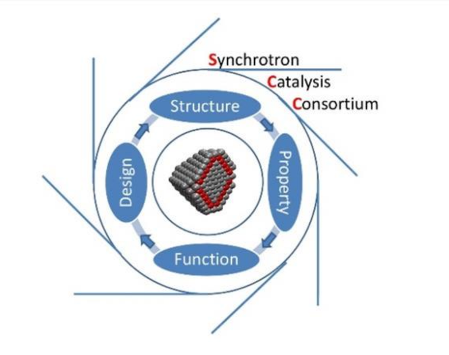


XAFS Short Course 2023

Brookhaven National Lab

Presented by Nick Marcella

NN approaches in XANES and EXAFS data analysis

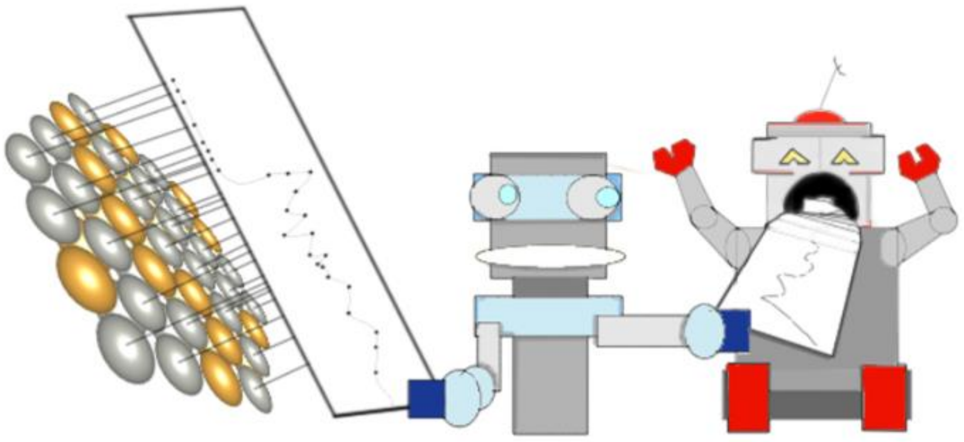
Outline: Neural Networks for XANES and EXAFS data analysis

1. What is a neural network used for? How do they work?

1. How can we frame XANES and EXAFS data analysis as a machine learning problem?

1. A few past examples.

1. Some guidance if you want to DIY.

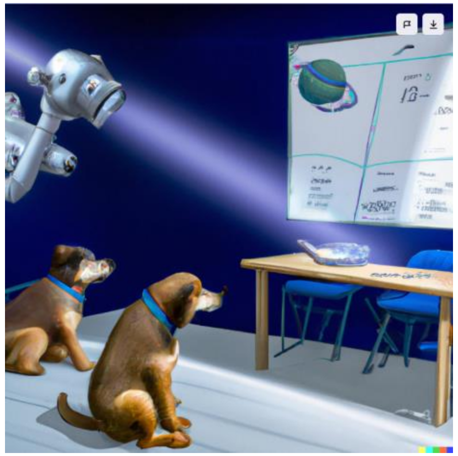
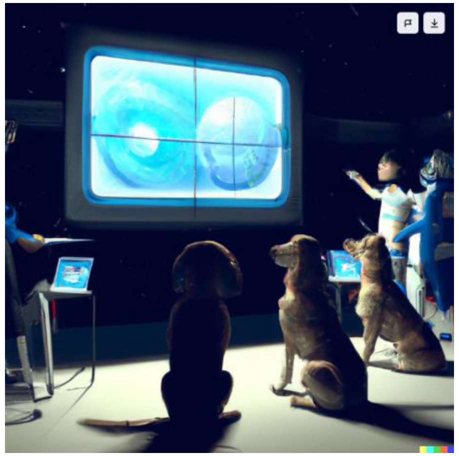


The (generative) AI revolution is currently underway.

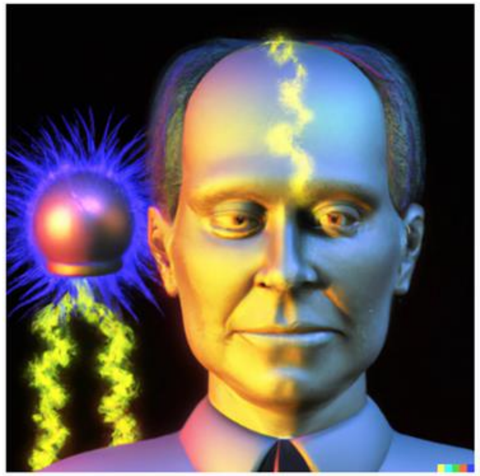
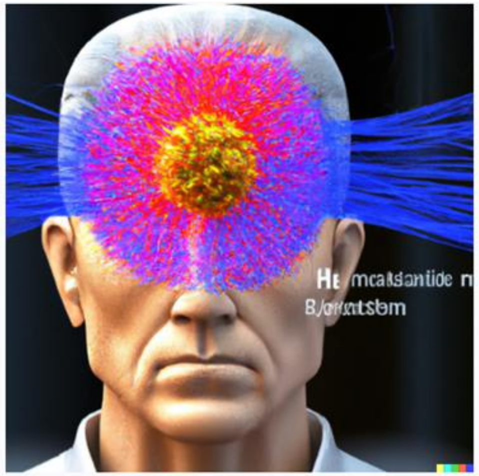
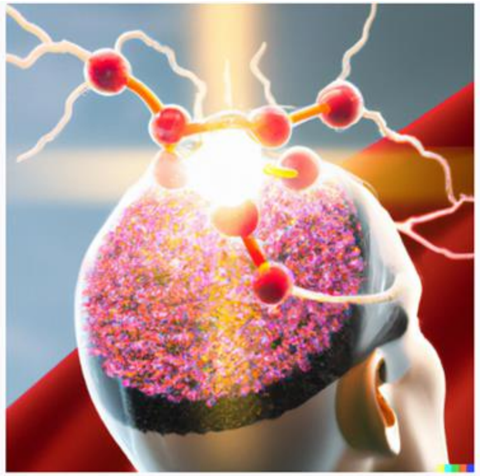
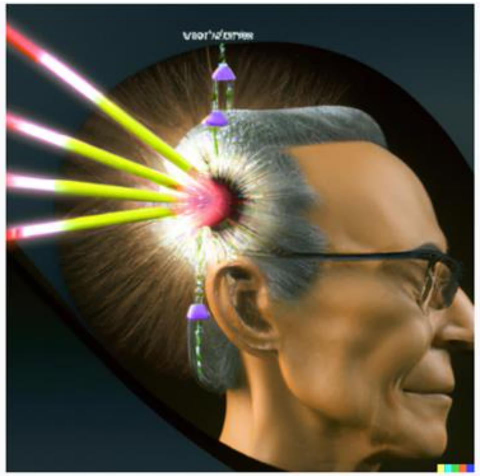
labs.openai.com

last year: 2022

“short course on x-ray spectroscopy. taught by dogs. in space.”

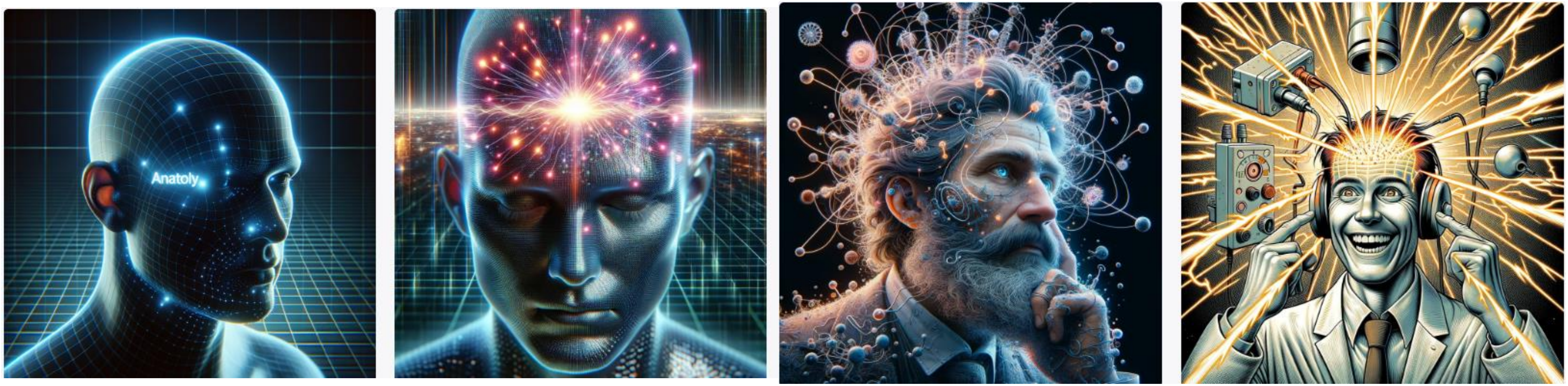



“3d render. photoelectron backscattering off Anatoly's head”

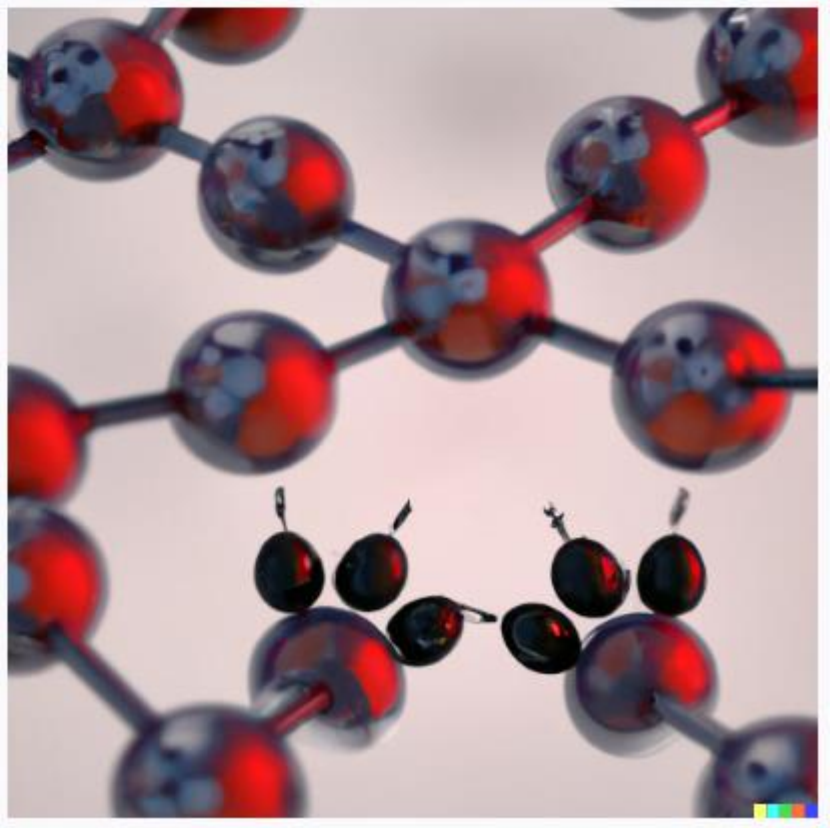





“short course on x-ray spectroscopy. taught by dogs. in space.”

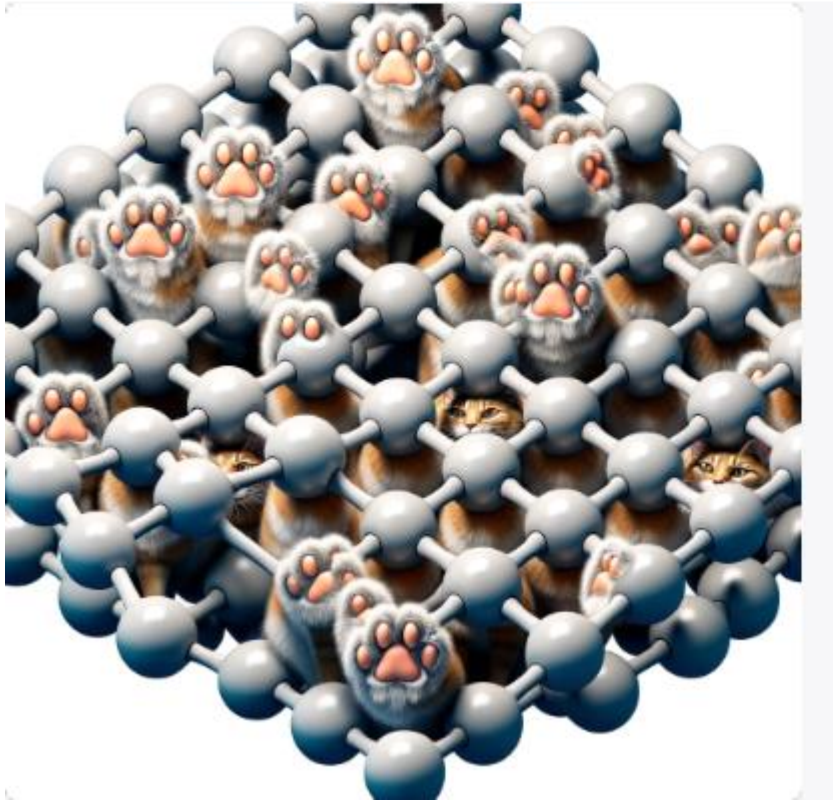
“3d render. photoelectron backscattering off Anatoly's head”



3d render . a lattice of atoms. but the atoms are cat paws



2022



2023

…. and now we have ChatGPT

N

Can you write me a Python function to generate chi(k) (using the EXAFS equation) with variables N, R, SS, S02, and placeholde
for now) for lambda(k), feff(k), and delt
resolution. If I'm missing anything, just

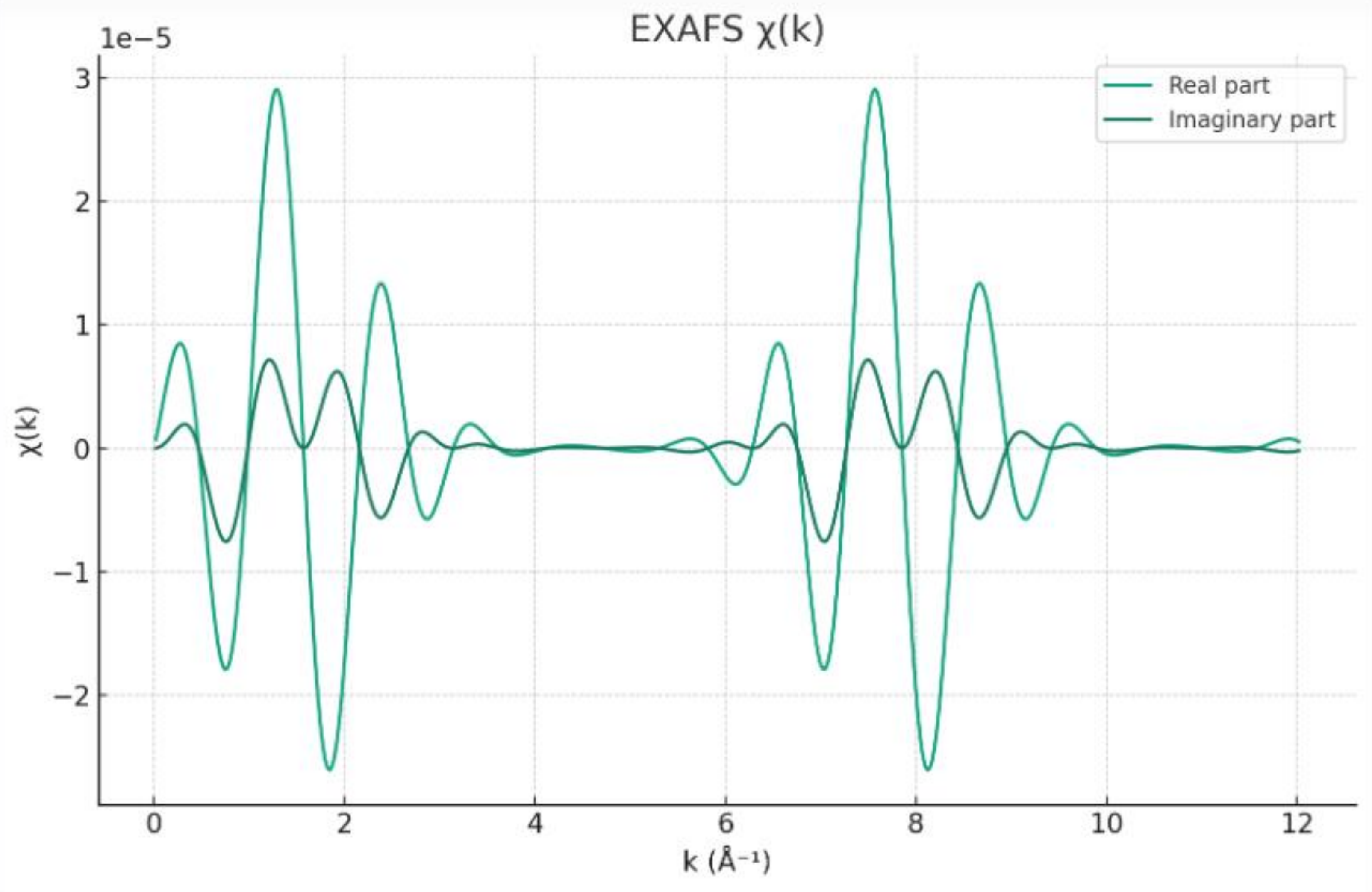
Sure, let's start by writing down the Extended X
equation:

$$\chi(k) = S_0^2 \sum_j N_j \frac{e^{2i\delta_j(k)}}{R_j^2} f_{\mathrm{eff},j}(k$$

Here, N_j is the number of backscattering atoms, R_j is the distance to the backscattering atoms, S_0^2 is the amplitude reduction factor, $\delta_j(k)$ is the phase shift, $f_{\mathrm{eff},j}(k)$ is the effective scattering amplitude, and $\lambda_j(k)$ is the mean free path.

