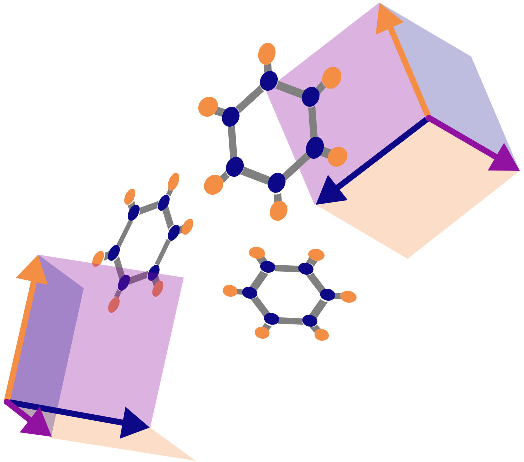
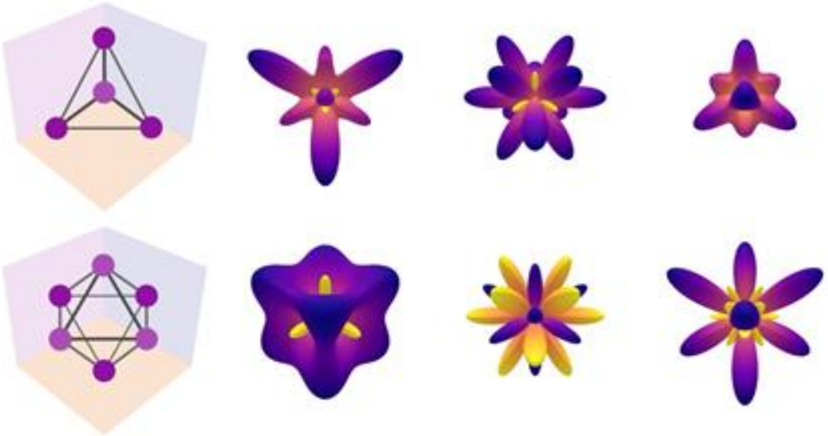
Applications of Euclidean Neural Networks for the Understanding and Design of Atomistic Systems

Slides: https://tinyurl.com/2025-aims-tess





Tess Smidt

Atomic Architects

MIT EECS

rle AT MIT RESEARCH LABORATORY OF ELECTRONICS AT MIT

Applications of Euclidean Neural Networks for the Understanding and Design of Atomistic Systems

Slides: https://tinyurl.com/2025-aims-tess

1. Properties of Euclidean neural networks *(E(3)NNs)*
3. Applications
    a. Interatomic potentials
    b. Generative models
4. *Tensor Products*

Tess Smidt

Atomic Architects

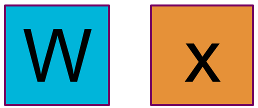
MIT EECS

rle AT MIT RESEARCH LABORATORY OF ELECTRONICS AT MIT

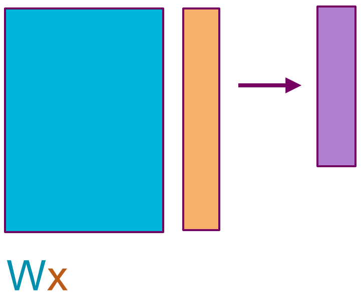
Neural networks are specially designed for different data types.
Assumptions about the data type are built into how the network operates.

W x

Neural networks are specially designed for different data types. Assumptions about the data type are built into how the network operates.

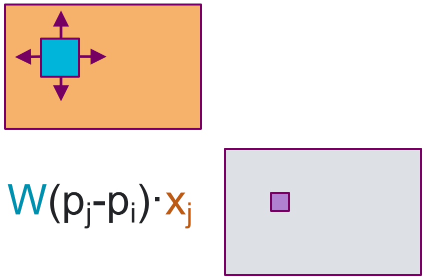


Arrays ⇨ Dense NN



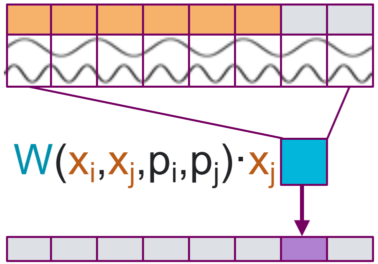
Features are independent, fixed length, and ordered. Weights are position sensitive.

Images ⇨ Conv. NN



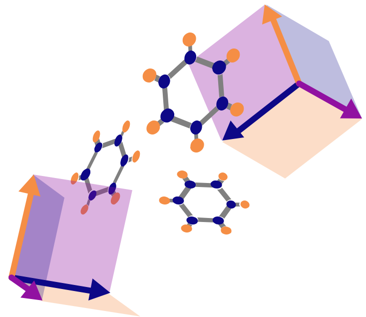
The same features can be found anywhere in an image. Filters operate on patches of image to create new pixel features (Locality).

"Tokens" ⇨ Transformer



Want to "pay attention" on the impact of every token on every other token modulated by "positional" embeddings.

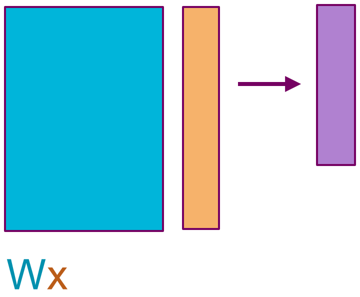
3D data ⇨ Euclidean NN



Data in 3D Euclidean space. Freedom to choose coordinate system. Data transforms predictably under change of coordinate system (***equivariance***).

**Neural networks are specially designed for different data types.
Assumptions about the data type are built into how the network operates.**

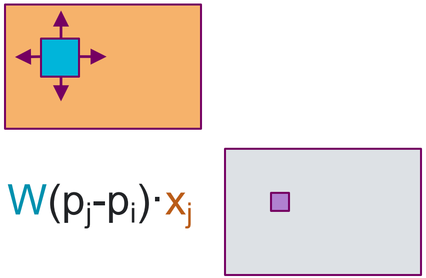
Arrays ⇨ Dense NN



Features are independent, fixed length, and ordered. Weights are position sensitive.

No symmetry!

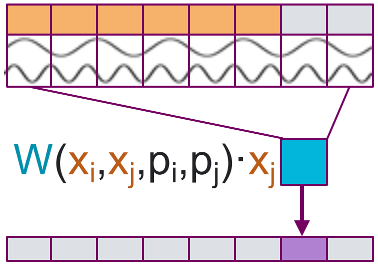
Images ⇨ Conv. NN



The same features can be found anywhere in an image. Filters operate on patches of image to create new pixel features (Locality).

Translation symmetry

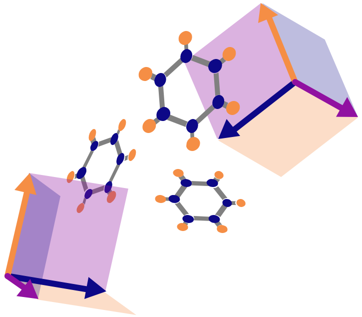
"Tokens" ⇨ Transformer



Want to "pay attention" on the impact of every token on every other token modulated by "positional" embeddings.

Permutation symmetry*

3D data ⇨ Euclidean NN



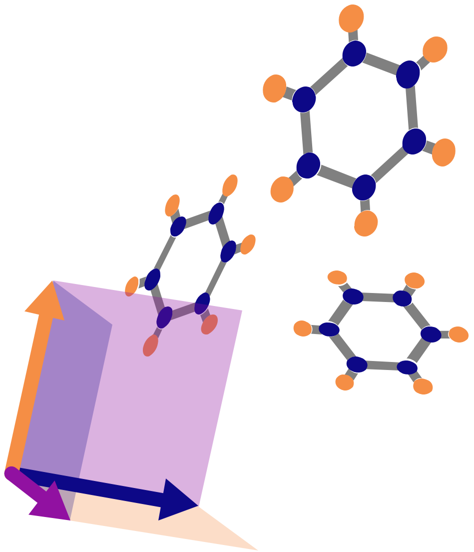
Data in 3D Euclidean space. Freedom to choose coordinate system. Data transforms predictably under change of coordinate system (***equivariance***).

Euclidean symmetry

Coordinate systems are useful for describing physical systems...

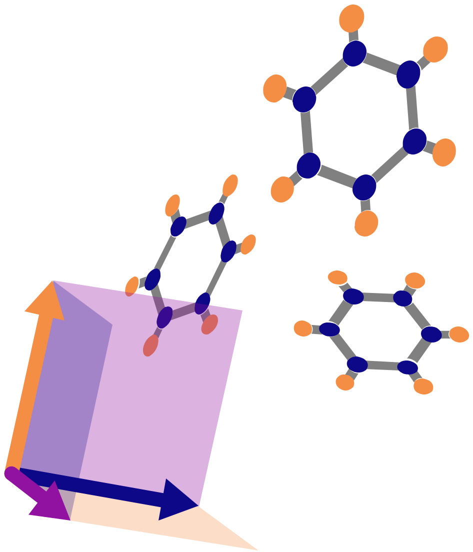
Coordinate systems are useful for describing physical systems...

We use them to describe where things are in 3D space, ...



Coordinate systems are useful for describing physical systems...

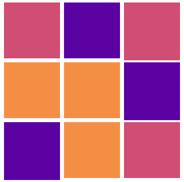
We use them to describe where things are in 3D space, ...



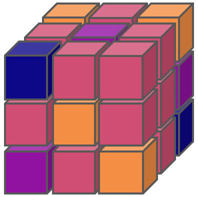
articulate properties that depend on direction, ...

3 vector
forces
displacements

3x3 matrix
moment of inertia
stress
strain



3x3x3 tensor
Levi-Civita tensor

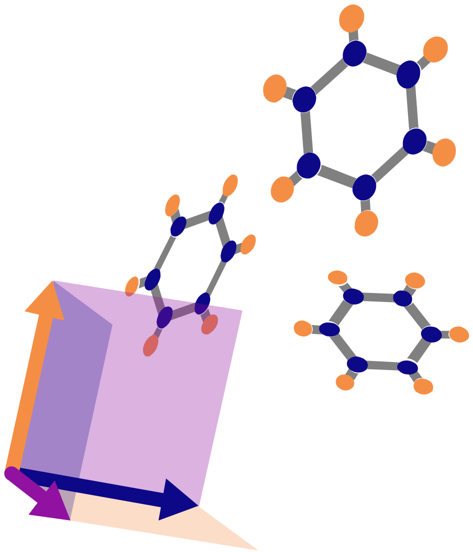


3x3x3x3 tensor
elasticity

...

Coordinate systems are useful for describing physical systems...

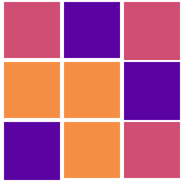
We use them to describe where things are in 3D space, ...



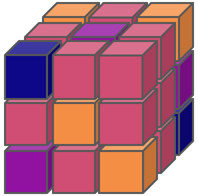
articulate properties that depend on direction, ...

3 vector
forces
displacements

3x3 matrix
moment of inertia
stress
strain



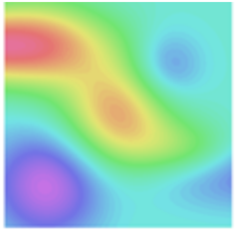
3x3x3 tensor
Levi-Civita tensor



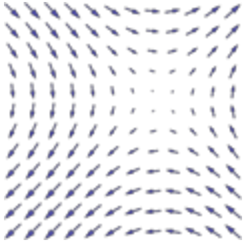
3x3x3x3 tensor
elasticity
...

and define tensor fields that vary over space.

scalar field
energy
pressure



vector field
velocity



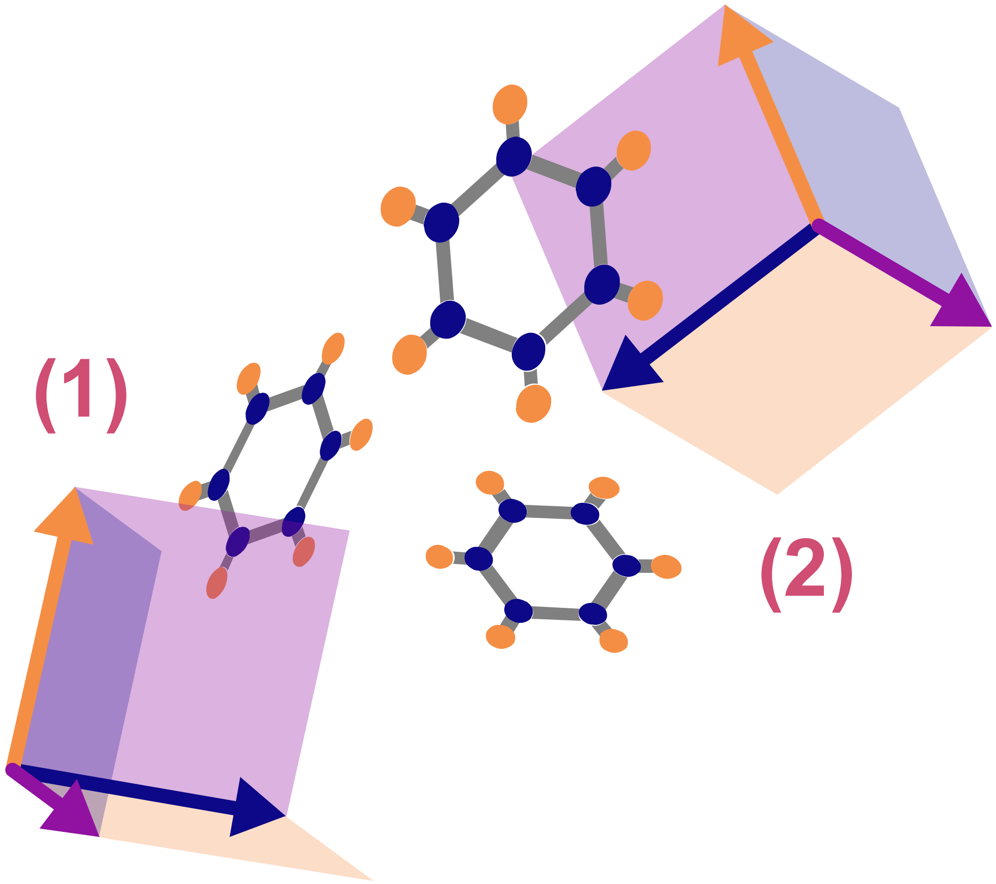
matrix and tensor fields
strain fields

Coordinate systems are useful for describing physical systems but *fundamentally arbitrary*.

