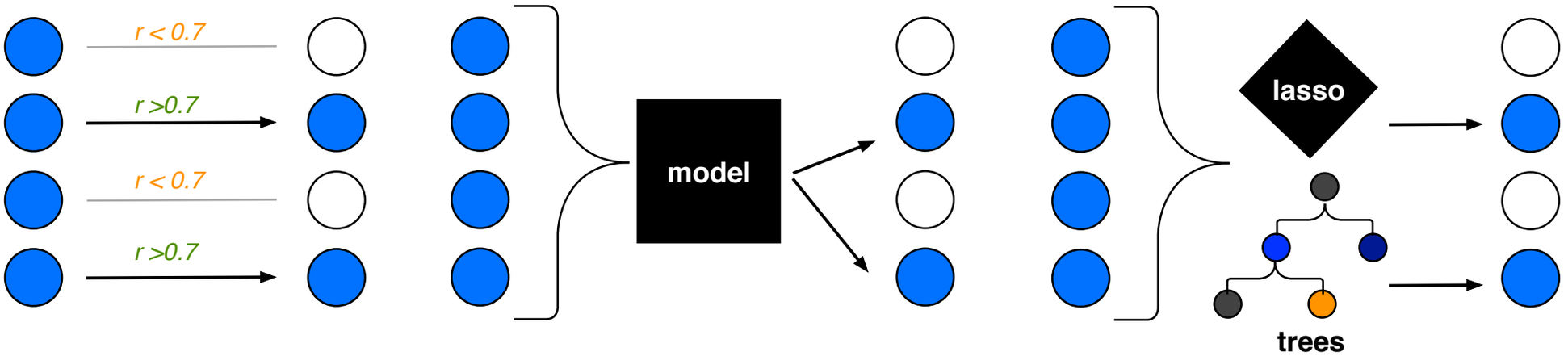
Methods that are based on similarity (*k*NN/KRR) might fail in high dimensional spaces!

We want to **reduce the dimensionality** of our feature matrix!

Domingos, P. A Few Useful Things to Know about Machine Learning. *Communications of the ACM* **2012**, *55* (10), 78..

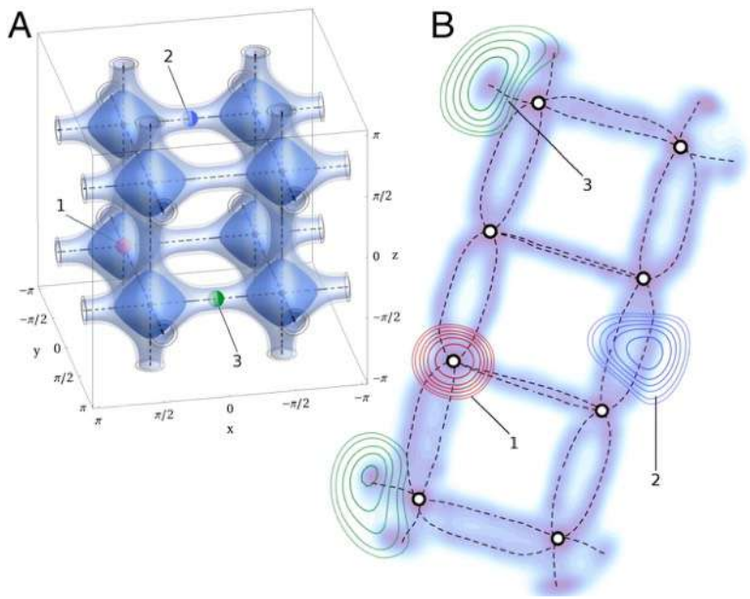
Feature Projection and Feature Selection

Reduce dimensionality of feature space by feature selection (compression)



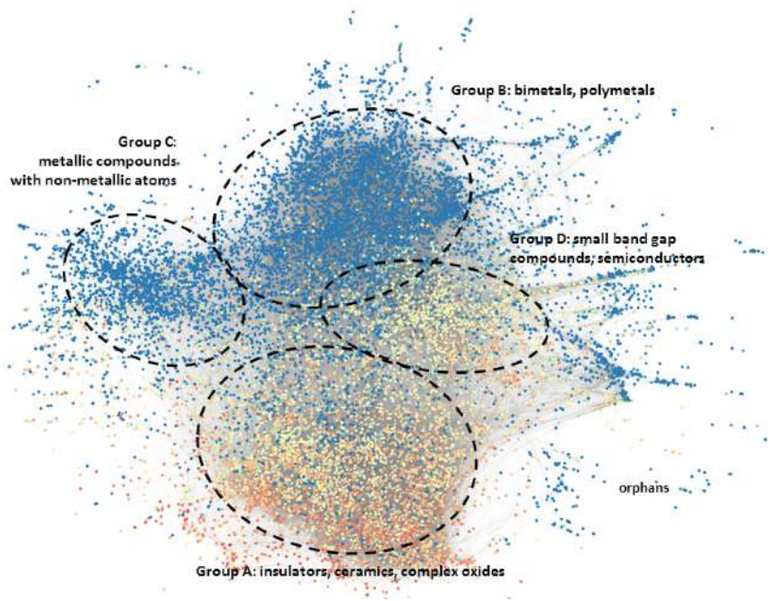
a Univariate filters. **b** Wrapper methods. **c** Shrinkage or direct.

Reduce size of feature space by dimensionality reduction (feature projection)



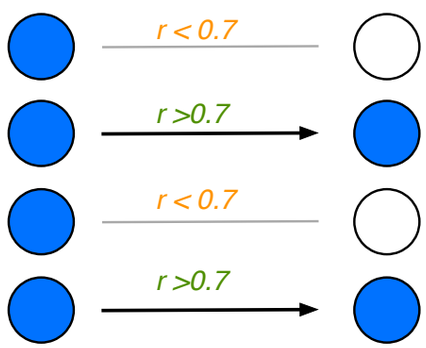
Janet, J. P.; Kulik, H. J. *J. Phys. Chem. A* **2017**, *121* (46), 8939–8954.

Visualize data and Materials Cartography



Tribello, G. A.; Ceriotti, M.; Parrinello, M. *PNAS* **2012**, *109* (14), 5196–5201.

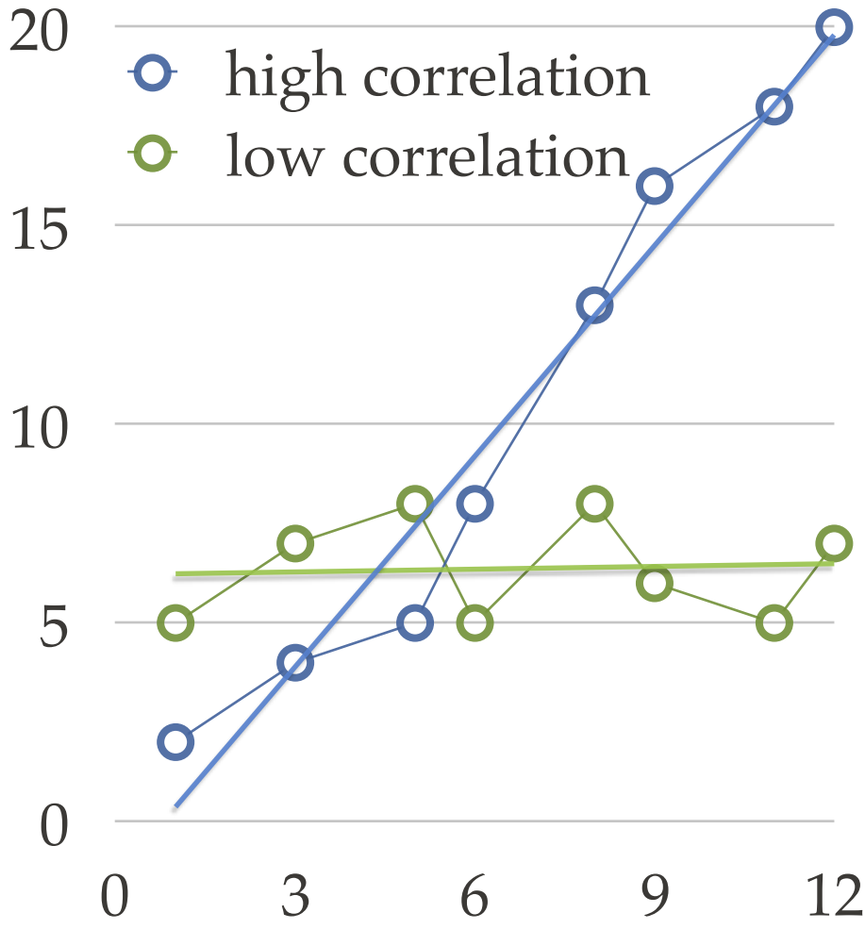
Feature Selection: Filter Methods



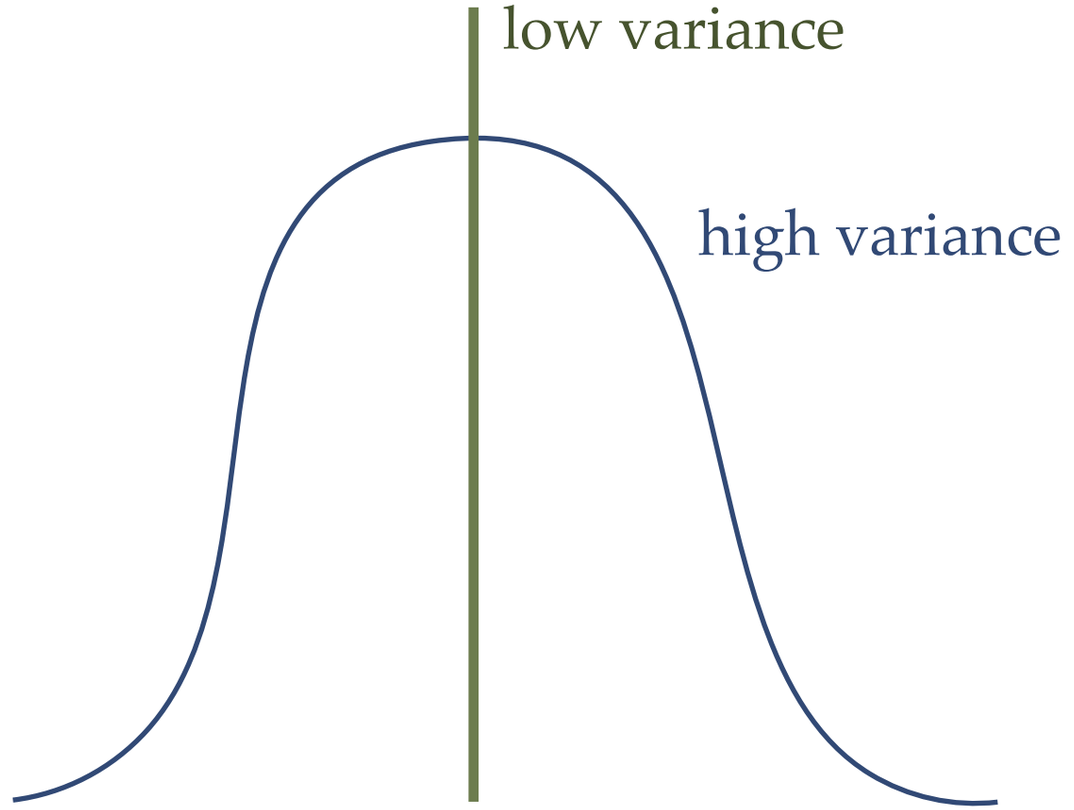
Easy and cheap

Interaction effects are not considered

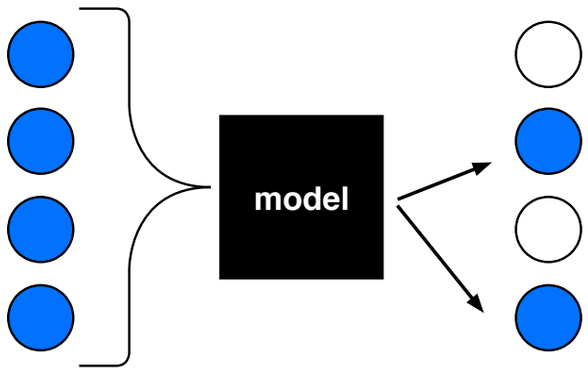
correlation threshold



variance threshold



Feature Selection: Wrapper Methods



Uses a good surrogate of the real objective

Expensive

For example recursive feature addition or elimination

$$\begin{pmatrix} x_1^1 & x_1^2 & x_1^3 & \dots & x_1^p \\ x_2^1 & x_2^2 & x_2^3 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & x_n^3 & \dots & x_n^p \end{pmatrix}$$

full design matrix

subset generation

$$\begin{pmatrix} x_1^1 & x_1^2 & x_1^3 \\ x_2^1 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & x_n^3 \end{pmatrix}$$

subset evaluation

stopping criterion

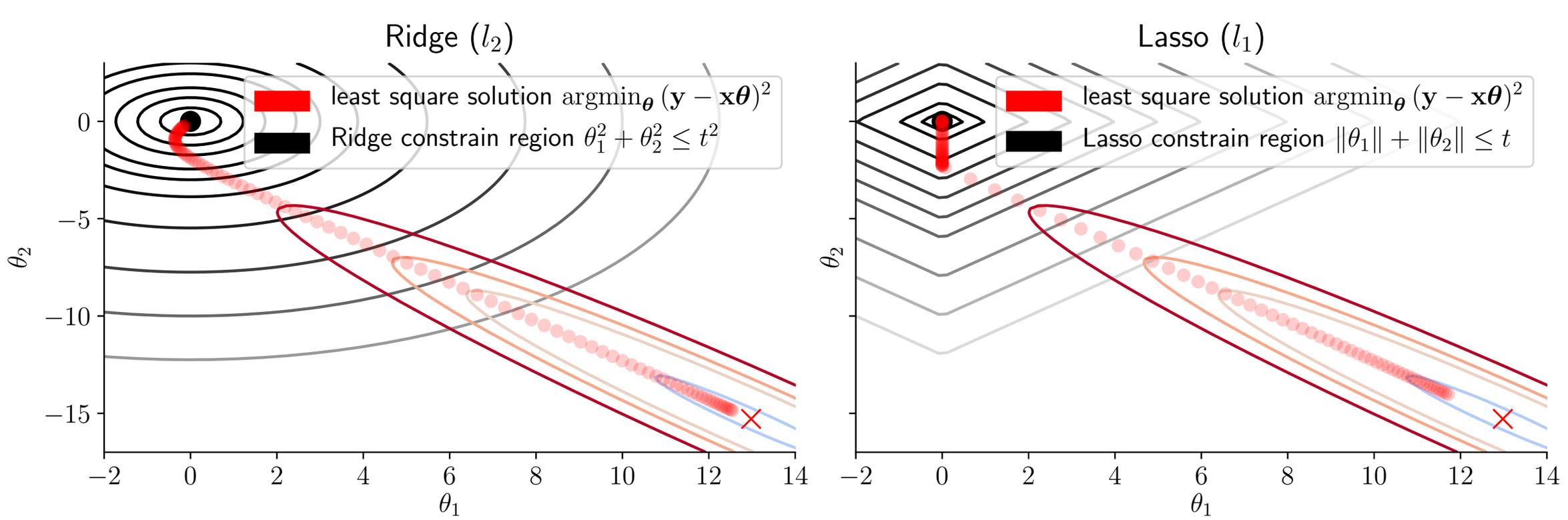
not fulfilled

Feature Selection: Just Relax Best Subset Selection

The basic problem: Best subset selection

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|^2 \text{ subject to } \quad \|\beta\|_0 \leq k \qquad \|\beta\|_0 = \sum_{j=1}^{p} 1\{\beta_j \neq 0\}$$