Now let's proceed to write a Python function for this. I'll use NumPy for numerical operations and Matplotlib for plotting. I'll use placeholder functions for $\lambda(k)$, $f_{\mathrm{eff}}(k)$, and $\delta(k)$ as you requested.

$$\chi(k) = \sum_j \frac{1}{kR_j^2} N_j S_0^2 f_j(k) e^{-2R_j/\lambda(k)} e^{-2k^2\sigma_j^2} \sin[2kR_j + \delta_j(k)]$$



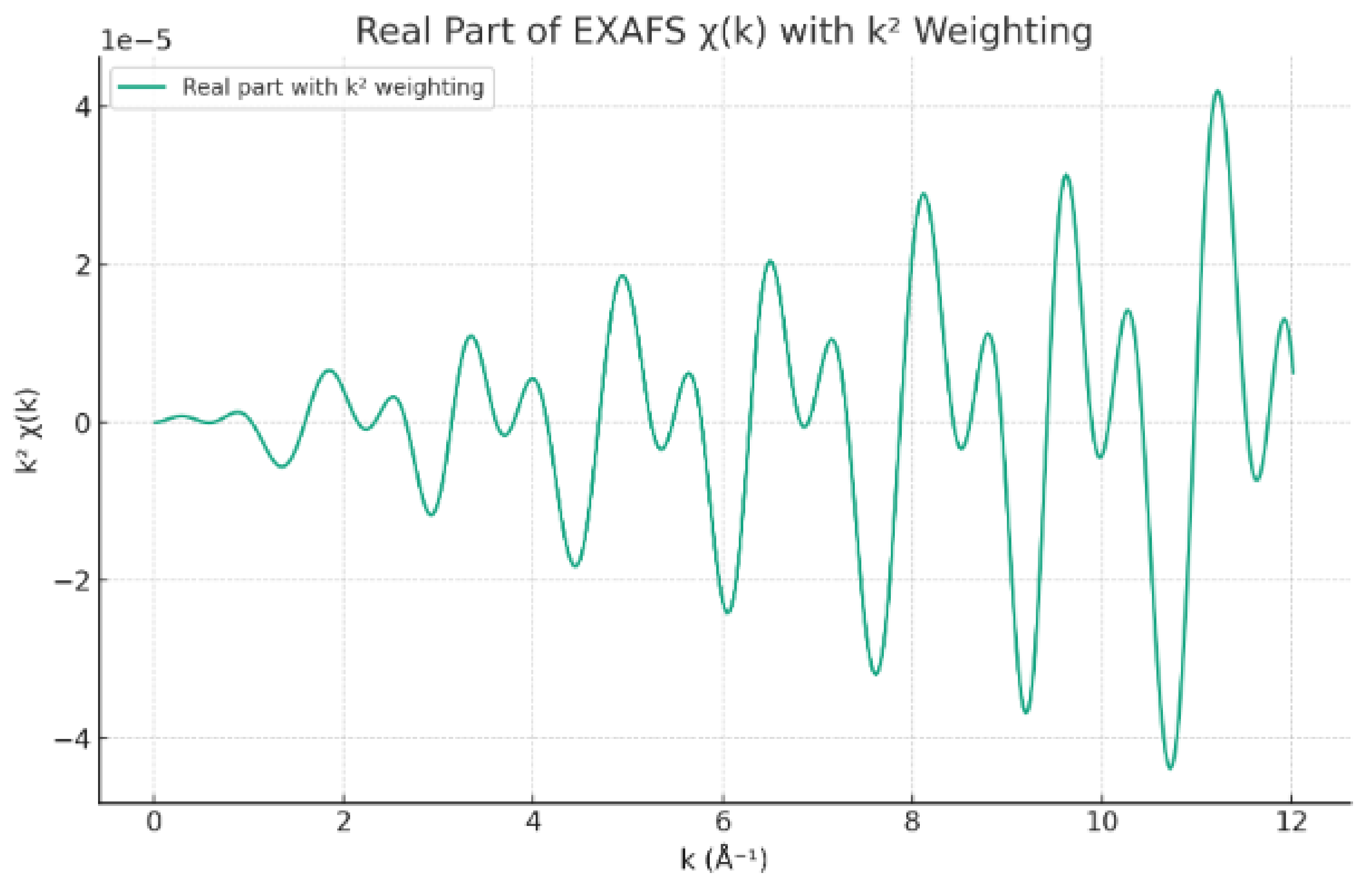
Here's the plot of $\chi(k)$ using the EXAFS equation with placeholder functions for $\lambda(k)$, $f_{\text{eff}}(k)$, and $\delta(k)$. I used example parameters for N, R, SS, and $S0^2$.

Let me know if you'd like to adjust anything!

N

can you show the real part with k^2 weighting

Finished working Show work

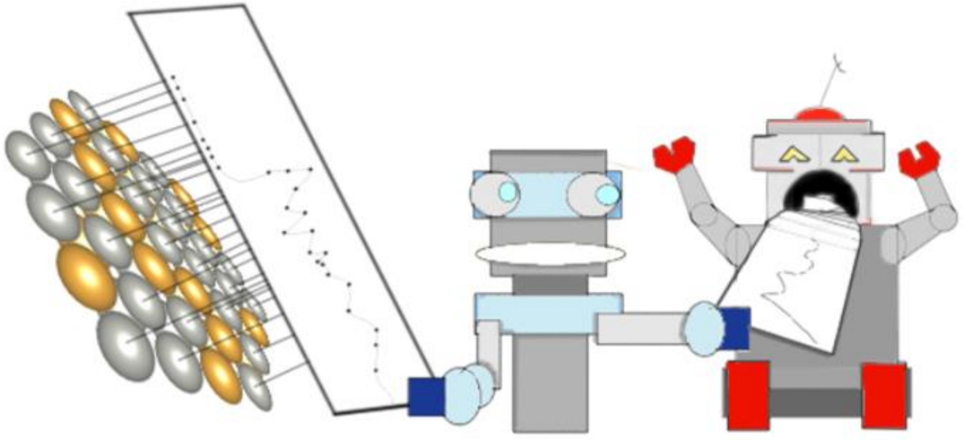


Here's the plot of the real part of $\chi(k)$ with k^2 weighting. Anything else you'd like to adjust?

Outline: Neural Networks for XANES and EXAFS data analysis

1. What is a neural network used for? How do they work?

1. How can we frame XANES and EXAFS data analysis as a machine learning problem?



1. A few past examples.

1. Some guidance if you want to DIY.

First, let's take a step back.

What's the point of modeling data, and data analysis in general?

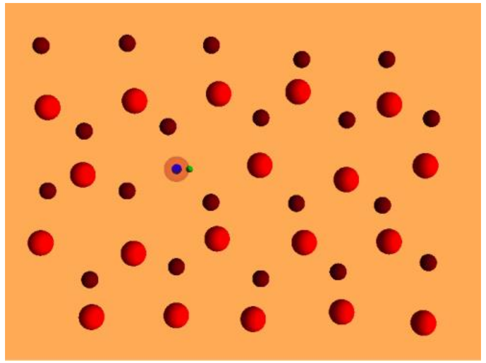
Generative factors → Observations → Model → Predictions

Parameters → Model

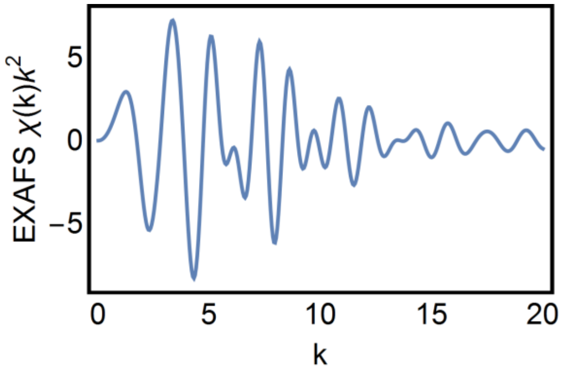
Parameters → Generative factors ?

- A model that explains the observations and can accurately predict new observations is general.
- The hope is that a model can be created in which the model parameters are functions of the generative factors. In this case, the model can be used to extract useful information from the observations.

Generative factors	Observations	Model



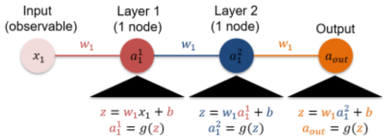
How deep should we go?



$$\chi(k) = \sum_j \frac{1}{kR_j^2} N_j S_0^2 f_j(k) e^{-2R_j/\lambda(k)} e^{-2k^2\sigma_j^2} \sin[2kR_j + \delta_j(k)]$$

$$\chi(k) = S_0^2 \int_0^\infty g(R) \frac{f(k,R)}{kR^2} \sin[2kR + \phi(k)]\, dR$$ **(More general)**

(Neural network)



(More general? more flexibility with parameters?)

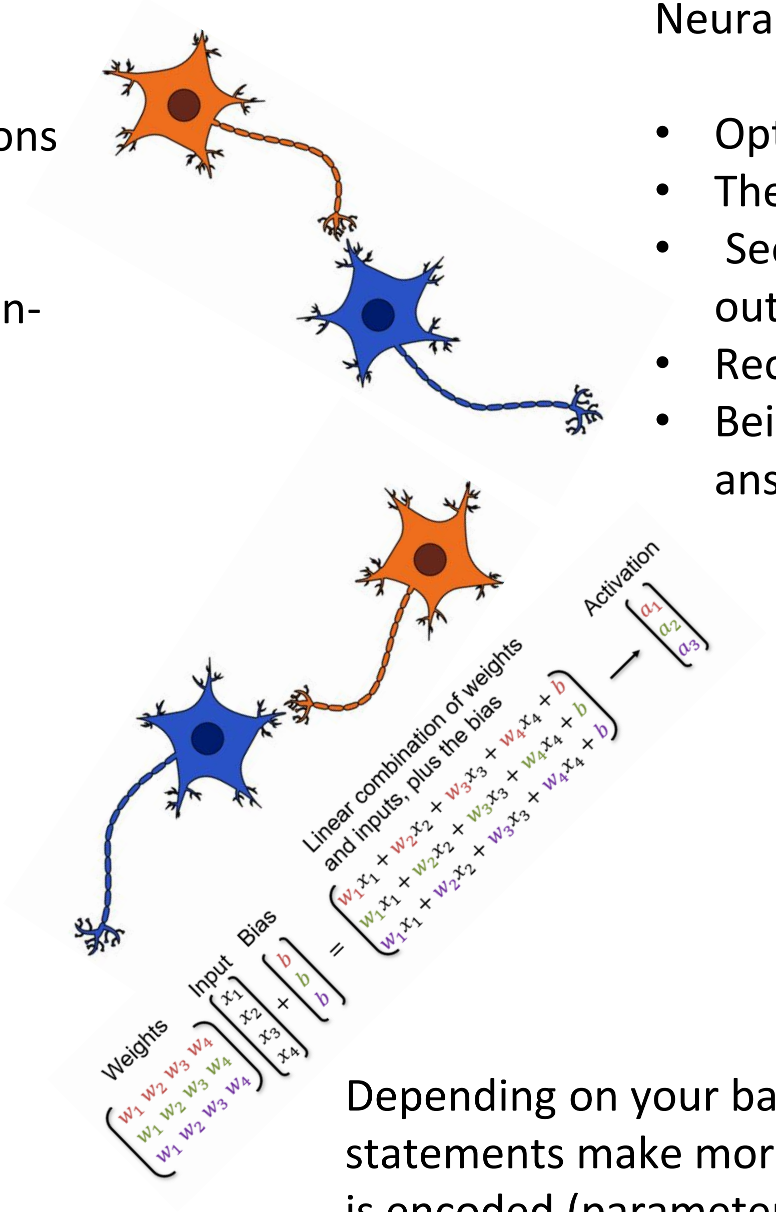
Neural Networks, conceptually.

Neural networks are

- Mathematical models of human neurons
- non-linear functions
- Universal approximators
- Chains of nodes linked together by non-linear functions
- A mathematical object constructed of linked matrices and/or vectors.

Neural networks are trained to

- Transform an input into an output.
- Map inputs to outputs.
- Predict outputs given an input
- Learn the relationship between the input and the output.
- Learn a representation of the input so that the representation allows for the prediction of the output.
- Minimize the difference between the input and the output

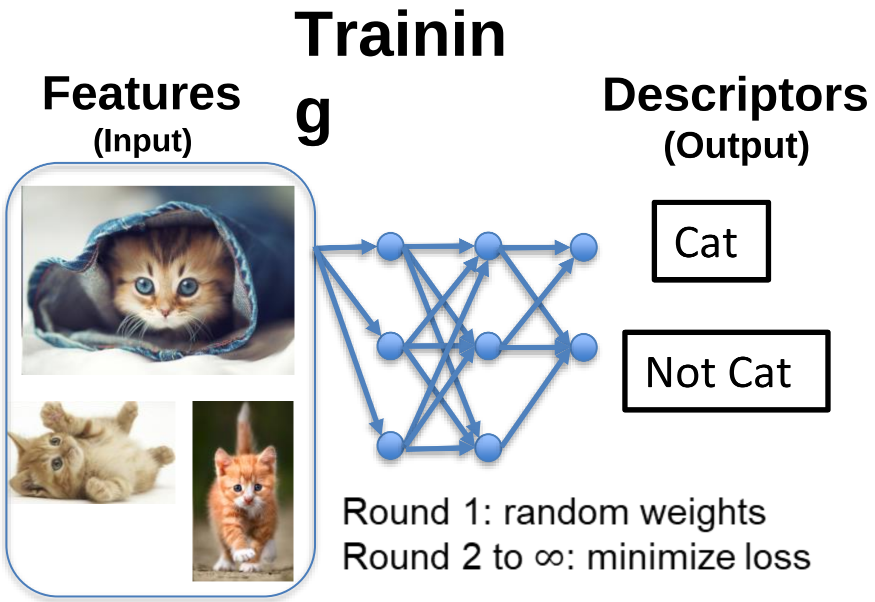


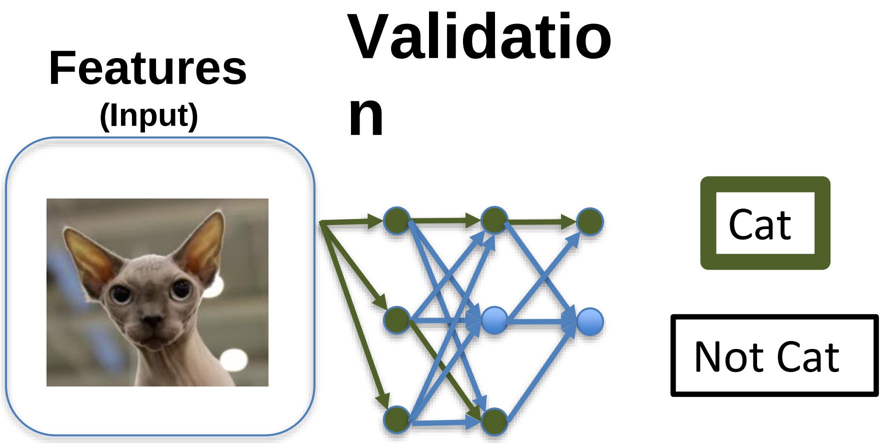
Neural networks learn by:

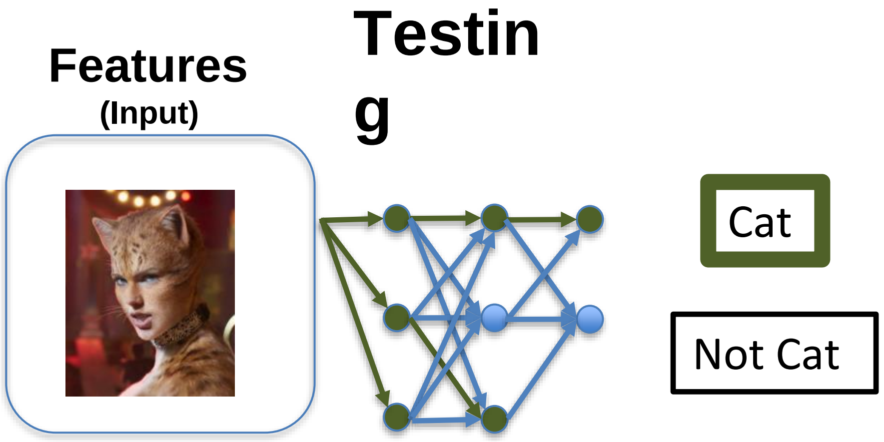
- Optimization of weights and biases.
- The minimization of a loss function.
- Seeing many examples of inputs and associated outputs.
- Receiving rewards when it gets the answer right.
- Being rewired or destroyed when it gets the answer wrong.

Depending on your background and context, one of these statements make more sense to you than the others. (the concept is encoded (parameterized) in terms of our experiences)

Neural Networks, ELI5.

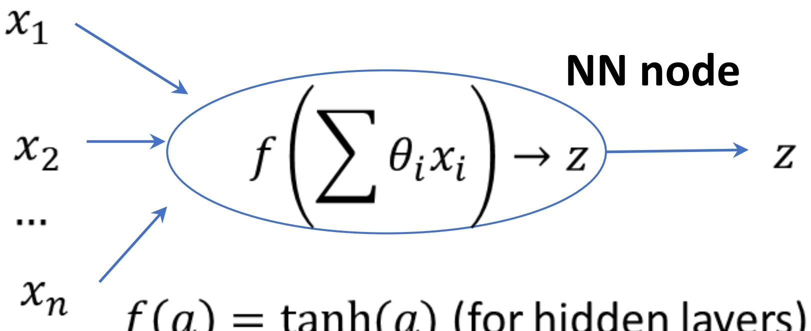


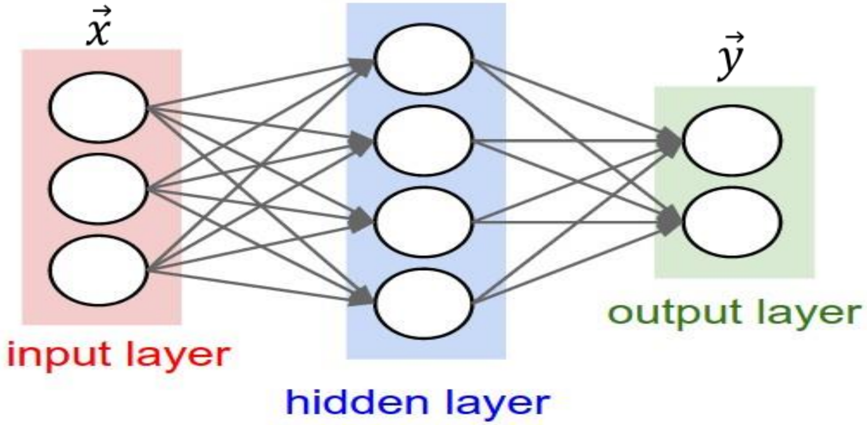




Neural Networks, ELI10.

