

# *2024 American Conference of Neutron Scattering*

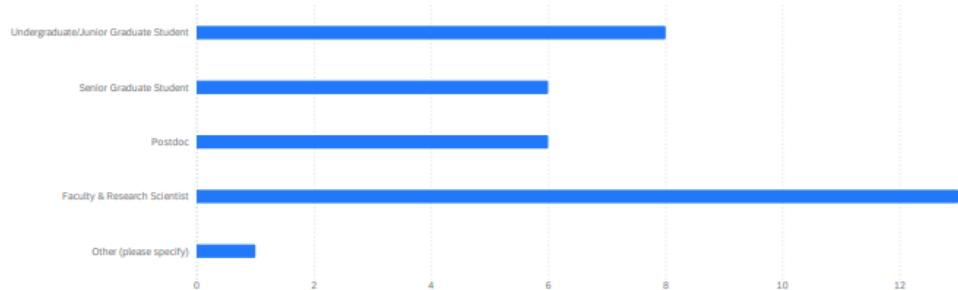
## *Artificial Intelligence for Scattering Tutorial*

### Introduction of Machine Learning for Scattering

**Mingda Li**  
mingda@mit.edu  
<http://qm.mit.edu/>

June 23, 2024

# Intended Learning Outcome



## Foundation

- Conceptual familiarity
- Literature reading
- Grant writing

## Application

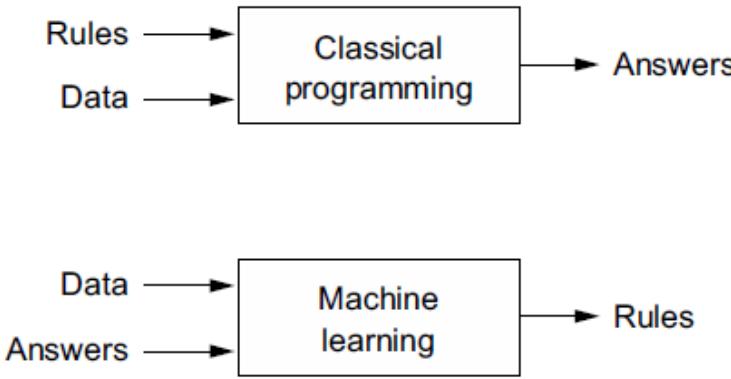
- Scattering data analysis
- Publication using AI
- Future AI-related jobs

# Today

- GT1: Machine learning crash course
- GT2: Materials Representations
- ST1: Symmetry in graph neural networks
- ST2: Neural differential equations
- ST3: Generative models
- Recent trends

- **GT1: Machine learning crash course**
- GT2: Materials Representations
- ST1: Symmetry in graph neural networks
- ST2: Neural differential equations
- ST3: Generative models
- Recent trends

# Machine learning



Neural networks ~ Deep learning  $\subset$  Machine learning  $\subset$  AI

## Analytical

$$f(\underbrace{x}_{\text{input}}; \underbrace{\theta}_{\text{learnable}}) = \underbrace{y}_{\text{output}}$$

$$Cost = \text{mean}((y - y_{true})^2)$$

$$\theta \leftarrow \theta - \underbrace{\eta}_{\text{Learning rate}} \times \nabla_{\theta} Cost$$

## Probabilistic

$$\hat{\theta} = \arg \max_{\theta} \{ p(y|x, \theta) \}$$

**Estimation:** accuracy of the predicted values of  $\hat{\theta}$

**Prediction:** accuracy of the prob.  
when inputting new data

# A highly incomplete kaleidoscope of ML models

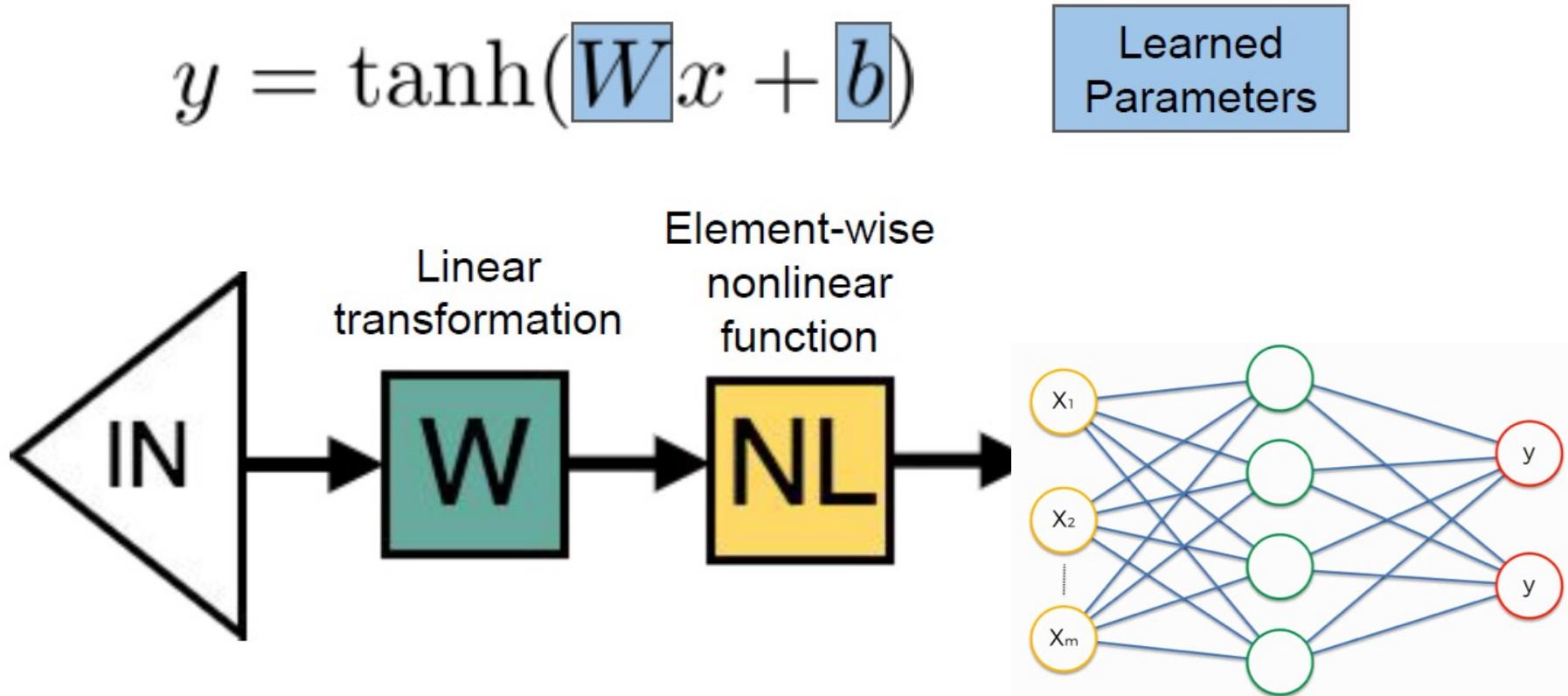
- Parametric: making implicit assumption of data, complexity NOT grow w/ data
- Non-parametric: fewer assumption of data, complexity grows w/ data
- Non-parametric does NOT mean no-parameter in model

	Supervised learning	Unsupervised Learning	Reinforcement Learning
Goals	Classification, regression, uncertainty quantification, etc.	Clustering, dimensionality reduction, denoising, anomaly detection, representation learning, etc.	Strategy learning under dynamical environment, sequential decision making and optimization, etc.
Parametric models	Linear regression; Logistic regression; SVM; NNs: FFNN, CNN, RNN (LSTM, GRU, ...)	Autoencoder; GMM; GAN; Diffusion model	Policy gradient learning; DQNs; Derivative free policy optimization
Non-parametric models	Decision trees; Gaussian process KNN; Kernel-SVM; Ensemble: Random forest, XGBoost	K-means clustering; PCA; Kernel PCA; t-SNE; DBSCAN	Tabular policy iteration; Q-learning

# What is a neural network?

A function with “learnable” parameters

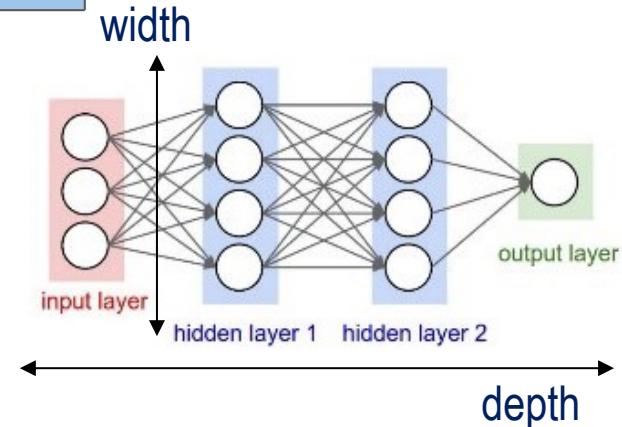
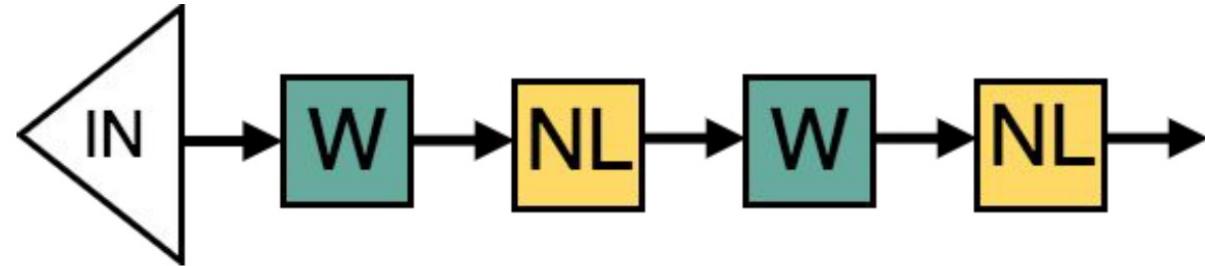
e.g. fully-connected neural network with output: “supervised”



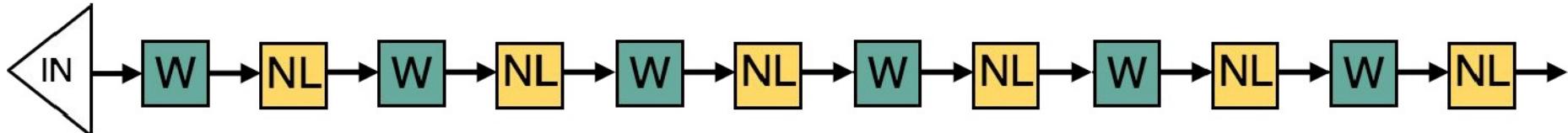
# What is deep learning?

$$y = \tanh(W_2 \tanh(W_1 x + b_1) + b_2)$$

Learned Parameters



- Neural networks with multiple layers can learn more complicated functions.
- Nonlinearity is key for representing the complexity.
- Deep learning with so many deep “hidden” layers in the middle



# Is it just fancy way of fitting?

- **No** (fundamental): Universal Approximation Theorem

For any integrable function  $f: R^n \rightarrow R$ , and any small positive number  $\varepsilon$ , There exists a fully connected neural network  $F_{NN}(x)$  with width  $w \leq n+4$ , satisfying

$$\int |f(x) - F_{NN}(x)| d^n x < \varepsilon$$

- A neural network can represent any function in principle.
- Manual fitting in x-ray/neutron scattering: each time we need a separate fitting model.

# Is it just fancy way of fitting?

- **No** (fundamental): Global minimum in possible

## Deep Learning without Poor Local Minima

Kenji Kawaguchi

In this paper, we prove a conjecture published in 1989 and also partially address an open problem announced at the Conference on Learning Theory (COLT) 2015. With no unrealistic assumption, we first prove the following statements for the squared loss function of deep linear neural networks with any depth and any widths: 1) the function is non-convex and non-concave, 2) every local minimum is a global minimum, 3) every critical point that is not a global minimum is a saddle point, and 4) there exist "bad" saddle points (where the Hessian has no negative eigenvalue) for the deeper networks (with more than three layers), whereas there is no bad saddle point for the shallow networks (with three layers). Moreover, for deep nonlinear neural networks, we prove the same four statements via a reduction to a deep linear model under the independence assumption adopted from recent work. As a result, we present an instance, for which we can answer the following question: how difficult is it to directly train a deep model in theory? It is more difficult than the classical machine learning models (because of the non-convexity), but not too difficult (because of the nonexistence of poor local minima). Furthermore, the mathematically proven existence of bad saddle points for deeper models would suggest a possible open problem. We note that even though we have advanced the theoretical foundations of deep learning and non-convex optimization, there is still a gap between theory and practice.

- Even not global minimum, many local minima are not bad.
- Manual fitting in x-ray/neutron scattering: we may run into local minimum that does not represent physical reality.

# Is it just fancy way of fitting?

- Not quite (practical): Favored at high data volume, high-dimensional data

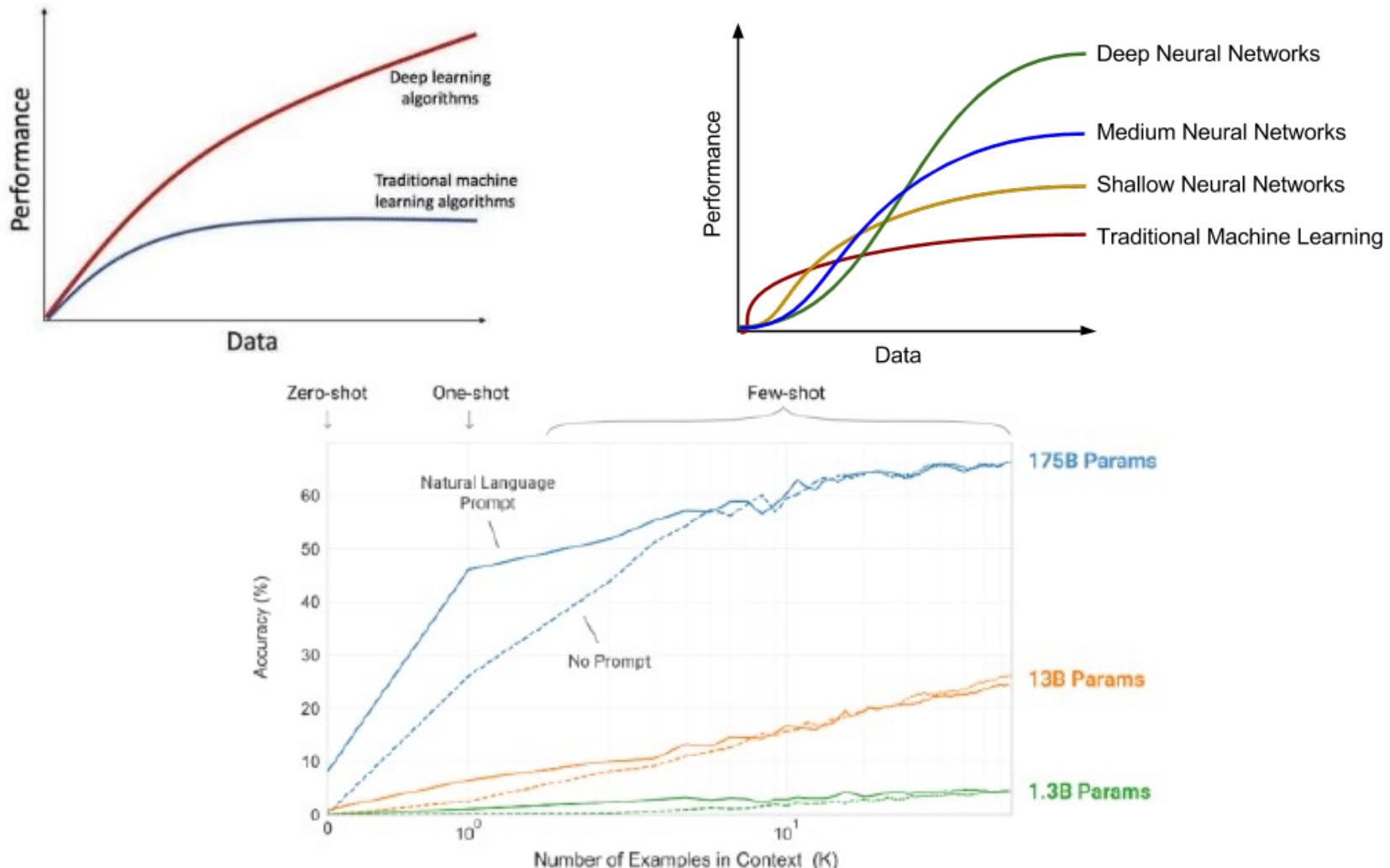
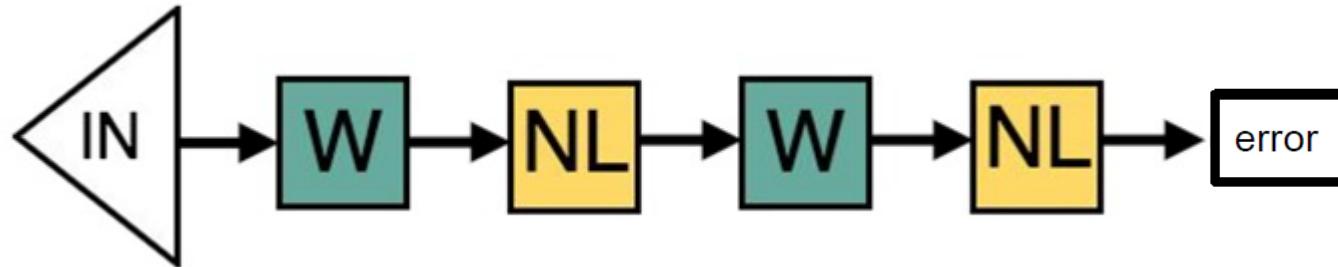


