Physics-Informed Machine Learning at the Atomic Scale

Martin Uhrín

Computational Atomistic Methods & Machine Learning, SIMaP, UGA

UGA

Materials informatics

AiiDA MATERIALSCLOUD

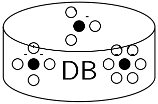
- High-throughput quantum mechanical calculations
- Turnkey workflows for calculation of properties
- Large quantities of readily reusable property calculations

Invariant and equivariant machine learning

milad e3nn

- Theory of atomistic structure representations
- Inversion of invariant fingerprints
- Highly data-efficient equivariant neural networks

Generative models



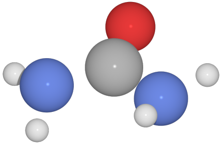
- Linking 3D geometry to properties
- Motif based construction of molecules and materials
- Coupling experiment and theory

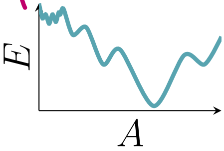
Physics inspired machine learning models

Accelerating property prediction with machine learning

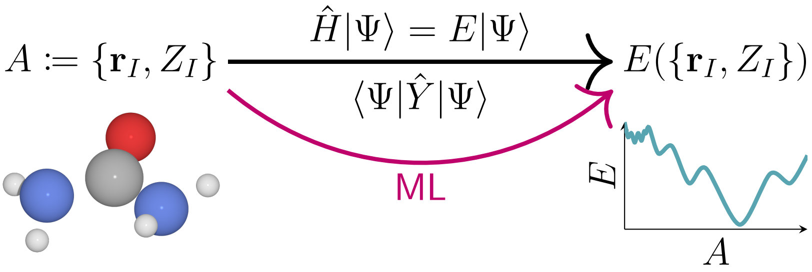
$$A := \{\mathbf{r}_I, Z_I\} \xrightarrow[\langle\Psi|\hat{Y}|\Psi\rangle]{\hat{H}|\Psi\rangle = E|\Psi\rangle} E(\{\mathbf{r}_I, Z_I\})$$

ML

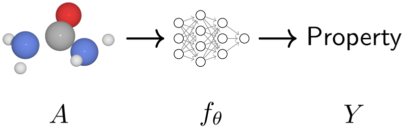




Accelerating property prediction with machine learning



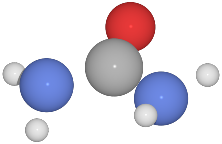
Our ML models learns to map from atomic coordinates to properties:
$f_\theta : A \to Y$,

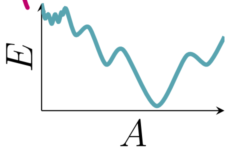


Accelerating property prediction with machine learning

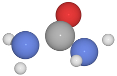
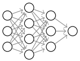
$$A := \{\mathbf{r}_I, Z_I\} \xrightarrow[\langle\Psi|\hat{Y}|\Psi\rangle]{\hat{H}|\Psi\rangle = E|\Psi\rangle} E(\{\mathbf{r}_I, Z_I\})$$

ML





Our ML models learns to map from atomic coordinates to properties:
$f_\theta : A \to Y$,

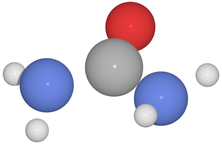
 $\longrightarrow$  $\longrightarrow$ Property

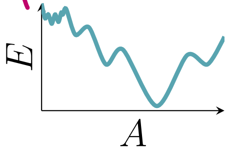
A $\quad$ f_θ $\quad$ Y

Accelerating property prediction with machine learning

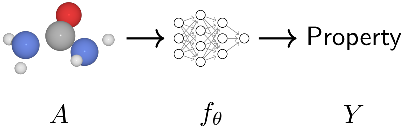
$$A := \{\mathbf{r}_I, Z_I\} \xrightarrow[\langle\Psi|\hat{Y}|\Psi\rangle]{\hat{H}|\Psi\rangle = E|\Psi\rangle} E(\{\mathbf{r}_I, Z_I\})$$

ML





Our ML models learns to map from atomic coordinates to properties: $f_\theta : A \to Y$,



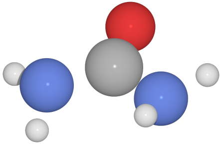
but A is not a suitable input. We want something that respects the symmetries of the underlying physical laws:

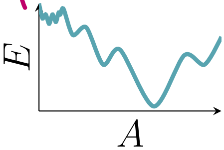
- permutation
- translation
- rotation

Accelerating property prediction with machine learning

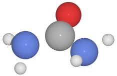
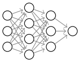
$$A := \{\mathbf{r}_I, Z_I\} \xrightarrow[\langle\Psi|\hat{Y}|\Psi\rangle]{\hat{H}|\Psi\rangle = E|\Psi\rangle} E(\{\mathbf{r}_I, Z_I\})$$

ML





Our ML models learns to map from atomic coordinates to properties:
$f_\theta : A \to Y$,

 $\longrightarrow$  $\longrightarrow$ Property

A f_θ Y

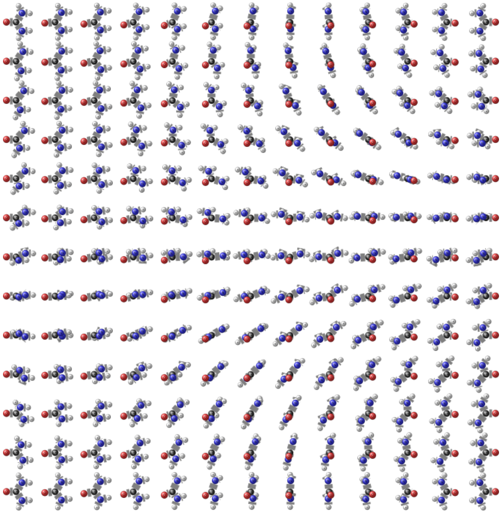
but A is not a suitable input. We want something that respects the symmetries of the underlying physical laws:

- permutation
- translation
- rotation

furthermore, for generative models we want representations that are:

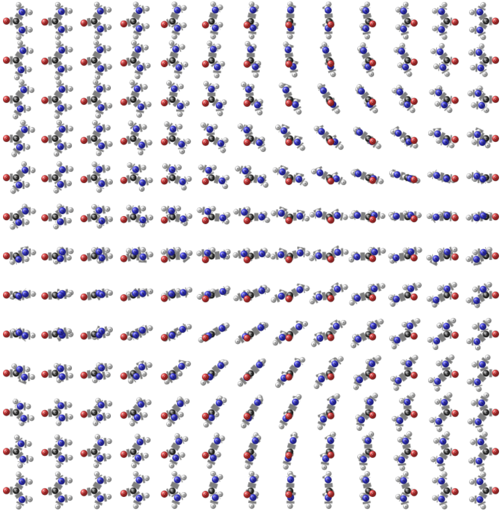
- smooth,
- continuous
- complete

Learning with no symmetry awareness



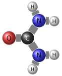
requires hundreds fold augmentation for O(3)

Learning with no symmetry awareness

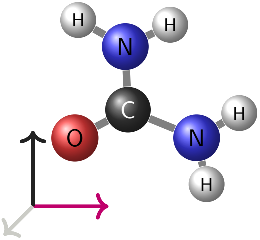


requires hundreds fold augmentation for O(3)

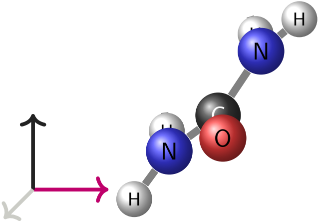
Learning with symmetry-awareness



Symmetry awareness

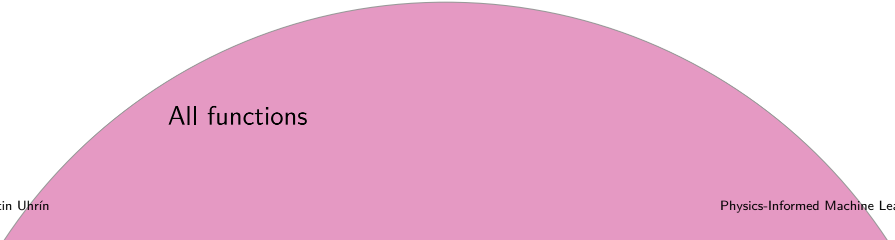


a)