Regression: find such $y = h\left(x, \vec{\theta}\right) \to \mathbb{R}^n$ that for given x "predicts" respective y value

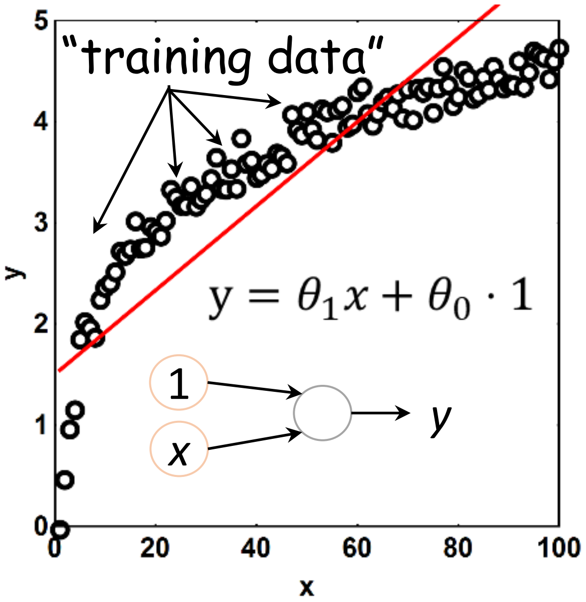




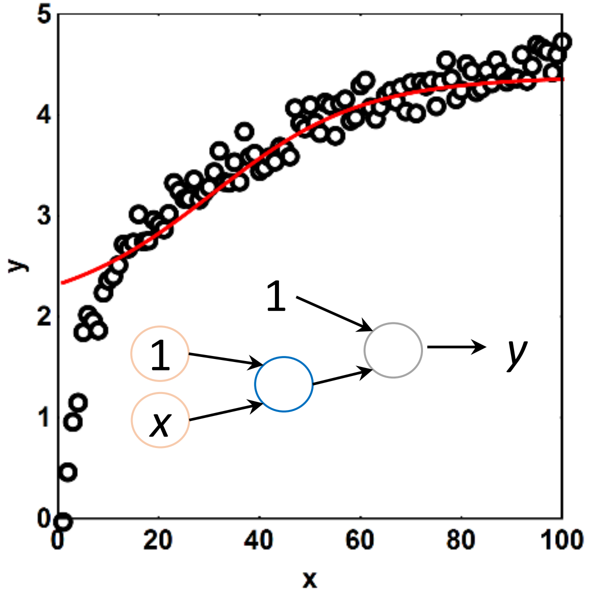
$f(a) = \tanh(a)$ (for hidden layers)

$f(a) = a$ (for output layer

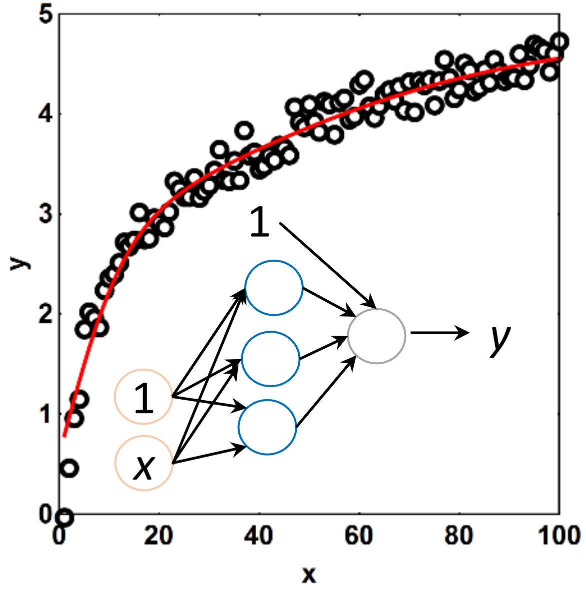
No hidden nodes:



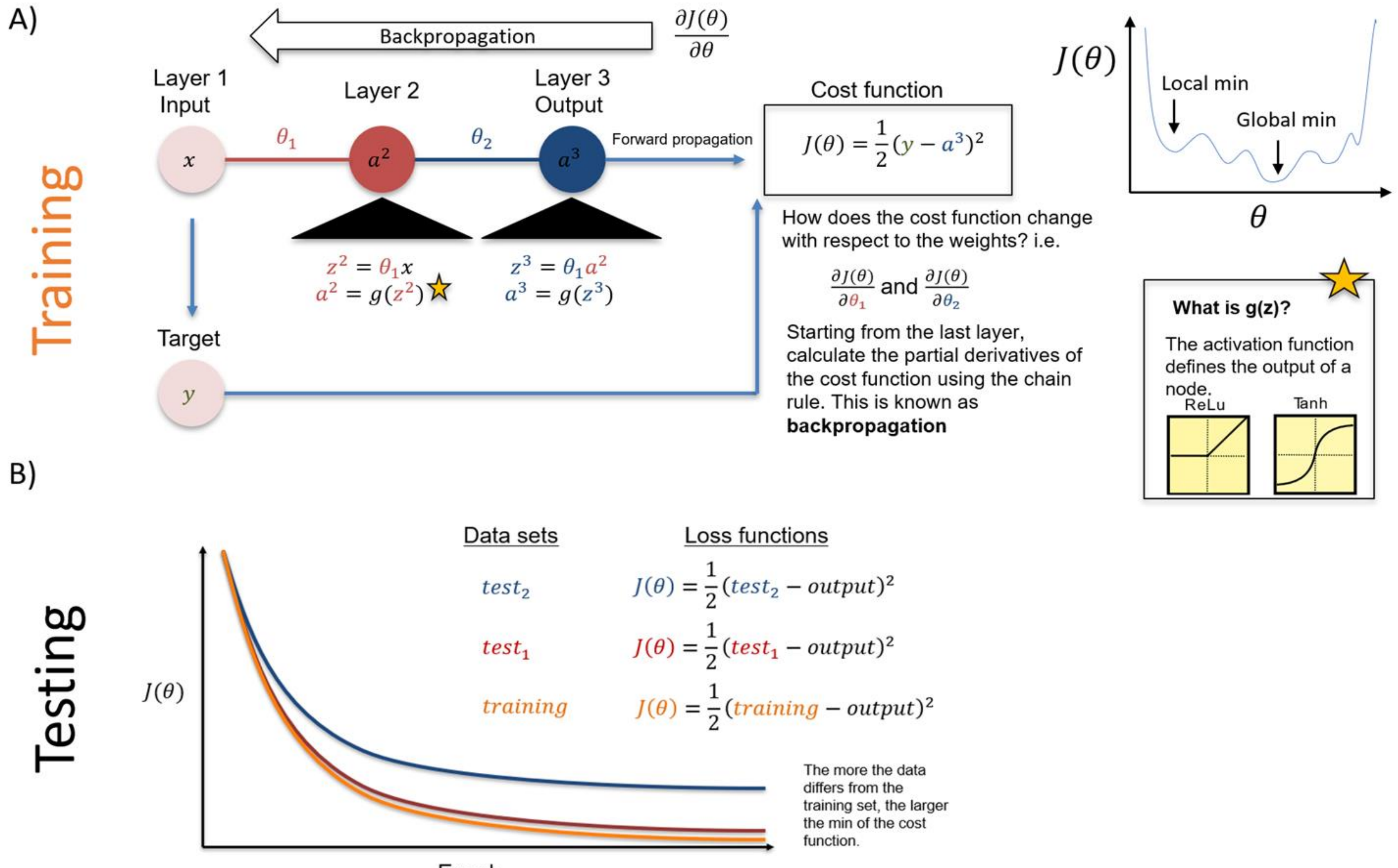
1 hidden node:



3 hidden nodes:



Neural Networks. training, and testing. ELI10.

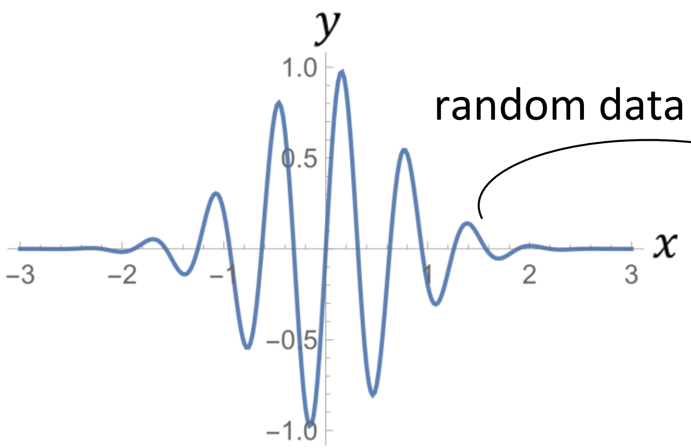


The training set is the only data that takes part in weight optimization.

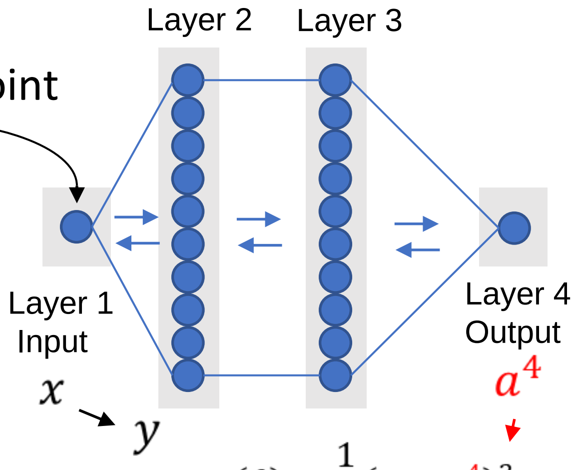
The training set needs to be robust!

Putting it all together. A real example.

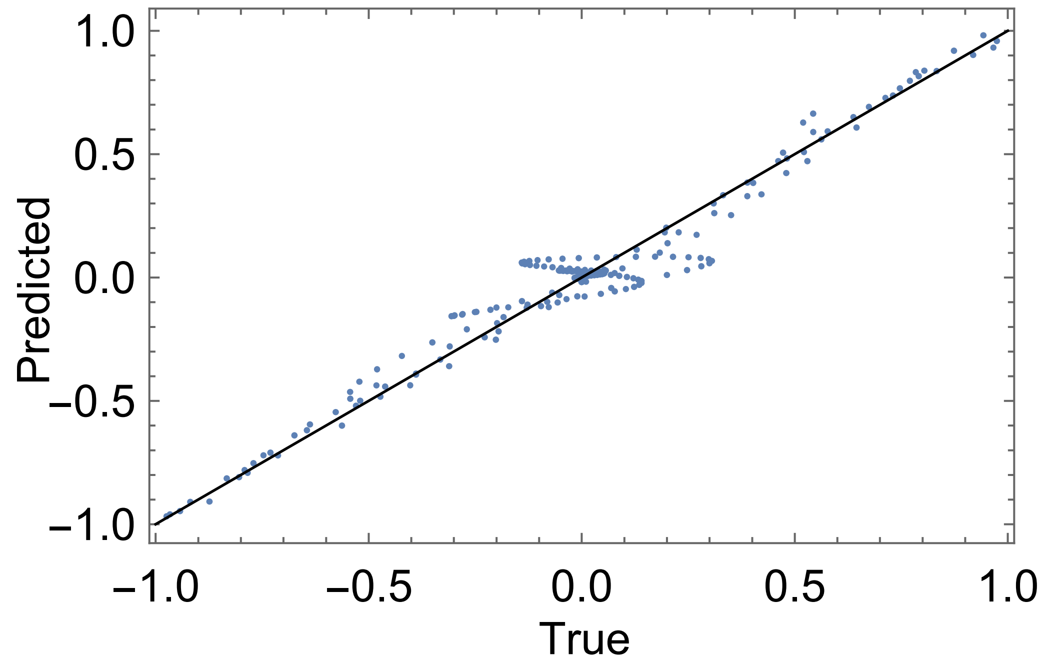
Some observations



Function



$$J(\theta) = \frac{1}{2}(y - a^4)^2$$

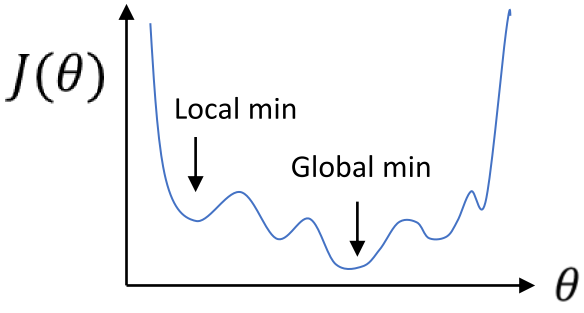


Optimization of the loss function

How does the cost function change with respect to the weights? i.e.

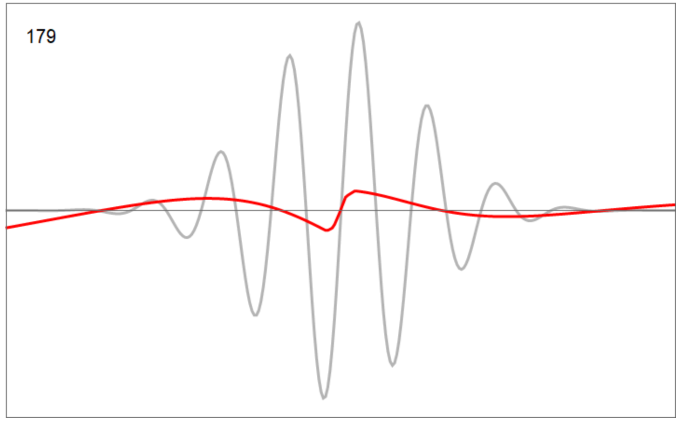
$$J(\theta) = \frac{1}{2}(y - a^4)^2$$

$$\frac{\partial J(\theta)}{\partial \theta_1}, \frac{\partial J(\theta)}{\partial \theta_2}, \text{and } \frac{\partial J(\theta)}{\partial \theta_3}$$