Figure taken from Language models are few-shot learners Advances in neural information processing systems 2020, 33, 1877

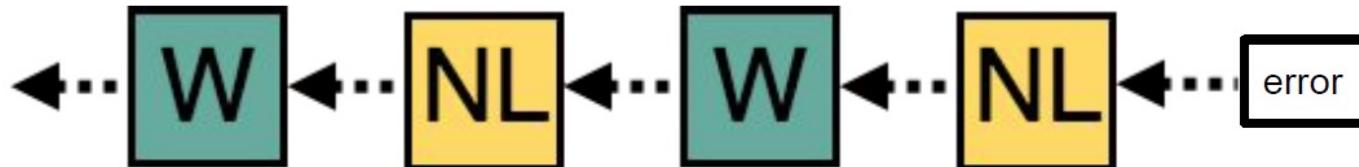
# How does the “learning” work?

- **Back propagation:** General way to update the parameters  $\mathbf{W}$  if the neural network is differentiable.
- Forward calculation using existing parameters  $\mathbf{W}$  to obtain the “error”, difference between “ground truth”  $y$  and predicted  $f(\mathbf{W}, \mathbf{x})$



- Use error to guide the update of the parameters

$$\Delta W_{ij} = -\eta \frac{\partial \text{error}(f(\mathbf{W}, \mathbf{x}), y)}{\partial W_{ij}}$$

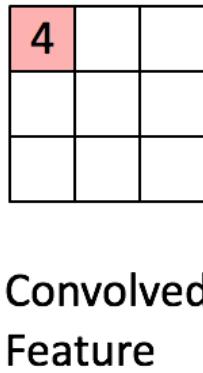


# Convolutional neural network

- Original data convolves with a **Kernel** to get a new layer of **reduced data**

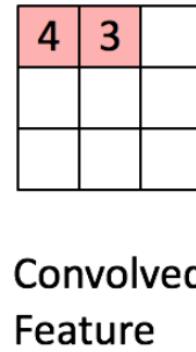
1 x1	1 x0	1 x1	0	0
0 x0	1 x1	1 x0	1	0
0 x1	0 x0	1 x1	1	1
0	0	1	1	0
0	1	1	0	0

Image



1	1 x1	1 x0	0 x1	0
0	1 x0	1 x1	1 x0	0
0	0 x1	1 x0	1 x1	1
0	0	1	1	0
0	1	1	0	0

Image



1	1	1 x1	0 x0	0 x1
0	1	1 x0	1 x1	0 x0
0	0	1 x1	1 x0	1 x1
0	0	1	1	0
0	1	1	0	0

Image

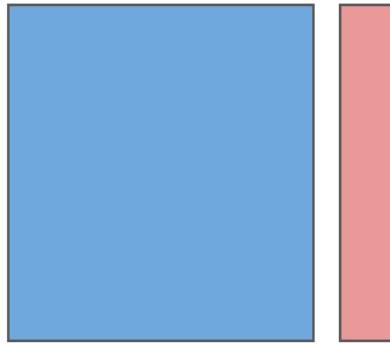
4	3	4

Convolved Feature

- Reduce the data, i.e., to lower dimension for easy processing and prediction, but still keep key features representing spatial/temporal correlation
- X-ray and neutron scattering:
  - SAXS/SANS, changing real-space size leads to multiple q-spaces to change → q space is correlated.

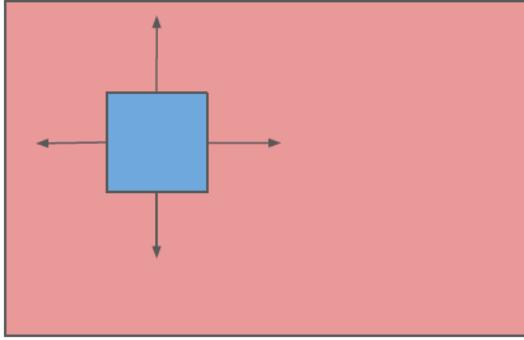
# Types of neural networks depend on data type

**Arrays  $\Rightarrow$  Dense NN**



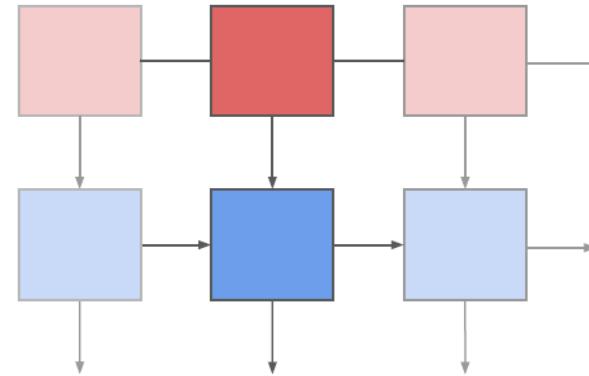
Components are independent.

**2D images  $\Rightarrow$  Convolutional NN**



The same features can be found anywhere in an image. Locality.

**Text  $\Rightarrow$  Recurrent NN**



Sequential data. Next input/output depends on input/output that has come before.

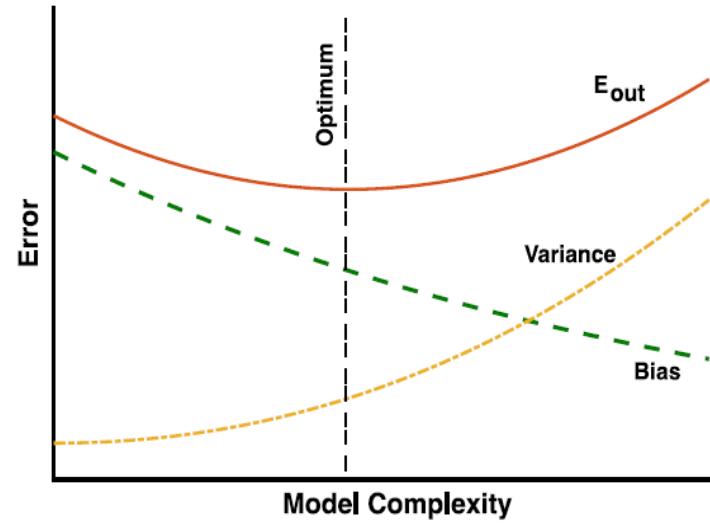
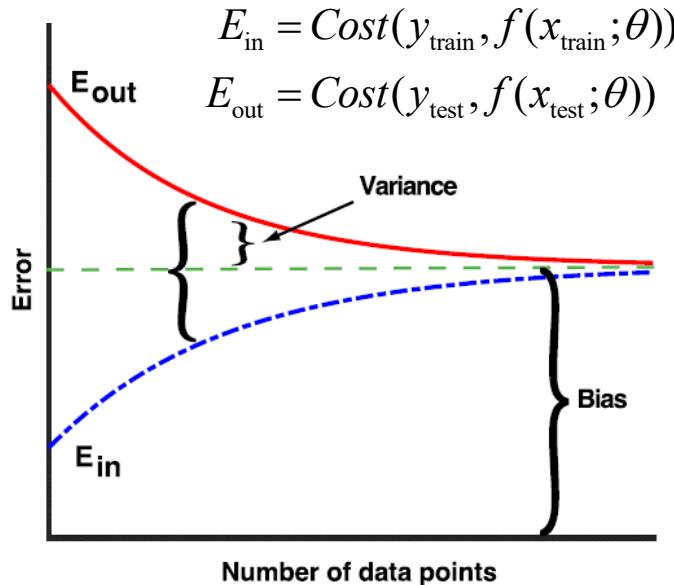
- X-ray and neutron scattering:

Generic fitting function

Correlative feature; 2D area detector, imaging

Time-resolved spectra, guiding optimized measurement plan

# Evaluation of machine learning model: Classical



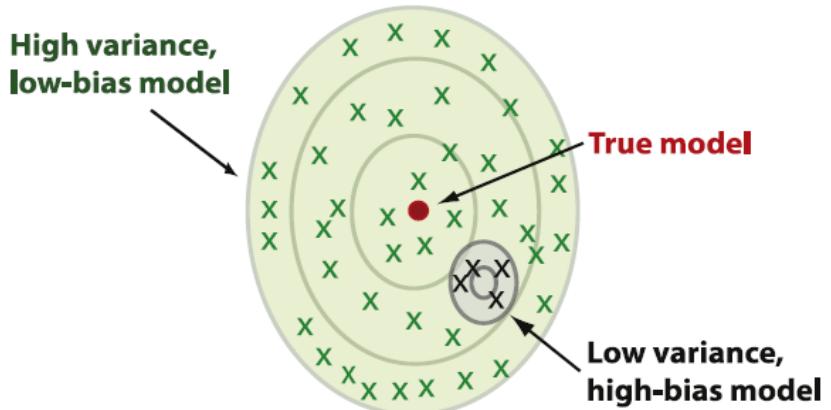
**Bias:** the best estimate of error with ideal  $\propto$  data

**Variance:** The gap between test vs ideal.

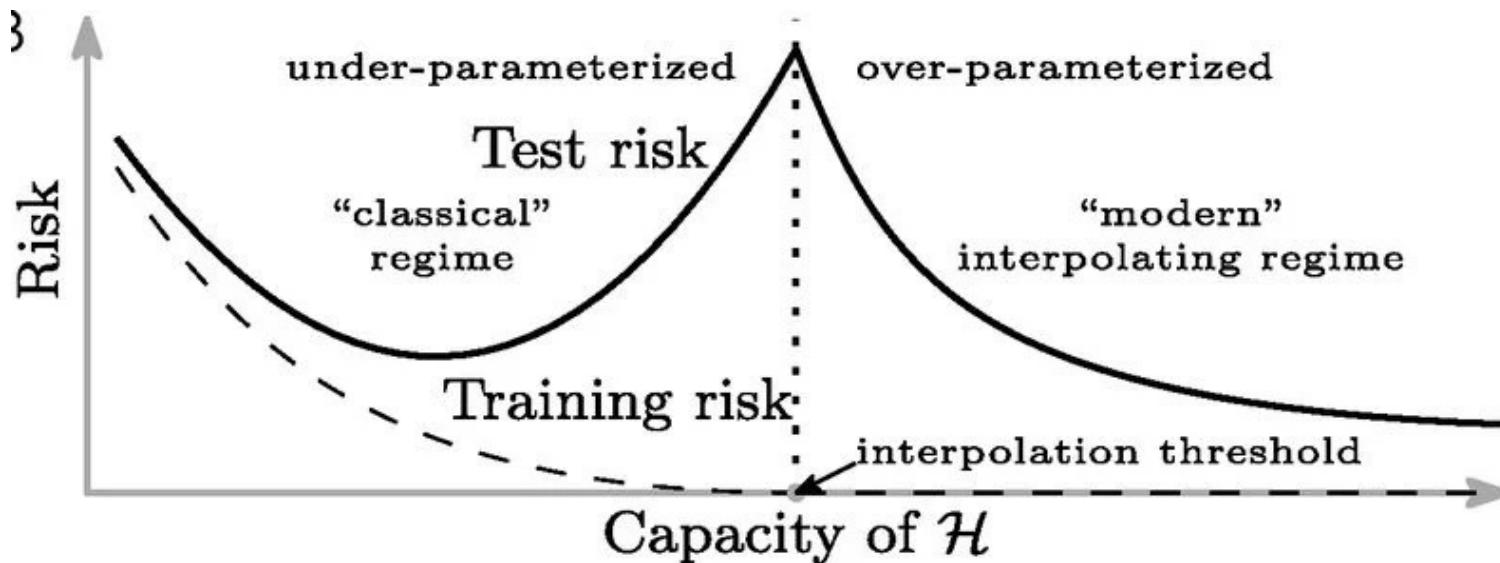
**Overfitting:** Low-bias, high-variance

**Underfitting:** High-bias, low-variance.

**Q: Can a model be neither under-fitted nor over-fitted?**



# Evaluation of machine learning model: Modern



- Classical picture with increased Test-set Error w.r.t model capacity
- When model increases beyond threshold, test-error decreases

