Introduction to GNNs as interatomic potentials

Ekin Doğuş Çubuk
cubuk@google.com

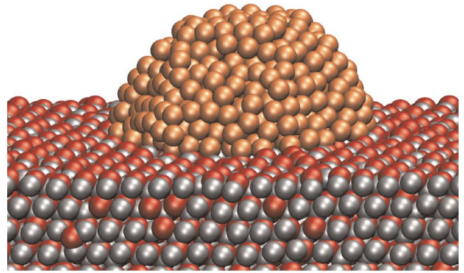
Outline

- Why GNNs for interatomic potentials?
- JAX-MD as one tool for connecting atomistic simulations to GNNs.
- Future work
- JAX-MD tutorials

Predicting energy from atomic positions

Atomic positions → Energy

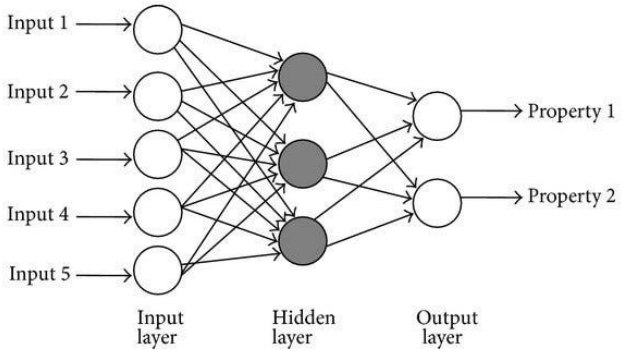
R^{Nx3} → R^{1}

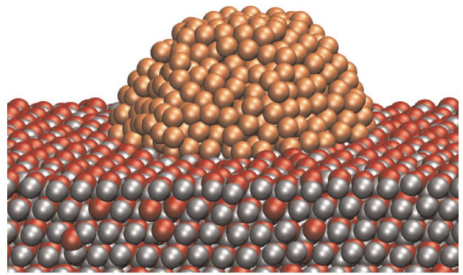


Predicting energy from atomic positions

Atomic positions → Energy

R^{Nx3} → R^{1}

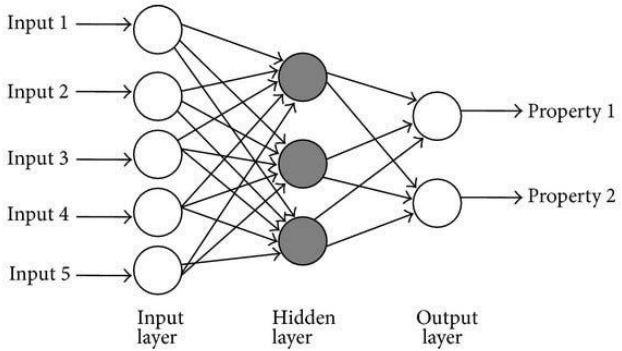




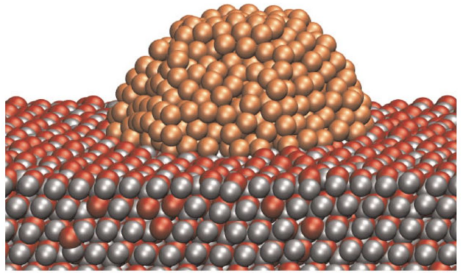
Issues with using the simplest Multi Layer Perceptron

Atomic positions → Energy

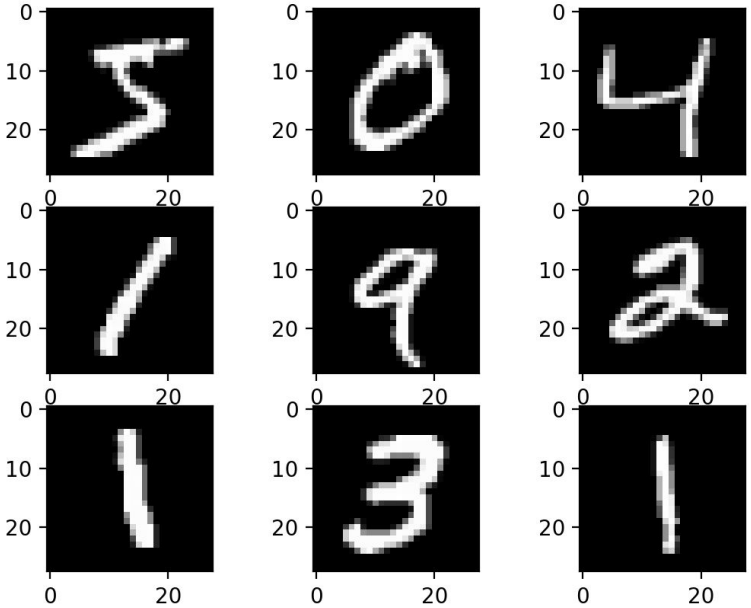
R^{Nx3} → R^1



- Invariances:
  - Rotations
  - Translations
  - Permutations
- Number of atoms
- Locality / weight-tying

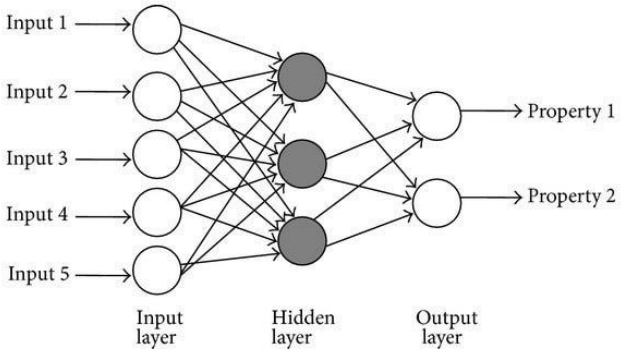


Issues with using the simplest Multi Layer Perceptron (MNIST)



Pixels → Number

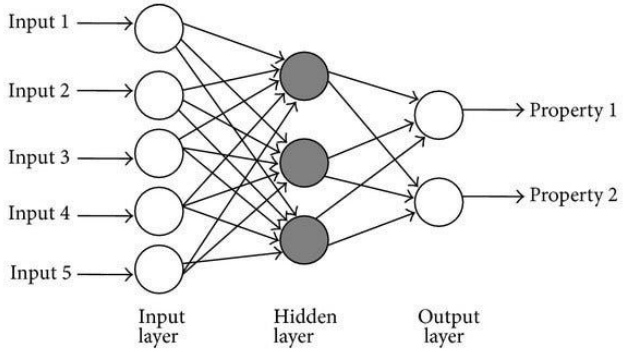
R^{28x28} → Z^{10}



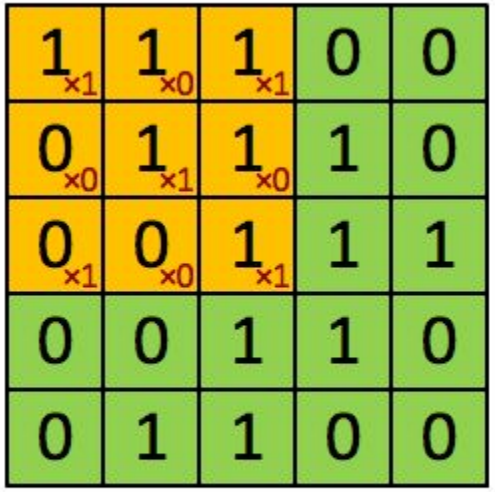
Issues with using the simplest Multi Layer Perceptron (ImageNet)

