MRS MATERIALS RESEARCH SOCIETY

Advancing materials. Improving the quality of life.

UCLA

How Machine Learning can Assist in Powder X-Ray Diffraction

Nathan Szymanski

Spring MRS 2026

CH08 Tutorial

April 26, 2026

MD^2 LAB

MATERIALS DESIGN THROUGH DYNAMICS

Samueli School of Engineering

Outline for this tutorial

10:30-10:45	Basics of powder X-ray diffraction (XRD)
10:45-11:00	How to simulate *realistic* XRD patterns
11:00-11:15	Conventional approaches to XRD analysis
11:15-11:30	Introduction to machine learning (ML)
11:30-12:00	How ML can be used for XRD analysis

Follow along with Google Colab

MRS CH08 Tutorial on ML for powder XRD

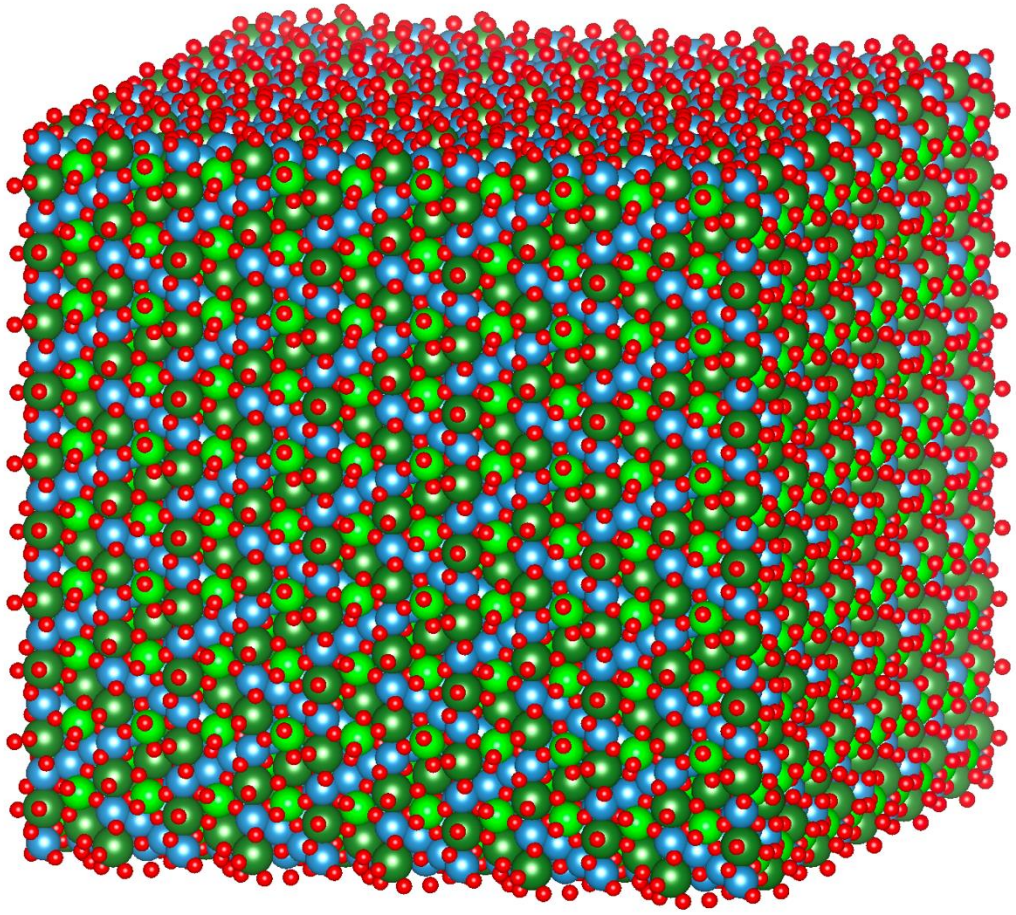
Open in Colab

Module	Topic	Open in Colab
01	Pattern Generation	▶ Open
02	Conventional Phase ID	▶ Open
03	ML-based Phase ID	▶ Open

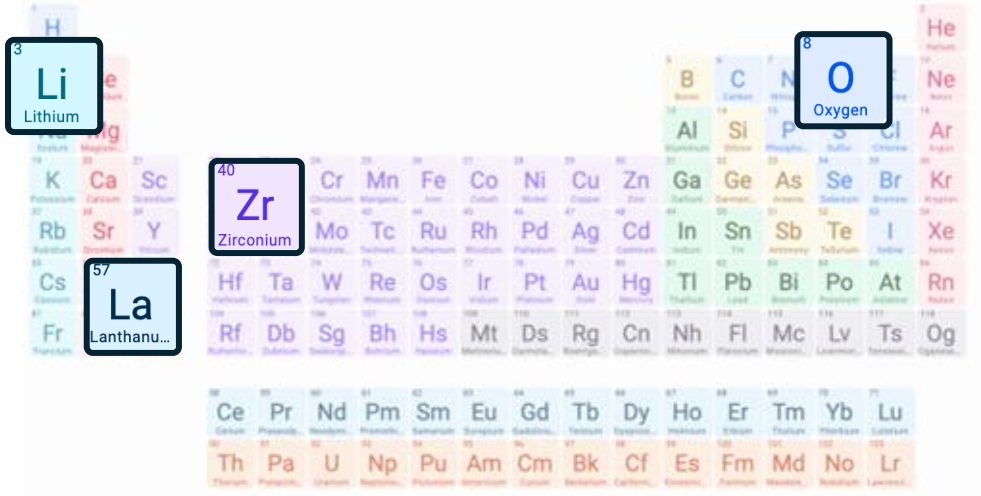
https://github.com/Szymanski-Group/MRS_CH08_Tutorial

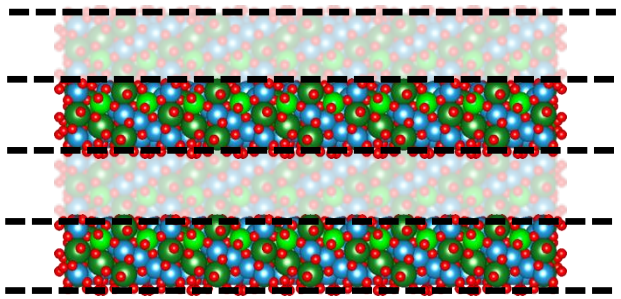
Change runtime type
Select **T4 GPU**

The structure of crystalline materials



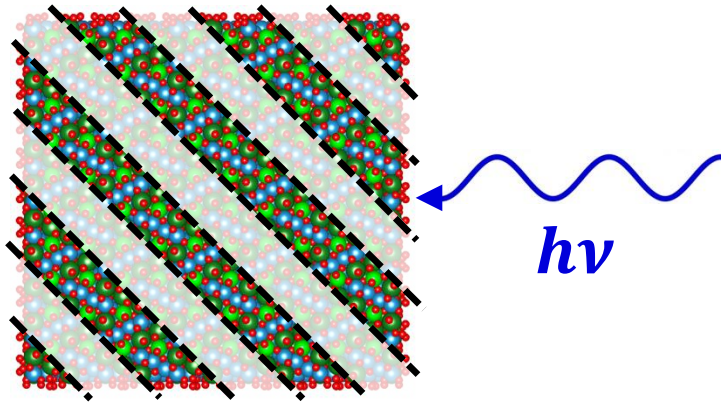
Li La Zr O



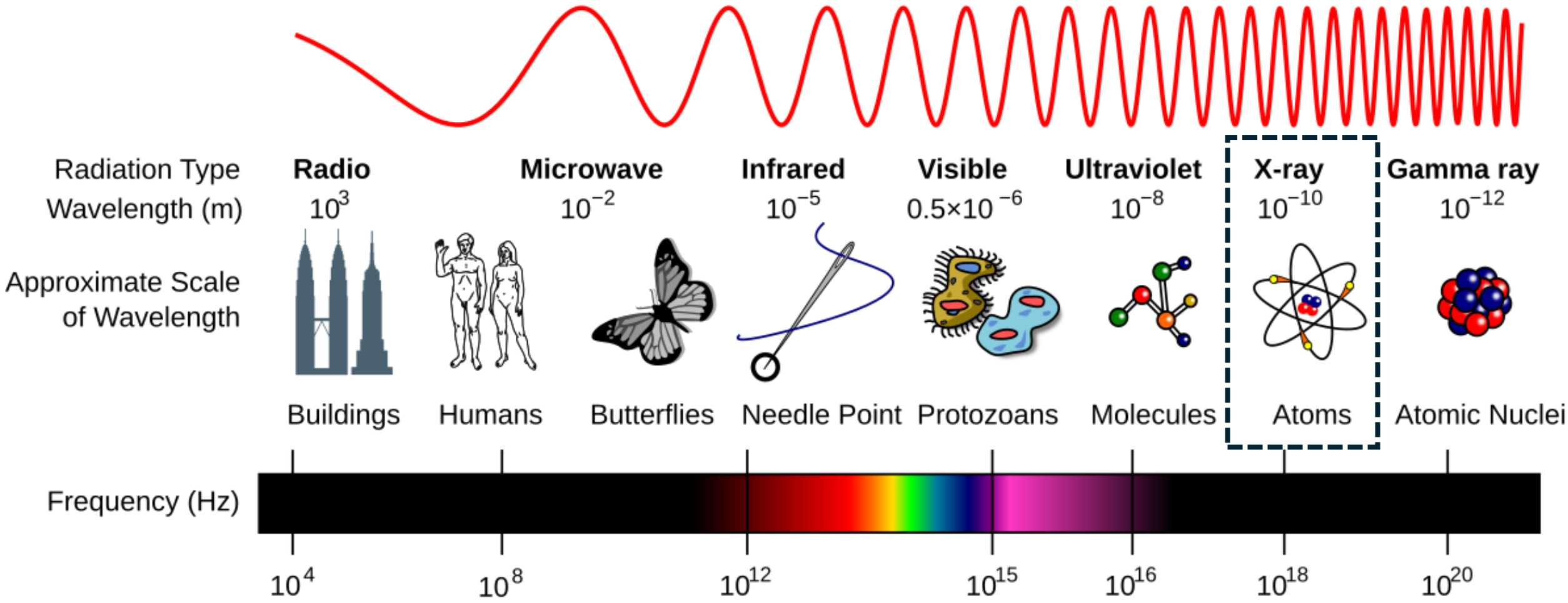


These structures have ***periodicity***

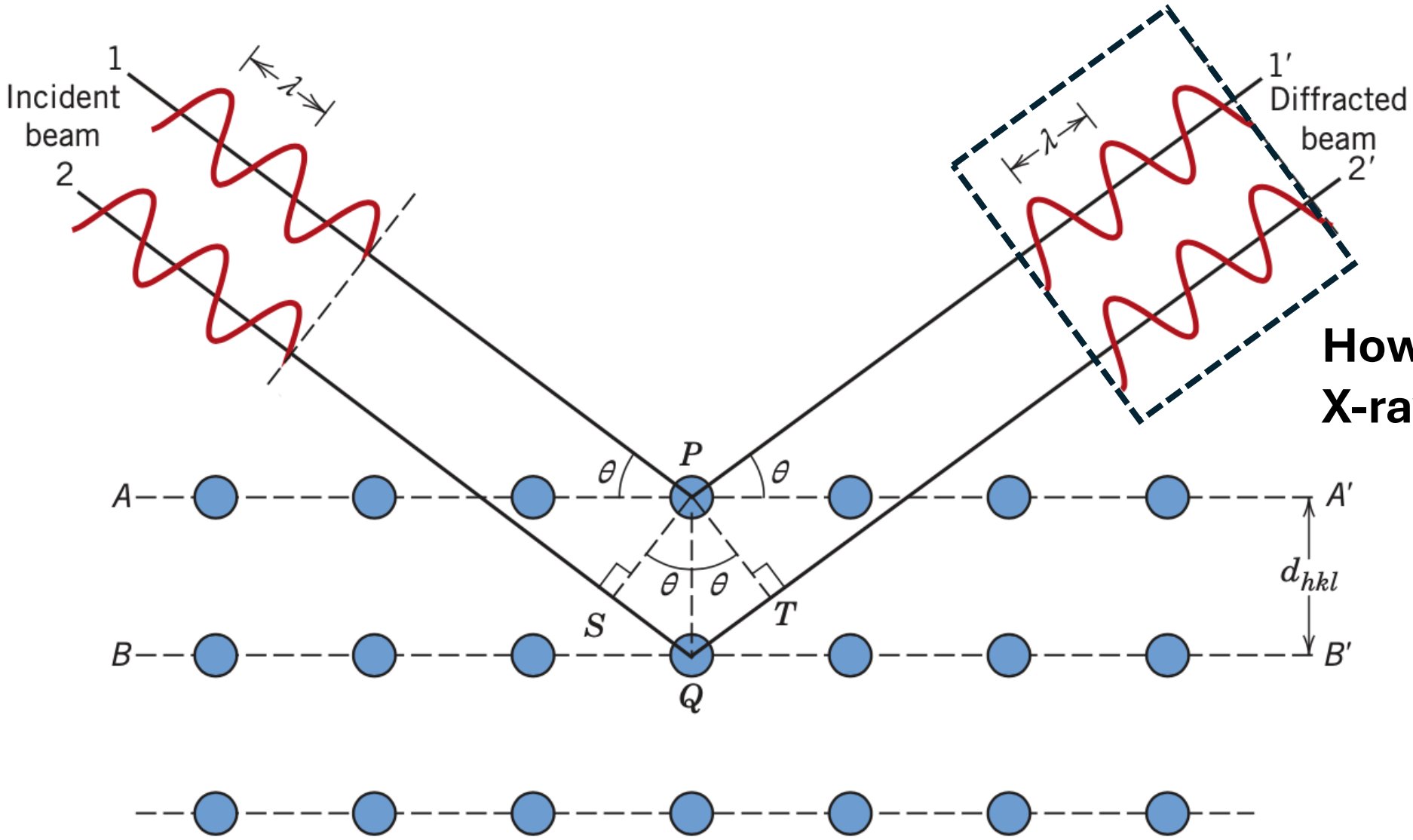
Diffraction exploits this



X-rays: the right wavelength to probe structure

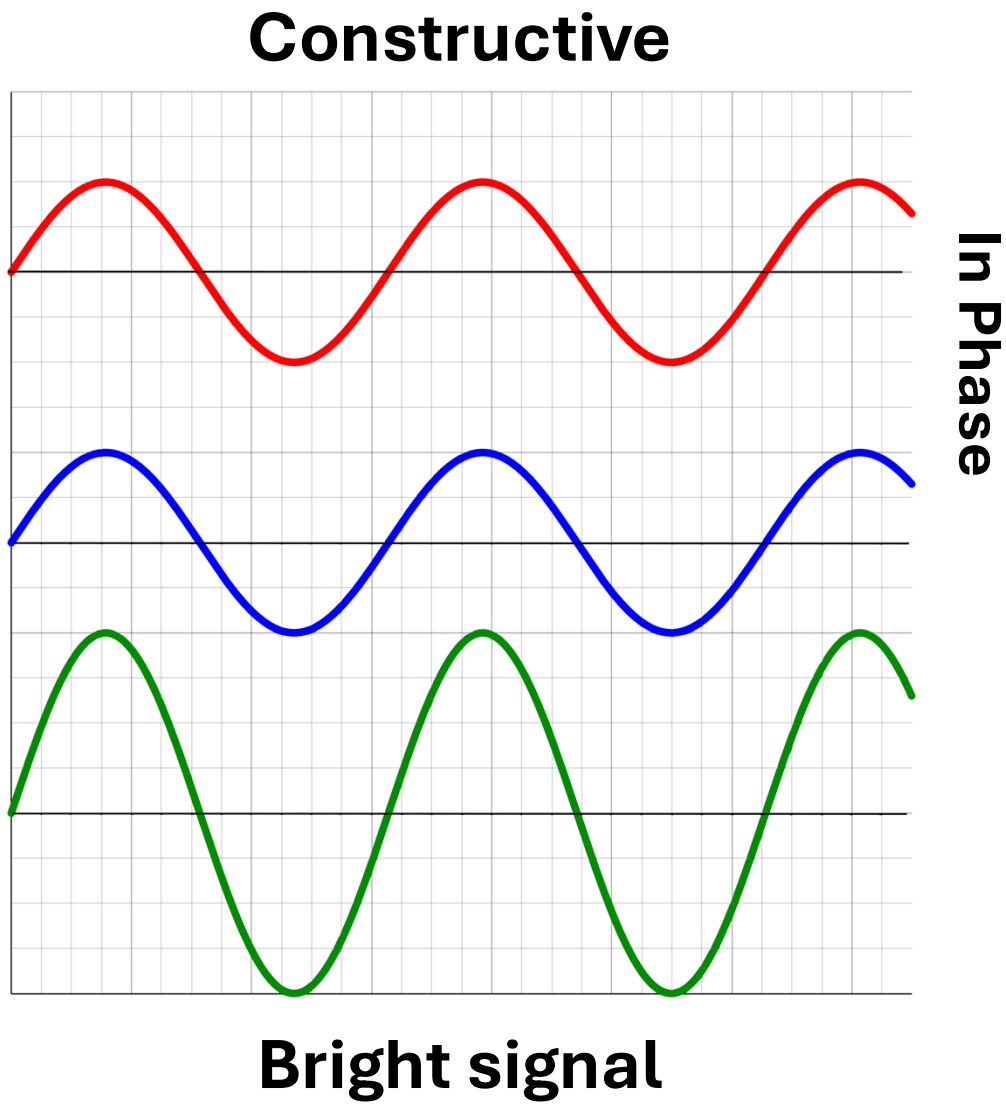


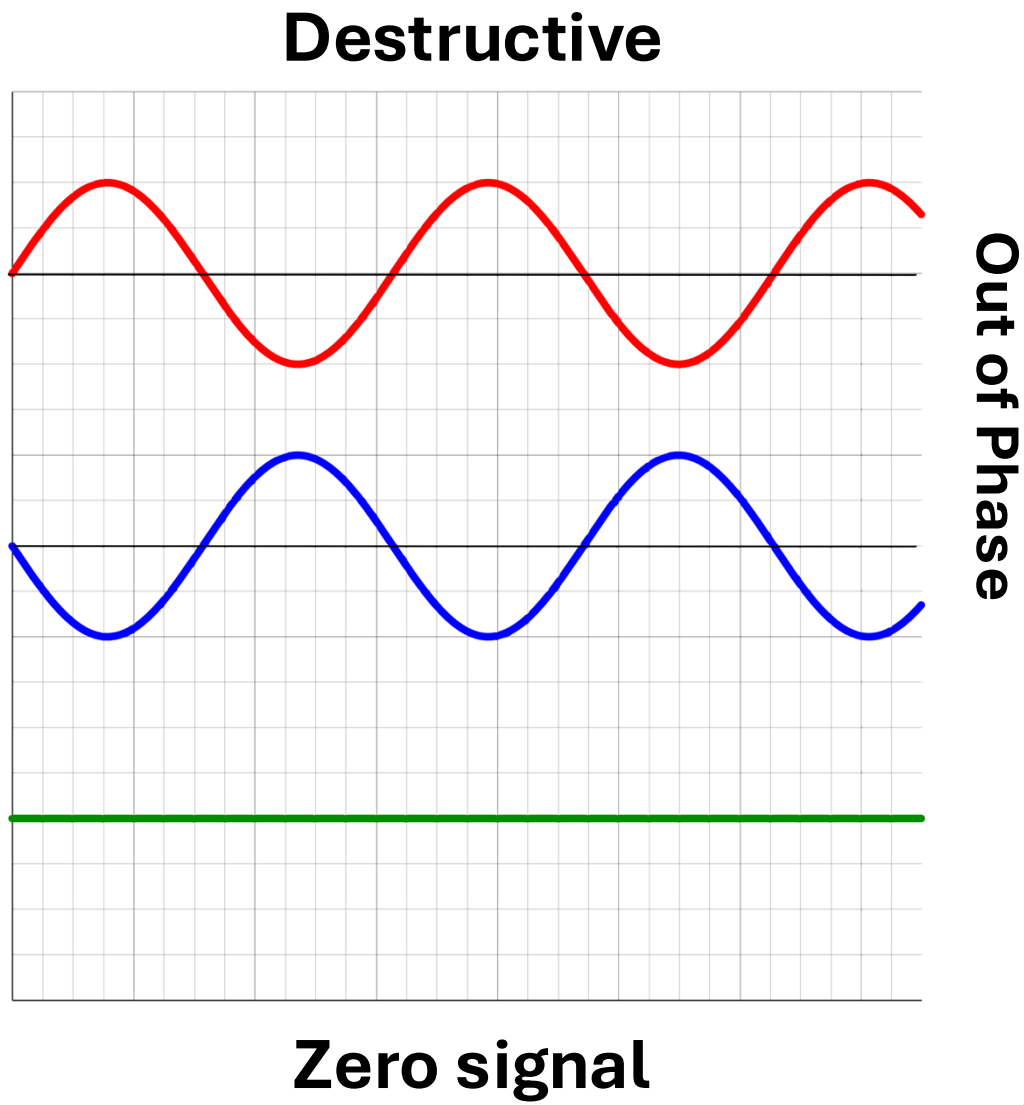
X-rays diffract from crystallographic planes



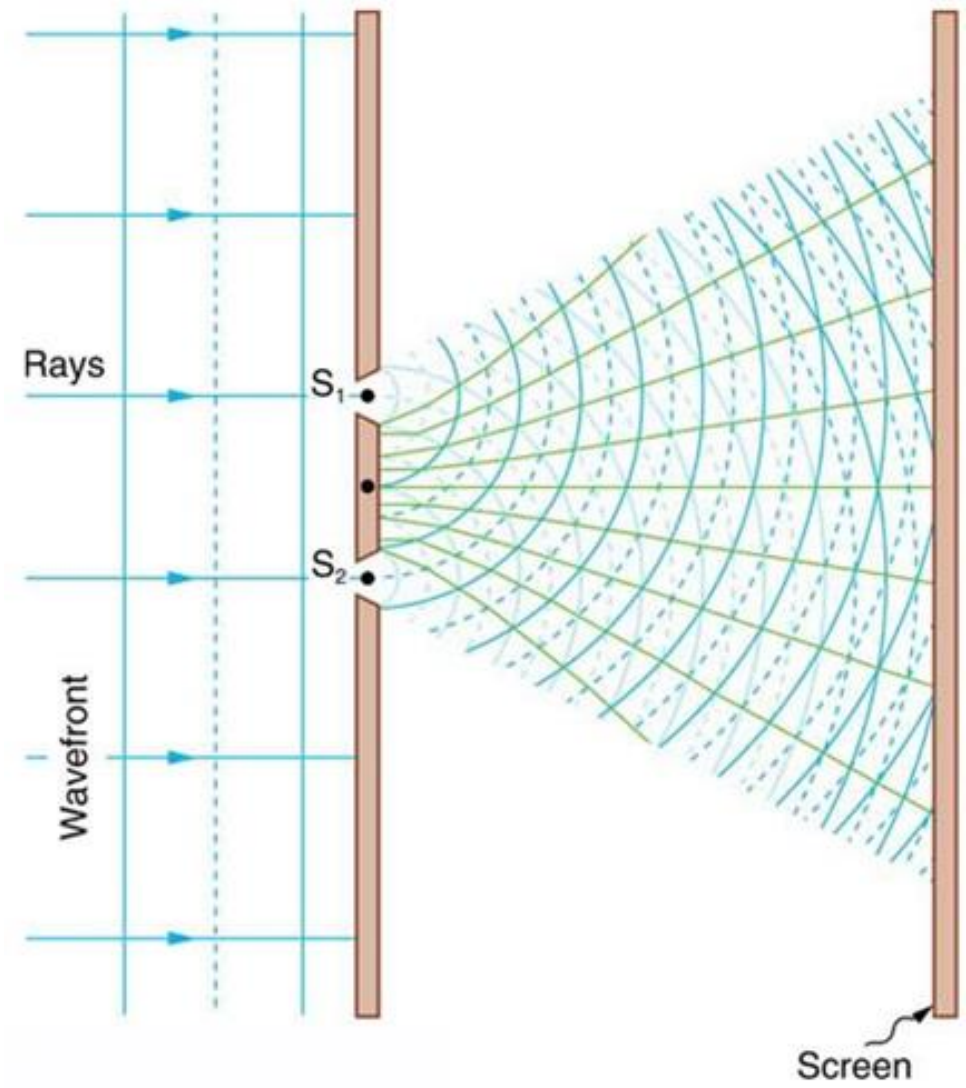
How do these X-rays interact?

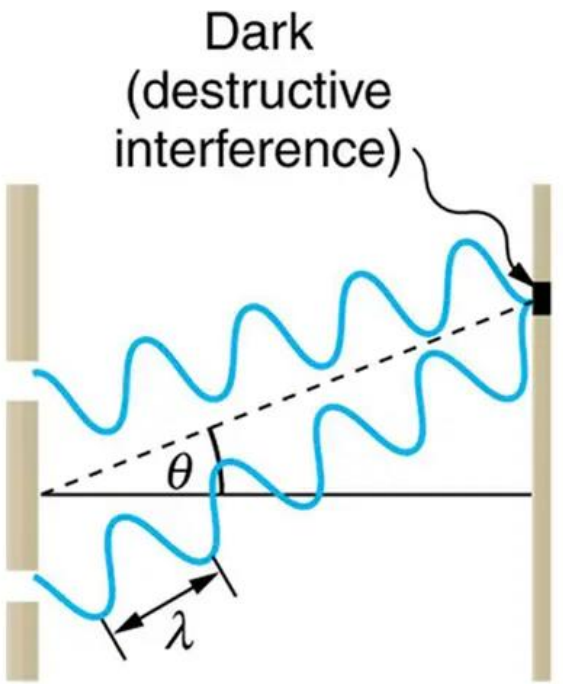
Constructive *vs.* Destructive interference

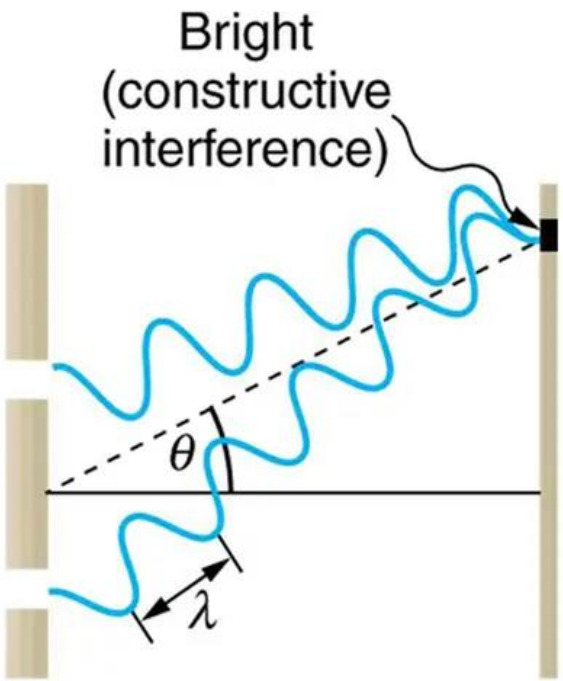




Interference patterns: Double-slit diffraction

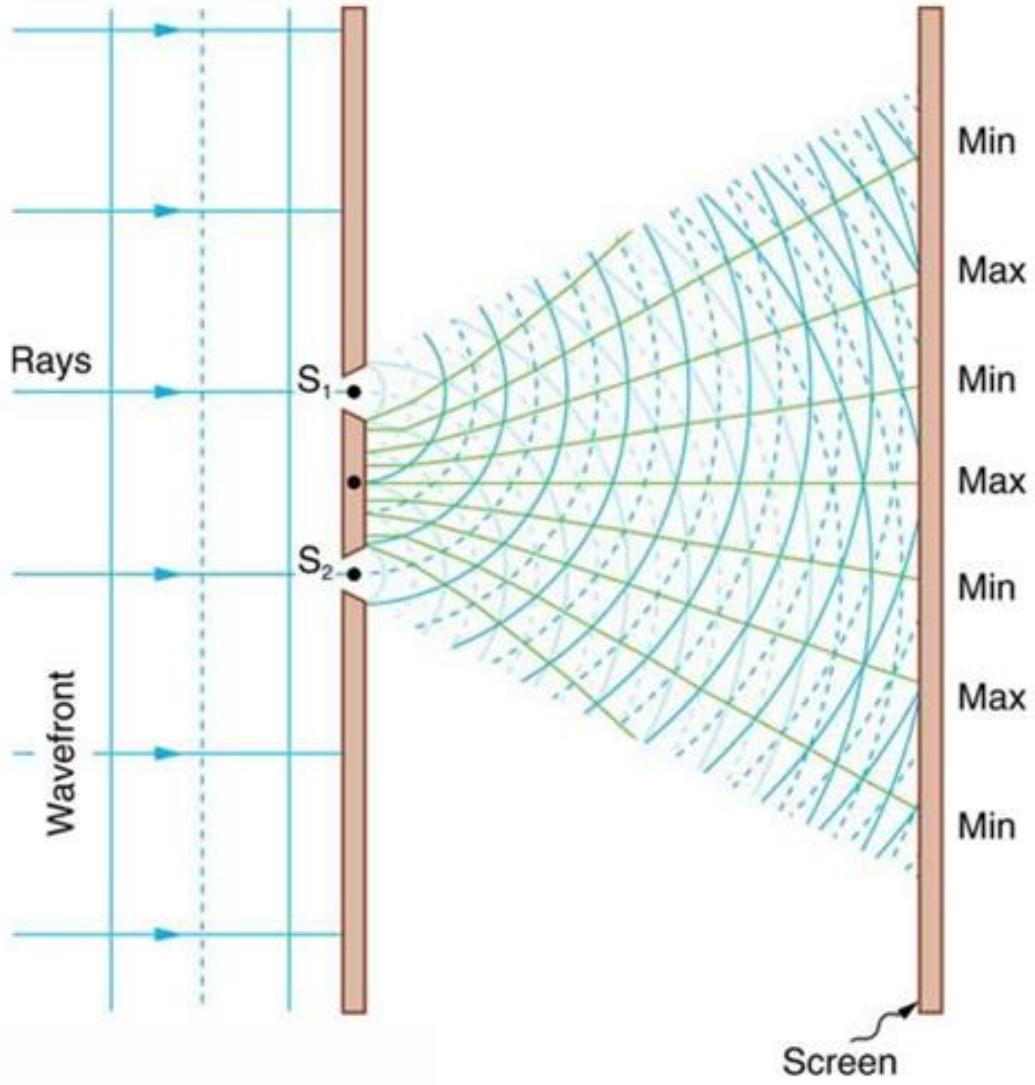


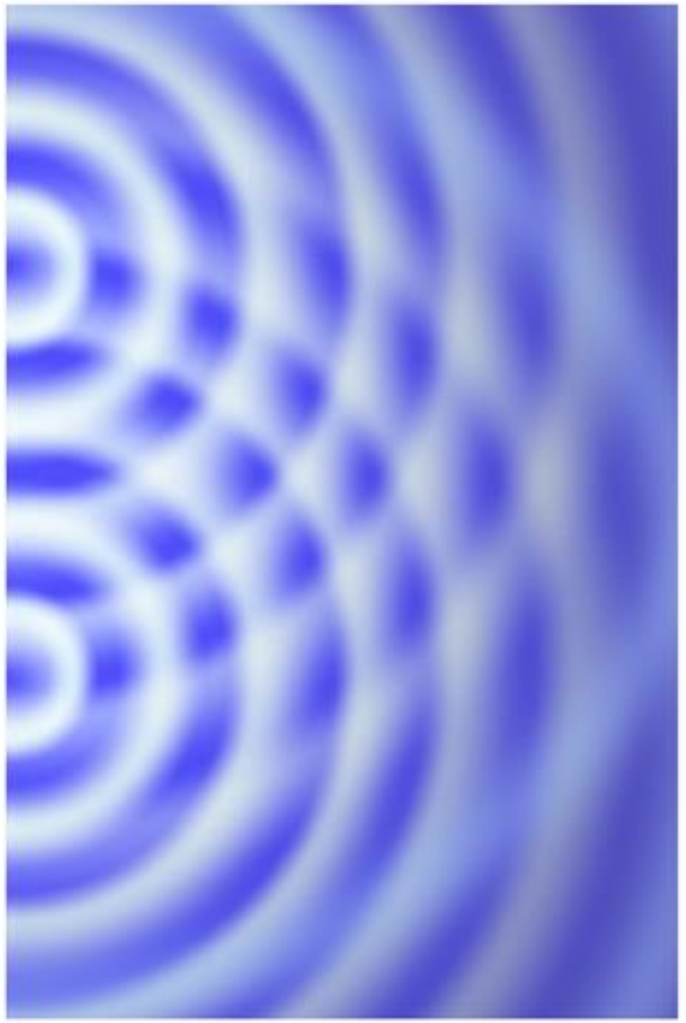




It is the **path length difference** between wave from each slit that leads to interference