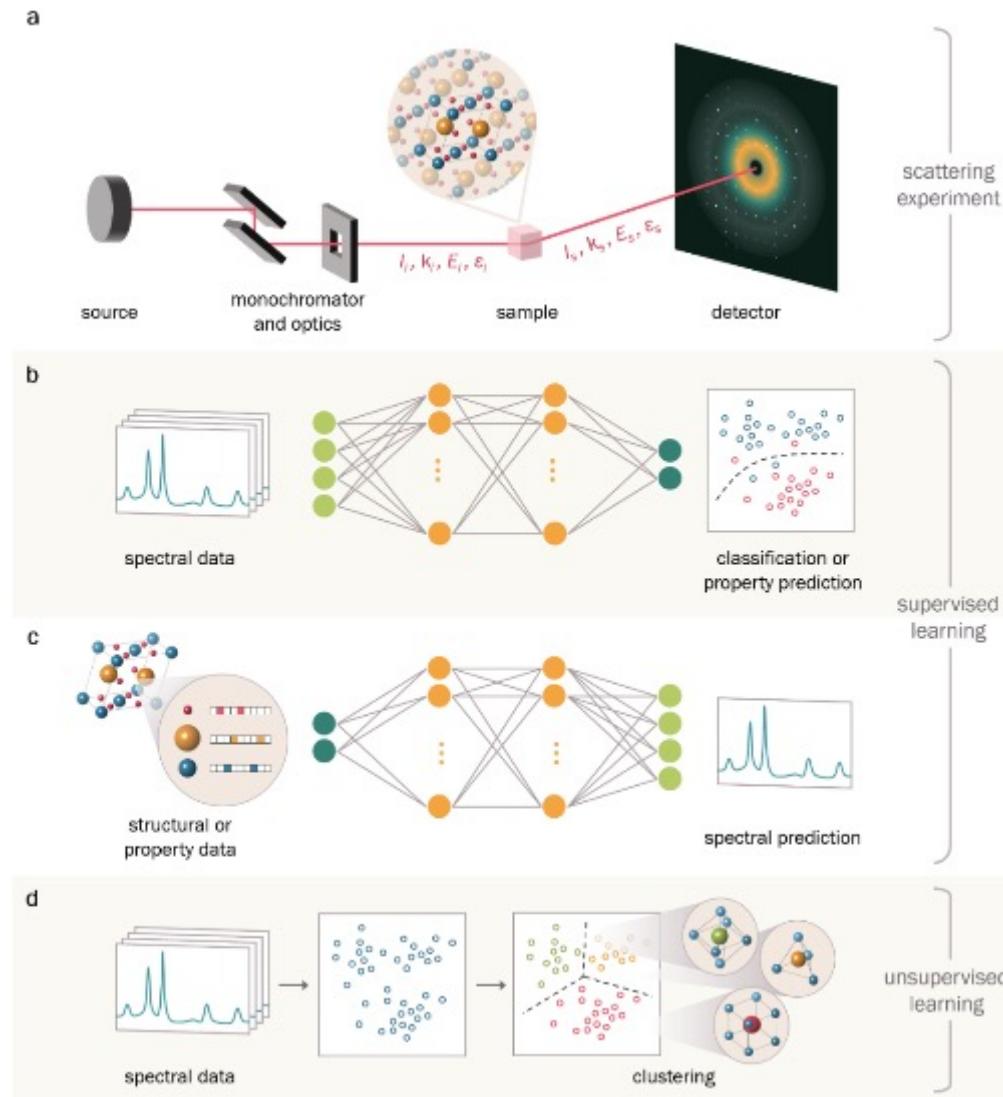
**Reconciling modern machine-learning  
practice and the classical bias-variance  
trade-off**

Mikhail Belkin , Daniel Hsu, Siyuan Ma, and Soumik Mandal [Authors Info & Affiliations](#)

Edited by Peter J. Bickel, University of California, Berkeley, CA, and approved July 2, 2019 (received for review February 21, 2019)

July 24, 2019 | 116 (32) 15849-15854 | <https://doi.org/10.1073/pnas.1903070116>

# Machine learning for scattering



"Machine learning in neutron and x-ray scattering and spectroscopies." Chem. Phys. Rev. **2**, 031301 (2021).

- GT1: Machine learning crash course
- **GT2: Materials Representations**
- ST1: Symmetry in graph neural networks
- ST2: Neural differential equations
- ST3: Generative models
- Recent trends

# Physics-Inspired Structural Representations for Molecules and Materials

Felix Musil, Andrea Grisafi, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti\*

Cite this: *Chem. Rev.* 2021, 121, 16, 9759–9815

Publication Date: July 26, 2021

<https://doi.org/10.1021/acs.chemrev.1c00021>

Copyright © 2021 The Authors. Published by American Chemical Society. This publication is licensed under CC-BY 4.0.

Perspective | [Open access](#) | Published: 16 March 2022

Article Views

31130

Altmetric

47

Citations

165

[LEARN ABOUT THESE METRICS](#)

## Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning

Marcel F. Langer, Alex Goëßmann & Matthias Rupp 

*njp Computational Materials* 8, Article number: 41 (2022) | [Cite this article](#)

Advanced Review |  [Open Access](#) |  

## A review of molecular representation in the age of machine learning

Daniel S. Wigh, Jonathan M. Goodman, Alexei A. Lapkin 

First published: 18 February 2022 | <https://doi.org/10.1002/wcms.1603> | Citations: 35

**ANNUAL REVIEW OF MATERIALS RESEARCH** Volume 53, 2023

Review Article | Open Access

## Representations of Materials for Machine Learning

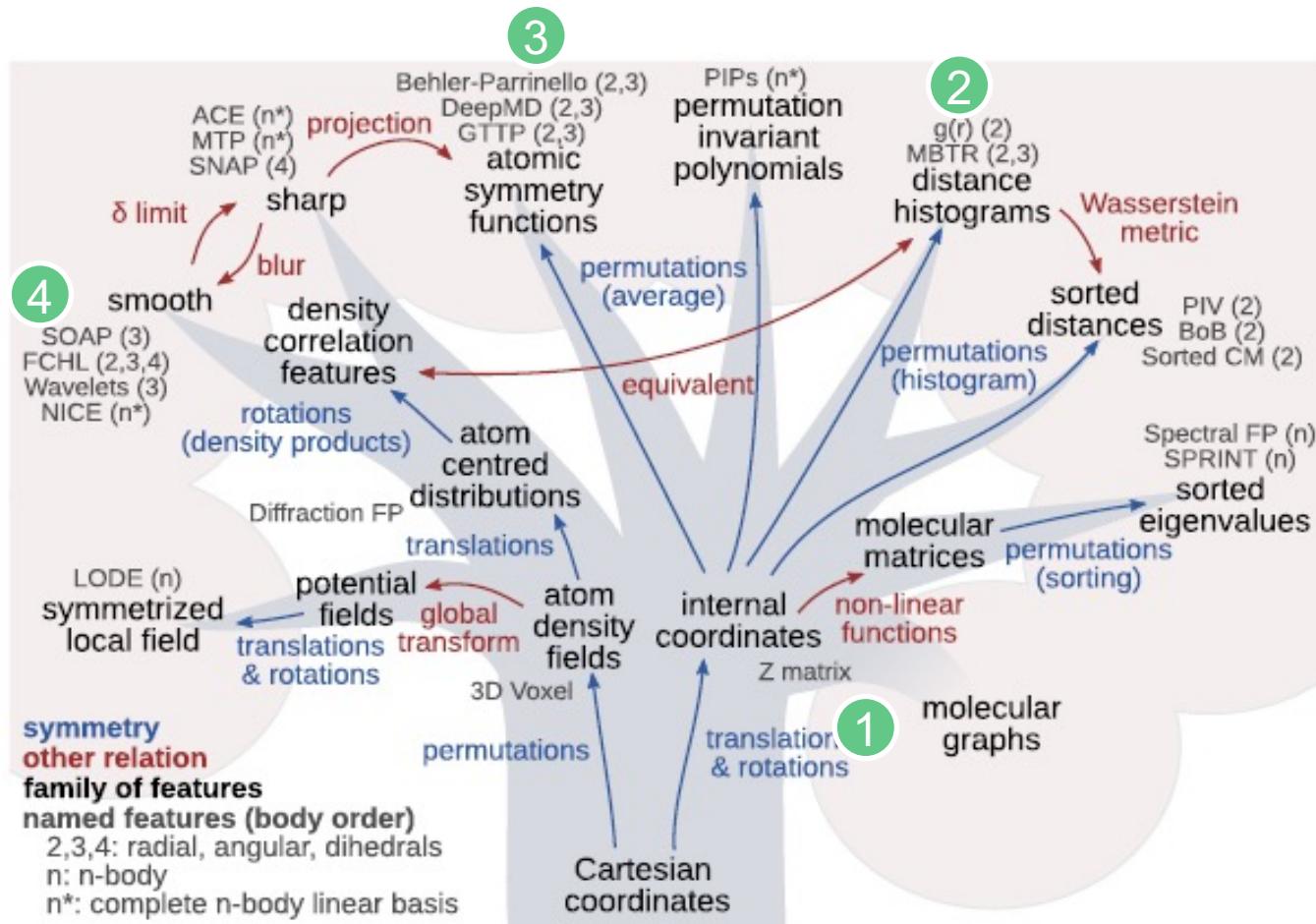
James Damewood<sup>1</sup>, Jessica Karaguesian<sup>1,2</sup>, Jaclyn R. Lunger<sup>1</sup>, Aik Rui Tan<sup>1</sup>, Mingrou Xie<sup>1,3</sup>, Jiayu Peng<sup>1</sup>, and Rafael Gómez-

Bombarelli<sup>1</sup>

 Group

Another pillar for successful ML but sometimes less emphasized than model:  
Representations

# Common Materials Representations



**Figure 2.** Phylogenetic tree of structural representations for materials and molecules. Arrows indicate the relationship between different groups of features. Lists of names, in gray, indicate the most common implementations for each class. Classes that appear as “leaves” of the tree are fully symmetric.

# 1 Z-matrix (Coulomb matrix) Representation

- For a molecule with  $N$  atoms located at  $\{\mathbf{R}_I\}$  and  $\{Z_I\}$

$$M_{IJ} = \begin{cases} 0.5Z_I^{2.4} & \text{for } I = J, \\ \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} & \text{for } I \neq J. \end{cases} \quad I, J = 1, 2, \dots, N$$

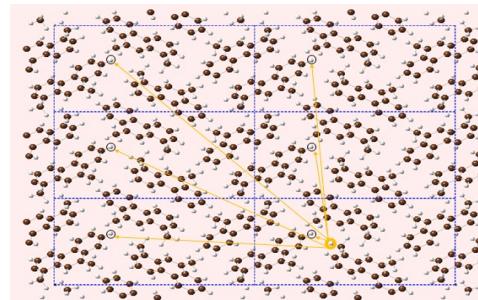
- With Eigenvalues  $\varepsilon_I$ , distance between two molecules

$$d(\mathbf{M}, \mathbf{M}') = \sqrt{\sum_{I=1}^N (\varepsilon_I - \varepsilon'_I)^2}$$

- Energy of a new molecule from training set  $\{\mathbf{M}_i\}$

$$E^{\text{est}}(\mathbf{M}) = \sum_{i=1}^N \alpha_i \exp\left[-\frac{1}{2\sigma^2} d(\mathbf{M}, \mathbf{M}_i)^2\right],$$

- Generalizable to infinite solids

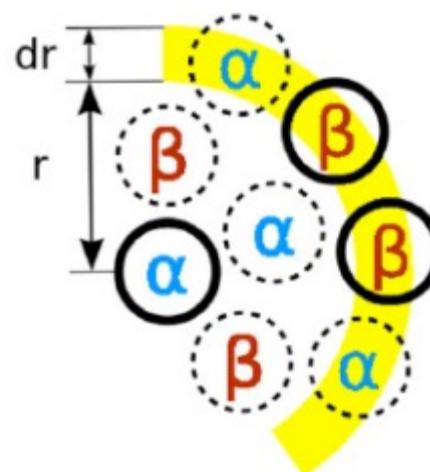
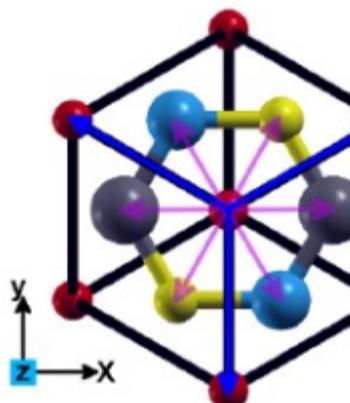


## 2 $g(r)$ Partial radial distribution function Representation

- Radial distribution for two atomic species

$$g_{\alpha\beta}(r) = \frac{1}{N_\alpha V_r} \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} \theta(d_{\alpha_i\beta_j} - r)\theta(r + dr - d_{\alpha_i\beta_j}),$$

- Counting the density of atoms in a thin shell away from center



### ③ Behler-Parrinello (Atomic-center-pairwise) representation

- Consideration both radial and angular information

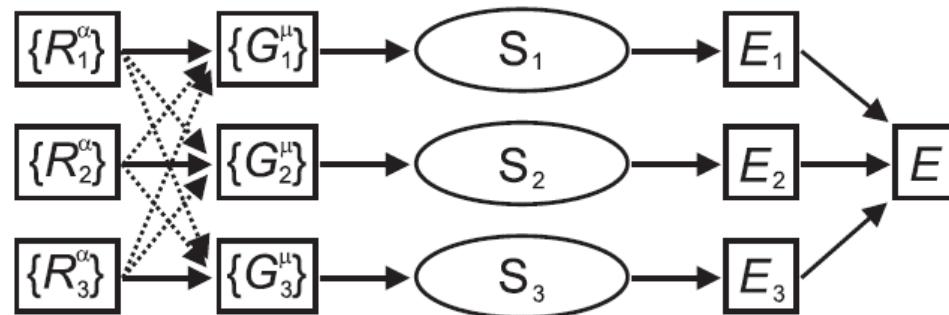
- Radial function:

$$G_i^1 = \sum_{j \neq i}^{\text{all}} e^{-\eta(R_{ij} - R_s)^2} f_c(R_{ij}). \quad f_c(r) = \begin{cases} [\cos(\frac{\pi r}{r_{\text{cut}}}) + 1]/2 & \text{for } r \leq r_{\text{cut}}, \\ 0 & \text{for } r > r_{\text{cut}}. \end{cases}$$

- Angular function:

$$G_i^2 = 2^{1-\zeta} \sum_{j,k \neq i}^{\text{all}} (1 + \lambda \cos \theta_{ijk})^\zeta \times e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk})$$

- Example on 3-atom solid:



## SOAP (Smooth Overlap of Atomic Positions) Representation

- Invariant local environment representation of atoms
- Atomic density (smoothed):  $\rho(\mathbf{r}) = \sum_i \exp(-\alpha|\mathbf{r} - \mathbf{r}_i|^2)$
- Basis function expansion:  $\rho(\mathbf{r}) = \sum_n \sum_{l=0}^l \sum_{m=-l}^l c_{nlm} g_n(r) Y_{lm}(\hat{\mathbf{r}}).$
- Power spectrum:  $p_{nn'l} \equiv \sum_m c_{nlm} (c_{n'lm})^*$
- Atomic environments as density overlap:  $S(\rho, \rho') = \int \rho(\mathbf{r}) \rho'(\mathbf{r}) d\mathbf{r}.$
- Similarity kernel:

$$k(\rho, \rho') = \int |S(\rho, \hat{R}\rho')|^n d\hat{R}$$

$$= \int d\hat{R} \left| \int \rho(\mathbf{r}) \rho'(\hat{R}\mathbf{r}) d\mathbf{r} \right|^n$$

$$k(\rho, \rho') = \sum_{n,n',l,m,m'} c_{nlm} (c'_{nlm'})^* (c_{nlm})^* c'_{n'lm'}$$

$$\equiv \sum_{n,n',l} p_{nn'l} p'_{nn'l}$$

# Recent MLIPs

[Submitted on 18 Dec 2015 (v1), last revised 23 Jul 2016 (this version, v2)]

## Moment Tensor Potentials: a class of systematically improvable interatomic potentials

Alexander V. Shapeev

Density functional theory offers a very accurate way of computing materials properties from first principles. However, it is too expensive for modelling large-scale molecular systems whose properties are, in contrast, computed using interatomic potentials. The present paper considers, from a mathematical point of view, the problem of constructing interatomic potentials that approximate a given quantum-mechanical interaction model. In particular, a new class of systematically improvable potentials is proposed, analyzed, and tested on an existing quantum-mechanical database.