Realtime model training

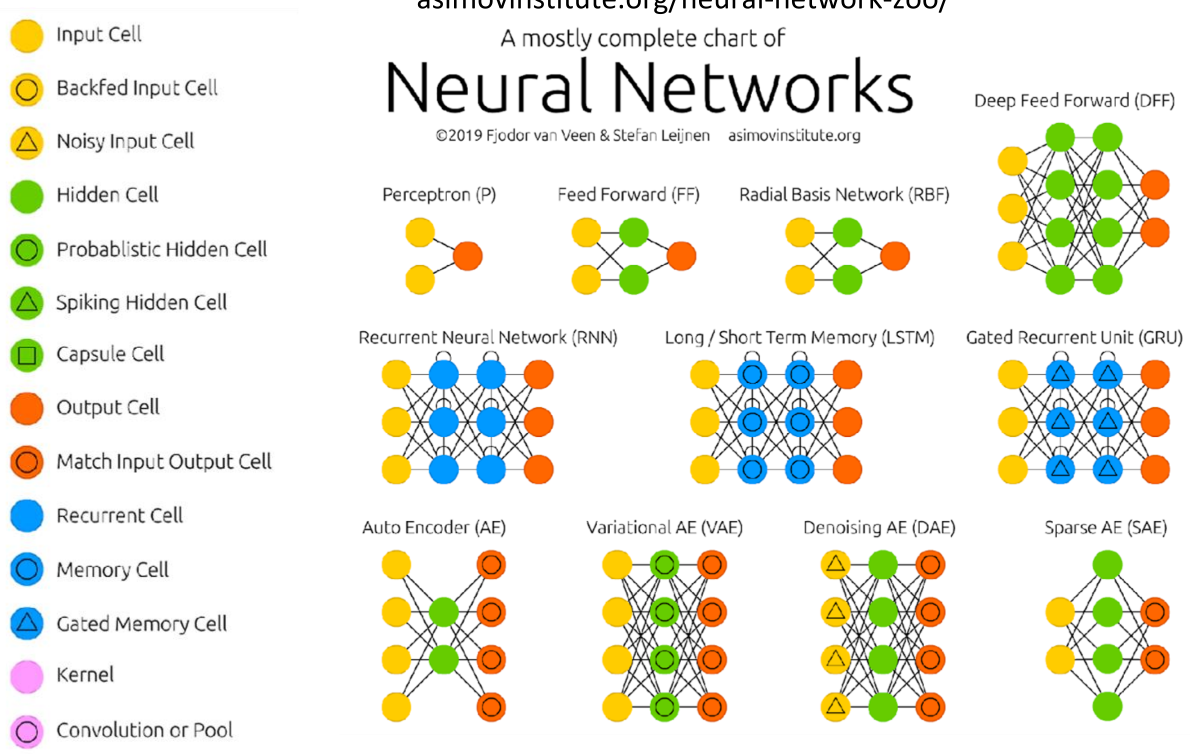
Iterations→

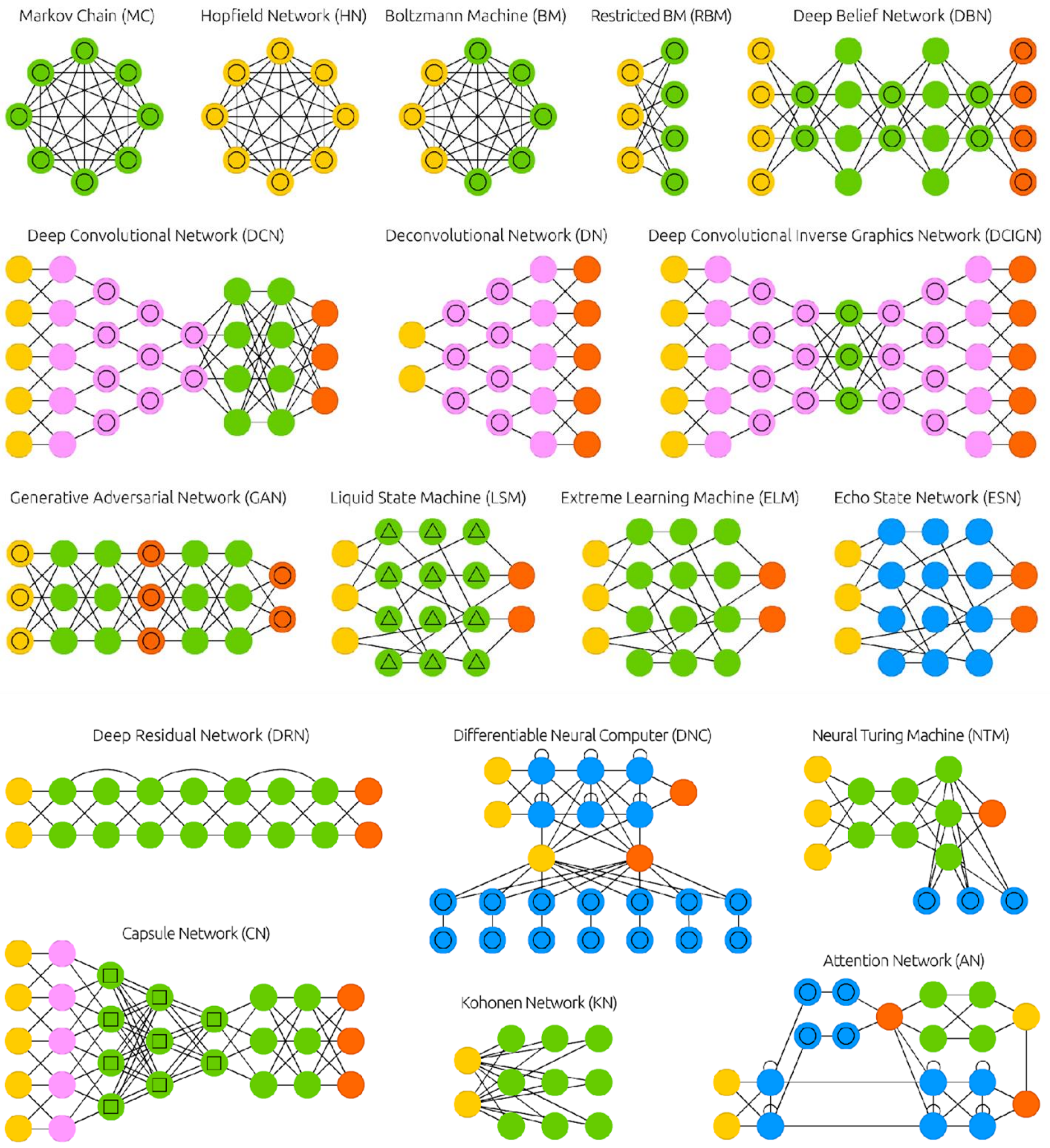


Vary θ so that, eventually, $a^4 = y$

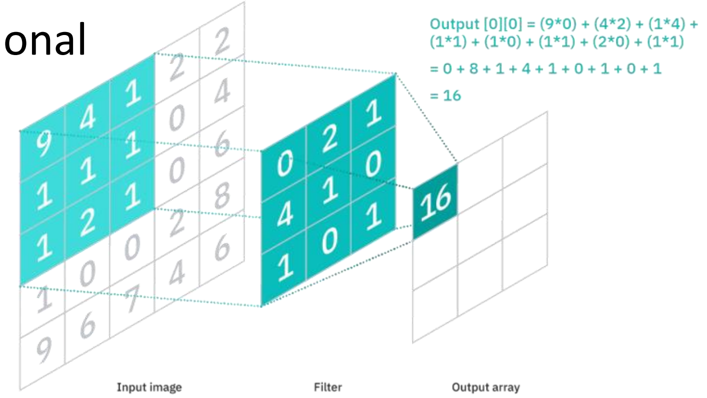
Neural Networks: Architectures

asimovinstitute.org/neural-network-zoo/





Convolutional

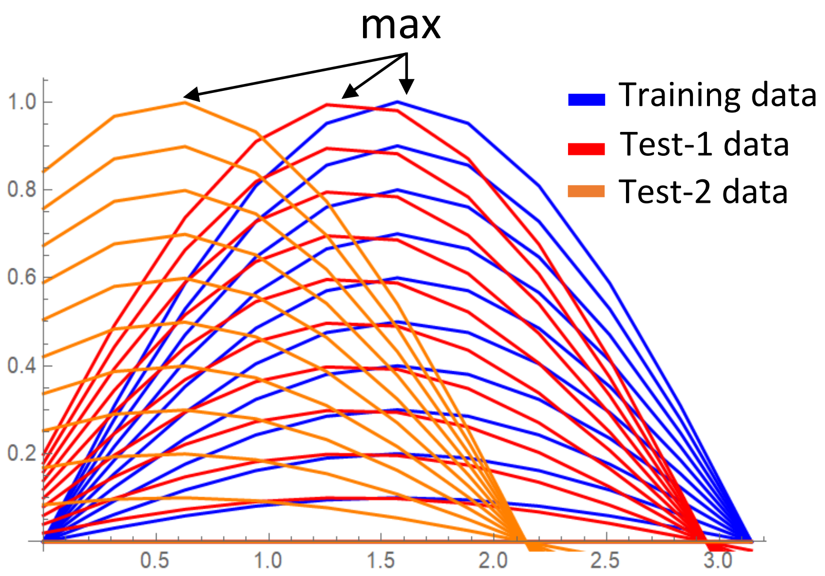


https://www.ibm.com/cloud/learn/convolutional-neural-networks

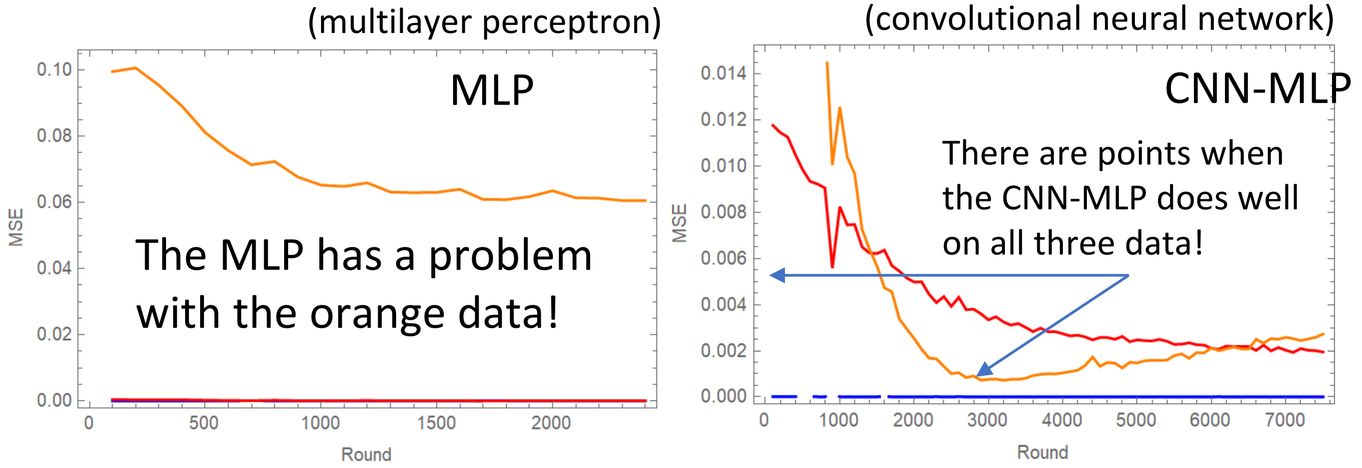
The choice of architecture influences generalization. Generalization is key.

EXAMPLE to demonstrate the idea

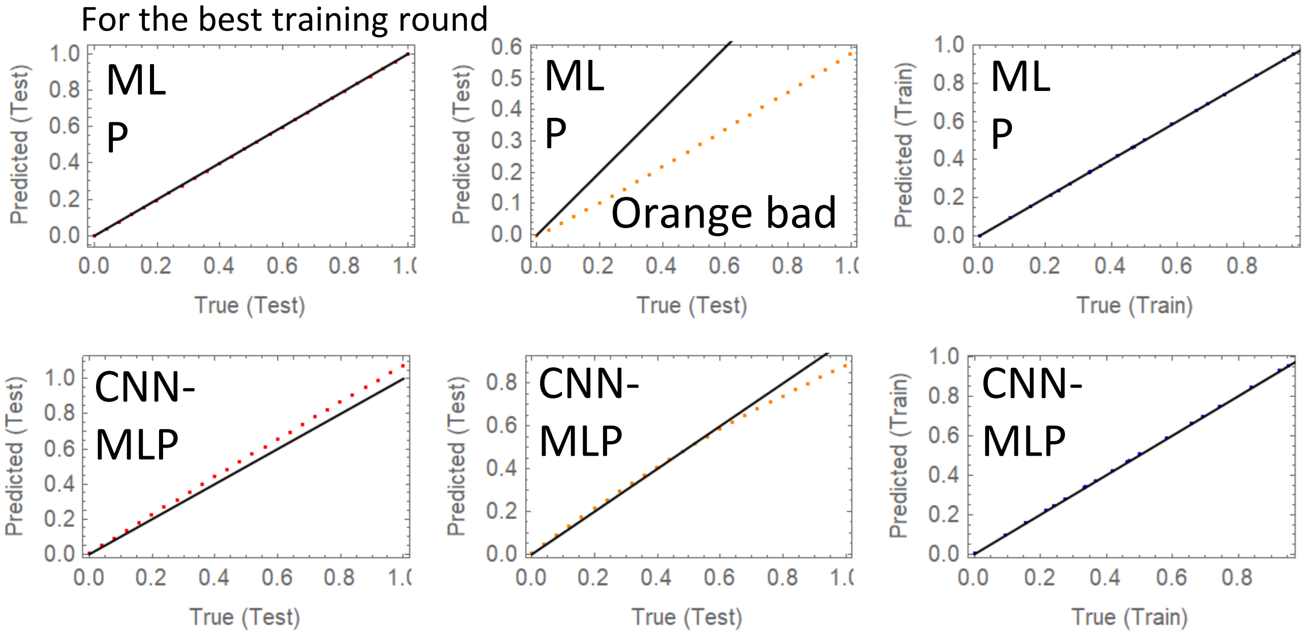
A)



B)



C)



All three ok! But could do better

A) The training data and two sets of test data.

A) Training MLP and CNN-MLP to predict the maximum value from each curve.

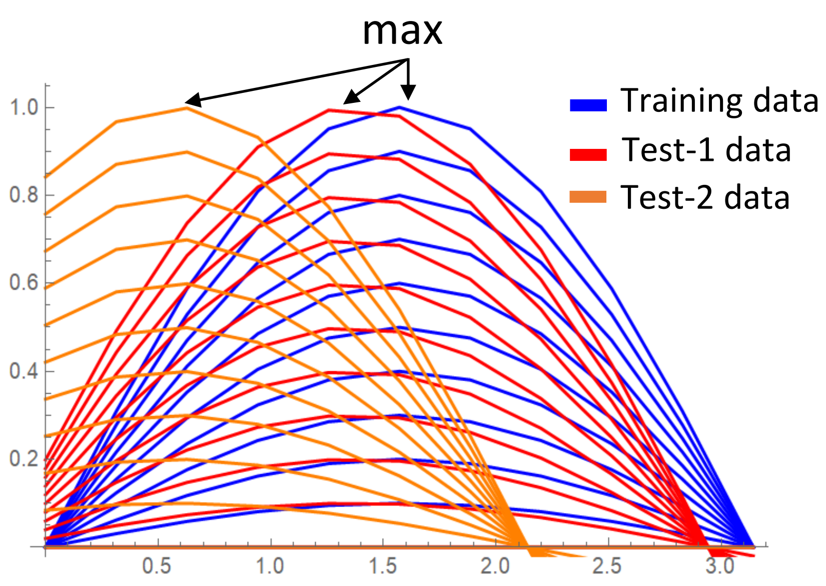
A) MLP and CNN-MLP predicted vs. true plots for training and testing data.

The choice of architecture influences generalization. Generalization is key.

- This is important because this is the exact situation that we are dealing with.

e.g., orange is **experiment**, blue and red are two different **theories**.

A)



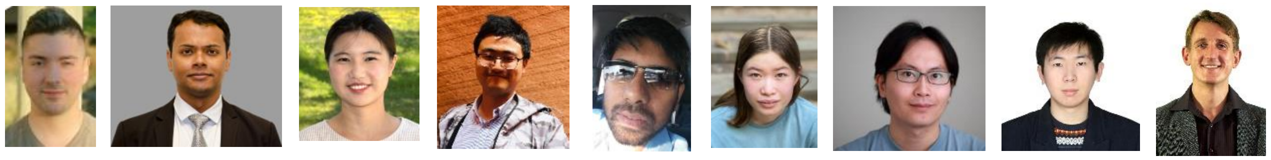
- Theory does not exactly match the experiment.
  - It can get close.
  - Qualitative trends are usually reproduced
- Bottom line: we need models that bridge the gap between theory and experiment.

What about uncertainties?

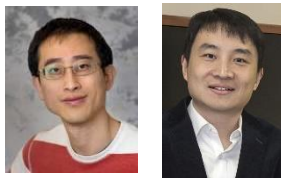
Aleatoric Uncertainty: Due to inherent noise in the data
Epistemic Uncertainty: Due to model uncertainty; the model doesn't know what it doesn't know.

Aleatoric	Epistemic
- Random stochastic differences in the model parameters that were determined in training - Noise in experimental data - Noise in theoretical data	- Correlated variables are not separated. or are just unknown. - The training data and experimental data are too different.
- Ensemble Methods: Train multiple models and use the variance of their predictions. - Bootstrapping: Resample the training data with replacement and train multiple models.	- Test-set benchmarking: Test multiple trained models on data that was not included in the training set. Use the variance of their predictions to estimate uncertainty.

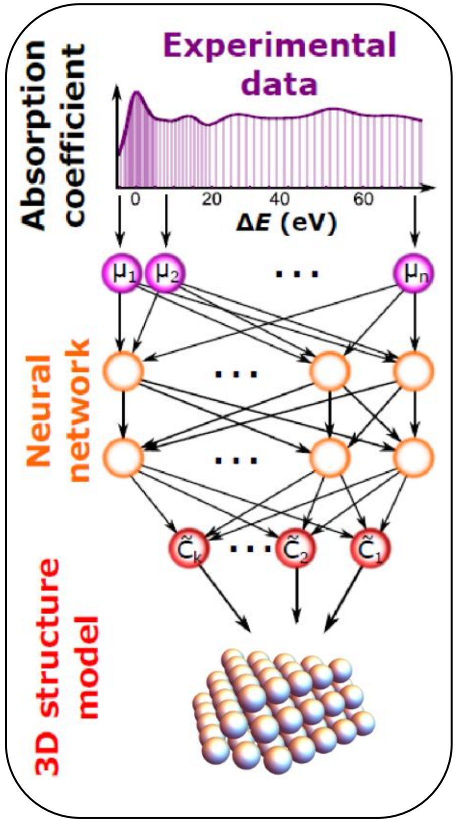
NNs in XANES and EXAFS

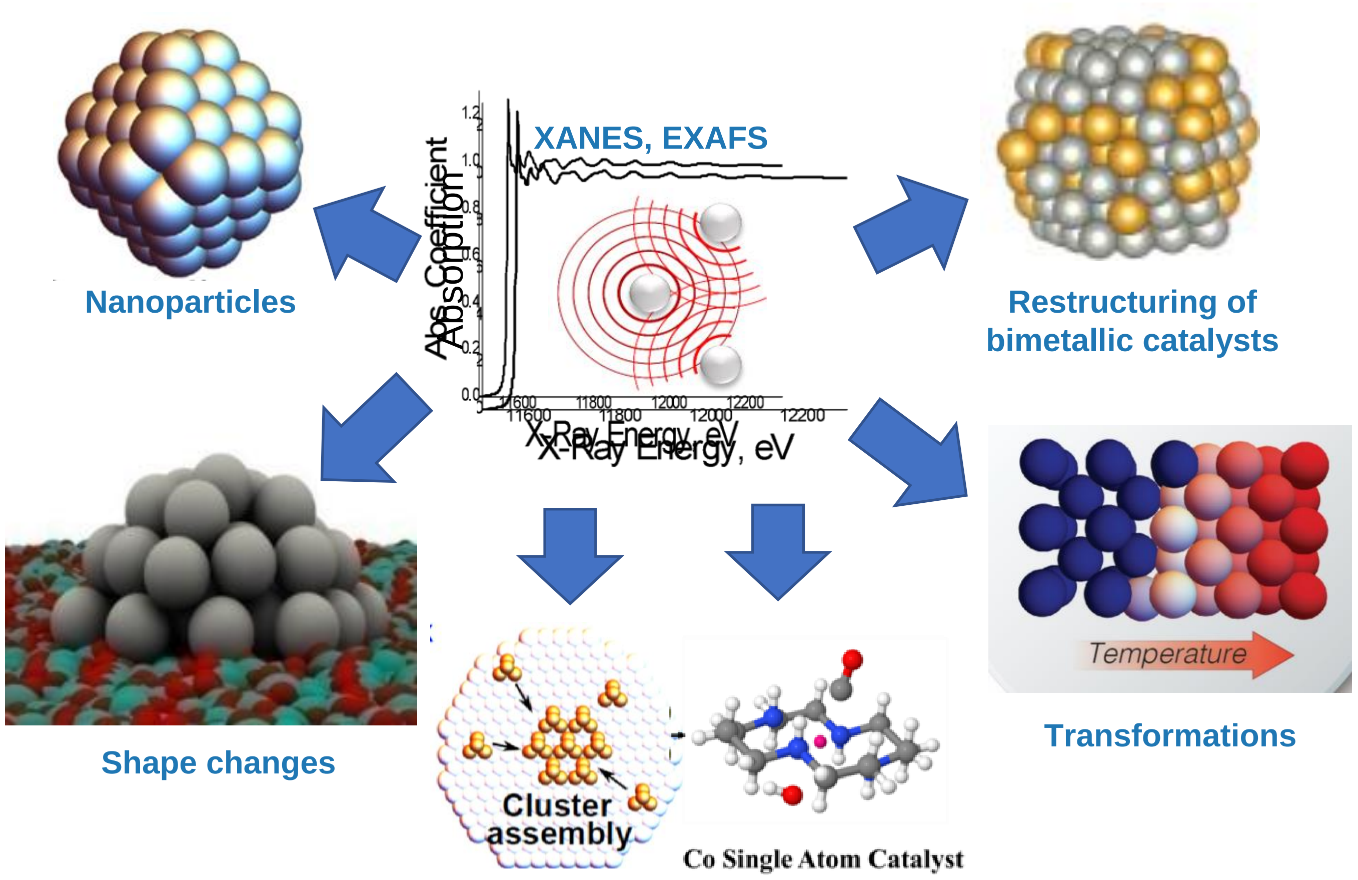


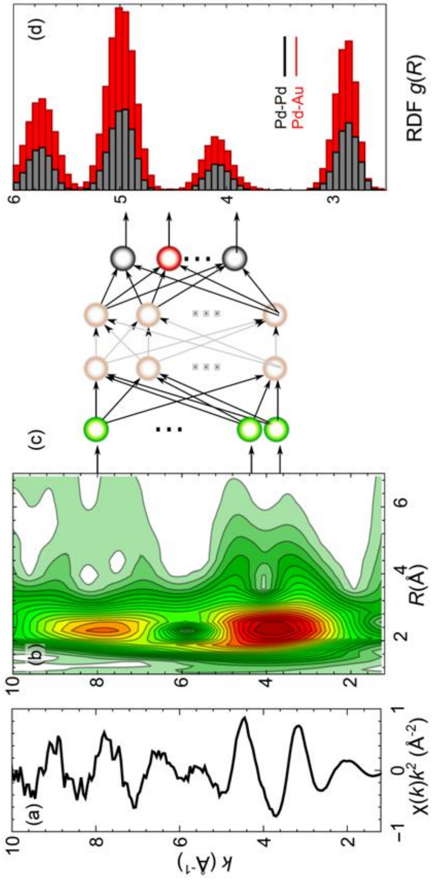
N. Marcella, P. Routh, S. Xiang,
K. Zheng, M. Mahboob, S. D'Halleweyn,
R. Shimogawa, Y. Liu, J. Timoshenko