Interference patterns: Double-slit diffraction

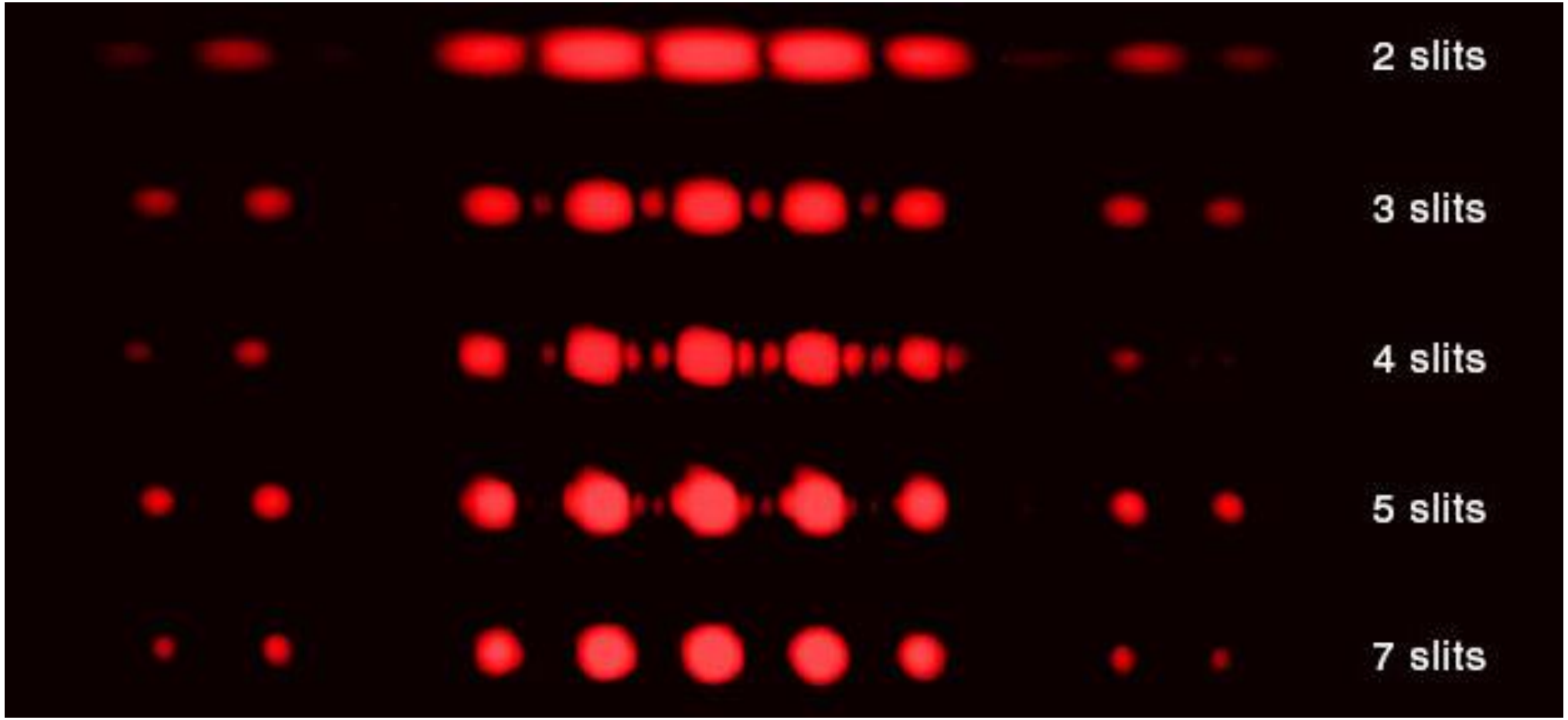




Interference pattern

Interference patterns: Double-slit diffraction

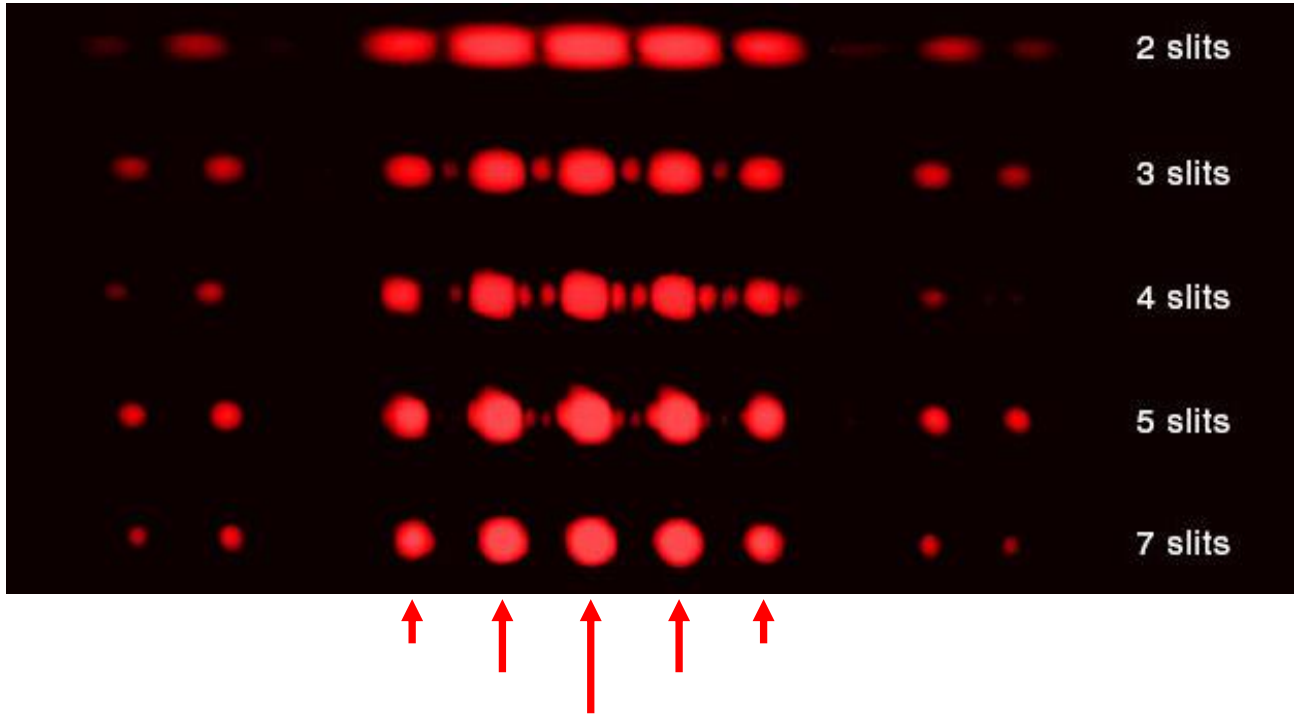
More slits → fewer visible spots → brighter and sharper



For comparison, crystals have ~10^{23} planes to diffract from!

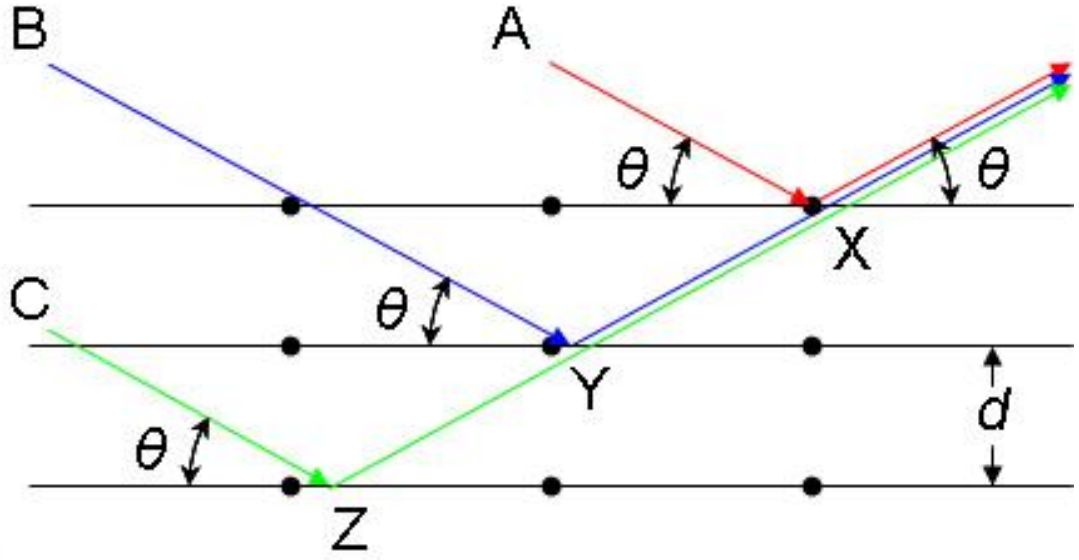
Interference patterns: Double-slit diffraction

More slits → fewer visible spots.
These spots are brighter and sharper.



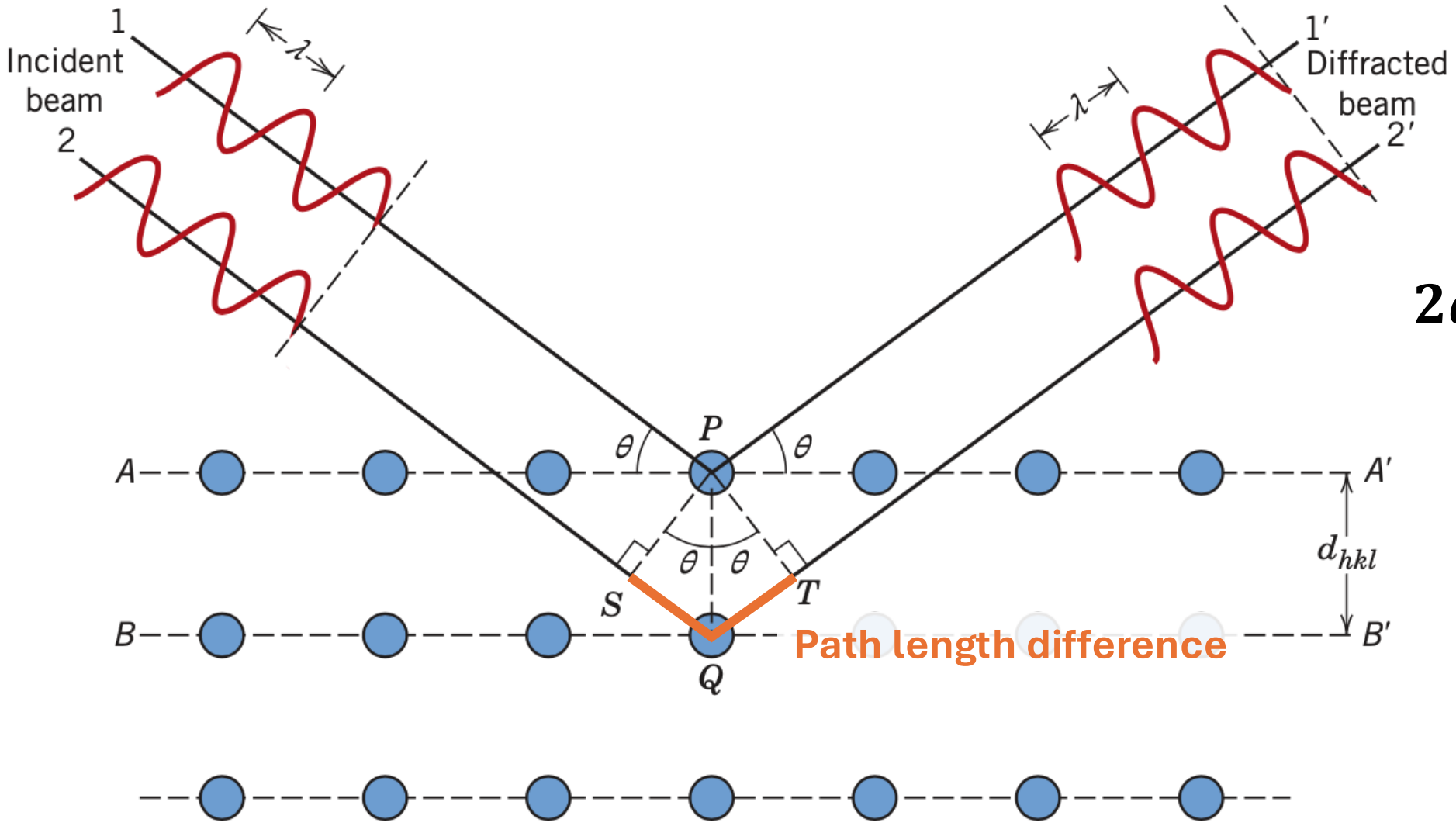
The location of these spots tell us about the slit spacing

For comparison, crystals have ~10^{23} planes to diffract from!



In this case, it is the *plane* spacing we learn about from diffraction patterns

Plane spacing from Bragg's law

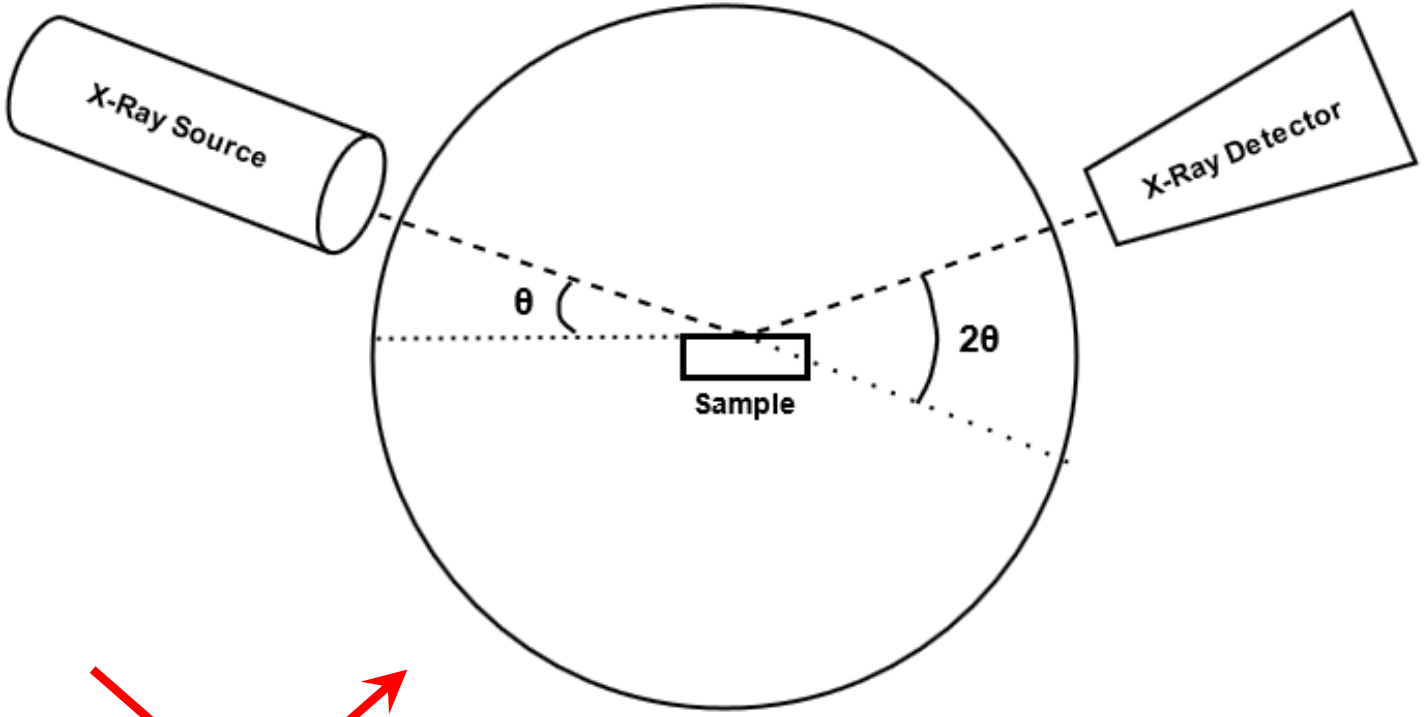


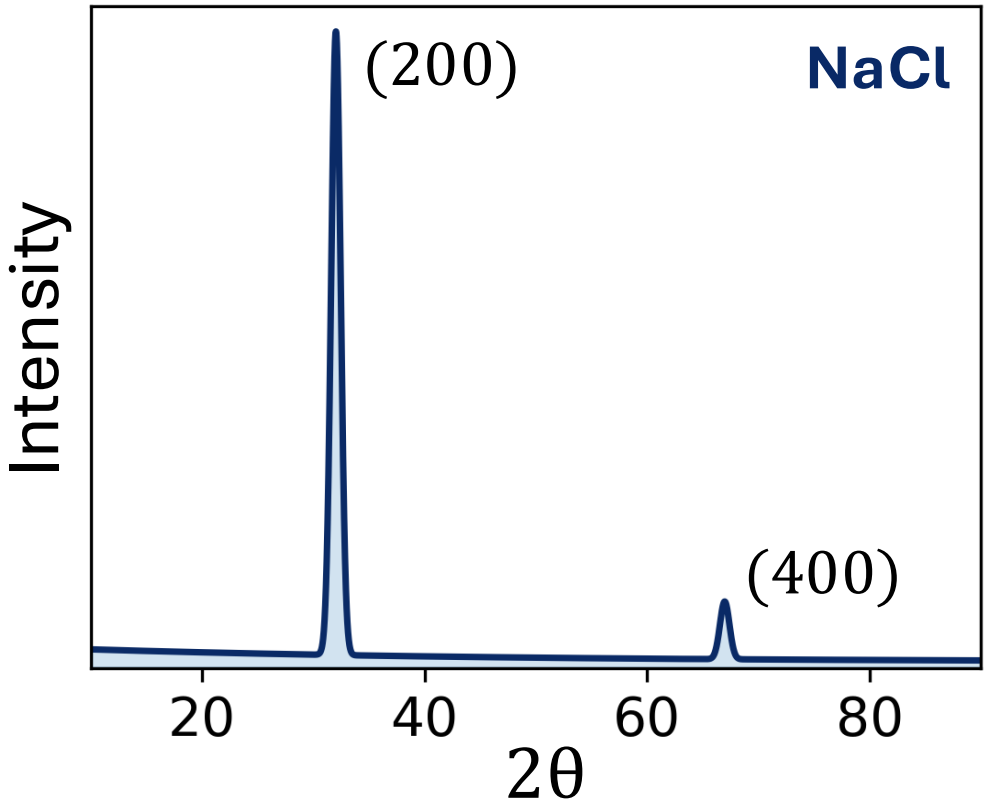
$$2d_{hkl}\sin\theta = n\lambda$$

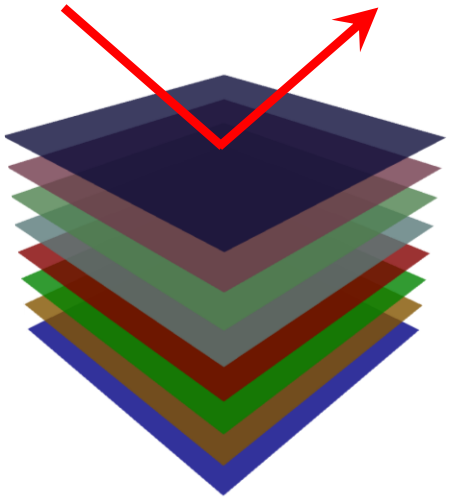
The path length difference is a multiple of the wavelength

→ **Constructive interference**

Bragg-Brentano setup

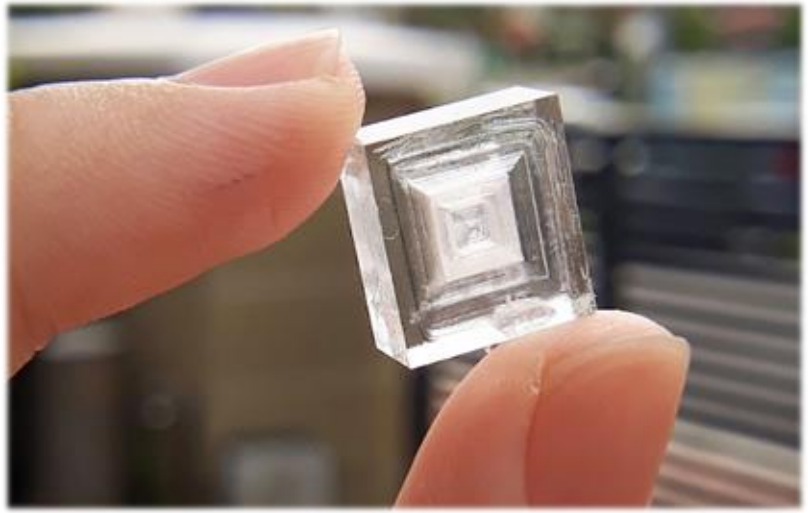




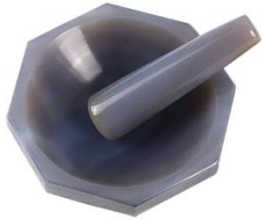


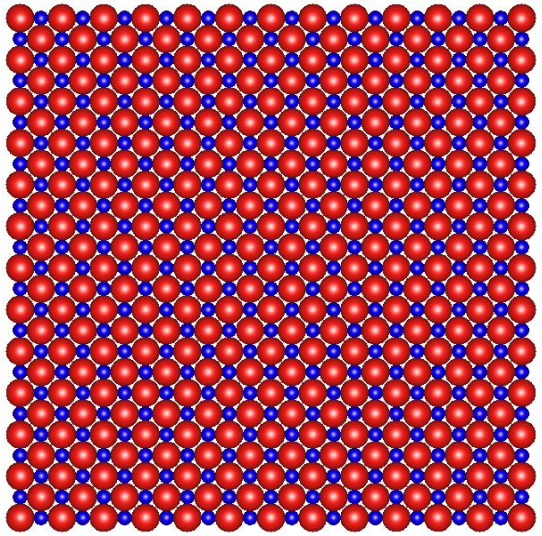
If the sample is a single crystal, **very few peaks** will be observed.
These correspond *only* to planes that are **parallel to the surface**.

Powder X-ray diffraction (XRD)

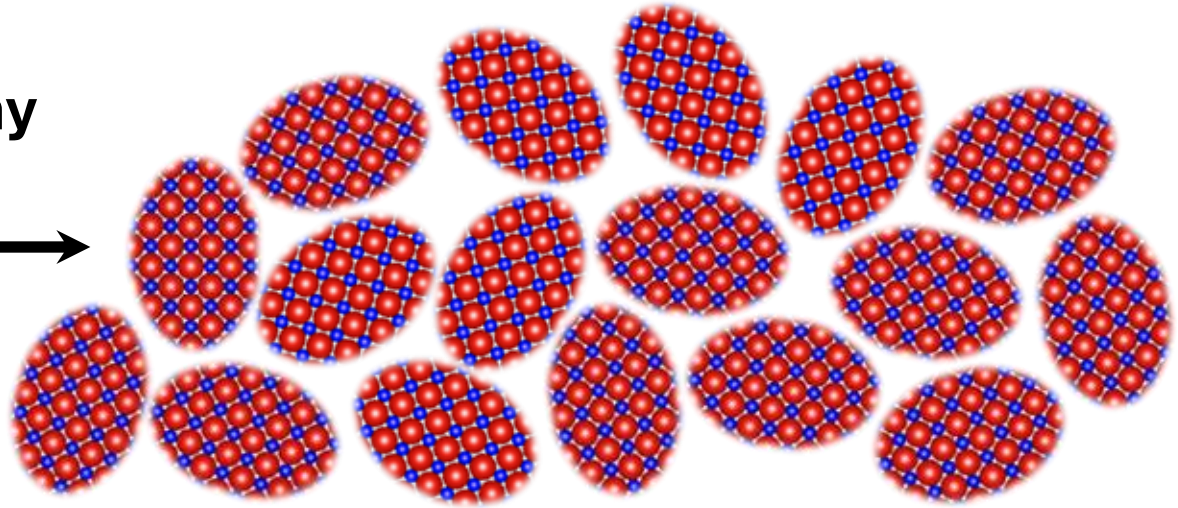


Grind into a fine powder

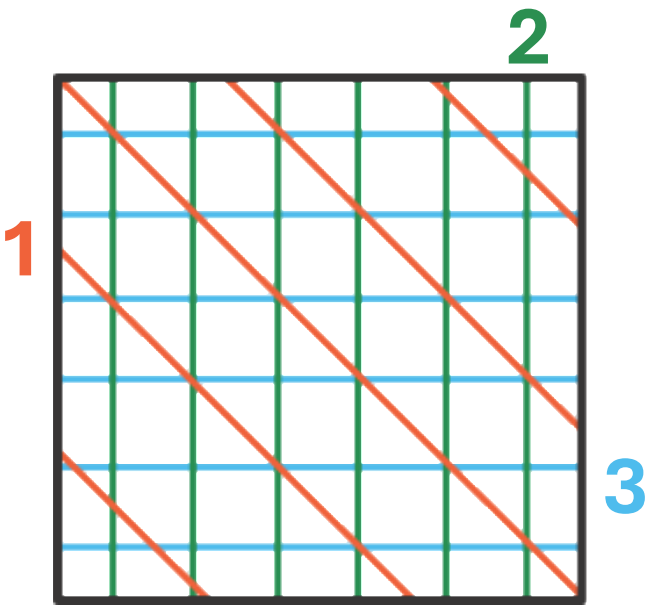




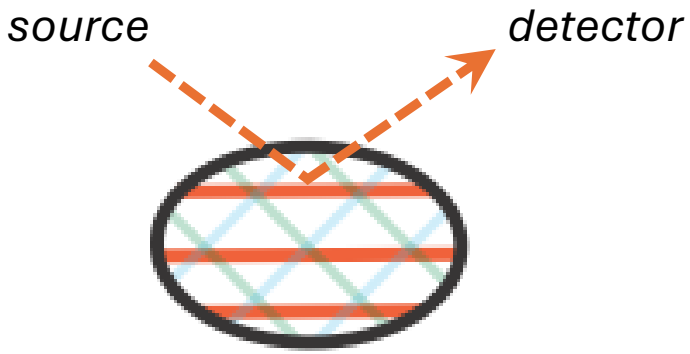
From one → many orientations



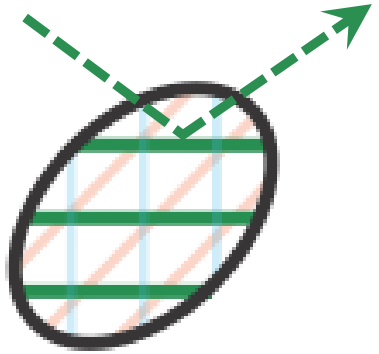
Powder X-ray diffraction (XRD)



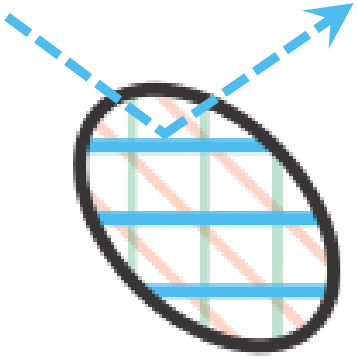
Suppose the single crystal has **three planes**



In the powder, some grains will be oriented for diffraction from **plane 1**



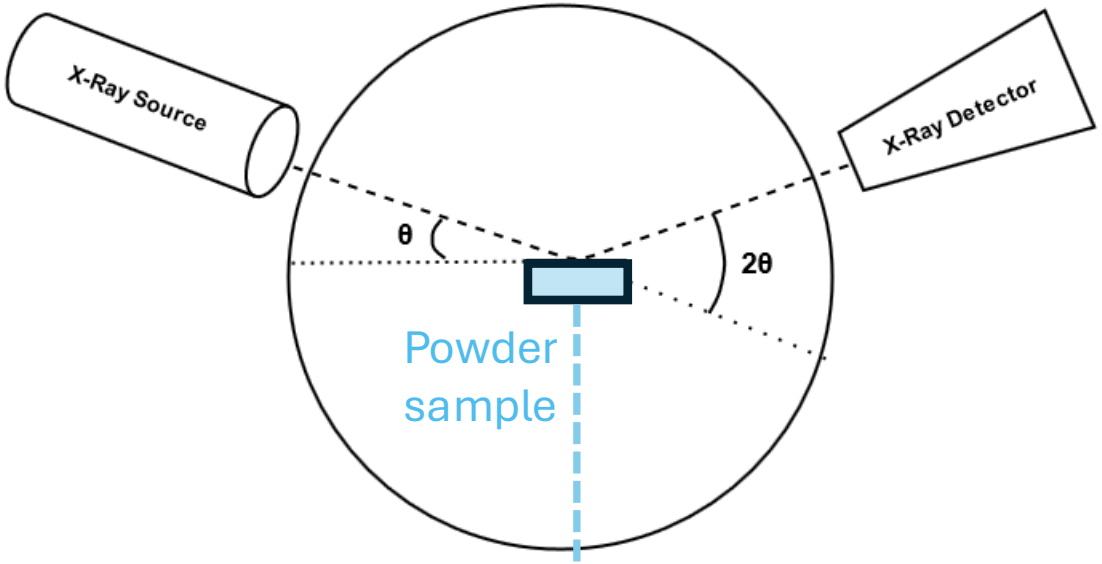
And some will be oriented for diffraction from **plane 2**

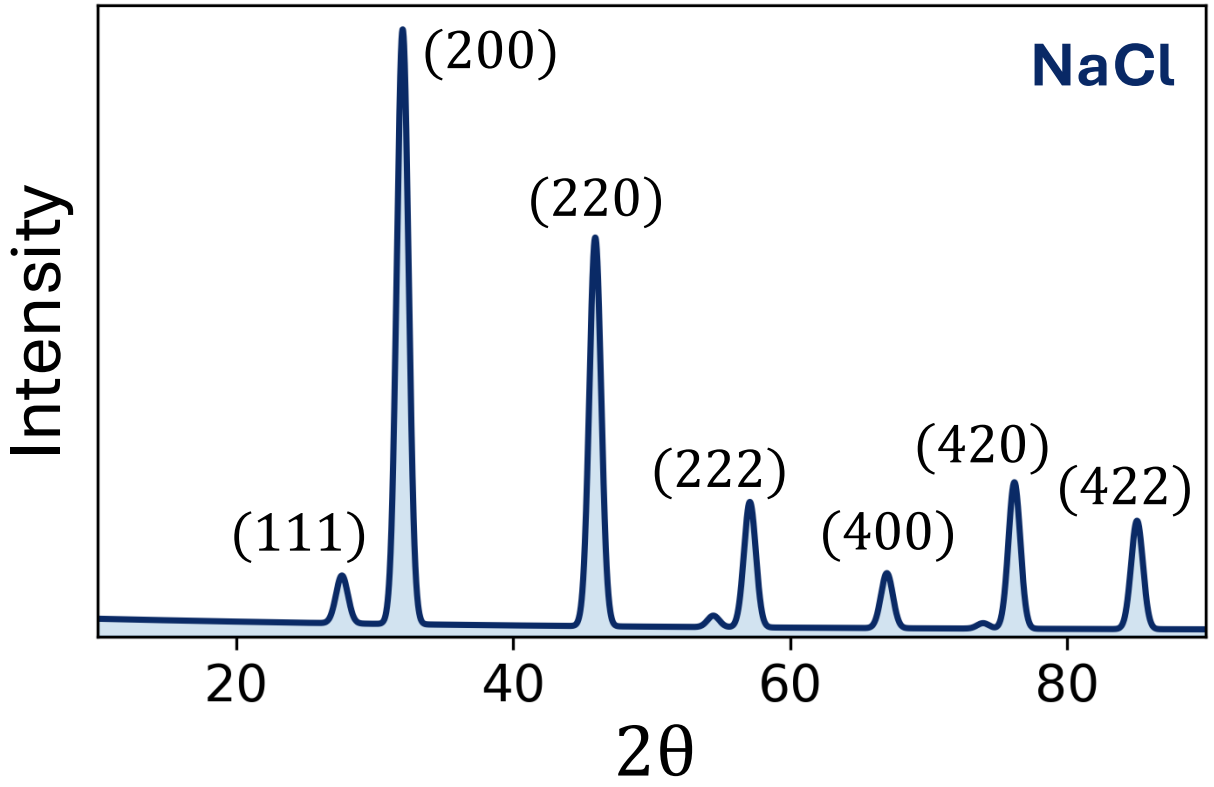


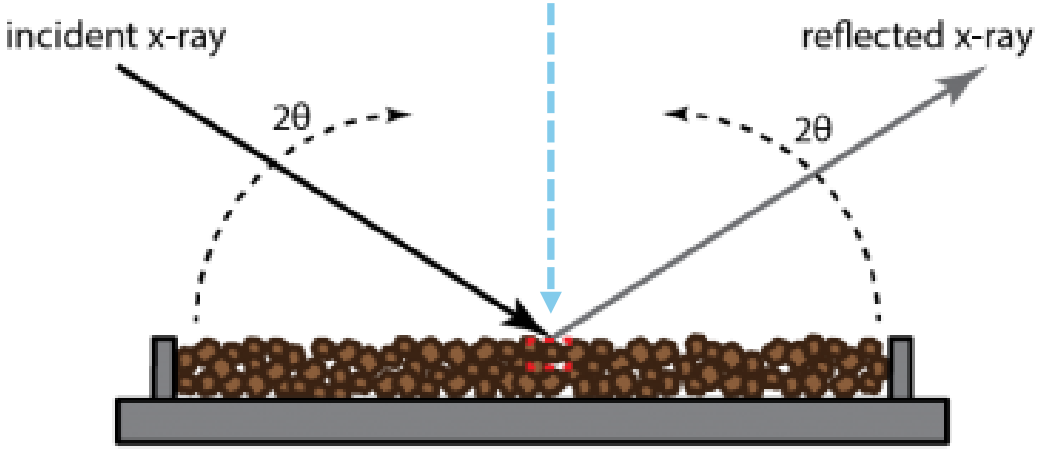
And some will be oriented for diffraction from **plane 3**

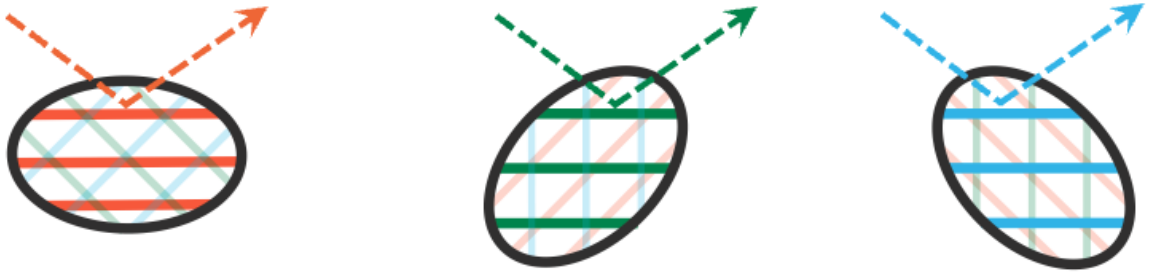
But in each case, a bright signal will be detected *only* when Bragg's law is satisfied for that family of planes: $\boldsymbol{2d_{hkl}\sin\theta = n\lambda}$