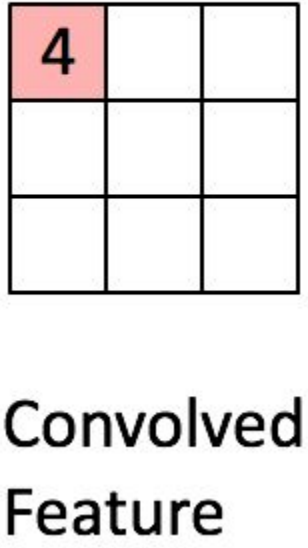
Pixels → Object label

$R^{224x224}$ → Z^{1000}



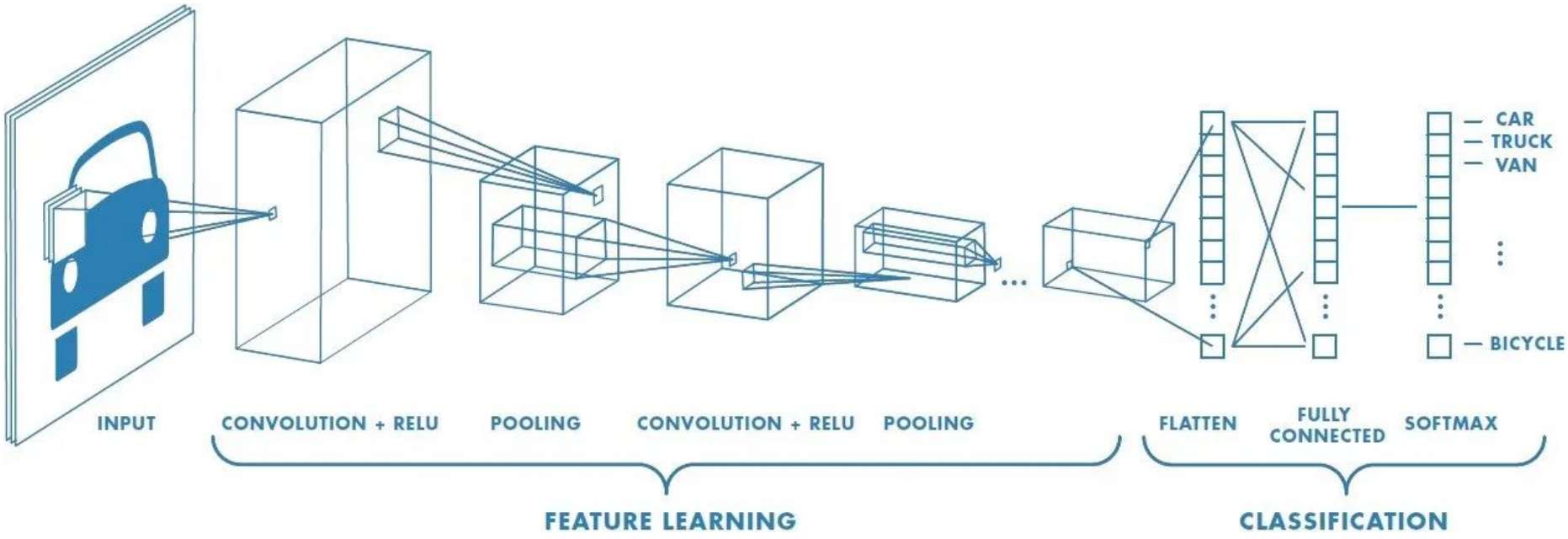
Convolutional networks



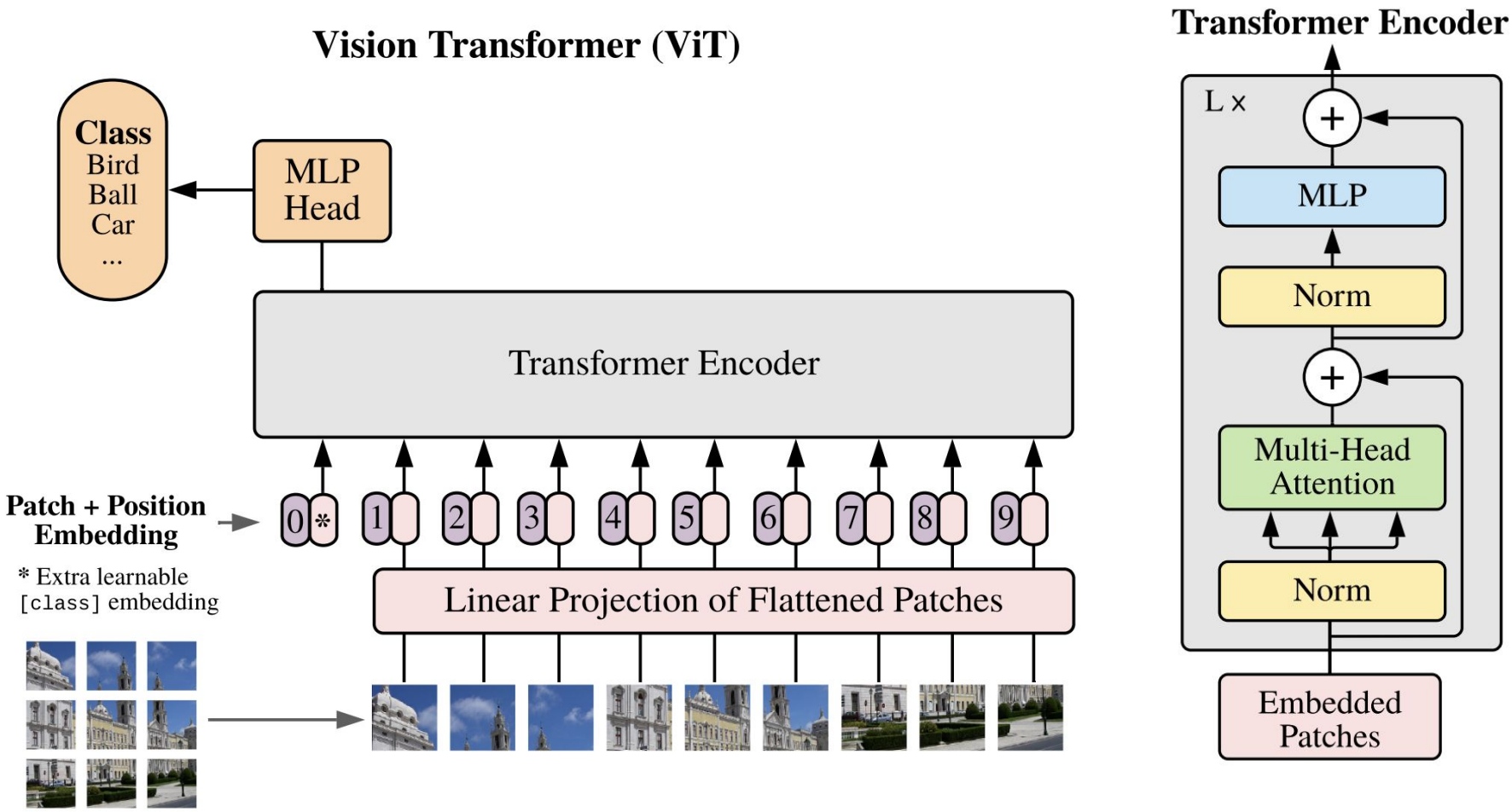


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

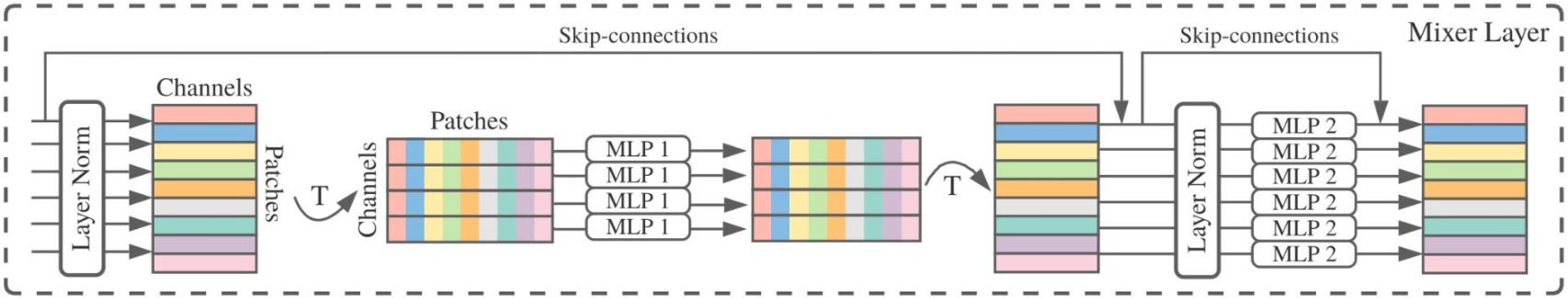
Convolutional networks

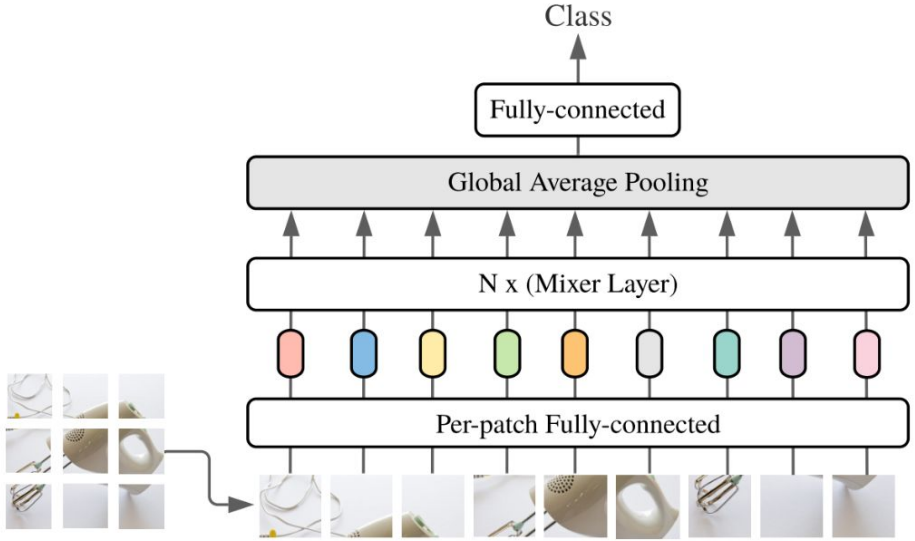


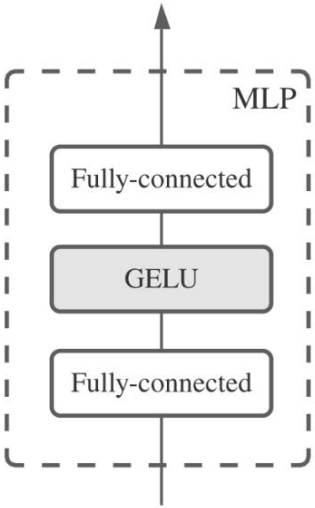
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 * 1 2 3 4 5 6 7 8 9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

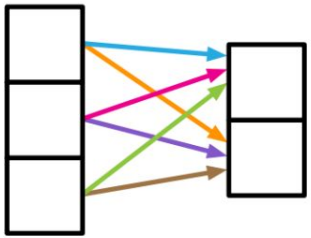
MLP MIXER


Skip-connections
Skip-connections
Mixer Layer
Channels
Layer Norm
Patches
T
Patches
Channels
MLP 1
MLP 1
MLP 1
MLP 1
T
Layer Norm
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2

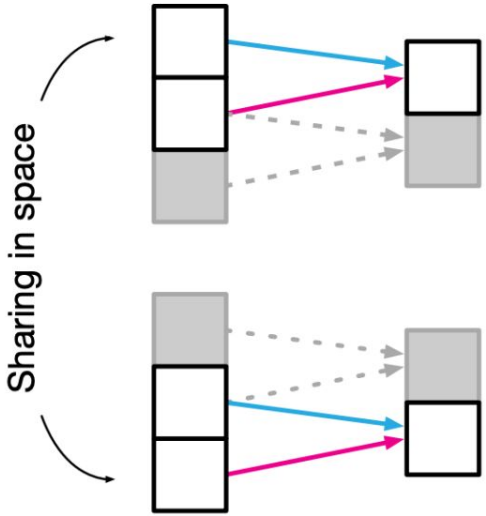

Class
Fully-connected
Global Average Pooling
N x (Mixer Layer)
Per-patch Fully-connected