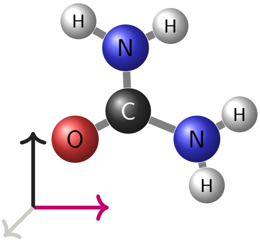


b)

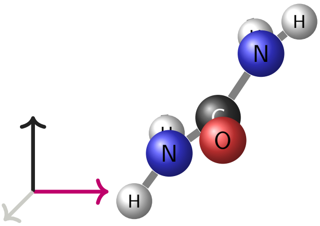
Regular learning model will *see* a) and b) as completely different inputs



Symmetry awareness



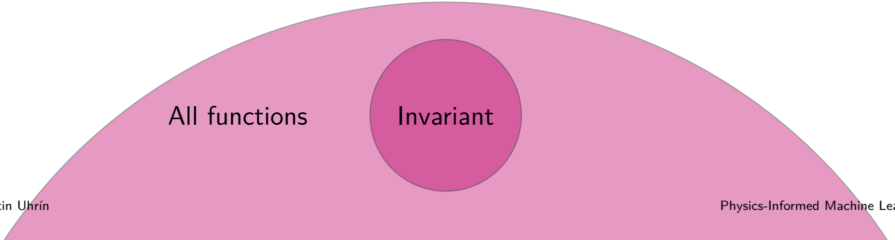
a)



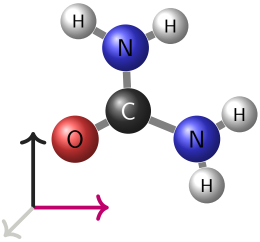
b)

Regular learning model will *see* a) and b) as completely different inputs

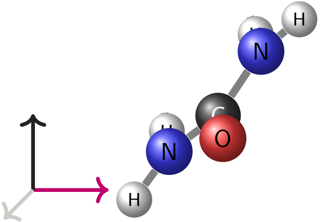
Invariant model will *see* the same inputs.



Symmetry awareness



a)

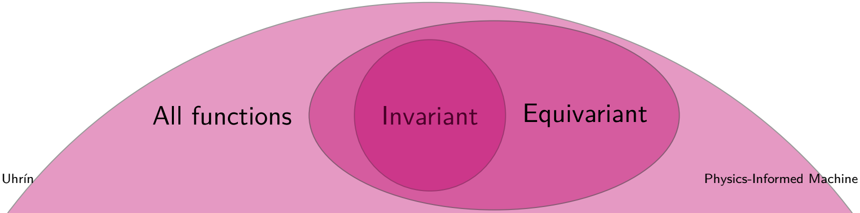


b)

Regular learning model will *see* a) and b) as completely different inputs

Invariant model will *see* the same inputs.

Equivariant model will *see* the same inputs, and carry around gemetric (non-scalar) information.



Lifting the hood on equivariant neural networks

Invariant model

$$f_\theta(D_X[g]x) = f_\theta(x)$$

for the group g.

Lifting the hood on equivariant neural networks

Invariant model

$$f_\theta(D_X[g]x) = f_\theta(x)$$

for the group g. Equivariant model:

$$D_Y[g]f_\theta(x) = f_\theta(D_X[g]x) \quad \forall g \in G, \forall x \in X$$

Natural way of extending ML models to handled non-scalar inputs and outputs (vectors, tensors),

Lifting the hood on equivariant neural networks

Invariant model

$$f_\theta(D_X[g]x) = f_\theta(x)$$

for the group g. Equivariant model:

$$D_Y[g]f_\theta(x) = f_\theta(D_X[g]x) \quad \forall g \in G, \forall x \in X$$

Natural way of extending ML models to handled non-scalar inputs and outputs (vectors, tensors),

Fully-connected layer

$$\phi\left(\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix}^\top \begin{bmatrix} w_1 & \\ w_2 & \\ w_3 & \cdots \\ w_4 & \\ w_5 & \\ w_6 & \end{bmatrix}\right) = \begin{bmatrix} x_3 & \cdots \end{bmatrix}$$

$\phi(x)$ - activation function

Lifting the hood on equivariant neural networks

Equivariant layer

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

Irreducible form:

$l = 0$ scalar - $w_1(x_1x_2 + y_1y_2 + z_1z_2)$

$l = 1$ vector - $w_2 \begin{bmatrix} y_1z_2 - z_1x_2 \\ z_1x_2 - x_1z_2 \\ x_1y_2 - x_2y_1 \end{bmatrix}$

$l = 2$ vector - $w_3 \begin{bmatrix} x_1z_2 + z_1x_2 \\ x_1y_2 + y_1x_2 \\ 2y_1y_2 - x_1x_2 - z_1z_2 \\ y_1z_2 - z_1y_2 \\ z_1z_2 - x_1x_2 \end{bmatrix}$

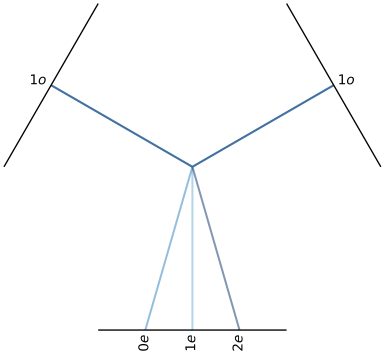
Fully-connected layer

$$\phi\left(\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix}^\top \begin{bmatrix} w_1 & \\ w_2 & \\ w_3 & \cdots \\ w_4 & \\ w_5 & \\ w_6 & \end{bmatrix} \right) = \begin{bmatrix} x_3 & \cdots \end{bmatrix}$$

$\phi(x)$ - activation function

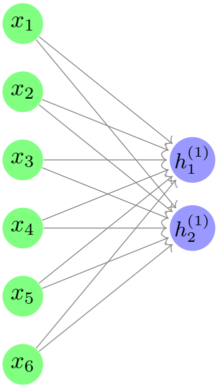
Lifting the hood on equivariant neural networks

Equivariant layer



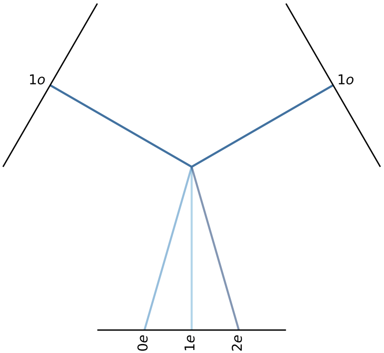
Outputs: $i \times 0e \oplus j \times 1e \oplus k \times 2e$

Fully-connected layer



Lifting the hood on equivariant neural networks

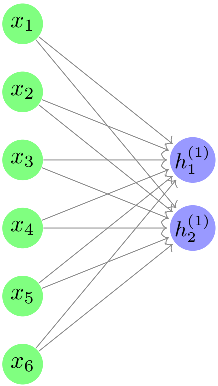
Equivariant layer



Outputs: $i \times 0e \oplus j \times 1e \oplus k \times 2e$

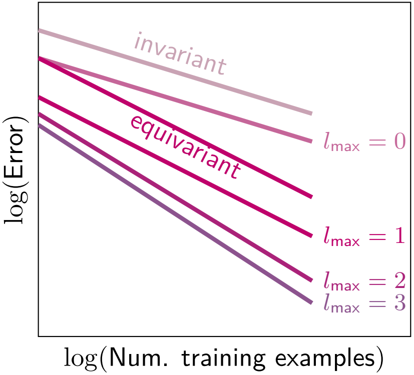
- Maintains symmetry (cannot *rotate* input)
- Minimal amount of parameters

Fully-connected layer



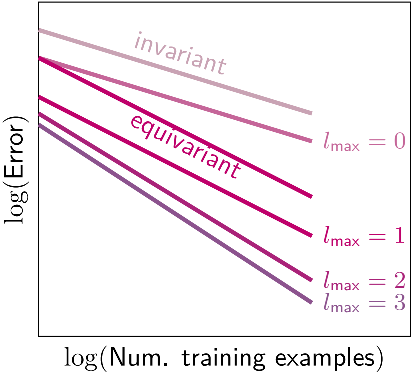
- No constraint on transformation (can scale/rotate/translate inputs)
- Many parameters