Coordinate systems are useful for describing physical systems but ***fundamentally arbitrary***.

(1) and (2) use different coordinate systems to describe the **same physical system**.

Traditional machine learning see (1) and (2) as completely different!

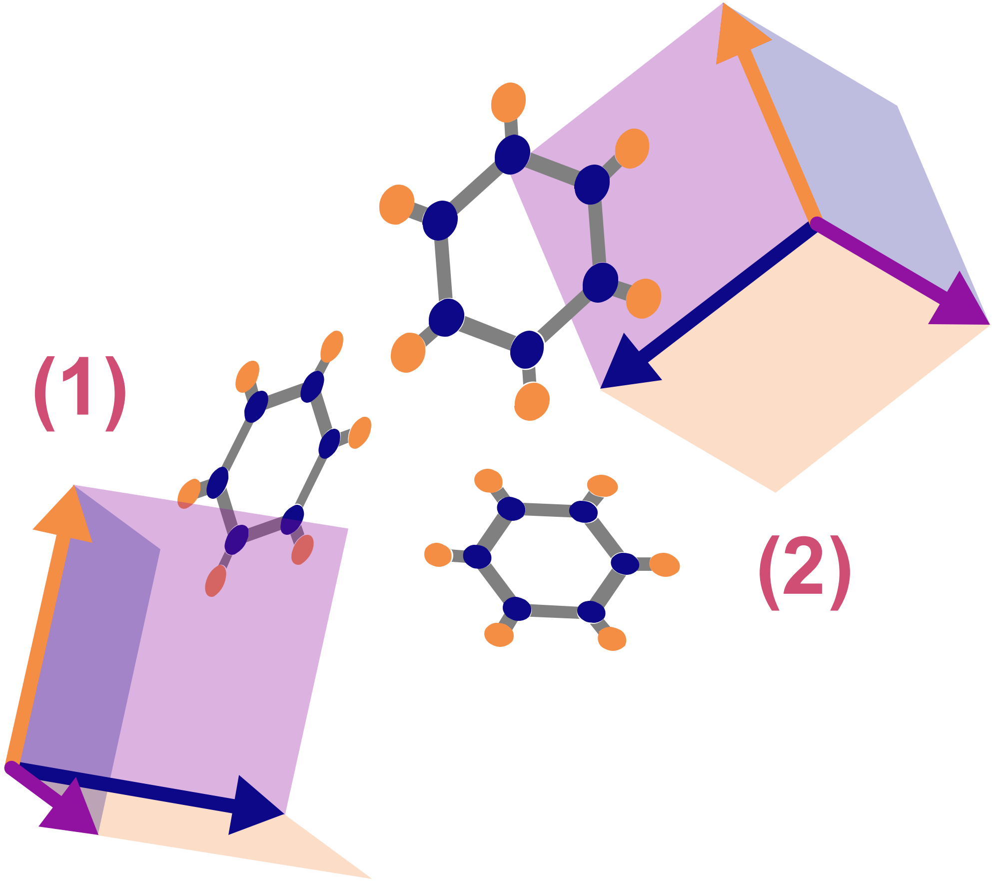


Coordinate systems are useful for describing physical systems but ***fundamentally arbitrary***.

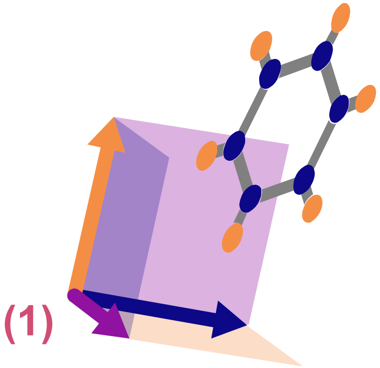
(1) and (2) use different coordinate systems to describe the **same physical system**.

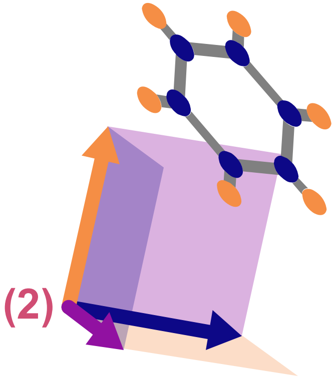
Traditional machine learning see (1) and (2) as completely different!

Can we use coordinates with ML without being biased or thrown off by them?



A neural network "equivariant" to E(3) sees (1) and (2) as the same system described differently even without training.



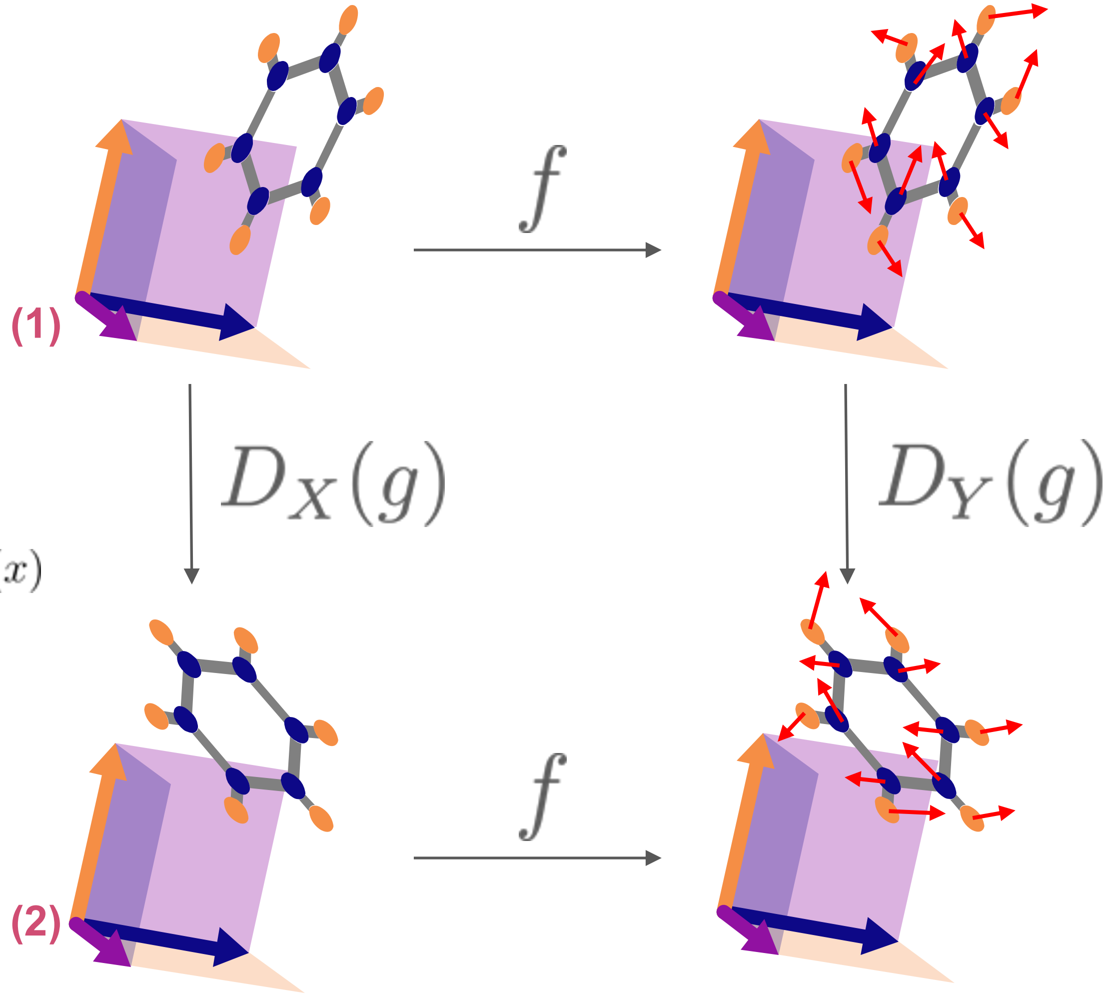


A neural network "equivariant" to E(3) sees (1) and (2) as the same system described differently even without training.

Equivariance:

If the coordinate system changes, the output changes accordingly.

$$f : X \to Y, f(D_X(g)x) = D_Y(g)f(x)$$

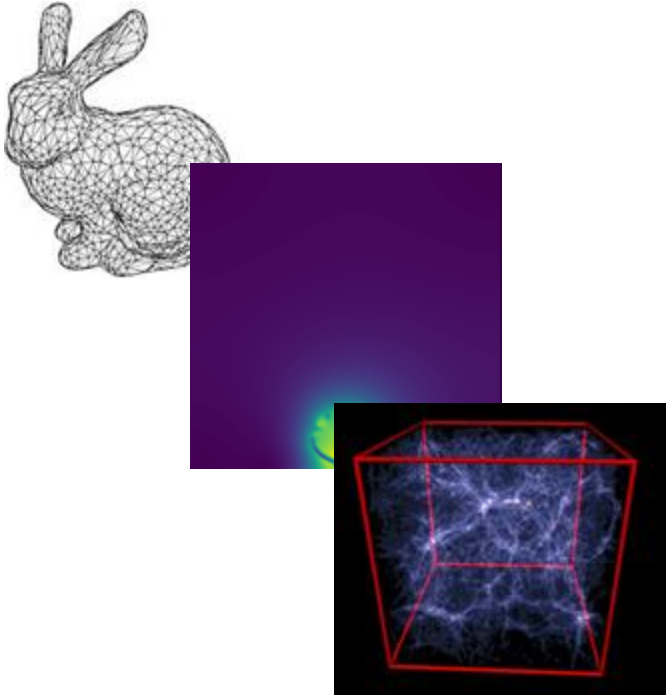


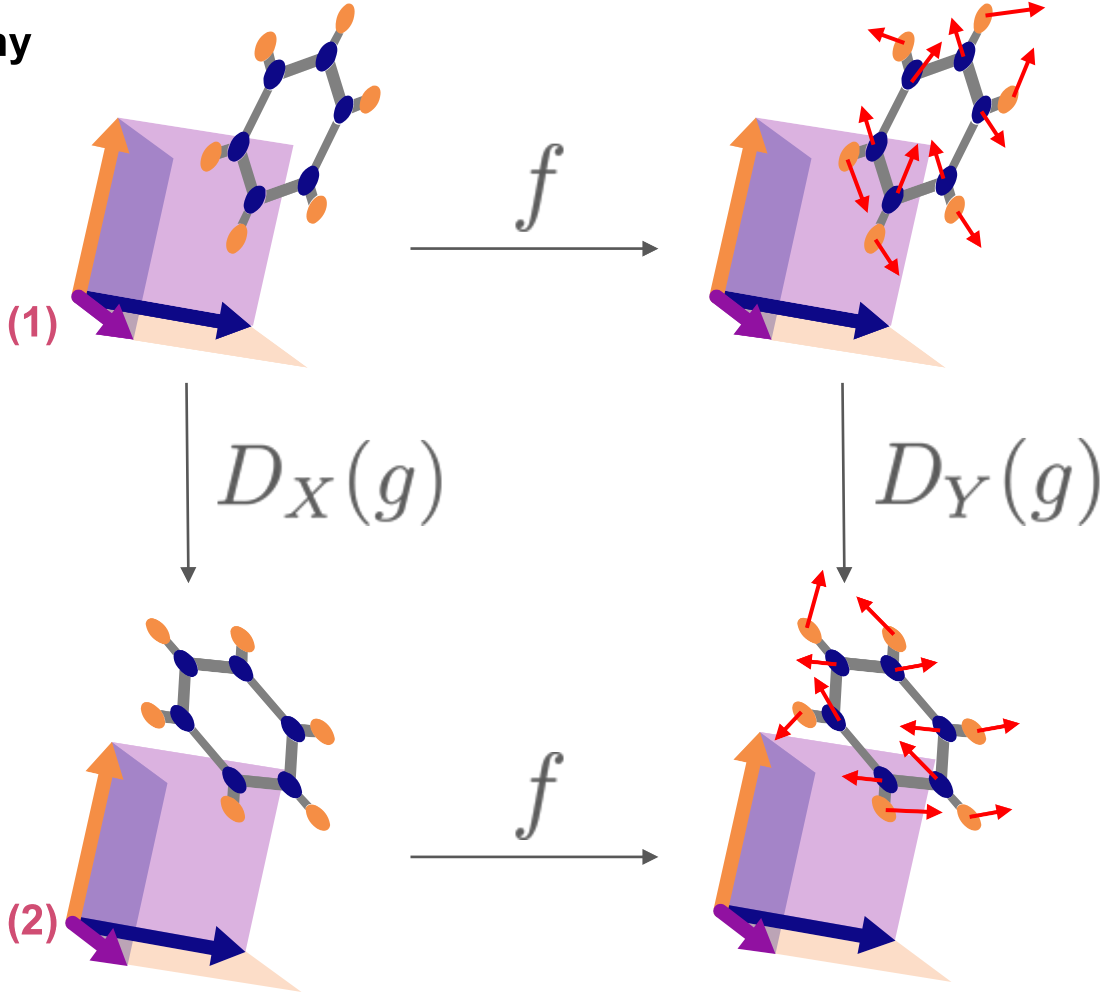
E(3) ***"equivariance"*** **applies to any type of 3D geometric data**

meshes, voxels, points, etc

at any length scale.

from the atomic to the cosmic





Consequence of equivariance!