MLP
Fully-connected
GELU
Fully-connected

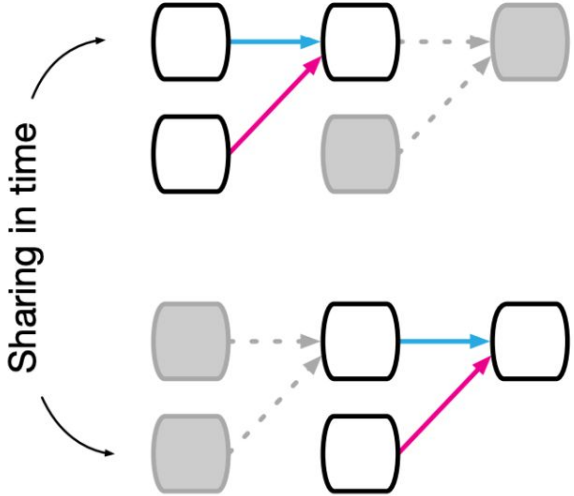
Weight-tying in different frameworks



(a) Fully connected



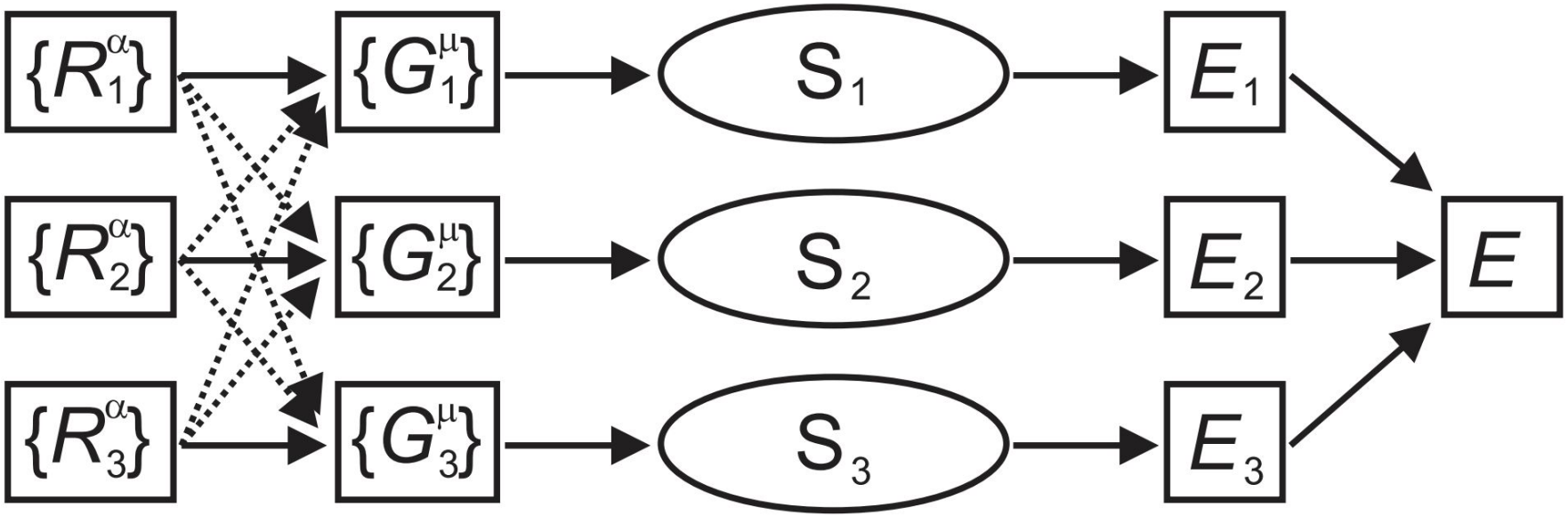
(b) Convolutional



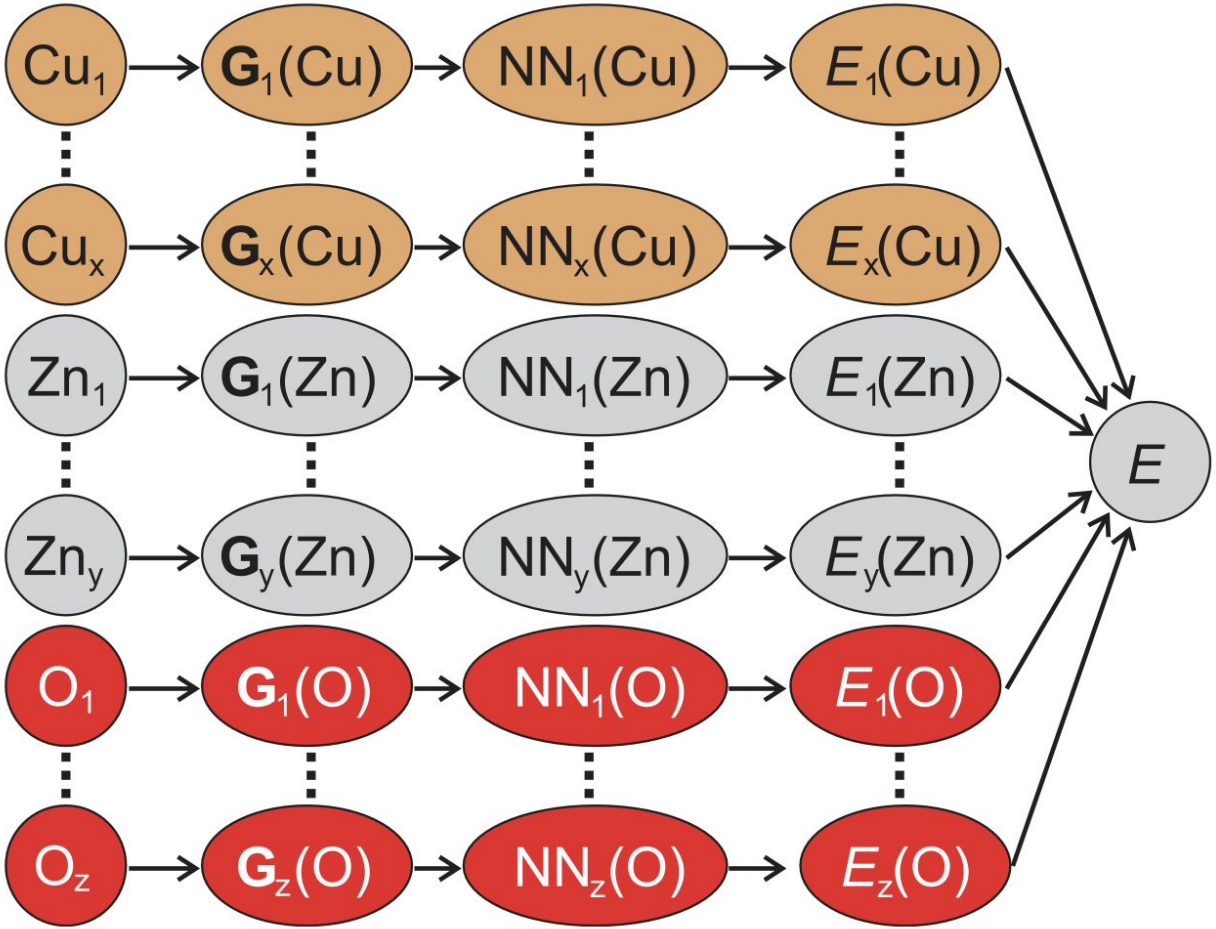
(c) Recurrent

Battaglia et al. arXiv:1806.01261

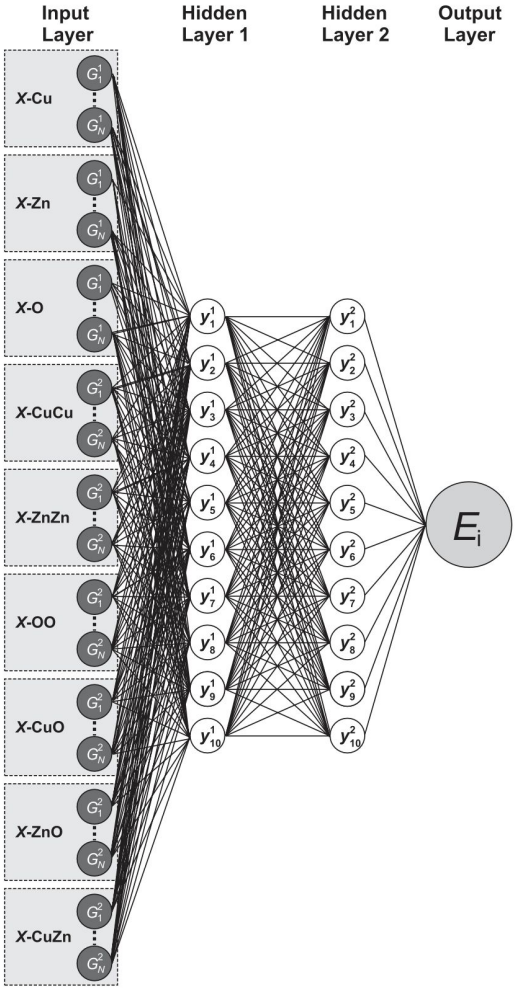
Behler-Parrinello Network

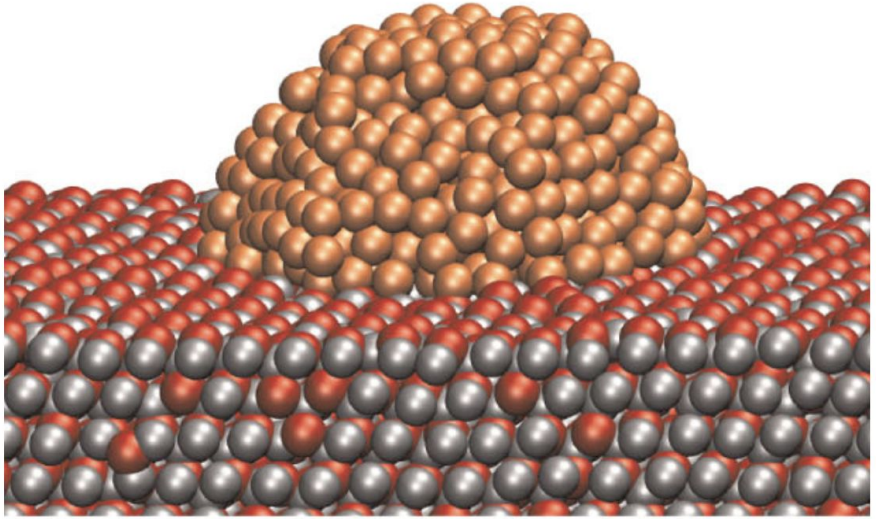


Behler, Parrinello, PRL 2007



Artrith, Hiller, Behler, PSS 2012

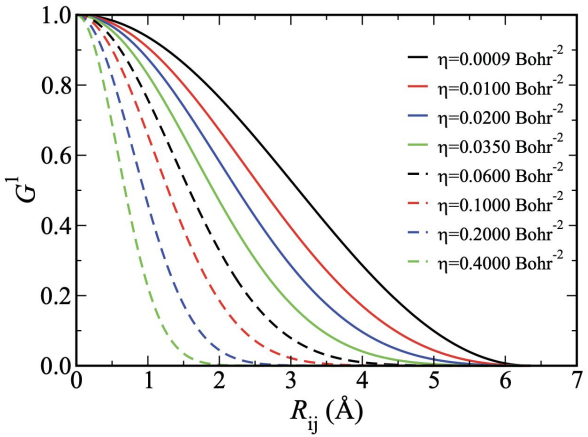




Artrith, Hiller, Behler, PSS 2012

Radial symmetry functions

$$G_i^1 = \sum_{j \neq i} e^{-\eta R_{ij}^2} \cdot f_c(R_{ij}).$$

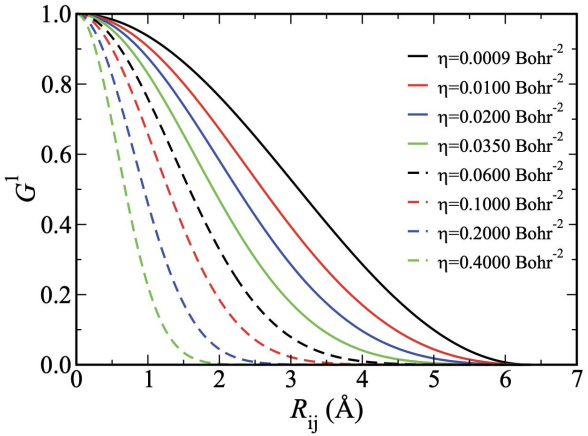


Artrith, Hiller, Behler, PSS 2012

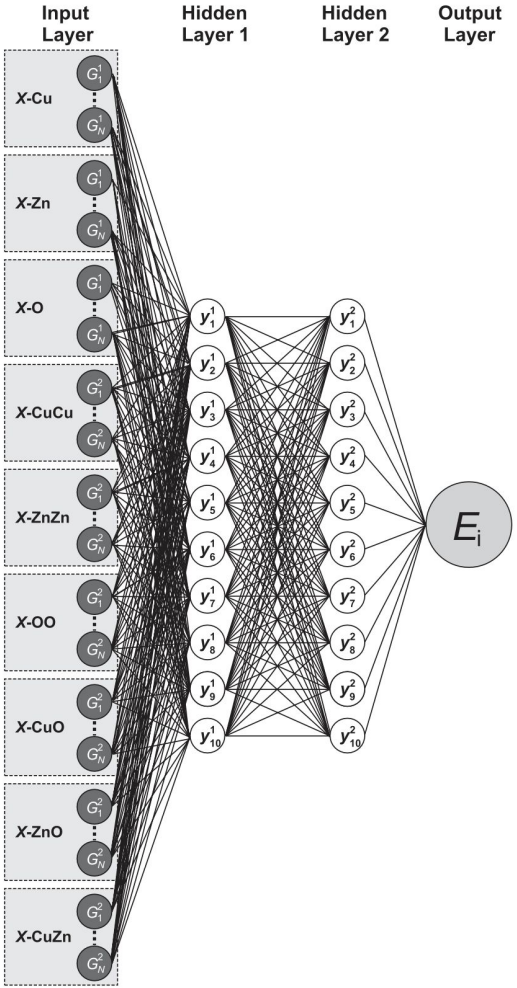
Angular symmetry functions

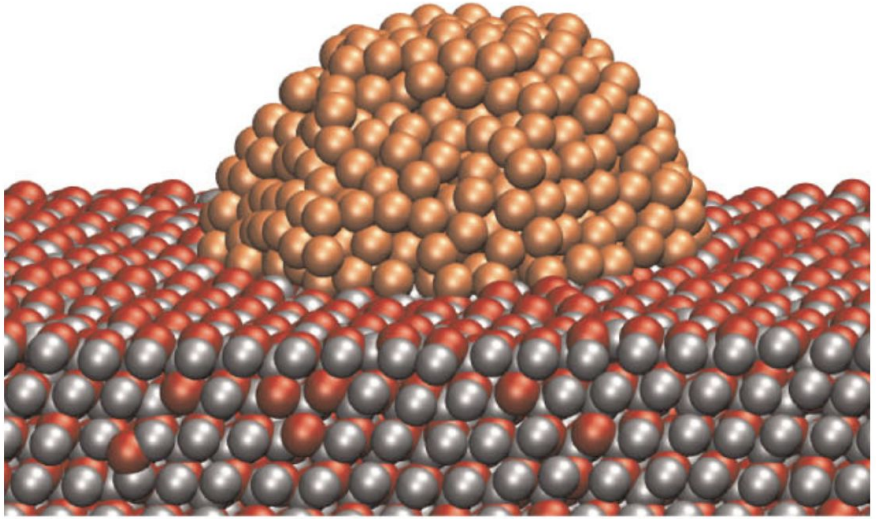
$$G_i^2 = 2^{1-\zeta} \sum_{j \neq i} \sum_{k \neq i,j} \left(1 + \lambda \cos\theta_{ijk}\right)^{\zeta} \cdot e^{-\eta\left(R_{ij}^2 + R_{ik}^2 + R_{jk}^2\right)}$$

$$\cdot f_c(R_{ij}) \cdot f_c(R_{ik}) \cdot f_c(R_{jk}),$$



Artrith, Hiller, Behler, PSS 2012





Artrith, Hiller, Behler, PSS 2012

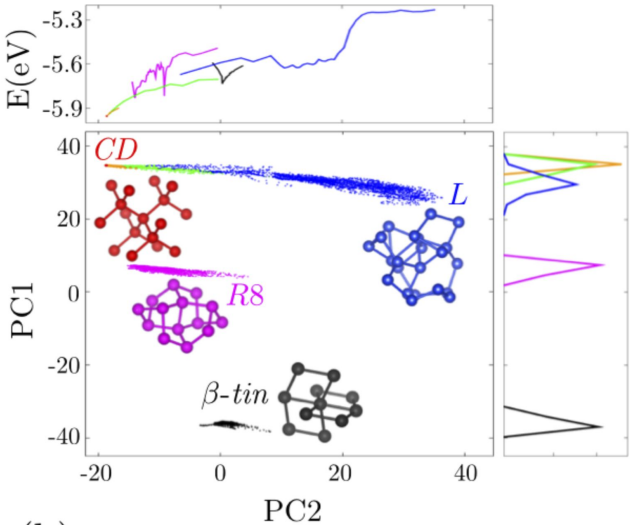
Limitations of the Behler-Parrinello Network

- For M atom types:
  - M radial distribution functions
  - M * (M-1) / 2 + M angular distribution functions
- Not clear how to stack layers in a useful way.
- It perhaps encodes too much physics.