Data efficiency



Geiger M., Smidt T., . . . , MU et al., (2022). e3nn/e3nn: 2022-04-13 (0.5.0). Zenodo. https://doi.org/10.5281/zenodo.6459381

Data efficiency



System		NequIP[a)]	NequIP[b)]	NequIP[c)]	DeepMD
Liquid Water	Energy	-	1.6	1.7	1.0
	Forces	11.9	49.4	11.6	40.4
Ice Ih (b)	Energy	-	2.5	4.3	0.7
	Forces	10.2	55.8	9.9	43.3
Ice Ih (c)	Energy	-	3.9	10.2	0.7
	Forces	12.0	27.7	11.7	26.8
Ice Ih (d)	Energy	-	2.6	12.7	0.8
	Forces	9.8	23.2	9.5	25.4

NequIP model trained on $< 0.1\%$ of DeepMD model

S. Batzner et al., Nature Communications **13**, 2453 (2022)

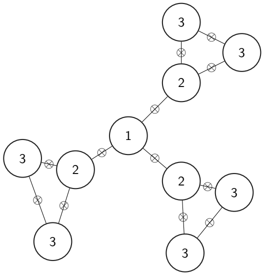
Geiger M., Smidt T., . . . , MU et al., (2022). e3nn/e3nn: 2022-04-13 (0.5.0). Zenodo. https://doi.org/10.5281/zenodo.6459381

python ecosystem for development of equivariant learning

Complete ecosystem built in JAX and `e3nn-jax`:

tensorial

- Native support for equivariant graph convolutions
- Complete flexibility over input and output types:
  - per-node, per-edge or global
  - arbitrary node attributes (scalars, tensors)
- Implemented in JAX: 4-20x speedup over pytorch



reax

- pytorch lightning-like model training

e3md

- Built in implementations of
  - NequIP
  - Allegro
  - MACE

```
def f(x):
return ** 2

# Evaluate gradient
jax.grad(f)(5.)

# Multiple inputs, scalar output
jax.grad(dft_energy(atom_positions))

# Multiple inputs, multiple outputs
jax.jacfwd(get_bands(atom_positions))
```

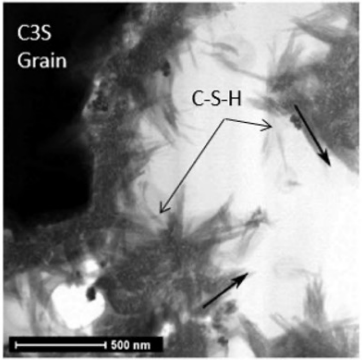
Electric and magnetic response

Calcium Silicate Hydrates

Laura Mismetti
IBM

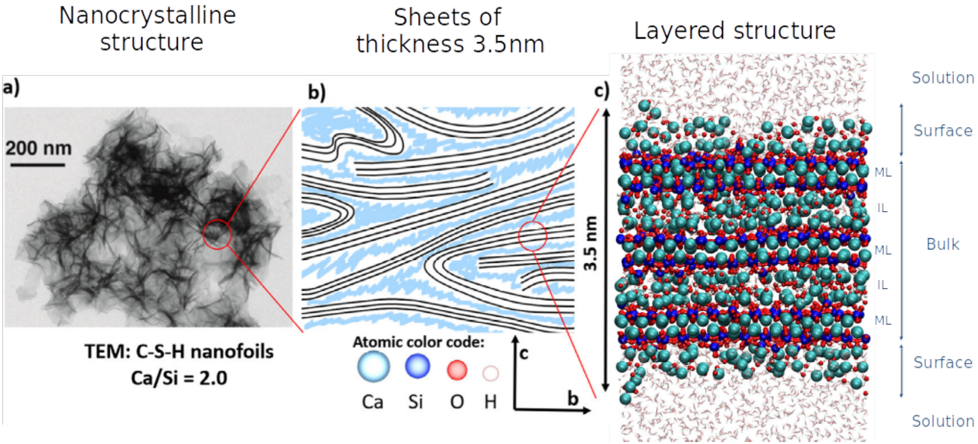
Ziga Casar
Princeton

- Main product of the hydration process of cement 50% hardened cement paste
- Its nucleation and growth are the controlling steps of the early hydration reaction
- Forms a nanoporous network which controls the ion transport (Cl-) and consequently influences the service-life of concrete structures



CSH micrograph

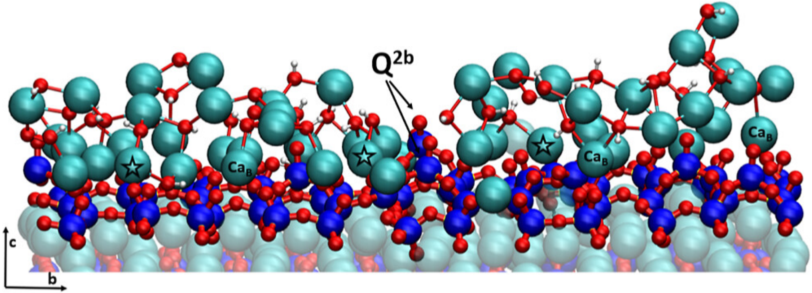
CSH Nanofoil Structure



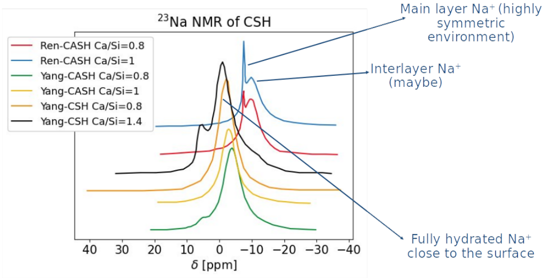
Interactions at CSH Surface

- Ions in solution (Na+, Cl-, . . .)
- Interactions at the interface could affect cohesion and long-term durability of cement-based materials
- High Na+ concentration: it seems it removes Ca+ and OH-, lowering the surface charge

- Nuclear Magnetic Resonance (NMR) to get info about the local environment
- Insights on Na+ incorporation sites



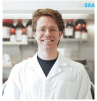
CSH Experimental ^{23}Na NMR



- Nuclear Magnetic Resonance (NMR) to get info about the local environment
- Sodium incorporation sites not yet understood

Goal: Deepen the understanding of sodium incorporation in CSH exploiting NMR data (simulations + experiments)

Magnetic response



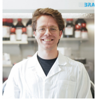
Mattia Ragni
UGA

$$E = U - \vec{\mu} \cdot \vec{B}$$

$\vec{\mu}$ - magnetic dipole moment

$\vec{B}$ - external magnetic field

Magnetic response



Mattia Ragni
UGA

$$E = U - \vec{\mu} \cdot \vec{B}$$

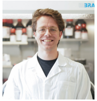
$\vec{\mu}$ - magnetic dipole moment

$\vec{B}$ - external magnetic field

NMR shielding tensor

$$\sigma^I_{\alpha\beta} = \frac{\partial^2 E}{\partial B_\alpha \partial \mu^I_\beta} = \vec{B} \otimes \vec{\mu}^I$$