But this is our hard problem (NP hard) …
… hence we relax the constraint to have a problem that is convex
… the Lasso gives use sparsity as the most feasible approximation to l_0



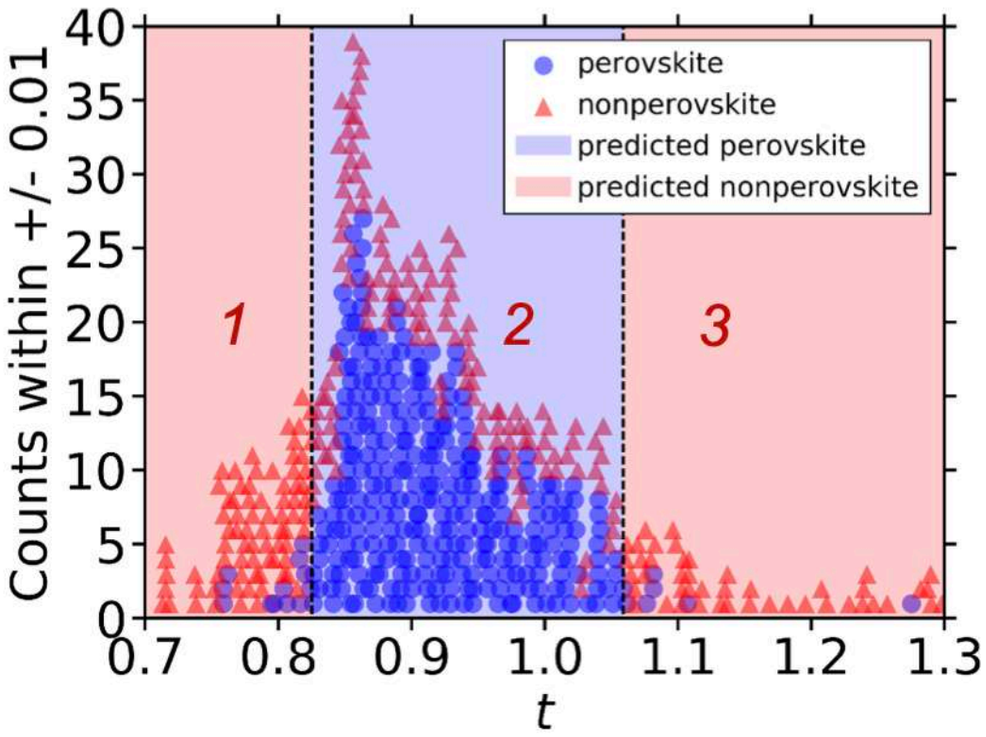
Hastie, T.; Tibshirani, R.; Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*; Monographs on statistics and applied probability; CRC Press, Taylor & Francis Group: Boca Raton, 2015.

Lasso in Practice: Finding New Tolerance Factors For Perovskites (Developing Causal Models)

primary features $\xrightarrow{\hat{R}}$ **billions** of feature candidates $\xrightarrow{\text{(SI) +Lasso}}$ subset of features

primary features $= \{r_A, r_B, n_a, n_b, \ldots\}$ $\quad \hat{R} = \{+, -, \cdot, \exp, \lg, ^{-1}, ^{2}, ^{3}, \sqrt{}, \|\cdot\|\}$

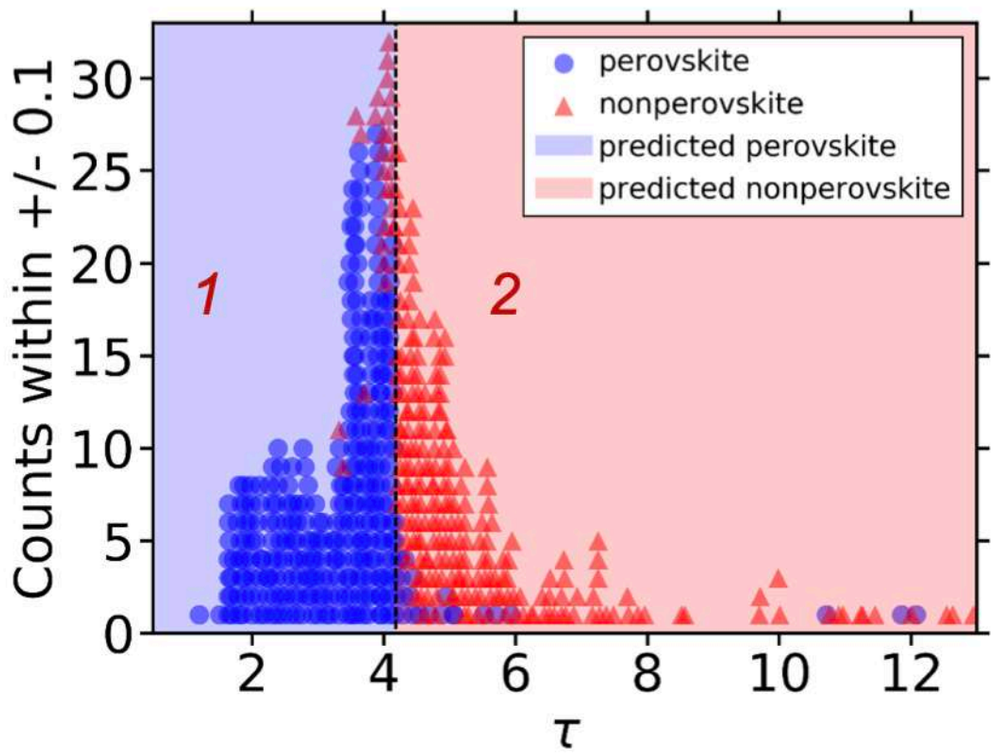
$$t = \frac{r_A + r_X}{\sqrt{2}(r_B + r_X)}$$



$0.825 < t < 1.059 \rightarrow$ perovskite

74% accuracy

$$\tau = \frac{r_X}{r_B} - n_A\left(n_A - \frac{r_A/r_B}{ln\, r_A/r_B}\right)$$



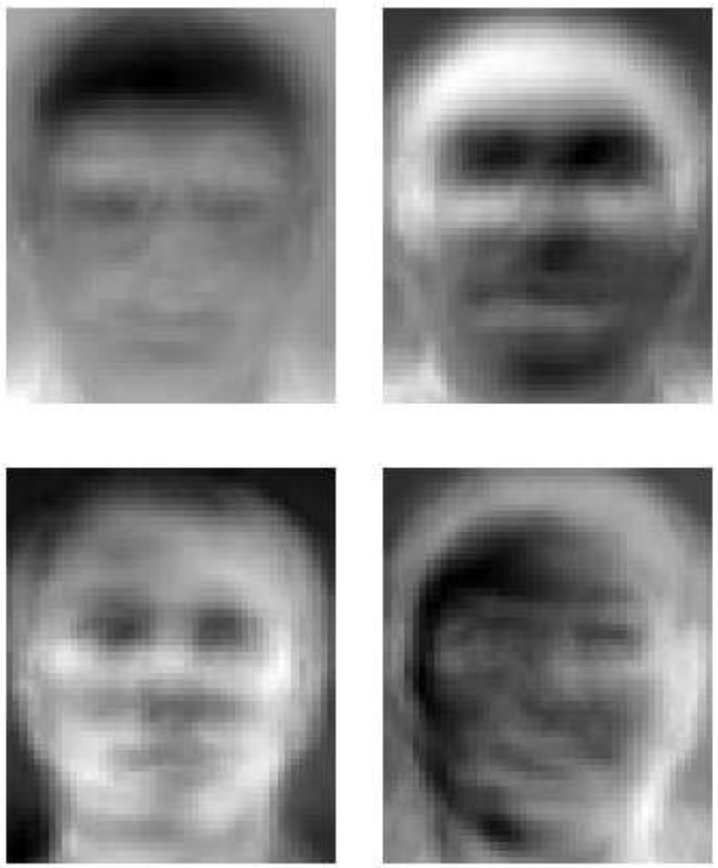
$\tau < 4.18 \rightarrow$ perovskite

92% accuracy

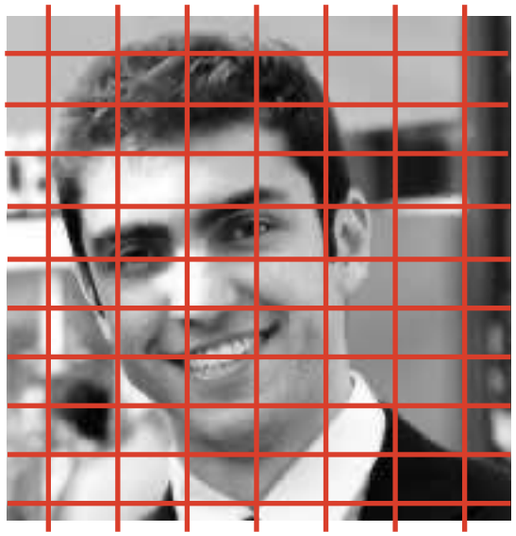
Ghiringhelli, *et al*. *Phys*. *Rev*. *Lett*. **2015**, *114* (10), 105503.
Ouyang, R.; *et al*. *Phys*. *Rev*. *Materials* **2018**, *2* (8), 083802.
Bartel, C.; *et al*., M. *Sci*. *Adv*. **2019**, *5* (2), eaav0693.

Feature Projection: Projecting High-Dimensional Data

Eigenfaces: Principal Component Analysis on face images to get "basis vectors" of face image space