Limitations of the Behler-Parrinello Network

- For M atom types:
  - M radial distribution functions
  - M * (M-1) / 2 + M angular distribution functions
- Not clear how to stack layers in a useful way.
- It perhaps encodes too much physics.

PCA of Symmetry Functions

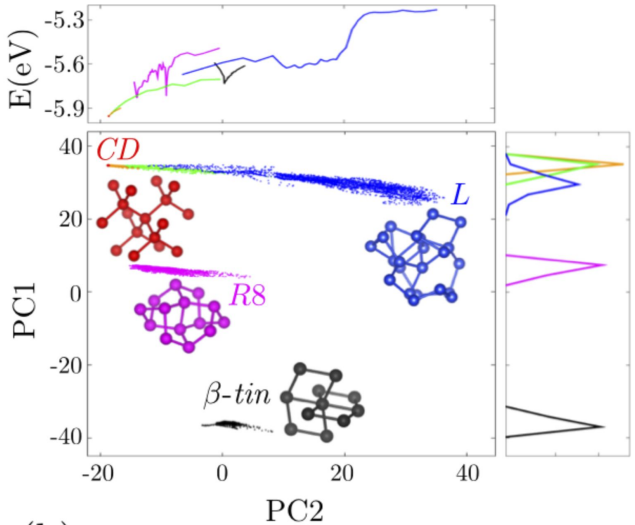


Cubuk et al. JCP 2017

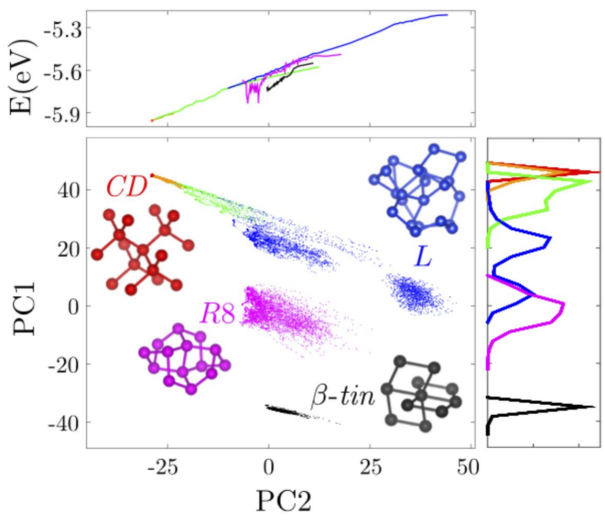
Limitations of the Behler-Parrinello Network

- For M atom types:
  - M radial distribution functions
  - M * (M-1) / 2 + M angular distribution functions
- Not clear how to stack layers in a useful way.
- It perhaps encodes too much physics.

PCA of Symmetry Functions



PCA of NN Representations



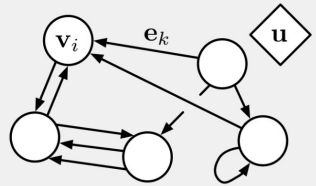
Cubuk et al. JCP 2017

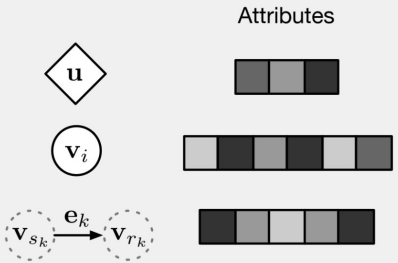
How about GNNs as interatomic potentials?

Can GNNs overcome these challenges?

- For M atom types: (use one hot representation for each atom type)
  - M radial distribution functions
  - M * (M-1) / 2 + M angular distribution functions
- Not clear how to stack layers in a useful way. (graph convolutions)
- It perhaps encodes too much physics. (less processed features)

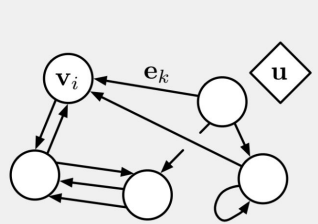
Box 3: Our definition of "graph"

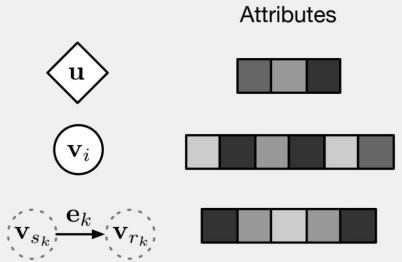




Battaglia et al. arXiv:1806.01261

Box 3: Our definition of "graph"

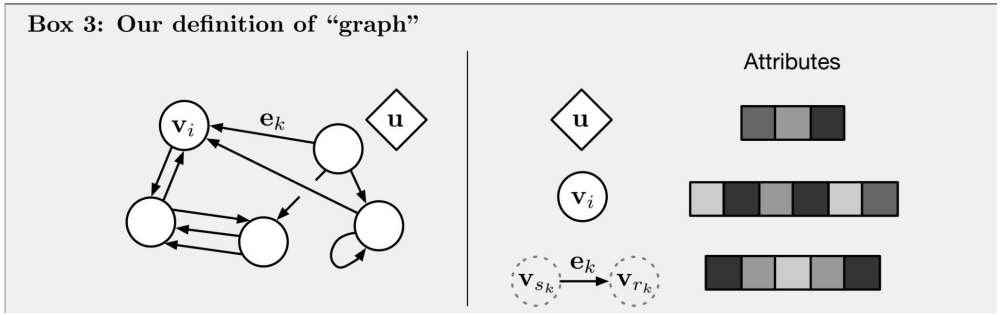




Algorithm 1 Steps of computation in a full GN block.