Powder X-ray diffraction (XRD)









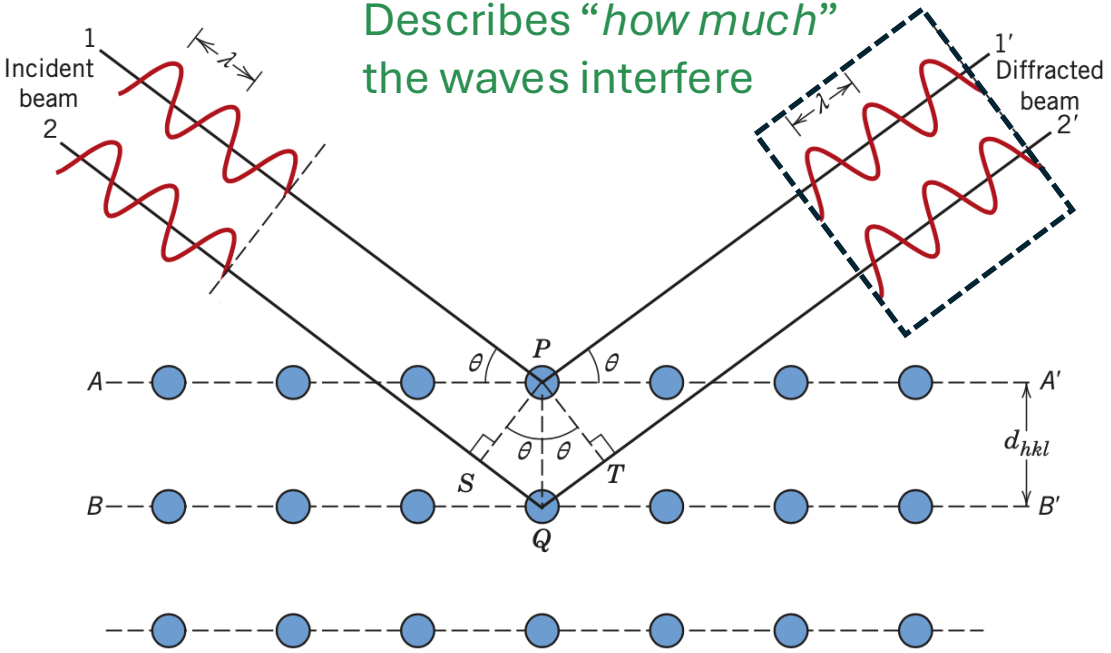
The structure factor determines peak intensity

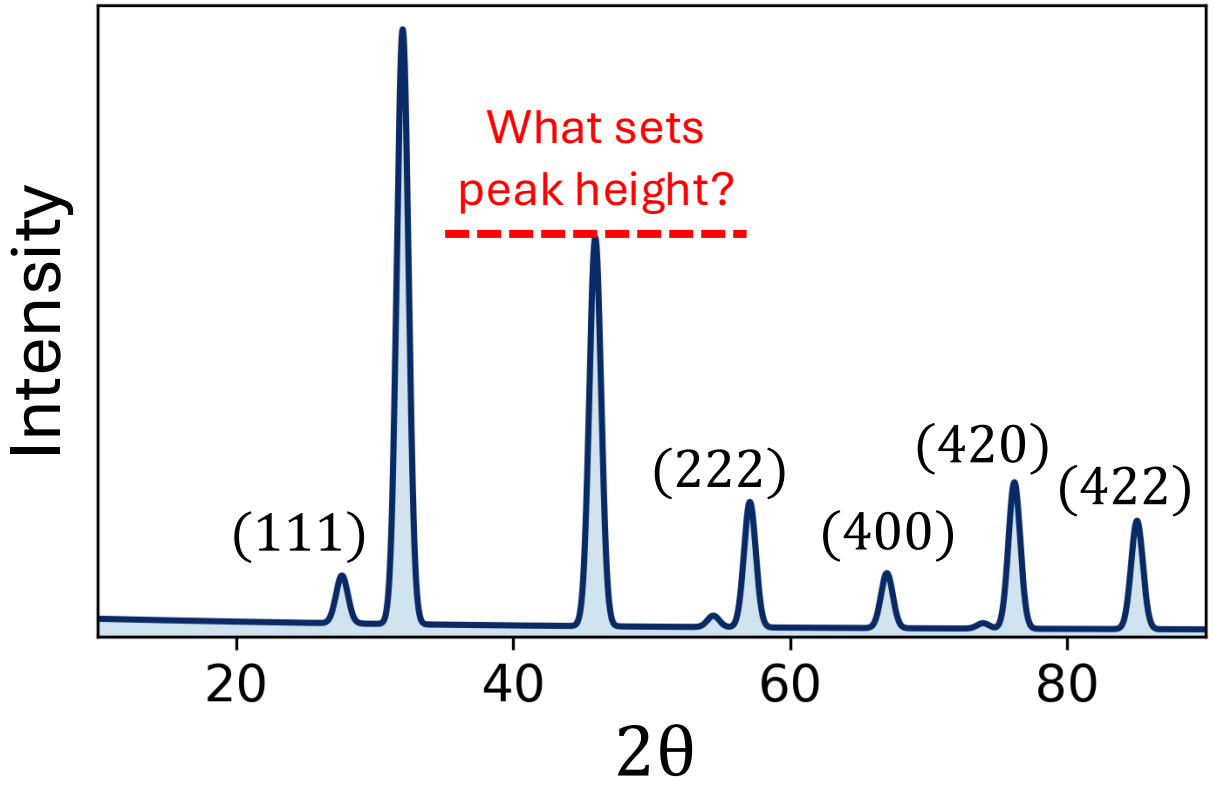
$$I_{hkl} \propto |F_{hkl}|^2$$

Structure Factor

↓

Describes "*how much*" the waves interfere

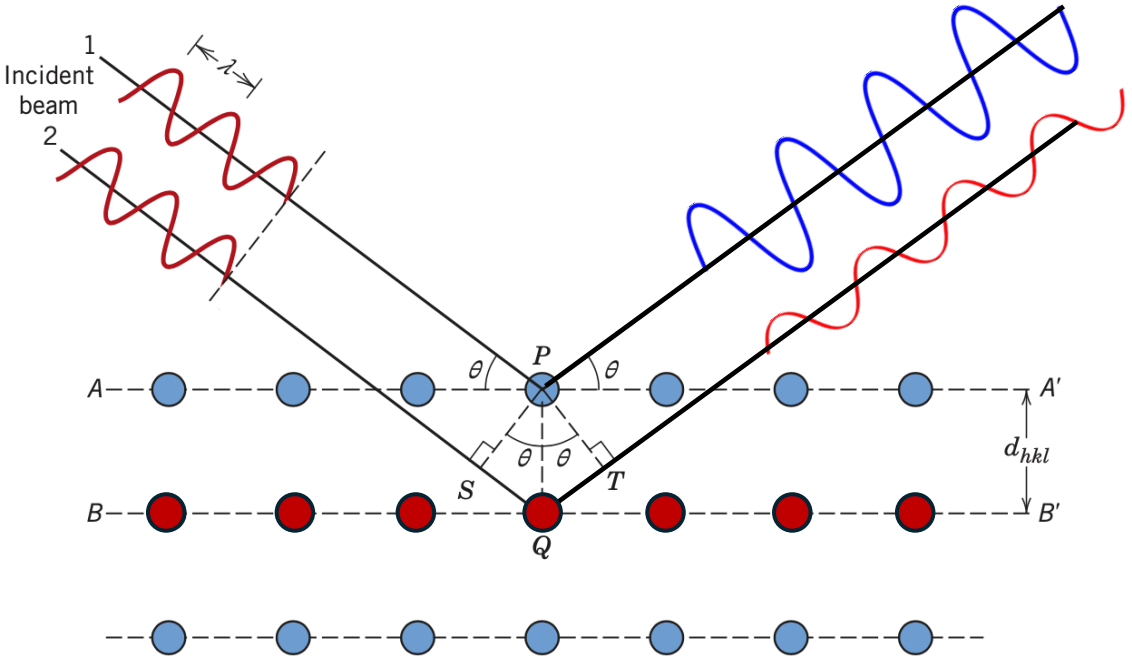


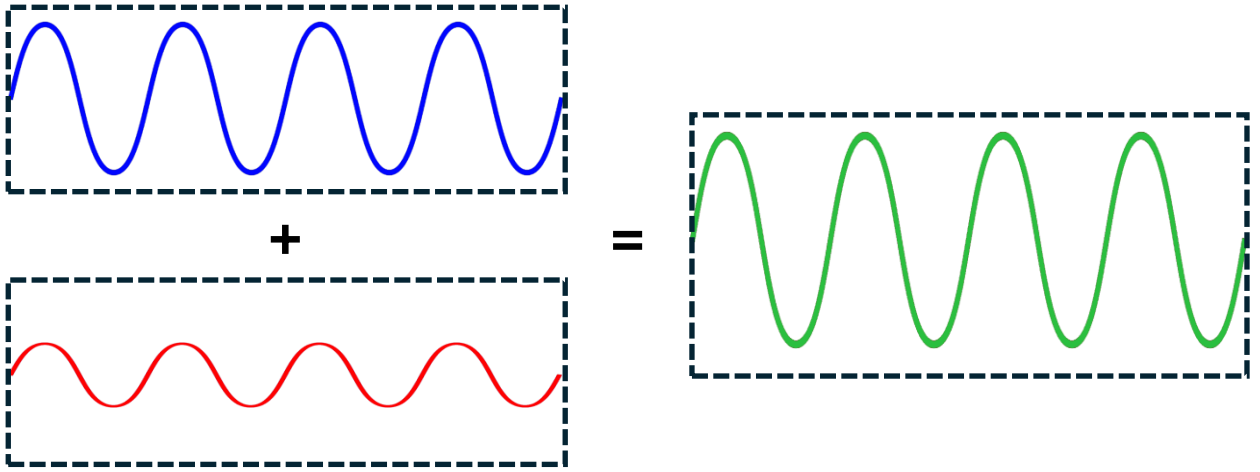


$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

Recommended Video: Pat's Perovskites

The structure factor determines peak intensity



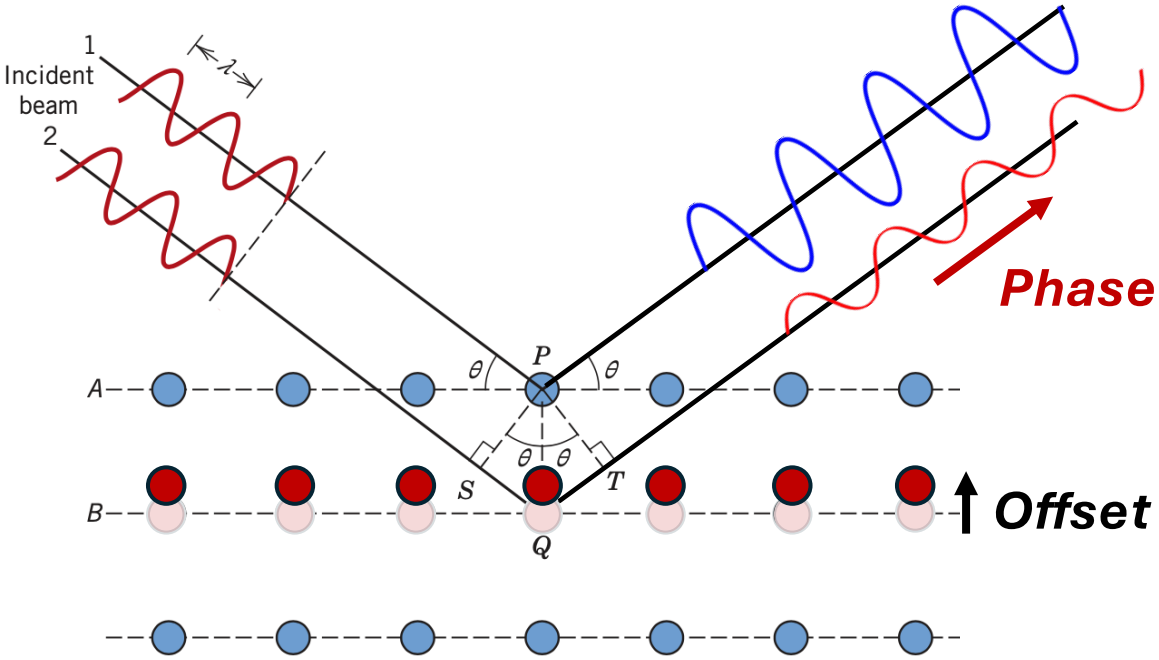


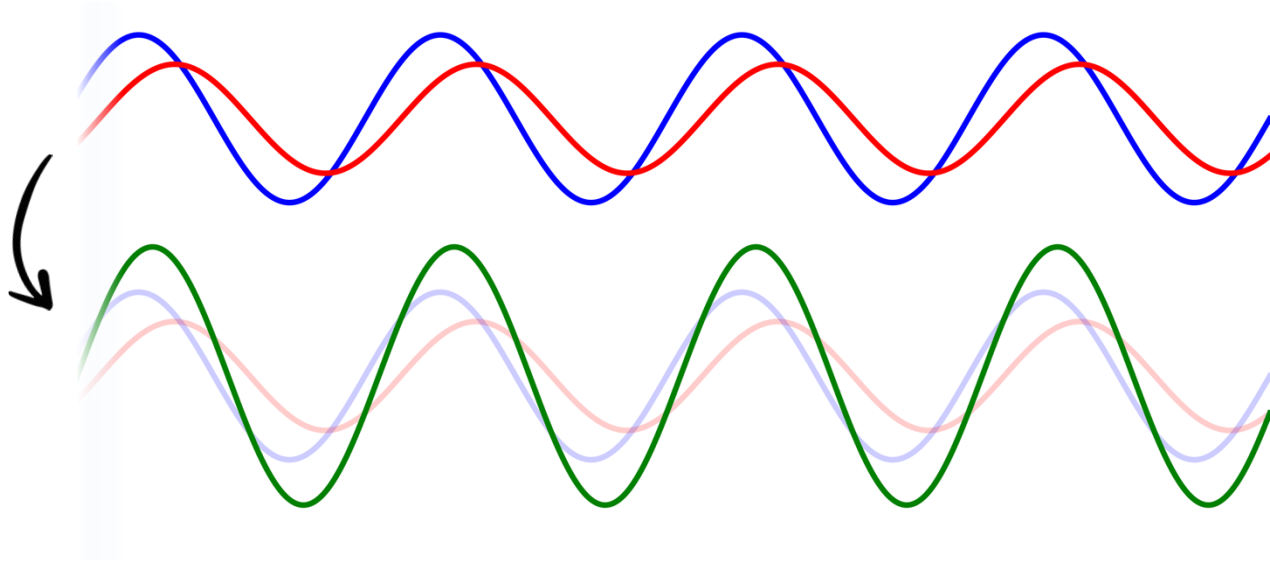
This **atomic form factor** describes how strongly atom j scatters X-rays

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

Recommended Video: Pat's Perovskites

The structure factor determines peak intensity





These **atomic coordinates** describe ***how in-phase*** the scattered wave are

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

Recommended Video: Pat's Perovskites

Simulating “stick” patterns with PyMatGen

Bragg’s Law → peak positions

$$2d_{hkl}\sin\theta = n\lambda$$

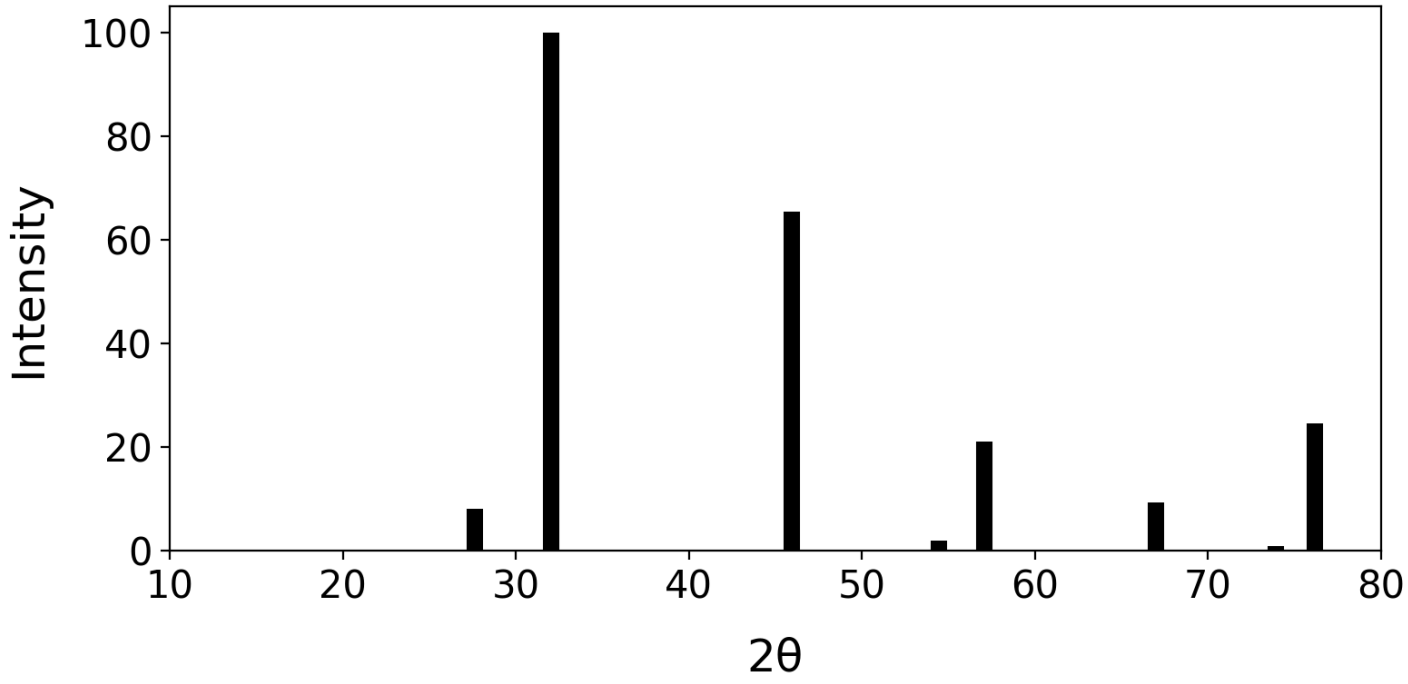
Structure Factor → peak intensities

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

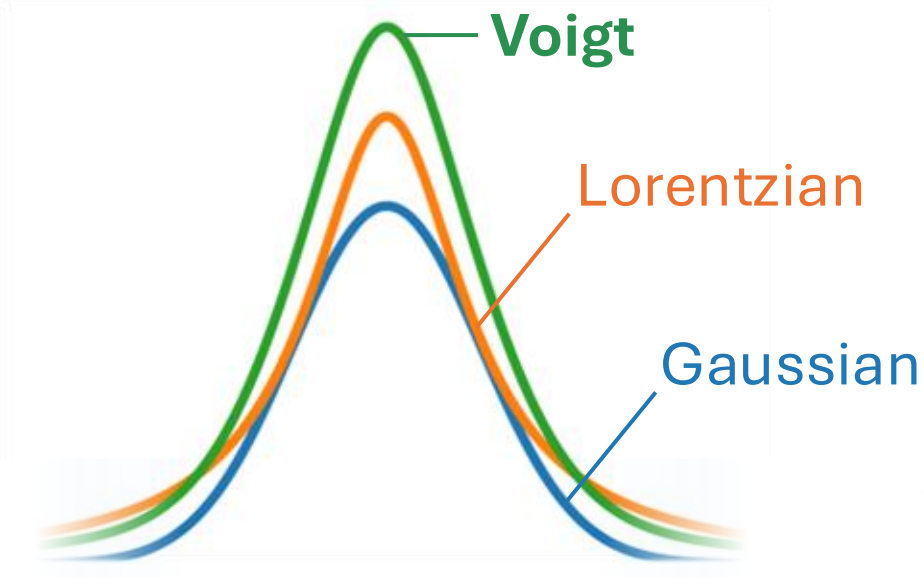
→ **$(2\theta, I)$ lists**

jupyter

pymatgen



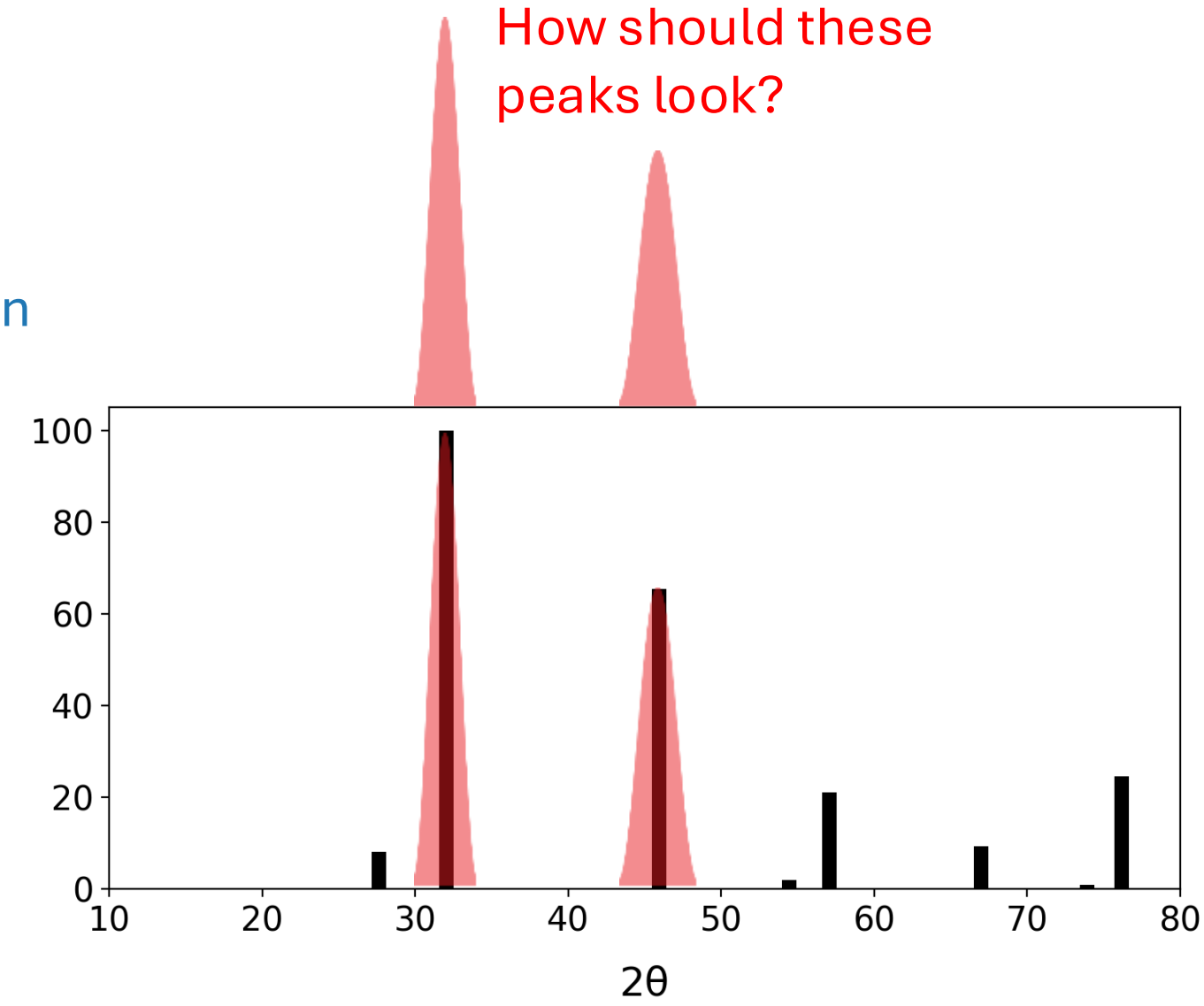
From “stick” to continuous patterns



$$V(x,\sigma,\gamma) = \int_{-\infty}^{\infty} G(x',\sigma)L(x-x',\gamma)dx'$$

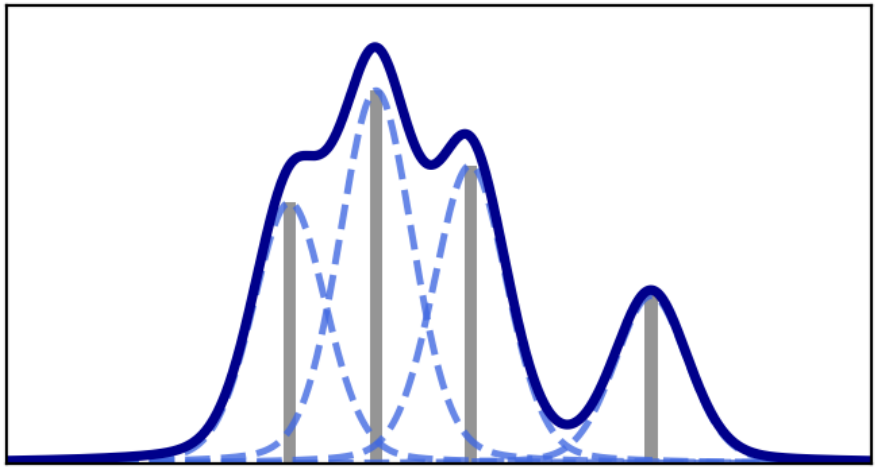
The **Voigt profile** is a *convolution* of a Gaussian and Lorentzian

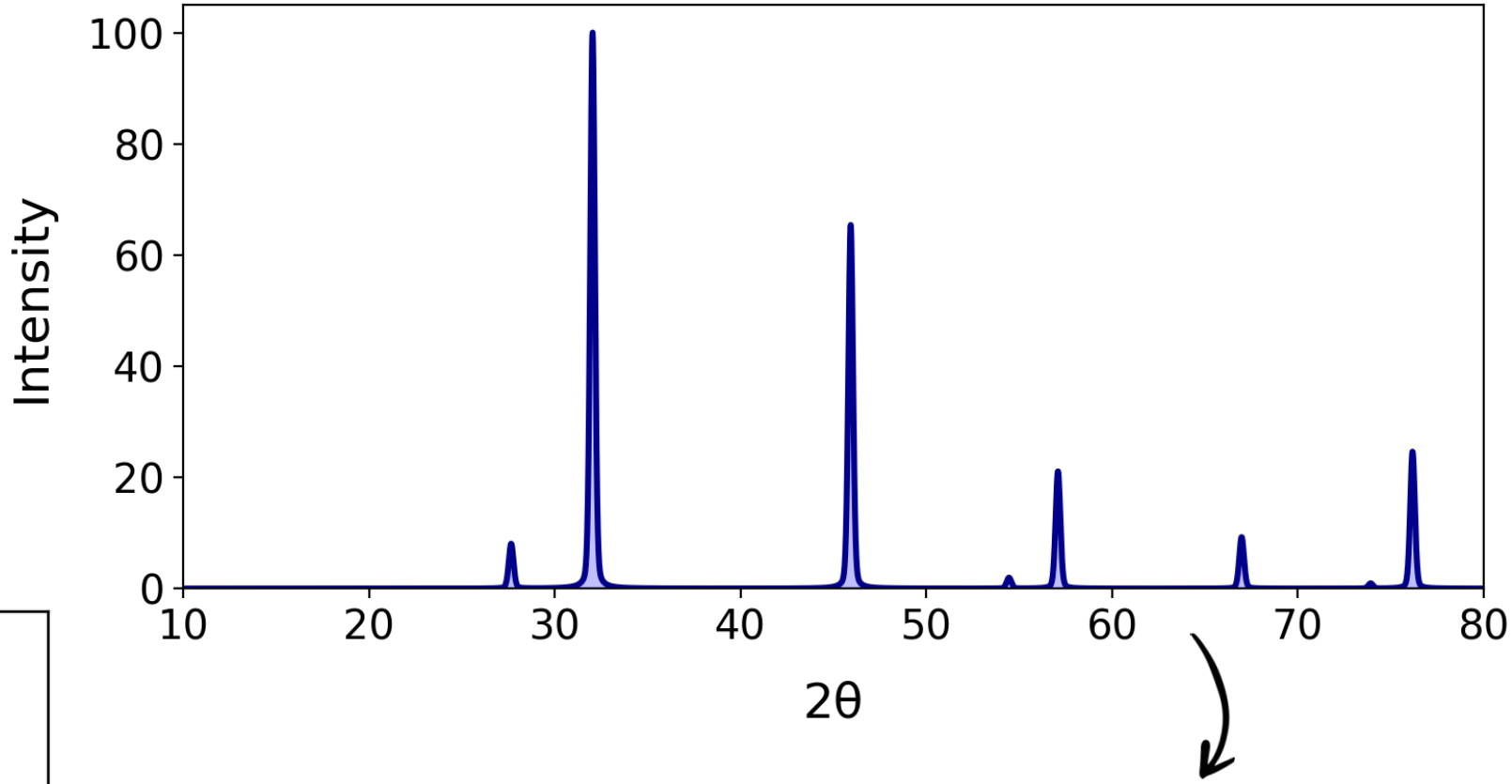
Or we can simply **sum them** → ***pseudo*-Voigt**



From “stick” to continuous patterns

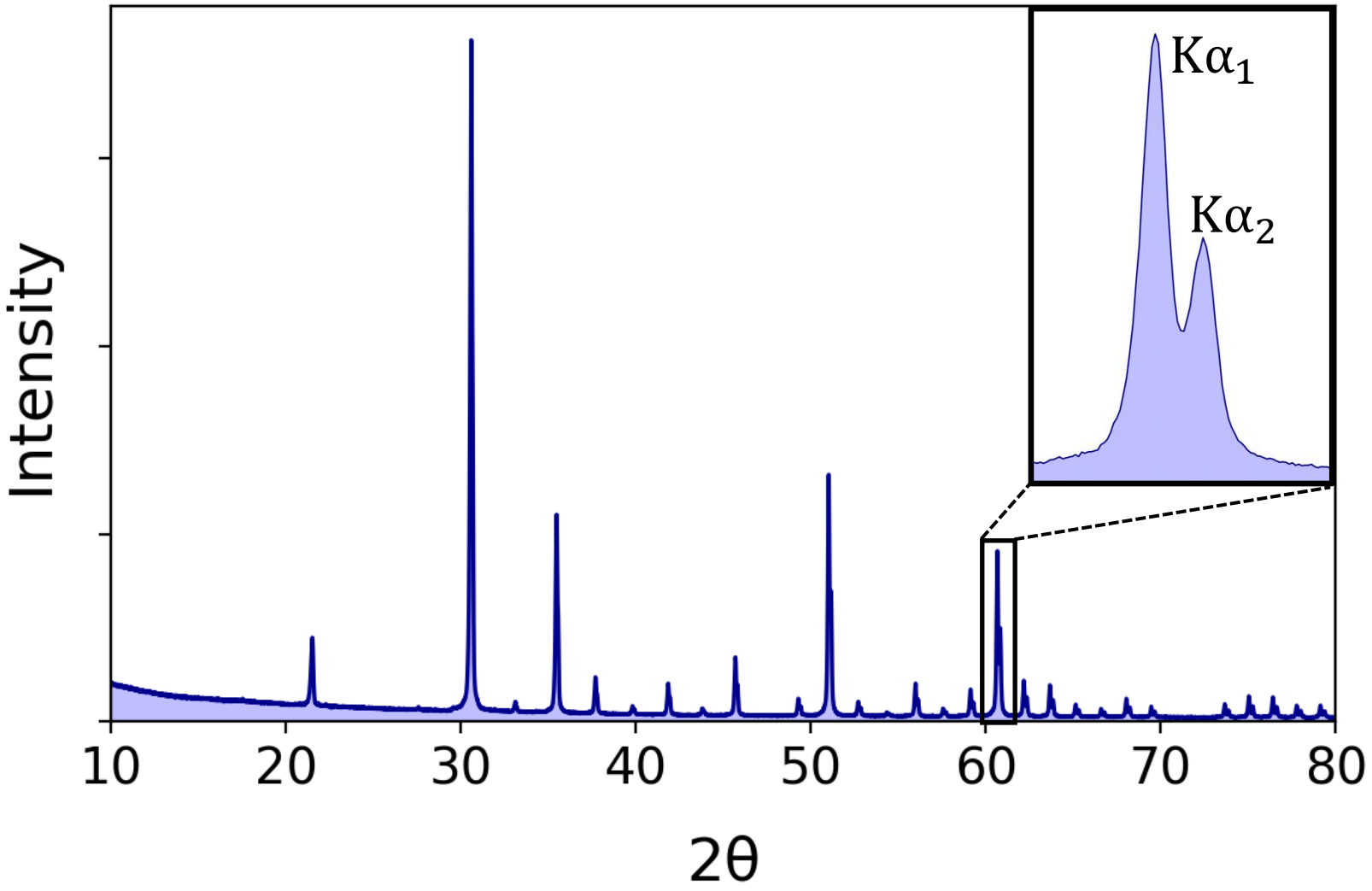
1) Compute stick patterns
2) **Define peak shape**
3) **Sum the profiles from all peaks on a densely sampled 2θ grid**
4) Normalize and plot





Does this pattern look ***realistic?***
Similar to experimental data?