<u>All data</u> **acted on by O(3) can be broken up into simpler "data types"** *(irreps)* **defined by...**

L

angular frequency (positive int)

rate of change under rotation

parity

even or **odd**

does not or does flip sign under inversion

Consequence of equivariance!

All data acted on by O(3) can be broken up into simpler "data types" *(irreps)* defined by...

L — **angular frequency** (positive int)
rate of change under rotation

parity — **even** or **odd**
does not or does flip sign under inversion

Some examples include...

L=0

Even parity
(scalars)
Classification labels

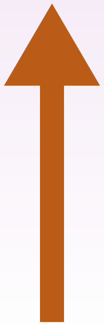
rabbit

Odd parity
(pseudoscalars)
Chirality or "handedness"

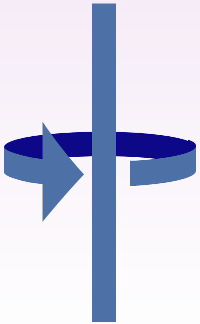
right hand

L=1

Odd parity
(vectors)
Coordinates

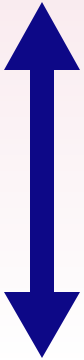


Even parity
(pseudovectors)
Rotation axes

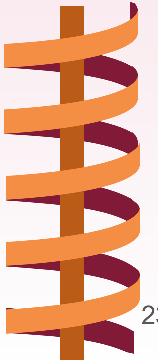


L=2

Even parity
Double-headed Ray

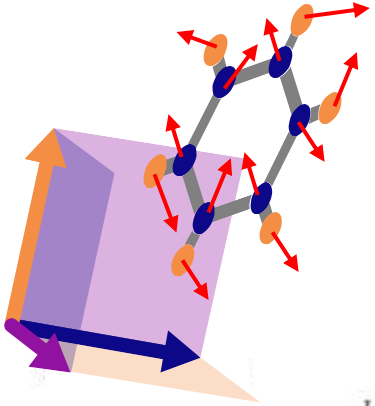


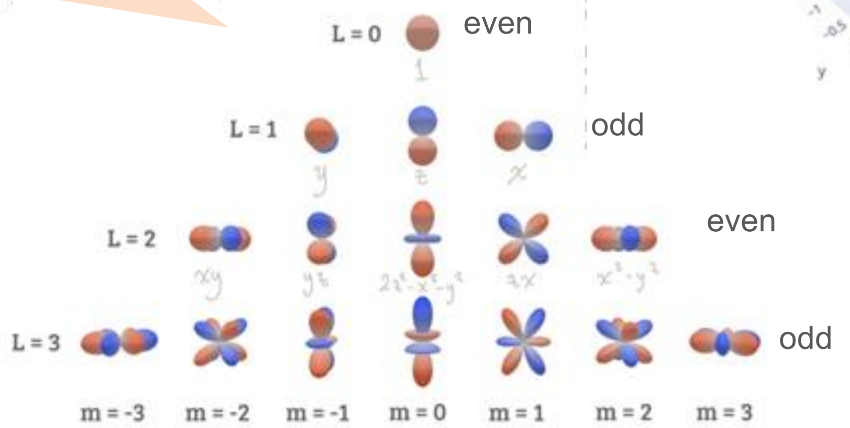
Odd Parity
Helix

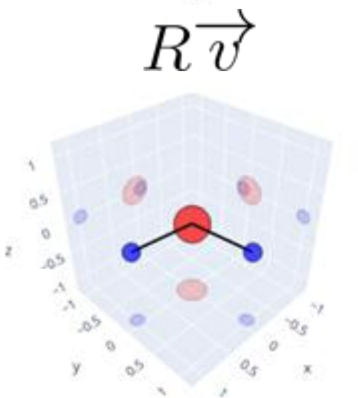


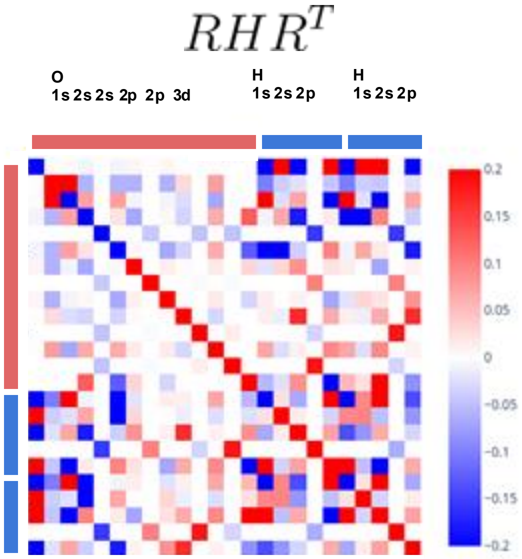
Irreps faithfully encode the “data types” of atomic systems.

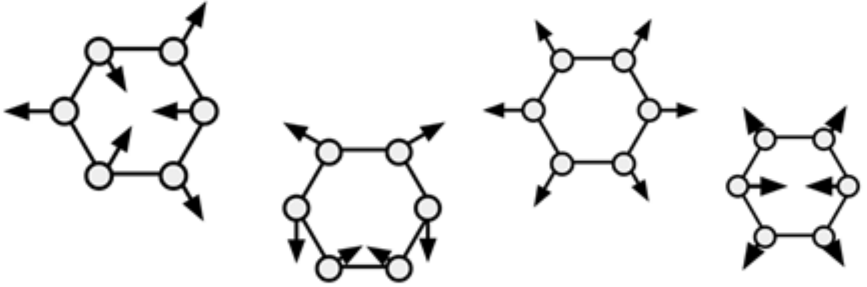
From structure and forces to atomic orbitals, Hamiltonians, and vibrational modes…









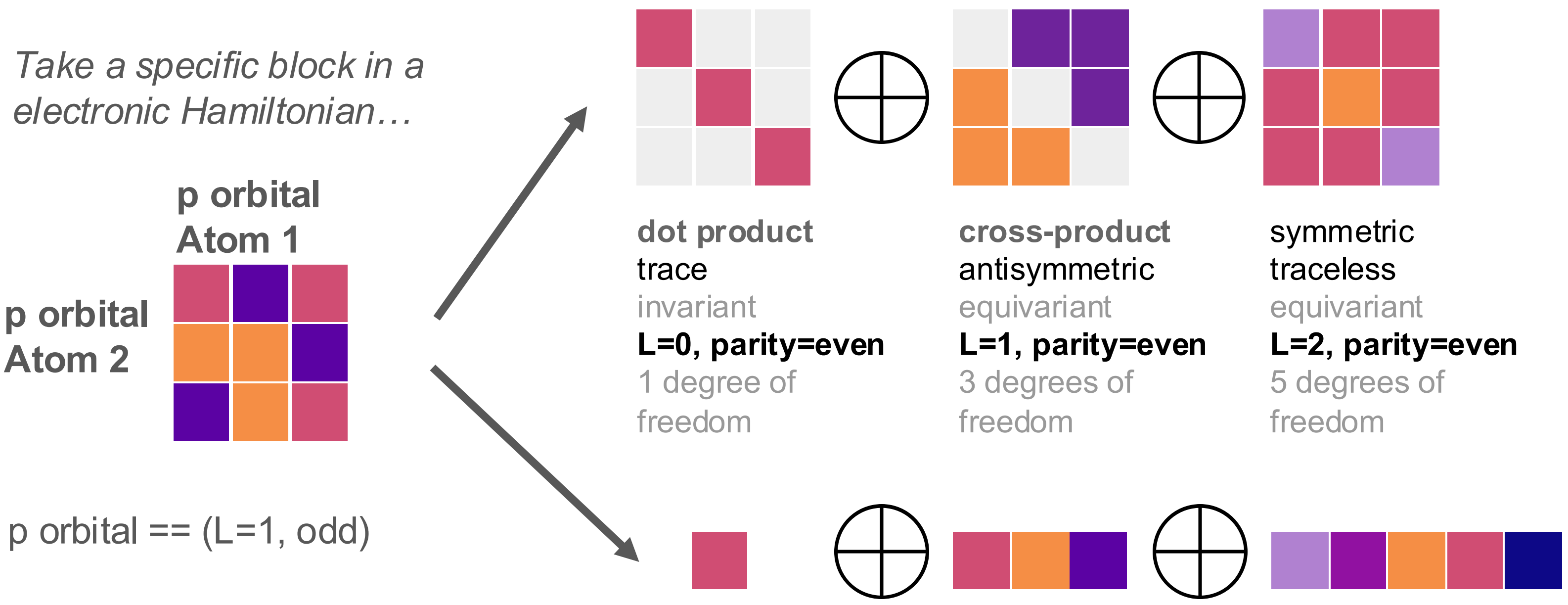


Irreps faithfully encode the “data types” of atomic systems.

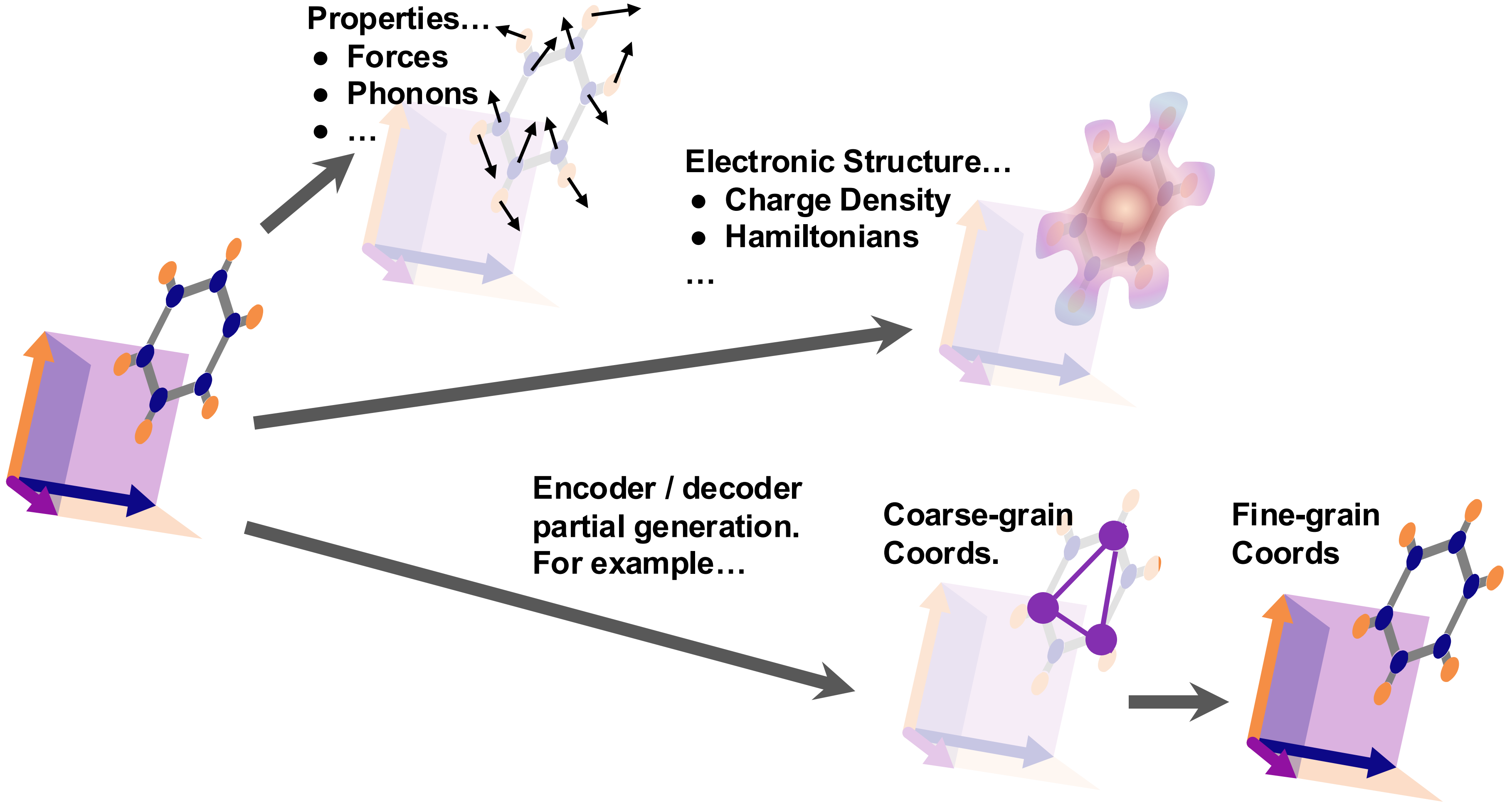
We can “change basis” from multi-index data types to irreps (“decompose to irreps”).