function GRAPHNETWORK(E, V, $\mathbf{u}$)
  for $k \in \{1 \dots N^e\}$ **do**
    $\mathbf{e}'_k \leftarrow \phi^e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u})$ ▷ 1. Compute updated edge attributes
  end for
  for $i \in \{1 \dots N^n\}$ **do**
    let $E'_i = \{(\mathbf{e}'_k, r_k, s_k)\}_{r_k=i,\ k=1:N^e}$
    $\bar{\mathbf{e}}'_i \leftarrow \rho^{e\to v}(E'_i)$ ▷ 2. Aggregate edge attributes per node
    $\mathbf{v}'_i \leftarrow \phi^v(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u})$ ▷ 3. Compute updated node attributes
  end for
  let $V' = \{\mathbf{v}'\}_{i=1:N^v}$
  let $E' = \{(\mathbf{e}'_k, r_k, s_k)\}_{k=1:N^e}$
  $\bar{\mathbf{e}}' \leftarrow \rho^{e\to u}(E')$ ▷ 4. Aggregate edge attributes globally
  $\bar{\mathbf{v}}' \leftarrow \rho^{v\to u}(V')$ ▷ 5. Aggregate node attributes globally
  $\mathbf{u}' \leftarrow \phi^u(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u})$ ▷ 6. Compute updated global attribute
  return $(E', V', \mathbf{u}')$
end function

Battaglia et al. arXiv:1806.01261

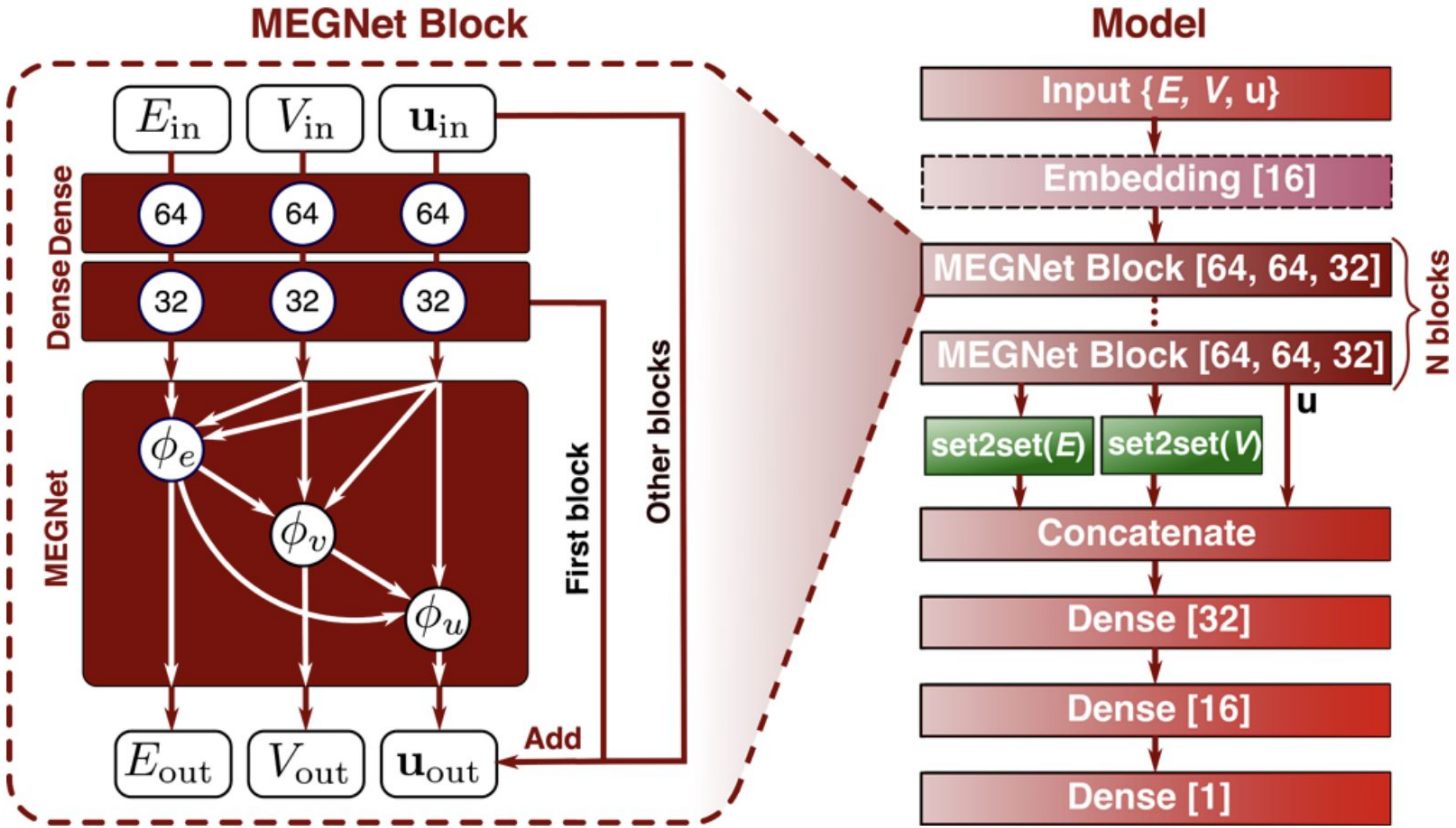


A GN block contains three "update" functions, ϕ, and three "aggregation" functions, ρ,

$$\mathbf{e}'_k = \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) \qquad \bar{\mathbf{e}}'_i = \rho^{e\to v}\left(E'_i\right)$$
$$\mathbf{v}'_i = \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) \qquad \bar{\mathbf{e}}' = \rho^{e\to u}\left(E'\right)$$
$$\mathbf{u}' = \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right) \qquad \bar{\mathbf{v}}' = \rho^{v\to u}\left(V'\right)$$

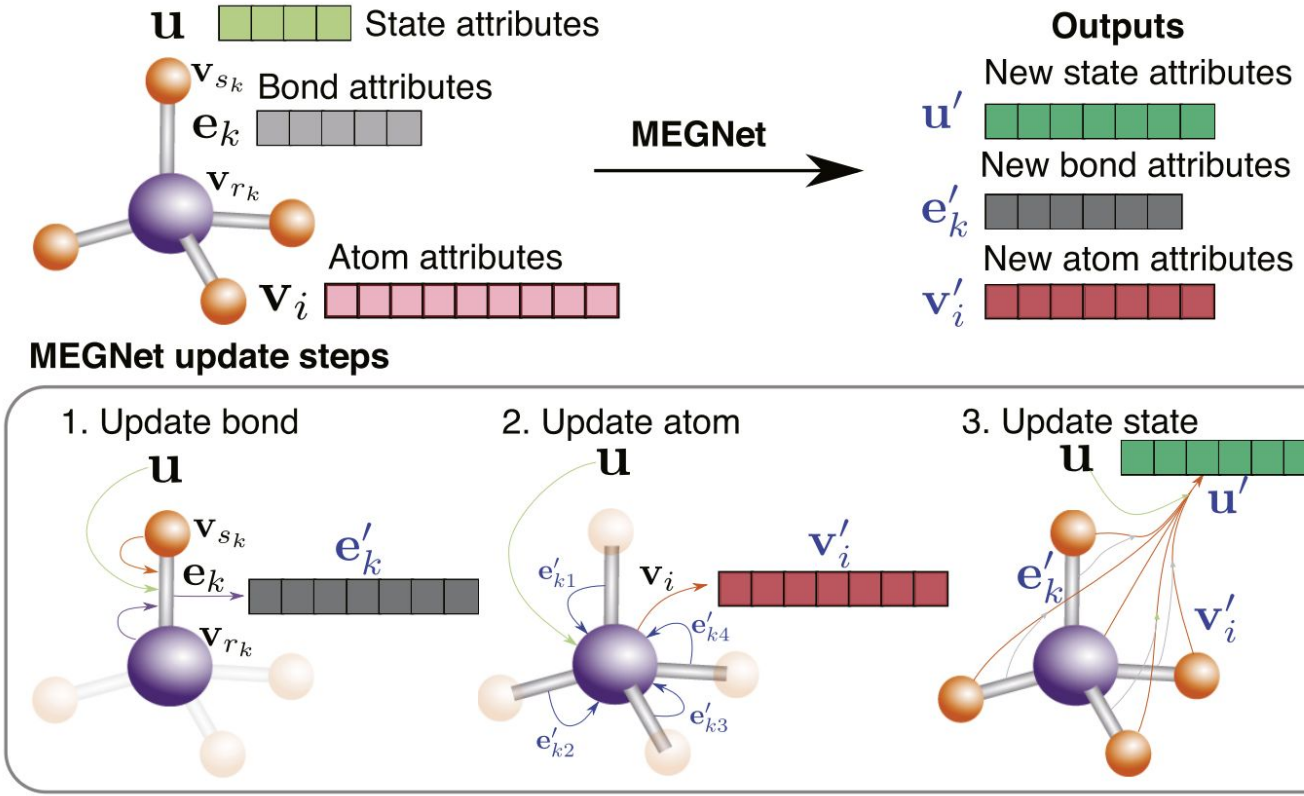
Battaglia et al. arXiv:1806.01261

MEGNet



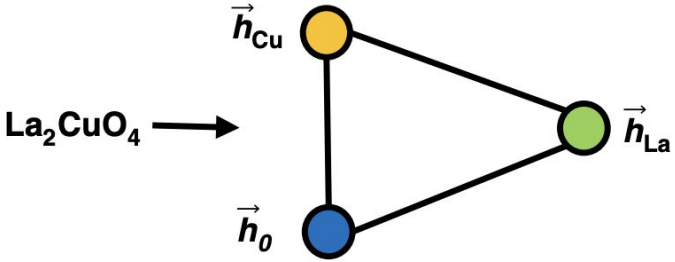
Chen et al. Chemistry of Mat. 2019

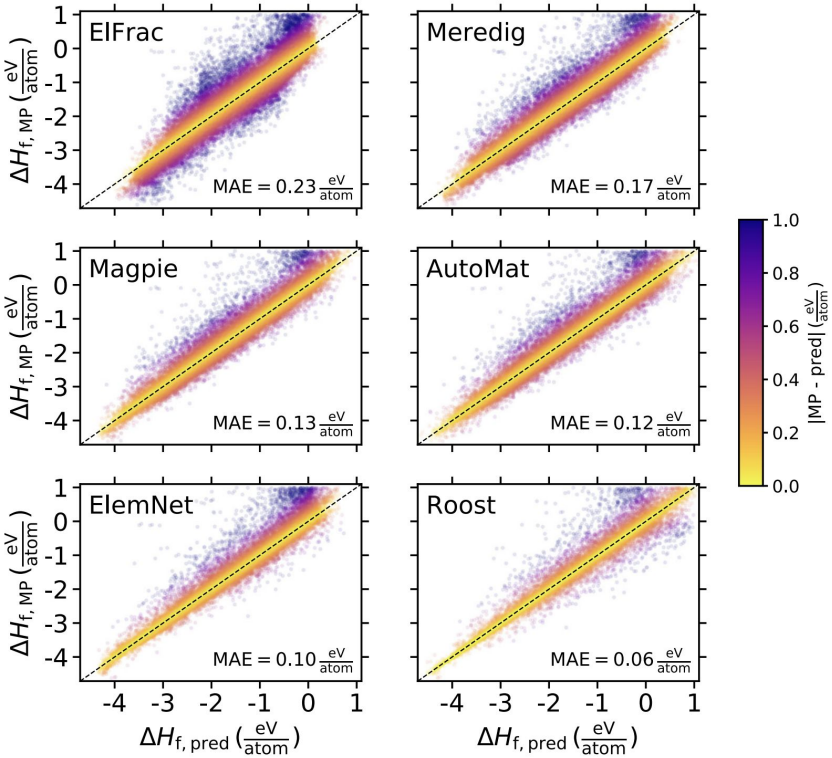
MEGNet



Chen et al. Chemistry of Mat. 2019

Roost: GNNs are also good for compositional predictions





Goodall, Lee, Nat. Comm. 2020

Bartel et al. npj Comp. Mat. 2020

Inductive bias and training set size: historical trends

Hand-crafted features → CNNs → ViTs

Inductive bias and training set size: historical trends

Hand-crafted features (10 images) → CNNs (1M images) → ViTs (1B images)