Real XRD patterns are more complicated



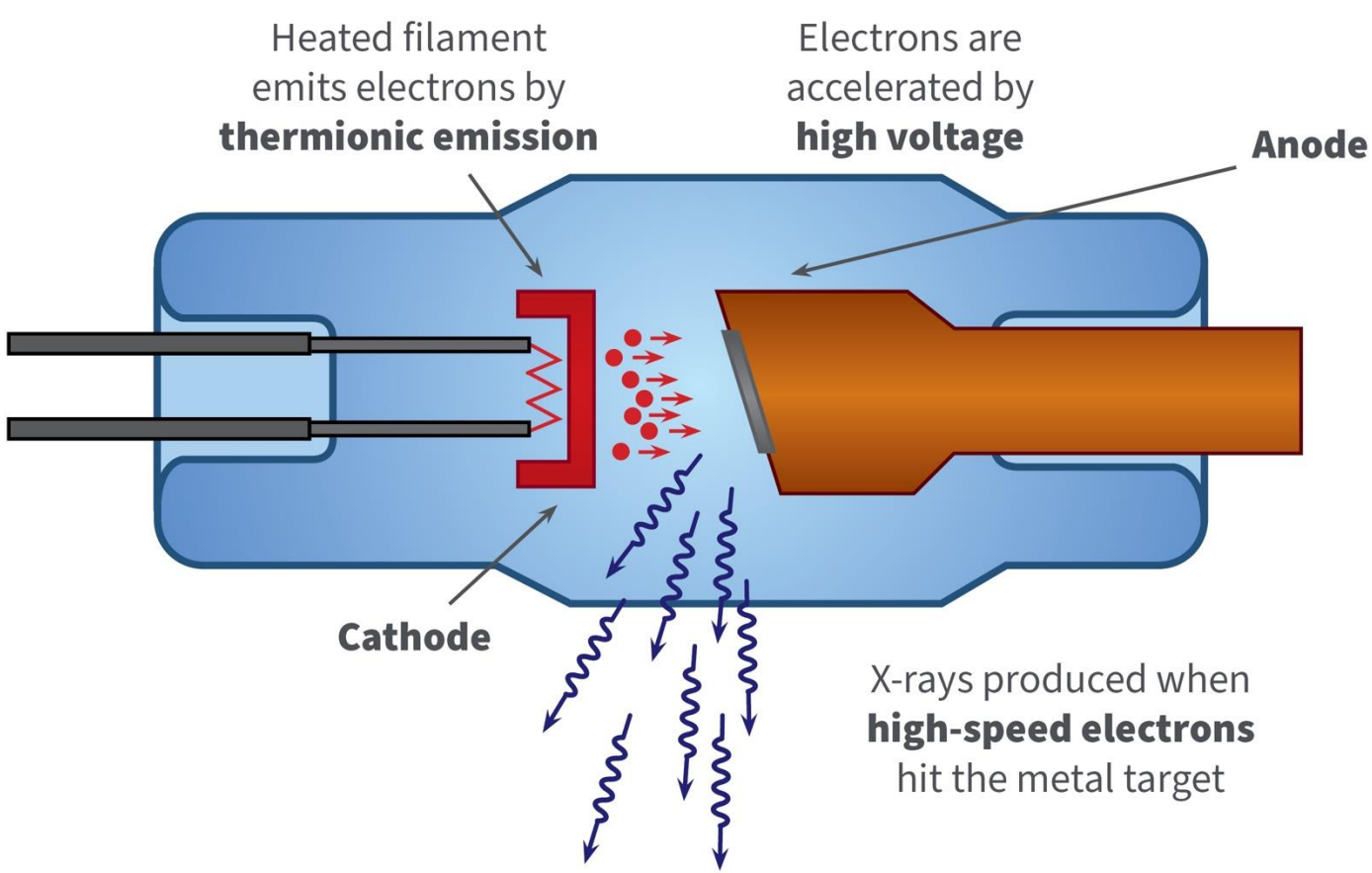
Notice that single peaks begin to **split**, especially at high 2θ

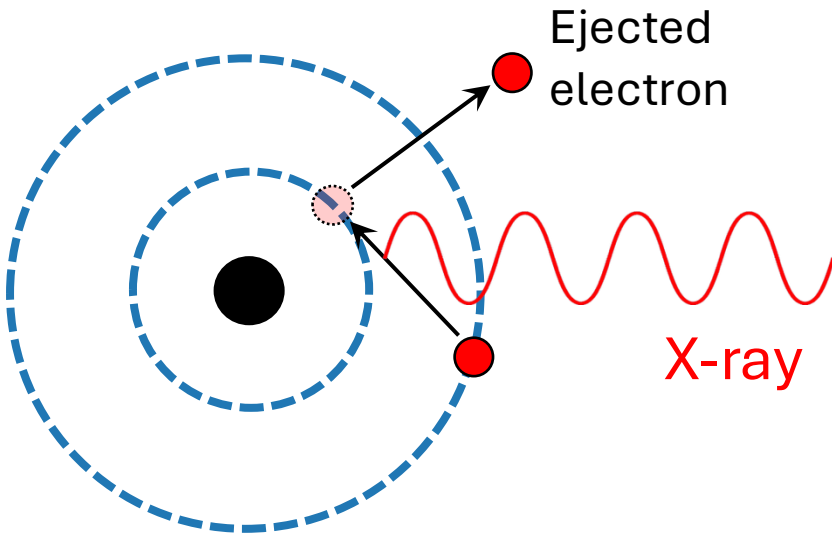
This occurs when the X-rays are not perfectly monochromatic – *i.e.*, with **> 1 wavelength**

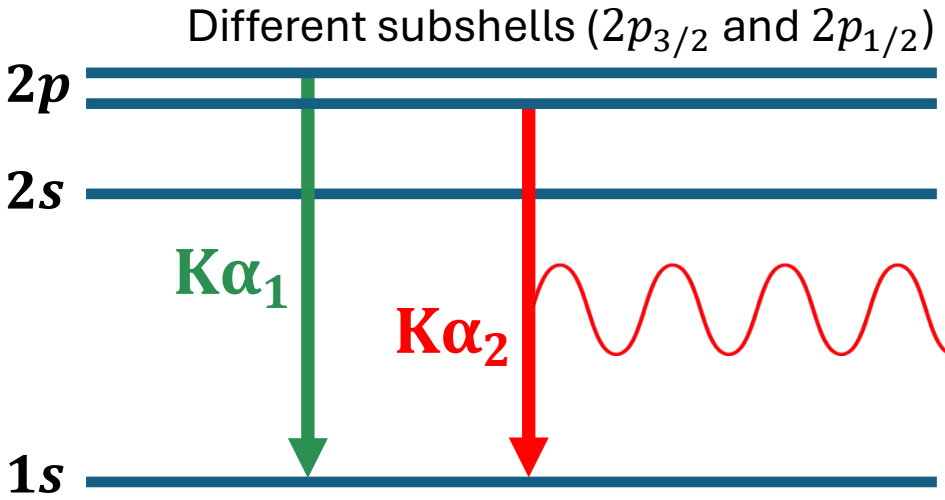
$$2d_{hkl}\sin\theta = n\lambda$$

θ_1 θ_2 λ_1 λ_2

How to model peak splitting (Cu $K\alpha_1$ *vs.* $K\alpha_2$)

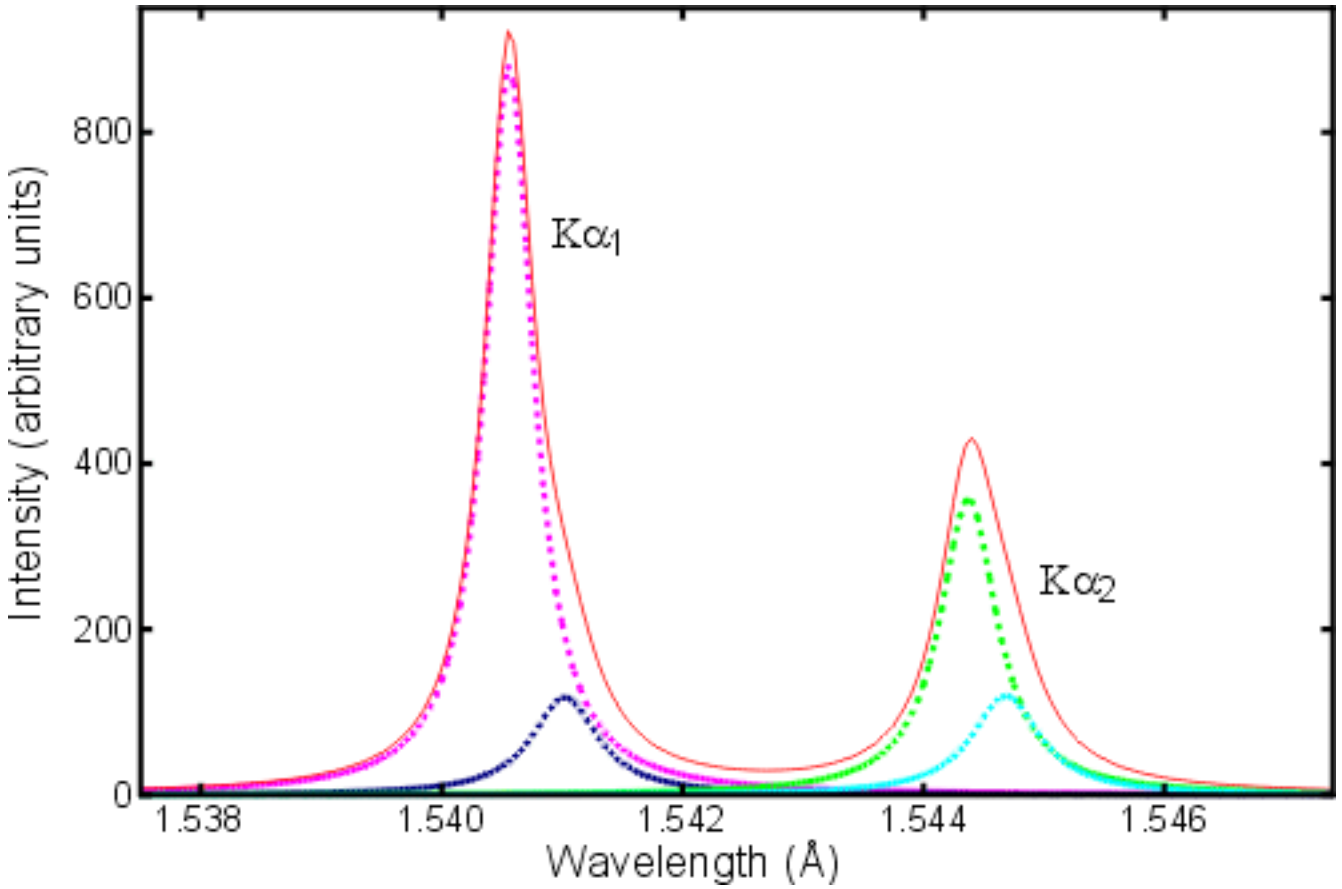






Accelerated electrons 1) smash into Cu, 2) ejecting core electrons, 3) **outer-shell electrons relax to fill vacancy, emitting X-rays**

How to model peak splitting (Cu $K\alpha_1$ *vs.* $K\alpha_2$)



We need to account for:

1) **Different wavelengths**

$$K\alpha_1 \rightarrow 1.540598\ \text{Å}$$
$$K\alpha_2 \rightarrow 1.544426\ \text{Å}$$

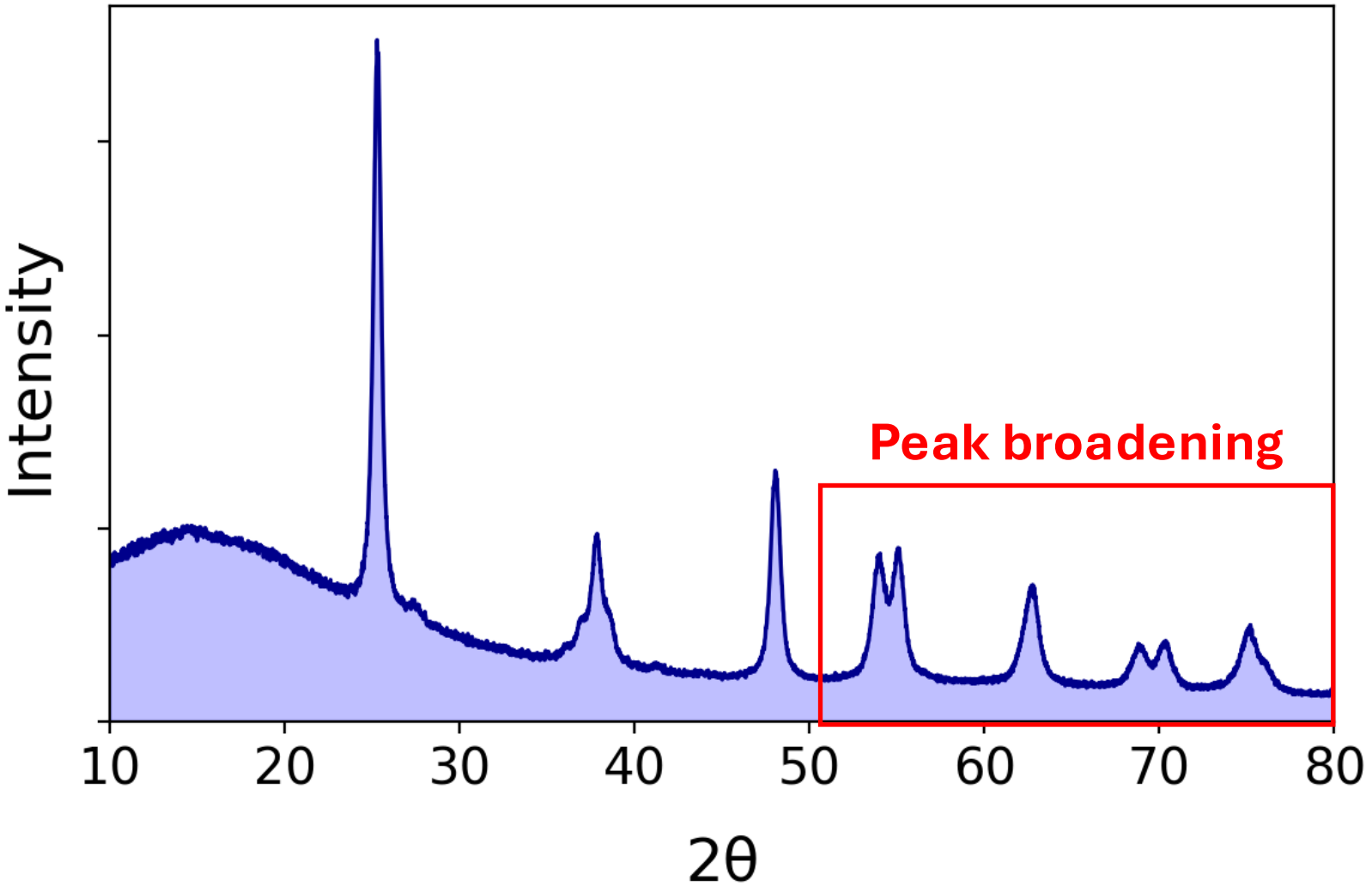
2) **Different intensities**

$$K\alpha_1 \sim 2\,K\alpha_2$$

Then we simply **add the patterns**

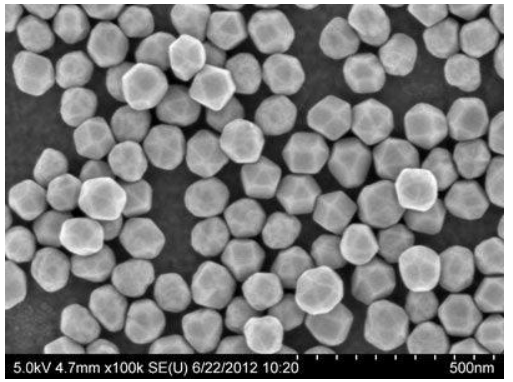
H. Berger in X-ray Spectrometry, 1986, **15**, 241-243.

Real XRD patterns are more complicated



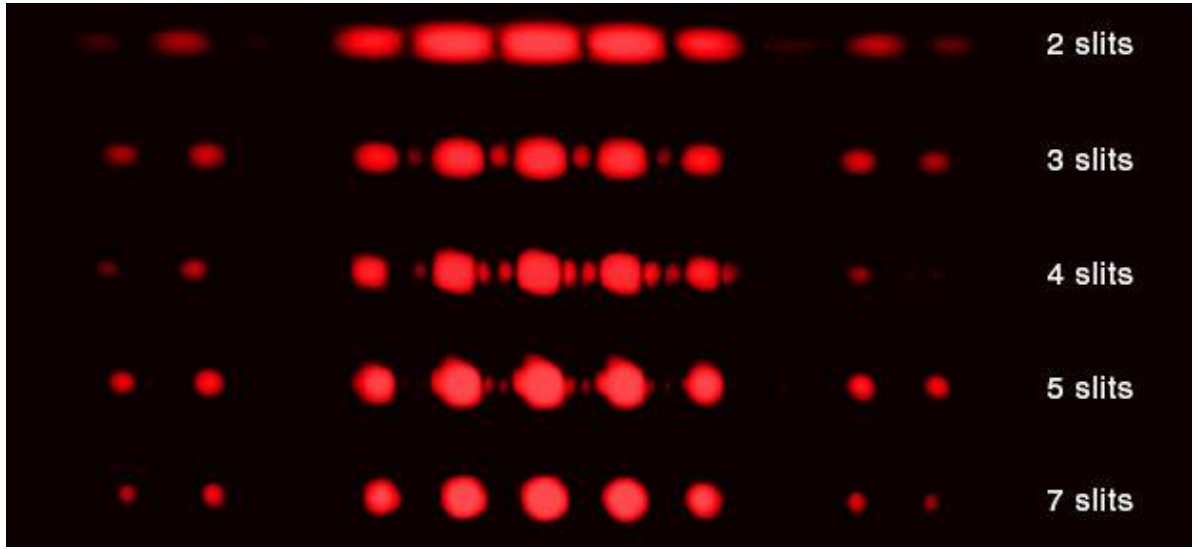
Notice that single **peaks get broader as 2θ increases**

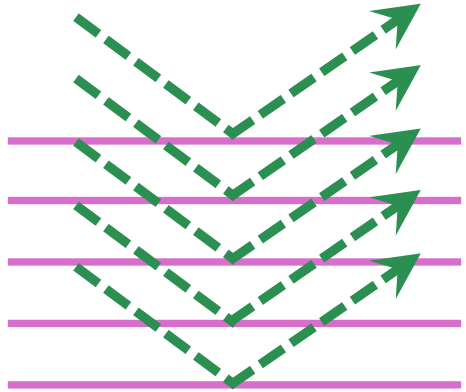
In this case, the peak broadening is caused by the sample having a **small crystallite size**



Domains < 100 nm

Peak broadening from small crystallites



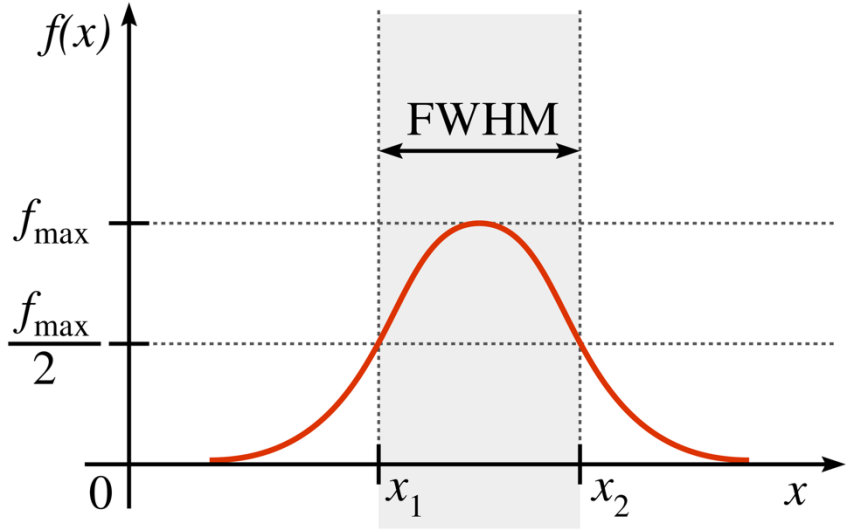


An insufficient number of places will lead to **incomplete** destructive interference near θ_{Bragg}

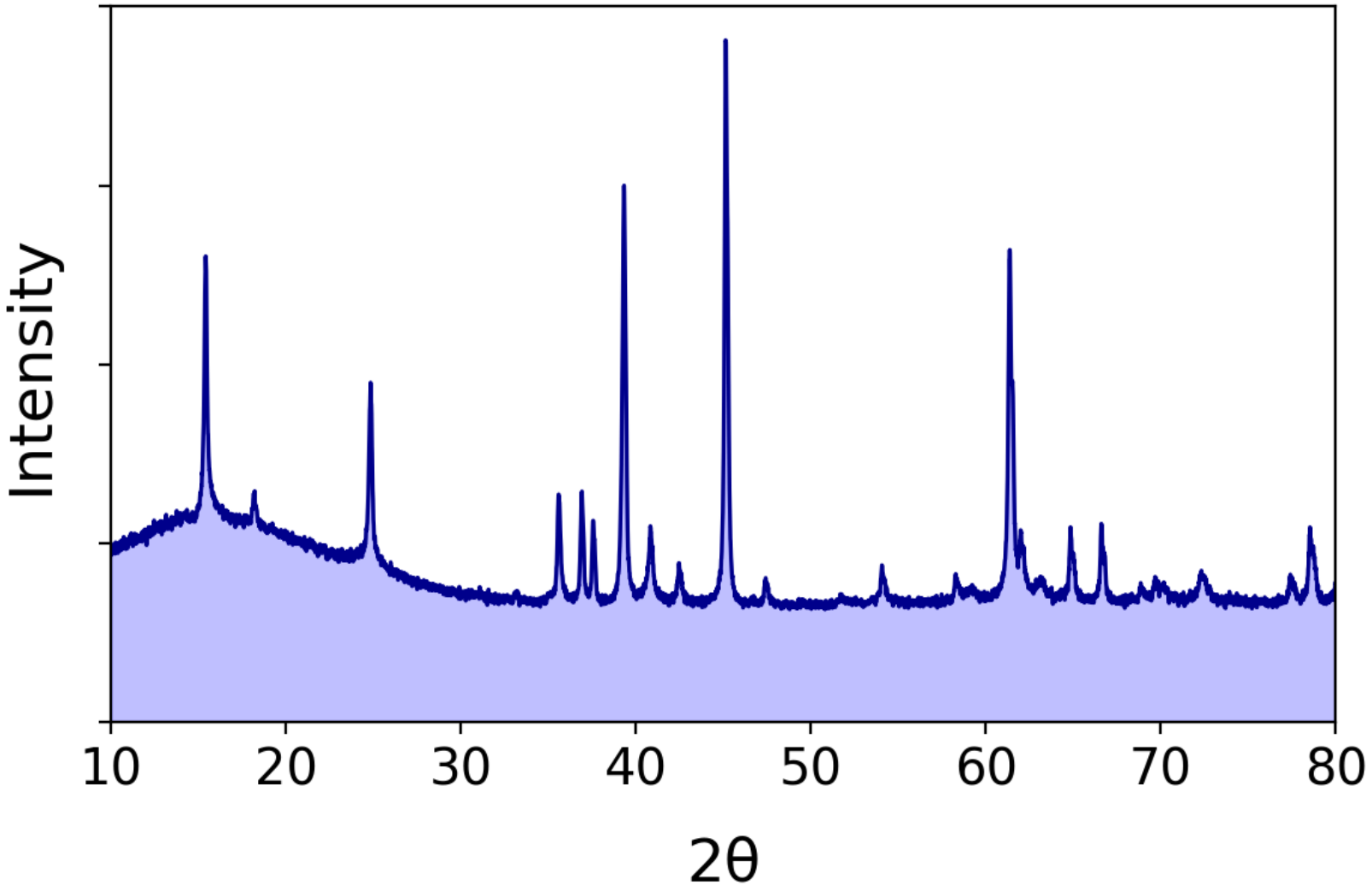
Scherrer equation:

$$D = \frac{K\lambda}{\beta \cos\theta}$$

D is the particle size
K is a shape factor
λ is the X-ray wavelength
β is the FWHM
θ is the Bragg angle



Real XRD patterns are more complicated

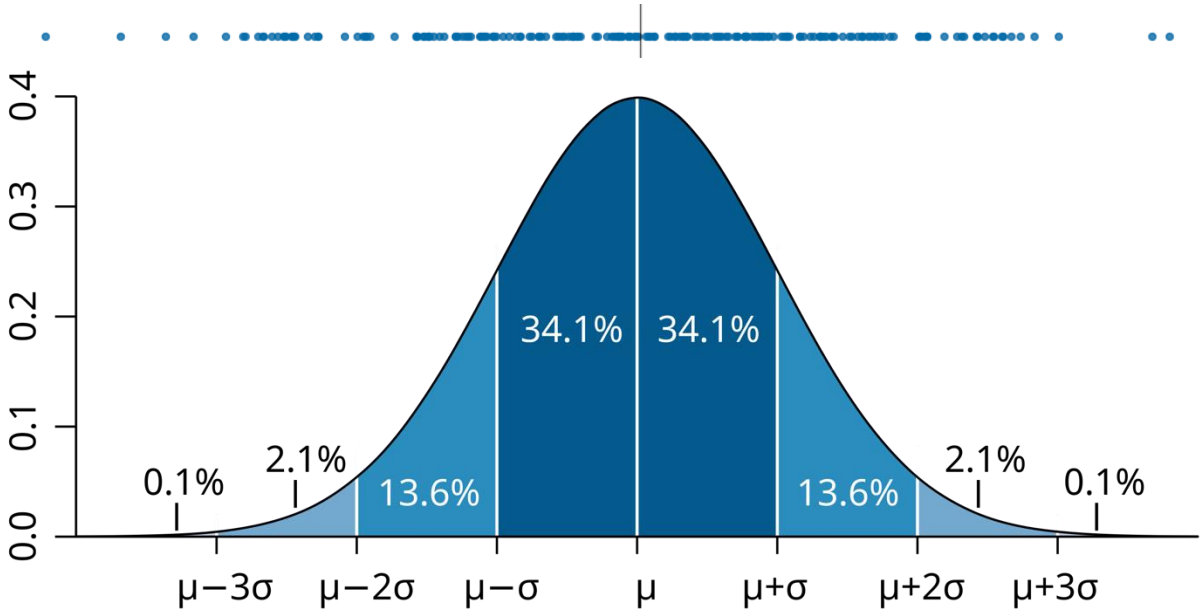


There can be **noisy background signal** with **diffuse "bumps"** that usually occur at low values of 2θ

Noise ↔ poor counting statistics

Diffuse signal ↔ amorphous films or impurities, thermal effects, and more

Creating diffuse and noisy baseline signal

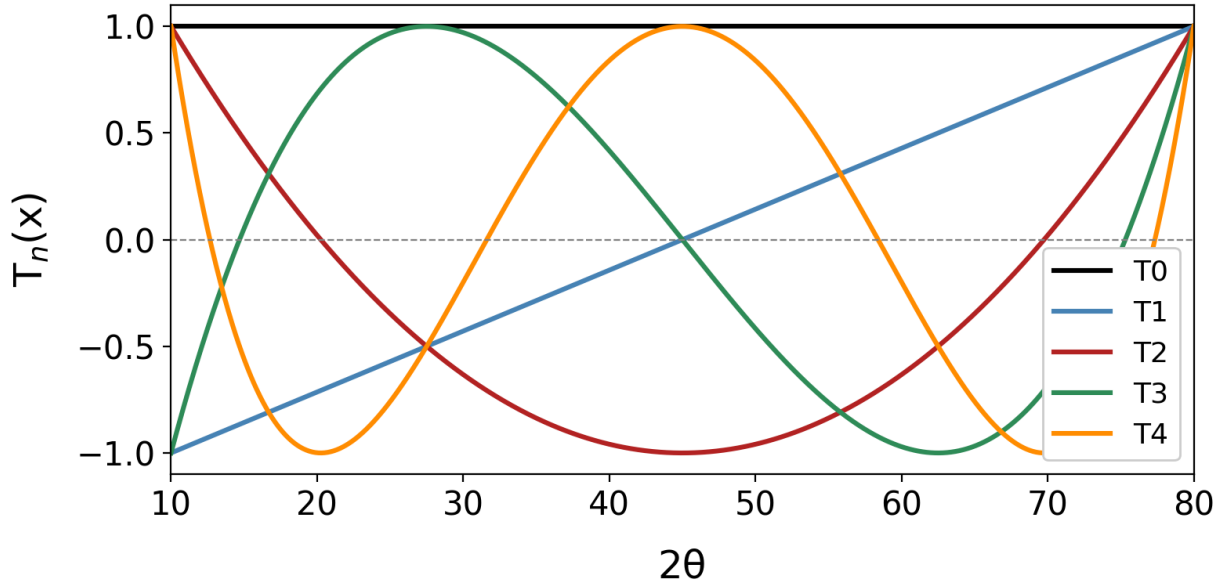


Gaussian noise:

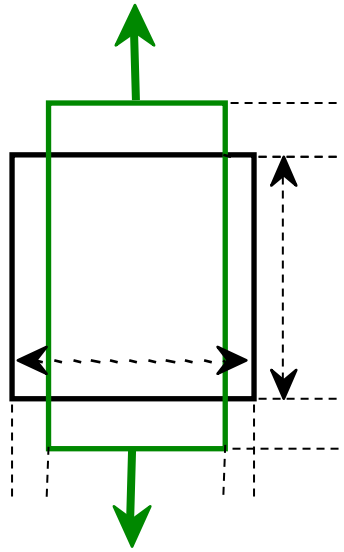
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

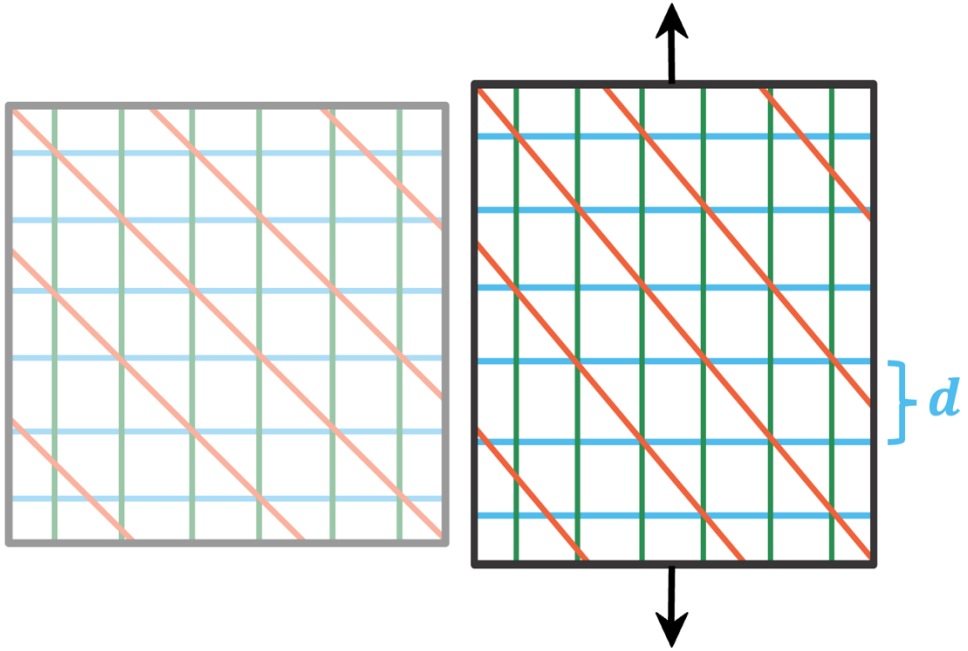
Chebyshev polynomials:

$T_0(x) = 1$
$T_1(x) = x$
$T_2(x) = 2x^2 - 1$
$T_3(x) = 4x^3 - 3x$
$T_4(x) = 16x^4 - 8x^2 + 1$



There are many other artifacts: Lattice strain

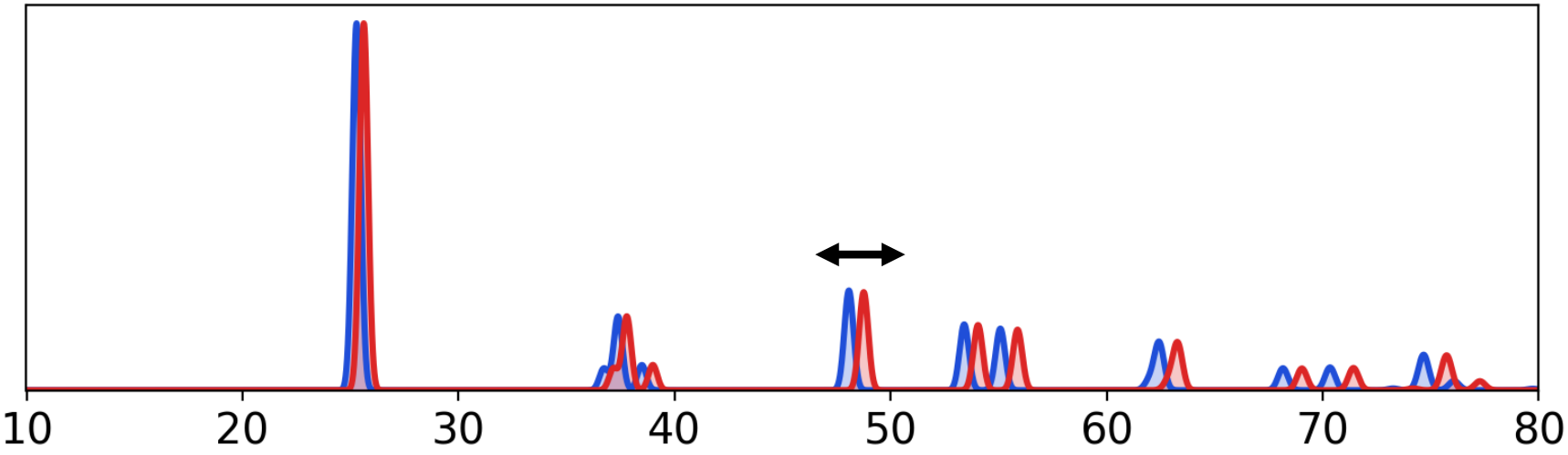




Lattice strain caused by thermal expansion, residual stresses, defects, *etc.*

$$2d_{hkl}\sin\theta = n\lambda$$

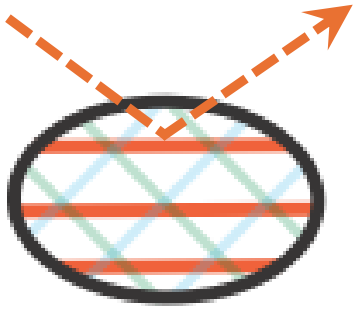
Change in d leads to change in peak position (θ)