Linear combination of basis vectors



One image is one column

faces

many faces

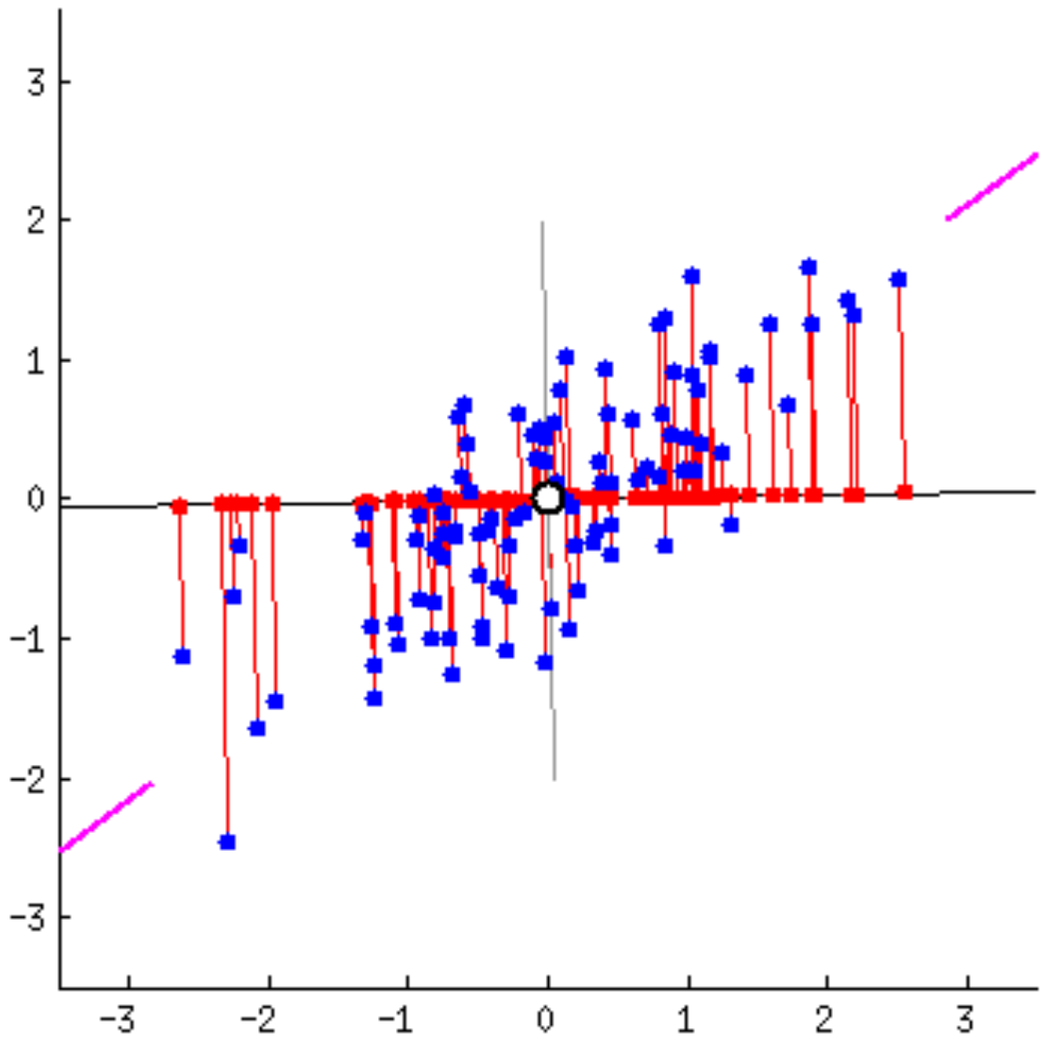
pixels

PCA

Find basis vectors

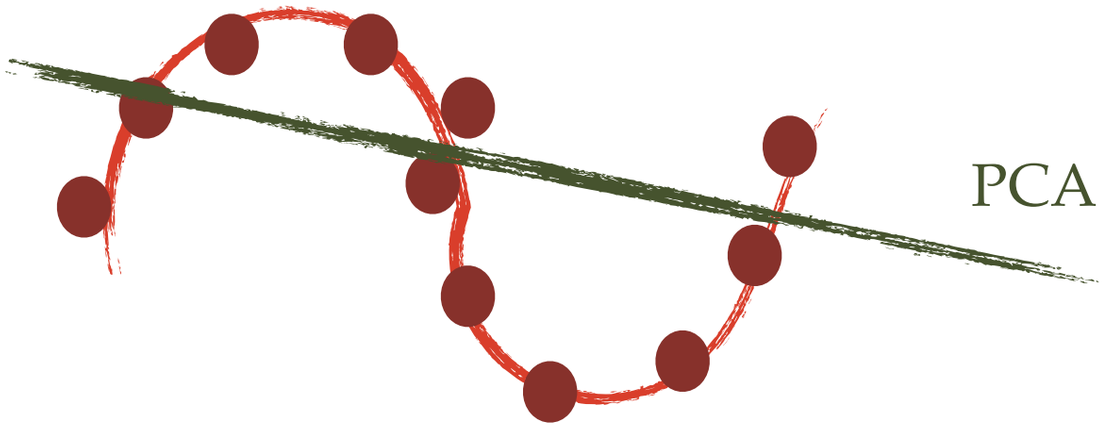
Navarrete, P.; Ruiz-Del-Solar, J. *Int. J. Patt. Recogn. Artif. Intell.* **2002**, *16* (07), 817–830.

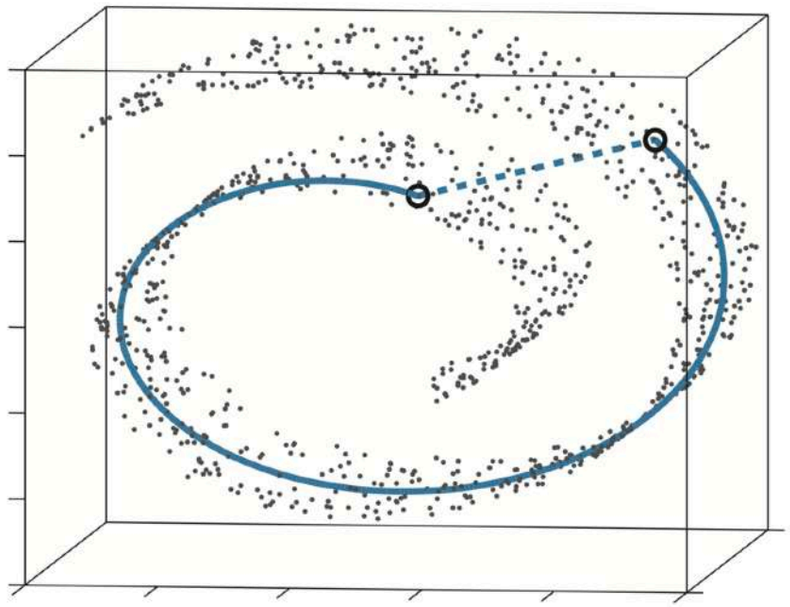
Feature Projection: Intuition for PCA



Caveats With PCA

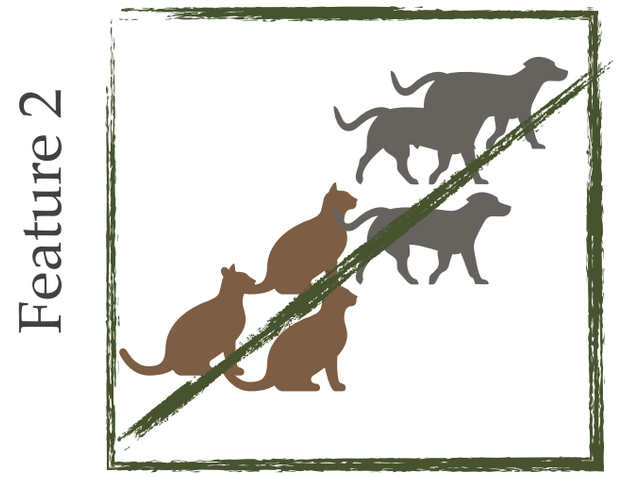
Non-linearity

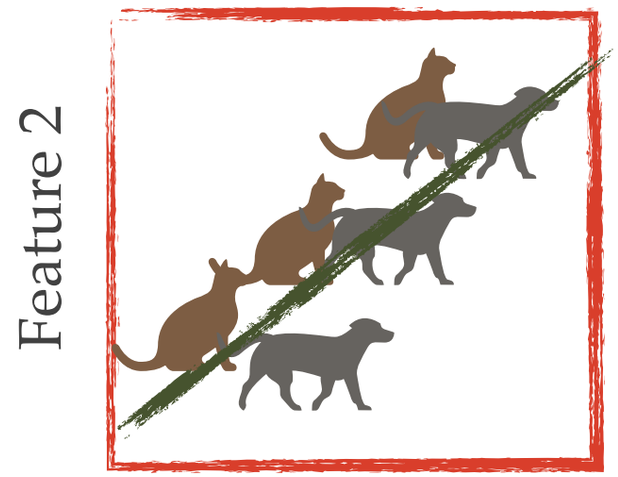




- Data is linearly uncorrelated
- But there is still a non-linear dependence

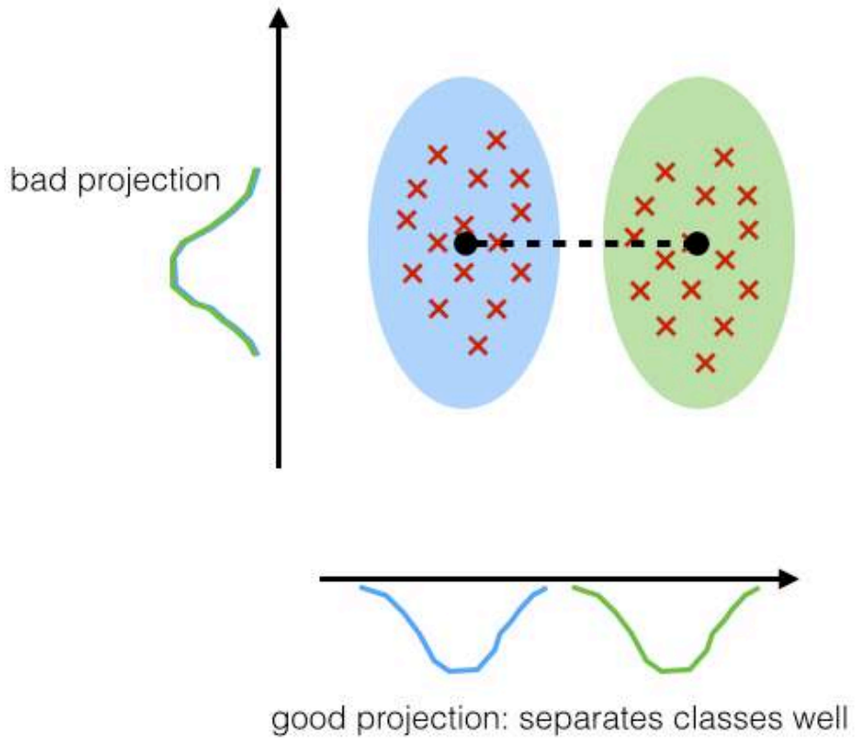
Higher variance feature is not more discriminative





Other Members of the Dimensionality Reduction Zoo

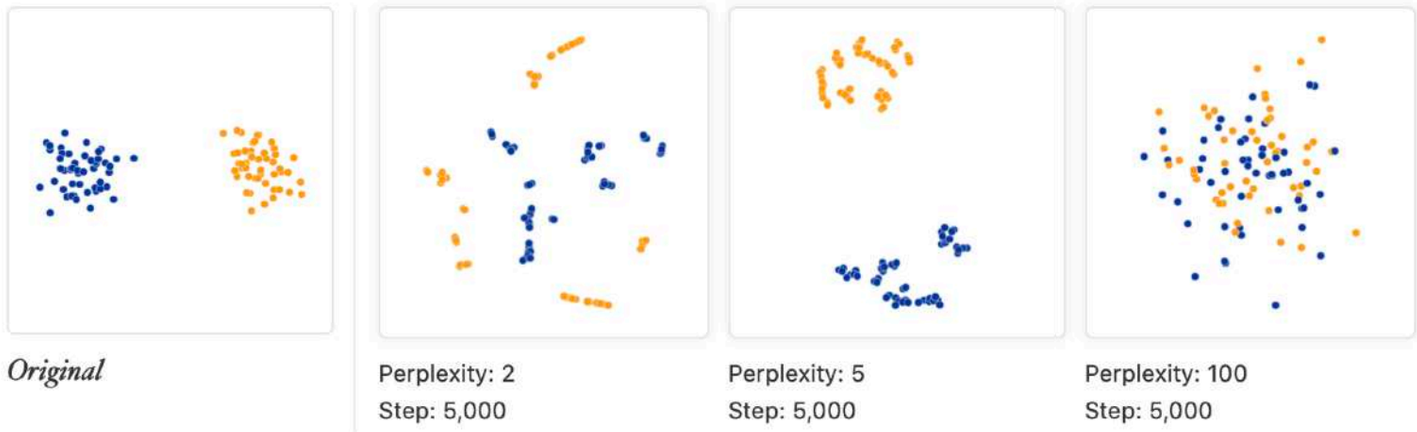
Linear Discriminant Analysis (LDA, Supervised Technique)



- math like for PCA
- maximizing component axes for class separation

t-distributed stochastic neighbor embedding (*t*-SNE)

- Non-linear
- Conditional probabilities that represent similarities
- Sensitive to perplexity (number of close neighbors)



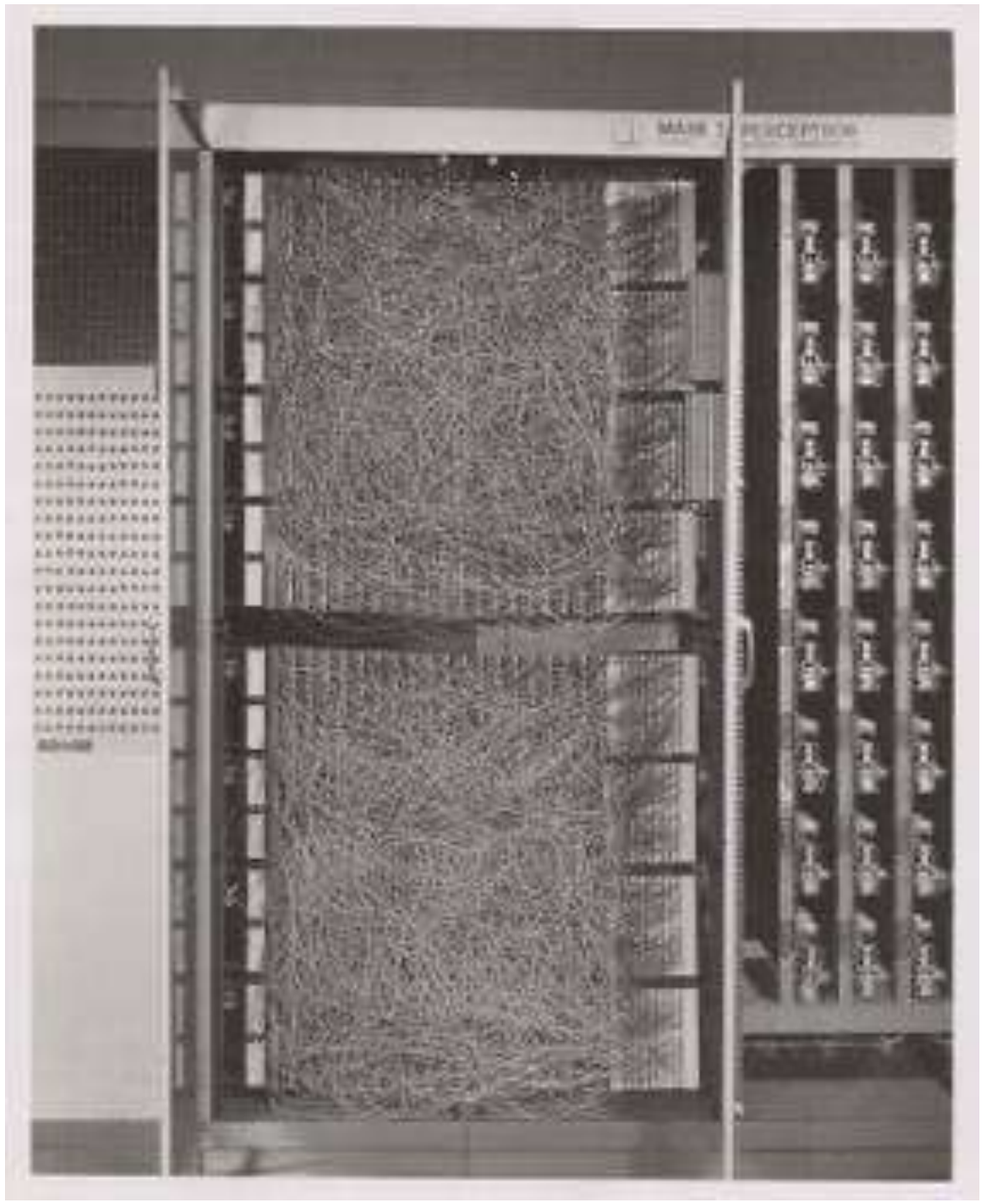
Wattenberg, et al., "How to Use t-SNE Effectively", *Distill*, **2016**.