Key part of operation that replaces multiplication in equivariant models

Outer product that decomposes to irreps = “tensor product decomposition”

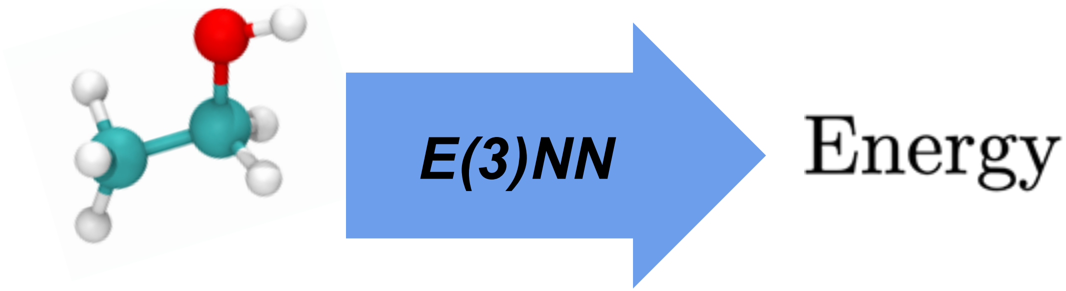


E(3)NNs have been used to build data-efficient and scalable models of physical processes.



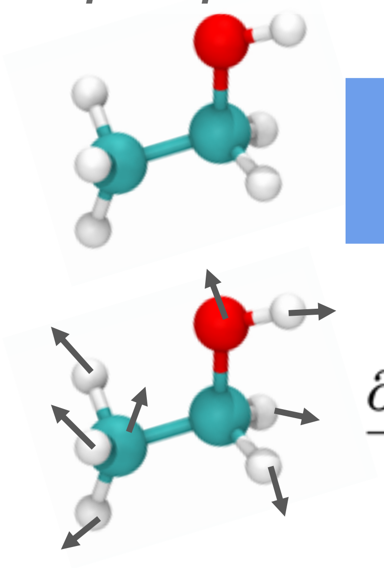
The first successful use case of E(3)NNs was (and continues to be) interatomic potentials.

Trained on “first-principles “ calculations (e.g. DFT)



The first successful use case of E(3)NNs was (and continues to be) interatomic potentials.

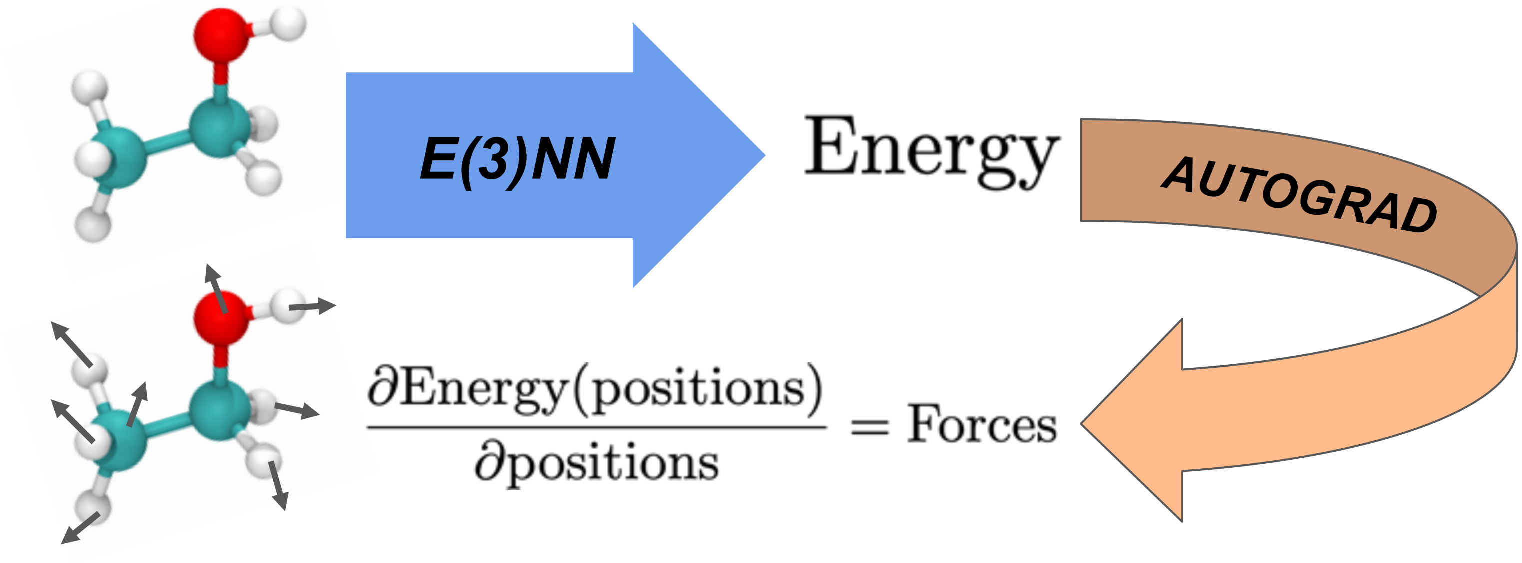
Trained on “first-principles “ calculations (e.g. DFT)



$$\text{E(3)NN} \rightarrow \text{Energy} \xrightarrow{\text{AUTOGRAD}} \frac{\partial \text{Energy}(\text{positions})}{\partial \text{positions}} = \text{Forces}$$

The first successful use case of E(3)NNs was (and continues to be) interatomic potentials.

Trained on "first-principles " calculations (e.g. DFT)



NequIP — Batzner et al (2022)

Allegro — Musaelian et al (2023)

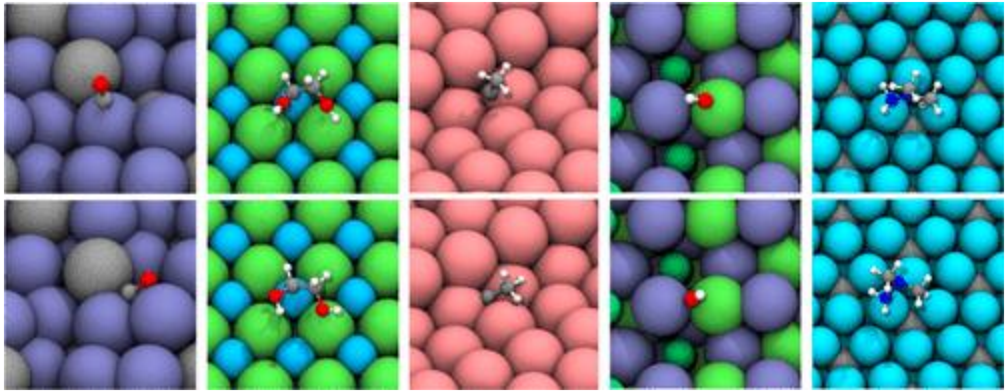
MACE — Batatia et al (2022)

torchmd-net — Thölke and De Fabritiis (2022)

...

Equiformer(V2) – We can adapt techniques in computer vision and NLP to atomistic domain and achieve scalable accuracy. A top performer on OC20, OC22, ODAC,...

Open Catalysis 2020 Dataset (examples)



Predict energy, forces of given configurations and relaxed structures.

Graph attention built from tensor products of irrep features

EquiformerV2
ICLR 2024
(arXiv:2306.12059)

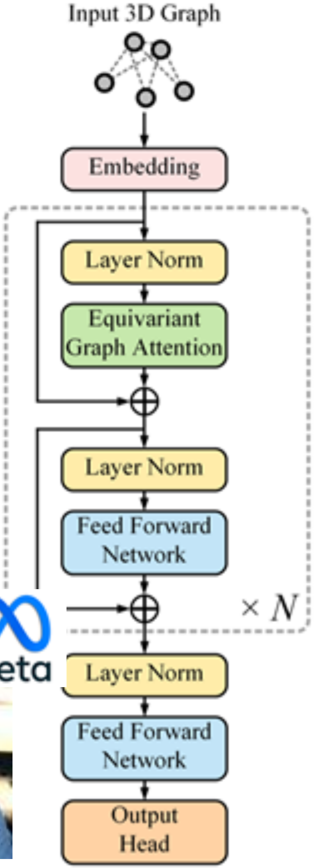
Equiformer:
Equivariant graph attention trans**former**
ICLR 2023
(arXiv:2206.11990)

First equivariant transformer to be state-of-art on multiple atomistic benchmarks (QM9, MD17, OC20).

Yi-Lun Liao

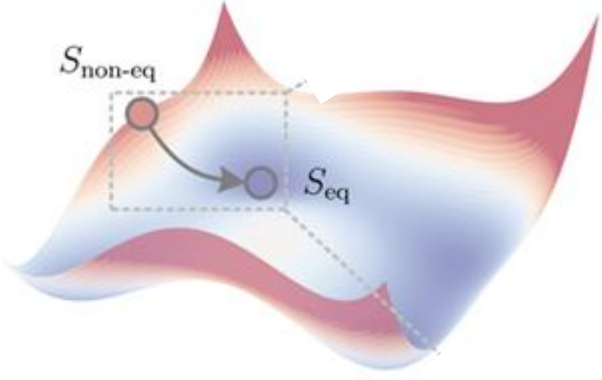
Abhishek Das

Brandon Wood



Generalizing Denoising to Non-Equilibrium Structures Improves Equivariant Force Fields
[TMLR 2024](), https://arxiv.org/abs/2403.09549

We can "augment" training of interatomic potentials by denoising non-equilibrium structures.



Potential energy surface

Yi-Lun Liao

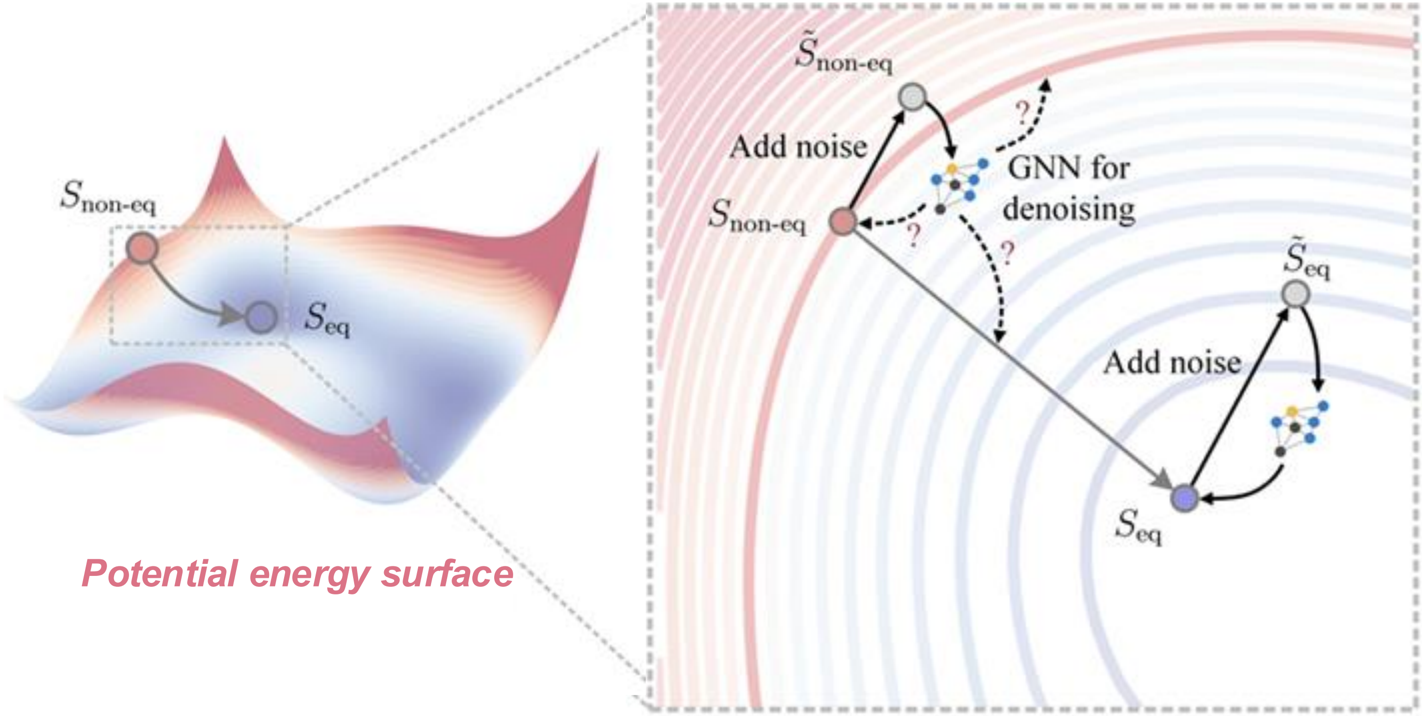
Abhishek Das

Muhammed Shuaib

Generalizing Denoising to Non-Equilibrium Structures Improves Equivariant Force Fields

[TMLR 2024](https://arxiv.org/abs/2403.09549), https://arxiv.org/abs/2403.09549

We can "augment" training of interatomic potentials by denoising non-equilibrium structures.