D. Lu,
Y. Lin

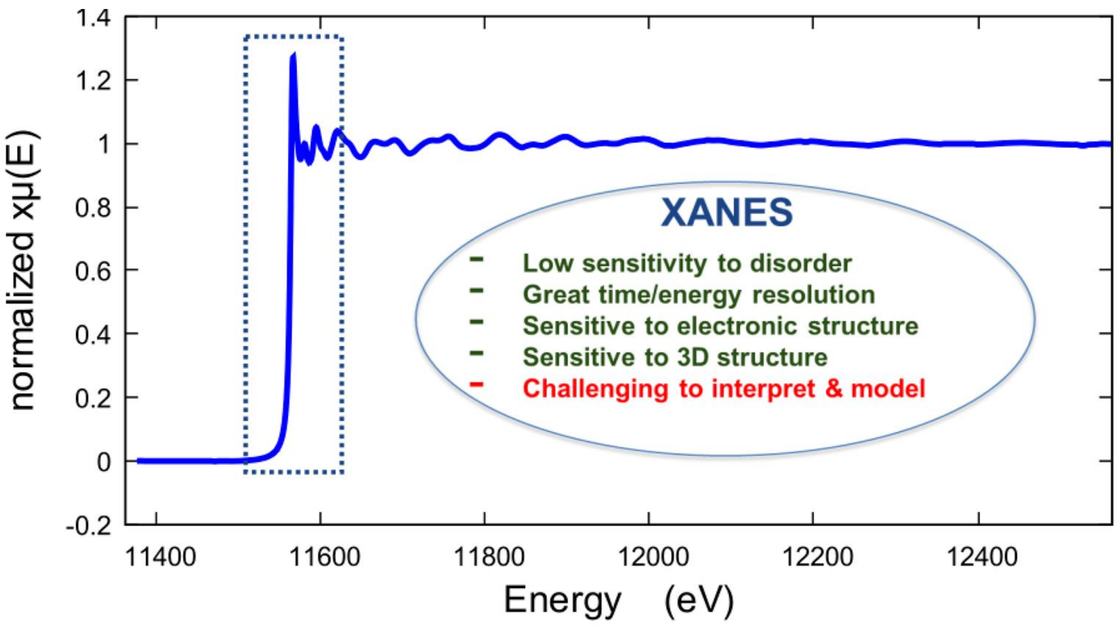




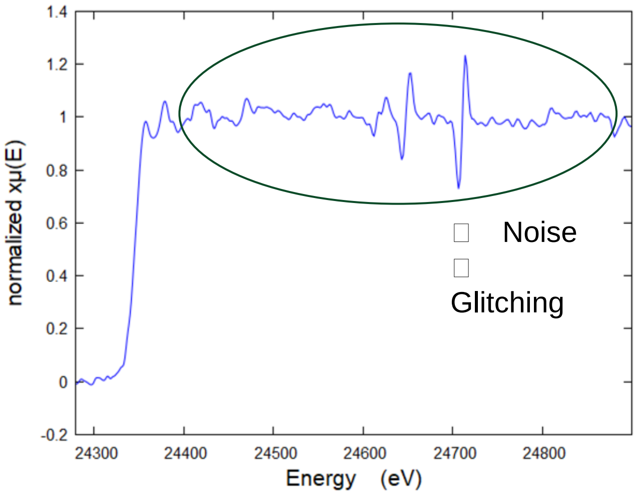


NNs for XANES

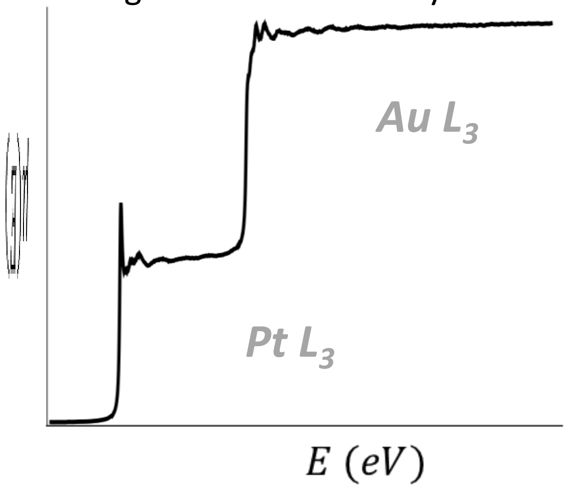
- When EXAFS is inadequate, **XANES** is usually available.



Sometimes…



In this case (AuPt):
- The Au edge cuts off the Pt edge.
- Pt contributions exist in the Au edge that are not easily fit.



Idea: we know the XANES is related to the local structure of the material being investigated. A neural network should be able to learn (model) the spectrum-structure relationship.

Training data:
- We need XANES spectra labeled with some structural characteristics.
- Getting the training set experimentally is impossible.
- We can use ab initio codes (FEFF, FDMNES, or OCEAN) to calculate the spectrum from atomistic models.

Testing data:
- Experimental XANES spectra with good quality EXAFS for labeling structure parameters.

NNs for XANES

Input: aligned, normalized, discretized XANES spectrum $\mu(E)$

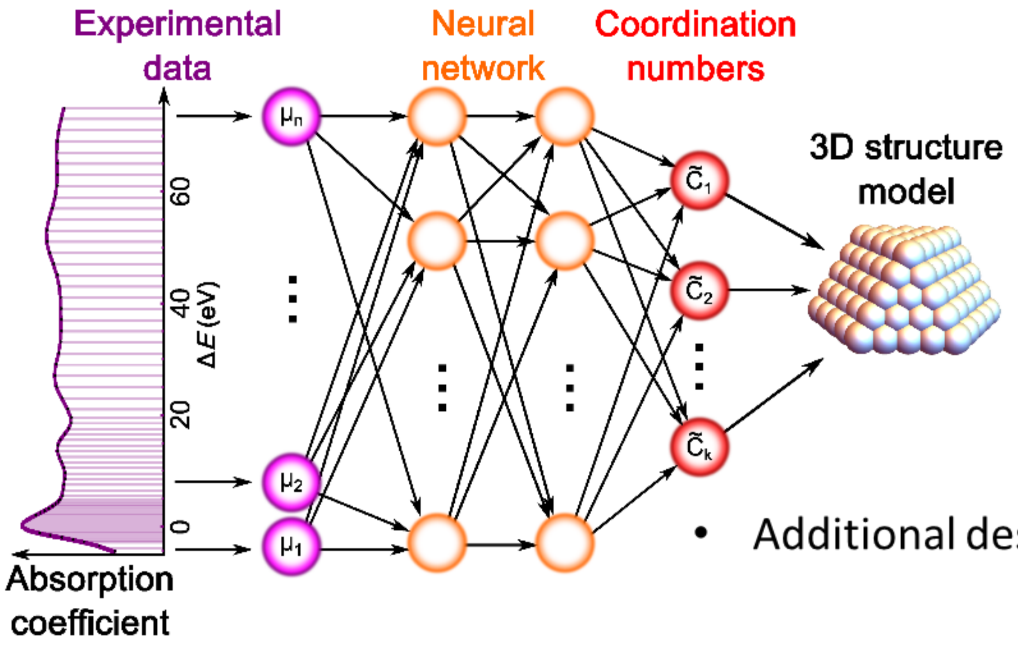
Output: average coordination numbers for the 1st, 2nd, 3rd, … coordination shells

- $C_i = \frac{1}{N_a} \sum_j c_{ij}$

Site-specific coord. numbers

Number of atoms in particle

- Can be used to identify metallic particle size/shape
- Can be directly compared with EXAFS



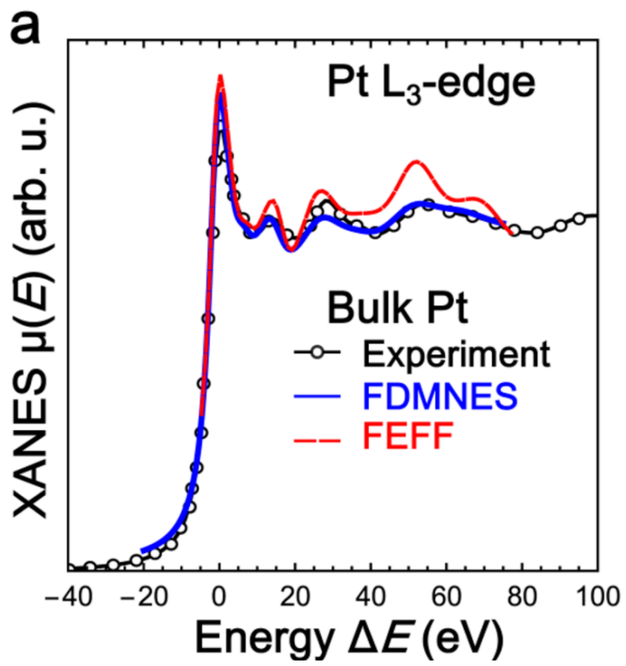
This approach was successful for determination of local structure in Pt nanoparticles:

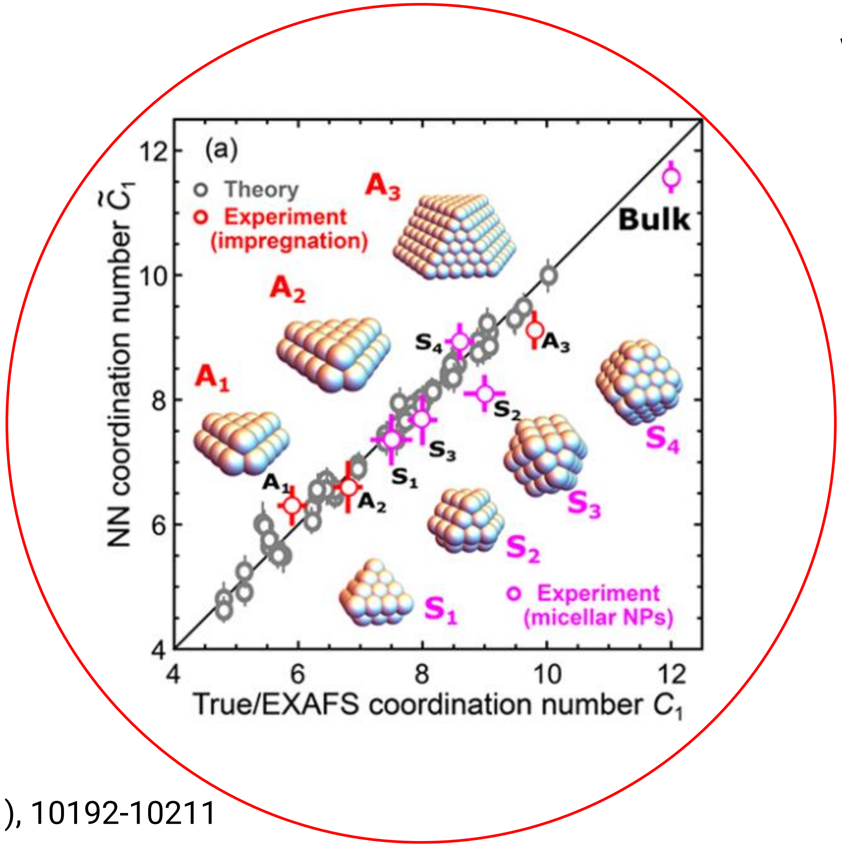
Timoshenko, Frenkel et al,
***J. Phys. Chem. Lett.*, 8, 5091 (2017)**

- Additional descriptor: effective interatomic distance R

What about the agreement between theory and experiment

Pt nanoparticles





Timoshenko, Frenkel. ACS Catalysis **2019** *9* (11), 10192-10211

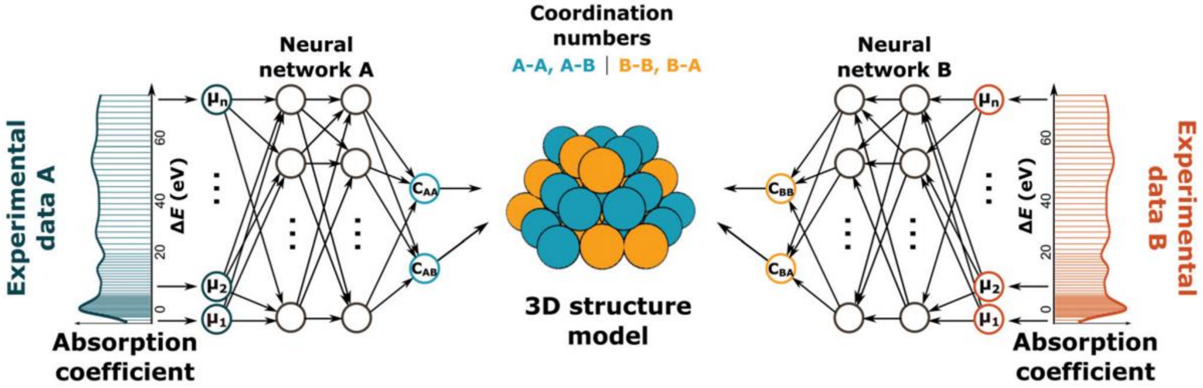
Validation of the model on well-defined systems is required.

If you can’t check the model against “true” values, an alternative validation scheme should be developed.

MLP works fine, no special treatment needed.

Marcella et al. Phys. Chem. Chem. Phys. 22, 18902-18910, 2020.

XANES inversion: spectrum to structure
(XANES)   (coordination number)

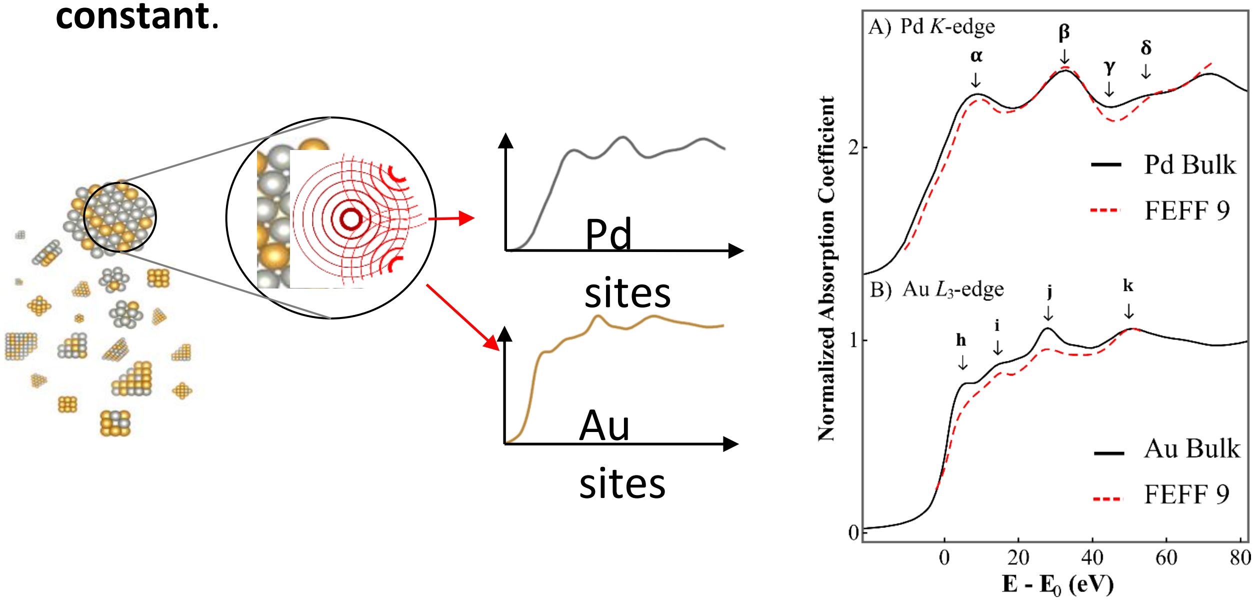


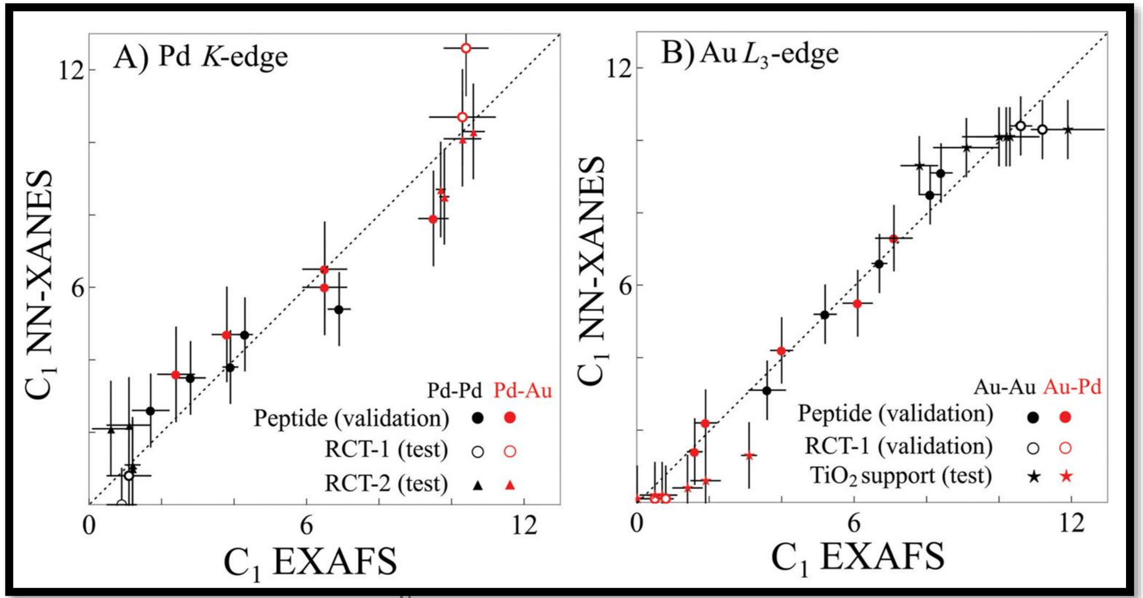
Training data:

- Large set of XANES spectra labeled with "True" coordination numbers.
- We can use ab initio code (FEFF9) to calculate the spectrum from atomistic models.
- **Diversity:** neural networks are interpolative. We include many variations in **particle size**, **shape**, **composition**, **compositional distribution**, **lattice constant**.