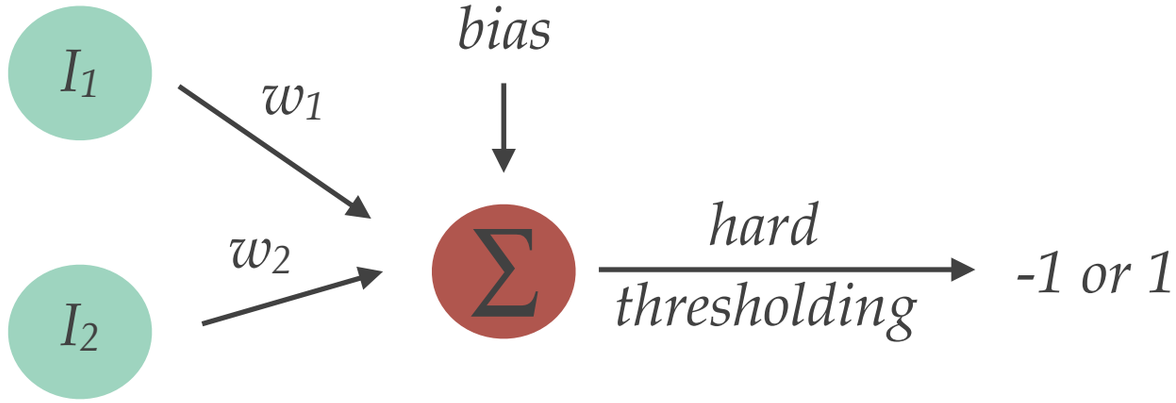
Talk by Laurens van der Maaten: https://bit.ly/2RaFJIw

Neural Networks: Perceptron (*"may eventually be able to learn, make decisions, and translate languages"*)

The Perceptron (Rosenblatt (1957))

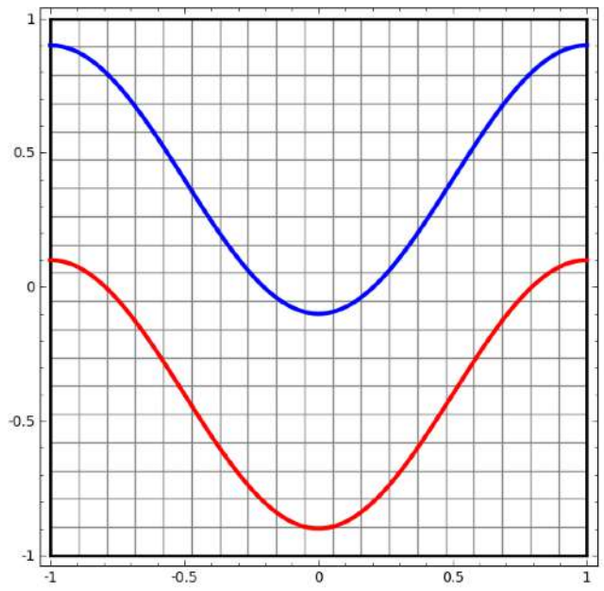
Mark 1 built for image recognition

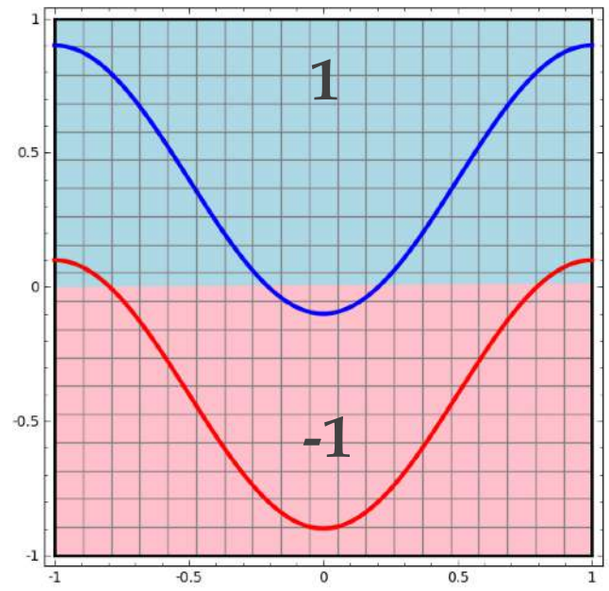




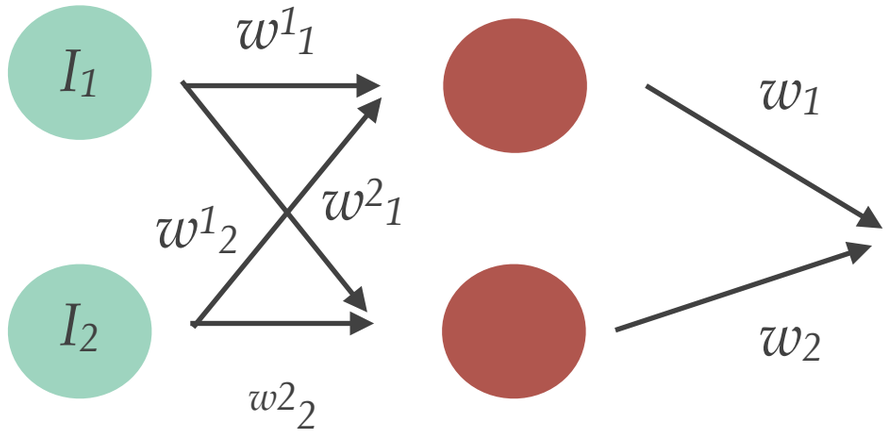
Try to classify which points belong to which curve.

Decision boundary: $w_1x_1 + w_2x_2 + b = 0$





Multilayer Perceptron: Representation Learning



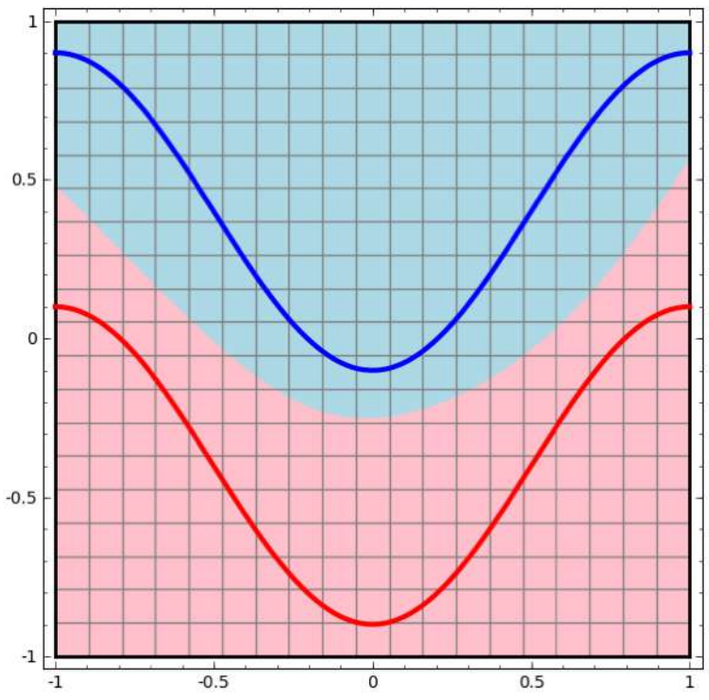
output

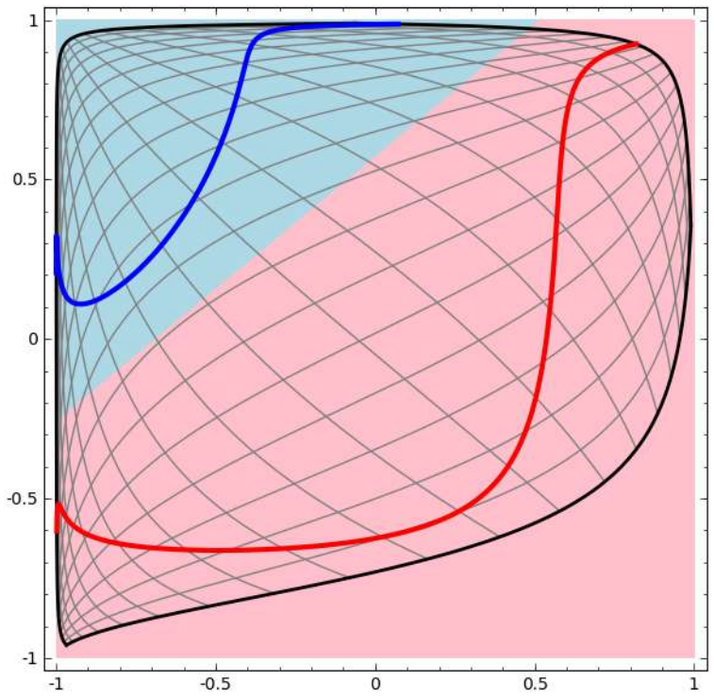
transform representation such that it is linearly separable

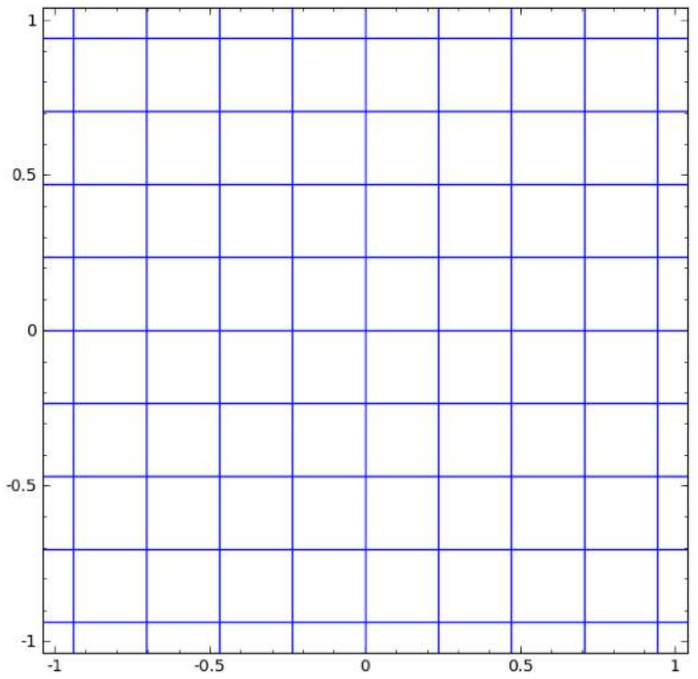
hidden layer, e.g. tanh

in tanh(**W**x+b):