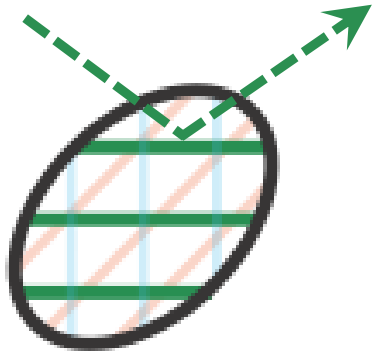
Peak positions shift in *non-uniform* ways, especially at high 2θ

There are many other artifacts: Texture

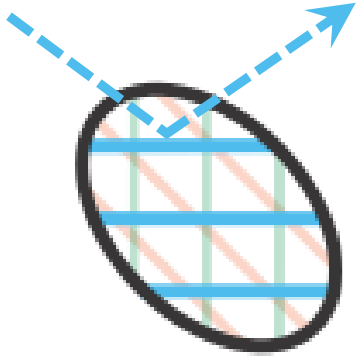
We typically assume that all **grain orientations appear equally** in the powder



In the powder, some grains will be oriented for diffraction from plane 1

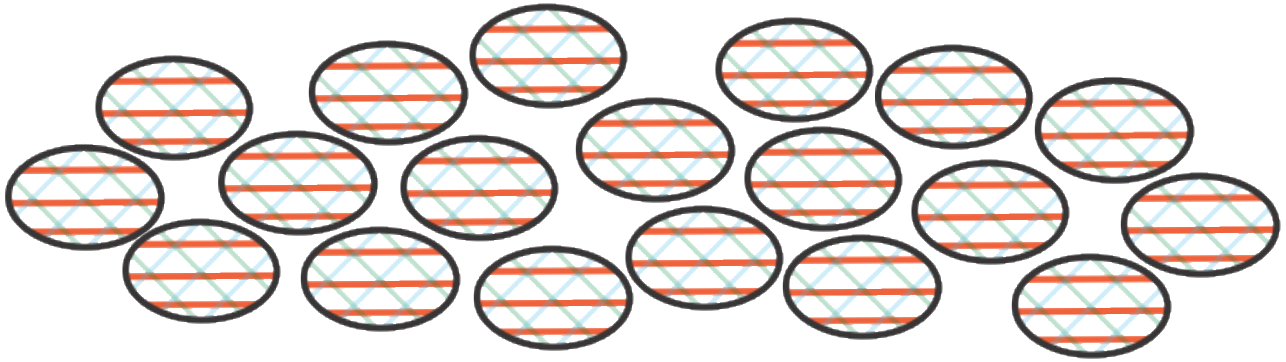


And some will be oriented for diffraction from plane 2

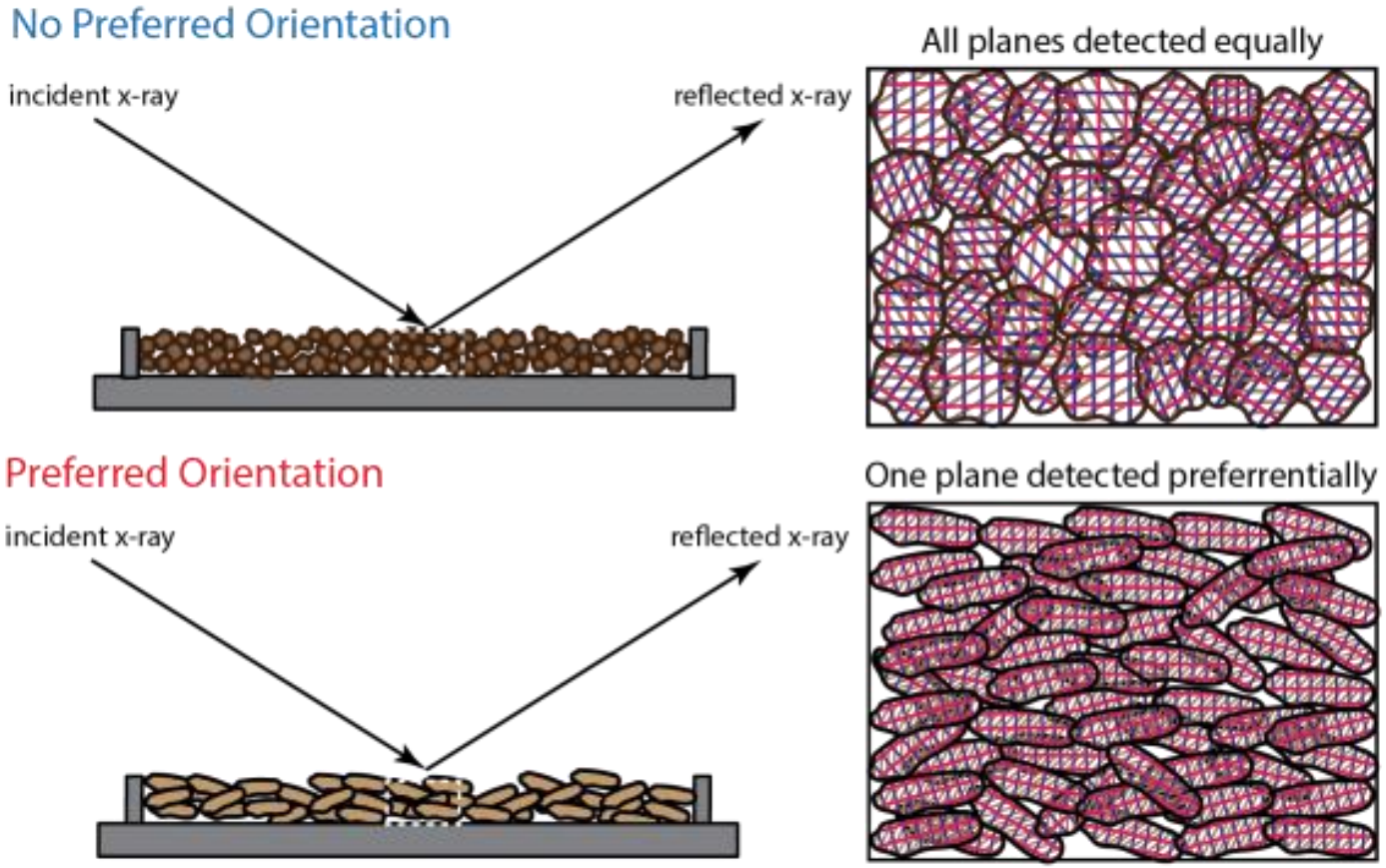


And some will be oriented for diffraction from plane 3

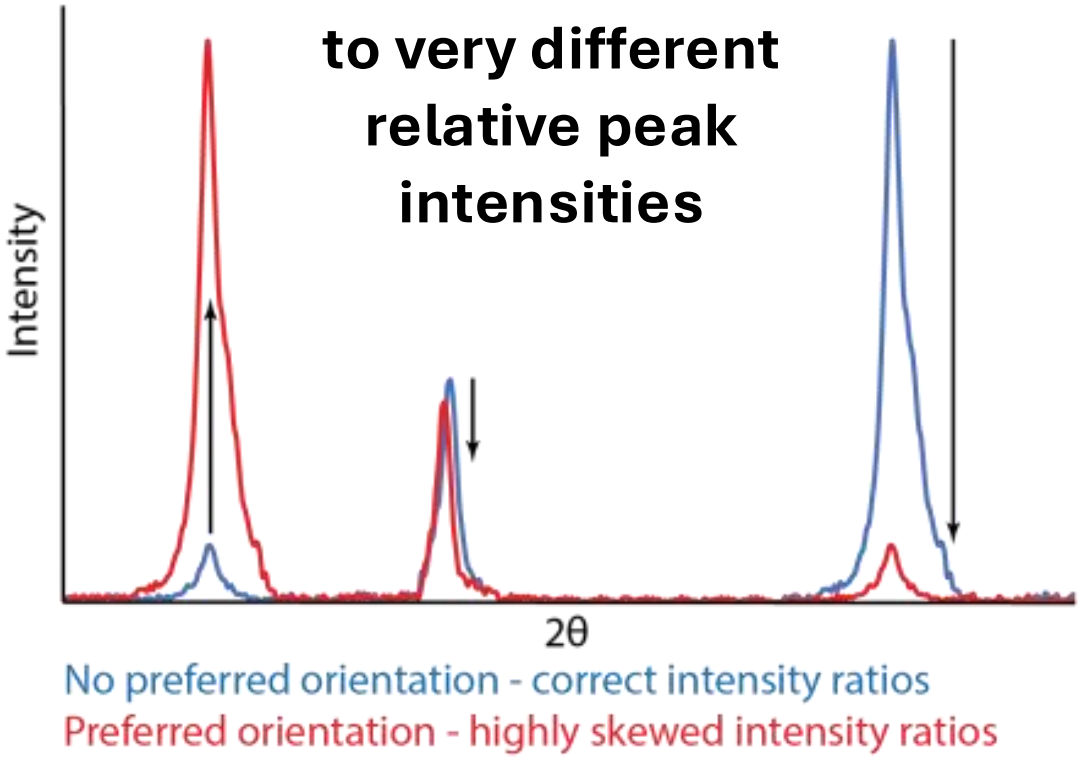
But what if there is a **preferred orientation (texture)**



There are many other artifacts: Texture

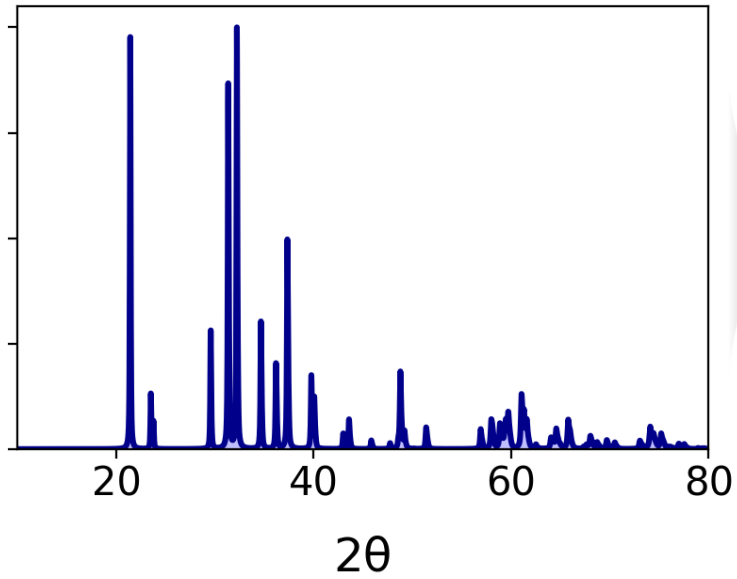


Texture can lead to very different relative peak intensities

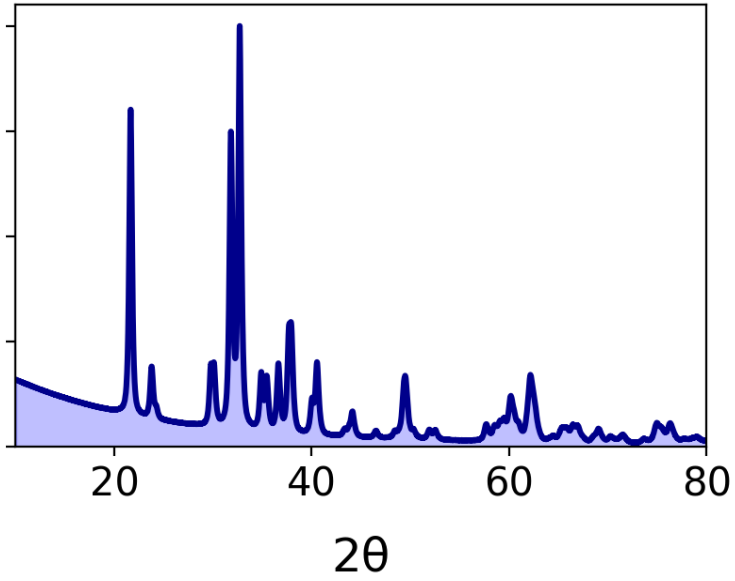


From the University of Alberta, X-ray diffraction Laboratory

Real patterns have some degree of *all* artifacts

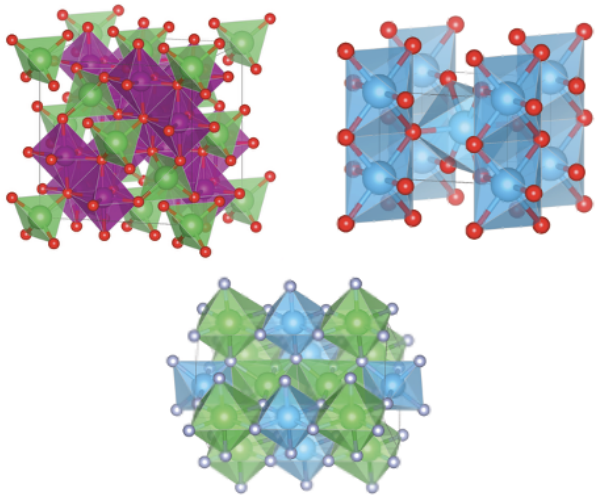


- Peak splitting ($K\alpha_1|K\alpha_2$)
- Peak broadening
- Measurement noise
- Lattice strain
- Preferred orientation
- Sample displacement
- Thermal attenuation
- Microstrain
- …



Can we go between structure ↔ pattern?

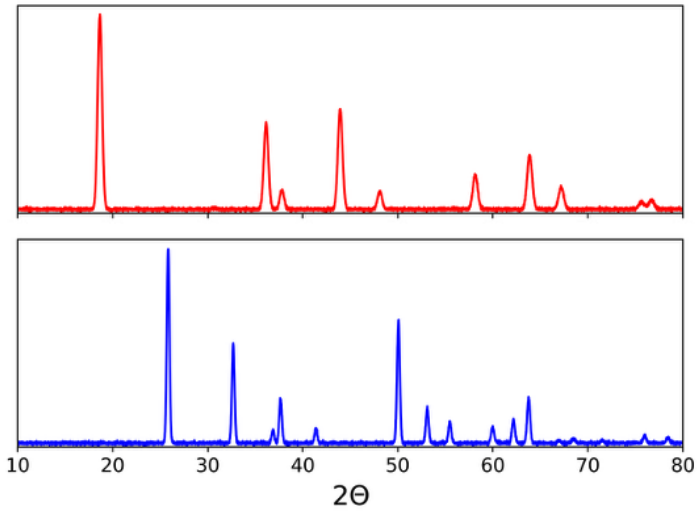
We can solve the "forward" problem: structure → XRD pattern



$$2d_{hkl}\sin\theta = n\lambda$$

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

Experimental artifacts



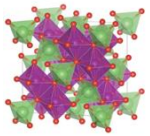
How to go from XRD to structure?

What about the "reverse " problem?

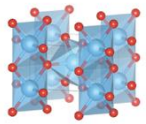
For known structures: Peak search-match

Assuming we have some **candidate structures** available, we can check which one best matches our XRD pattern through **Figures of Merit (FoM)**

PDF-5+ COD ICSD MATERIALS PROJECT



2θ	25.8°	31.9°	36.6°	38.5°
I	100	64.0	6.5	17.1



2θ	19.6°	25.7°	32.2°	38.0°
I	92.0	100	26.5	19.1

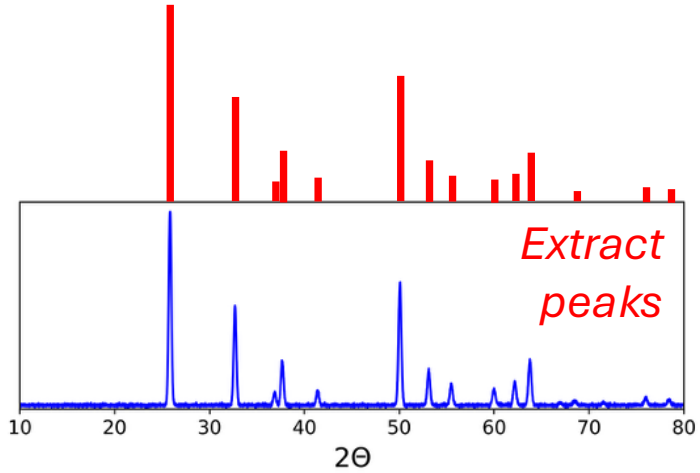
...

Popular **FoMs** include **de Wolff** and **Smith & Snyder:**

For each, we identify the **top-N peaks** and compare:

$$M_N \propto^{-1} \sum_{i}^{N} |Q^i_{\text{obs}} - Q^i_{\text{calc}}|$$

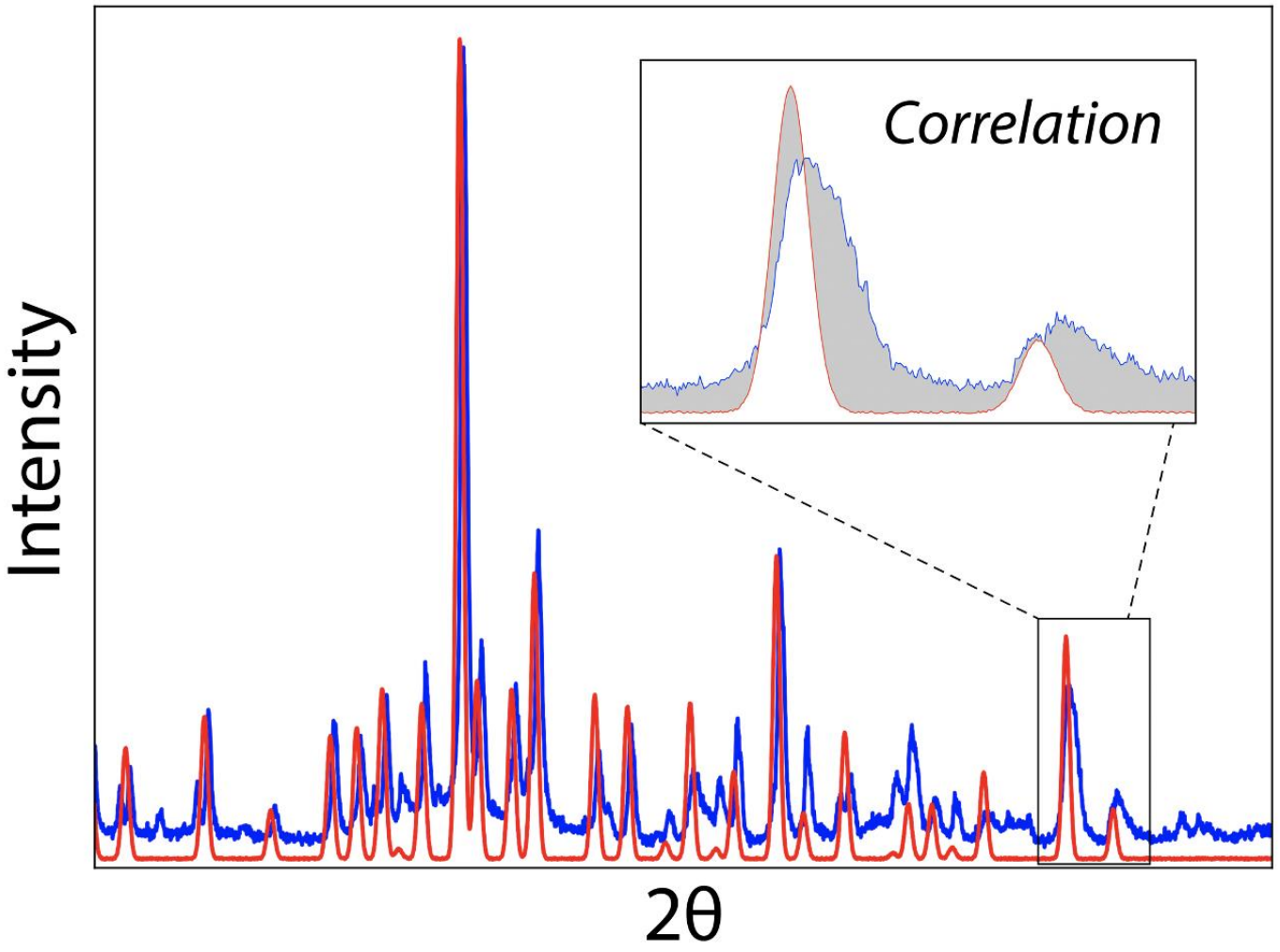
Reciprocal space: $Q = \frac{4\pi}{\lambda} \sin\theta$



2θ	26.1°	32.3°	36.9°	38.7°
I	100	62.2	5.2	18.5

For known structures: Full-profile matching

Measured pattern *vs.* Simulated pattern



We can avoid peak detection altogether by ***comparing*** entire (continuous) profiles.

Correlation metrics:

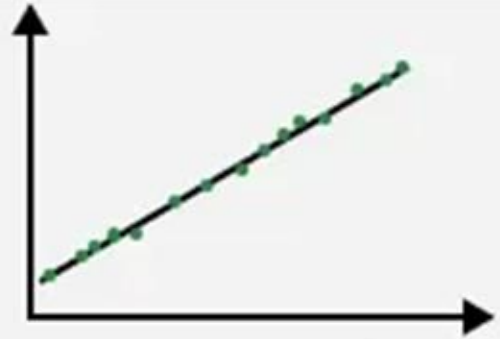
- Pearson coefficient

$$\rho = \frac{\sum(y_{\text{obs}} - \bar{y}_{\text{obs}})(y_{\text{calc}} - \bar{y}_{\text{calc}})}{\sigma_{\text{obs}}\sigma_{\text{calc}}}$$

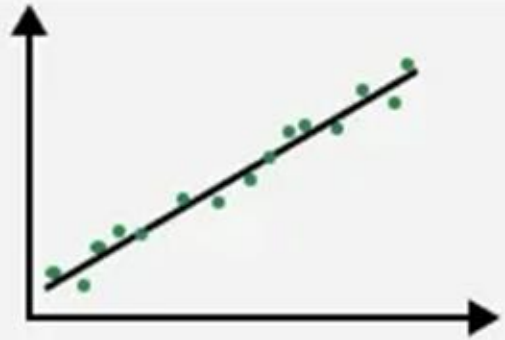
- Cosine similarity

$$\cos\theta = \frac{\sum_i y^i_{\text{obs}} y^i_{\text{calc}}}{|\boldsymbol{y}_{\text{obs}}||\boldsymbol{y}_{\text{calc}}|}$$

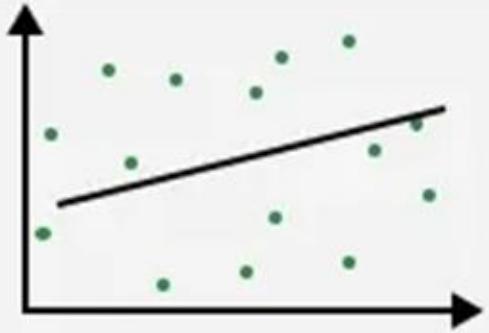
Pearson correlation coefficient



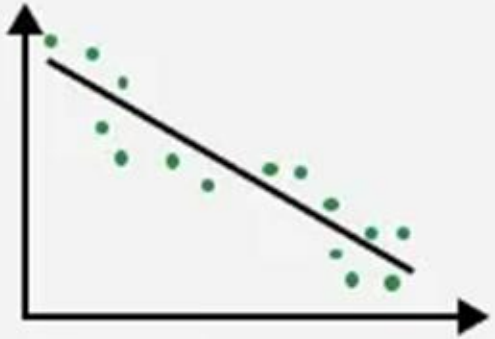
1. Strong Positive Correlation



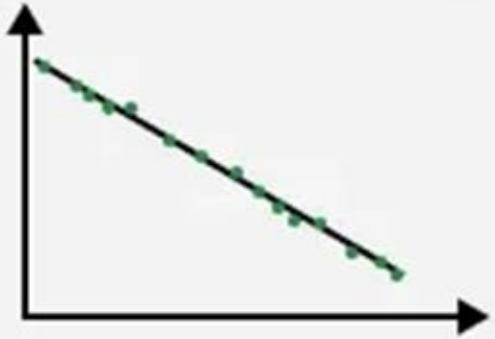
2. Medium Positive Correlation



3. Weak / No Correlation



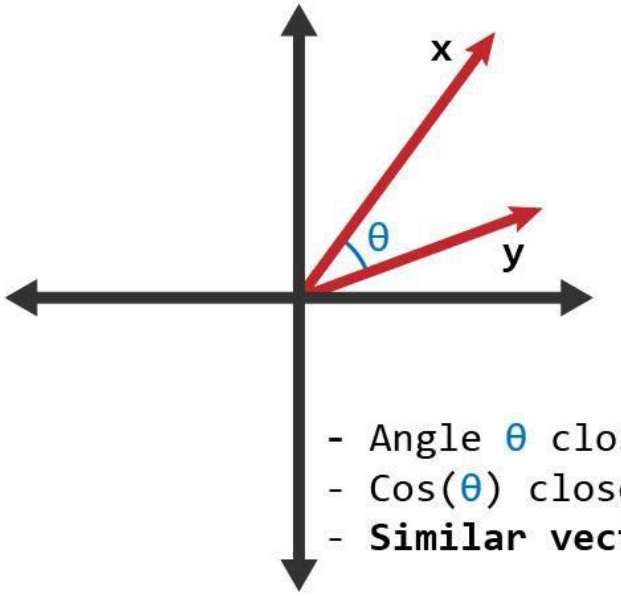
4. Medium Negative Correlation



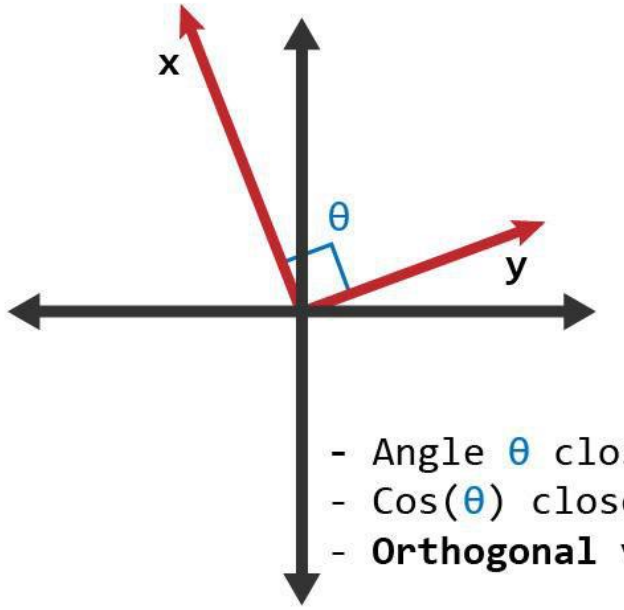
5. Strong Negative Correlation

From GeeksforGeeks.com

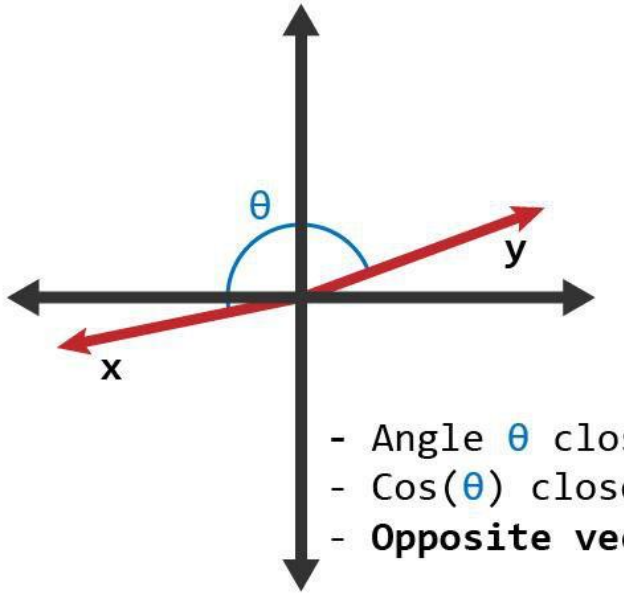
Cosine similarity



- Angle θ close to 0
- Cos(θ) close to 1
- **Similar vectors**

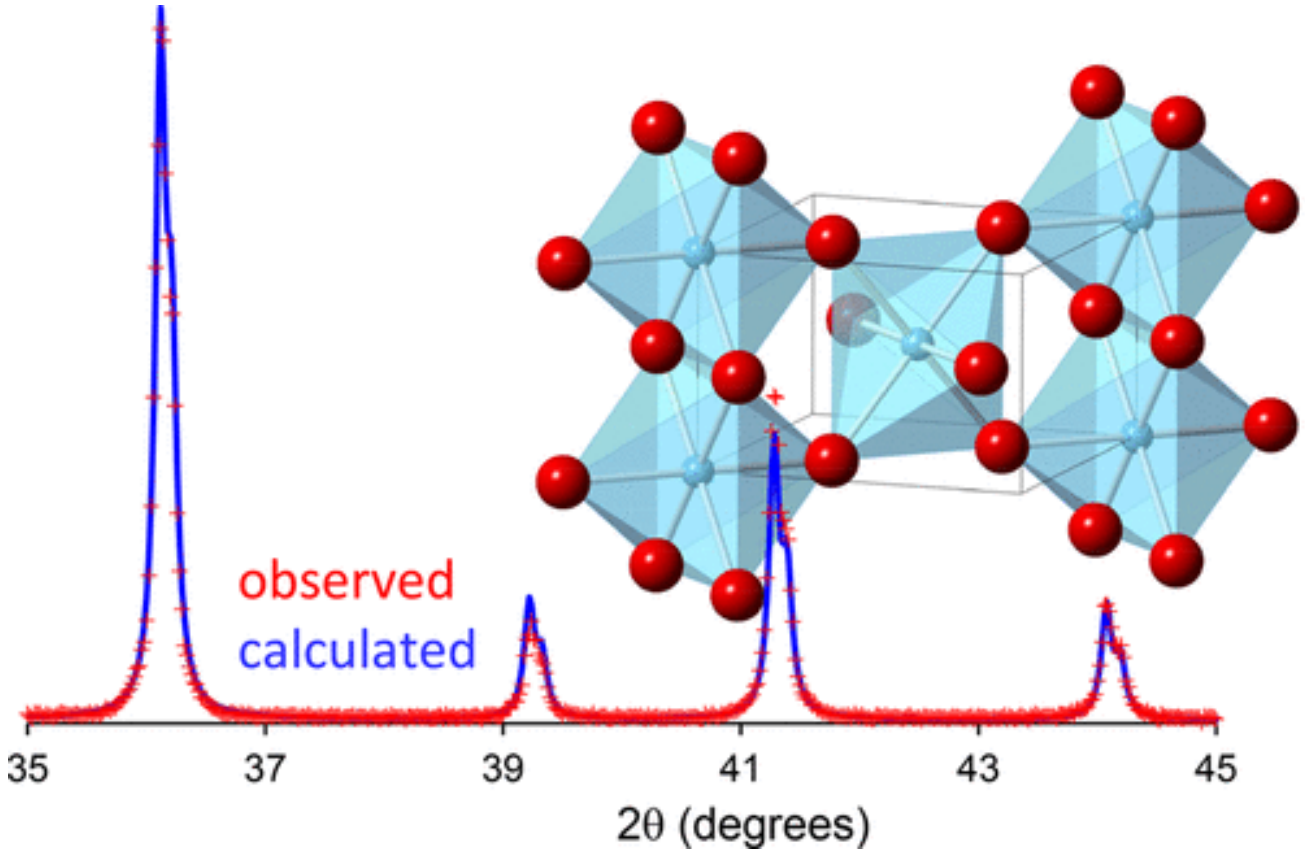


- Angle θ close to 90
- Cos(θ) close to 0
- **Orthogonal vectors**



- Angle θ close to 180
- Cos(θ) close to -1
- **Opposite vectors**

For known structures: Rietveld refinement



In **Rietveld refinement,** we actively tune the unit cell lattice parameters (+ site positions and occupancies, experimental artifacts, and more…) to **minimize the difference:**

$$y_{\mathbf{diff}} = \sum_i w_i \left(y^i_{\mathbf{obs}} - y^i_{\mathbf{calc}} \right)^2$$