Article | Published: 28 November 2022

## A universal graph deep learning interatomic potential for the periodic table

Chi Chen & Shyue Ping Ong

Nature Computational Science 2, 718–728 (2022) | Cite this article

5215 Accesses | 55 Citations | 187 Altmetric | Metrics

Element	M3GNet	M3GNet-all	EAM	MEAM	NNP	MTP
Energy ( $10^{-3}$ eV per atom)						
Ni	0.9	1.9	8.5	23.0	2.3	0.8
Cu	1.8	2.3	7.5	10.5	1.7	0.5
Li	2.5	4.7	368.6	—	1.0	0.7
Mo	6.3	6.8	68.0	36.4	5.7	3.9
Si	9.6	6.8	—	111.7	9.9	3.0
Ge	9.4	5.9	—	—	11.0	3.7
Force ( $10^{-3}$ eV Å $^{-1}$ )						
Ni	37.4	37.0	110	330	67.3	26.9
Cu	17.0	16.9	120	240	63.0	13.5
Li	22.1	24.5	140	—	63.4	13.2
Mo	193.7	271.4	520	220	198.7	148.1
Si	102.8	126.2	—	400	174.2	88.1
Ge	76.4	78.4	—	—	124.3	70.3

Article | Open access | Published: 04 May 2022

## E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt & Boris Kozinsky

Nature Communications 13, Article number: 2453 (2022) | Cite this article

51k Accesses | 238 Citations | 85 Altmetric | Metrics

Article | Open access | Published: 14 September 2023

## CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling

Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J. Bartel & Gerbrand Ceder

Nature Machine Intelligence 5, 1031–1041 (2023) | Cite this article

11k Accesses | 7 Citations | 36 Altmetric | Metrics

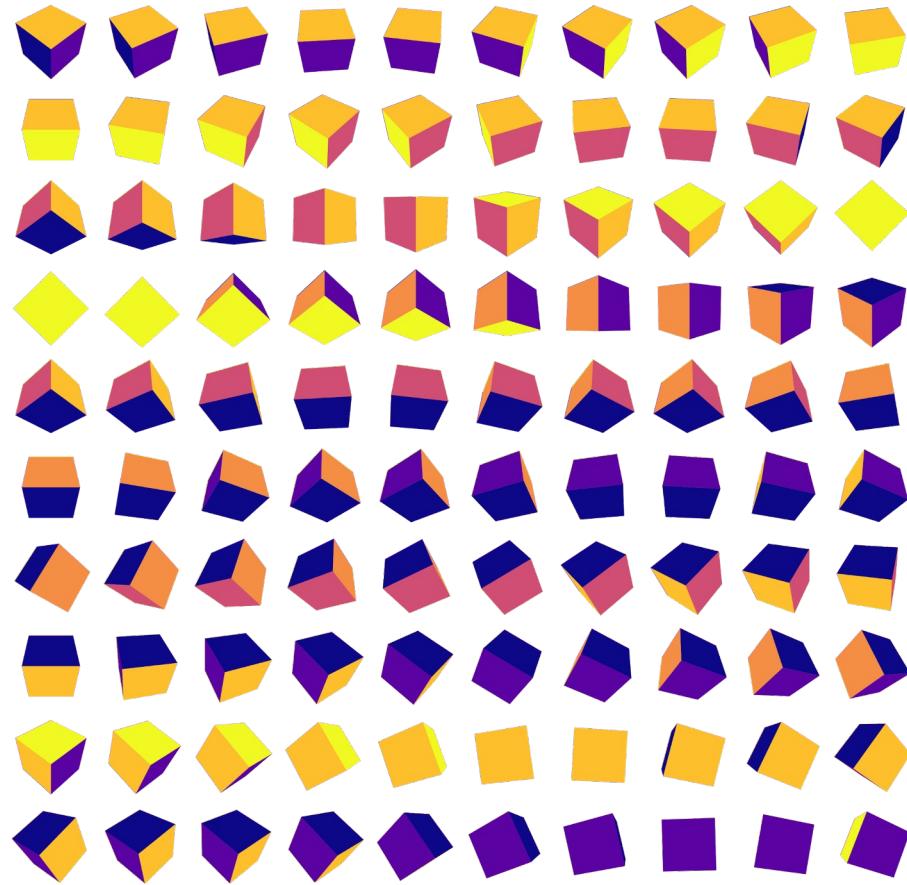
## MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, Gábor Csányi

Creating fast and accurate force fields is a long-standing challenge in computational chemistry and materials science. Recently, several equivariant message passing neural networks (MPNNs) have been shown to outperform models built using other approaches in terms of accuracy. However, most MPNNs suffer from high computational cost and poor scalability. We propose that these limitations arise because MPNNs only pass two-body messages leading to a direct relationship between the number of layers and the expressivity of the network. In this work, we introduce MACE, a new equivariant MPNN model that uses higher body order messages. In particular, we show that using four-body messages reduces the required number of message passing iterations to just two, resulting in a fast and highly parallelizable model, reaching or exceeding state-of-the-art accuracy on the rMD17, 3BPA, and AcAc benchmark tasks. We also demonstrate that using higher order messages leads to an improved steepness of the learning curves.

- GT1: Machine learning crash course
- GT2: Materials Representations
- **ST1: Symmetry in graph neural networks**
- ST2: Neural differential equations
- ST3: Generative models
- Recent trends

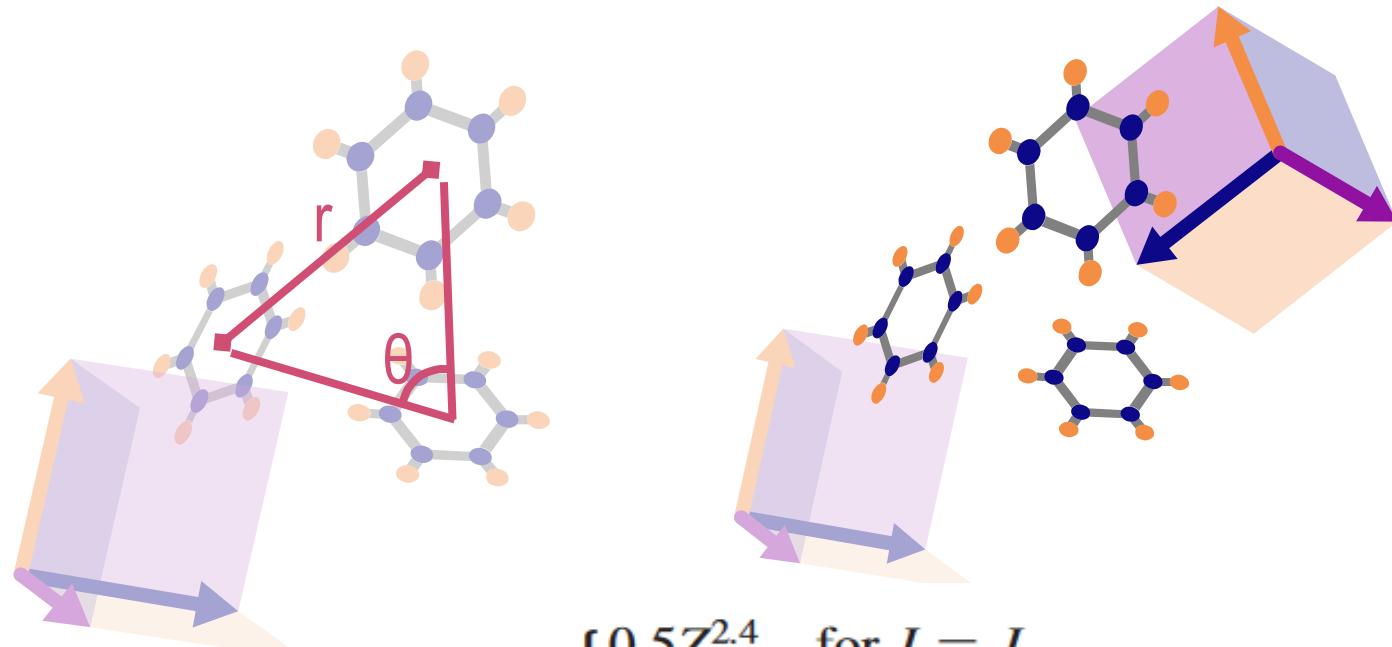
# Poor-man augmentation, Invariance, and Equivariance: Poor-man data augmentation



training with symmetry



# Invariant representations



$$M_{IJ} = \begin{cases} 0.5Z_I^{2.4} & \text{for } I = J, \\ \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} & \text{for } I \neq J. \end{cases}$$

$$d(\mathbf{M}, \mathbf{M}') = \sqrt{\sum_{I=1}^N (\varepsilon_I - \varepsilon'_I)^2}$$

- Focus on distances  $r$ , angles  $\theta$ , and invariant properties (e.g. Coulomb matrix, SOAP)

# Group and group representations

**Group  $G$ :** Set of elements combined with a binary operation

- Identity:  $I \in G$
- Closure:  $gh \in G$
- Associativity:  $(gh)k = g(hk)$
- Inverse:  $g^{-1} \in G$

**Group representation  $D(g, x)$ :**

Map a group to linear transform of a vector space (e.g. a matrix), preserving the group structure

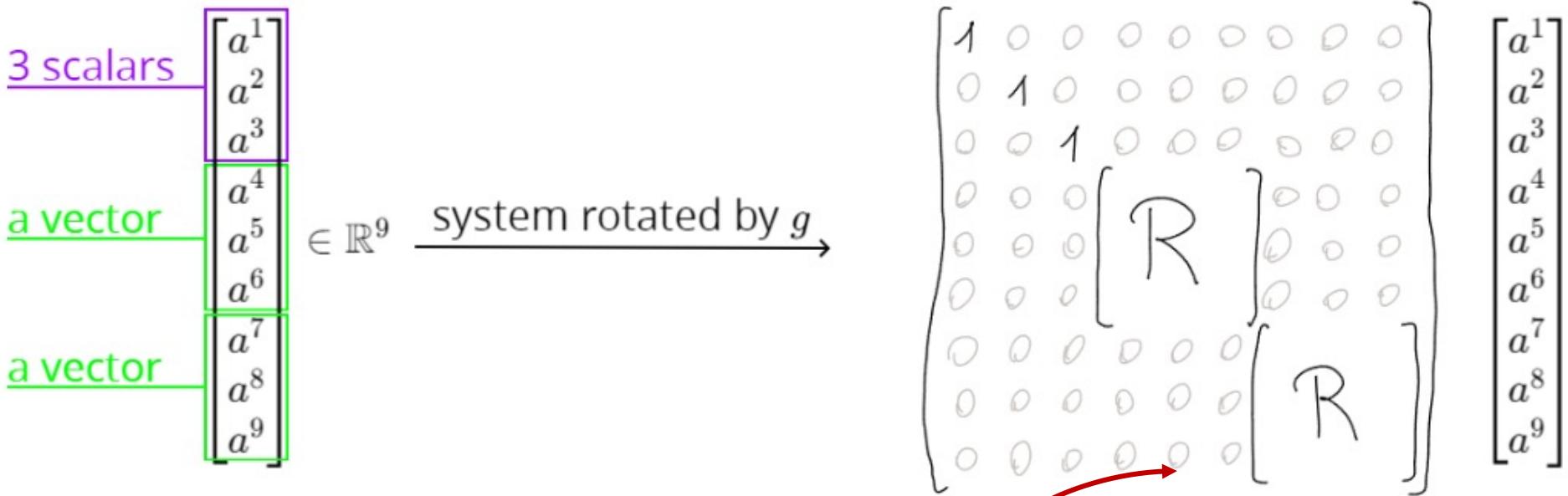
- $g \in G, x \in V$
- Linear:  $D(g, x + y) = D(g, x) + D(g, y)$
- Group structure:  
 $D(gh, x) = D(g, D(h, x))$   
 $D(hg, x) = D(h, D(g, x))$

**Group representation  $D(g)x$  :**

An Equivalent notation

- $D(g)$  is mapping:  $V \rightarrow V$
- Linear:  $D(g)(x + y) = D(g)x + D(g)y$
- Group structure:  
 $D(gh) = D(g)D(h)$   
 $D(hg) = D(h)D(g)$

# Example of representation

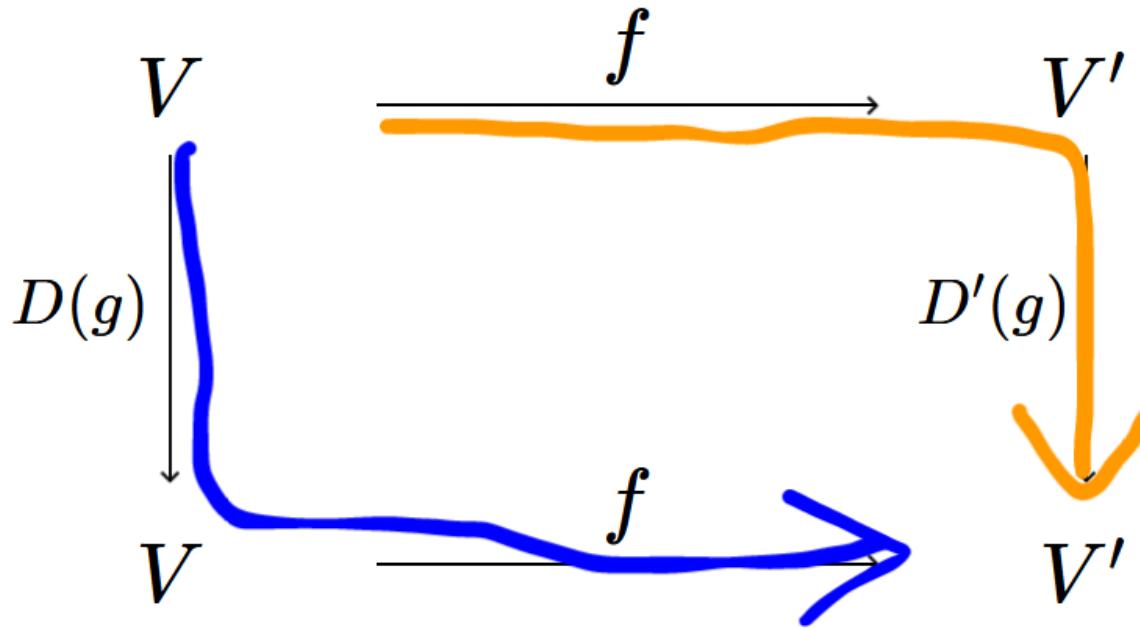


- Vector space = 3 scalars + 2 vectors
- Rotation group.
- Representation  $D$
- Code (Afternoon):

```
1 irreps = "3x0e + 2x1o"
```

## Equivariance: how a function $f$ behaves w.r.t group actions

- $x \in V$
- $f: V \rightarrow V'$
- $D(g)$ : group action on  $V$
- $D'(g)$ : group action on  $V'$
- Equivariance:  $f$  commutes with group actions  $D(g)$  and  $D'(g)$



$$f(D(g)x) = D'(g)f(x)$$

## Tensor product

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$  transforming with  $D(g)$

$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$  transforming with  $D'(g)$