Inductive bias and training set size: historical trends

Hand-crafted features → CNNs → ViTs

Hand-crafted features → LSTMs → Transformers

Hand-crafted features → BP NN → GNNs

JAX, M.D.

A Framework for Differentiable Physics

Samuel S. Schoenholz
Google Research: Brain Team
schsam@google.com

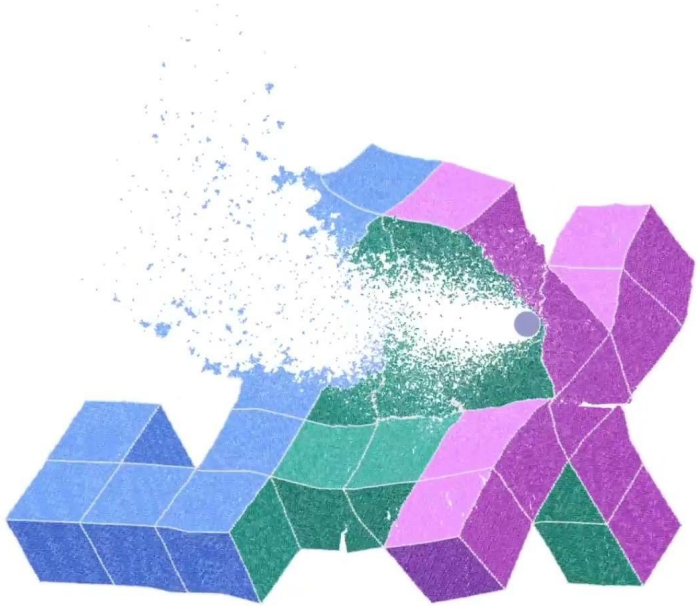
Ekin D. Cubuk
Google Research: Brain Team
cubuk@google.com

JAX, M.D.

A Framework for Differentiable Physics

Samuel S. Schoenholz
Google Research: Brain Team
schsam@google.com

Ekin D. Cubuk
Google Research: Brain Team
cubuk@google.com

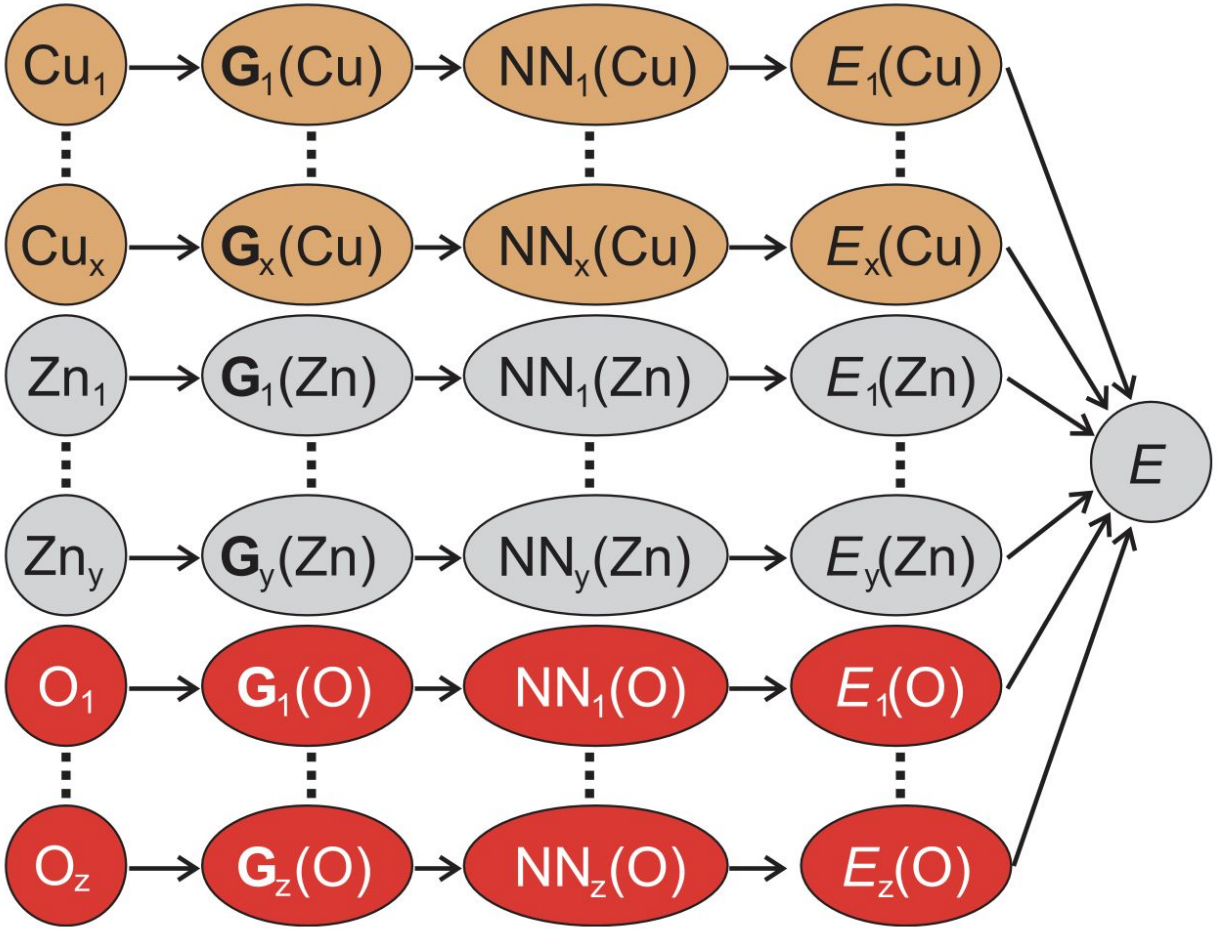


84,000 Rigid Bodies
336,000 Total Particles
P100 GPU in public Colab

Goals

Easy to express complicated simulations

Everything written in Python
Derivatives computed using automatic differentiation



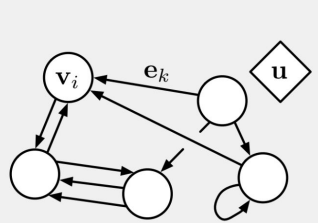
$$F_{i,\alpha} = -\frac{\partial E}{\partial R_{i,\alpha}}$$

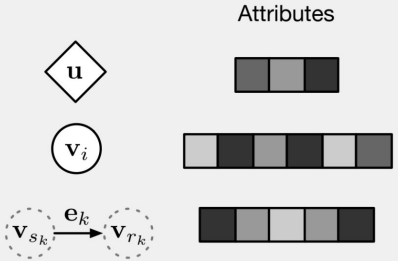
$$= -\sum_{k=1}^{M} \frac{\partial E_k}{\partial R_{i,\alpha}}$$

$$= -\sum_{k=1}^{M} \sum_{j=1}^{N_k} \frac{\partial E_k}{\partial G_{k,j}} \frac{\partial G_{k,j}}{\partial R_{i,\alpha}}.$$

Artrith, Hiller, Behler, PSS 2012

Box 3: Our definition of "graph"





$$F_{i,\alpha} = -\frac{\partial E}{\partial R_{i,\alpha}}$$

Battaglia et al. arXiv:1806.01261

Goals

Easy to express complicated simulations

Everything written in Python
Derivatives computed using automatic differentiation

My PhD: Fortran -> Lua -> C++ -> Python/Matlab

Goals

Easy to express complicated simulations

Everything written in Python
Derivatives computed using automatic differentiation

Fast enough to do high quality research

Python code gets Just-In-Time compiled to CPU / GPU / TPU via XLA
Efficient spatial partitioning strategies [O(N) vs $O(N^2)$]

Goals

Easy to express complicated simulations

Everything written in Python
Derivatives computed using automatic differentiation

Fast enough to do high quality research

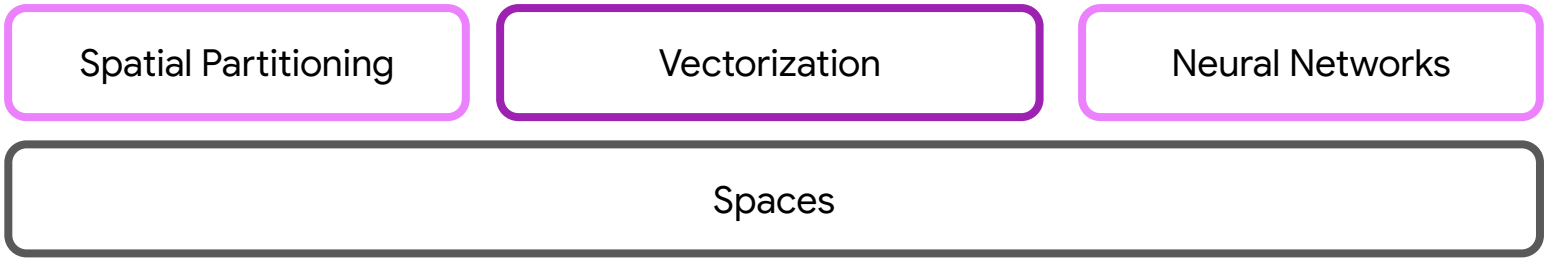
Python code gets Just-In-Time compiled to CPU / GPU / TPU via XLA
Efficient spatial partitioning strategies [O(N) vs $O(N^2)$]