- rotate (**W**)
- Translate (b)
- point-wise application of tanh



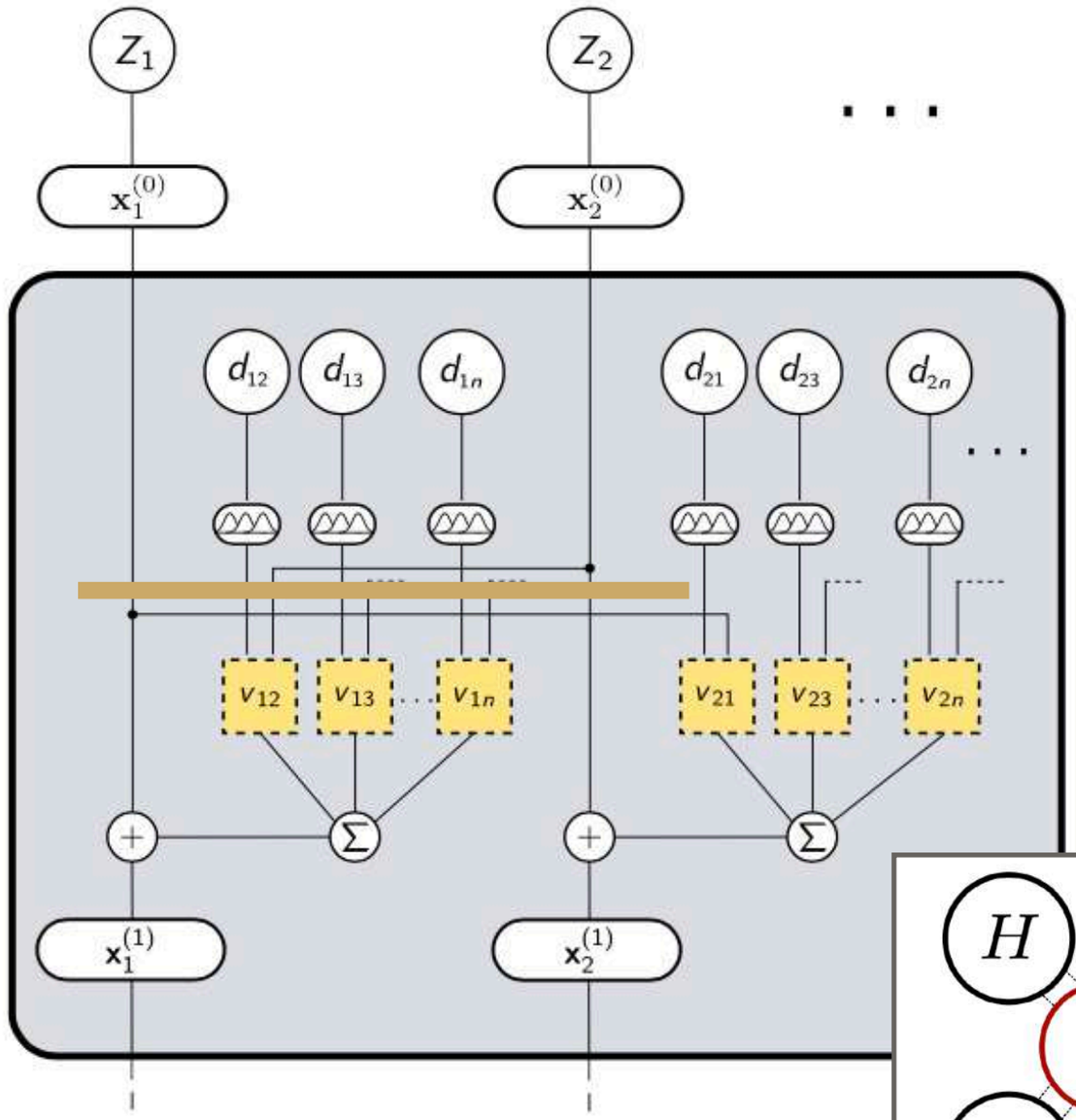




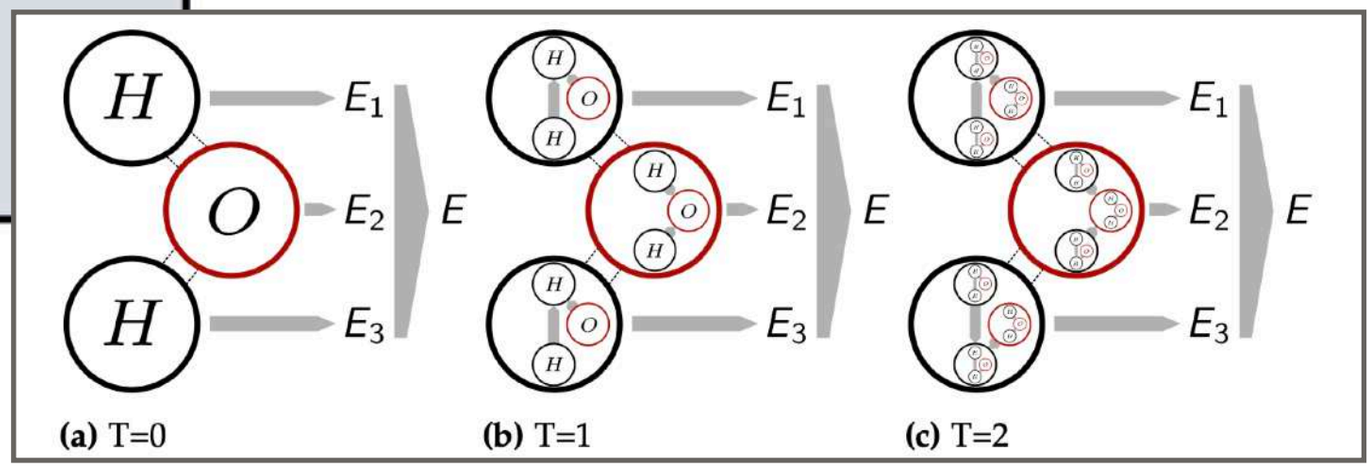
Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*. The MIT Press: Cambridge, Massachusetts, 2016.

LeCun, Y.; Bengio, Y.; Hinton, G. *Nature* **2015**, *521* (7553), 436–444.
http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Message Passing Neural Networks



- Do not use highly engineered features, like symmetry function, but directly ***Z*** and ***r.***
- Transferable across **Z**

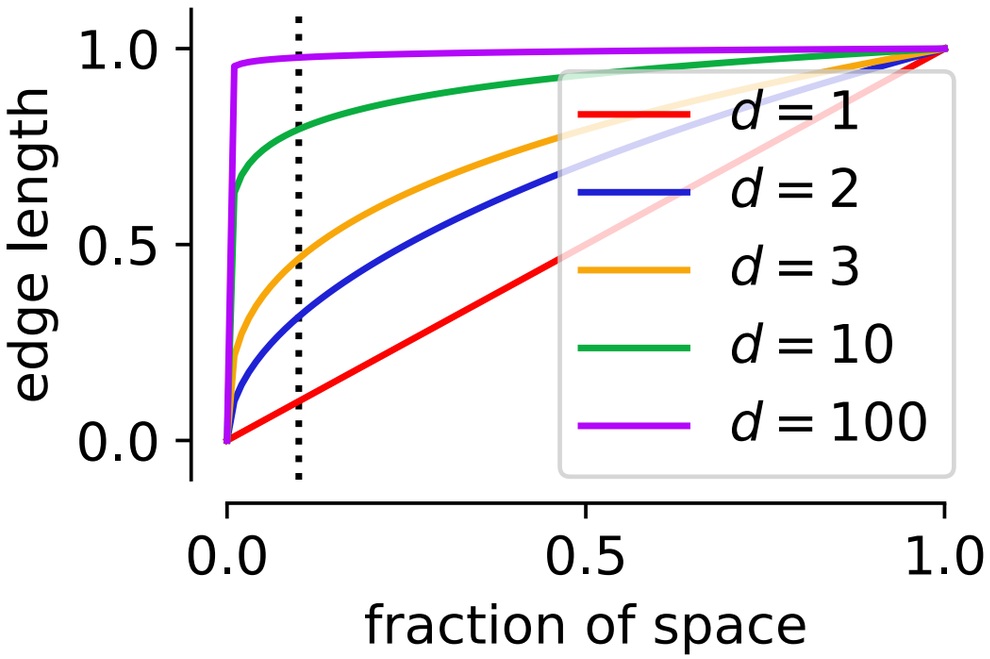


Schütt, K. Learning Representations of Atomistic Systems with Deep Neural Networks TU Berlin, 2018.

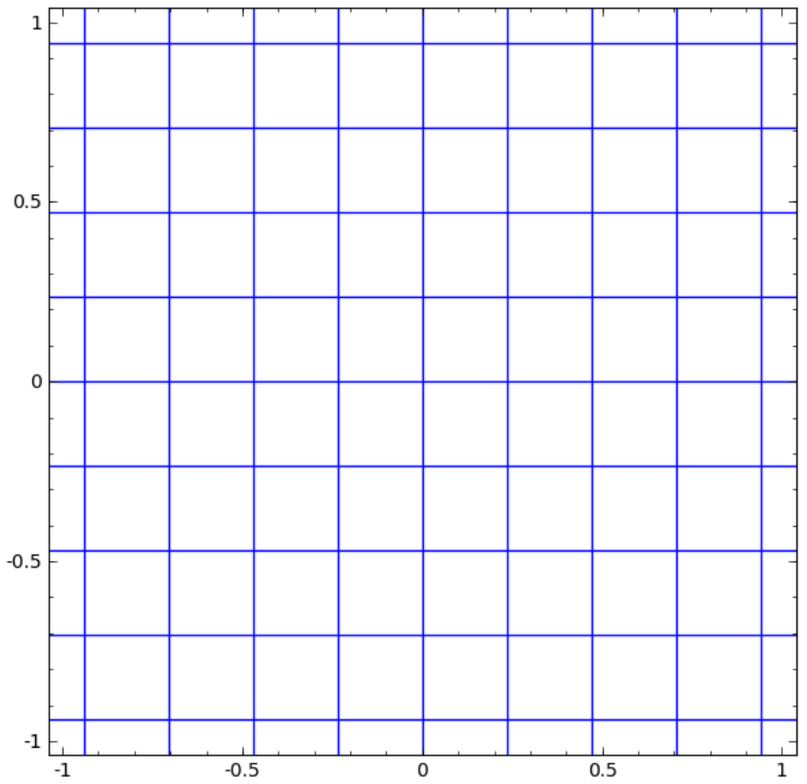
Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E.*arXiv:1704.01212 [cs]* **2017**.

Summary of Feature Engineering and Learning

- Curse of Dimensionality



- Neural Networks can do representation learning

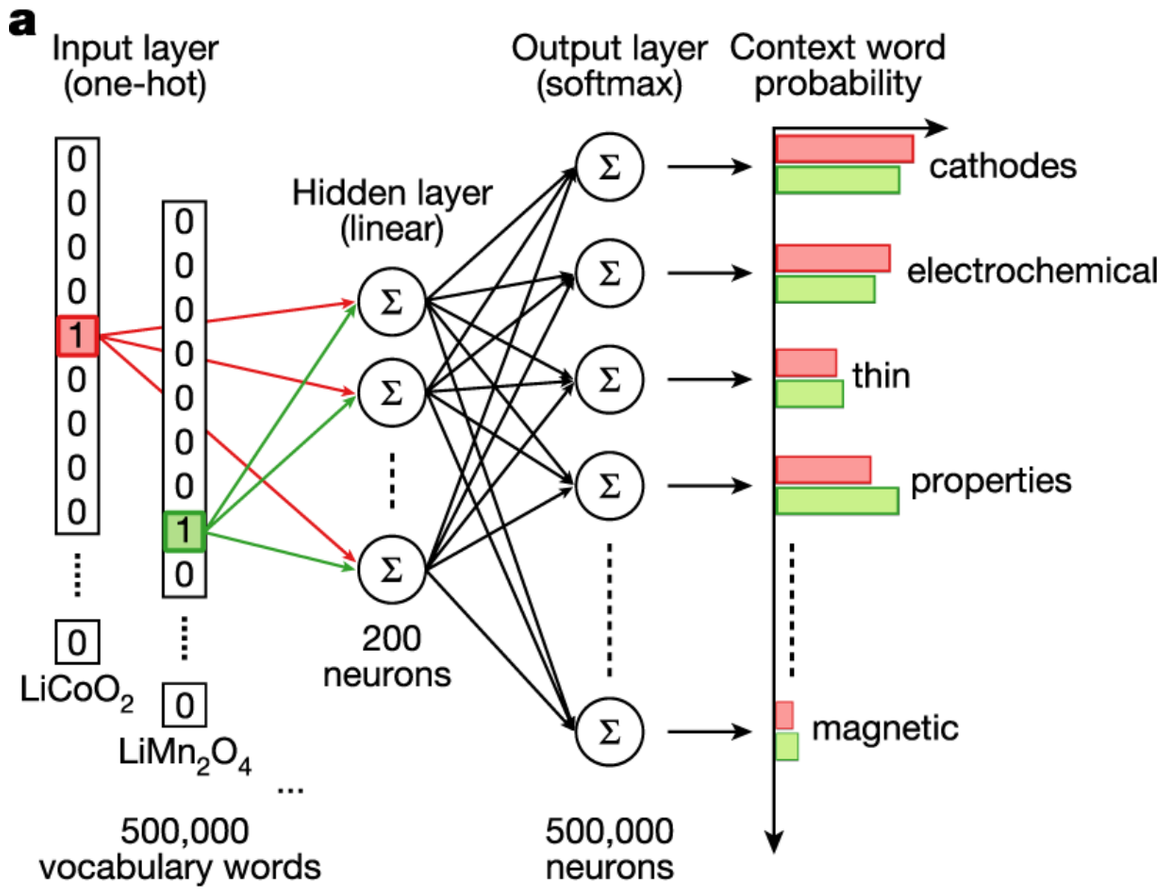


Now, (advanced) applications.

Word Embeddings: Learning on > 1 Million of Abstracts

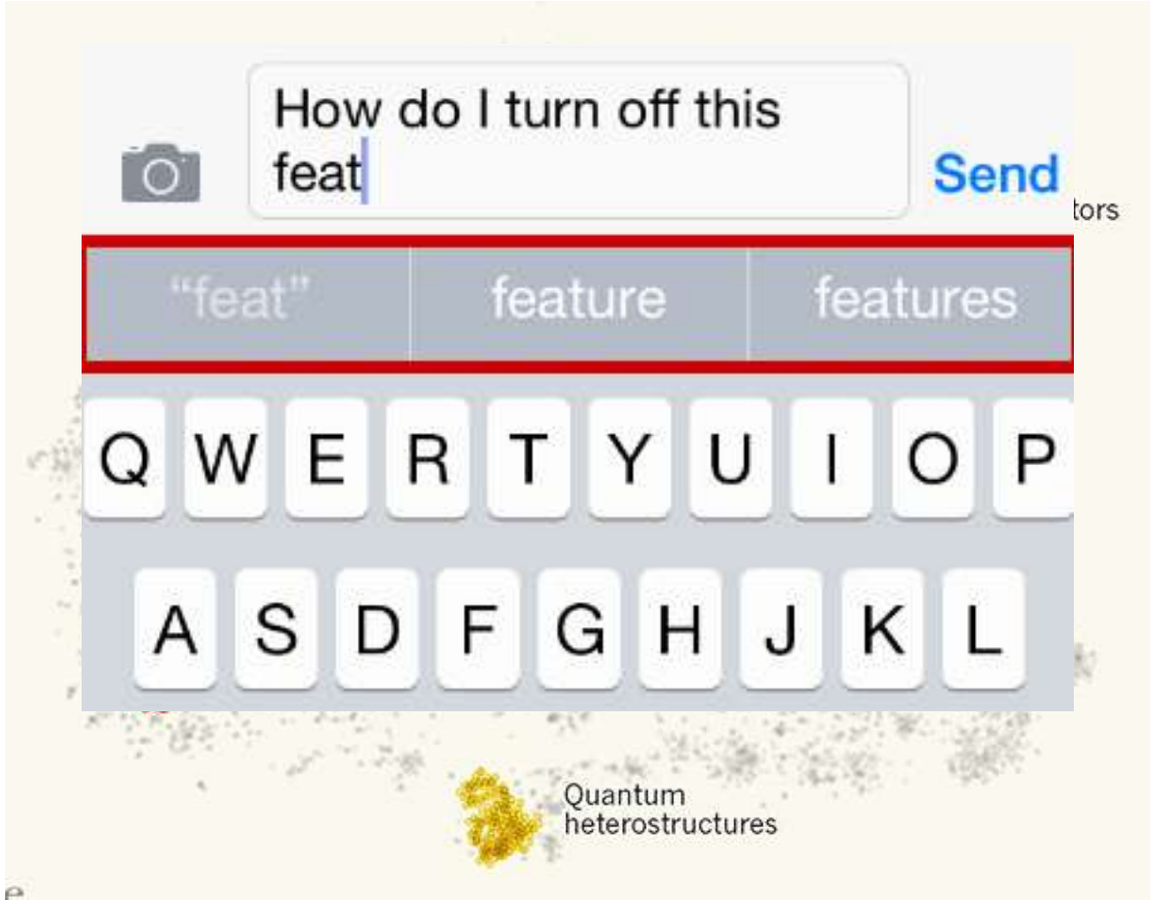
ferromagnetic−NiFe + IrMn ≈ antiferromagnetic

single layer NN is trained to predict all context words for the given target word.



for similar words the context words are the same

Projection of embeddings onto two dimensions (t-SNE).