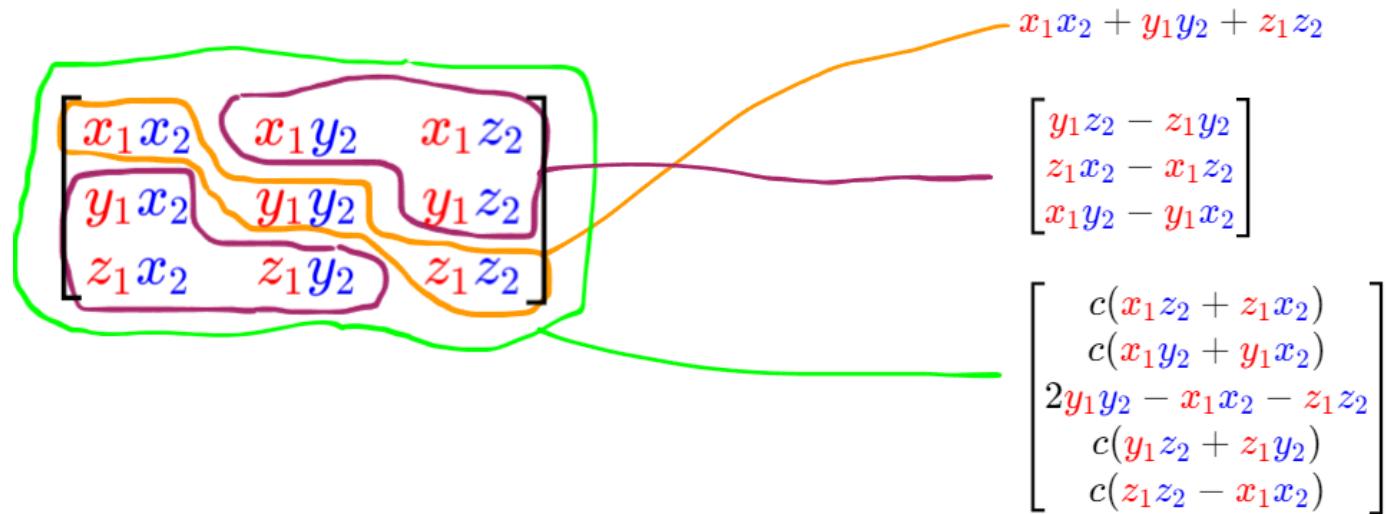
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \otimes \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} x_1y_1 & x_1y_2 & x_1y_3 & x_1y_4 & x_1y_5 \\ x_2y_1 & x_2y_2 & x_2y_3 & x_2y_4 & x_2y_5 \\ x_3y_1 & x_3y_2 & x_3y_3 & x_3y_4 & x_3y_5 \end{bmatrix}$$

transforms with  $D(g) \otimes D'(g)$

## Irreducible representation: $3 \times 3 = 1 + 3 + 5$

- Large representation (after tensor product) can be decomposed
- Tensor product of two 3D Irreps. can be decomposed into 3 Irreps

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \otimes \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1x_2 & x_1y_2 & x_1z_2 \\ y_1x_2 & y_1y_2 & y_1z_2 \\ z_1x_2 & z_1y_2 & z_1z_2 \end{bmatrix}$$

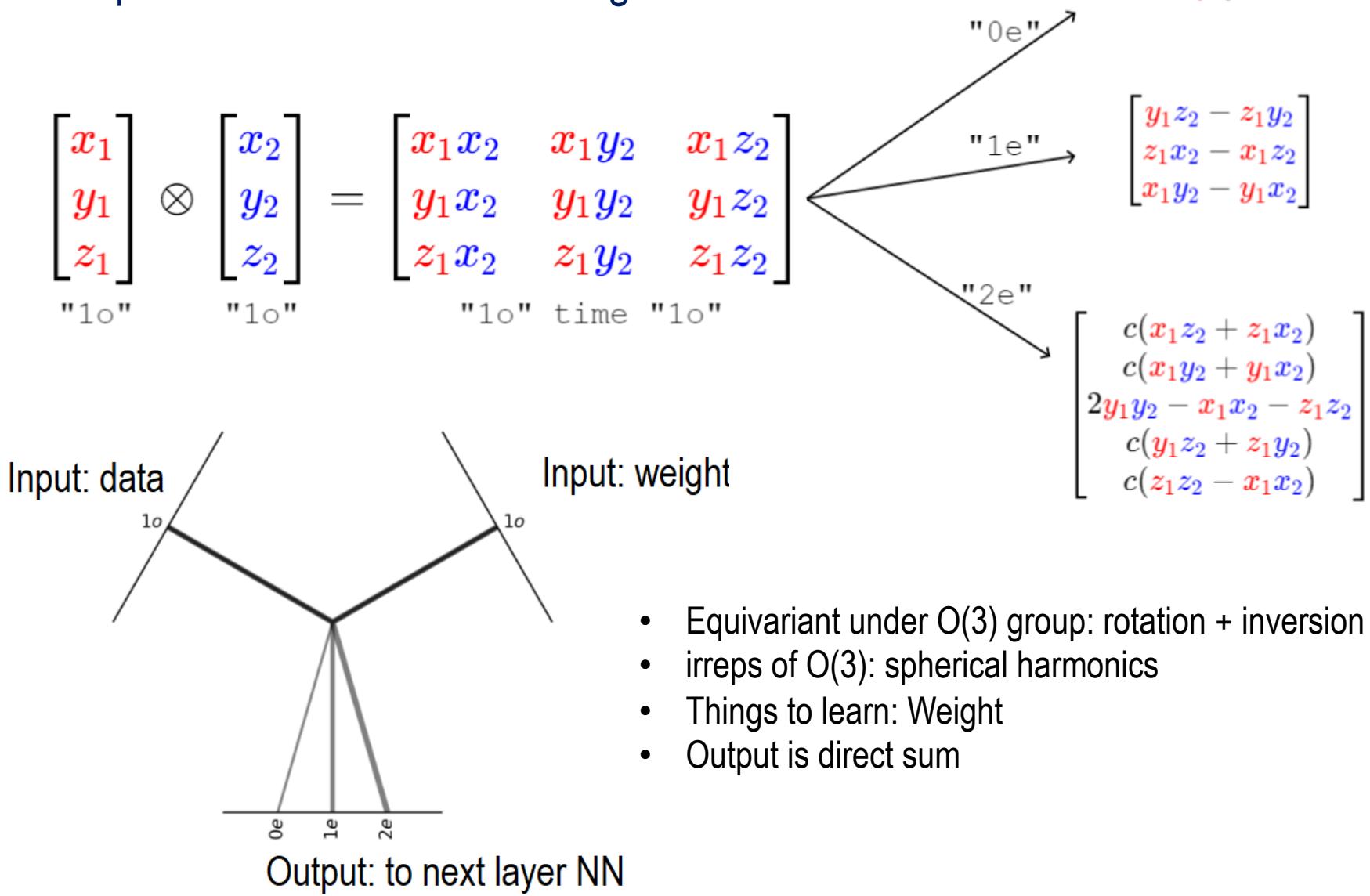


$$S_1 \otimes S_2 = |S_1 - S_2| \oplus |S_1 - S_2| + 1 \oplus \dots \oplus S_1 + S_2$$

$$S_1 = 1(S_{z1} = -1, 0, 1), S_2 = 1(S_{z2} = -1, 0, 1)$$

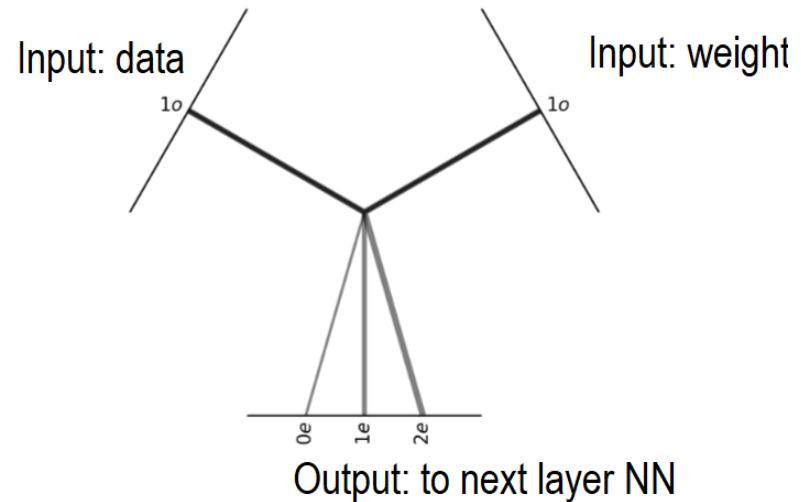
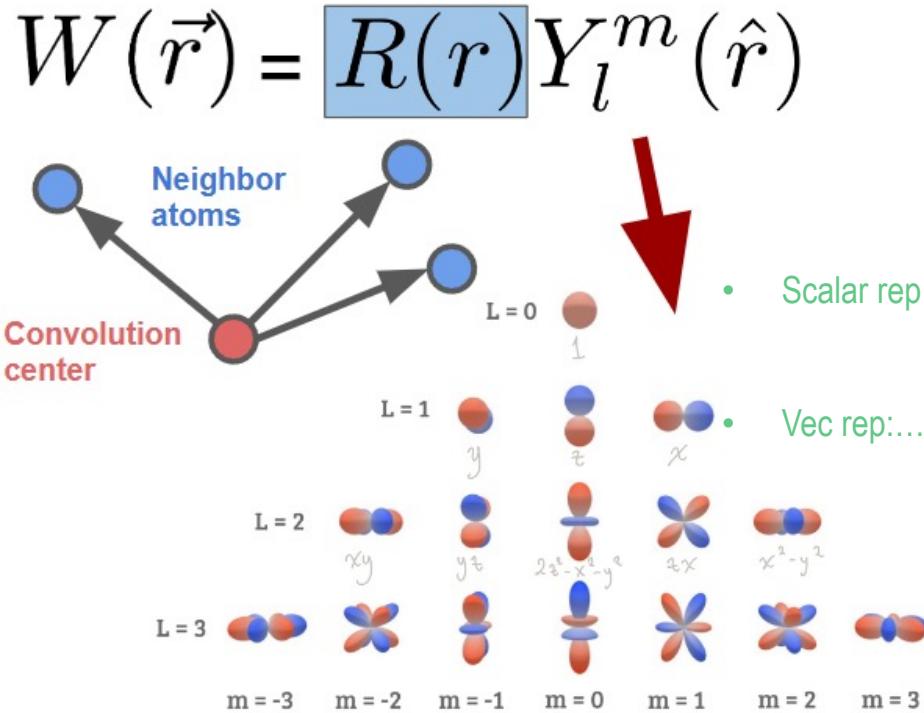
$$S = S_1 + S_2 = 0(S_z = 0), 1(S_z = -1, 0, 1), 2(S_z = -2, -1, 0, 1, 2)$$

# One step closer to Machine learning



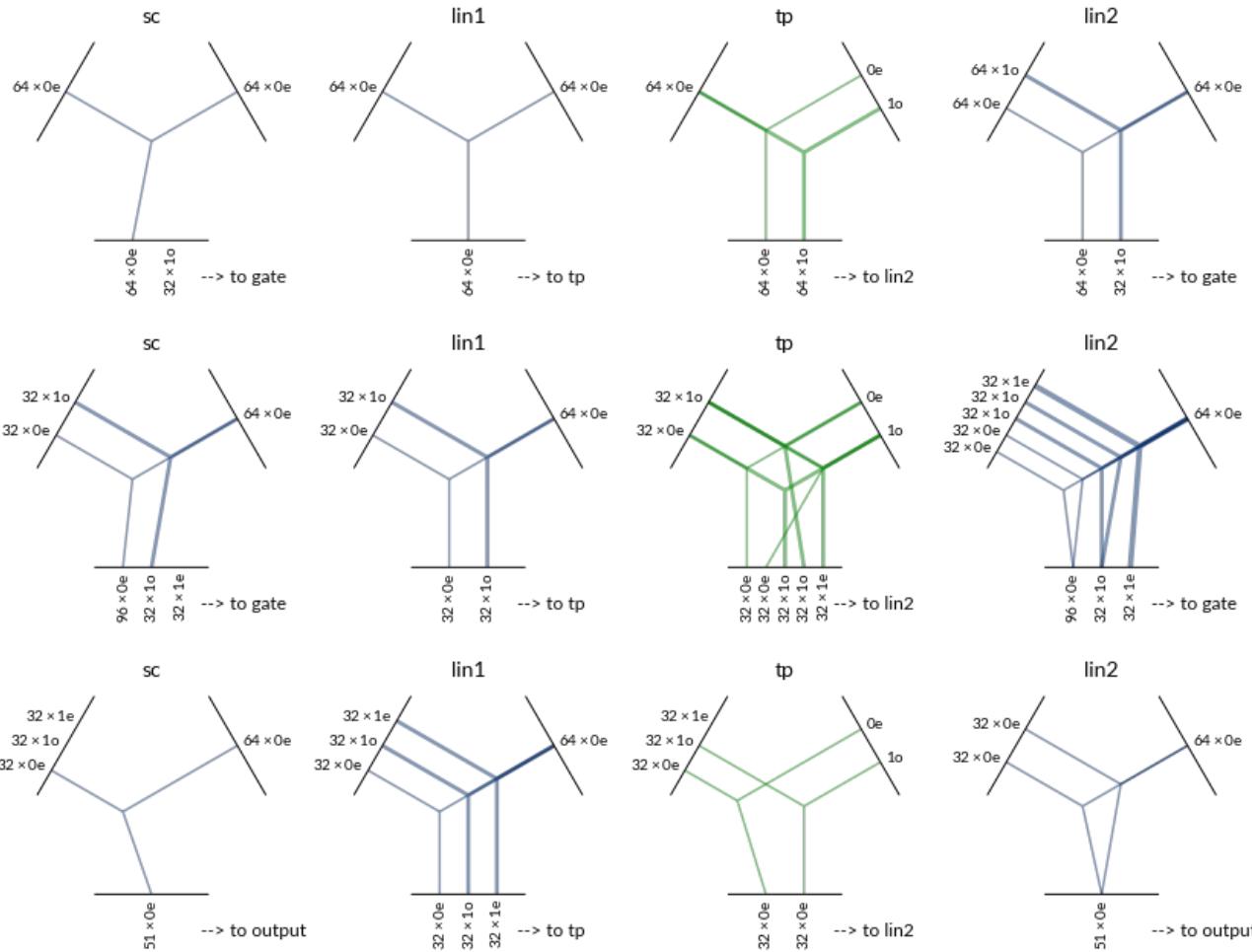
- Equivariant under  $O(3)$  group: rotation + inversion
- irreps of  $O(3)$ : spherical harmonics
- Things to learn: Weight
- Output is direct sum

# One further step closer: Weight in E3NN



- Angular part:  $Y_{lm}(\hat{r})$ , i.e. Irreps of  $O(3)$  Group
- Radial part:  $R(|\vec{r}|)$  as a FFNN.

# Tensor product with weights in E3NN

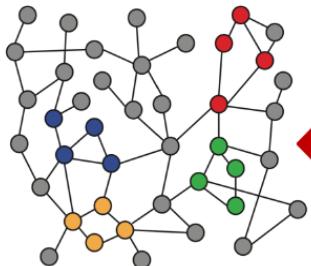


- Recall in FFNN,  

$$y = \tanh(Wx + b)$$
- Weight \* Data
- Here, the same idea  

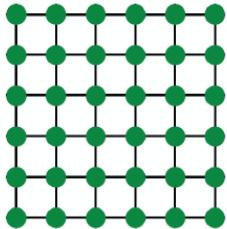
$$\text{Data} \otimes \text{Weight}$$
- Output dimension can be reduced (automatically)

# GNNs: where they come from

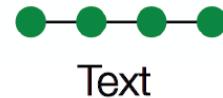


Networks

VS.