Magnetic response



Mattia Ragni
UGA

NMR shielding tensor

$$\sigma^I_{\alpha\beta} = \frac{\partial^2 E}{\partial B_\alpha \partial \mu^I_\beta} = \vec{B} \otimes \vec{\mu}^I$$

rank 2 non-symmetric tensor, irreps $0e \oplus 1e \oplus 2e$

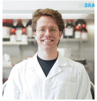
$$E = U - \vec{\mu} \cdot \vec{B}$$

$\vec{\mu}$ - magnetic dipole moment

$\vec{B}$ - external magnetic field

$$\sigma_{\alpha\beta} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

Magnetic response



Mattia Ragni
UGA

$$E = U - \vec{\mu} \cdot \vec{B}$$

$\vec{\mu}$ - magnetic dipole moment

$\vec{B}$ - external magnetic field

NMR shielding tensor

$$\sigma^I_{\alpha\beta} = \frac{\partial^2 E}{\partial B_\alpha \partial \mu^I_\beta} = \vec{B} \otimes \vec{\mu}^I$$

$$\sigma_{\alpha\beta} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

rank 2 non-symmetric tensor, irreps $0e \oplus 1e \oplus 2e$

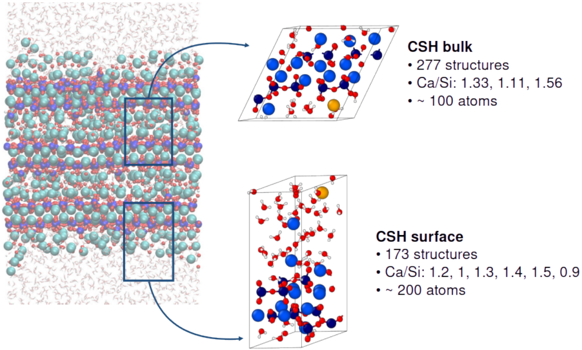
Model - GNN:

- R^I
- Z^I

$\longrightarrow$ `tensorial` EGNN $\longrightarrow$

- σ^I

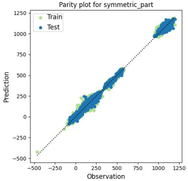
Dataset generation

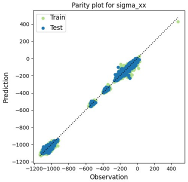


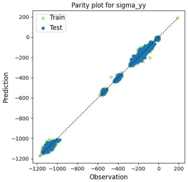
Bulk and surface CSH structures with water

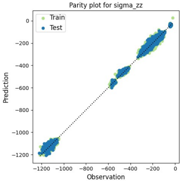
- Representative CSH bulk and surface (containing Na+) generated with the brick model
- Pre-relaxation with empirical FF + DFT relaxation
- QE+GIPAW method to calculate NMR tensor (3x3) for every atom

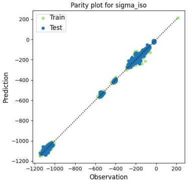
Training results

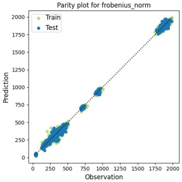


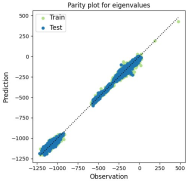












Electric response

Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Electric response

Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

Born effective charge tensor

$$\mathcal{Z}^{I}_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta} = \vec{\epsilon} \otimes \vec{\tau}^I$$

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Electric response


Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

Born effective charge tensor

$$\mathcal{Z}^I_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta} = \vec{\epsilon} \otimes \vec{\tau}^I$$

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Raman tensor - Can be calculated via derivatives of the enthaly:

$$\chi^I_{\alpha\beta\gamma} = -\frac{1}{\Omega} \frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma}$$
$$= -\frac{1}{\Omega} \vec{\epsilon} \otimes \vec{\epsilon} \otimes \vec{\tau}^I$$

Electric response


Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Born effective charge tensor

$$\mathcal{Z}^I_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta} = \vec{\epsilon} \otimes \vec{\tau}^I$$

rank 2 non-symmetric tensor, irreps $0e \oplus 1e \oplus 2e$

Raman tensor - Can be calculated via derivatives of the enthaly:

$$\chi^I_{\alpha\beta\gamma} = -\frac{1}{\Omega} \frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma}$$
$$= -\frac{1}{\Omega} \vec{\epsilon} \otimes \vec{\epsilon} \otimes \vec{\tau}^I$$

Electric response

Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Born effective charge tensor

$$\mathcal{Z}^I_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta} = \vec{\epsilon} \otimes \vec{\tau}^I$$

rank 2 non-symmetric tensor, irreps $0e \oplus 1e \oplus 2e$

Raman tensor - Can be calculated via derivatives of the enthaly:

$$\chi^I_{\alpha\beta\gamma} = -\frac{1}{\Omega} \frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma} = -\frac{1}{\Omega} \vec{\epsilon} \otimes \vec{\epsilon} \otimes \vec{\tau}^I$$

rank 3 tensor symmetric in α, β irreps $2 \times 1o \oplus 2o \oplus 3o$

Electric response


Lorenzo Bastonero
Uni Bremen

Alessandro D'Urso
EPFL

$$E = U - \vec{P} \cdot \vec{\epsilon}$$

$\vec{P}$ - polarization

$\vec{\epsilon}$ - external electric field

Born effective charge tensor

$$\mathcal{Z}^I_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta} = \vec{\epsilon} \otimes \vec{\tau}^I$$

rank 2 non-symmetric tensor, irreps $0e \oplus 1e \oplus 2e$

Raman tensor - Can be calculated via derivatives of the enthaly:

$$\chi^I_{\alpha\beta\gamma} = -\frac{1}{\Omega} \frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma}$$
$$= -\frac{1}{\Omega} \vec{\epsilon} \otimes \vec{\epsilon} \otimes \vec{\tau}^I$$

rank 3 tensor symmetric in α, β irreps $2 \times 1o \oplus 2o \oplus 3o$

Model - double headed EGNN:

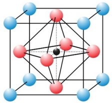
- R^I
- Z^I

$\longrightarrow$ $\boxed{\texttt{tensorial}\ \text{EGNN}}$ $\longrightarrow$

- $\mathcal{Z}^I$
- χ^I

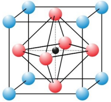
Electric response results

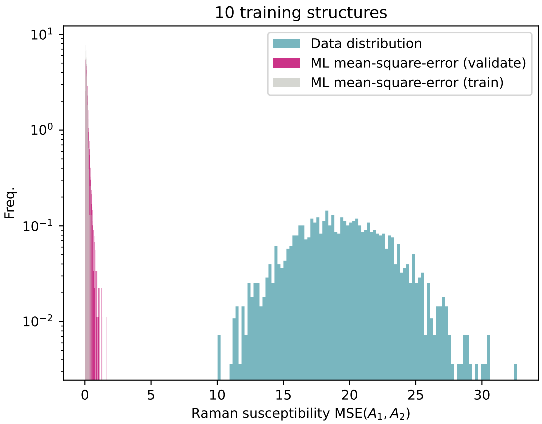
50 structures $BaTiO_3$, 135 atoms, at 400K with random displacements from equilibrium.



Electric response results

50 structures $BaTiO_3$, 135 atoms, at 400K with random displacements from equilibrium.

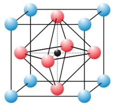


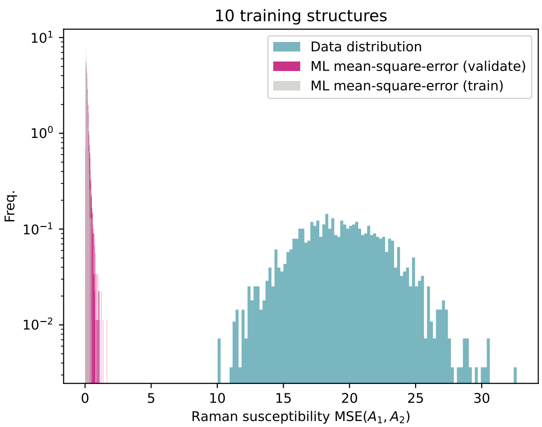


Raman susceptibilities

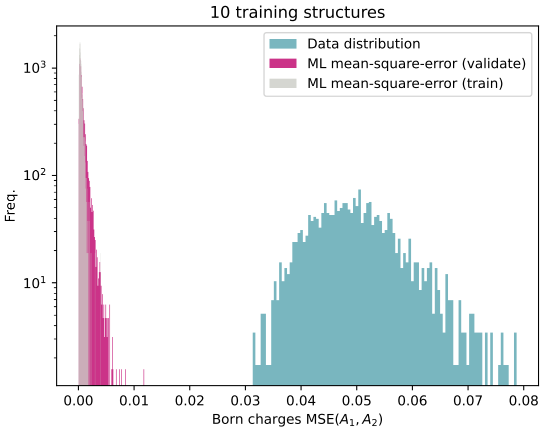
Electric response results

50 structures $BaTiO_3$, 135 atoms, at 400K with random displacements from equilibrium.