Yi-Lun Liao

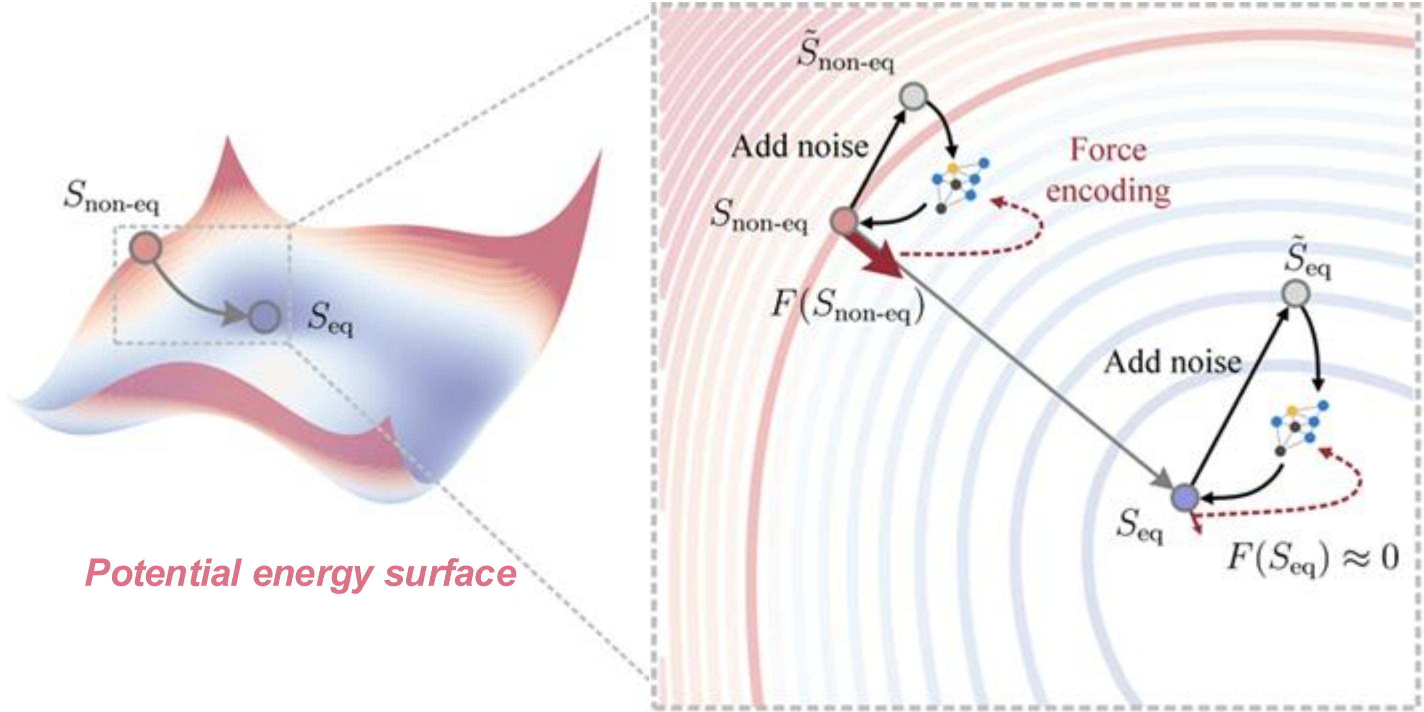
Abhishek Das

Muhammed Shuaib

Generalizing Denoising to Non-Equilibrium Structures Improves Equivariant Force Fields

[TMLR 2024](#), https://arxiv.org/abs/2403.09549

We can "augment" training of interatomic potentials by denoising non-equilibrium structures.



Yi-Lun Liao

Abhishek Das

Muhammed Shuaib

Generalizing Denoising to Non-Equilibrium Structures Improves Equivariant Force Fields

TMLR 2024, https://arxiv.org/abs/2403.09549

We can "augment" training of interatomic potentials by denoising non-equilibrium structures.

	EquiformerV2				EquiformerV2 + DeNS			
Epochs	forces	energy	Number of parameters	training time (GPU-hours)	forces	energy	Number of parameters	training time (GPU-hours)
12	20.46	285	83M	1398	19.09	269	89M	1501
20	19.78	280	83M	2330	18.58	260	89M	2501
30	19.42	278	83M	3495	18.02	251	89M	3752
	eSCN				eSCN + DeNS			
20	20.61	290	52M	1802	19.14	268	52M	1829

Force MAE in meV/Å
Energy MAE in meV

Yi-Lun Liao

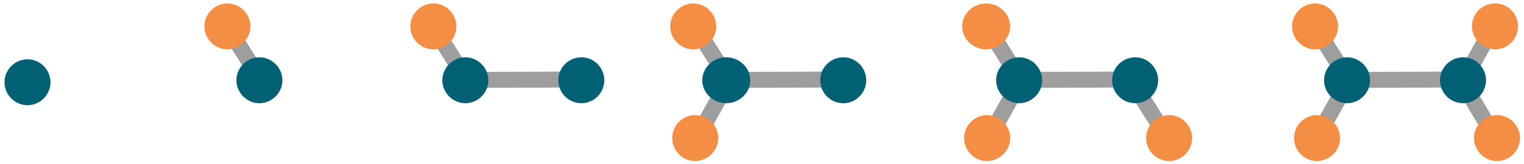
Abhishek Das

Muhammed Shuaib

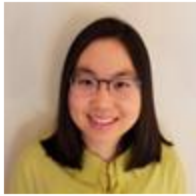
ENNs can also use spherical harmonics to efficiently represent spatial distributions

Symphony: (ICLR 2024, arXiv:2311.16199)

Symphony generates molecules one atom at a time…



Ameya Daigavane



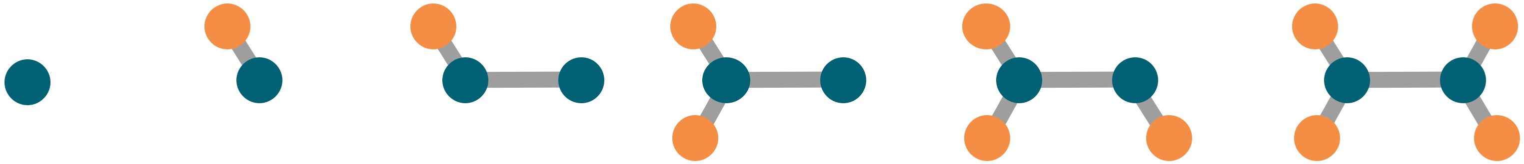
Song Kim

Mario Geiger

ENNs can also use spherical harmonics to efficiently represent spatial distributions

Symphony: (ICLR 2024, arXiv:2311.16199)

Symphony generates molecules one atom at a time...



and breaks each generation step into several predictions.

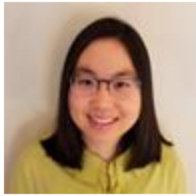
Sample ***"focus"*** for generation
Can also predict STOP

Sample new
"atom type"

Predict distribution of
possible positions

Sample
position

Ameya Daigavane



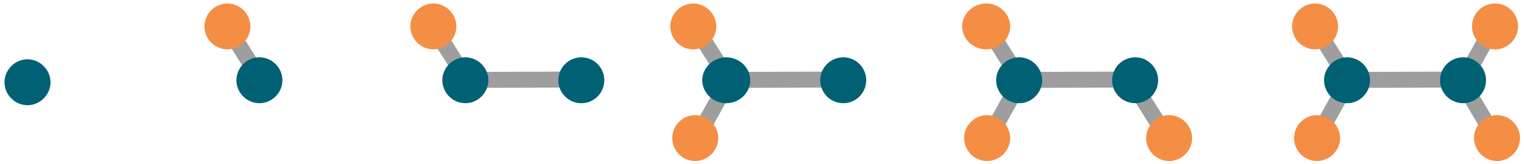
Song Kim

Mario Geiger

ENNs can also use spherical harmonics to efficiently represent spatial distributions

Symphony: (ICLR 2024, arXiv:2311.16199)

Symphony generates molecules one atom at a time...



and breaks each generation step into several predictions.

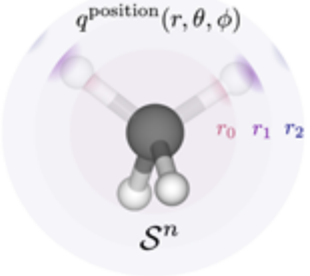
Sample ***"focus"*** for generation
Can also predict STOP

Sample new
"atom type"

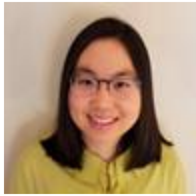
Predict distribution of
possible positions

Sample
position

Positions sampled from spherical harmonics distributions…
a natural datatype for ENNs.



Ameya Daigavane



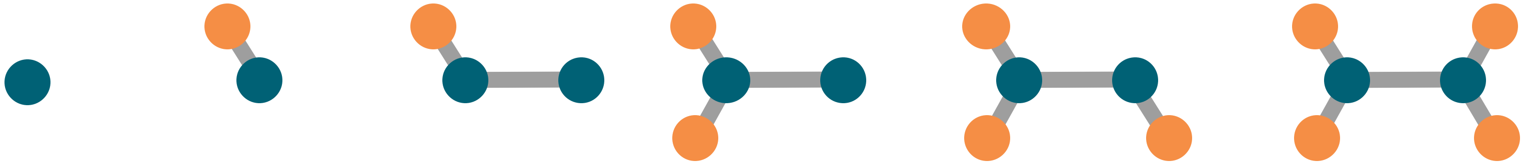
Song Kim

Mario Geiger

ENNs can also use spherical harmonics to efficiently represent spatial distributions

Symphony: (ICLR 2024, arXiv:2311.16199)

Symphony generates molecules one atom at a time...



and breaks each generation step into several predictions.

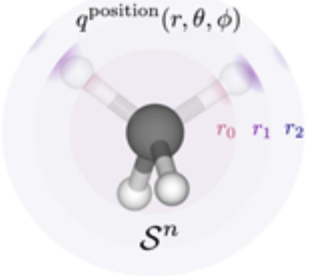
Sample ***"focus"*** for generation
Can also predict STOP

Sample new
"atom type"

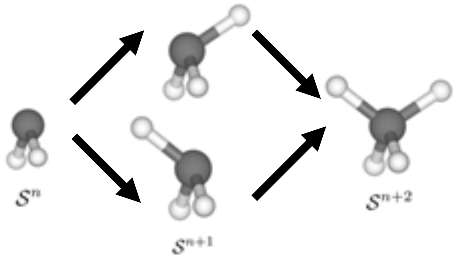
Predict distribution of
possible positions

Sample
position

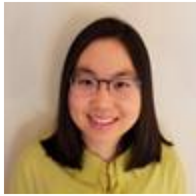
Positions sampled
from spherical harmonics
distributions…
a natural datatype for ENNs.



Sampling is
symmetric



Ameya Daigavane

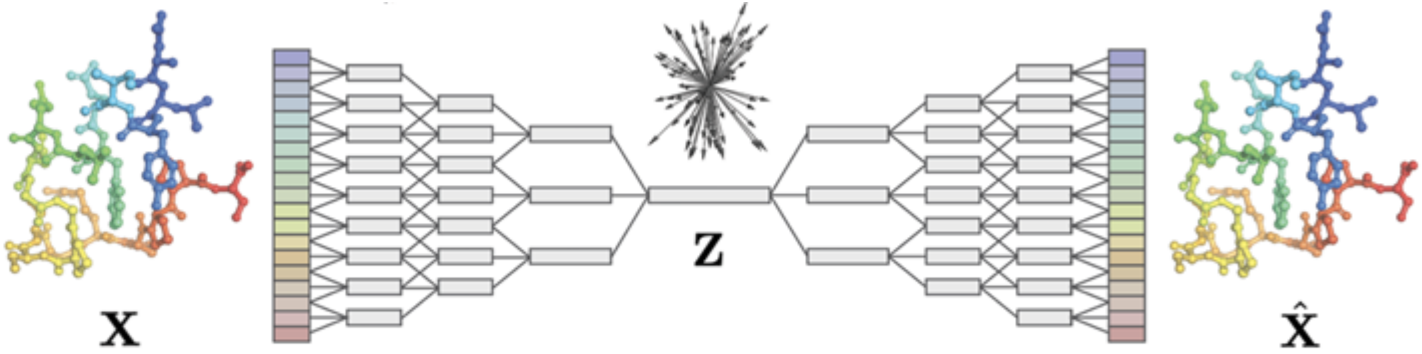


Song Kim

Mario Geiger

ENNs learn lossless, physical, coarsened representations of large atomic systems.

Ophiuchus: Hierarchical Coarse-Graining of Proteins (ICLR GEM Workshop 2024, arXiv:2310.02508)



Iteratively coarsen along backbone.

Iteratively build up backbone.

Allan Costa

Ilan Mitnikov

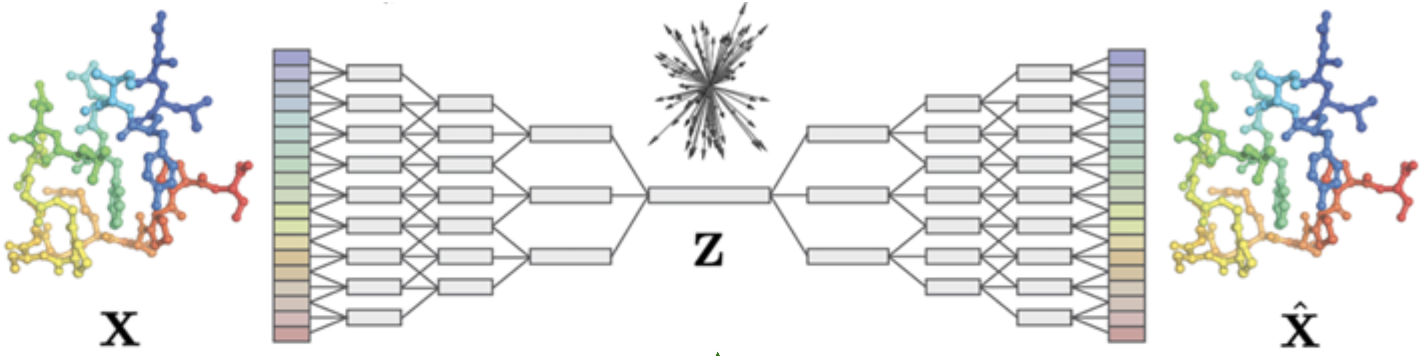
Mario Geiger

Manvitha Ponnapati

Joe Jacobson

ENNs learn lossless, physical, coarsened representations of large atomic systems.