Validation / Testing data:

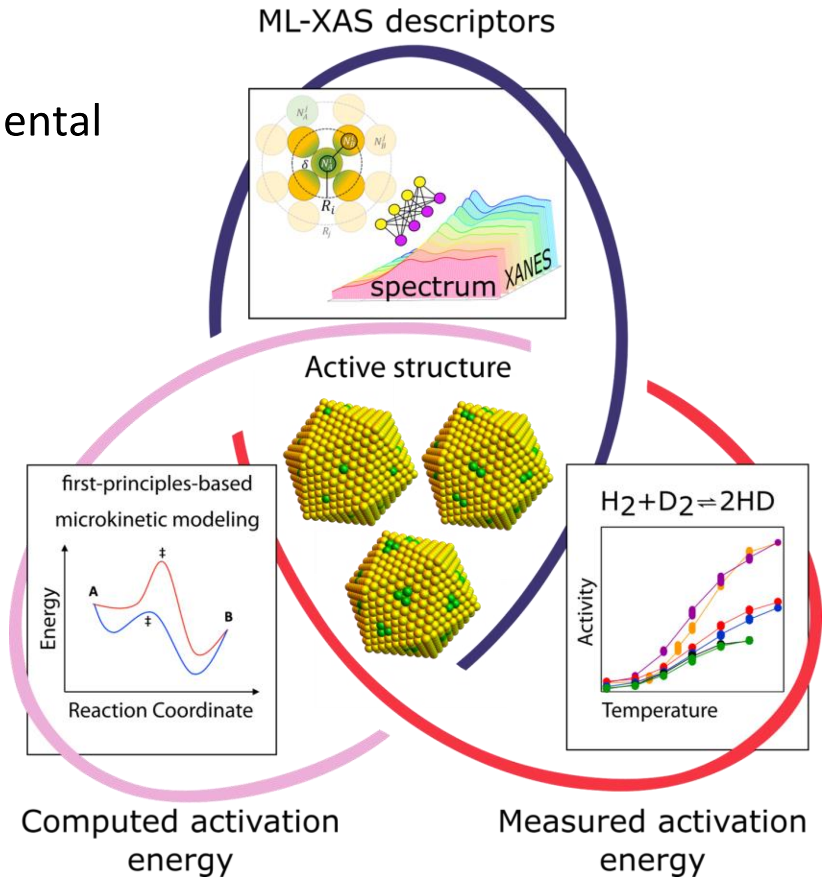
- Experimental XANES spectra with good quality EXAFS for labeling structure parameters.





Decoding reactive structures

Refining the active sites with experimental and theoretical activity modeling

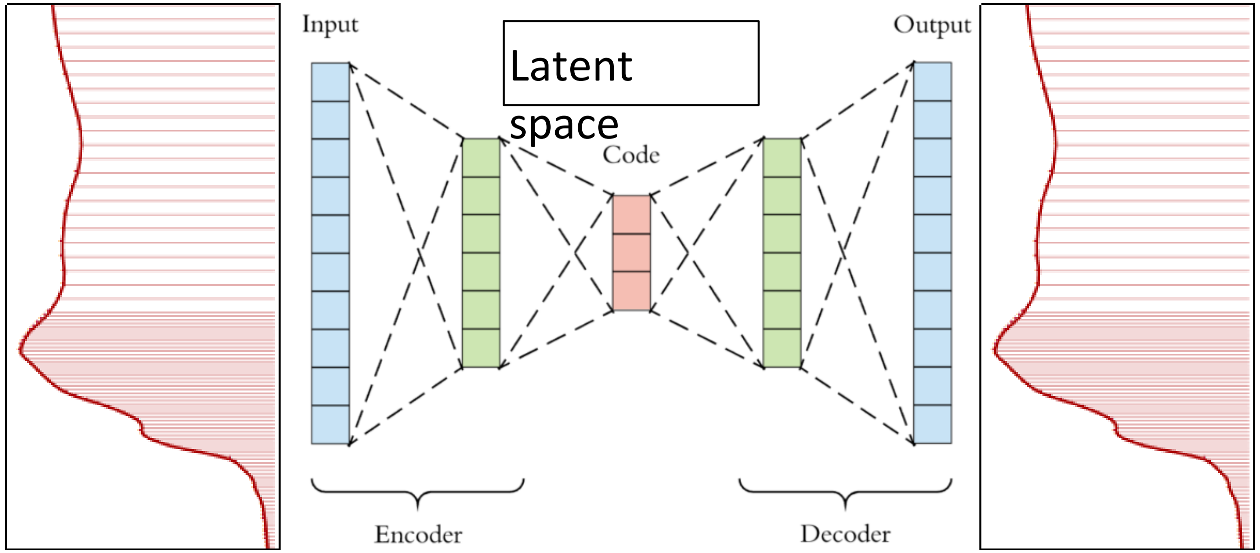


Marcella et al. Nature Commun. 13, 832 (2022)

NNs for XANES: unsupervised learning

Previous machine learning works are supervised, i.e., they require us to label our training data

The future may rely on unsupervised approaches. Routh, Liu, Marcella et al. J. Phys. Chem. Lett. (Perspective) 12, 2086-2094, 2021.



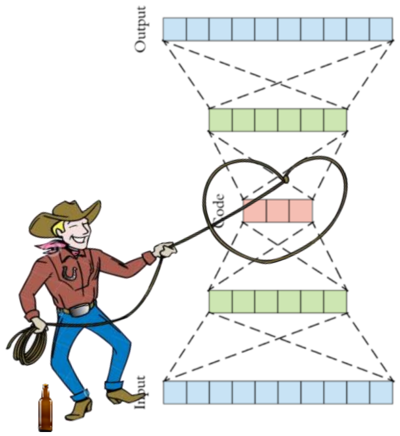
Latent space

"bottleneck"

Key findings:

- Autoencoder creates compressed latent representation of the input space.
- Dimensionality of the latent space is related to the information content in the input space
- Unsupervised and generative modeling allows to learn latent variables and correlate them with physical variables (descriptors)

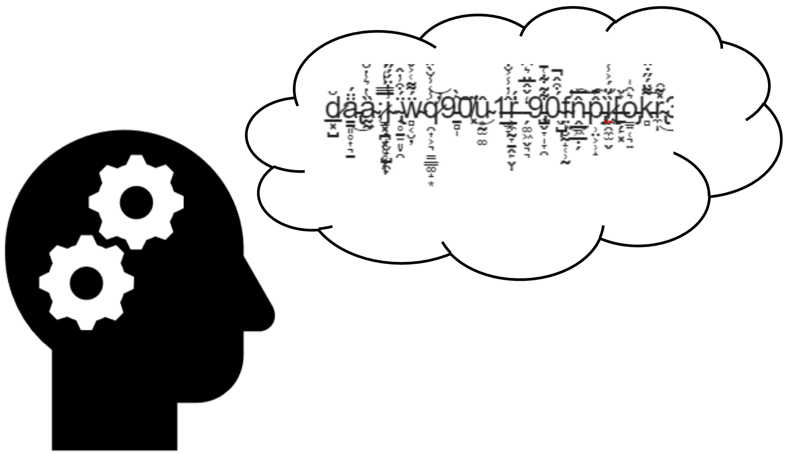
NNs for XANES: unsupervised learning



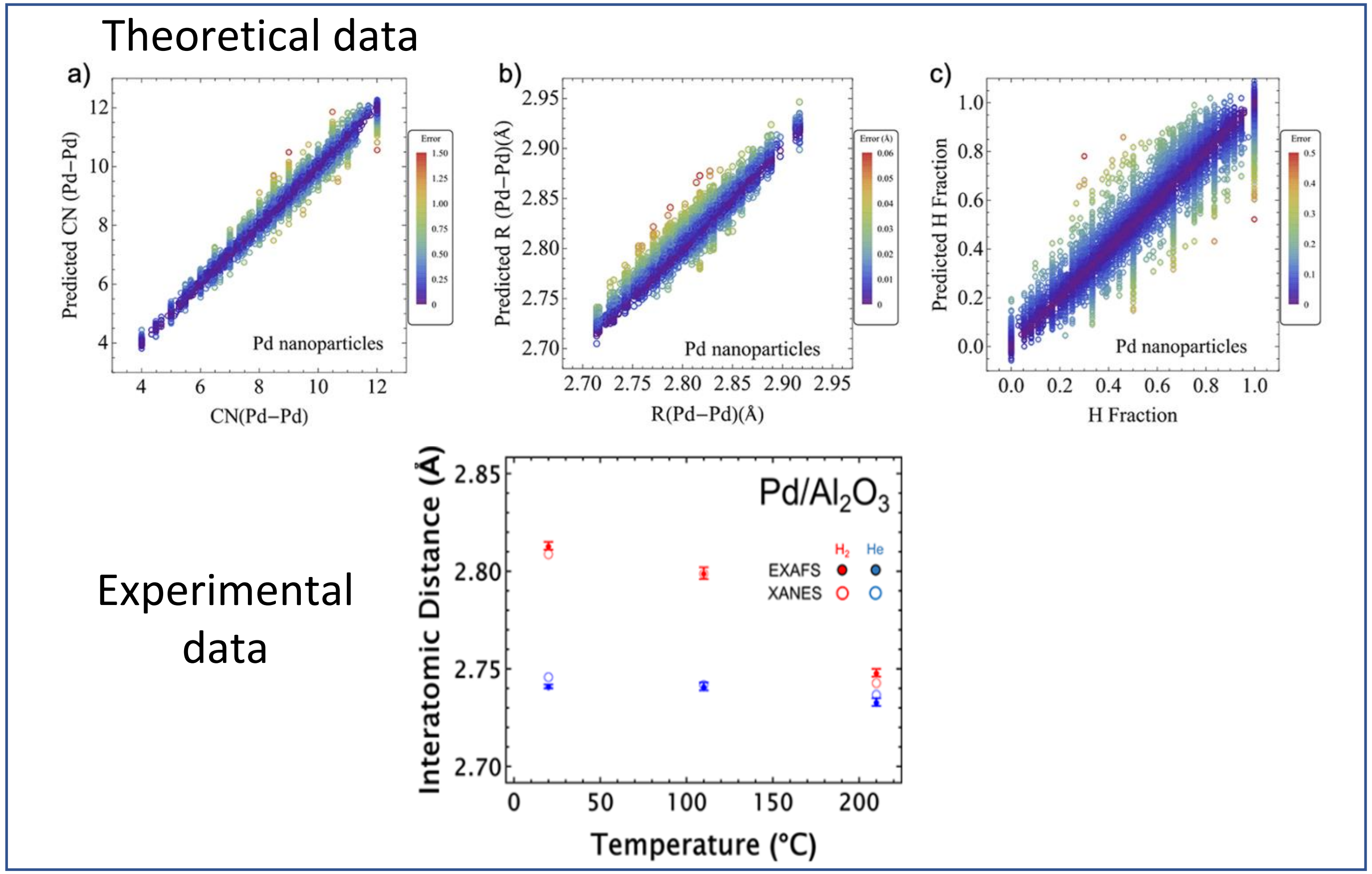
Example latent space

Pd nanoparticles

If we can decode the latent space, we have access to all varying information contained in the XANES spectrum.



Neural Network



The autoencoder has no idea what coordination number, distance, or hydrogen fraction is, however, we find this information stored in the latent space.

NNs for EXAFS

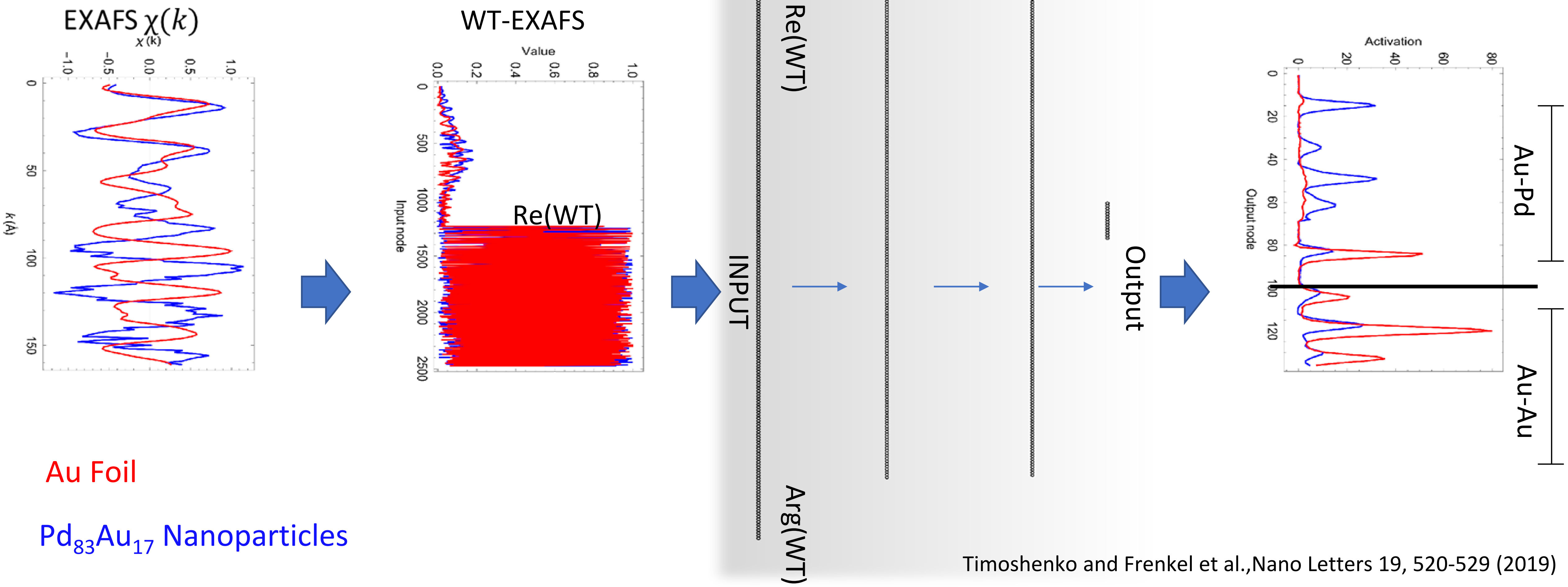
NN-EXAFS – workflow

Use **n**eural **n**etwork to extract *g(r)* from **EXAFS**

*(**absorber**-specific radial distribution function)*

Au in this case

Fully connected Multilayer Perceptron (MLP) using Tanh activations



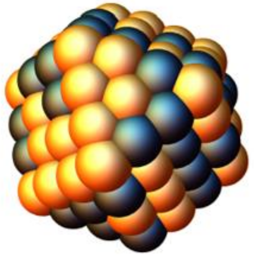
Au Foil

$Pd_{83}Au_{17}$ Nanoparticles

Timoshenko and Frenkel et al.,Nano Letters 19, 520-529 (2019)

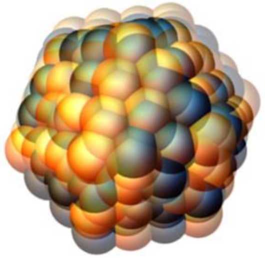
Produce training data:

3D Structure model (xyz coordinates)

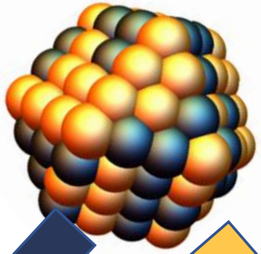


Jiggle

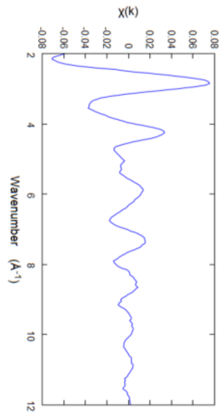
MD for x time steps
@ some temp



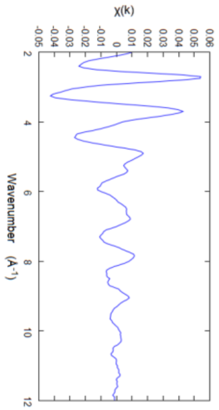
Average structure over x time steps



FEFF
$\chi(k)^{Pd}$



FEFF
$\chi(k)^{Au}$

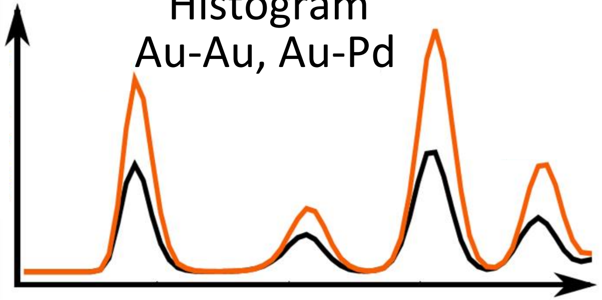


Build in:
1) Particle size
2) Crystallographic structure
3) Lattice parameter
4) Composition
5) Segregation motifs
6) Whatever you want...

Distance matrix for each time step.