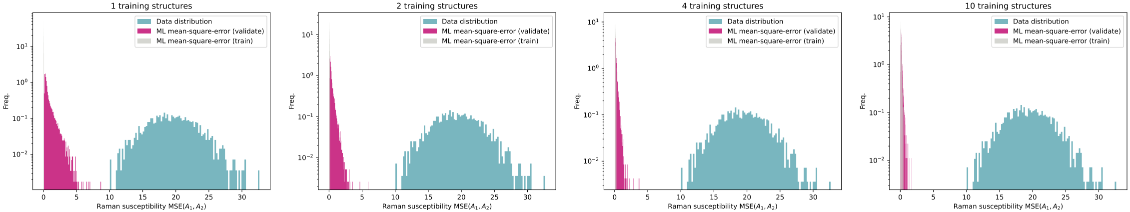
Raman susceptibilities



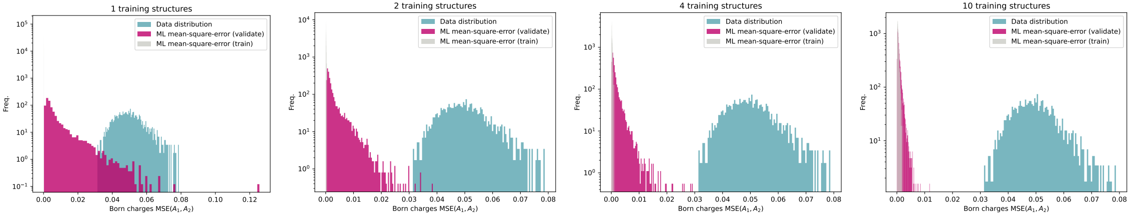
Born effective charges

Accuracy as function of training set size

Raman susceptibilities



Born effective charges



Injecting more physics

Born effective charge tensor

$$\mathcal{Z}^{I}_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta}$$

Raman tensor

$$\chi^{I}_{\alpha\beta\gamma} = -\frac{1}{\Omega}\frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma}$$

Injecting more physics

Born effective charge tensor

$$\mathcal{Z}^I_{\alpha\beta} = \frac{\partial^2 E}{\partial \epsilon_\alpha \partial \tau^I_\beta}$$

Raman tensor

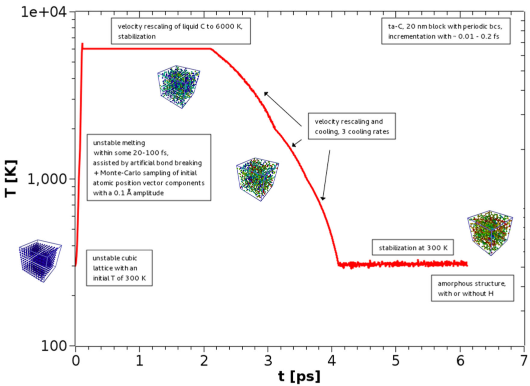
$$\chi^I_{\alpha\beta\gamma} = -\frac{1}{\Omega}\frac{\partial^3 E}{\partial \epsilon_\alpha \partial \epsilon_\beta \partial \tau^I_\gamma}$$

Model

$$\begin{matrix} R^I \\ Z^I \\ \vec{\epsilon} \end{matrix} \longrightarrow \boxed{\begin{matrix}\texttt{tensorial} \\ \text{EGNN}\end{matrix}} \longrightarrow E \xrightarrow{\partial/\partial} \begin{matrix} \mathcal{Z}^I \\ \chi^I \end{matrix}$$

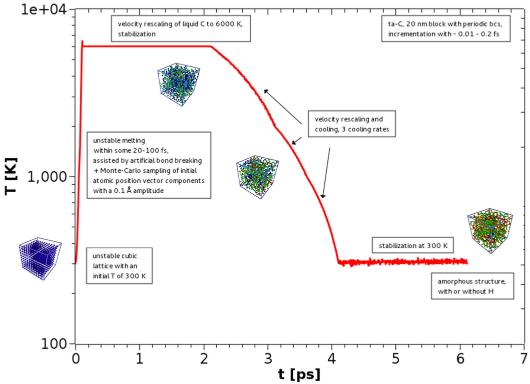
Generative models of atomic configurations

Example: generating atomic configurations in amoprhous solids



Timo Hakala, Kenneth Holmberg and Anssi Laukkanen. Lubricants. 9. 30. (2021).

Example: generating atomic configurations in amoprhous solids

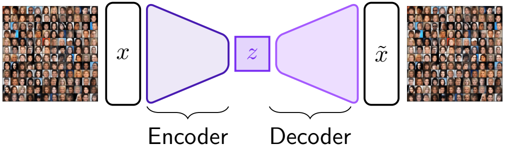


Given an example structure(s), can we teach a generative machine learning model to generate novel examples, bypassing the need for further molecular dynamics?

Timo Hakala, Kenneth Holmberg and Anssi Laukkanen. Lubricants. 9. 30. (2021).

The Variational Autoencoder

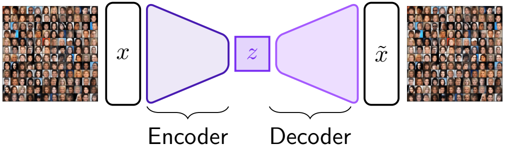
The autoencoder



$$\mathcal{L} = (x - \tilde{x})^2$$

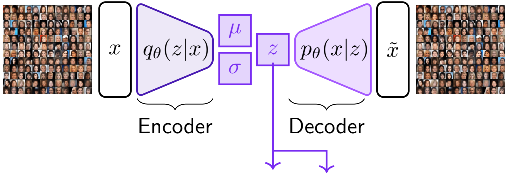
The Variational Autoencoder

The autoencoder

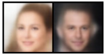


$$\mathcal{L} = (x - \tilde{x})^2$$

The *variational* autoencoder



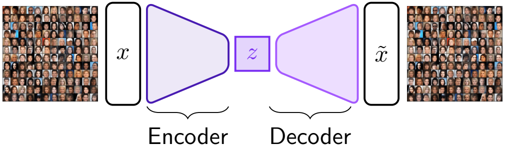
$$z \sim \mathcal{N}(\mu, \sigma)$$



$$\mathcal{L} = (x - \tilde{x})^2 + \sum_j KL(q_j(z|x)||p(z))$$

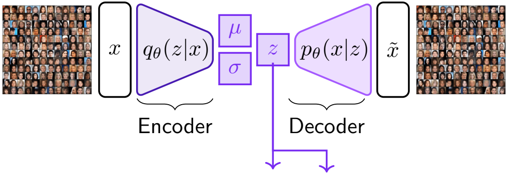
The Variational Autoencoder

The autoencoder

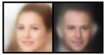


$$\mathcal{L} = (x - \tilde{x})^2$$

The *variational* autoencoder



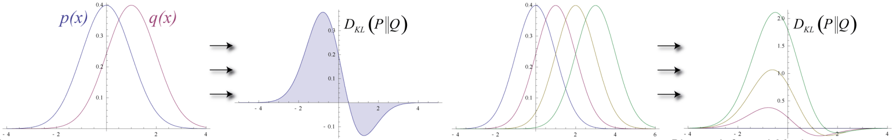
$$z \sim \mathcal{N}(\mu, \sigma)$$



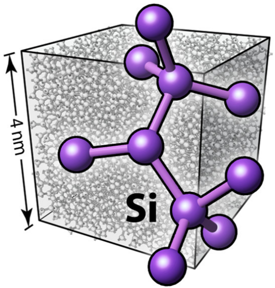
$$\mathcal{L} = (x - \tilde{x})^2 + \sum_j KL(q_j(z|x)||p(z))$$

Kullback-Leibler divergence

$$D_{\mathsf{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) \, dx$$



Symmetry-aware representation of local atomic environments



V. L. Deringer et al., Journal of Physical Chemistry Letters **9**, 2879–2885 (2018)

Direct representation

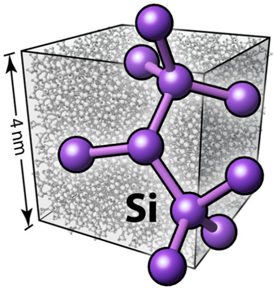
$$x = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{bmatrix}$$

Not a suitable input to a learning model. Consider

$$x' = xQ$$

where Q is a rotation matrix.