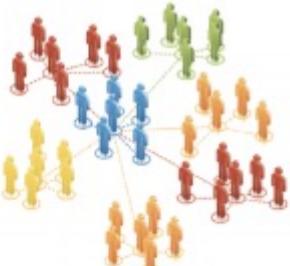


Images

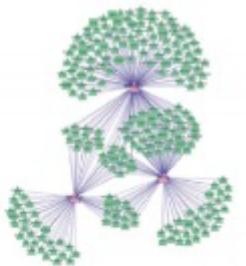


Text

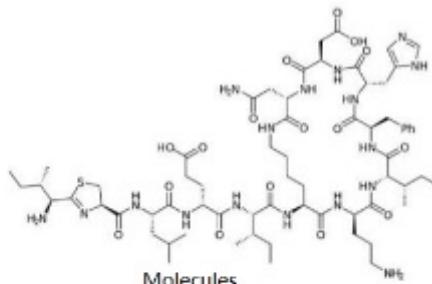
- More flexible than structured data
- Exist in various contexts
- Naturally represent molecules and solids
- Flexibly input new, unseen compounds



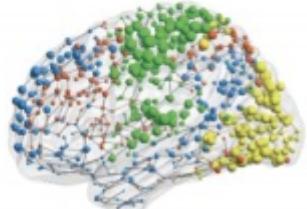
Social networks



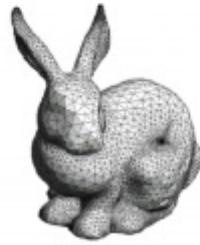
Regulatory networks



Molecules



Functional networks



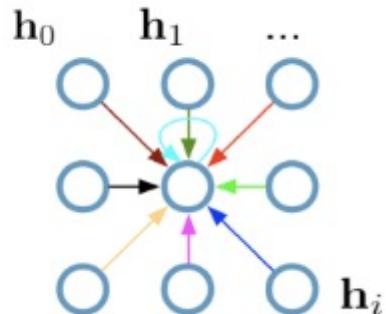
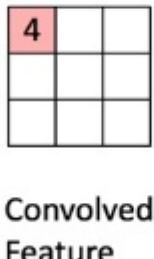
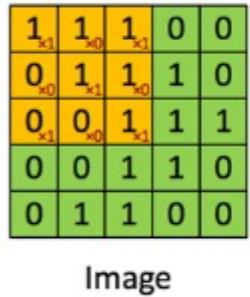
3D shapes



Maps

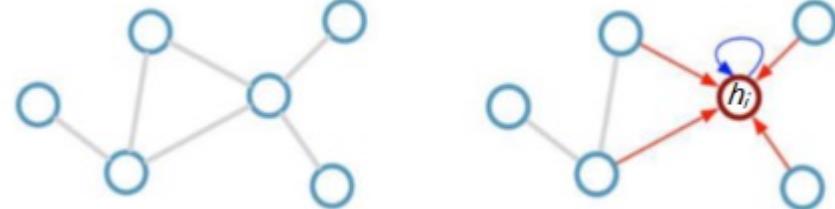
# One more thing needed: embedding as input

Convolutional neural network (CNN)



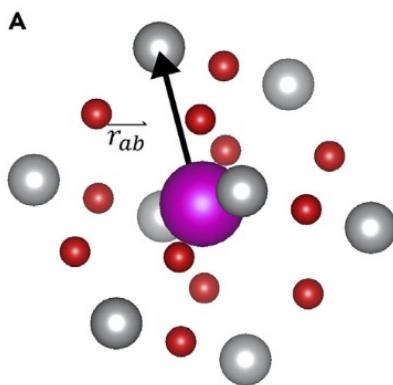
$$h_4^{(l+1)} = \sigma \left( W_0^{(l)} h_0^{(l)} + W_1^{(l)} h_1^{(l)} + W_2^{(l)} h_2^{(l)} \dots + W_8^{(l)} h_8^{(l)} \right)$$

General Graph convolutional network (GCN)



$$h_i^{(l+1)} = \sigma \left( W_0^{(l)} h_i^{(l)} + \sum_{j=i's \text{ neighbors}} \frac{1}{c_{ij}} W_1^{(l)} h_j^{(l)} \right)$$

GNN in materials  
e.g. E3NN



$x_a^{prop}$

Fe [...,  $m_{Fe}$ , 0, 0, ...]  
Co [..., 0,  $m_{Co}$ , 0, ...]  
Ni [..., 0, 0,  $m_{Ni}$ , ...]  
Index  $i$  ...,  $Z_{Fe}$ ,  $Z_{Co}$ ,  $Z_{Ni}$ , ...

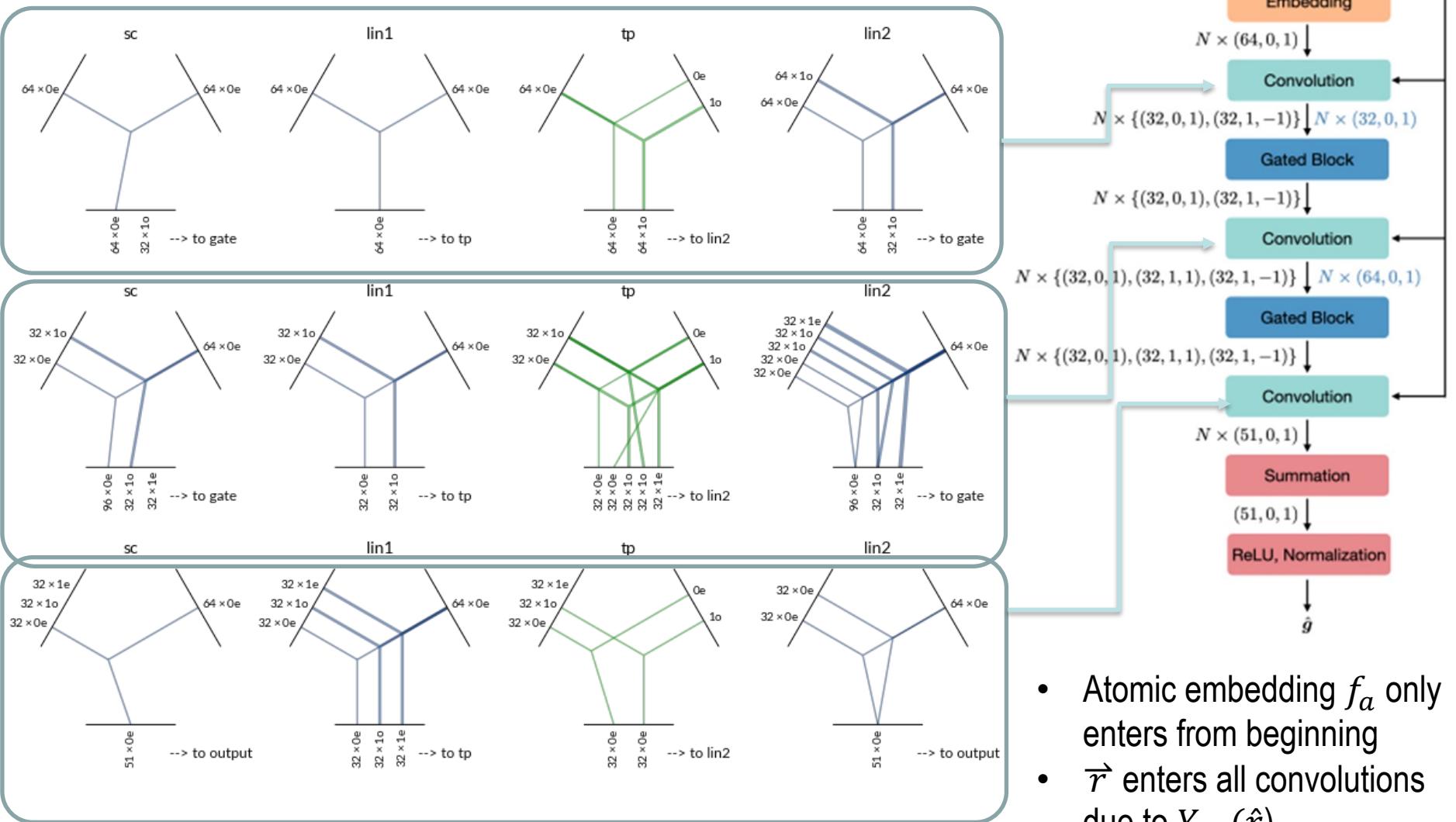
$m$ : atomic weight  
 $a$ : atomic radius  
 $\chi$ : electronegativity  
 $P$ : dipole polarizability  
 $Z$ : atomic number

$x_a^{mag}$

Fe [...,  $a_{Fe}$ , 0, 0, ...,  $\chi_{Fe}$ , 0, 0, ...,  $P_{Fe}$ , 0, 0, ...]  
Co [..., 0,  $a_{Co}$ , 0, ..., 0,  $\chi_{Co}$ , 0, ..., 0,  $P_{Co}$ , 0, ...]  
Ni [..., 0, 0,  $a_{Ni}$ , ..., 0, 0,  $\chi_{Ni}$ , ..., 0, 0,  $P_{Ni}$ , ...]

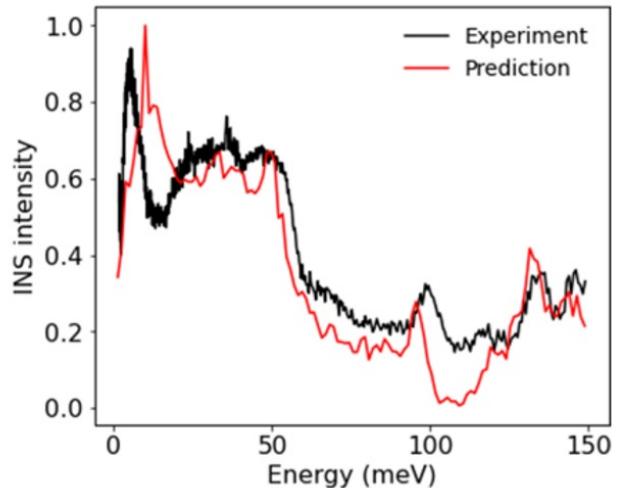
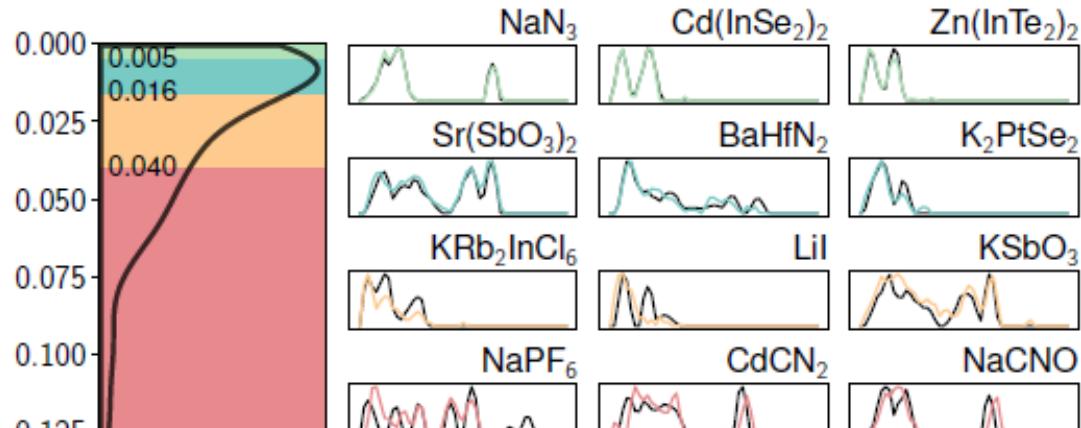
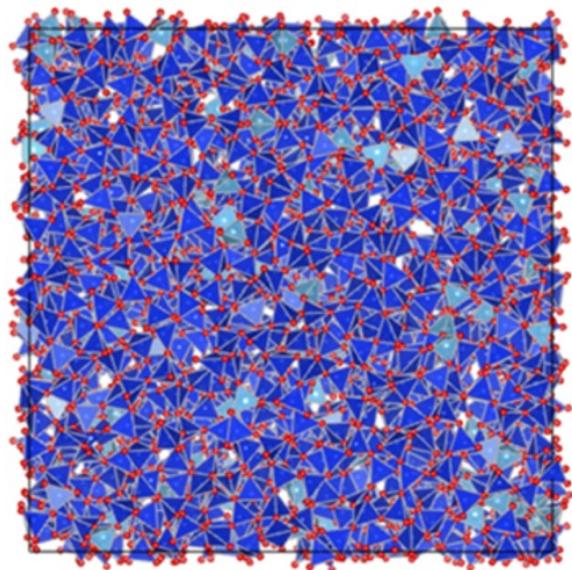
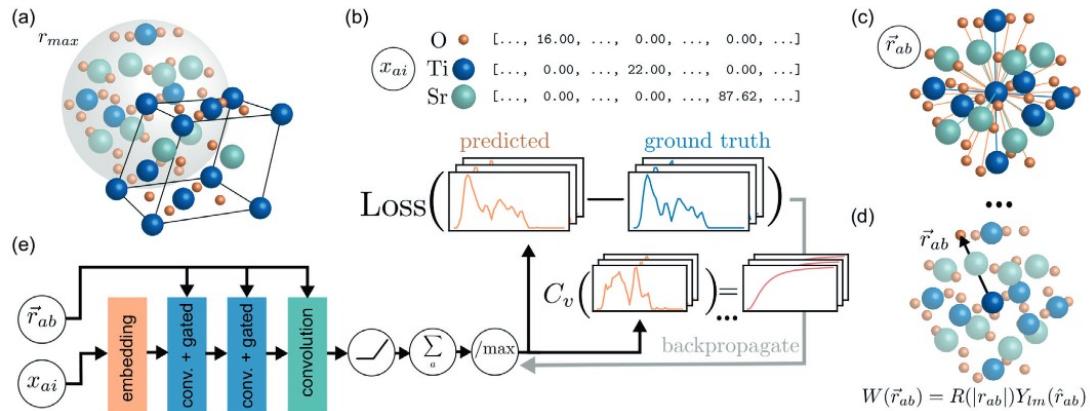
$$f'_a = \frac{1}{\sqrt{Z}} \sum_{b: |\vec{r}_{ab}| < r_{max}} f_b \otimes (h(\|\vec{r}_{ab}\|)) Y(\vec{r}_{ab} / \|\vec{r}_{ab}\|)$$

# Put everything together for machine learning



- Atomic embedding  $f_a$  only enters from beginning
- $\vec{r}$  enters all convolutions due to  $Y_{lm}(\vec{r})$

# Example on E3NN: Phonon DOS



- GT1: Machine learning crash course
- GT2: Materials Representations
- ST1: Symmetry in graph neural networks
- **ST2: Neural differential equations**
- ST3: Generative models
- Recent trends

# Physics Informed Neural Network (PINN)

- Learn hidden info (including equation parameters) from observables
- Start from an equation:  $u' + F(u, t) = 0$
- Represent variable as Neural network:  $\hat{u}(t) = NN_{\theta}(t)$
- Observe data  $d_i$  at time  $t_i$ , residual  $\hat{e}(t) = \hat{u}' + F(\hat{u}, t)$ . Loss function  $\mathcal{L}(\hat{u}, \theta)$  contains uninformed (data) + informed (ODE)

$$\mathcal{L}(\hat{u}, \theta) = \sum_{i=0}^N \|\hat{u}_i - d_i\|^2 + \sum_{i=0}^N \|\hat{e}_i\|^2$$