Histogram
Au-Au, Au-Pd

g(r)



r (Å)

NN-Model Validation with experimental data

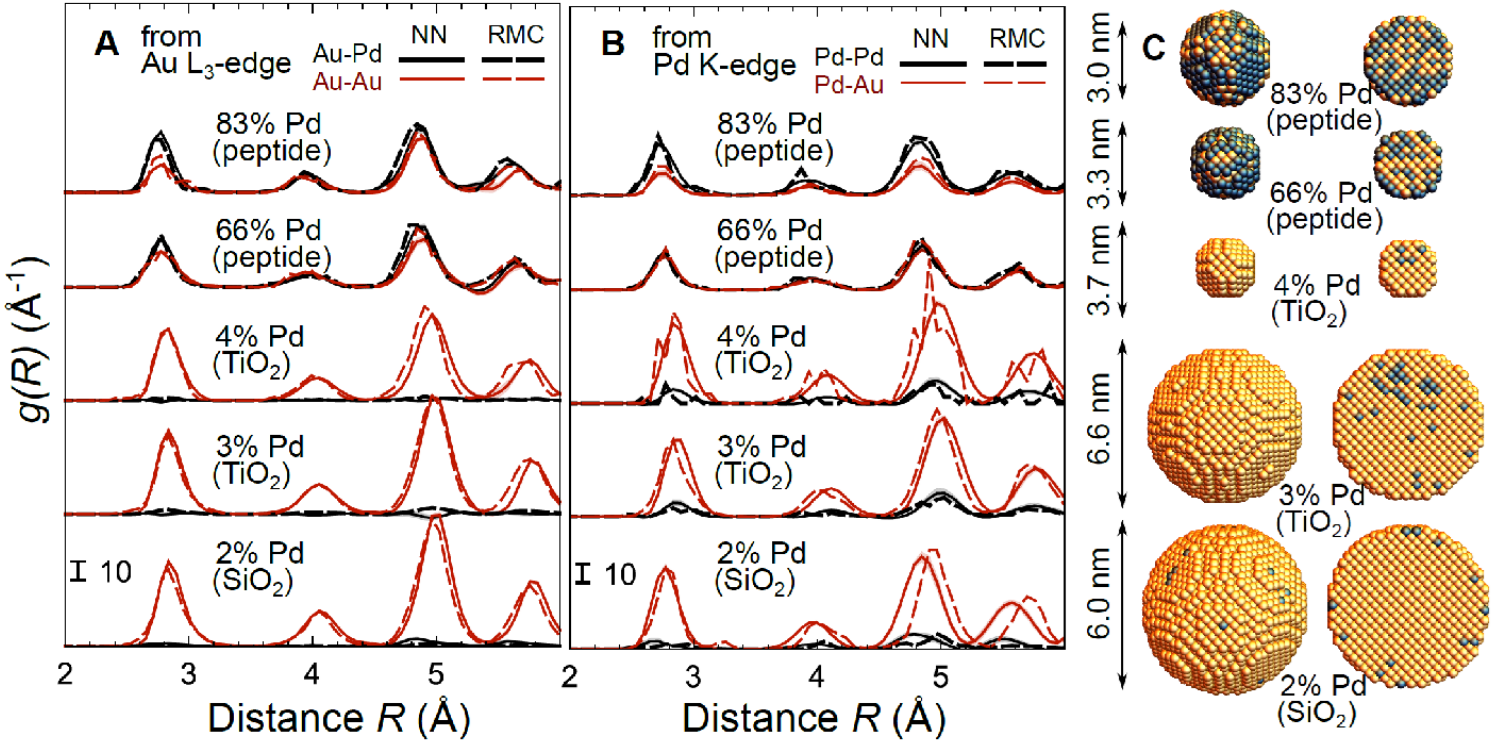
How do we prove that the trained NN-model is valid?

Use **RMC-EXAFS** fitting to obtain g(r) for experimental $\chi(k)$.

- This is computationally expensive, but it's the only way to see if NNs predictions are accurate.

Requires well-known samples (i.e. the starting model used in RMC fitting is a good approximation).

- Complimentary data from TEM, compositional measurements, insights from synthesis.

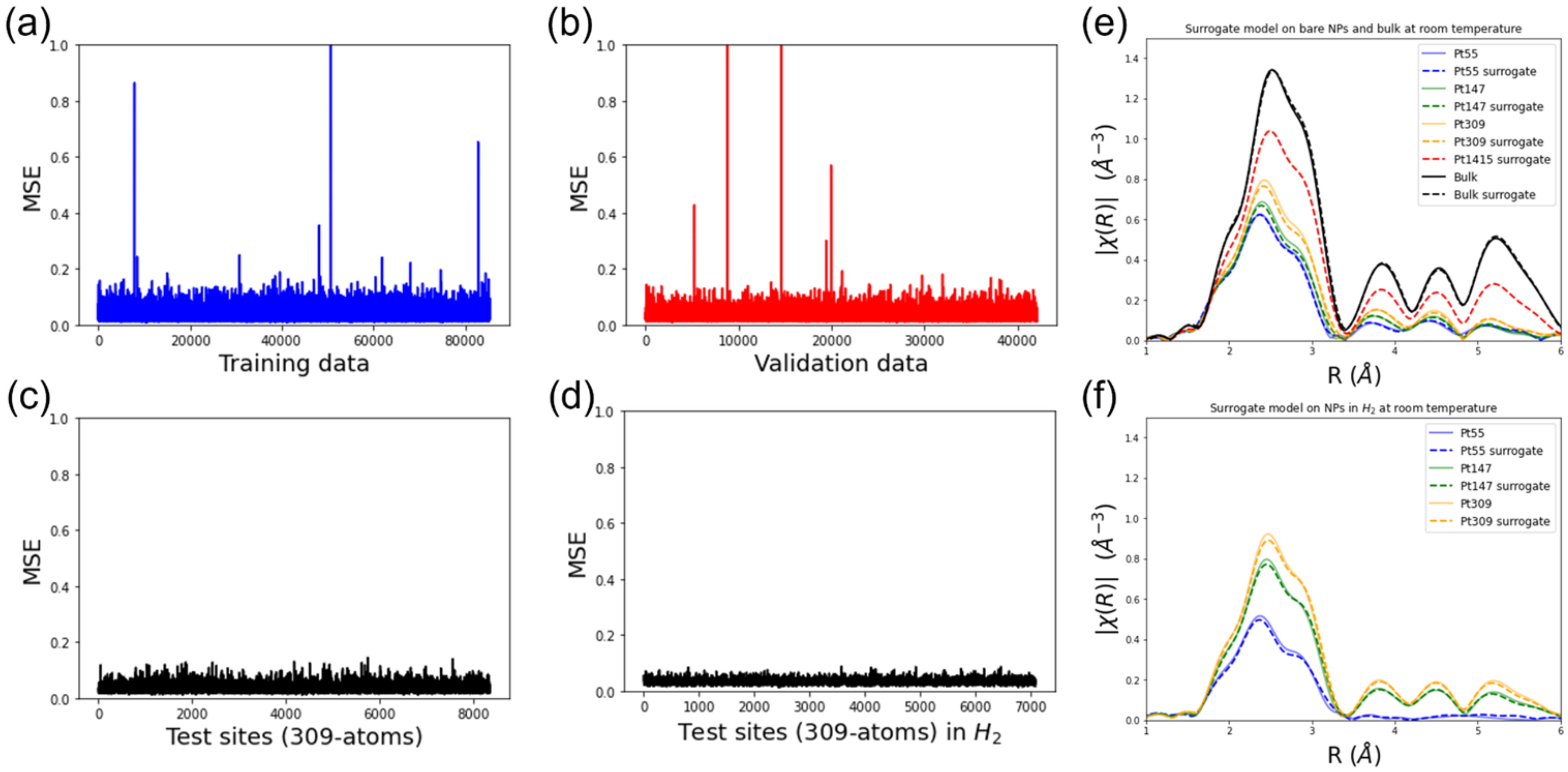


Reconstruction of PRDFs in bimetallic compounds by NN method: Au—Pd and Au—Au (A) and Pd—Pd and Pd—Au (B) PRDFs obtained from Au L_3-edge and Pd K-edge EXAFS for PdAu NPs with different Pd concentrations.

Timoshenko and Frenkel et al.,Nano Letters 19, 520-529 (2019)

NNs as theory surrogate

Unraveling the catalytic effect of hydrogen adsorption on Pt nanoparticle shape-change
arXiv:2306.00901 [cond-mat.mtrl-sci], 2023

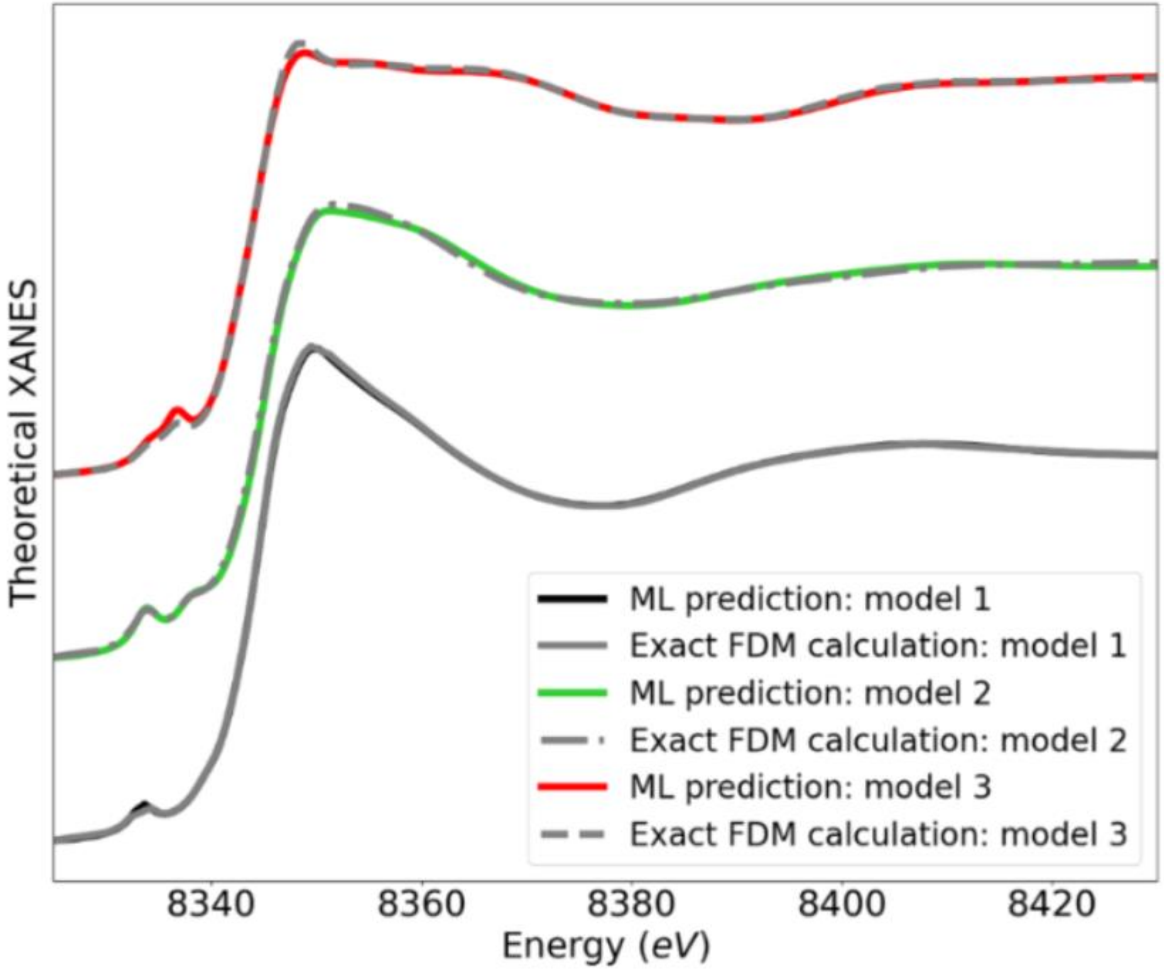


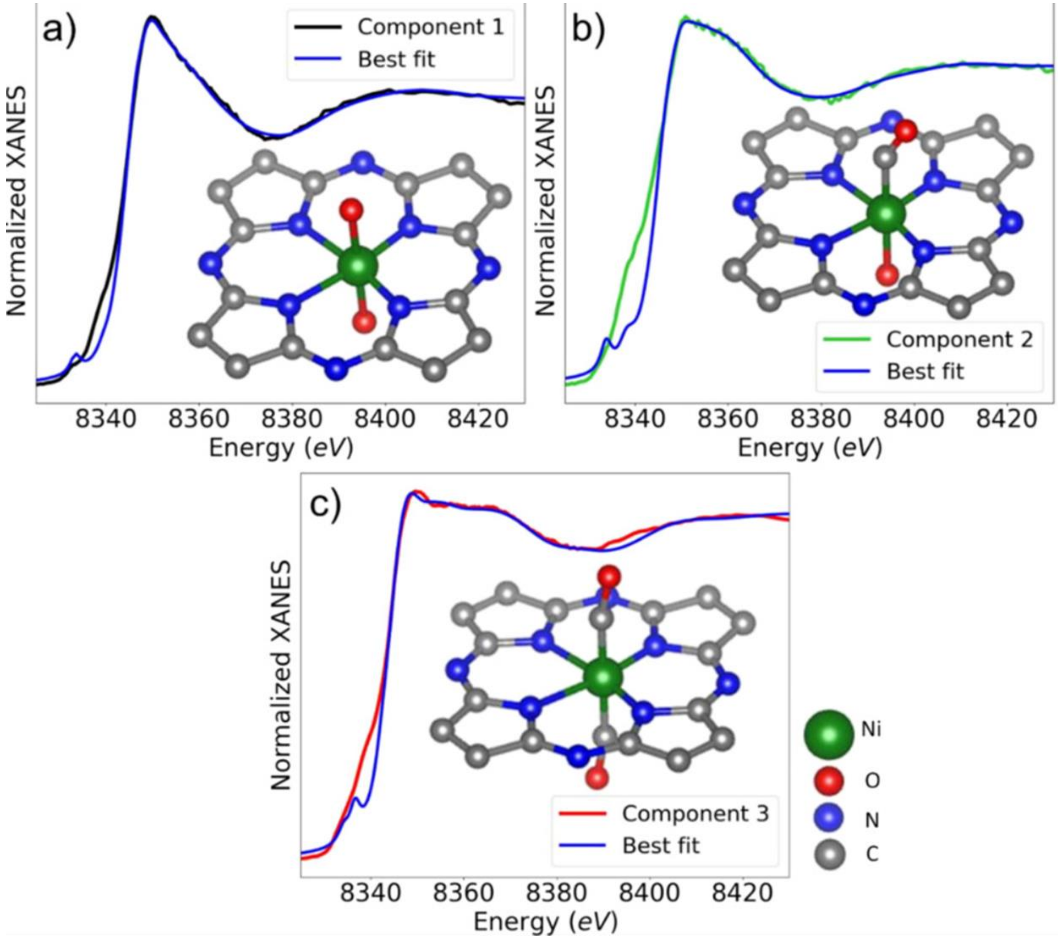
NNs as theory surrogate

Input → Output

Structure → Spectrum

A. Martini et al. J. Am. Chem. Soc.
2023, 145, 31, 17351−17366

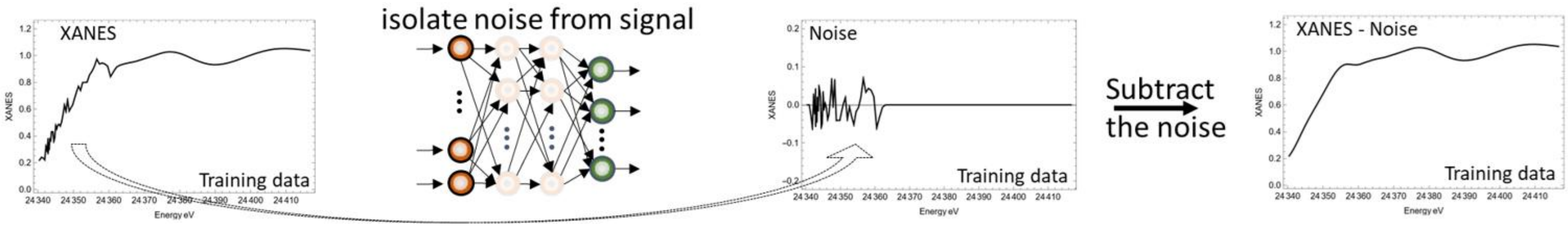




NN for denoising

Input → Output

XANES Noise



NN for counting

Input → Output

xyz CN

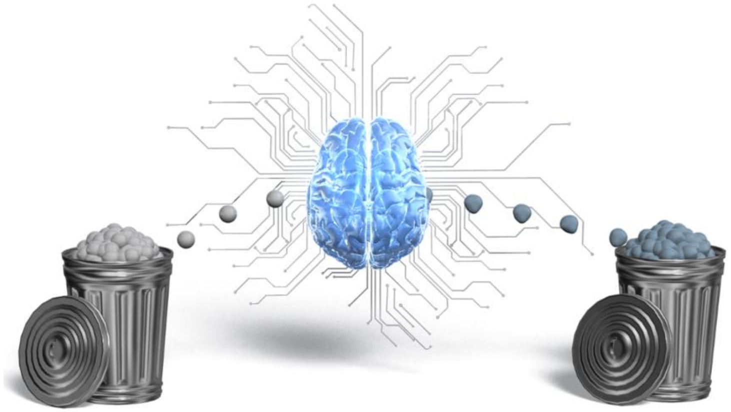
DIY

Implementing a machine learning algorithm has been made relatively easy by the various software packages and *billions* of online tutorials, books, videos, etc…

e.g.
Mathematica https://www.wolfram.com/mathematica/
Python (most common, various libraries) https://www.anaconda.com/
MATLAB https://www.mathworks.com/products/matlab.html

1) Frame your observations in terms of generating factors. Decide what parameters are reasonable to extract from the signal.

2) The hard part is related to the **training data**, because at the end of the day, "garbage in, garbage out."