Symmetry-aware representation of local atomic environments



V. L. Deringer et al., Journal of Physical Chemistry Letters **9**, 2879–2885 (2018)

Direct representation

$$x = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{bmatrix}$$

Not a suitable input to a learning model. Consider

$$x' = xQ$$

where Q is a rotation matrix.

Symmetry-aware representation

$$G = \begin{bmatrix} x^1 \cdot x^1 & \cdots & x^1 \cdot x^N \\ \vdots & \ddots & \vdots \\ x^N \cdot x^1 & \cdots & x^N \cdot x^N \end{bmatrix} = \begin{bmatrix} \|x^1\|^2 & \cdots & \|x^1\|\|x^N\| \cos\theta_{1N} \\ \vdots & \ddots & \vdots \\ \|x^N\|\|x^1\| \cos\theta_{N1} & \cdots & \|x^N\|^2 \end{bmatrix}$$

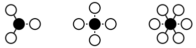
This representation is rotationally invariant.

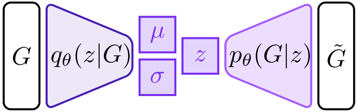
$$G = xx^T = (xQ)(xQ)^T = xQQ^Tx^T = xIx^T$$

Variational autoencoder for atomic motifs

Training

1. For each atom in unit cell, extract local atomic environment up to r_{cut}. Keeps closest n atoms



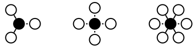


minimise
$\mathcal{L} = (G - \tilde{G})^2 + \sum_j KL(q_j(z|G)||p(z))$

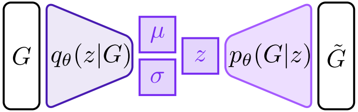
Variational autoencoder for atomic motifs

Training

1. For each atom in unit cell, extract local atomic environment up to r_{cut}. Keeps closest n atoms



2. Calculate Gram matrix $x^i \cdot x^j$, keep upper triangular part, $j \geq i$

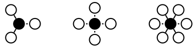


minimise
$\mathcal{L} = (G - \tilde{G})^2 + \sum_j KL(q_j(z|G)||p(z))$

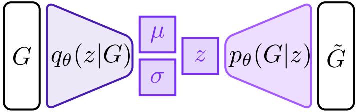
Variational autoencoder for atomic motifs

Training

1. For each atom in unit cell, extract local atomic environment up to r_{cut}. Keeps closest n atoms



2. Calculate Gram matrix $x^i \cdot x^j$, keep upper triangular part, $j \geq i$
3. Generate permutation copies of atom labels i (data augmentation) e.g. $[1,2,3],[1,3,2],[2,1,3]$, etc

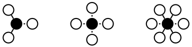


minimise
$\mathcal{L} = (G - \tilde{G})^2 + \sum_j KL(q_j(z|G)||p(z))$

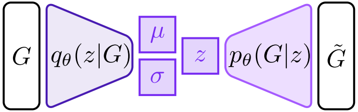
Variational autoencoder for atomic motifs

Training

1. For each atom in unit cell, extract local atomic environment up to r_{cut}. Keeps closest n atoms



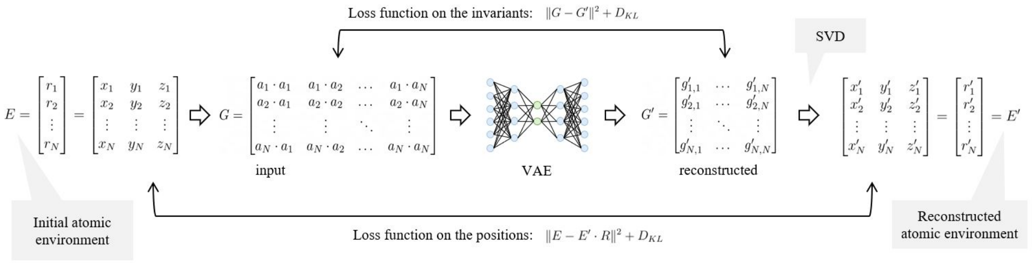
2. Calculate Gram matrix $x^i \cdot x^j$, keep upper triangular part, $j \geq i$
3. Generate permutation copies of atom labels i (data augmentation) e.g. $[1, 2, 3], [1, 3, 2], [2, 1, 3]$, etc
4. Train VAE using gradient-based optimisation



minimise
$\mathcal{L} = (G - \tilde{G})^2 + \sum_j KL(q_j(z|G)||p(z))$

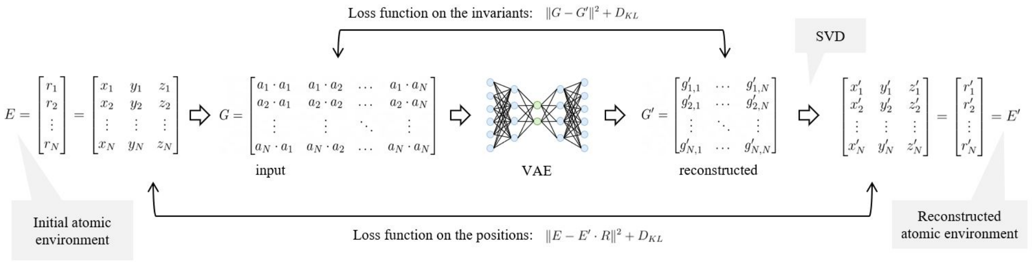
Training and generating

Training



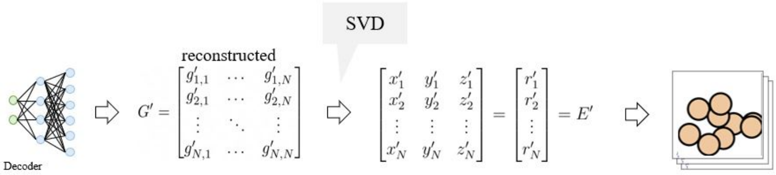
Training and generating

Training



Generating

Draw n_Z samples from $\mathcal{N}(0, 1)$



Variational autoencoder for atomic motifs

All tests performed in amorphous Si, 512 atom unit cells generated using ML potential trained on DFT[1].

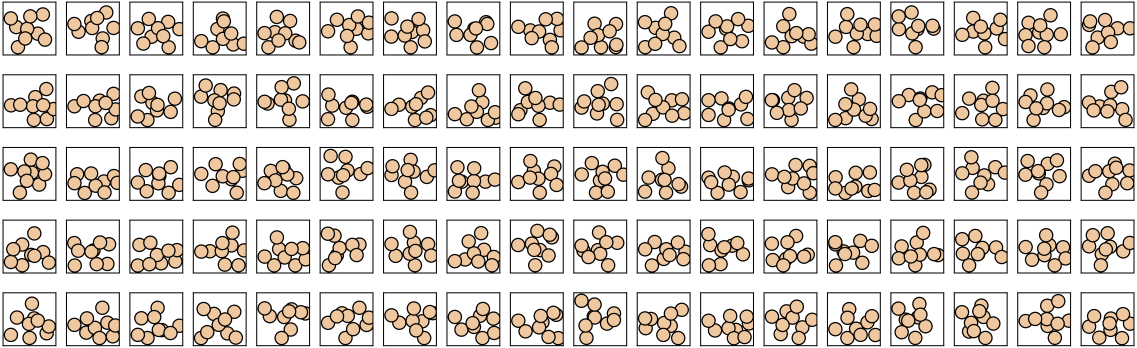
- Radial cutoff: 4 Å
- 8 atoms per environment
- latent space: 8 neurons (compared to $3n - 6 = 18$ DoFs)
- Encoder architecture 36-28-18-8 with tanh activations