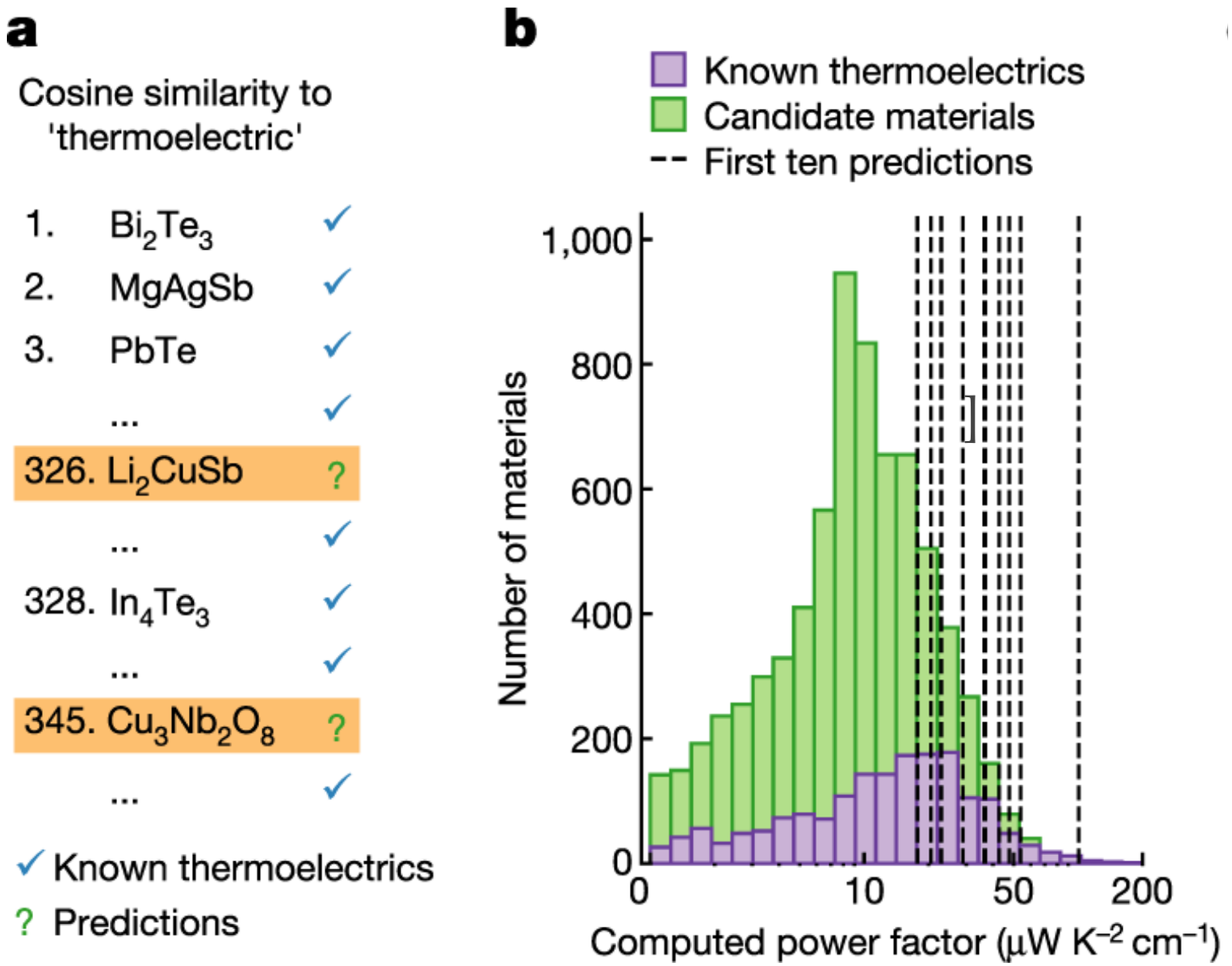


Tshitoyan, V.; *et al*.. *Nature* **2019**, *571* (7763), 95.

Perspective: Isayev, O. *Nature* **2019**, *571* (7763), 42–43.

Word Embeddings: Learning on > 1 Million of Abstracts

Embeddings can also be used for predictions.



- Top ten predictions even slightly higher than known average
- Better rank correlation with experiments than DFT
- Training data is important: model trained on all Wikipedia articles performs worse

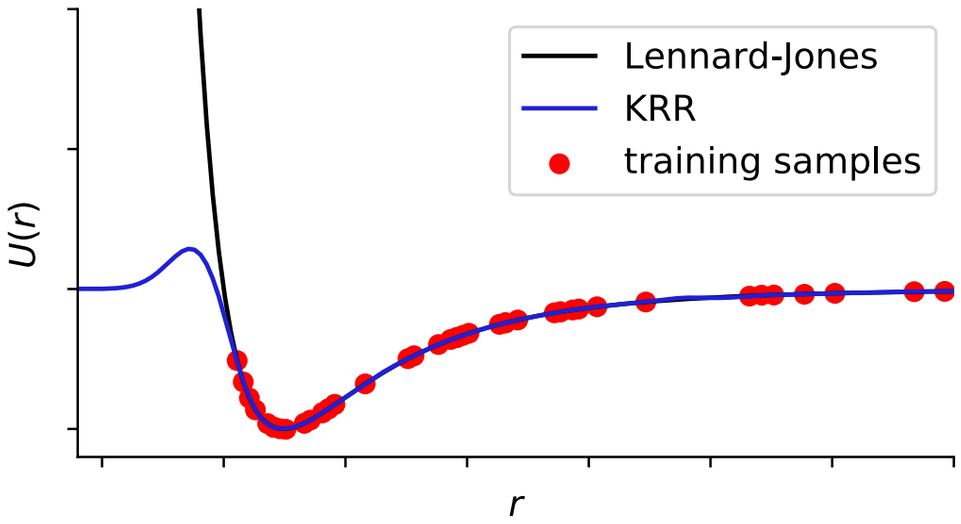
Tshitoyan, V.; *et al*.. *Nature* **2019**, *571* (7763), 95.

GPR for Active Learning of $U(\mathbf{X})$

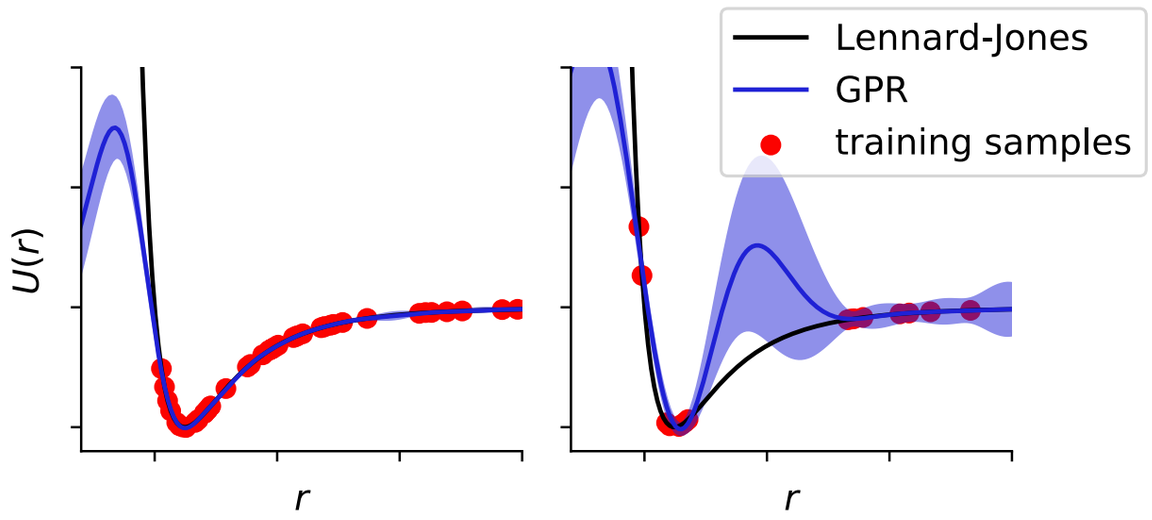
Bayesian version of KRR: Gaussian Process Regression (GPR)

KRR fails if there is no data …

… but cannot warn us

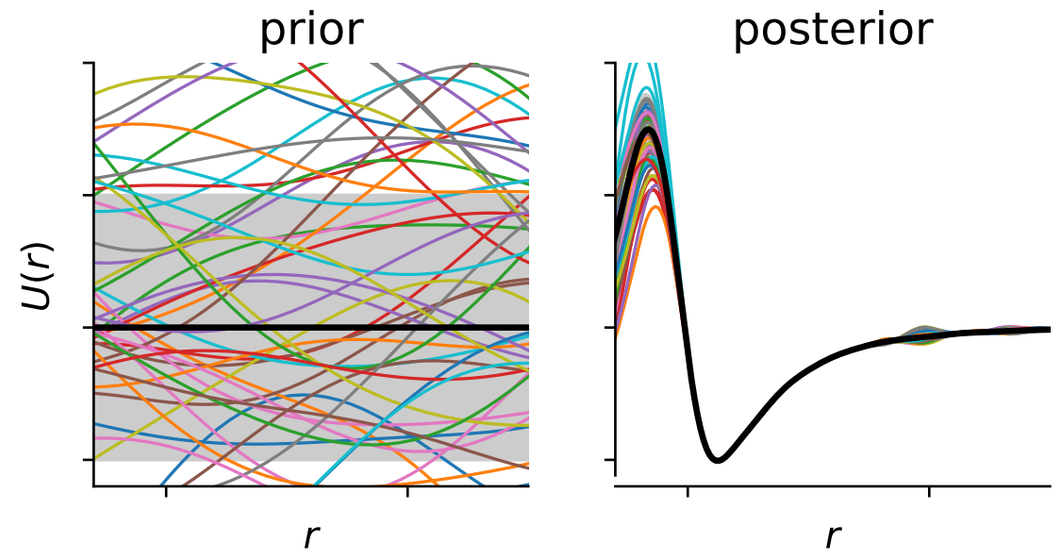


GPR gives uncertainty estimate



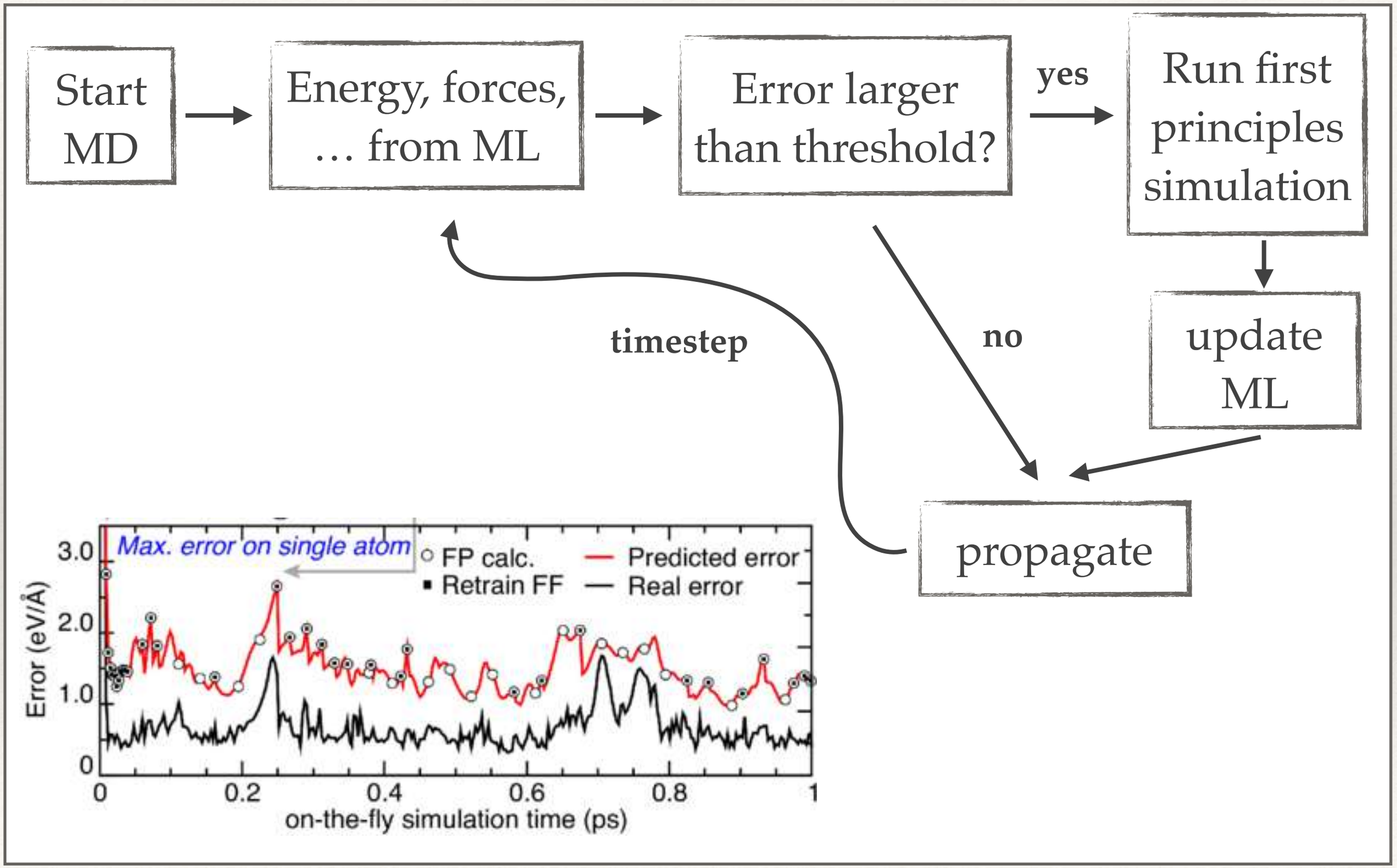
$$\underbrace{P(\theta|D)}_{\text{posterior}} \propto \underbrace{P(D|\theta)}_{\text{likelihood}} \underbrace{P(\theta)}_{\text{prior}}$$