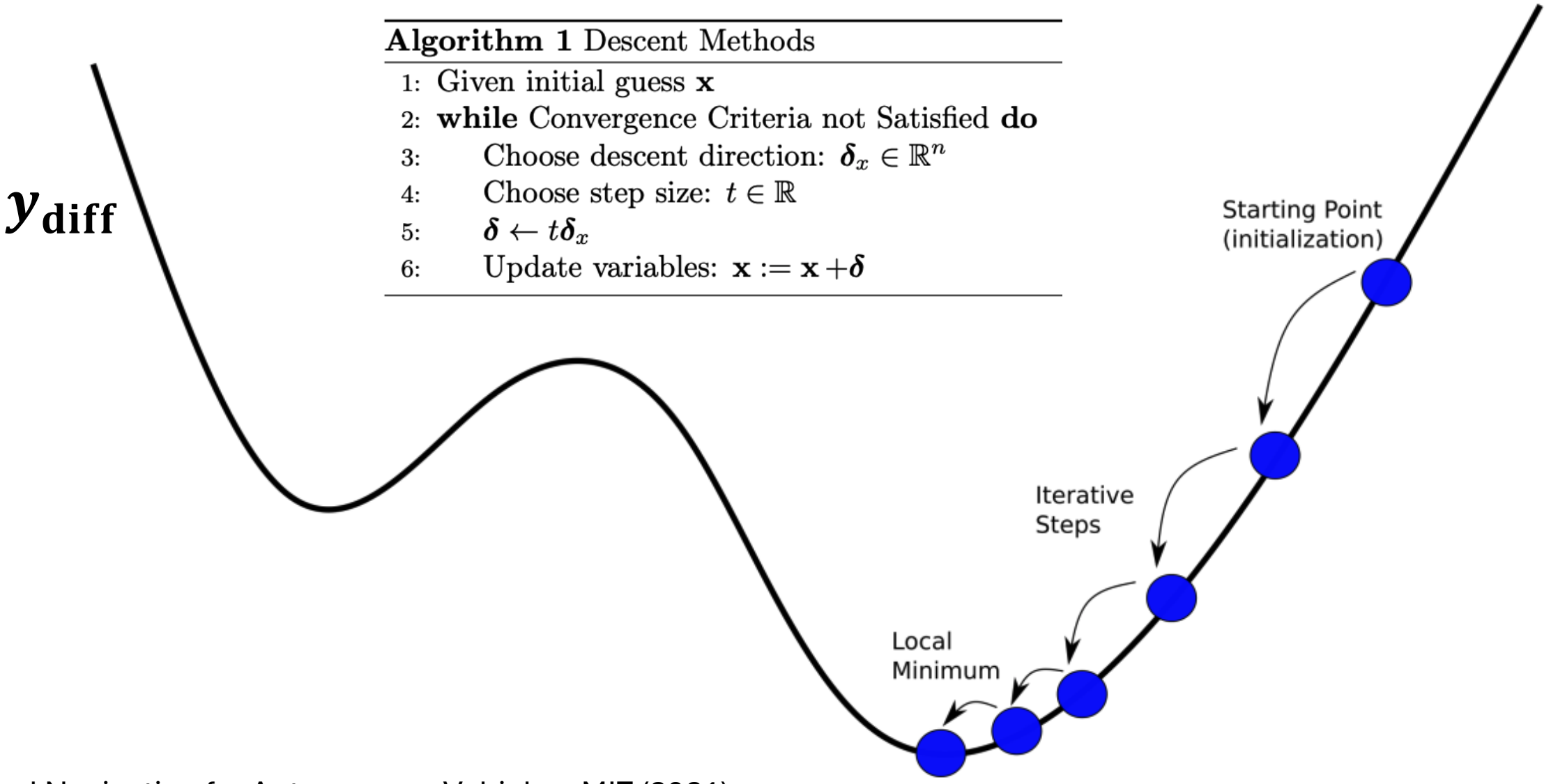
This is a ***non-linear least-squares*** optimization problem that we can **solve iteratively**

J. S. O. Evans & I. R. Evans, J. Chem. Education (2020).

Non-linear optimization

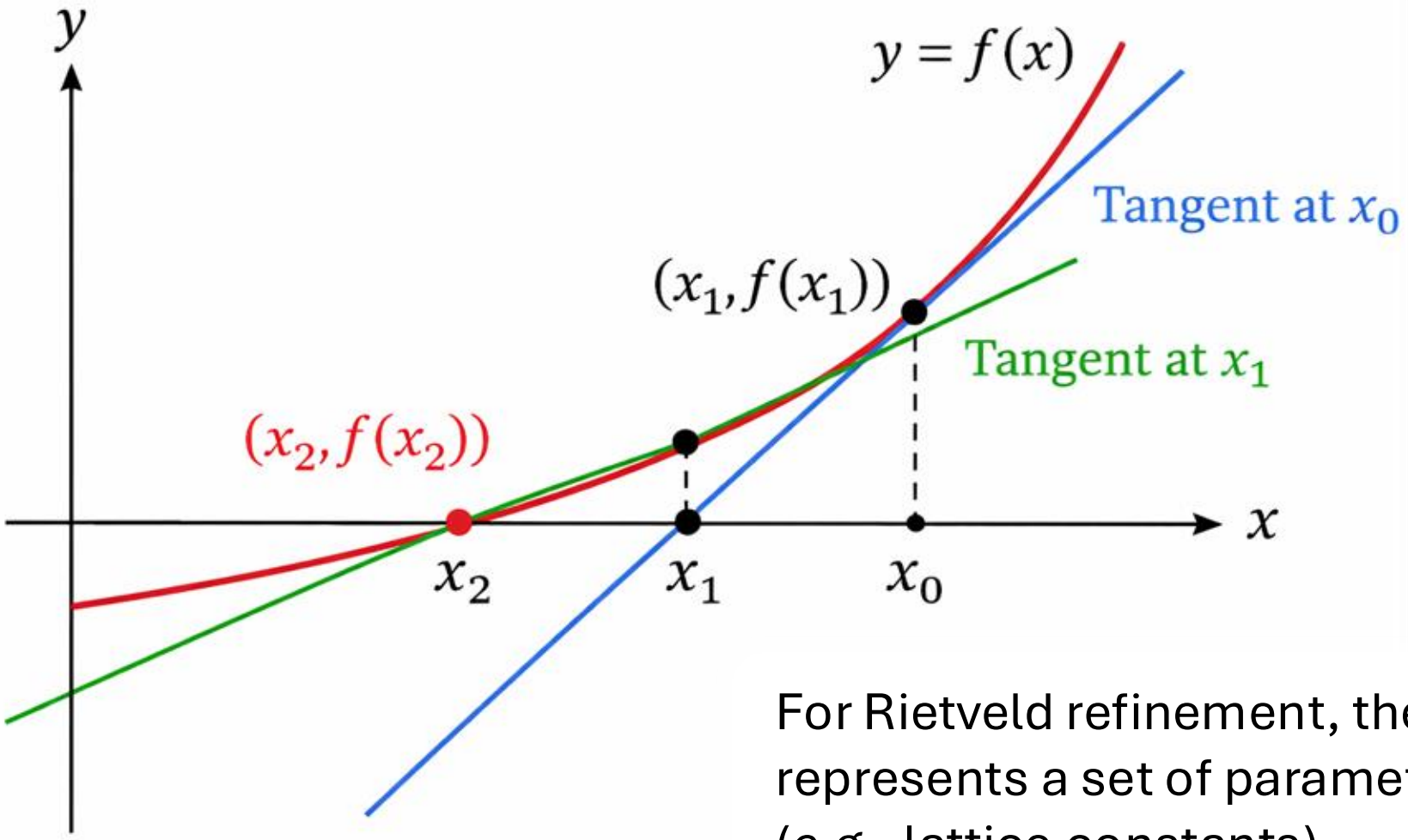
Algorithm 1 Descent Methods

1: Given initial guess $\mathbf{x}$
2: **while** Convergence Criteria not Satisfied **do**
3: Choose descent direction: $\boldsymbol{\delta}_x \in \mathbb{R}^n$
4: Choose step size: $t \in \mathbb{R}$
5: $\boldsymbol{\delta} \leftarrow t\boldsymbol{\delta}_x$
6: Update variables: $\mathbf{x} := \mathbf{x} + \boldsymbol{\delta}$



Visual Navigation for Autonomous Vehicles, MIT (2021).

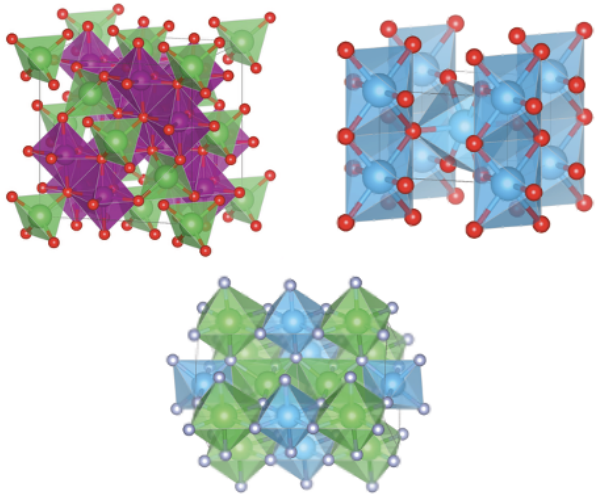
Newton's method



$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

For Rietveld refinement, the *x*-axis represents a set of parameters (*e.g.*, lattice constants)

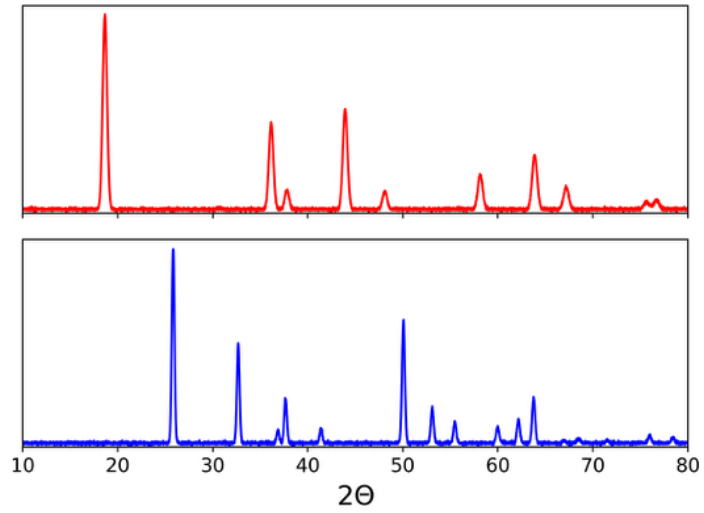
Machine learning (ML) as an alternative solution



$$2d_{hkl}\sin\theta = n\lambda$$

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j + ky_j + lz_j)}$$

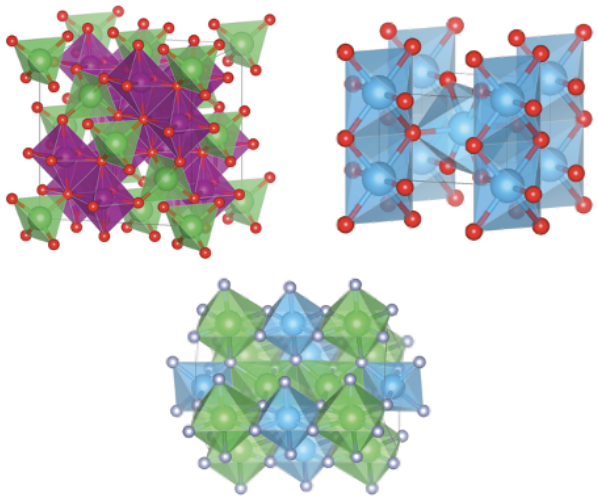
Experimental artifacts



How to go from XRD to structure?

- **Experimental artifacts** modify peaks
- **Multi-phase mixtures** are common
- The pattern **may not be unique**

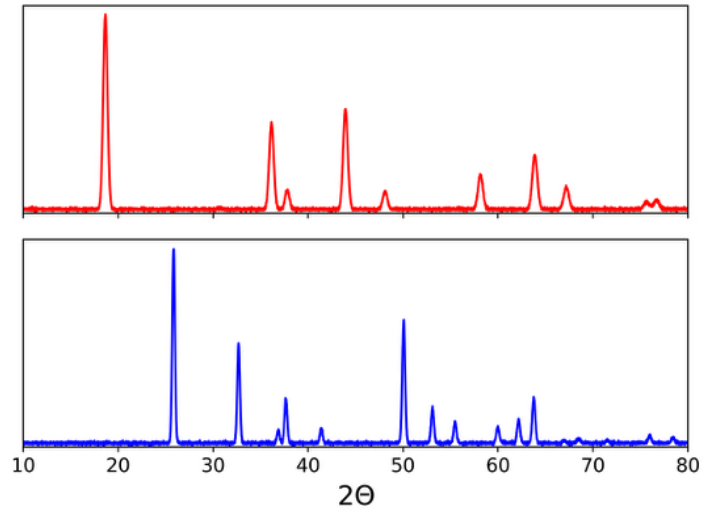
Machine learning (ML) as an alternative solution



$$2d_{hkl}\sin\theta = n\lambda$$

$$F_{hkl} = \sum_j f_j e^{2\pi i(hx_j+ky_j+lz_j)}$$

Experimental artifacts



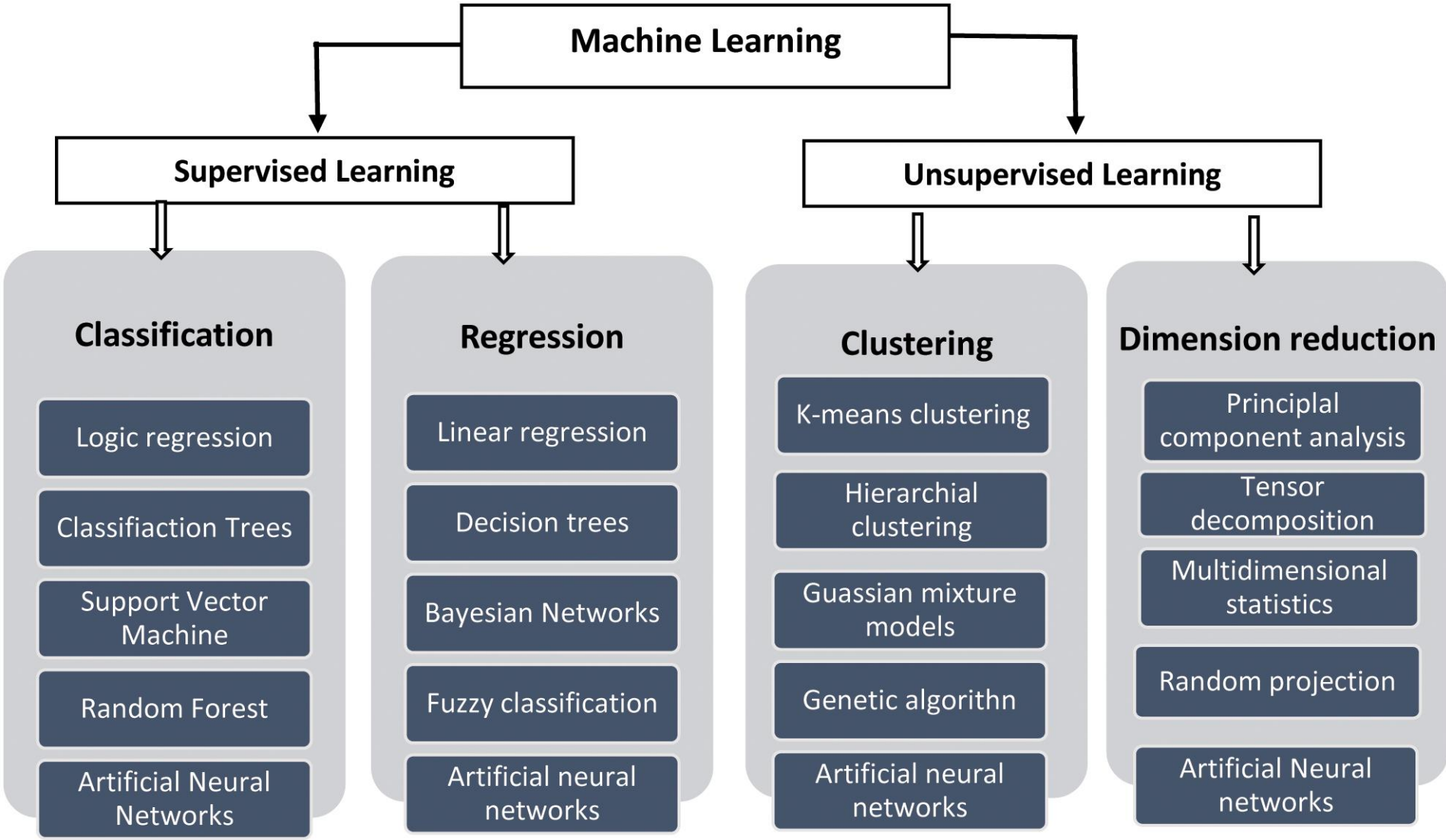
Train ML to solve reverse problem!

- **Experimental artifacts** modify peaks
- **Multi-phase mixtures** are common

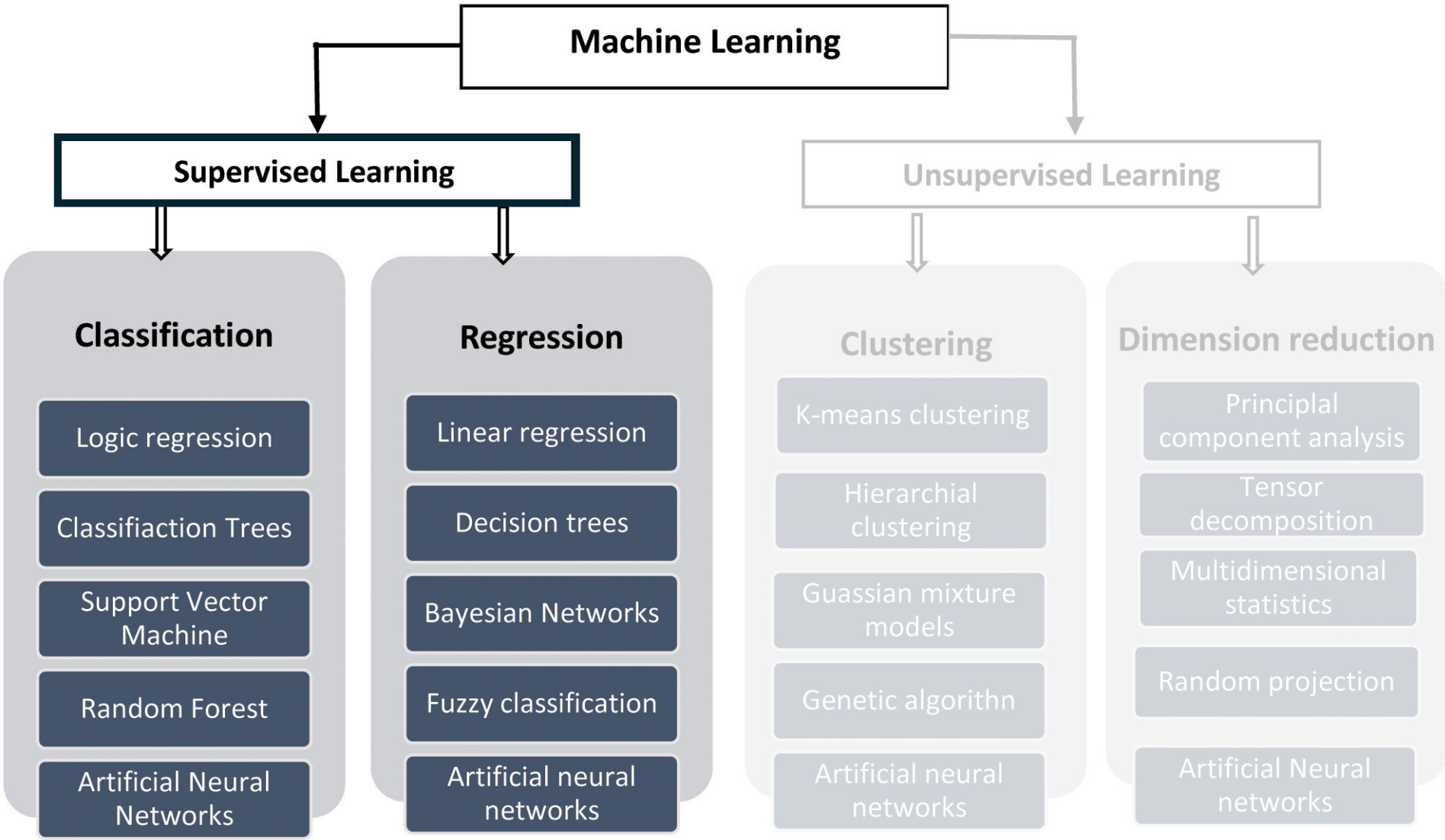
} These can be ***simulated*** and used to train ML models

- The pattern **may not be unique** ⟶ ML can be ***probabilistic***

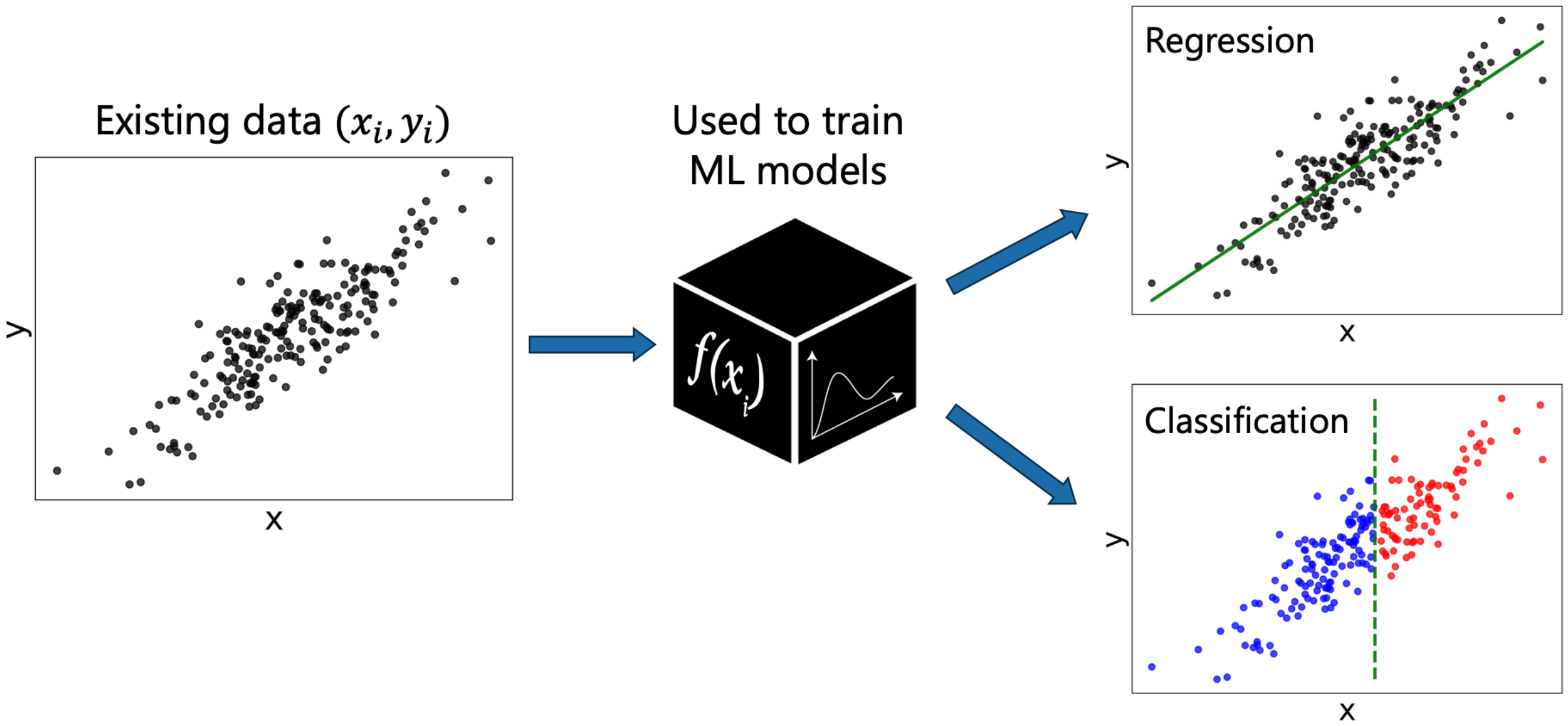
The basics of ML: learning *trends* from data



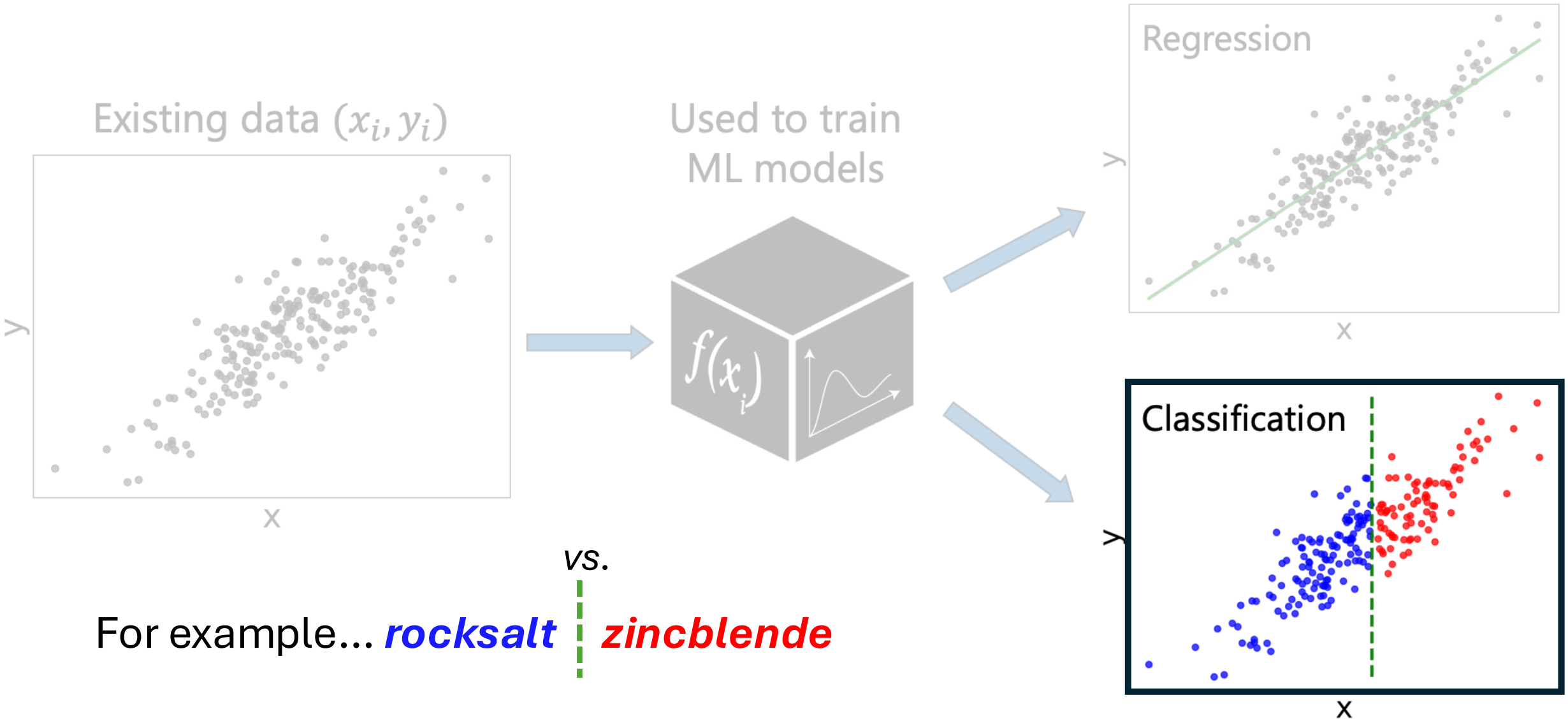
The basics of ML: learning *trends* from data



Supervised learning from *labeled* data



Supervised learning from *labeled* data

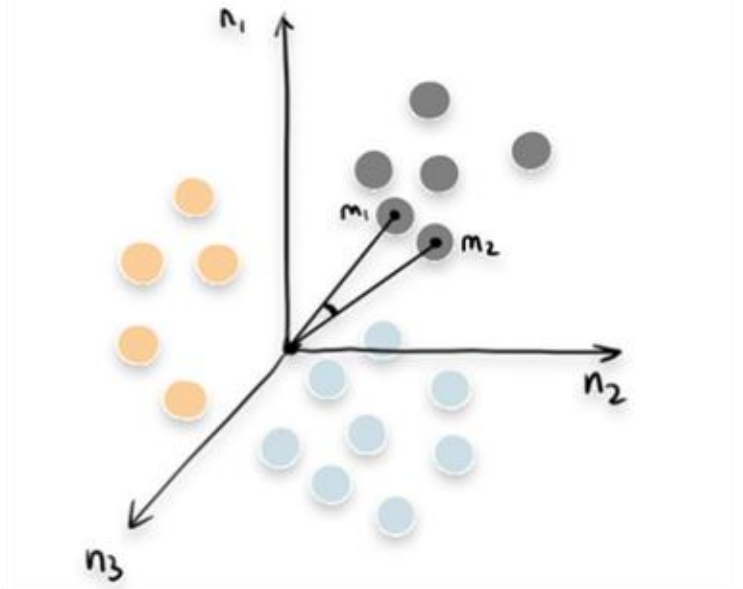


vs.

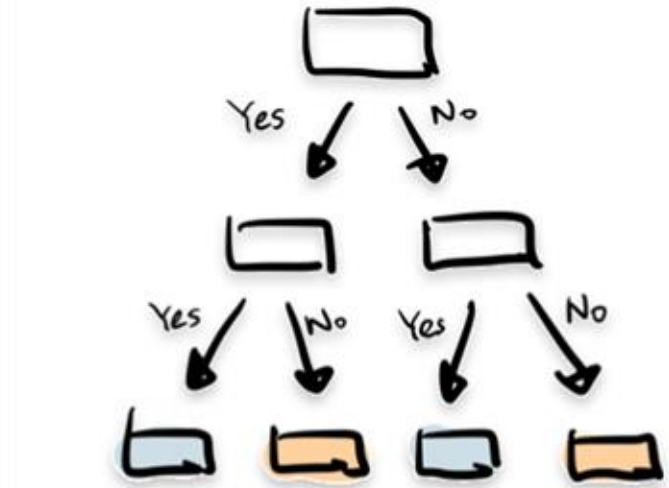
For example… ***rocksalt*** | ***zincblende***

Traditional models for *classification*

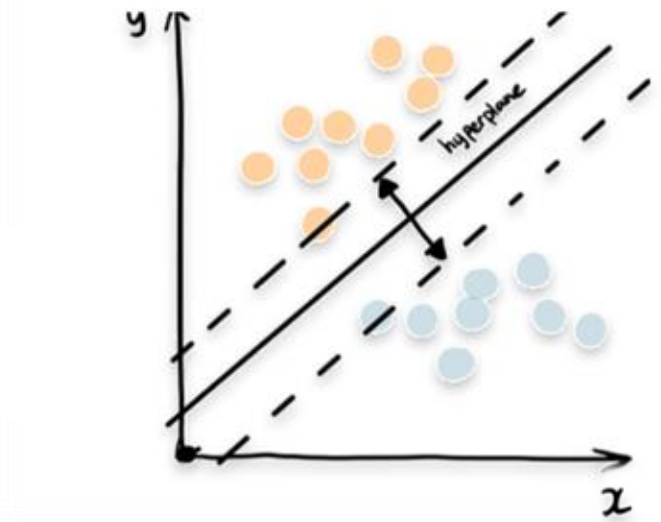
K Nearest Neighbour



Decision Tree



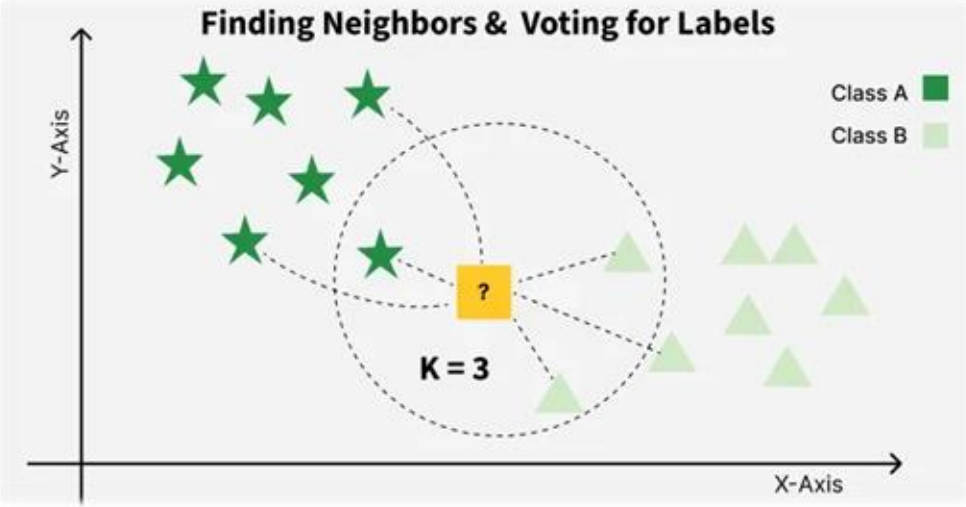
Support Vector Machine



From *Towards Data Science*.