Data normalisation

- $G'_{ii} = (G_{ii} - \mu_{\text{diag}})/\sigma_{\text{diag}}$
- $G'_{ij} = (G_{ij} - \mu_{\text{off-diag}})/\sigma_{\text{off-diag}}, i < j$

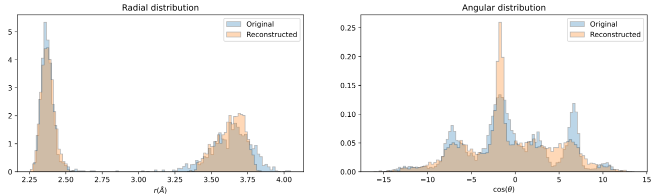
V. L. Deringer et al., Journal of Physical Chemistry Letters **9**, 2879–2885 (2018)

Example generated environments



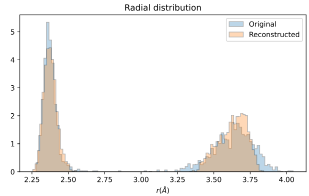
Model performance

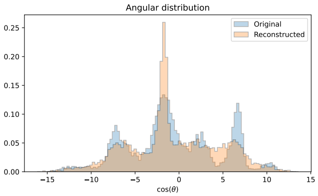
Training reconstruction

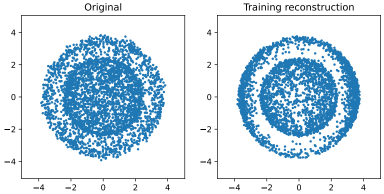


Model performance

Training reconstruction

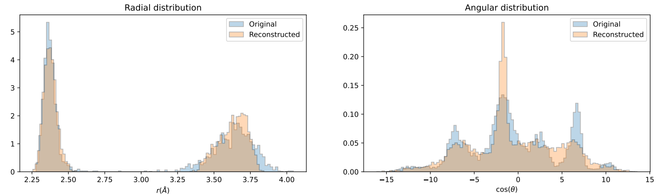


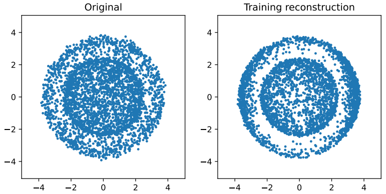




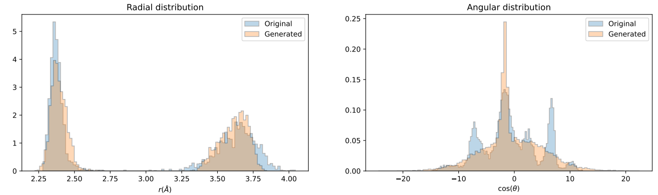
Model performance

Training reconstruction



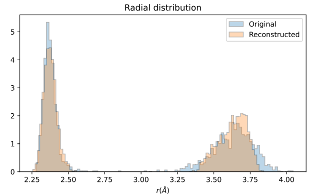


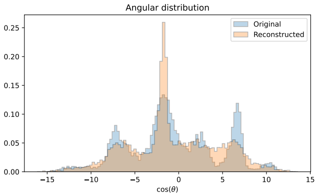
Generated samples

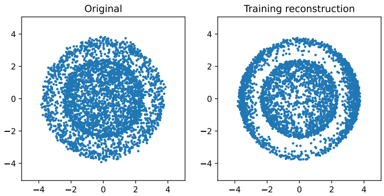


Model performance

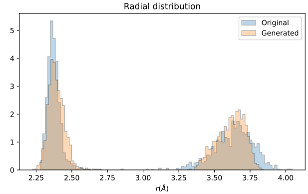
Training reconstruction

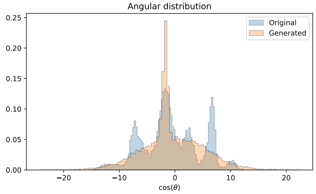


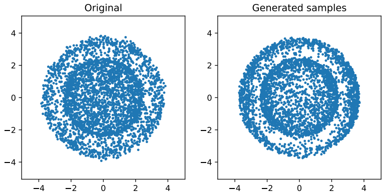




Generated samples

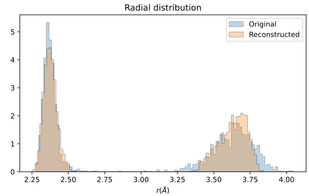


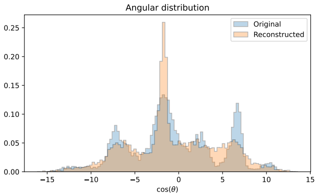


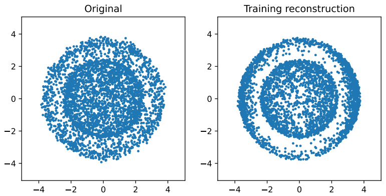


Model performance

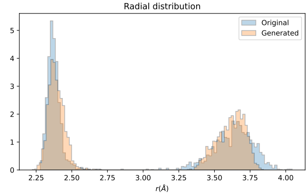
Training reconstruction

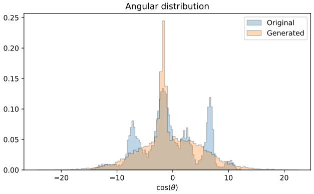


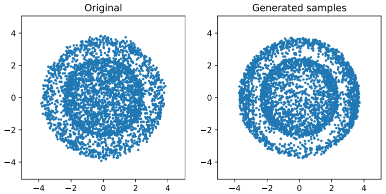


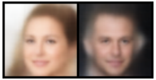


Generated samples

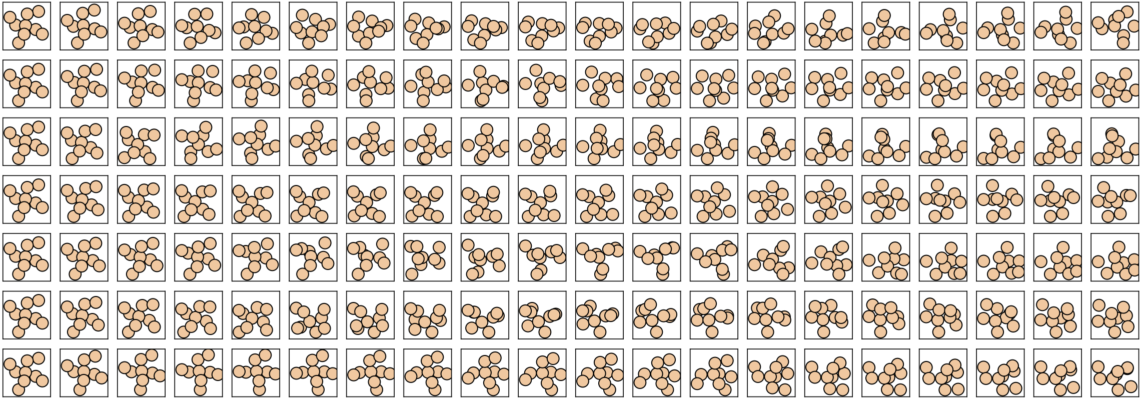




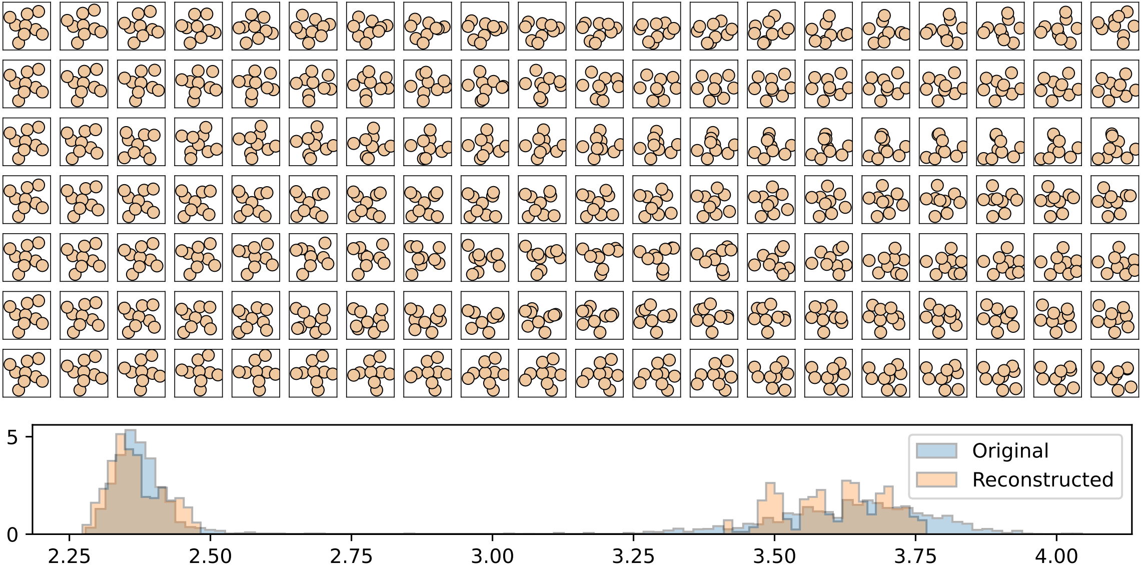




Interpolating between motifs



Interpolating between motifs



Atom infilling: Building complete unit cells

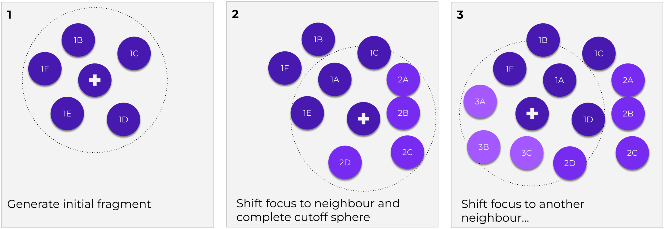
Image inpainting

T. F. Chan and J. Shen, Communications on Pure and Applied Mathematics **58**, 579–619 (2005)

Atom infilling: Building complete unit cells

Image inpainting

Environment infilling



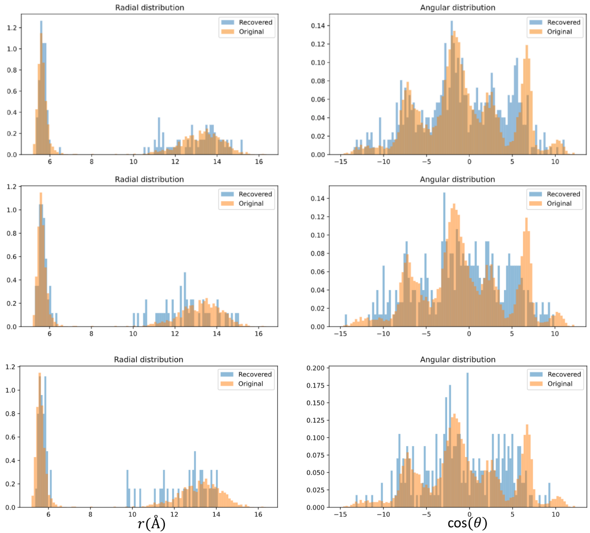
T. F. Chan and J. Shen, Communications on Pure and Applied Mathematics **58**, 579–619 (2005)

Atom infilling results

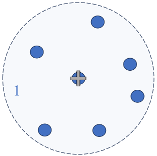
Fixed atoms: 7

Fixed atoms: 6

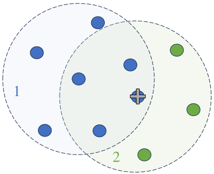
Fixed atoms: 5



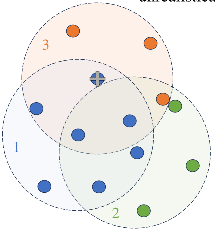
Atom infilling results



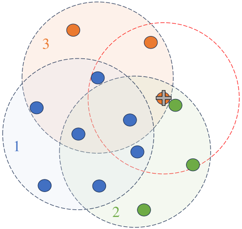
1. Initial VAE generated atomic environment



2. Filling missing atoms in the second environment



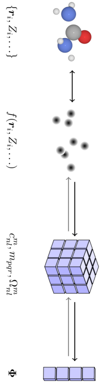
3. Filling missing atoms in the third environment



4. Environment with unrealistically close atoms

Next steps

Complete, invertible, representations

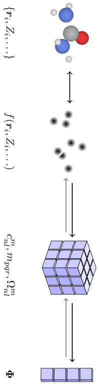


Permutational invariance