GPR uses data to update a prior distribution of functions to a posterior distribution



Rasmussen, C. E. Gaussian Processes in Machine Learning. In *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003,*

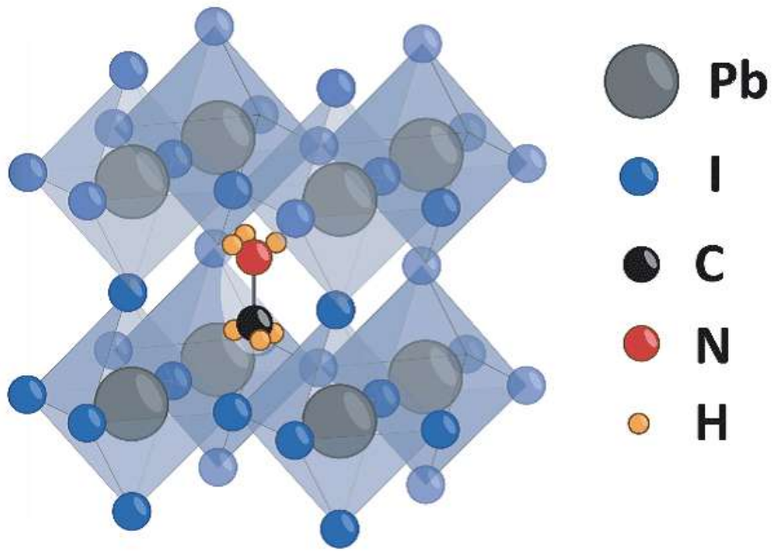
GPR for Active Learning of *U(X)*: Skip 99% of FP Calculations



Jinnouchi, R.; Lahnsteiner, J.; Karsai, F.; Kresse, G.; Bokdam, M. *Phys. Rev. Lett.* **2019**, *122* (22), 225701.

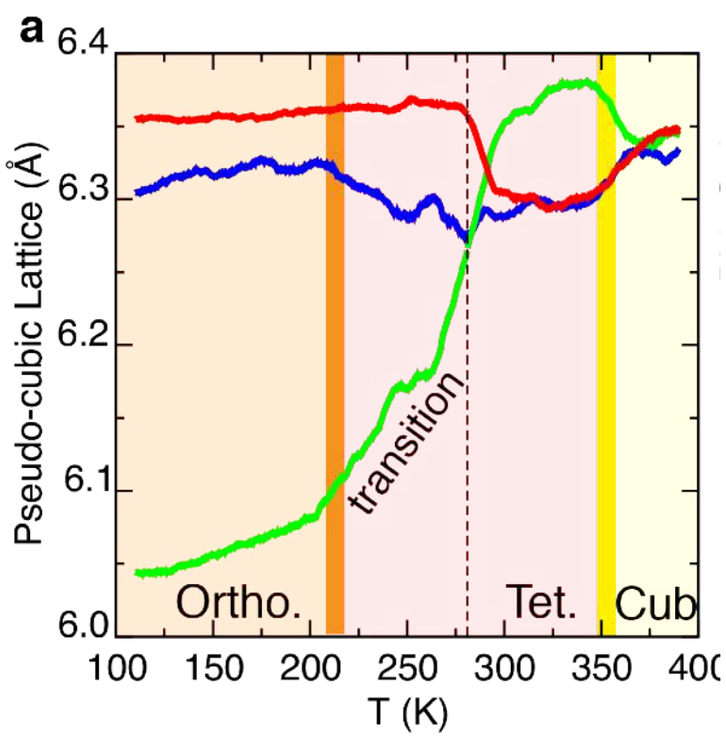
Implemented in VASP6.0.

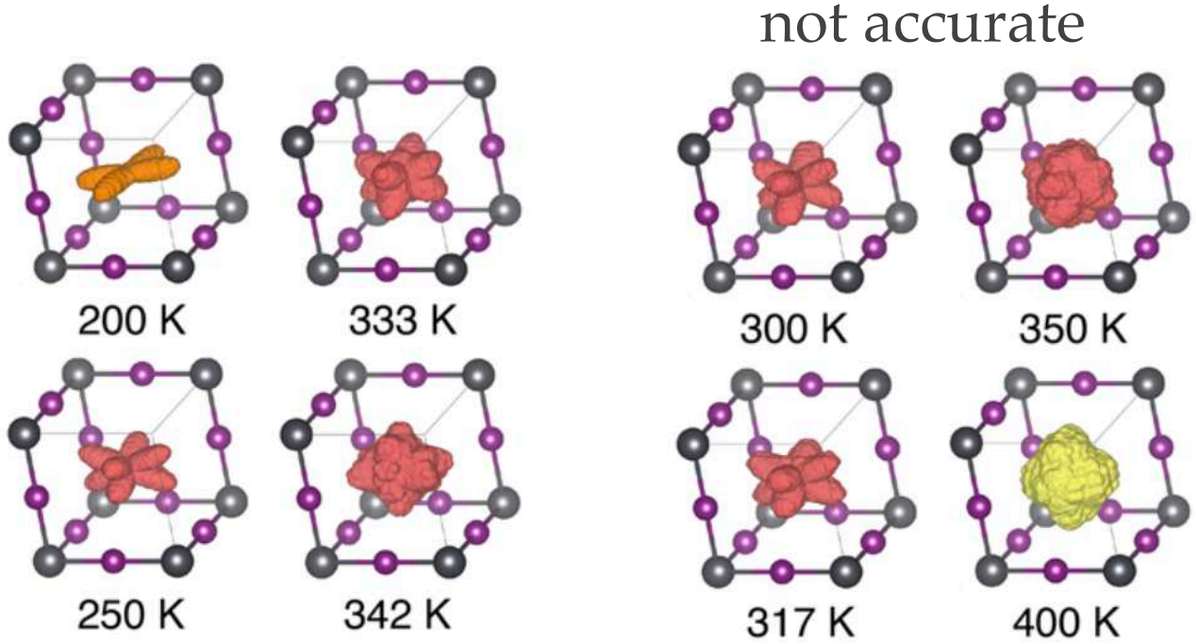
GPR for Active Learning of $U(\mathbf{X})$



Methyl ammonium lead halide perovskites

- Slow rotational dynamics
- Entropy driven phase transition
- Existing force-fields are not accurate



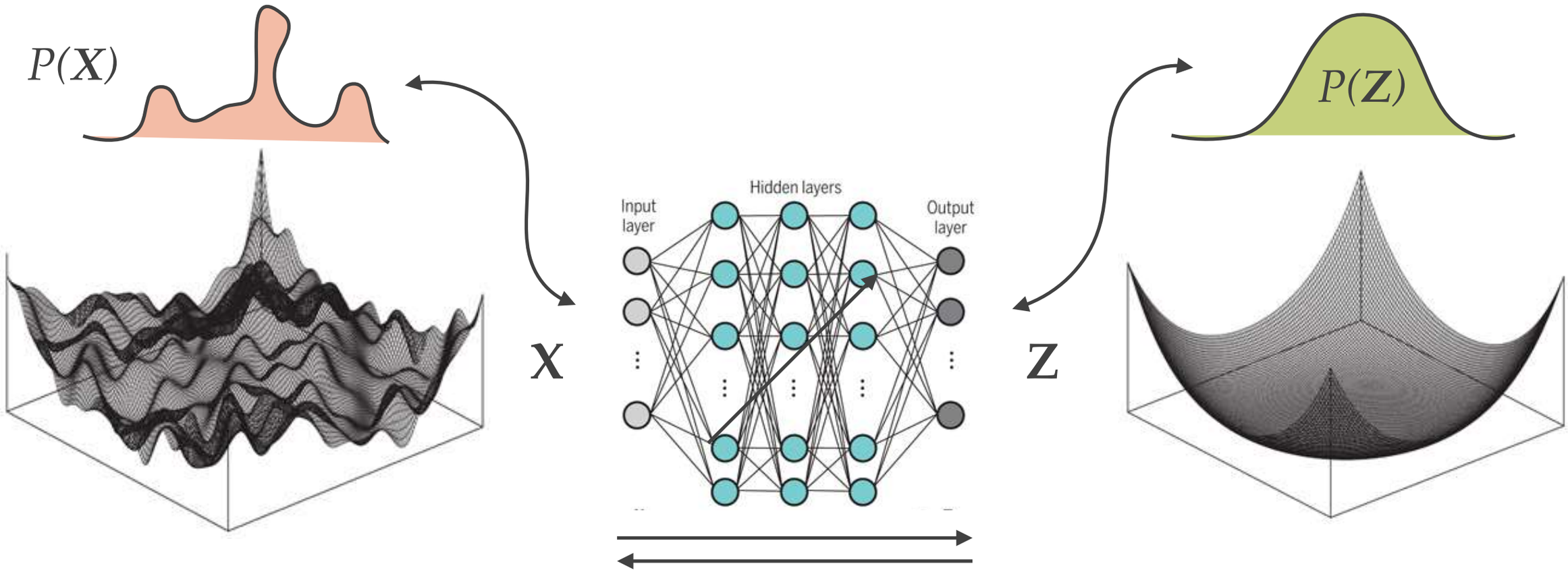


Jinnouchi, R.; Lahnsteiner, J.; Karsai, F.; Kresse, G.; Bokdam, M. *Phys. Rev. Lett.* **2019**, *122* (22), 225701.

Implemented in VASP6.0.

Boltzmann Generators: A New Approach for Sampling of Microstates in One Shot (Statistically Independent)

Molecular Simulations: $U(\boldsymbol{X})$ given, need approach to sample $P(\boldsymbol{X})$



$-kT\lg(P(\mathbf{X}))$

Landscapes are often rugged, hard to sample

Invertible neural network does the invertible mapping

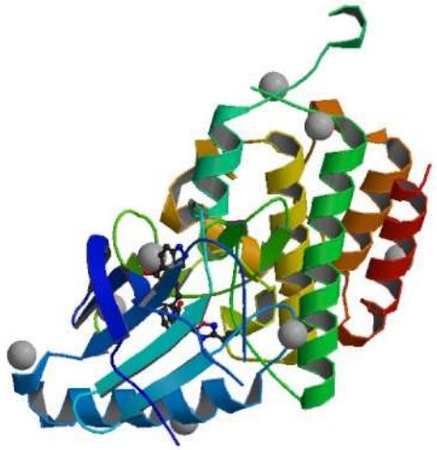
$-kT\lg(P(\mathbf{Z}))$

… this is why we bias some simulations to have low energy states close to each other (flat sampling)