Ophiuchus: Hierarchical Coarse-Graining of Proteins (ICLR GEM Workshop 2024, arXiv:2310.02508)



Iteratively coarsen along backbone.

Latent has no geometry, just tensor features.

Can generate latent with separate diffusion model.

Iteratively build up backbone.

Allan Costa

Ilan Mitnikov

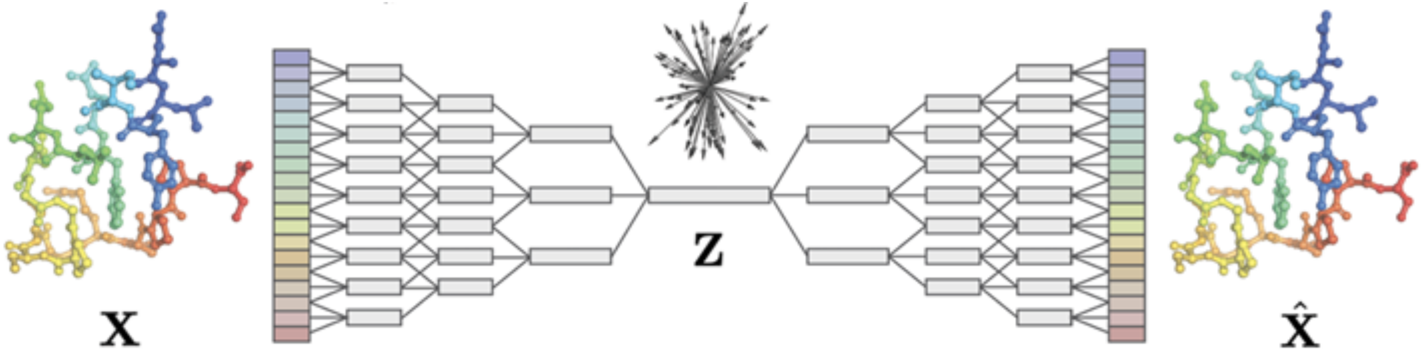
Mario Geiger

Manvitha Ponnapati

Joe Jacobson

ENNs learn lossless, physical, coarsened representations of large atomic systems.

Ophiuchus: Hierarchical Coarse-Graining of Proteins (ICLR GEM Workshop 2024, arXiv:2310.02508)



Allan Costa

Ilan Mitnikov

Mario Geiger

Manvitha Ponnapati

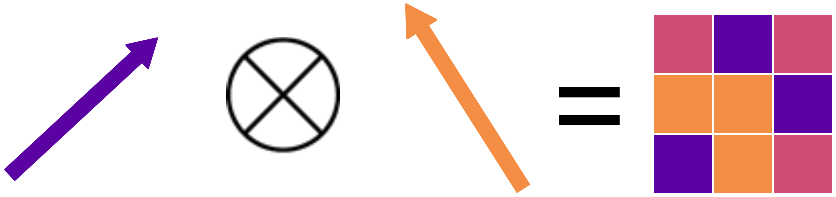
Joe Jacobson

Latent interpolation (z) between protein conformations are physical.

The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

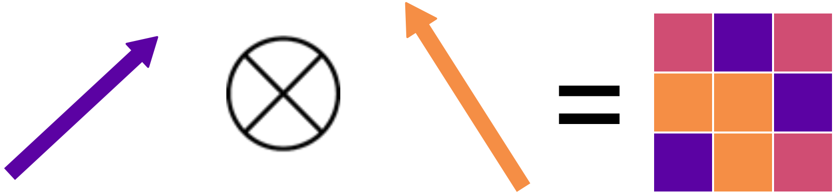
The most general way to interact equivariant objects (like vectors) is the outer product.



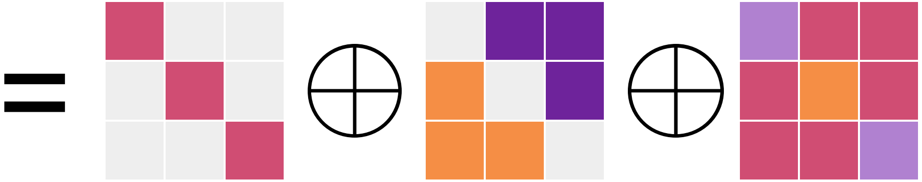
Not a scalar or vector.
Does not have a single angular frequency.
Can be decomposed into "irreps".

The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

The most general way to interact equivariant objects (like vectors) is the outer product.



Not a scalar or vector.
Does not have a single angular frequency.
Can be decomposed into "irreps".



dot product
trace
invariant
L=0, parity=even
1 degree of freedom

cross-product
antisymmetric
equivariant
L=1, parity=even
3 degrees of freedom

symmetric
traceless
equivariant
L=2, parity=even
5 degrees of freedom

The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

Most general tensor product does *<u>not scale favorably</u>*.
It preserves dimension.

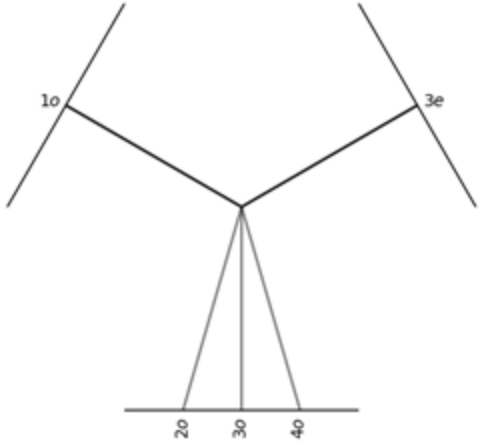
$$\mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}^{N \times M}$$

The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

Most general tensor product does *not scale favorably*.
It preserves dimension.

$$\mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^{N \times M}$$

$1o \otimes 3e$
3 x 7 = 21 dof, 3 paths



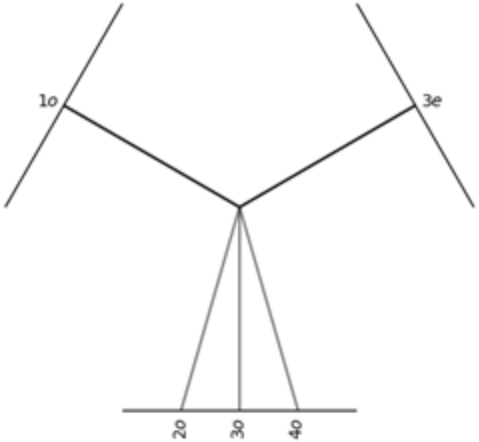
The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

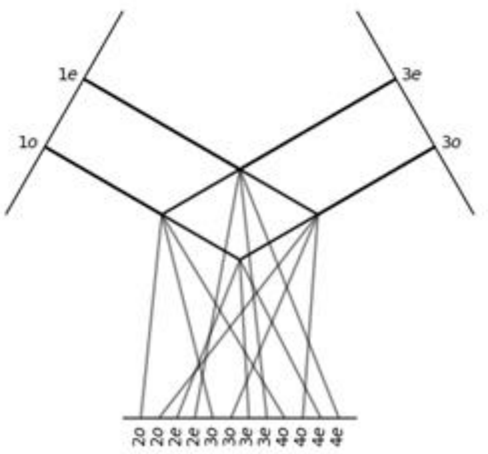
Most general tensor product does *not scale favorably*.
It preserves dimension.

$$\mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}^{N \times M}$$

1o ⊗ 3e
3 x 7 = 21 dof, 3 paths

(1o ⊕ 1e) ⊗ (3e ⊕ 3o)
6 x 14 = 84 dof, 12 paths



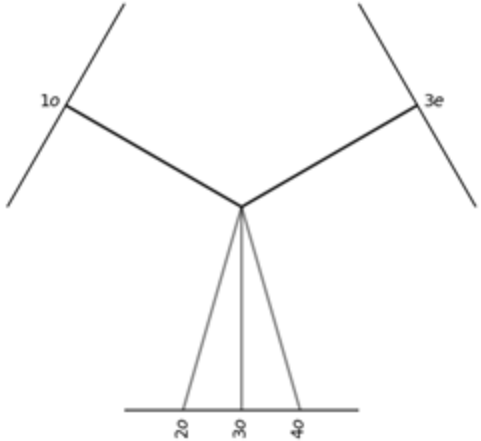


The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

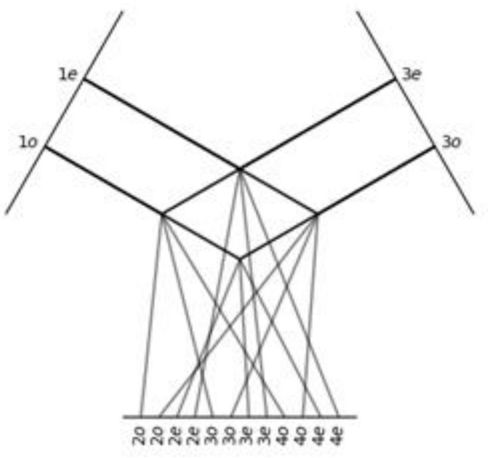
Most general tensor product does *<u>not scale favorably</u>*.
It preserves dimension.

$$\mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}^{N \times M}$$

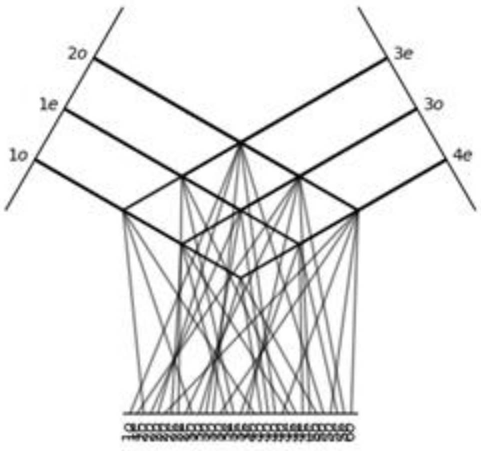
$1o \otimes 3e$
3 x 7 = 21 dof, 3 paths



$(1o \oplus 1e) \otimes (3e \oplus 3o)$
6 x 14 = 84 dof, 12 paths



$(1o \oplus 1e + 2o) \otimes (3e \oplus 3o + 4e)$
11 x 23 = 253 dof, 33 paths



The catch? There's an abstraction vs. implementation gap.
Biggest pain point? Equivariant "multiplication" == tensor product *decomposition*

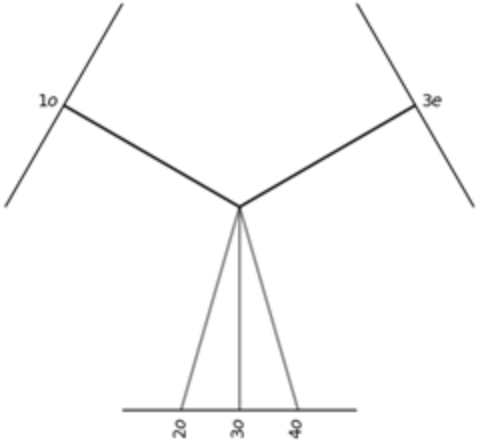
Most general tensor product does *not scale favorably*.
It preserves dimension.

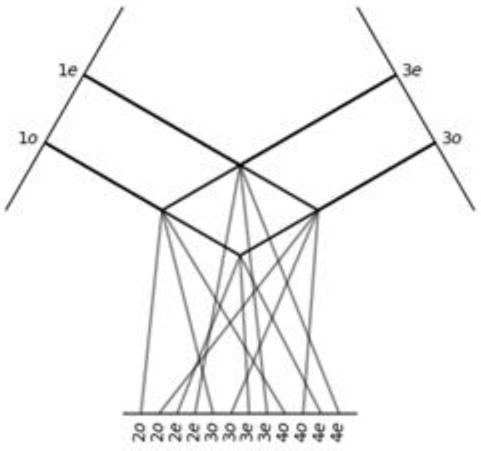
$$\mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^{N \times M}$$

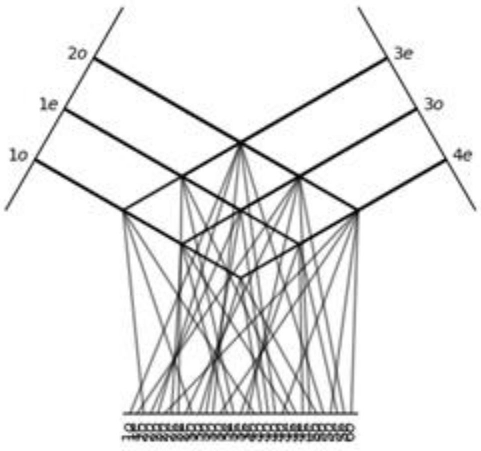
1o ⊗ 3e
3 x 7 = 21 dof, 3 paths

(1o ⊕ 1e) ⊗ (3e ⊕ 3o)
6 x 14 = 84 dof, 12 paths

(1o ⊕ 1e + 2o) ⊗ (3e ⊕ 3o + 4e)
11 x 23 = 253 dof, 33 paths







In order to maintain equivariance, <u>we can only throw out entire "paths"</u>
(path == set of 3 lines connecting irrep 1 in, irrep 2 in and irrep out).