M. Uhrin, Physical Review B **104**, 144110 (2021)

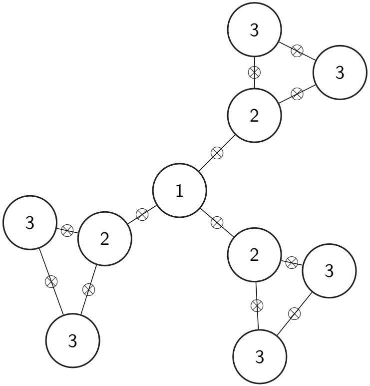
Next steps

Complete, invertible, representations



Permutational invariance

Equivariant graph neural networks

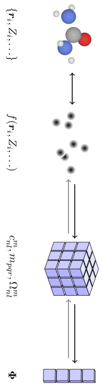


Long(er) sightedness

M. Uhrin, Physical Review B **104**, 144110 (2021)

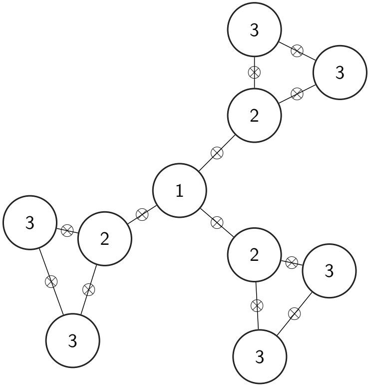
Next steps

Complete, invertible, representations



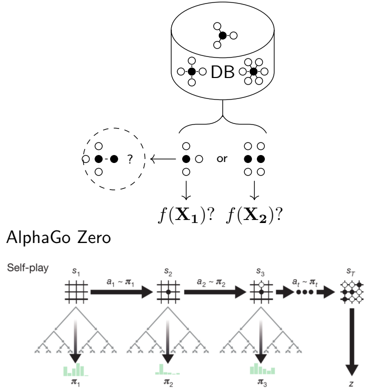
Permutational invariance

Equivariant graph neural networks



Long(er) sightedness

Reinforcement learning



Property conditioning

M. Uhrin, Physical Review B **104**, 144110 (2021)

Summary

Conclusion

Euclidean neural networks naturally encode the assumption that atomic systems exist in 3D Euclidean space

This inductive bias can make them more robust, data-efficient and accurate when predicting properties

GNNs give us a flexible way to express model inputs and outputs and perform multi-target learning

Conclusion

Euclidean neural networks naturally encode the assumption that atomic systems exist in 3D Euclidean space

This inductive bias can make them more robust, data-efficient and accurate when predicting properties

GNNs give us a flexible way to express model inputs and outputs and perform multi-target learning

Where is this going?

Multiscale approaches to bridge time and length scales (coarse graining, etc)

Conclusion

Euclidean neural networks naturally encode the assumption that atomic systems exist in 3D Euclidean space

This inductive bias can make them more robust, data-efficient and accurate when predicting properties

GNNs give us a flexible way to express model inputs and outputs and perform multi-target learning

Where is this going?

Multiscale approaches to bridge time and length scales (coarse graining, etc)

Coupling to experiment i) using ML to help with interpretation, ii) using experiment to train ML models

Generative models that can output atomic configurations with high probability of having the desired properties

Acknowledgements

Nicola Marzari
EPFL

Tess Smidt
MIT

Mario Geiger
Nvidia

Swiss National Science Foundation

MARVEL
NATIONAL CENTRE OF COMPETENCE IN RESEARCH

DOME 4.0

MIAI Grenoble Alpes
Multidisciplinary Institute In Artificial Intelligence

cost
EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

DAEMON

Tutorial time!

In the tutorial you will see how to predict electric field response properties using equivariant graph neural networks.

- Can run on cluster or on Google Colab (recommended - see links in github)
- Remember to change kernel to T5

Backup

Accuracy as function of num. training iterations