Toyao et. al, ACS Catal. **2020** *10* (3), 2260-2297

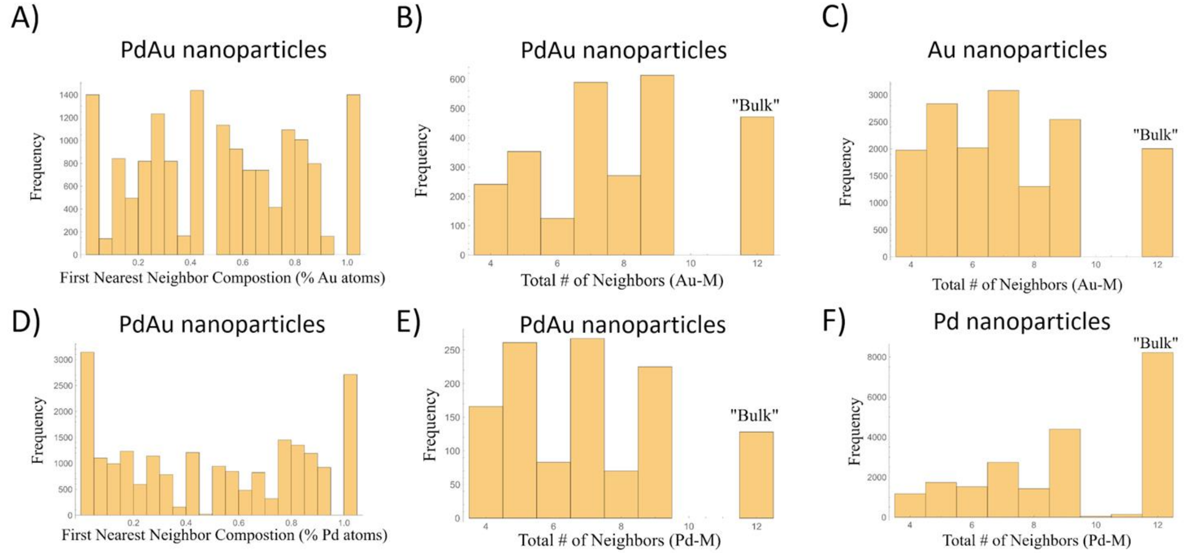
If we want to "Use neural network to extract descriptors from the XANES and EXAFS", then we need:

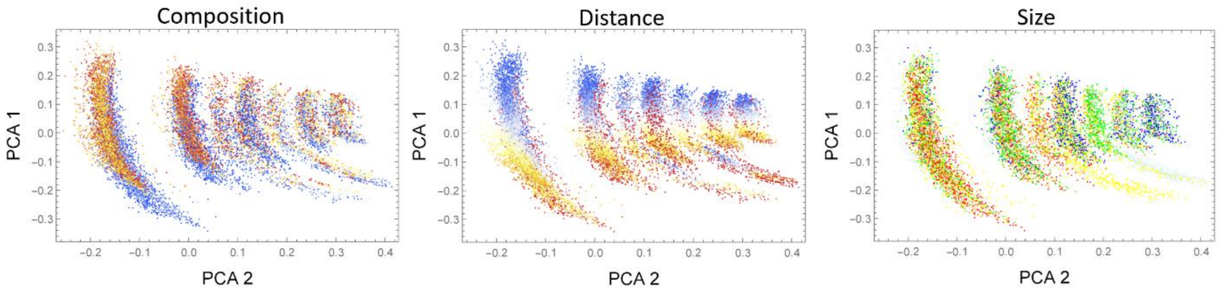
1) to have training data for which we know this relationship.
2) Determine the best way to preprocess the data
3) Find a way to validate the NN model – ideally using experimental data for which we know the relationship.

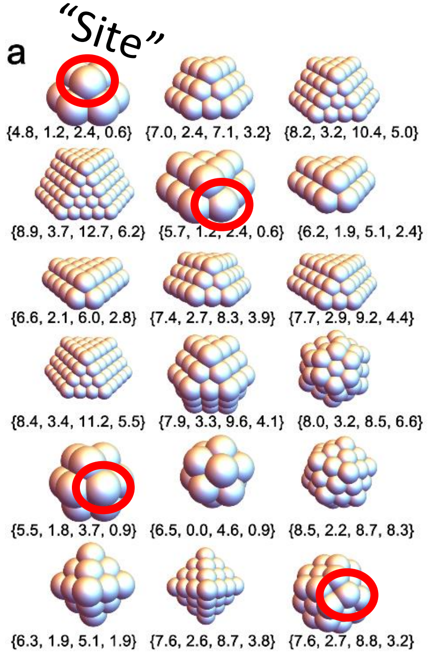
Training data

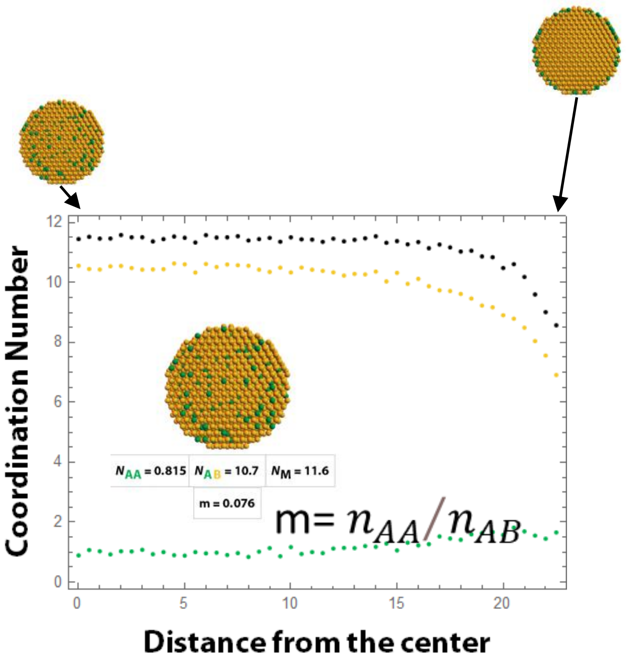
What do you expect the real system to look like, how might it behave, is it dynamic? You must create a training dataset that interpolates the entire space of possibilities

e.g., training data PdAu XANES NN.









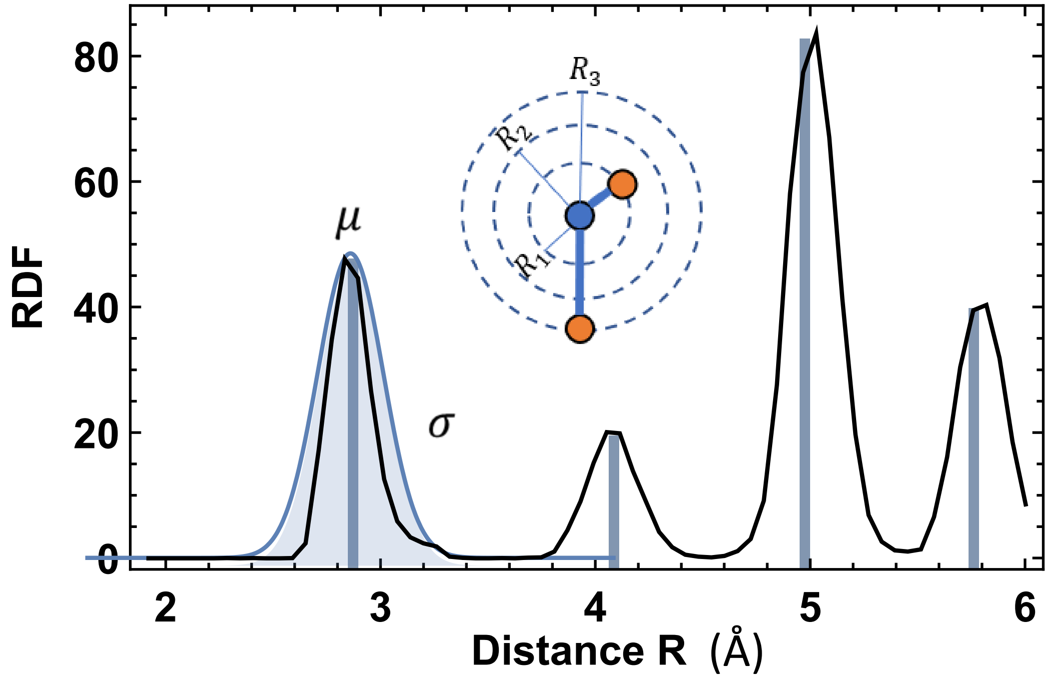
Configuration space is huge in bimetallic NPs

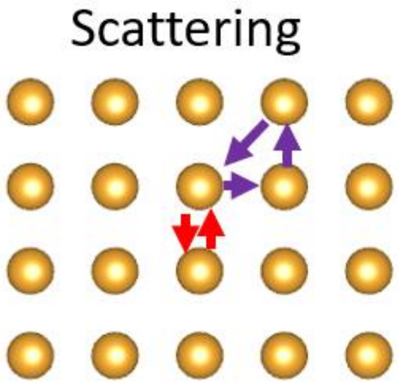
1) calculate site-specific XANES spectra for a **few** particle models (for each site we know also the site-specific coord. numbers c_i and interatomic distance R)

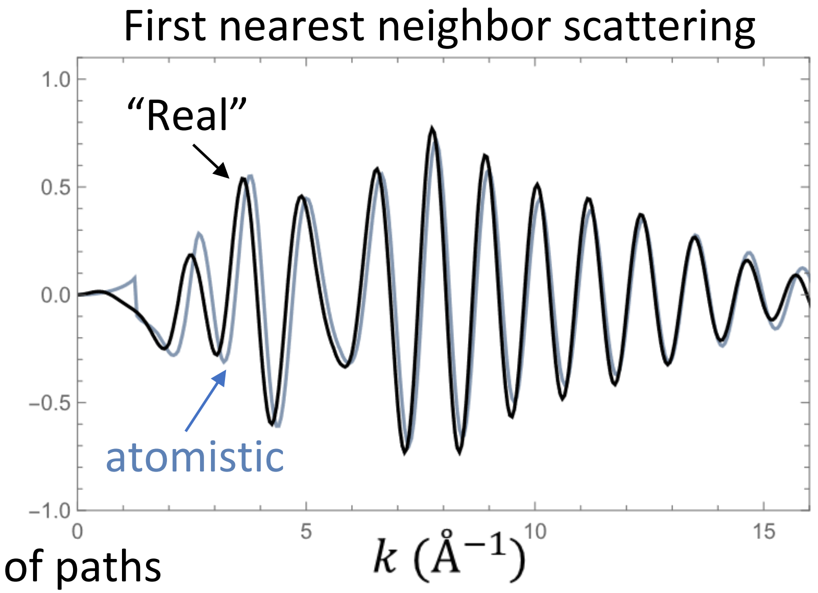
2) Pick randomly n of calculated spectra, and generate **artificial averaged spectrum** as $\mu(E) = \frac{1}{n}\sum_{j=1}^{n}\mu_j(E)$ Corresponding averaged coord. number is $C_i = \frac{1}{n}\sum_{j=1}^{n} c_{ij}$

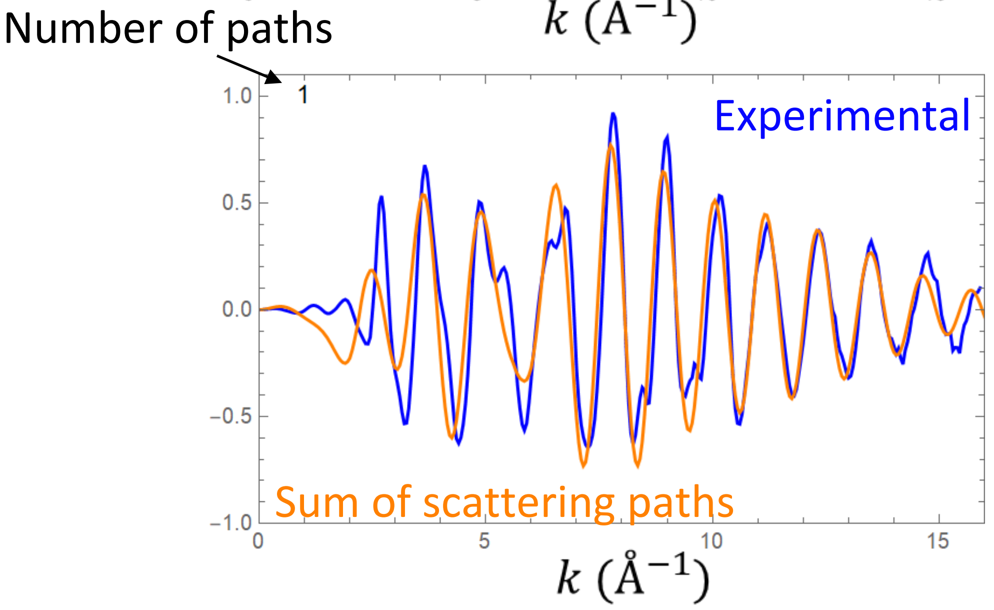
3) Repeat (2) as many times as needed, to generate **thousands of training examples**

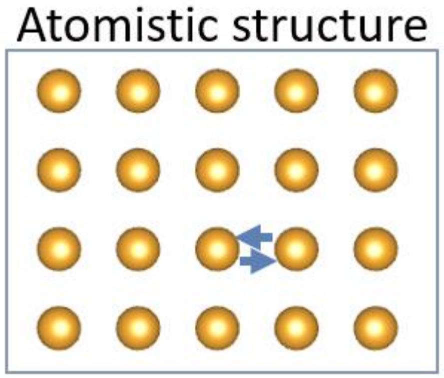
If you are looking at EXAFS, don't forget the training space must include dynamic and static disorder!

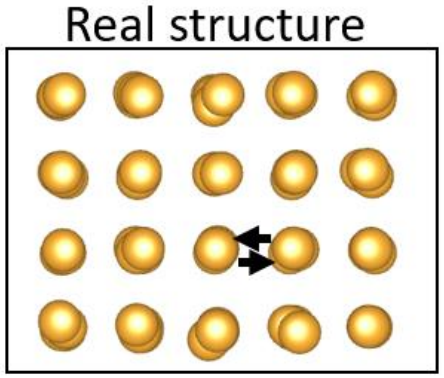








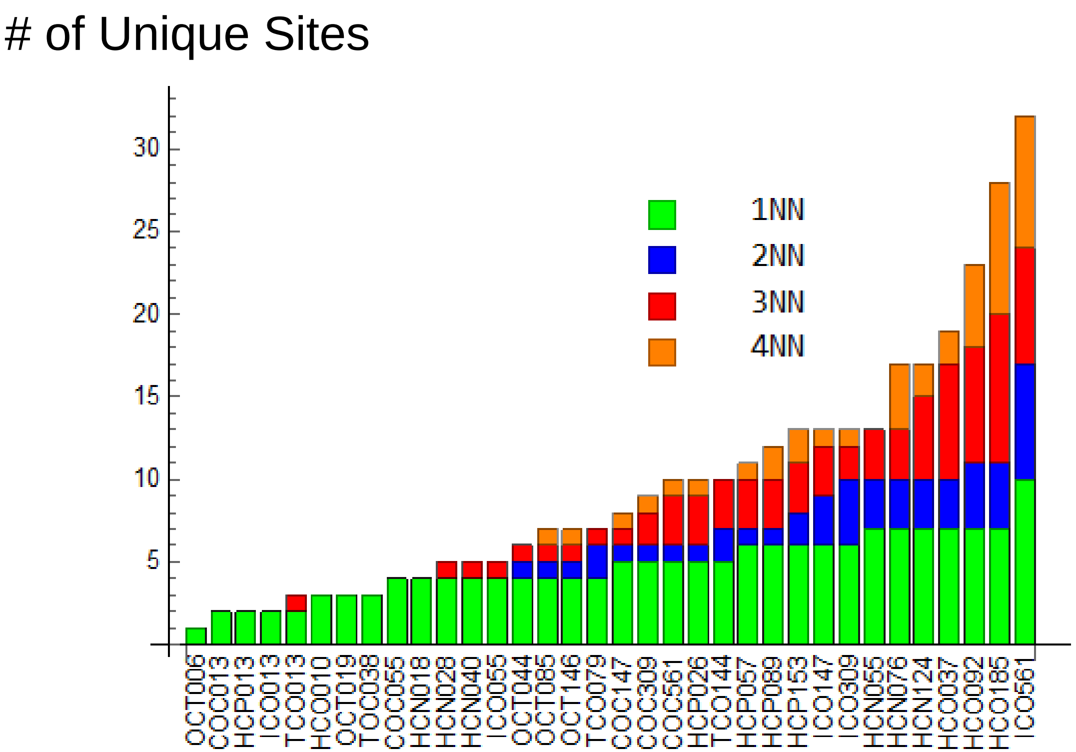


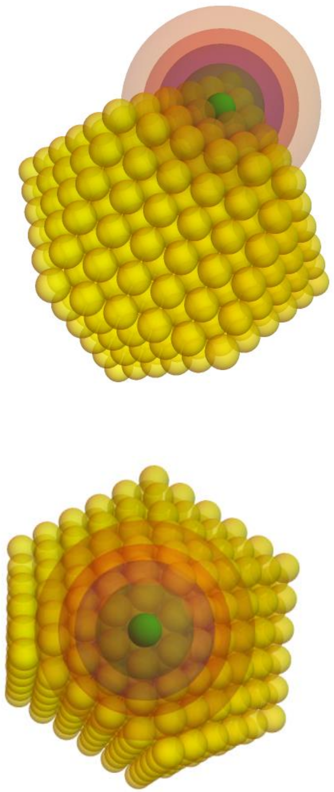


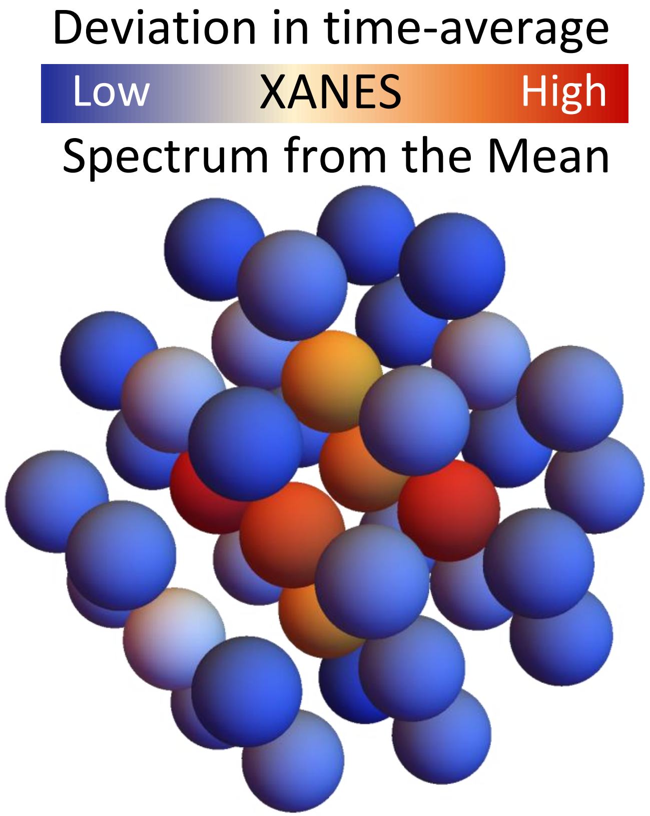
In some cases, you can decrease the size of the training data set by performing a sensitivity analysis. For example, some nanoparticles may have symmetrically-equivalent sites that can be approximated by one unique site. If interested in dynamics, some sites may be less sensitive to changes in thermal vibration than others, and thus one could sample them less.

Static cases

Dynamic cases







“Unique” depends on the size of the radius considered around the site