How to scale?
How to prune?

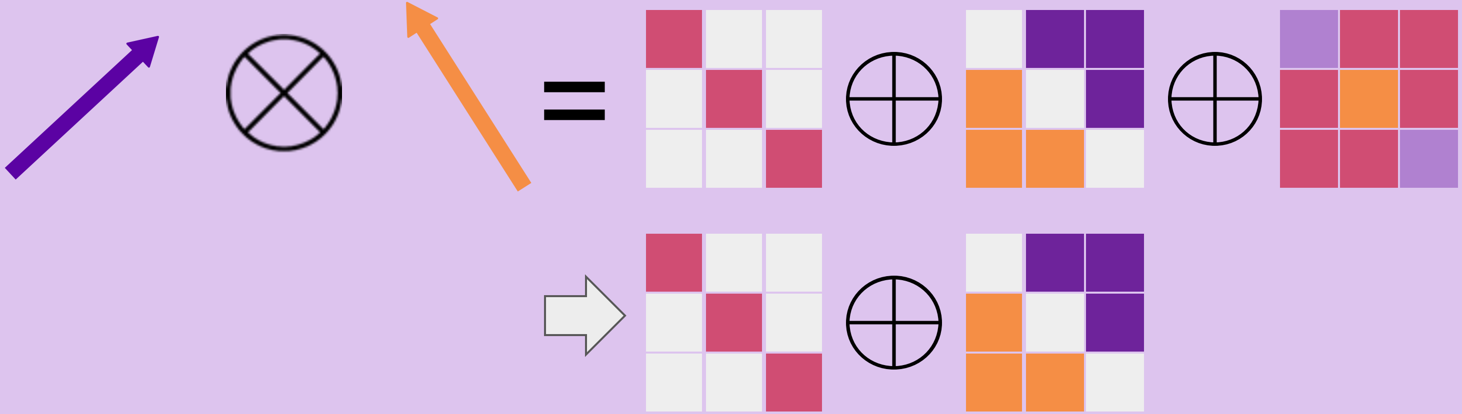
The catch? There’s an abstraction vs. implementation gap.
Biggest pain point? Equivariant “multiplication” == tensor product *decomposition*

Most general tensor product does *not scale favorably*.
It preserves dimension.

$$\mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}^{N \times M}$$

1o ⊗ 3 … ⊕ 3o + 4e)
3 x 7 … ns

Example with dropping “higher order paths”...



In order to maintain equivariance, <u>we can only throw out entire “paths”</u>
(path == set of 3 lines connecting irrep 1 in, irrep 2 in and irrep out).

How to scale?
How to prune?

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations

YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt

https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

YuQing Xie

Mit Kotak

Ameya Daigavane

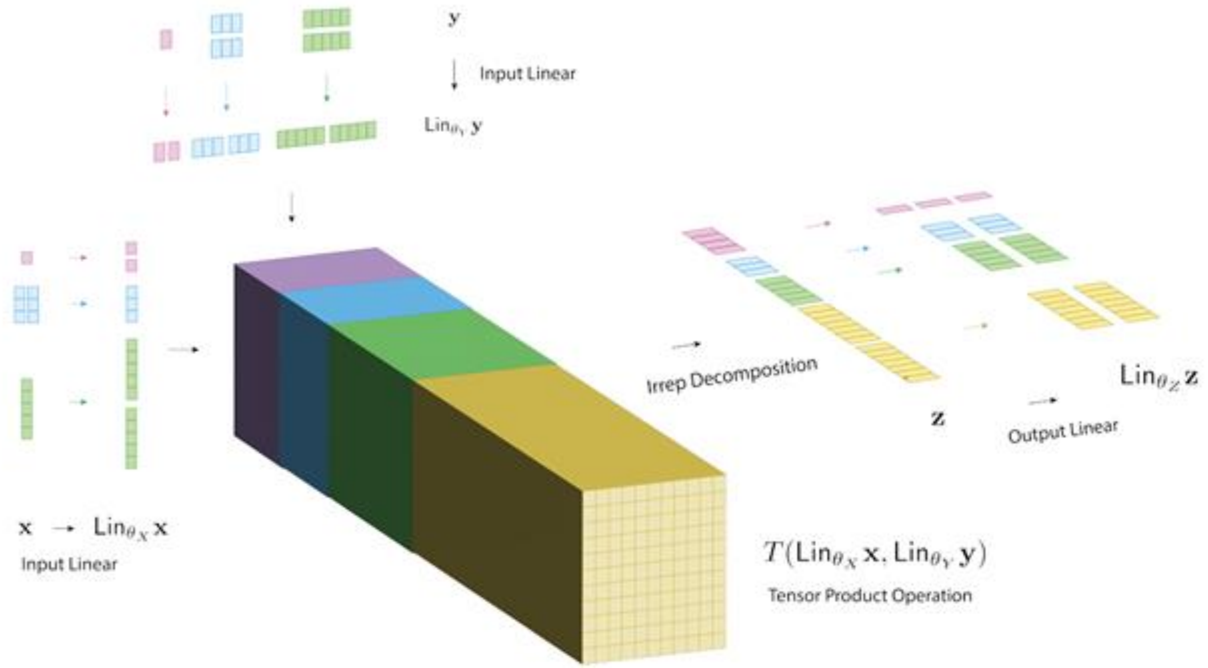


Diagram of "General" / **Clebsch-Gordan Tensor Product** (**CGTP**) *without* visualizing sparsity.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations

YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt

https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

YuQing Xie

Mit Kotak

Ameya Daigavane

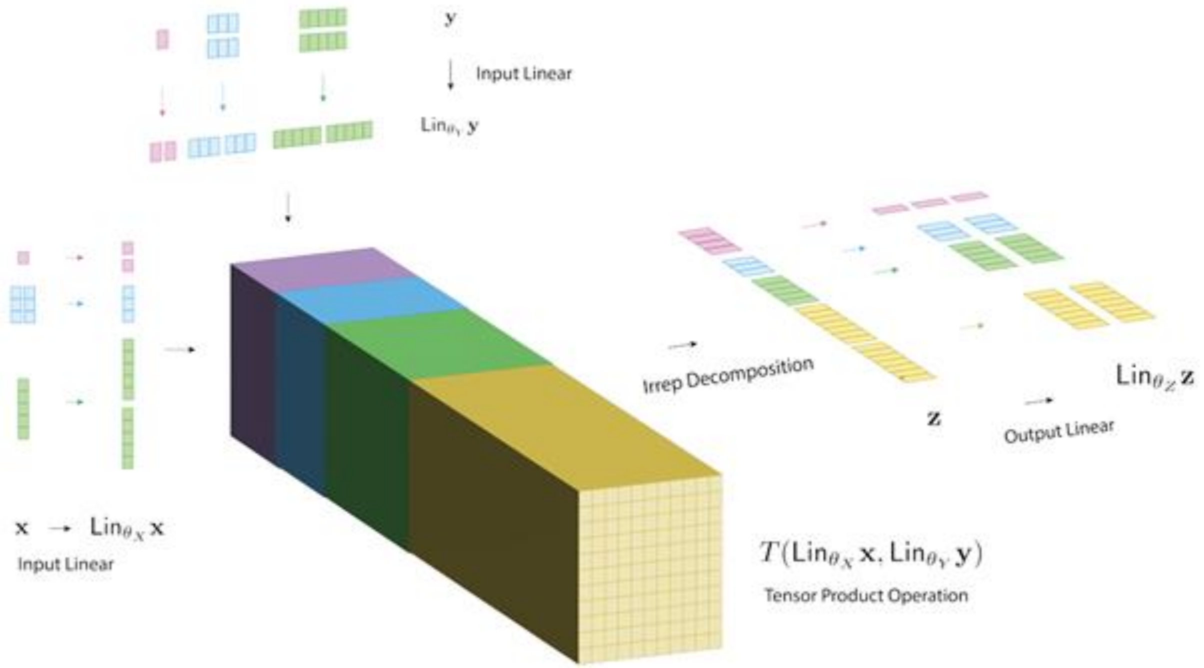


Diagram of "General" / **Clebsch-Gordan Tensor Product** (**CGTP**) *without* visualizing sparsity.

Spoiler: Any proposed "faster TP" can be emulated with a CGTP with specific dropped paths or path summing.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt
https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Gaunt Tensor Product (GTP)
Luo, Chen, Krishnapriyan
https://openreview.net/forum?id=mhyQXJ6JsK
ICLR 2024 Spotlight

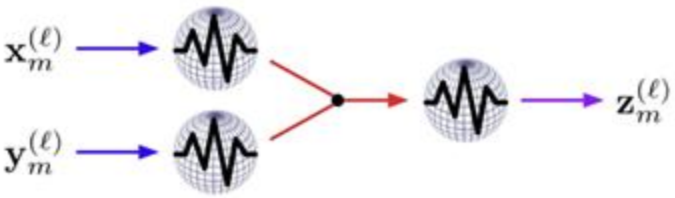


Figure 2. Schematic of GTP. We interpret input irreps as scalar SH coefficients to create spherical signals. We then take pointwise products of the two signals to create a new signal which we decompose back into scalar SH coefficients.

Matrix Tensor Product (MTP)
Unke, Maennel
In e3x: https://arxiv.org/abs/2401.07595

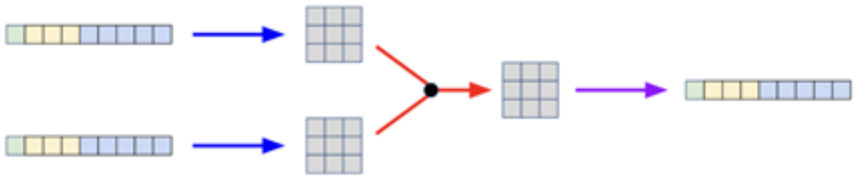


Figure 3. Schematic of the process in taking a matrix tensor product. We embed input irreps into a tensor product rep. We then interact using matrix multiplication before decomposing the resulting tensor product rep back into a direct sum of irreps.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt
https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Gaunt Tensor Product (GTP)
Luo, Chen, Krishnapriyan
https://openreview.net/forum?id=mhyQXJ6JsK
ICLR 2024 Spotlight

Matrix Tensor Product (MTP)
Unke, Maennel
In e3x: https://arxiv.org/abs/2401.07595

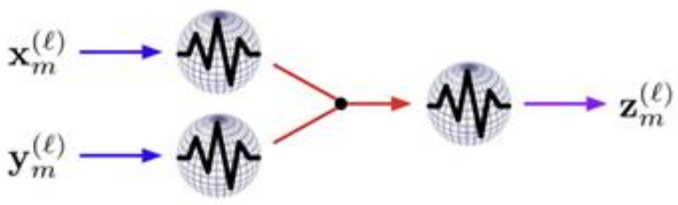


Figure 2. Schematic of GTP. We interpret input irreps as scalar SH coefficients to create spherical signals. We then take pointwise products of the two signals to create a new signal which we decompose back into scalar SH coefficients.

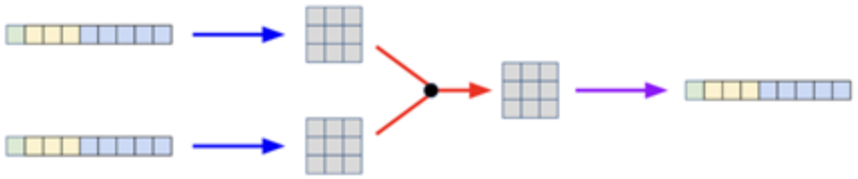


Figure 3. Schematic of the process in taking a matrix tensor product. We embed input irreps into a tensor product rep. We then interact using matrix multiplication before decomposing the resulting tensor product rep back into a direct sum of irreps.

GTP lacks antisymmetric interactions (e.g., cannot represent cross products), which makes it unable to classify chiral structures.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt
https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Gaunt Tensor Product (GTP)
Luo, Chen, Krishnapriyan
https://openreview.net/forum?id=mhyQXJ6JsK
ICLR 2024 Spotlight

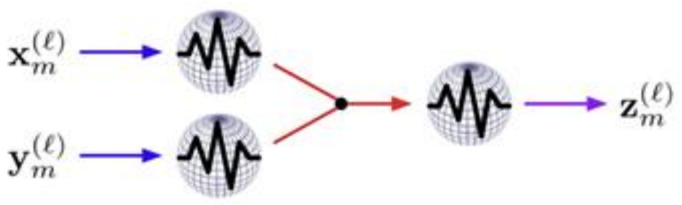


Figure 2. Schematic of GTP. We interpret input irreps as scalar SH coefficients to create spherical signals. We then take pointwise products of the two signals to create a new signal which we decompose back into scalar SH coefficients.

GTP lacks antisymmetric interactions (e.g., cannot represent cross products), which makes it unable to classify chiral structures.

Matrix Tensor Product (MTP)
Unke, Maennel
In e3x: https://arxiv.org/abs/2401.07595

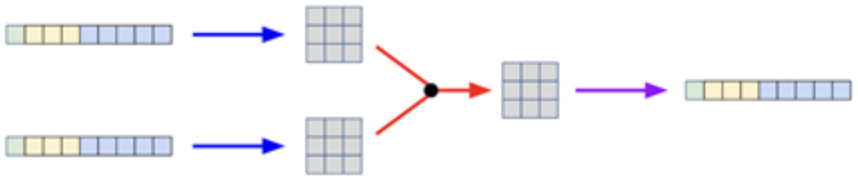


Figure 3. Schematic of the process in taking a matrix tensor product. We embed input irreps into a tensor product rep. We then interact using matrix multiplication before decomposing the resulting tensor product rep back into a direct sum of irreps.