K-Nearest Neighbor (KNN) algorithms

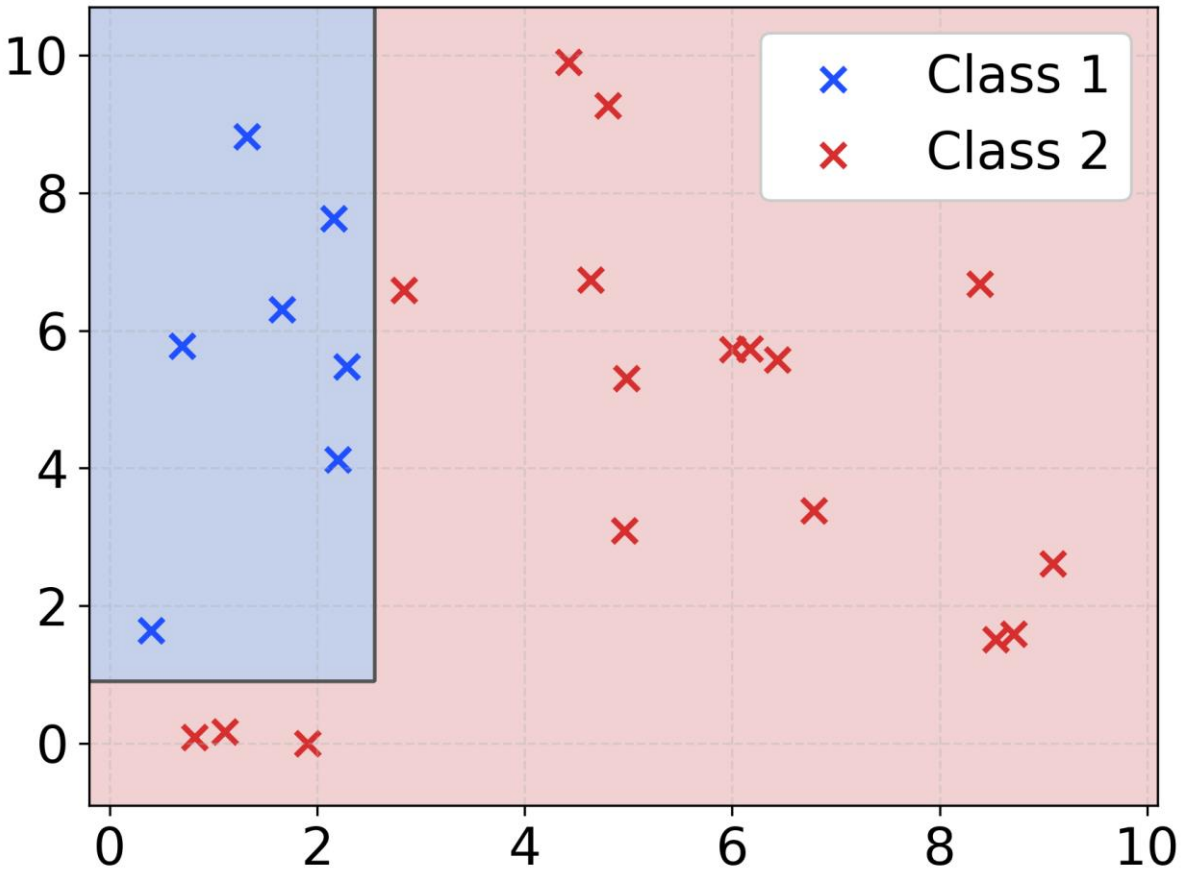
- Choose *k* (the number of neighbors)
- Compute distance to all points for a new sample
- Select the *k*-nearest neighbors
- Make classification based on majority vote



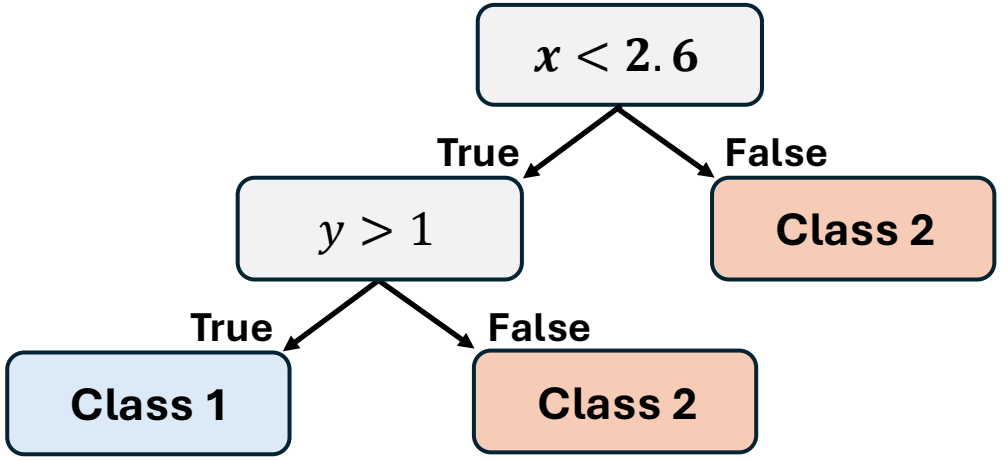
Each axis is a feature/descriptor of the data.
In our case, it would be the pattern itself, $I(2\theta)$, or some lower-dimensional representation of the pattern (*e.g.*, unsupervised learning).

From *GeeksForGeeks.com*

Decision trees

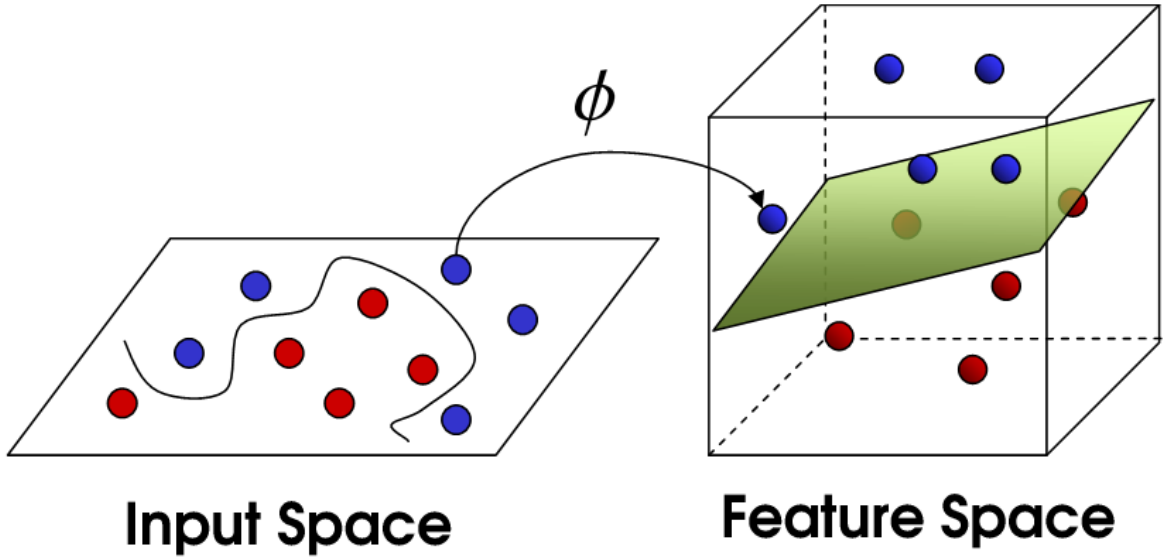


- Start with all labeled data at the root
- Find the feature + threshold that best separates different classes
- Split the data into branches
- Repeat recursively on each branch

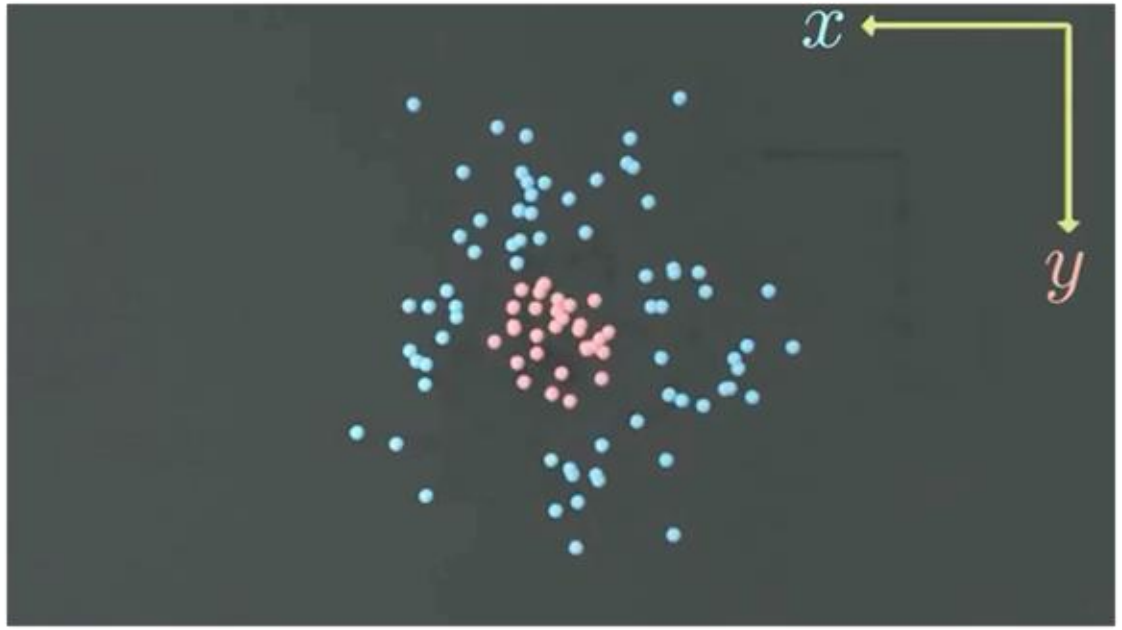


Support vector machines (SVMs)

What if the boundaries separating classes are highly **non-linear**?

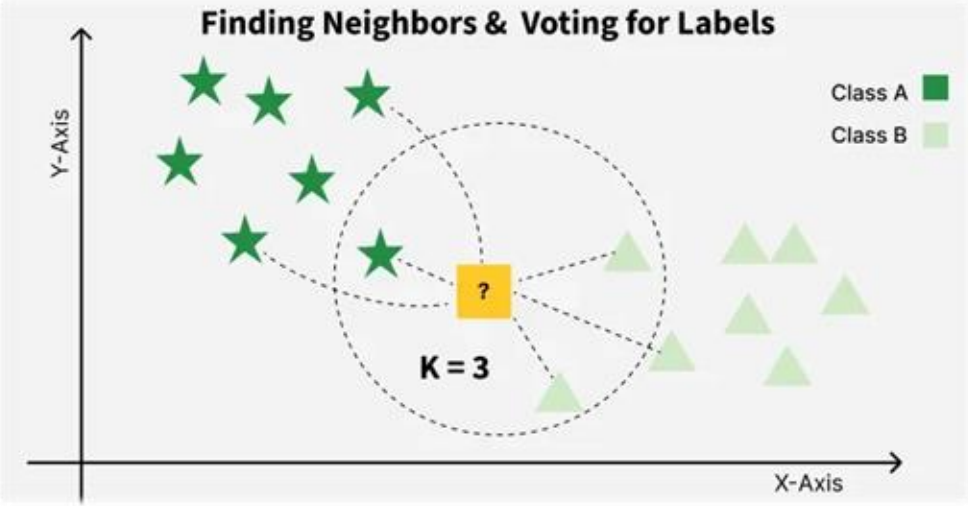


SVMs map data into a higher dimensional space where the **boundaries become linear**



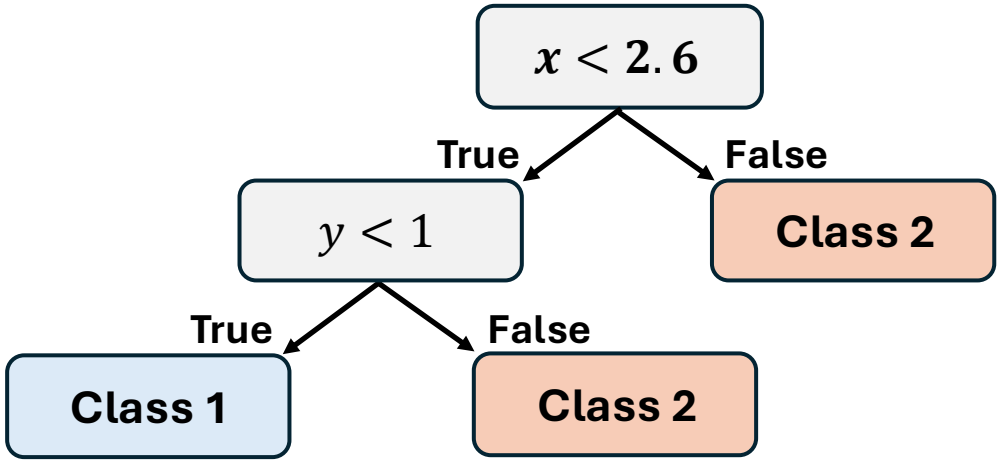
- Use a ***kernel function*** to compute distance in the high-dimensional space
- Find the optimal **hyperplane** in that space, separate different classes

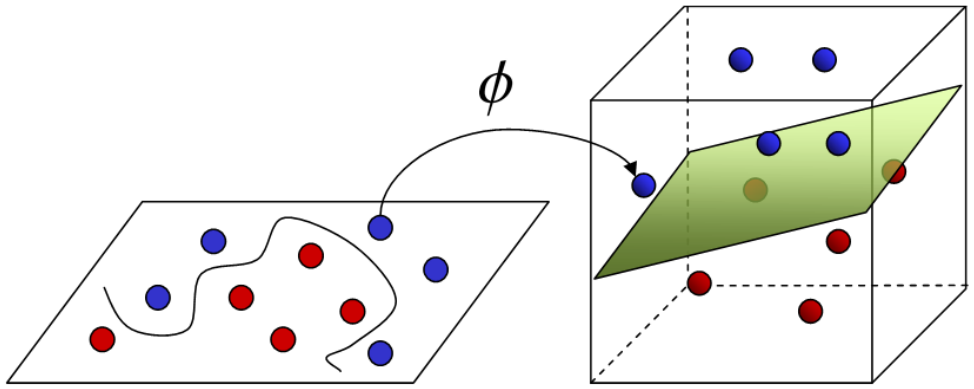
Model training and evaluation



How do we know which *k* is best?

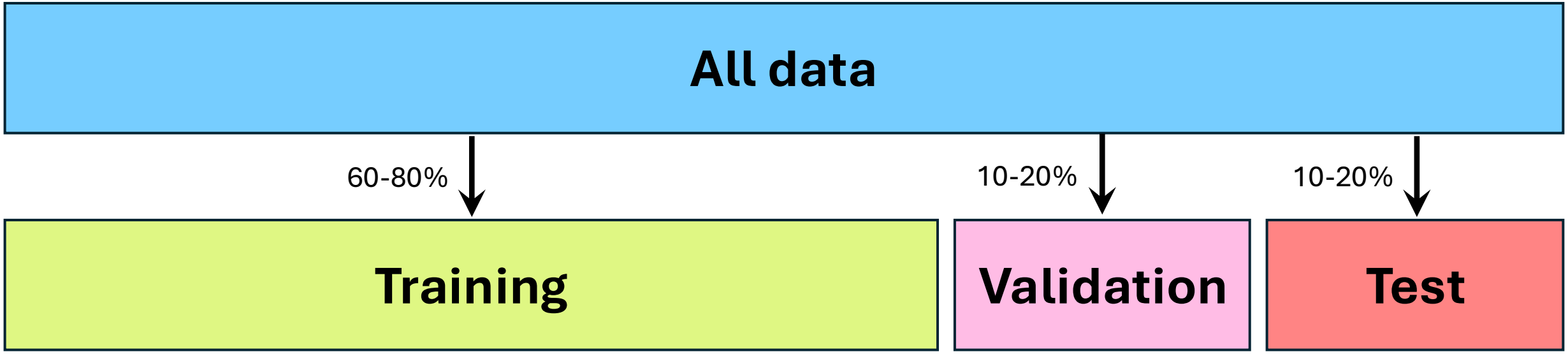
Or how many "decisions" to be made?
At what thresholds?





What is the right kernel or non-linear mapping?

Model training and evaluation



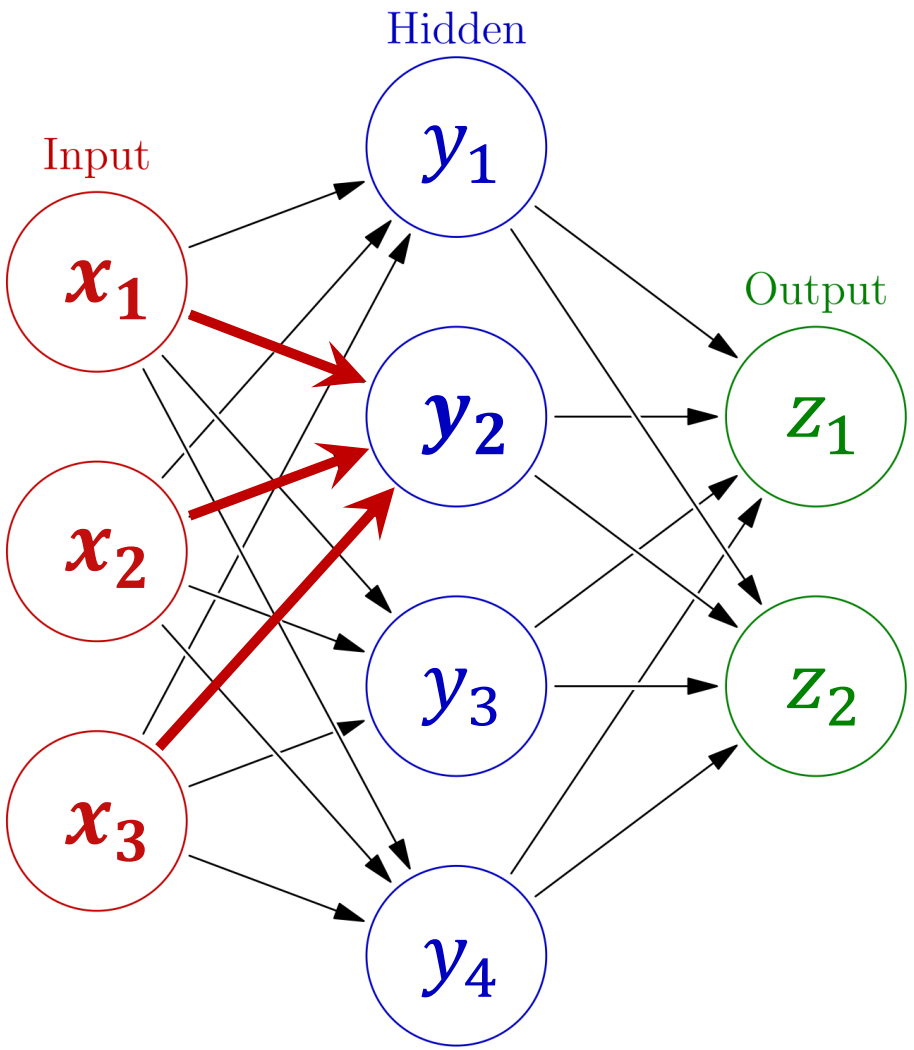
Model learns the task

Which model is the best?

How good is the model on "new" data?

Test data should be as realistic as possible!
Preferably from experimental measurements.

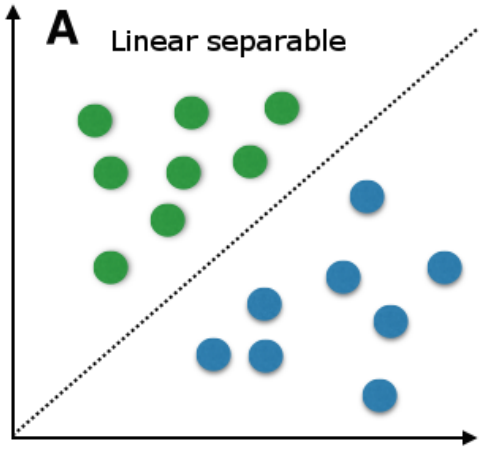
Deep learning models for classification

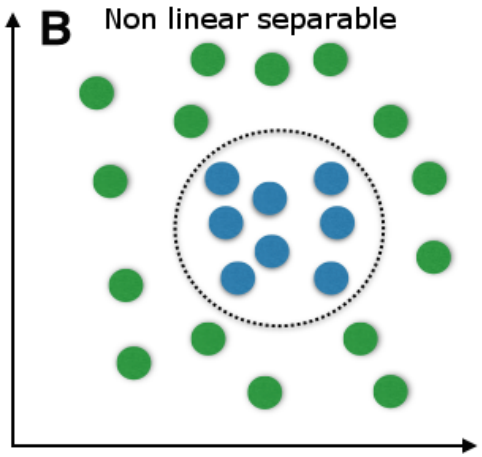


Without activation, the neuron in each layer is just a **weighted sum** of the neurons in the previous layer:

$$y_2 = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

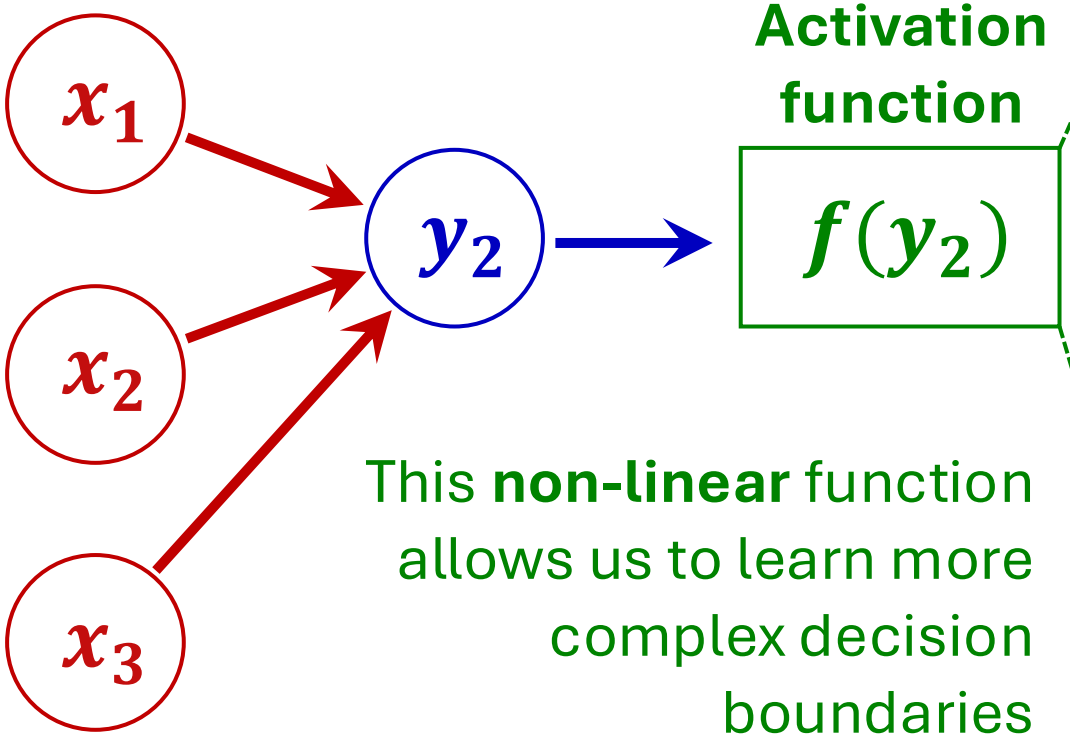
The product of two linear mappings is just another linear map…





What about this case?

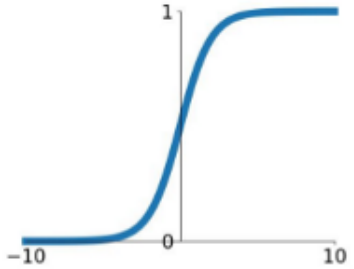
Deep learning models for classification



This **non-linear** function allows us to learn more complex decision boundaries

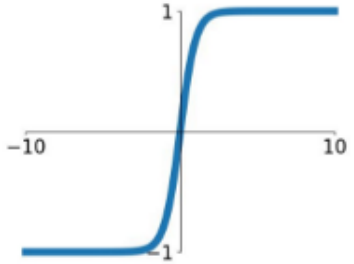
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



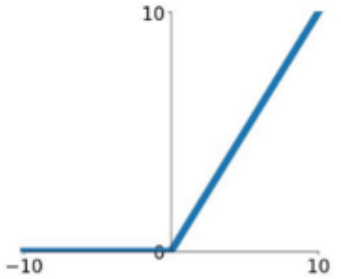
tanh

$$\tanh(x)$$



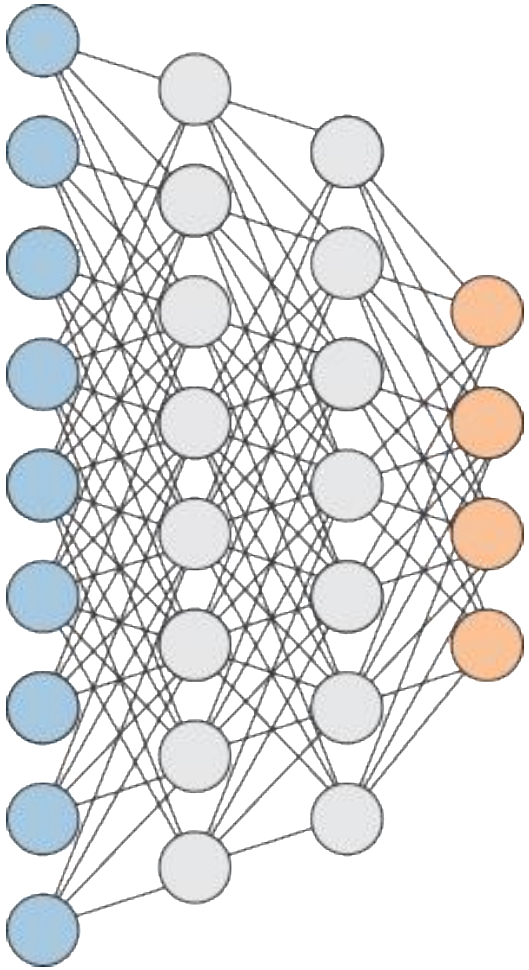
ReLU

$$\max(0, x)$$



Training process for deep learning models

Input: *entire* XRD pattern



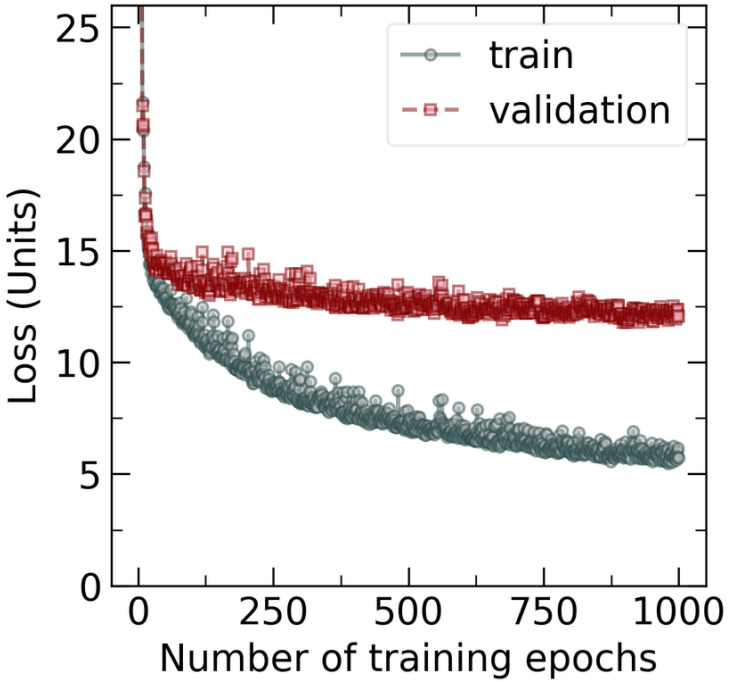
Phase #1

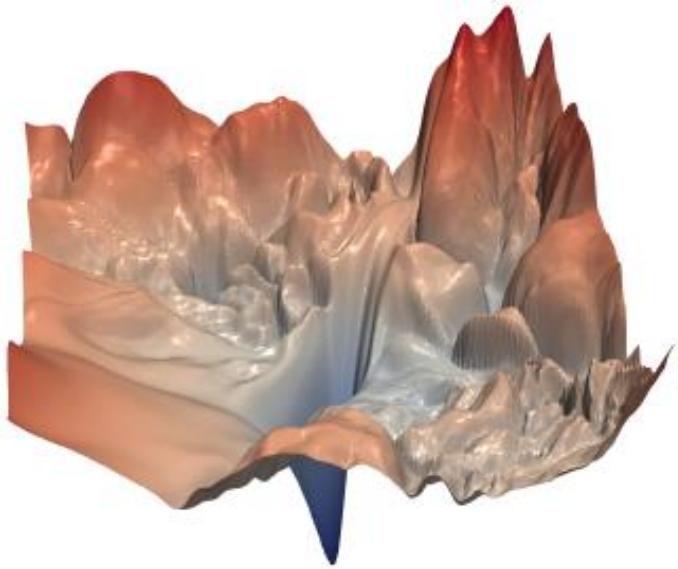
Phase #2

Phase #3

Phase #4

Finding the optimal weights (w_{ij}) that minimize some loss (error) function is a tricky and highly **non-convex optimization** problem…

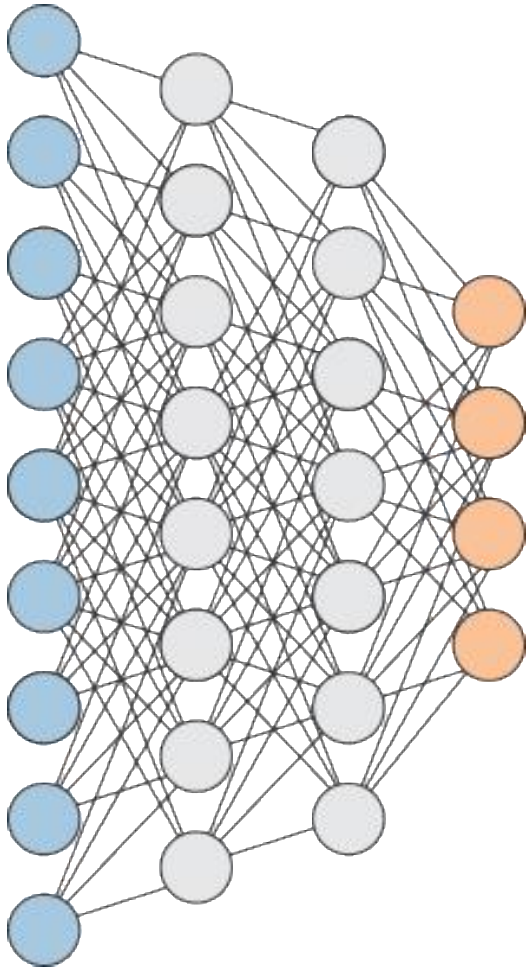




Convolutional neural networks (CNNs)

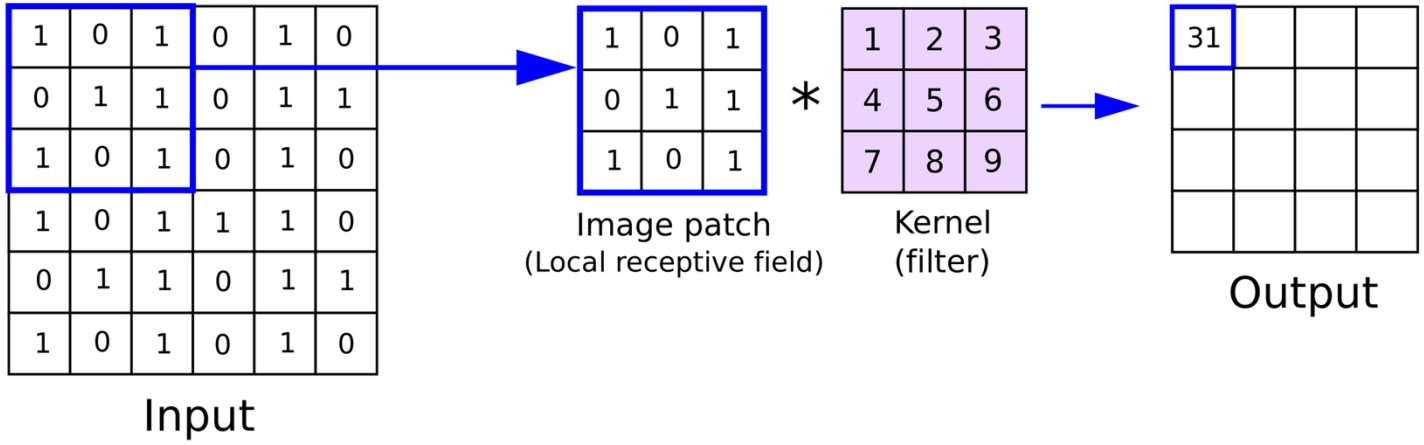
Input: *entire* XRD pattern

Conv. Filters
(Feature extractors)



Before passing data to the neural network, we extract key features using ***convolutional filters.*** These are translationally invariant!

It is **the features** which then get passed to the neural network, not the raw data.