ML as a first class citizen

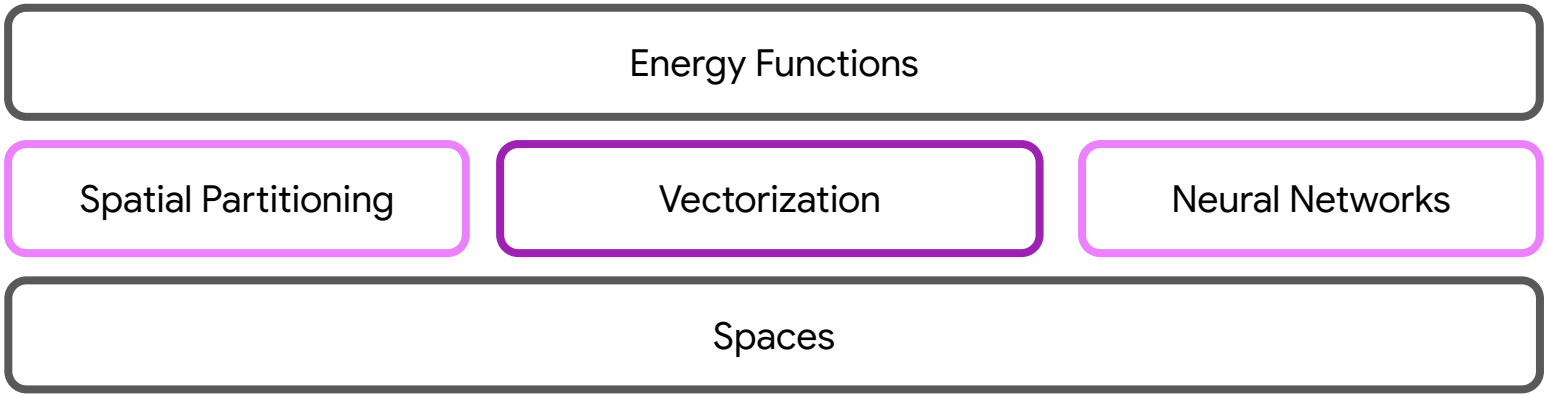
Any function can be a neural network
Whole simulations should be differentiable