Noé, F.; Olsson, S.; Köhler, J.; Wu, H. *Science* **2019**, *365* (6457),

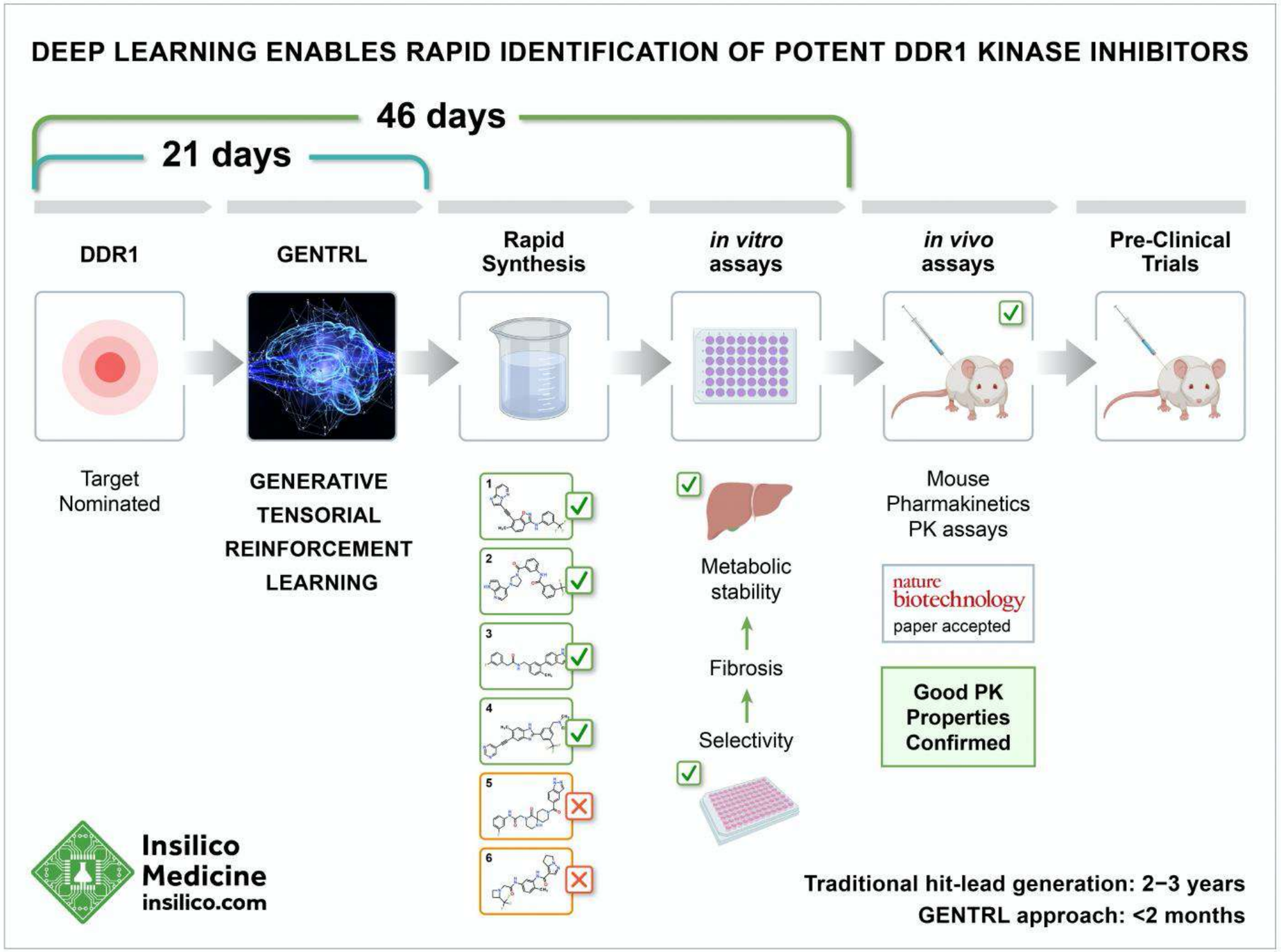
Perspective: Tuckerman, M. E. *Science* **2019**, *365* (6457), 982–983.

21 Days to Drug Candidate: Using Reinforcement Learning to Expedite Drug Discovery



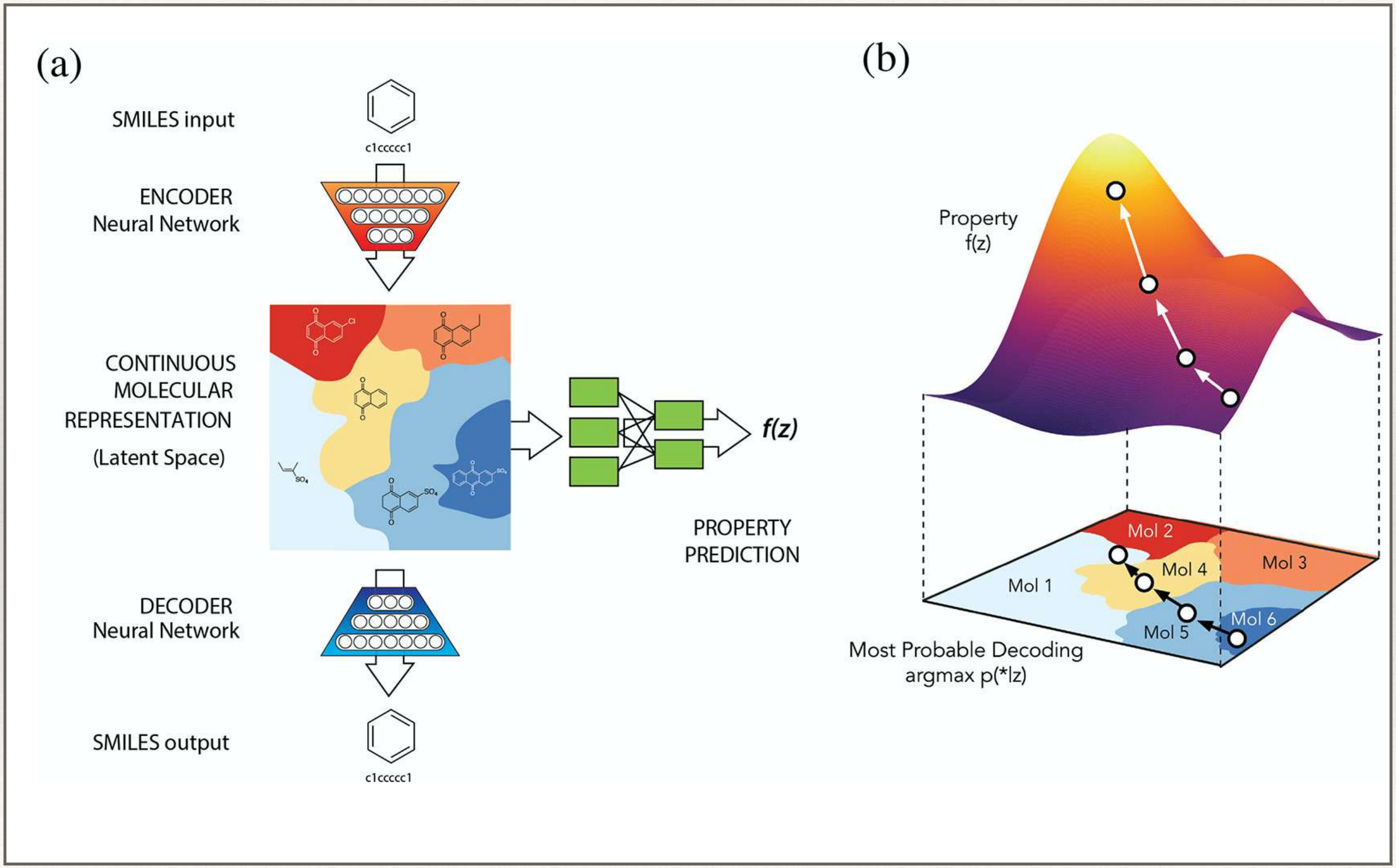
DDR1 (Target for fibrosis)

Model that takes synthetic feasibility, novelty, and biological activity into account



Zhavoronkov, A.; et al. *Nat Biotechnol* **2019**, *37* (9), 1038–1040.

21 Days to Drug Candidate: Using Reinforcement Learning to Expedite Drug Discovery



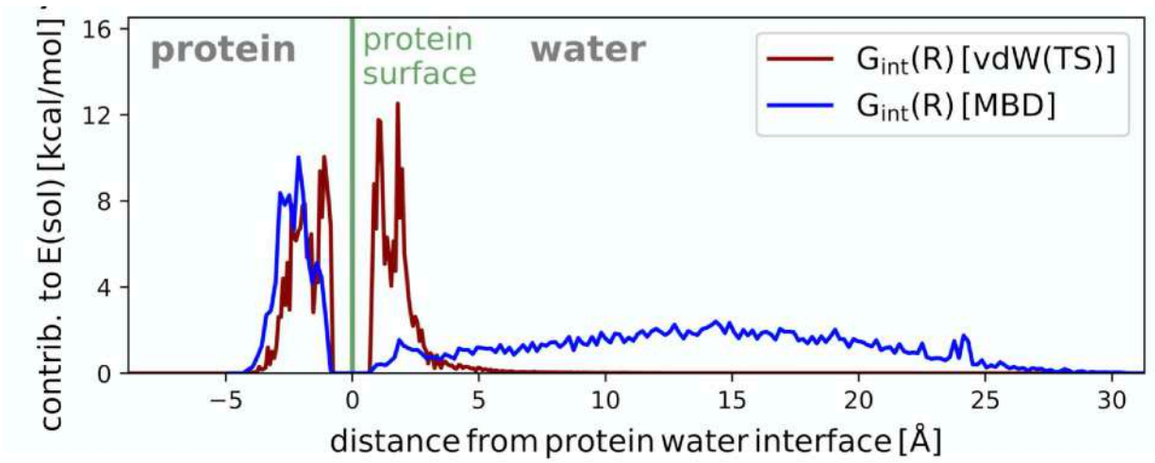
Gómez-Bombarelli, R.; *et al*.. *ACS Cent. Sci.* **2018**, *4* (2), 268–276.

Challenges for the Field

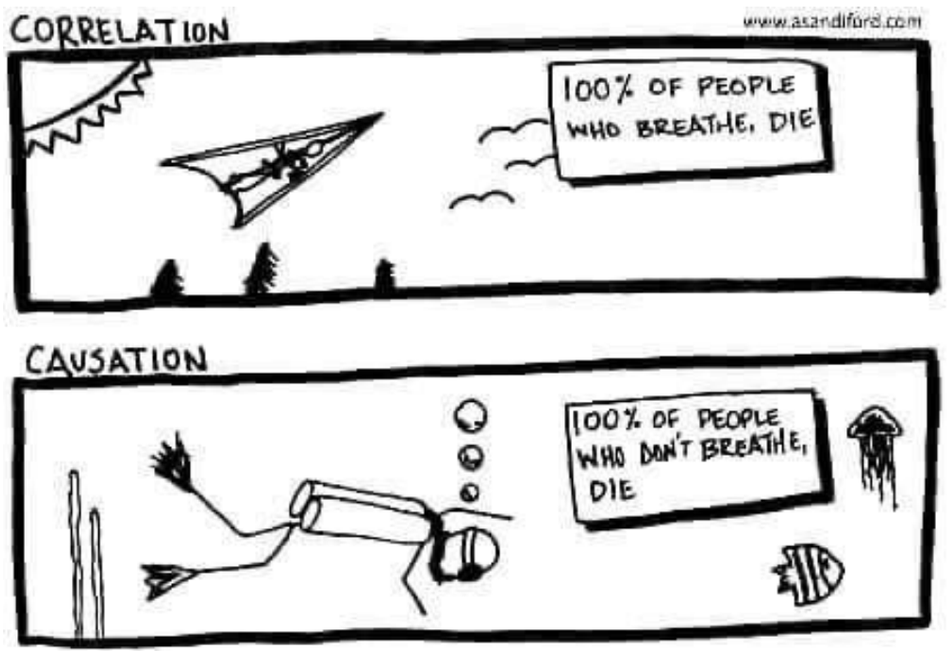
Benchmarks, Reproducibility and Comparability

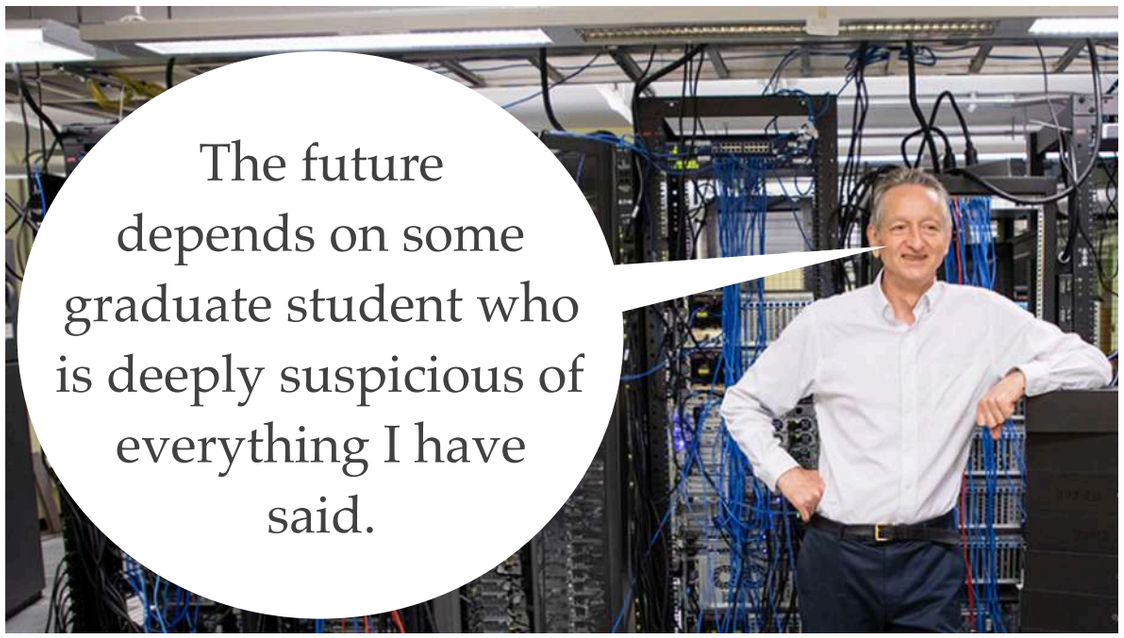
	MNIST	CIFAR10	ImageNet
FGS	0.996	0.911	0.881
JSMA	0.995	0.966	-
Deepfool	0.996	0.960	0.908
CarliniL2	0.989	0.929	0.907
BIM	0.994	0.907	0.820

Incorporation of long-range interactions



Causal Inference





Pearl, J. Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. *arXiv:1801.04016 [cs, stat]* **2018**.

Stöhr, M.; Tkatchenko, A. *Sci. Adv.* **2019**, *5* (12), eaax0024.

For the Exercise Session

Instructions on

https://github.com/kjappelbaum/ml_molsim2020

As soon as you come to the room

download and install

mini