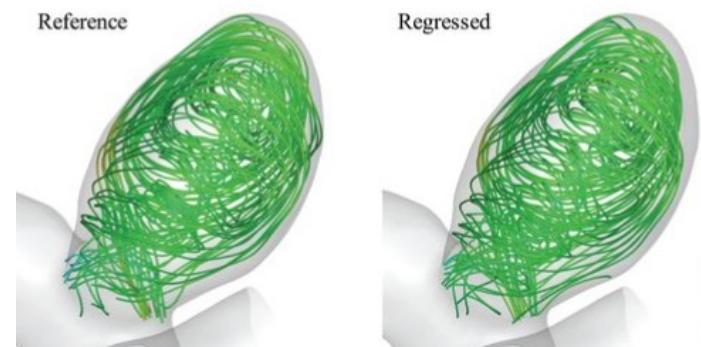
# Example of PINN: learn velocity fields from concentration



- Observed: concentration  $c_t + uc_x + vc_y + wc_z = \text{Pe}^{-1}(c_{xx} + c_{yy} + c_{zz}),$
- Hidden but governing: Navier-Stokes equation

$$\begin{aligned} u_t + uu_x + vu_y + wu_z &= -p_x + \text{Re}^{-1}(u_{xx} + u_{yy} + u_{zz}), \\ v_t + uv_x + vv_y + wv_z &= -p_y + \text{Re}^{-1}(v_{xx} + v_{yy} + v_{zz}), \\ w_t + uw_x + vw_y + ww_z &= -p_z + \text{Re}^{-1}(w_{xx} + w_{yy} + w_{zz}), \\ u_x + v_y + w_z &= 0. \end{aligned}$$

- Loss function: 
$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N |c(t^n, x^n, y^n, z^n) - c^n|^2 + \sum_{i=1}^5 \frac{1}{M} \sum_{m=1}^M |e_i(t^m, x^m, y^m, z^m)|^2$$
- Recovered: Velocity fields

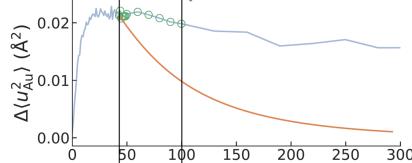


# Neural ODE (with adjoint state):

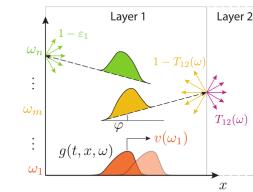
	PINN	Neural ODE
Neural network	Solution	Solution derivative
Loss	Direct add PDE to Loss	With Lagrangian multiplier
Advantage	Intuitive, solve PDE	Solved vanishing gradient
Disadvantage	Vanishing-gradient	Need numerical integration

$$L(\theta) = \int_{t_0}^{t_f} l(g(t, \theta)) dt + \int_{t_0}^{t_f} \lambda(t) \left( \frac{\partial}{\partial t} g(t, \theta) - h(g(t, \theta), \theta) \right) dt$$

Summing up the loss at each data point



A constantly zero term, times a Lagrangian multiplier

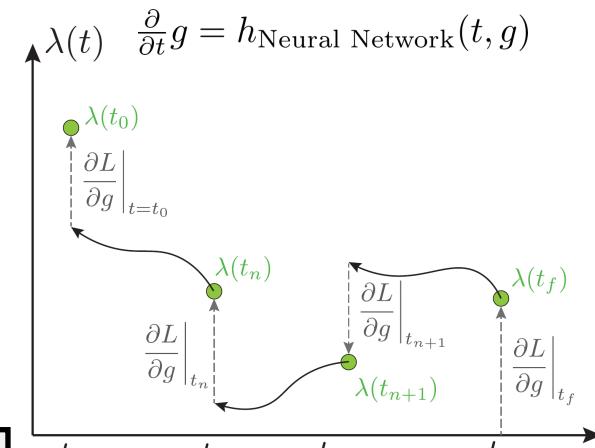


Taking the derivative:

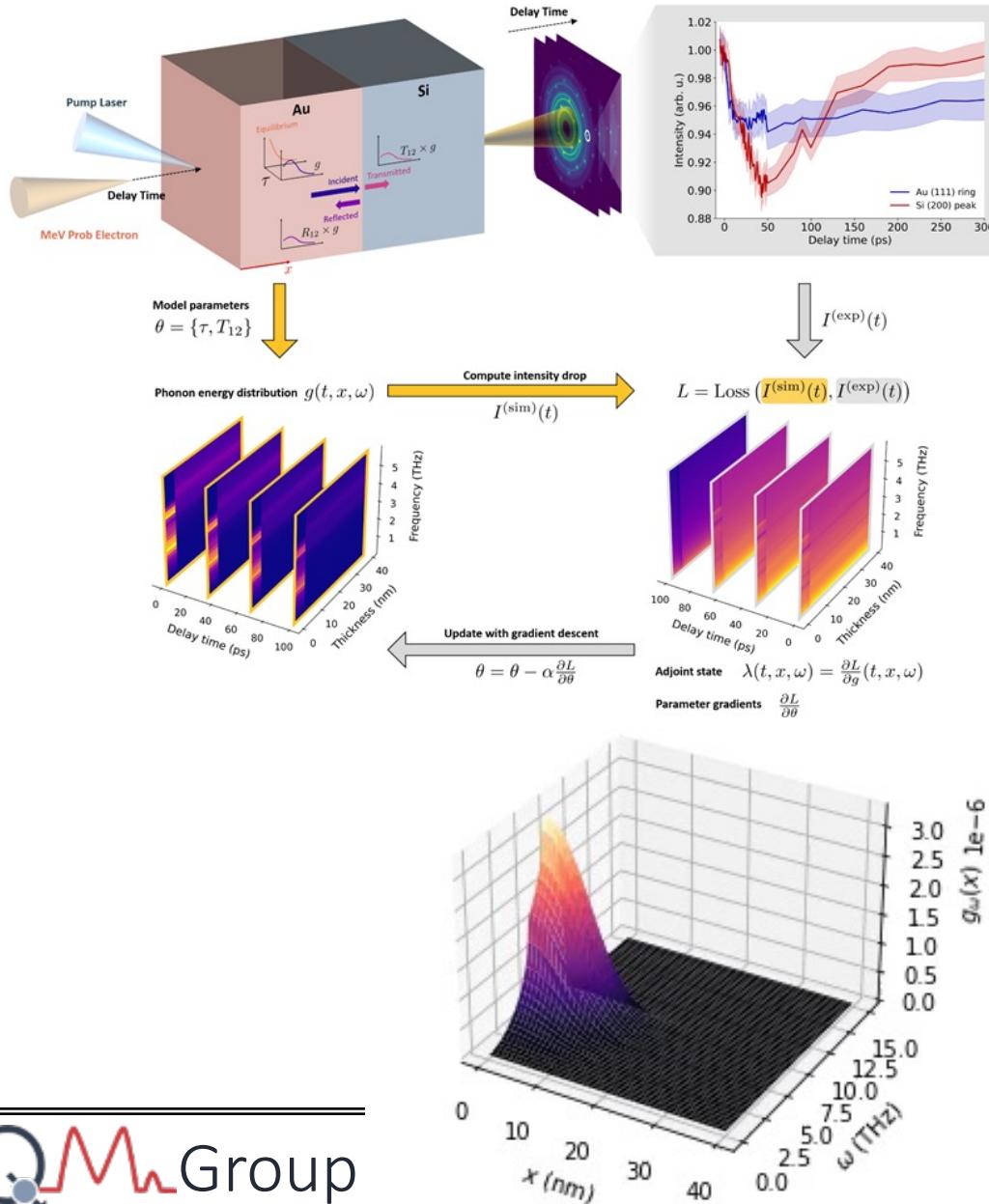
$$\frac{\partial L}{\partial \theta} = \int_{t_0}^{t_f} \left( \frac{\partial l}{\partial \theta} - \lambda \frac{\partial h}{\partial \theta} \right) dt + \lambda(t_f) \frac{\partial g}{\partial \theta} \Big|_{t_f} - \int_{t_0}^{t_f} \left( \frac{\partial \lambda}{\partial t} + \lambda \frac{\partial h}{\partial g} - \frac{\partial l}{\partial g} \right) dt = \eta(t_0)$$

Backward:  $\frac{\partial \eta}{\partial t} = -\lambda \frac{\partial f}{\partial \theta}, \quad \eta(t_f) = 0 \quad \frac{\partial \lambda}{\partial t} = -\lambda \frac{\partial h}{\partial g} + \frac{\partial L}{\partial g} \Big|_{t=t_0}, \quad \lambda(t_0) = 0$

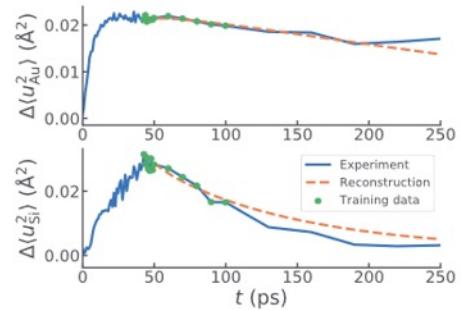
Forward:  $\frac{\partial}{\partial t} g(t, \theta) = h(g(t, \theta), \theta) = -\mu v_g \frac{\partial g}{\partial x} - \frac{g - g^{\text{eq}}}{\tau}$



# Example of Neural ODE: Ultrafast diffraction



- Observed: TR-diffraction



- Governing: Boltzmann transport of phonons (derivative as  $h=NN$ )  

$$h = -\mu v_g \frac{\partial f}{\partial x} - \frac{g-g^{\text{eq}}}{\tau}$$
- Uncover relaxation  $\tau(\omega)$ , transmission across interface  $T(\omega)$ .
- Further uncover real-time, real-space, frequency-resolved phonon dynamics

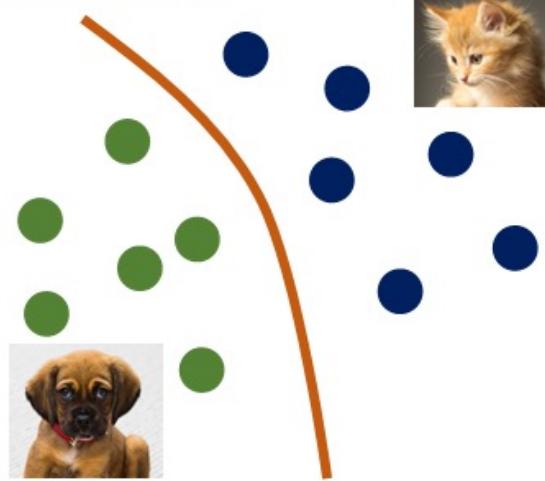
- GT1: Machine learning crash course
- GT2: Materials Representations
- ST1: Symmetry in graph neural networks
- ST2: Neural differential equations
- **ST3: Generative models**
- Recent trends

# Generative model

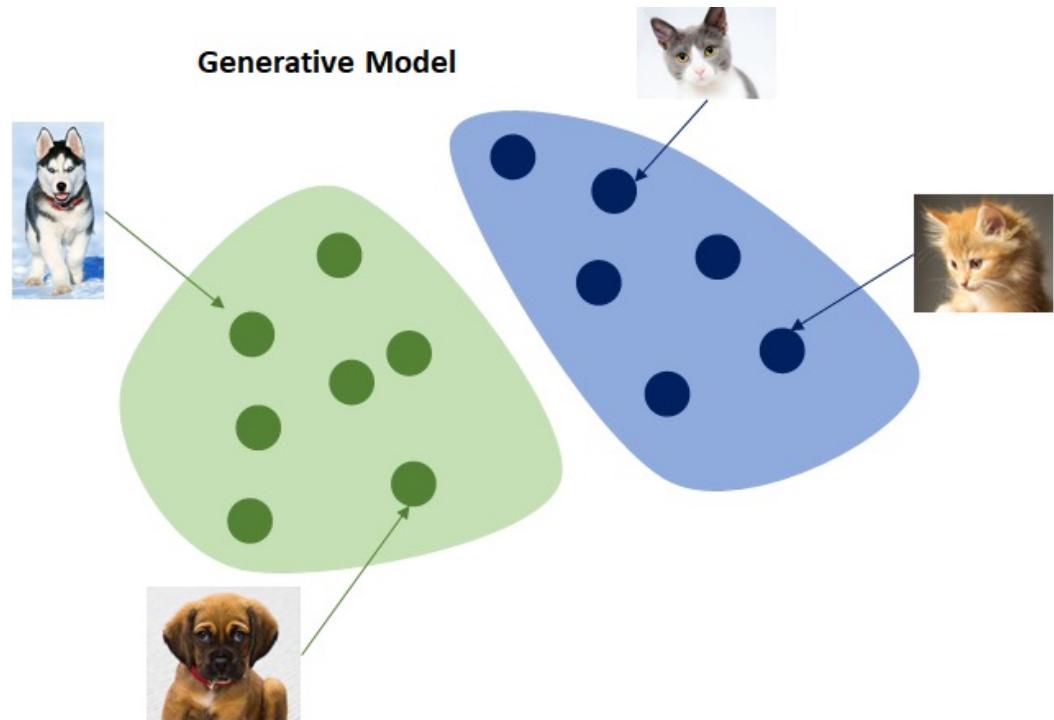
Discriminant model: predict the property of input images. Classification, regression.

Generative model: learn the distribution of the data based on the dataset, and sample from the dataset.