MTP can represent antisymmetric interactions but merges same-type outputs, limiting expressive granularity.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations

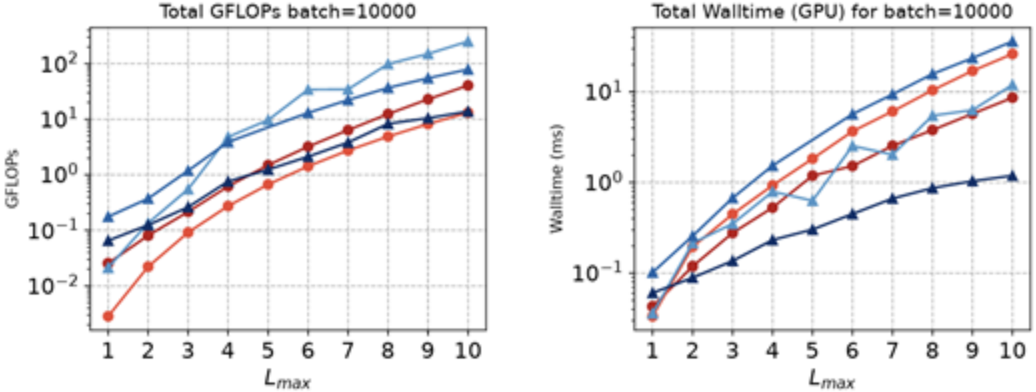
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt

https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

YuQing Xie

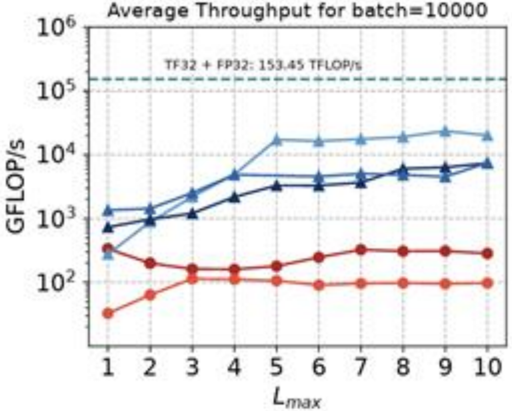
Mit Kotak

Ameya Daigavane



Many metrics to use to profile and benchmark.

Which you use matters.



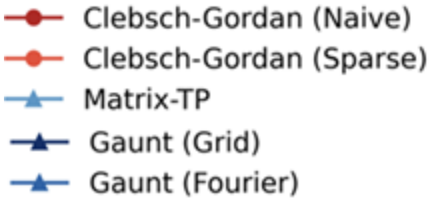


Figure 4. Top: Analysis of tensor products compute scaling on a RTX A5500 GPU: Total GFLOPs (Left), Total walltime (Middle), and Average throughput in GFLOPs/s (Right). Bottom: Analysis of tensor products compute scaling per path on RTX A5500 GPU: (Left) Total Walltime / Expressivity, (Right) Total GFLOPs / Expressivity. Batch refers to the number of tensor products performed in parallel.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations

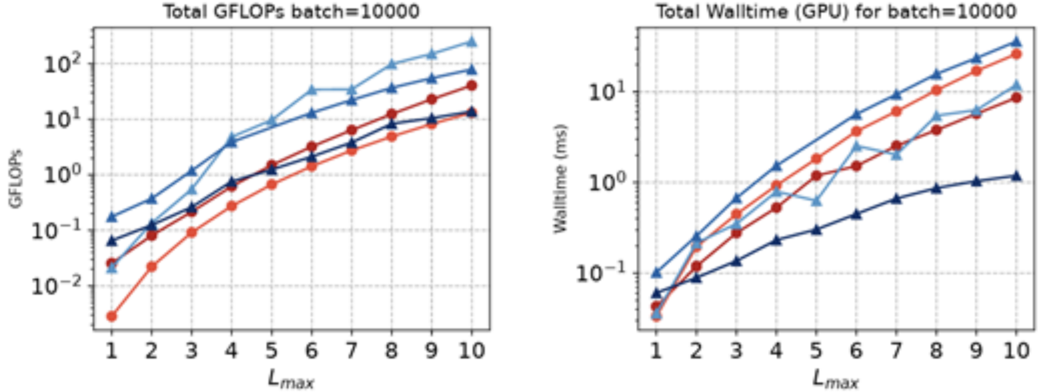
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt

https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

YuQing Xie

Mit Kotak

Ameya Daigavane



"Normalized" here means Normalized by number of paths – proxy for expressivity.

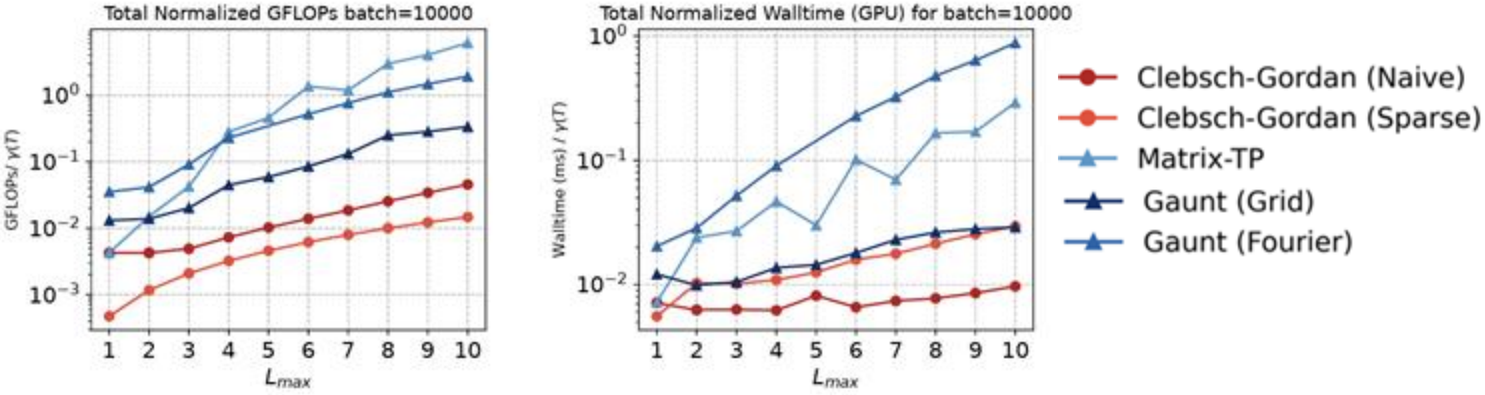


Figure 4. Top: Analysis of tensor products compute scaling on a RTX A5500 GPU: Total GFLOPs (Left), Total walltime (Middle), and Average throughput in GFLOPs/s (Right). Bottom: Analysis of tensor products compute scaling per path on RTX A5500 GPU: (Left) Total Walltime / Expressivity, (Right) Total GFLOPs / Expressivity. Batch refers to the number of tensor products performed in parallel.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations

YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt

https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Tensor Product Operation	Expressivity	Runtime	Runtime / Expressivity
CGTP (Naive)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^6)$	$\mathcal{O}(L^3)$
CGTP (Sparse)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^5)$	$\mathcal{O}(L^2)$
GTP (Fourier)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (Grid)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (S2FFT)	$\mathcal{O}(L)$	$\mathcal{O}(L^2 \log^2 L)$	$\mathcal{O}(L \log^2 L)$
MTP (Naive)	$\mathcal{O}(L)$	$\mathcal{O}(L^4)$	$\mathcal{O}(L^3)$
MTP (Sparse)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$

Asymptotic runtime normalized by expressivity is similar for all methods.

BUT, asymptotic runtime does not predict practical performance due to
e.g. GPU utilization and FLOPs.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt
https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Tensor Product Operation	Expressivity	Runtime	Runtime / Expressivity
CGTP (Naive)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^6)$	$\mathcal{O}(L^3)$
CGTP (Sparse)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^5)$	$\mathcal{O}(L^2)$
GTP (Fourier)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (Grid)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (S2FFT)	$\mathcal{O}(L)$	$\mathcal{O}(L^2 \log^2 L)$	$\mathcal{O}(L \log^2 L)$
MTP (Naive)	$\mathcal{O}(L)$	$\mathcal{O}(L^4)$	$\mathcal{O}(L^3)$
MTP (Sparse)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$

Asymptotic runtime normalized by expressivity is similar for all methods.

BUT, asymptotic runtime does not predict practical performance due to e.g. GPU utilization and FLOPs.

CGTP has low GPU utilization but, when normalized for expressivity, is still the most efficient.

GTP performs better in wall-clock time due to higher GPU efficiency.

MTP takes care of path dropping implicitly.

The Price of Freedom: Exploring Tradeoffs in Equivariant Tensor Product Operations
YuQing Xie, Ameya Daigavane, Mit Kotak, Tess Smidt
https://openreview.net/forum?id=EvIwwGYTLc (ICML 2025)

Tensor Product Operation	Expressivity	Runtime	Runtime / Expressivity
CGTP (Naive)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^6)$	$\mathcal{O}(L^3)$
CGTP (Sparse)	$\mathcal{O}(L^3)$	$\mathcal{O}(L^5)$	$\mathcal{O}(L^2)$
GTP (Fourier)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (Grid)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$
GTP (S2FFT)	$\mathcal{O}(L)$	$\mathcal{O}(L^2 \log^2 L)$	$\mathcal{O}(L \log^2 L)$
MTP (Naive)	$\mathcal{O}(L)$	$\mathcal{O}(L^4)$	$\mathcal{O}(L^3)$
MTP (Sparse)	$\mathcal{O}(L)$	$\mathcal{O}(L^3)$	$\mathcal{O}(L^2)$

Asymptotic runtime normalized by expressivity is similar for all methods.

BUT, asymptotic runtime does not predict practical performance due to e.g. GPU utilization and FLOPs.

CGTP has low GPU utilization but, when normalized for expressivity, is still the most efficient.

GTP performs better in wall-clock time due to higher GPU efficiency.

MTP takes care of path dropping implicitly.

Code for paper: https://github.com/atomicarchitects/PriceofFreedom
Efforts on compiling strategies:
cuEquivariance | openEquivariance | “newquip”

Thanks to the group…

Atomic Architects

MIT EECS

rle AT MIT RESEARCH LABORATORY OF ELECTRONICS AT MIT

Thanks to the group, our collaborators, …

Abhishek Das

Brandon Wood

Muhammed Shuaibi

Mario Geiger

Joe Jacobson
Ilan Mitnikov
Manvitha Ponnapati

Thanks to the group, our collaborators, and our funding!

DOE ICDI grant
DE-SC0022215

DOE CSGF

The NSF Institute for Artificial Intelligence and Fundamental Interactions

GRFP
NSF GRADUATE RESEARCH FELLOWSHIP PROGRAM

CSGRAD4US & mentoring program
Computer and Information Science and Engineering Graduate Fellowships

PHY 2019786

AFOSR
Young Investigator
Program

MIT **SuperUROP**

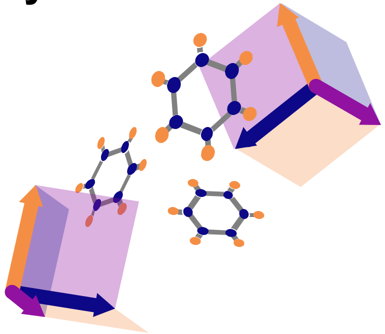
MathWorks

ANALOG DEVICES

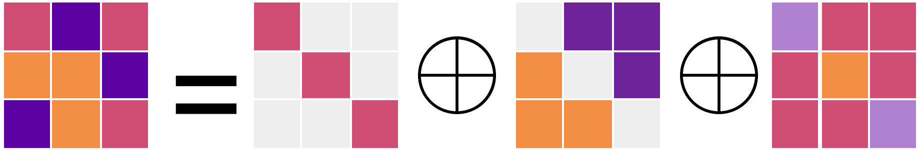
X.Org Foundation

TAKEAWAYS

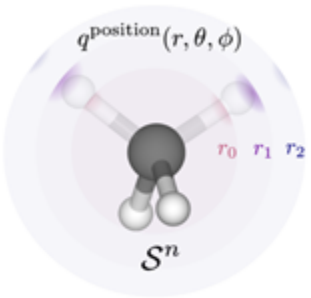
Euclidean neural networks are built with the powerful assumption that atomic systems exist in 3D Euclidean space.

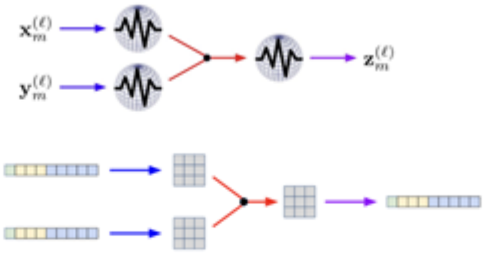


Symmetry induces data types (irreps), rules of interaction (tensor products), and other emergent properties (degeneracy).



There are many ways to use these properties to build interesting "modules".





Many opportunities for design and optimization!

- **Tensor product ops that balance expressivity and scalability**
- **Generative models**
- **…**

Slides: https://tinyurl.com/2025-aims-tess

Tess Smidt | tsmidt@mit.edu | **Atomic Architects**