Organization

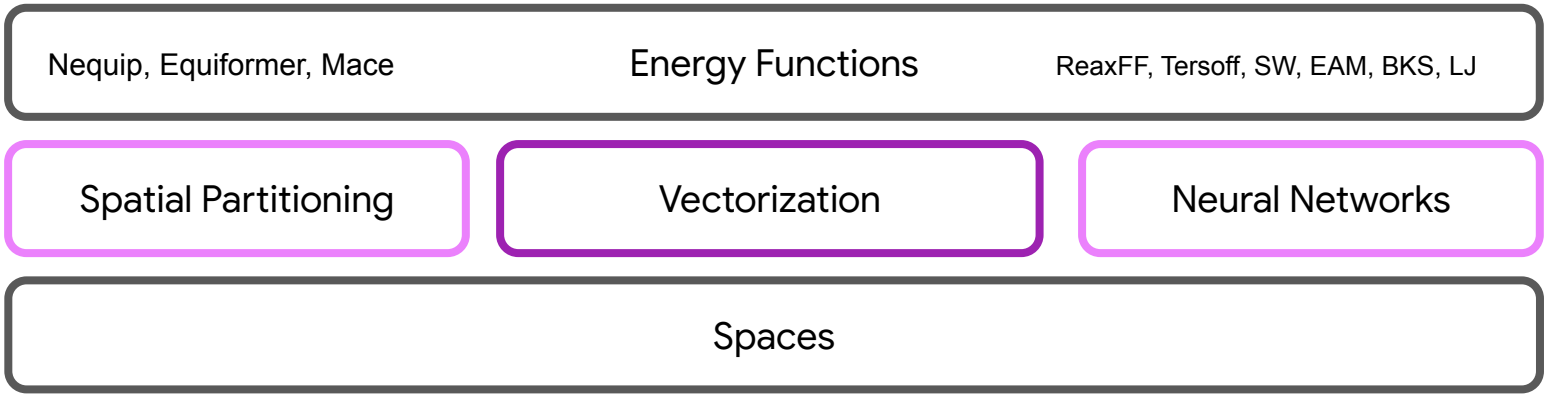
Organization



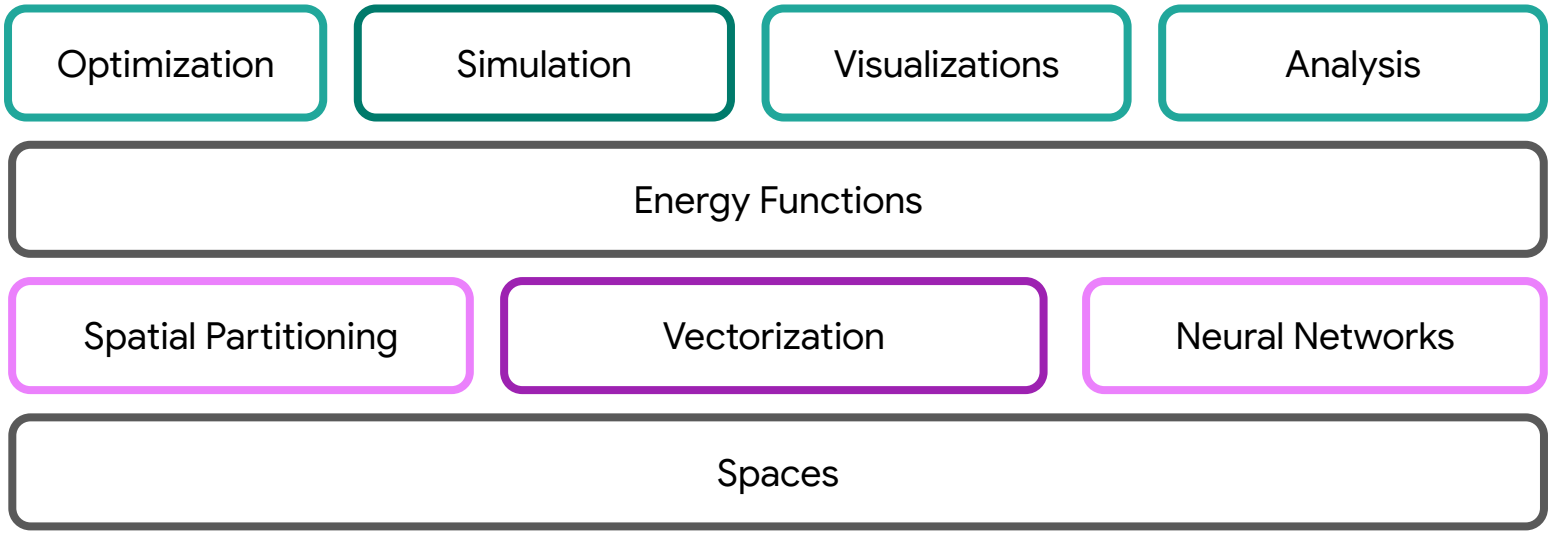
Organization



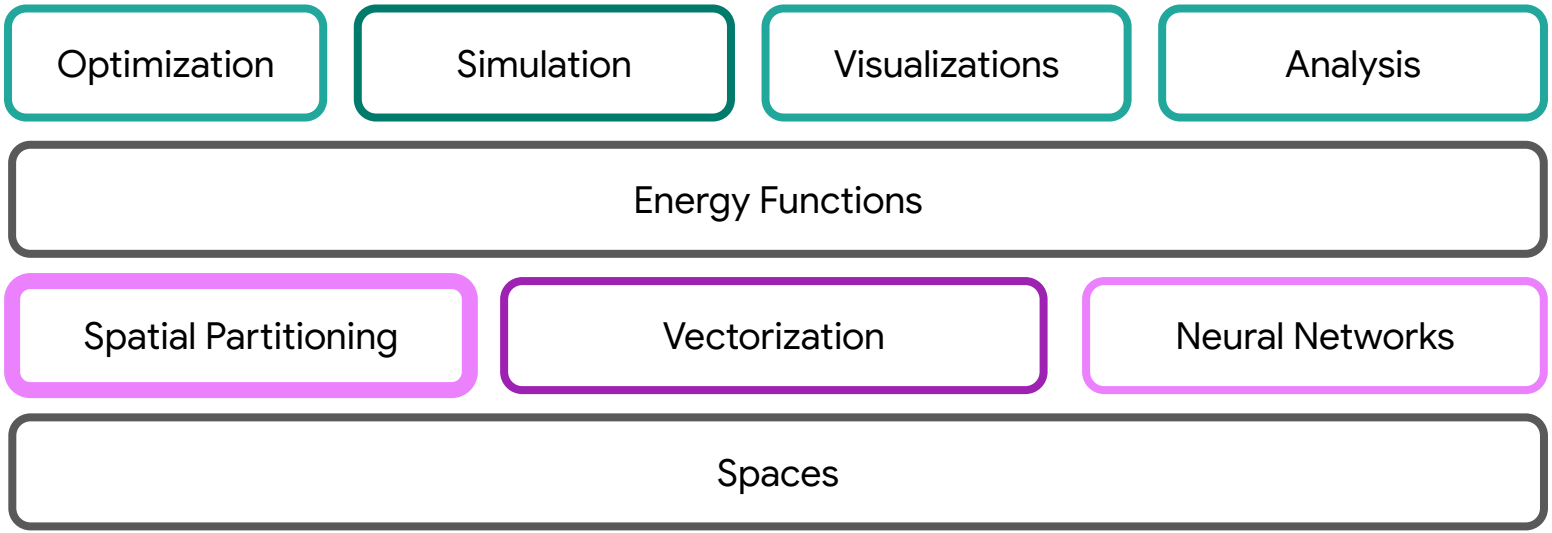
Organization



Organization



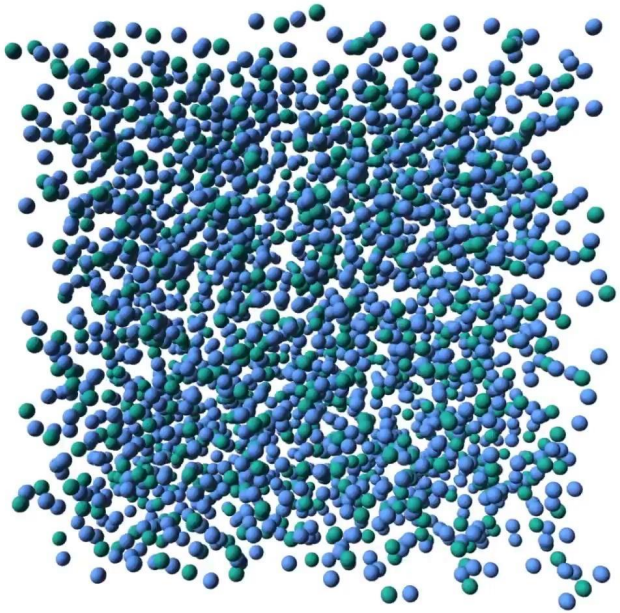
Organization



Spatial Partitioning

Efficiently split systems

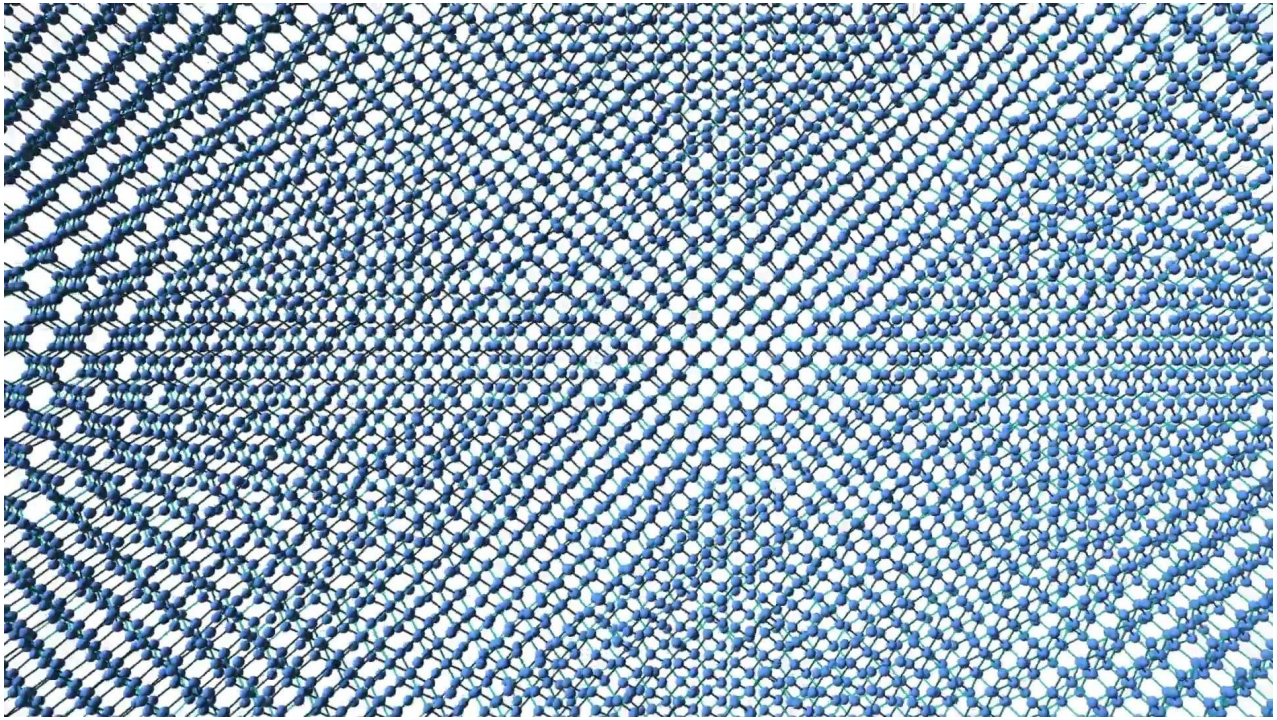
- Into cubic cells
- Into lists of neighbors



Neural Network Integration

Supports

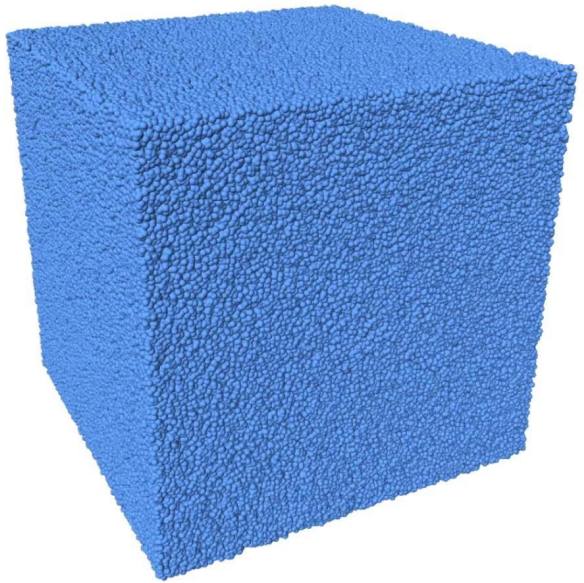
- Common featurizations
- Graph Networks
- Jraph integration
  github.com/deepmind/jraph
- Neighbor lists



Neural Network Integration

Supports

- Common featurizations
- Graph Networks
- Jraph integration
  github.com/deepmind/jraph
- Neighbor lists



Future work for GNNs as Interatomic Potentials

- Currently not good enough for modeling the whole periodic table.

- Stress predictions need to improve.

- Starting from bad inputs (e.g. AIRSS) needs to improve.

A dataset for future work for GNNs as Interatomic Potentials

- Stress predictions need to improve.
- Starting from bad inputs (e.g. AIRSS) needs to improve.

Crystal Structure Search with Random Relaxations Using Graph Networks

---

Gowoon Cheon
Stanford University
gcheon@stanford.edu

Lusann Yang
Google Research
lusann@google.com

Kevin McCloskey
Google Research
mccloskey@google.com

Evan J. Reed
Stanford University
evanreed@stanford.edu

Ekin D. Cubuk
Google Research
cubuk@google.com

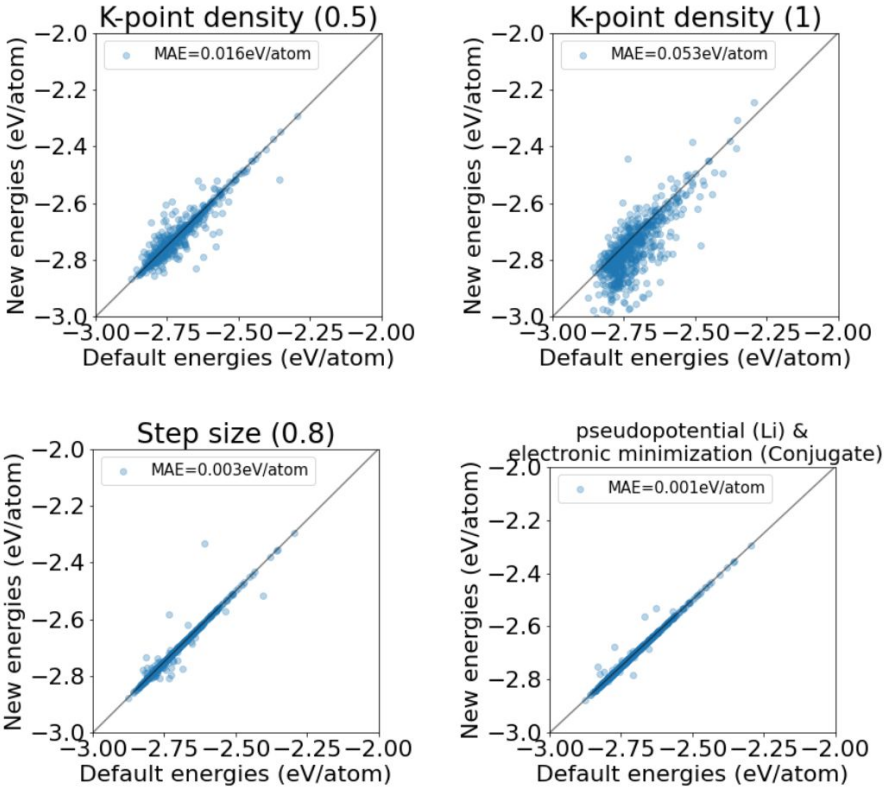
Database of >100,000 AIRSS calculations

Li-Si, Li-O, Si-O and Li-Si-O with diverse stoichiometries

system	#stoichiometry	#structures
Li-Si	5	60000
Li-O	2	10000
Si-O	1	33000
Li-Si-O	2	10000

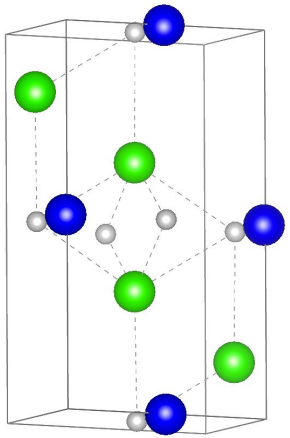
+ Out-of-domain generalization test sets with different DFT parameters
+ MD

Energies of final structures relaxed with different DFT parameters

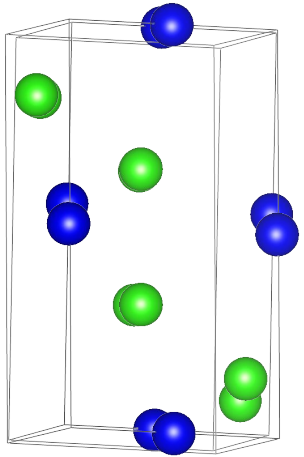


We learn crystal relaxations from random structures data using graph neural networks (GNN)

- We train graph neural networks on random structure relaxations data[1,2] of Li-Si
- We modify GNN architecture[3] to directly output stress measurements
- We investigate generalizability of GNN model relaxations on out-of-domain data
- Gaussian noise data augmentation helps generalizability



Fictitious atoms are added on cell planes (grey)



Gaussian noise is added to atom positions and lattice vectors

1.Deringer, V. L., Proserpio, D. M., Csányi, G. & Pickard, C. J. *Faraday Discuss.* **211**, 45–59 (2018).
2.Deringer, V. L., Pickard, C. J. & Proserpio, D. M. *Angewandte Chemie International Edition* **59**, 15880–15885 (2020).
3. Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. *Chem. Mater.* **31**, 3564–3572 (2019).
4. Sanchez-Gonzalez, A. et al. *arXiv:2002.09405* (ICLR 2020).

GNNs offer a large improvement over MEAM

GNN models were trained on Li-Si systems up to **34** atoms, **excluding $Li_{15}Si_4$**

38-atom experimental structure of $Li_{15}Si_4$ was found with GNNs

Model	Gaussian noise (std. dev)	Force MAE	Stress MAE	Relaxed structure match*	Matches in trajectory**
GNN	0	0.033 (2.1×10^{-4})	1.05 (0.021)	0.40 (0.015)	0.65 (0.015)
GNN	0.01	**0.031** (4.5×10^{-4})	**0.98** (0.047)	0.42 (0.007)	**0.67** (0.006)
GNN	0.02	0.032 (1.4×10^{-4})	1.01 (0.023)	**0.43** (0.005)	**0.67** (0.000)
GNN	0.03	0.034 (1.9×10^{-4})	1.06 (0.050)	0.42 (0.007)	0.66 (0.006)
MEAM	-	0.55	9.11	0.18	0.39