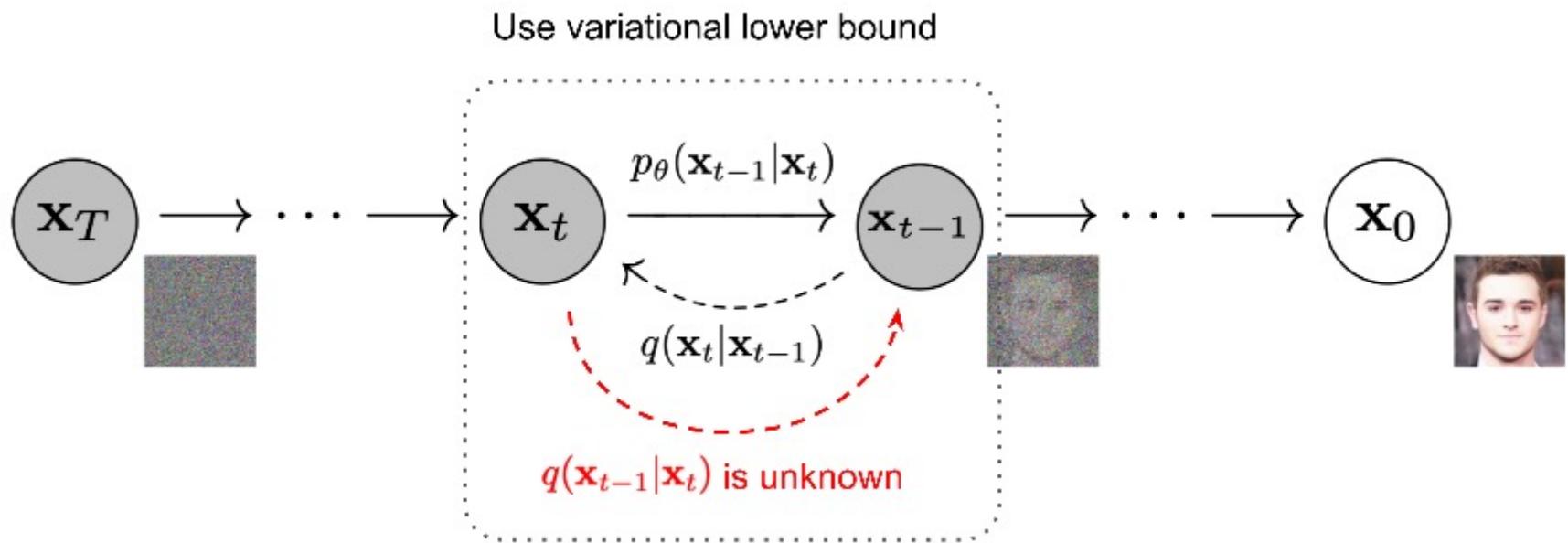
Discriminant Model



Generative Model



# Image generation by diffusion model



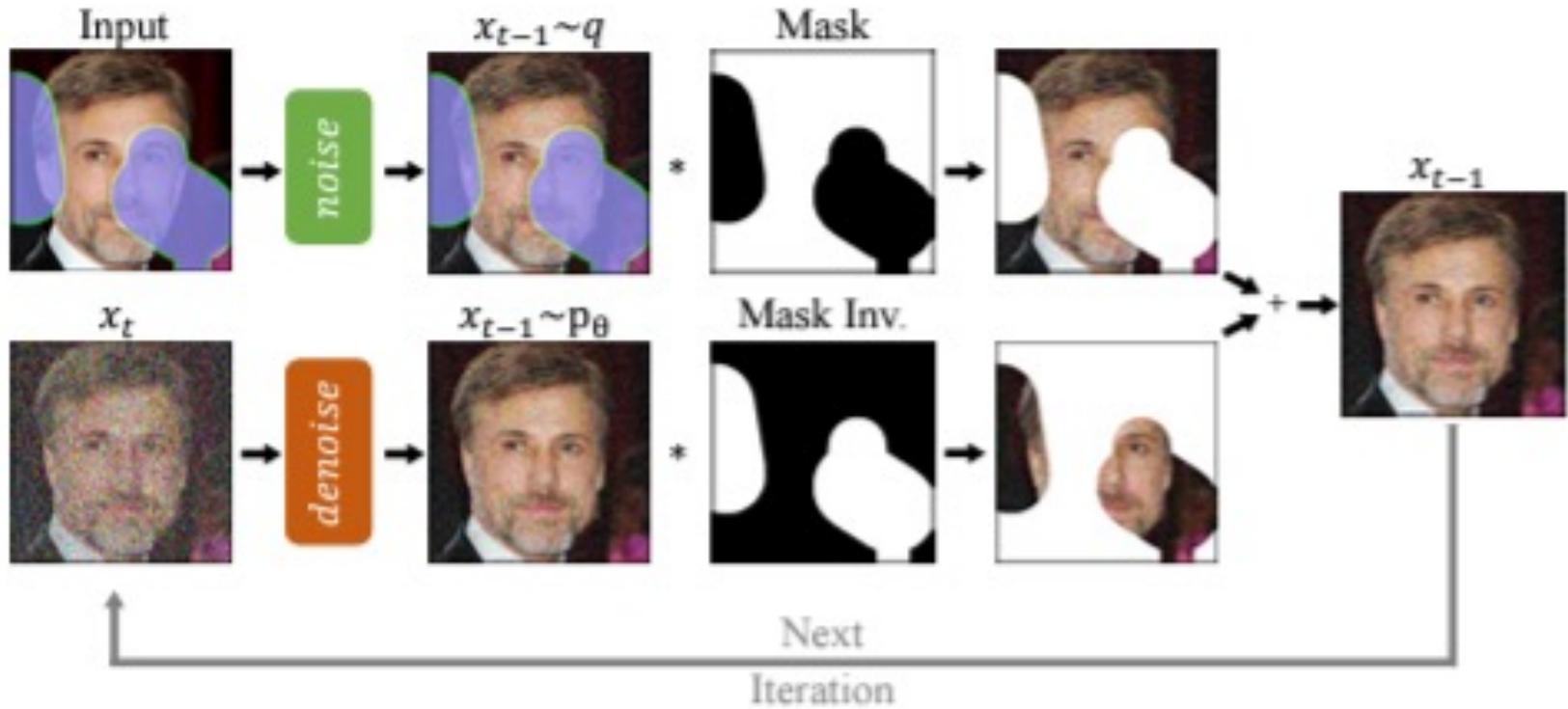
(1) Add noise to the image

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := p(\mathbf{x}_T) \prod_{t=1}^T \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

(2) Denoise the image

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

# Imprinting process masked generation



$$x_{t-1}^{\text{known}} \sim \mathcal{N} \left( \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) \mathbf{I} \right) \quad (1) \text{ Add noise to the known area.}$$

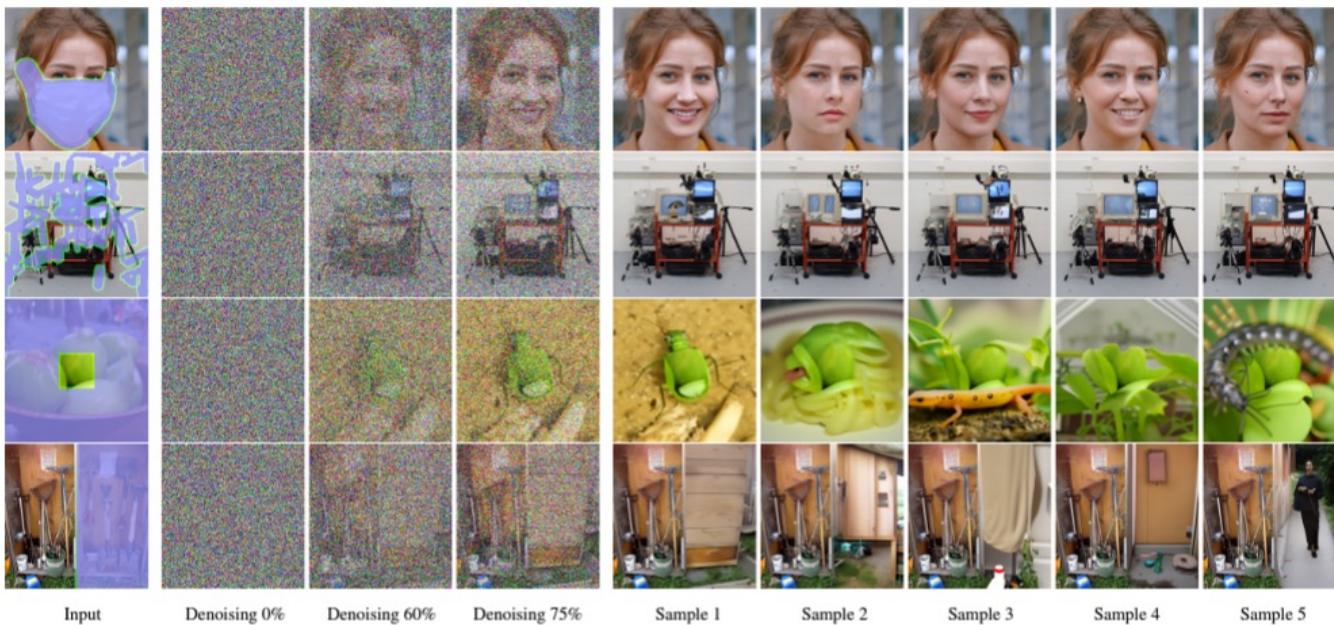
$$x_{t-1}^{\text{unknown}} \sim \mathcal{N} (\mu_\theta (x_t, t), \Sigma_\theta (x_t, t)) \quad (2) \text{ Denoise the unknown area.}$$

$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}} \quad (3) \text{ Combine known and unknown area.}$$

# RePaint: Inpainting using Denoising Diffusion Probabilistic Models

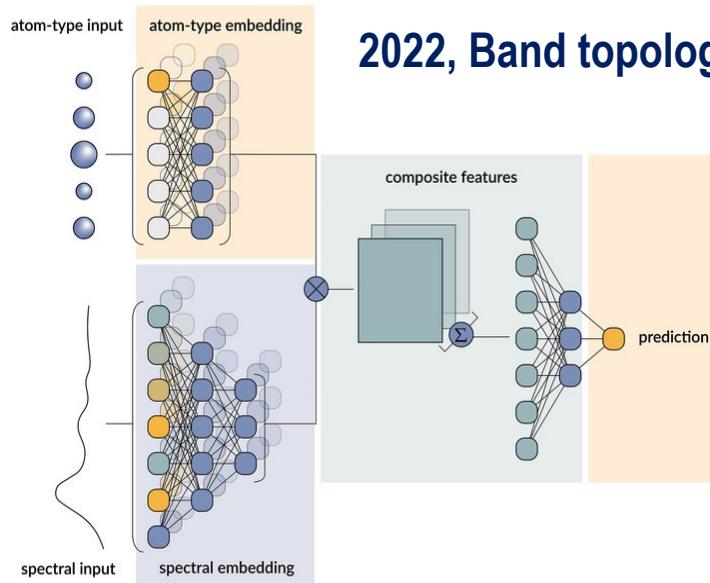


Input Image.



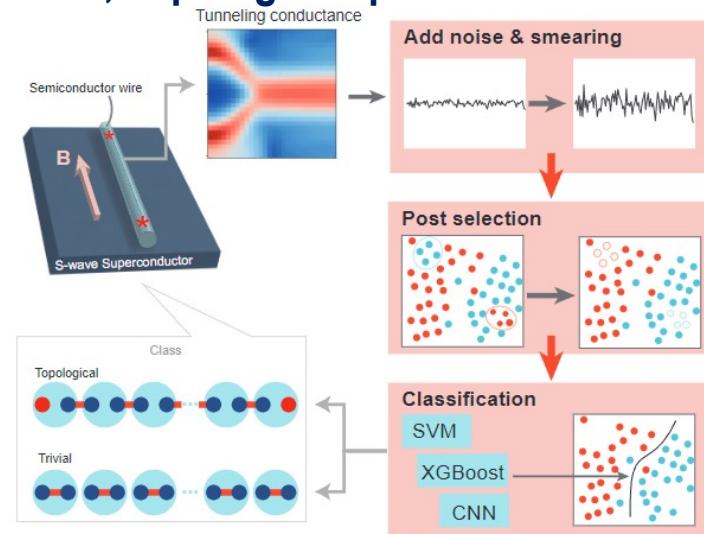
- GT1: Machine learning crash course
- GT2: Materials Representations
- ST1: Symmetry in graph neural networks
- ST2: Neural differential equations
- ST3: Generative models
- **Recent trends**

# Trend 1: From existing data to generating new data



2022, Band topology

2024, Topological qubits

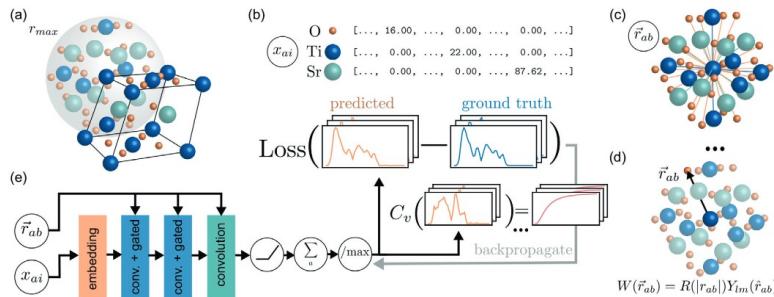


- **Problem:** Topological materials hold great promise for next-generation electronic devices, yet identifying topology has been challenging
- **Solution:** We design a CNN classifier, showing the topology is already encoded in simple X-ray absorption spectra
- **Data:** pre-existing X-ray absorption database  $\cap$  topological database.
- **Ref:** NA, ML, Advanced Materials 34, 202204113 (2022)

- **Problem:** Majorana zero modes (MZM). Is basis for topological quantum computing but is challenging to identify due to spurious signals.
- **Solution:** We build a machine-learning that identifies the MZMs from experimental signals.
- **Data:** Generated from scratch from Hamiltonian models + quantum transport calculations
- **Feature:** Prove this challenge is machine-classifiable, with 94% accuracy.
- **Ref:** MC, ML, Matter 7, 1-14 (2024)

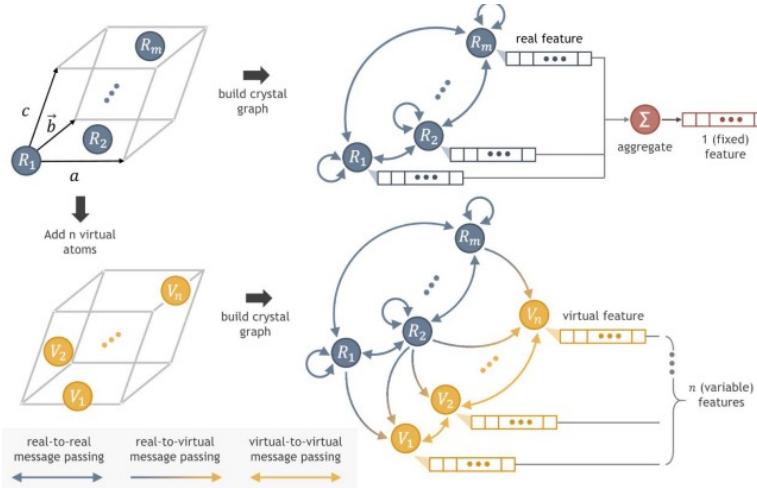
# Trend 2: From existing model to building new model

## 2021, Phonon DOS



- Problem:** Phonons are main heat carriers in semiconductors, yet phonon properties are challenging to measure or compute
- Solution:** We build E3NN-based framework that takes in crystal structure and outputs phonon DOS
- Ref:** ZC, ML, Advanced Science 8, 2004214 (2021)

## 2024, Phonon dispersion



- Problem:** Phonon dispersion is critical for thermal properties but hard to compute or measure
- Solution:** We design a whole architecture called “virtual nodes” that predicts full phonon dispersion relations from crystal structures
- Feature:** Order-of-magnitudes faster than any existing method, including MLIPs, without higher accuracy than MLIP
- Ref:** RO, ..., YQ Cheng, ML, arXiv:2301.02197, *Nature Computational Science In Production* (2024)

# Acknowledgements



Ryotaro Okabe,  
**Chemistry**



Abhijatmedhi  
Chotrattanapituk, **EECS**



Tiya Boonkird,  
**NSE**



Thanh Nguyen  
(PhD 2024)



Chuliang Fu,  
postdoc



Zhantao Chen,  
PhD' 22



Nina Andrejevic,  
PhD' 22



Mouyang Cheng,  
**DMSE**



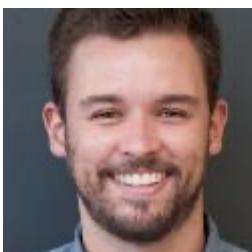
Gene Siriviboon,  
**Physics**



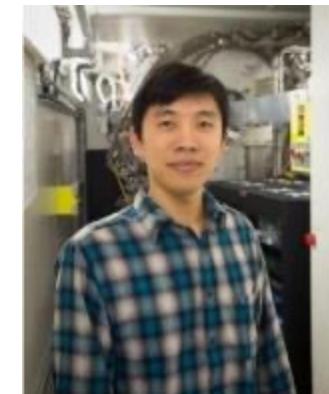
Nathan Drucke,  
PhD'24



Manasi Mandal,  
postdoc



Michael Landry,  
Postdoc



YQ Cheng, ORNL