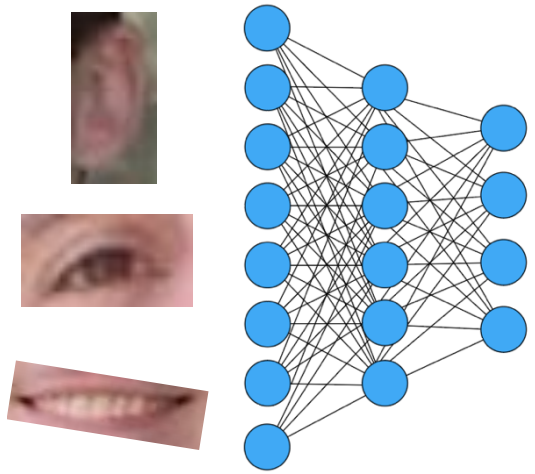
Convolutional neural networks (CNNs)

4032×3024 ~ **12 million pixels!**

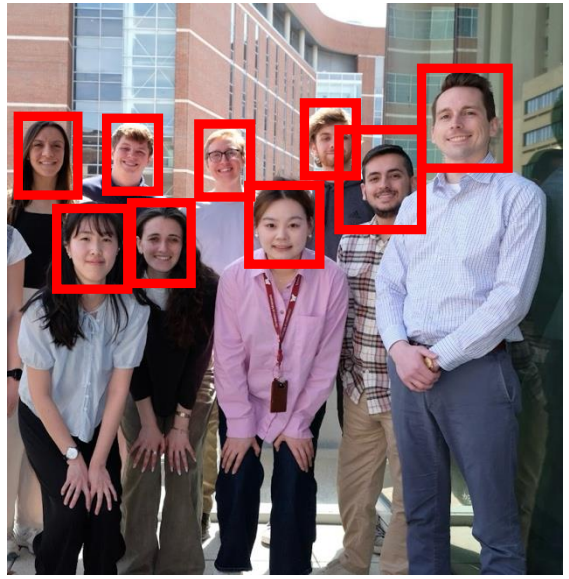
$$\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Convolution

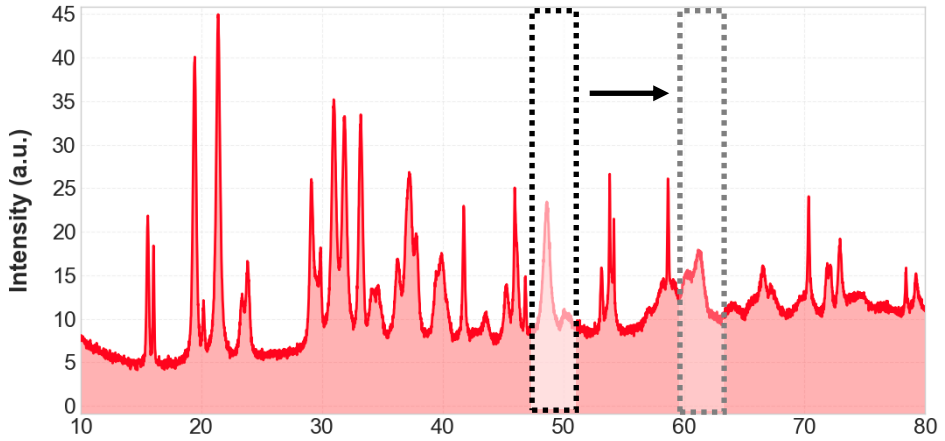
Edges,
corners, *etc.*



Faces identified

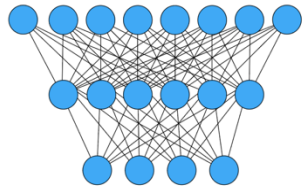


For 1D patterns,
these matrices
become vectors



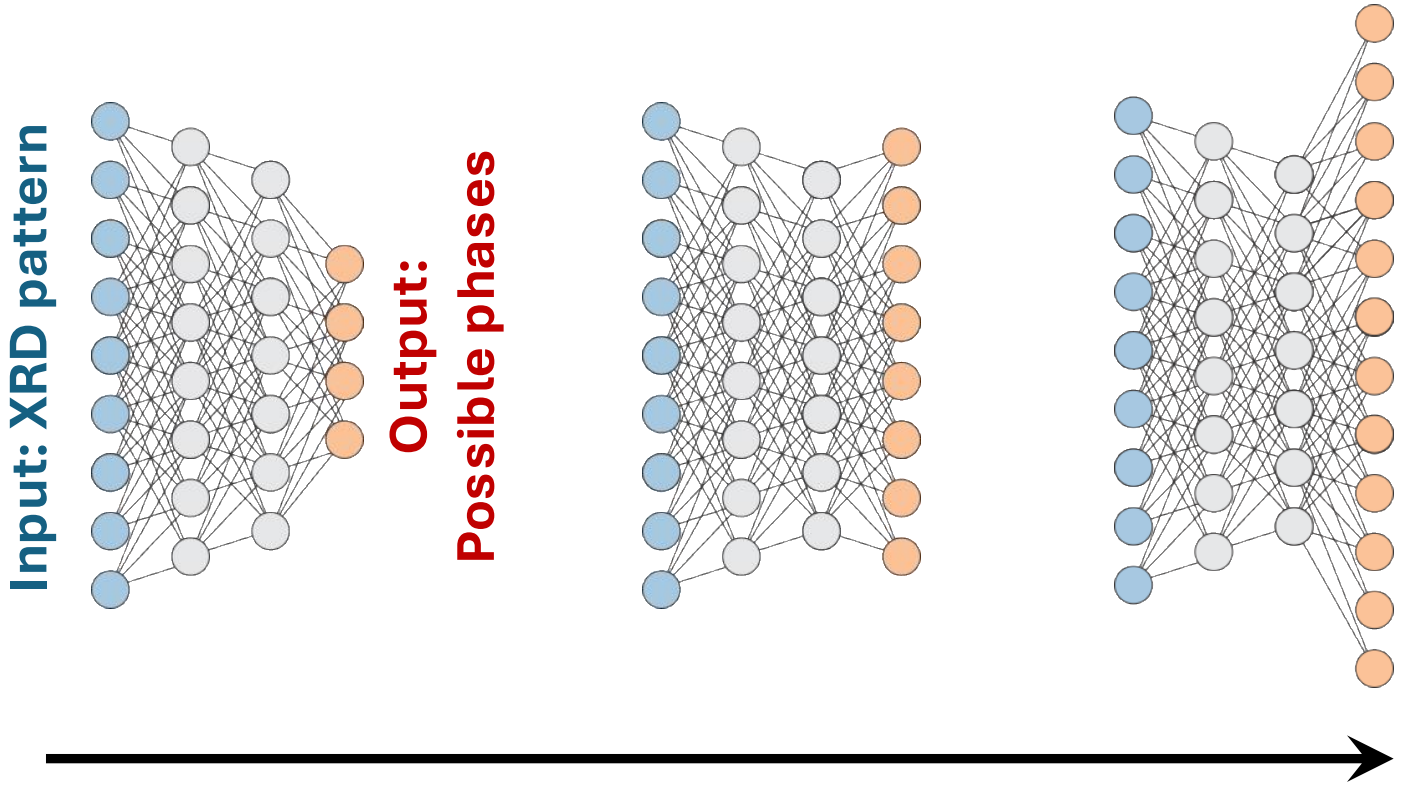
$$\begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_N \end{pmatrix}$$

Background subtraction,
peak detection, *etc.*

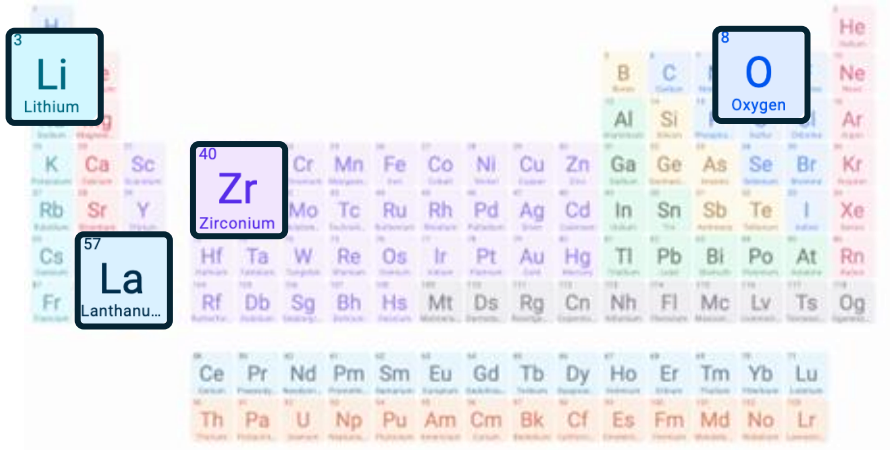


Phase(s)

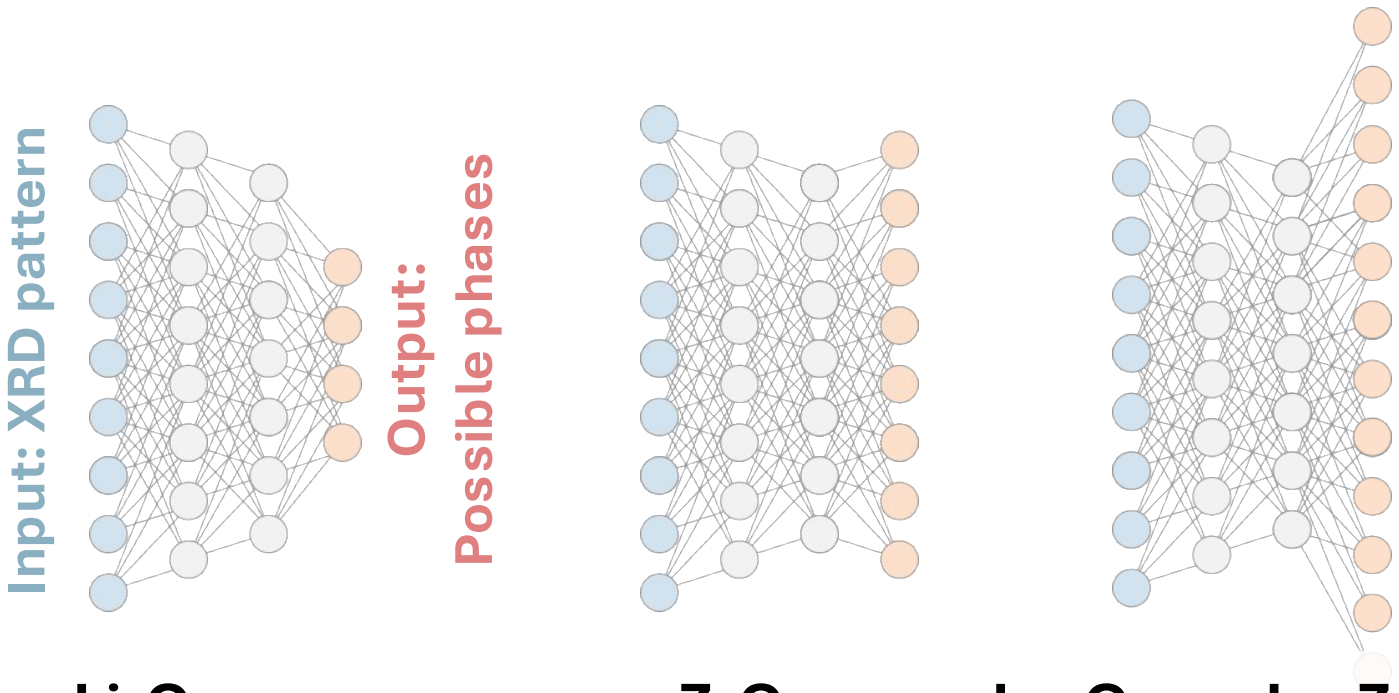
Another challenge: too many choices



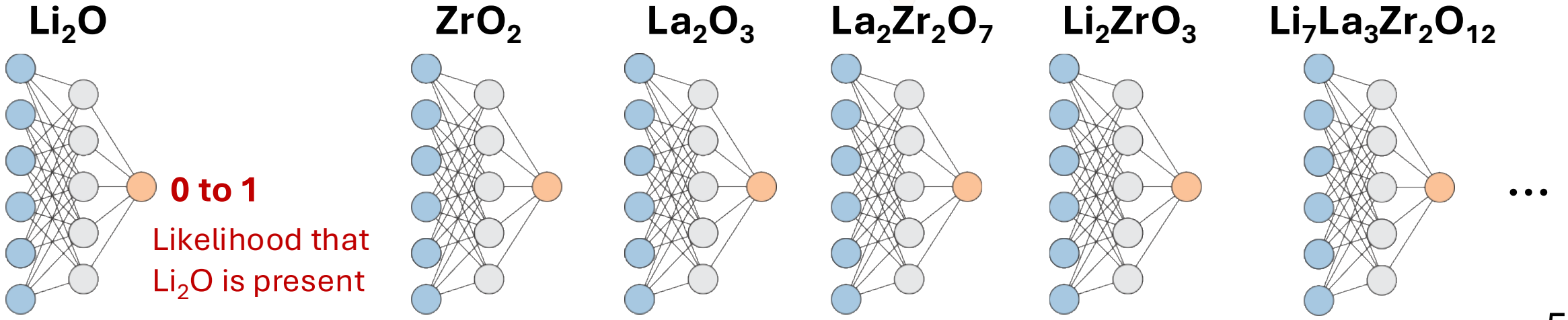
- ✗ Giving the model too many choices leads to confusion
- ✗ More phases = combinatorial explosion of training data



Solution: train a mixture of experts



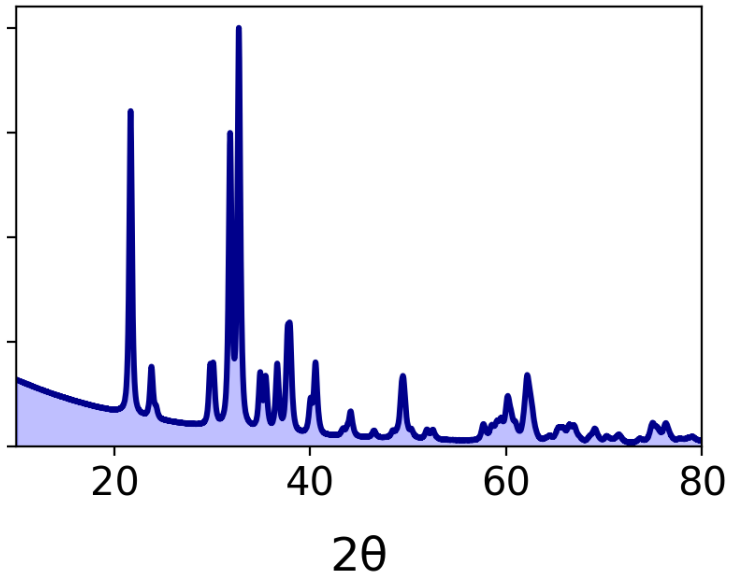
- ✗ Giving the model too many choices leads to confusion
- ✗ More phases = combinatorial explosion of training data

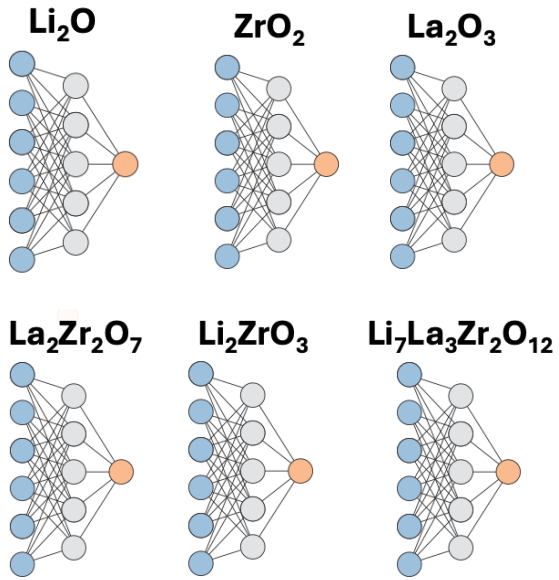


0 to 1

Likelihood that Li_2O is present

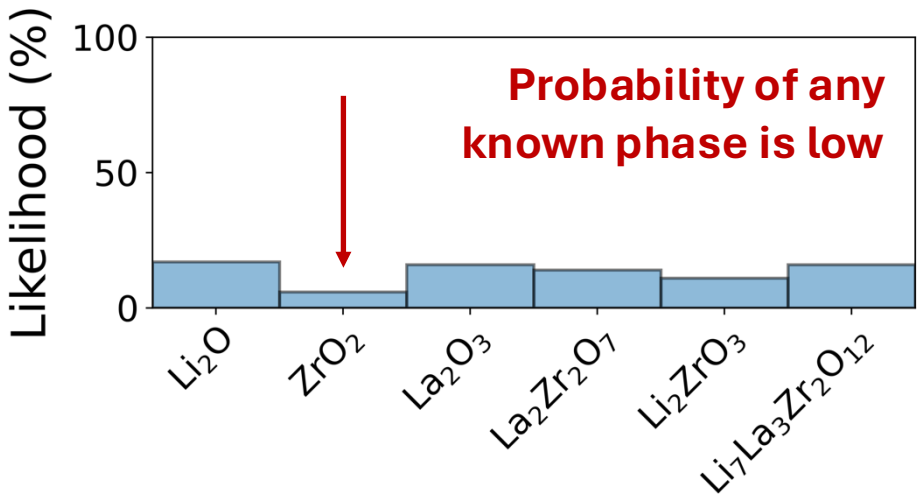
What about *new* phases, not in the dataset?





We may encounter phases that are *not* in our training set.

Can ML help **propose a structure**?



Generative AI can propose new structures

Generative AI *creates new content* by learning trends from existing data

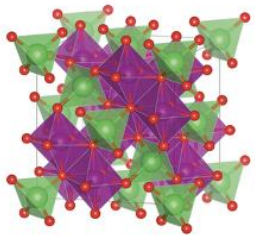
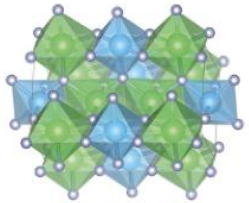
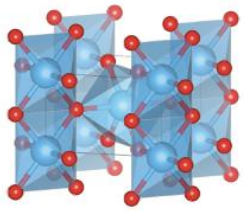
ChatGPT

deepseek

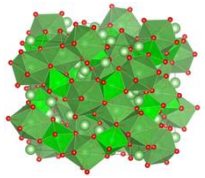
Claude

Gemini

This content may be text…

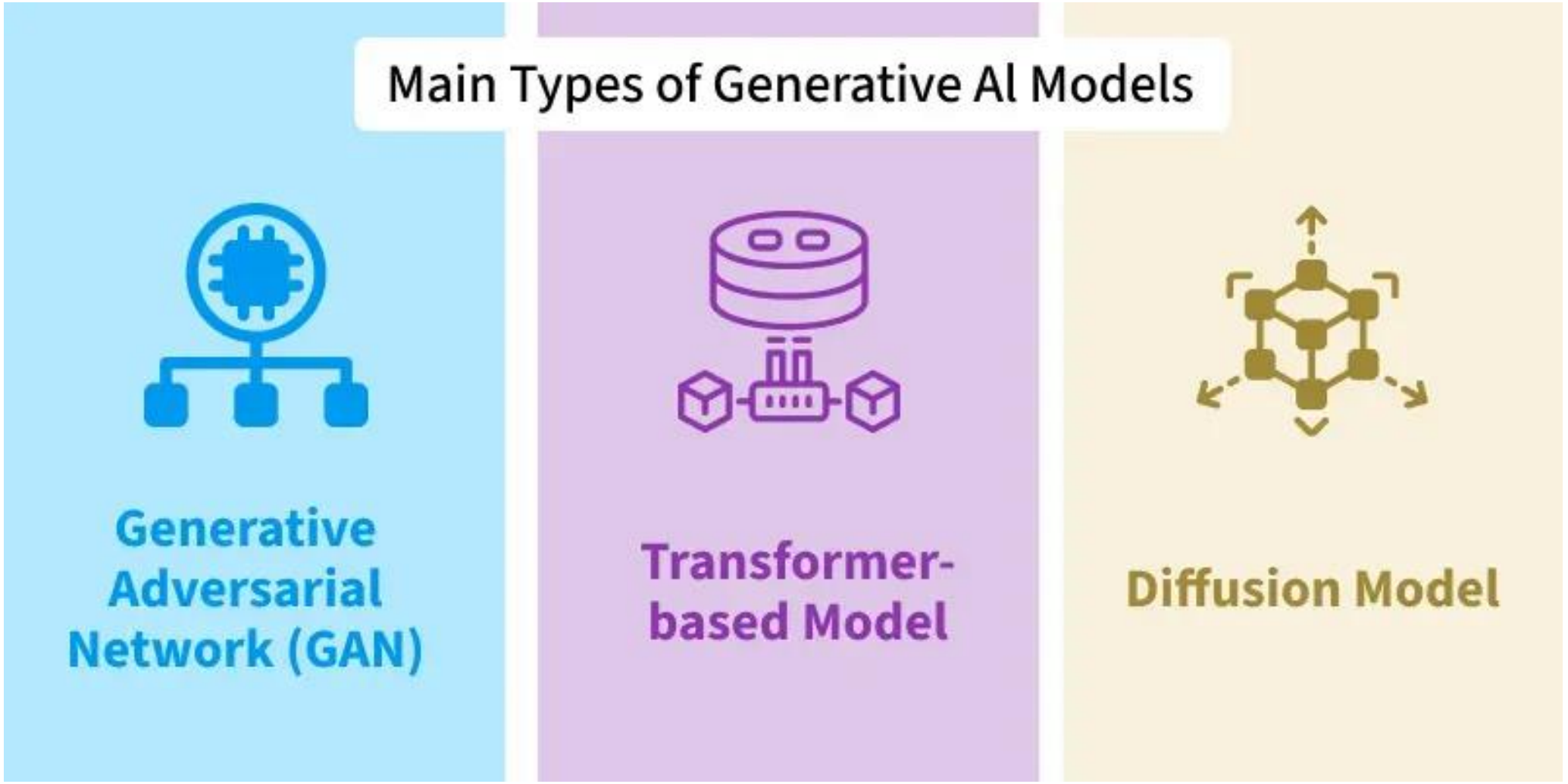
…Or structures!






…Or images…

Types of generative AI models



Types of generative AI models

Main Types of Generative AI M

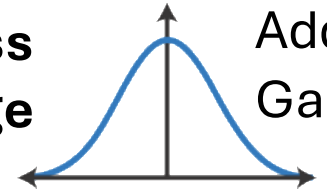
Generative Adversarial Network (GAN)

Transformer-based Model

Diffusion Model

Diffusion models for image generation

Forward process
"noises" a clean image



Adding a small amount of Gaussian noise $\hat{\varepsilon}_{t_i}$ at each step

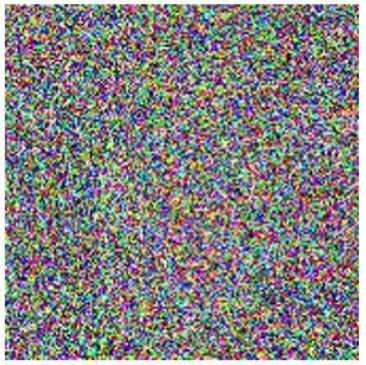
Clean image

$\hat{\varepsilon}_{t_1}$

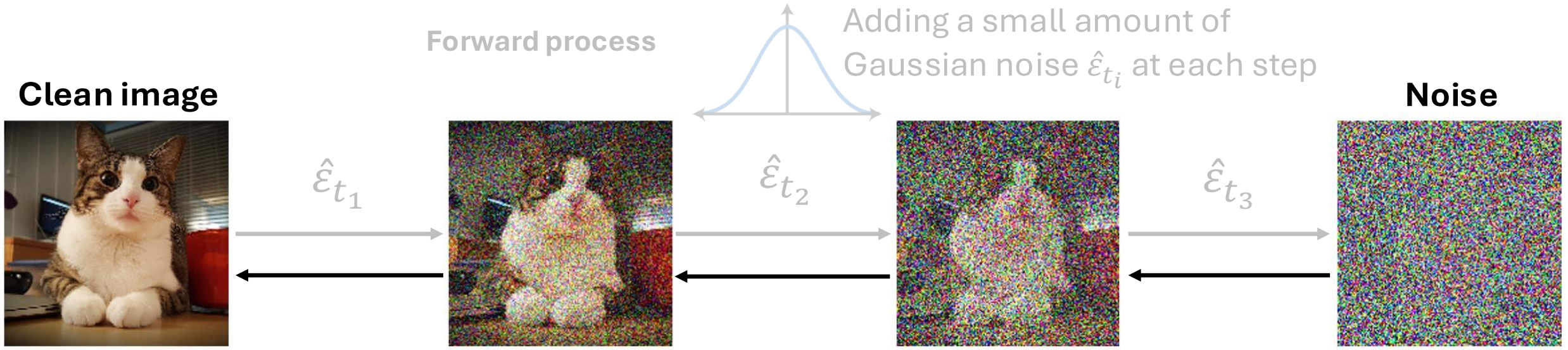
$\hat{\varepsilon}_{t_2}$

$\hat{\varepsilon}_{t_3}$

Noise

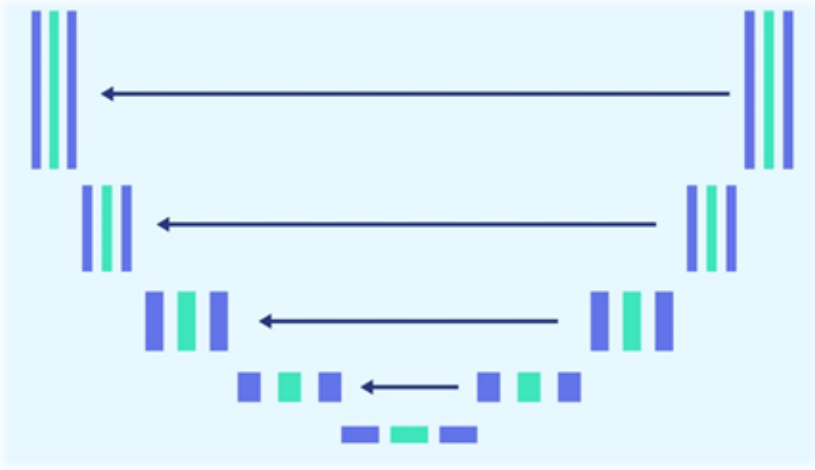


Diffusion models for image generation



Output:
Noise $\hat{\varepsilon}_{t_i}$ added to create this image

This can be subtracted to create a slightly more denoised image



U-Net architecture

Input:
Noisy image

Reverse process creates a clean image from noise

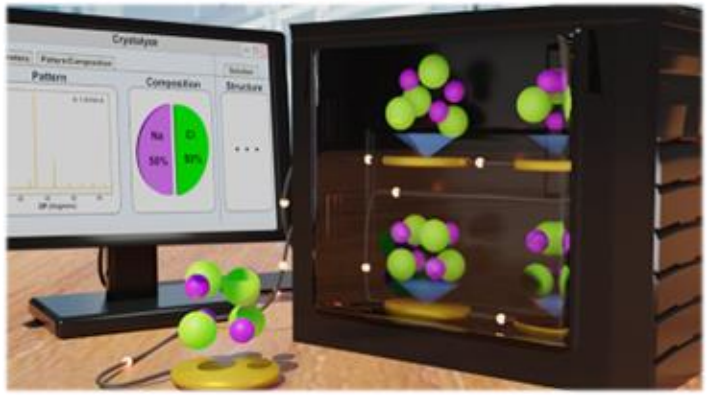
It’s called *diffusion* model for a reason

Diffusion models: XRD → Crystal structure

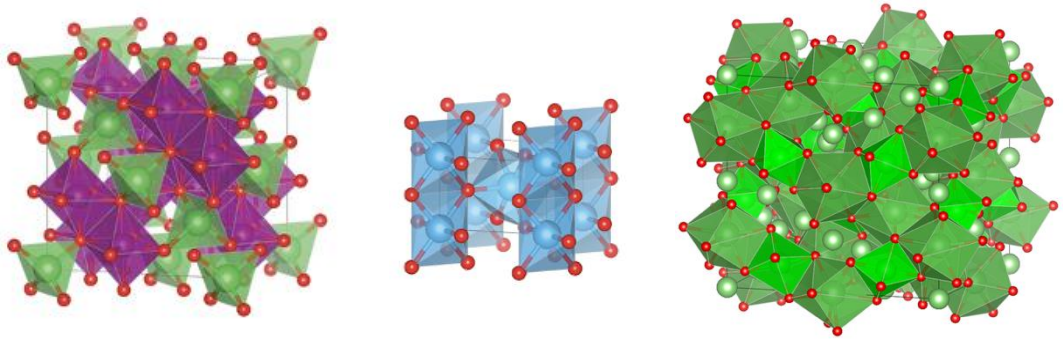
JACS
JOURNAL OF THE AMERICAN CHEMICAL SOCIETY
pubs.acs.org/JACS
Article

Crystal Structure Determination from Powder Diffraction Patterns with Generative Machine Learning

Eric A. Riesel,‖ Tsach Mackey,‖ Hamed Nilforoshan, Minkai Xu, Catherine K. Badding, Alison B. Altman, Jure Leskovec,* and Danna E. Freedman*

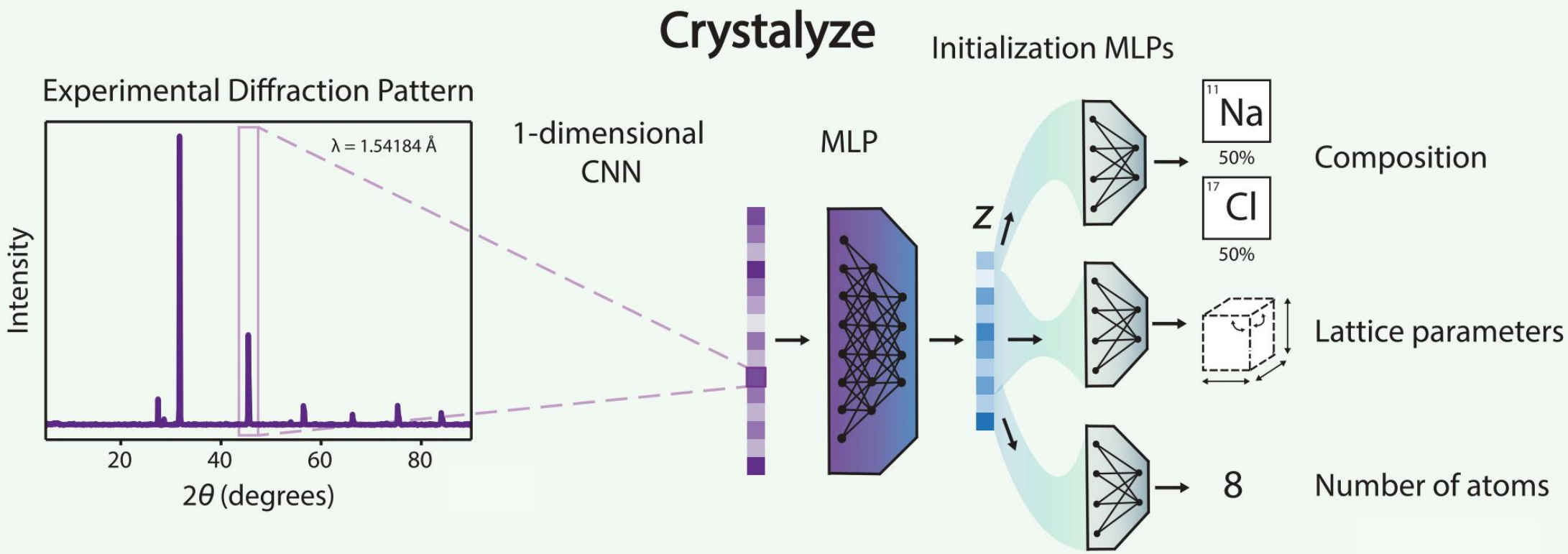


For images, we can fix the dimensions and number of pixels

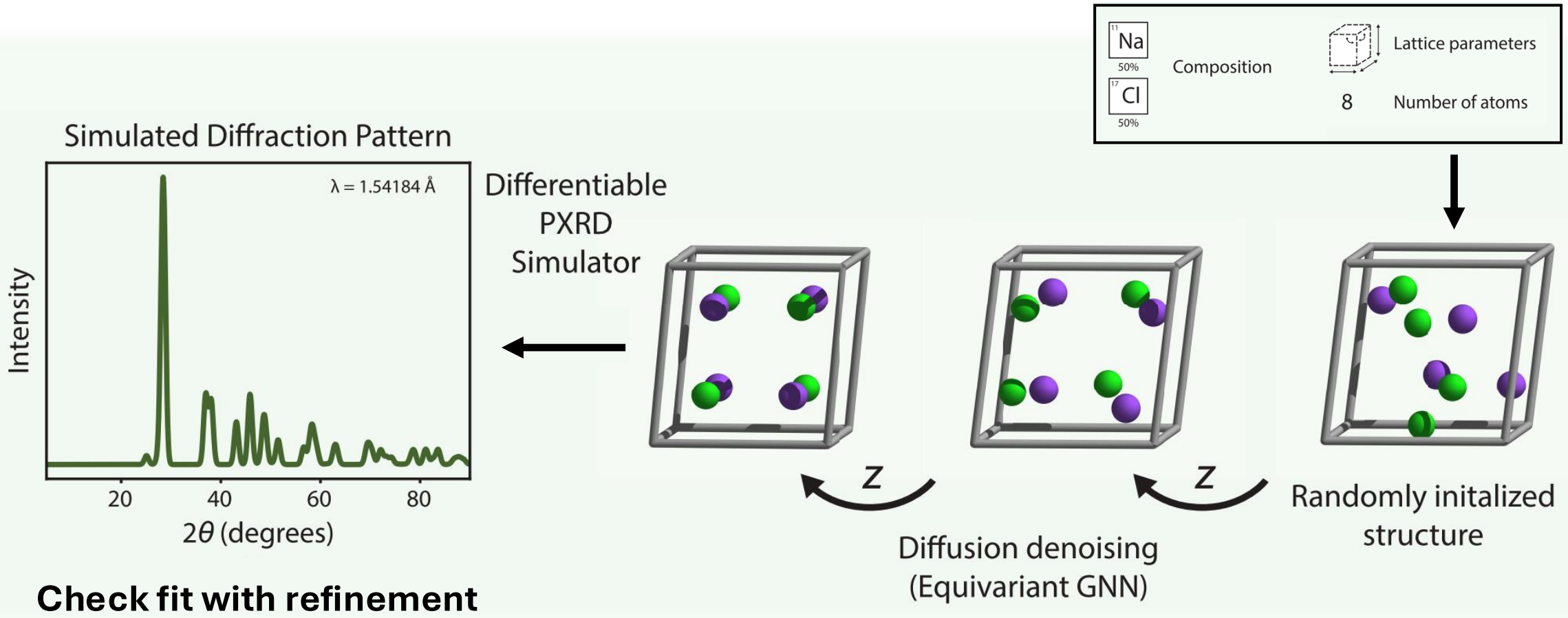


But for crystal structures, the lattice parameters and number of atoms per cell will vary…***Predict them***

Diffusion models: XRD → Crystal structure



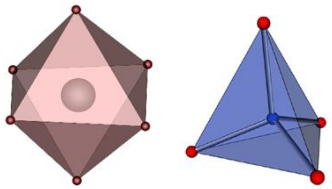
Diffusion models: XRD → Crystal structure



Assessing structure validity

Bond lengths

Coordination environments



pymatgen

Oxidation states

Cr	+1 +2 +3 +4 +5 +6
Mn	+1 +2 +3 +4 +5 +6 +7
Fe	+1 +2 +3 +4 +5 +6

Bond valence sum

$$V_i = \sum_j \exp\left[\left(R_{ij} - d_{ij}\right)/B\right]$$

"Instant" Materials Properties with Machine Learning Interatomic Potentials	Ong, Shyue Ping	United States	Tutorial Instructor	3:00 PM 4:00 PM

You can even assess thermodynamic stability using ***MLIPs***...Stay tuned!

MRS MATERIALS RESEARCH SOCIETY

Advancing materials. Improving the quality of life.

UCLA

Questions?

MD^2 LAB
MATERIALS DESIGN THROUGH DYNAMICS

Samueli School of Engineering