Rotation Invariant Machine Learning for Highly Symmetric Molecular Configurations

Student: Logan Bolton

Mentor: Tomáš Karella
Mentor: Roxana Bujack
Mentor: Nicholas Lubbers

08/05/2026

LA-UR-26-27984

About Me

- **Grew up in Alabama**
- **Graduated from Auburn University in May 2026**
  - Bachelor's in Computer Science
  - Minor in Statistics
- **Starting a Master's in Computer Science**
  - New York University in Fall 2027

Hike at Wheeler Peak in Taos

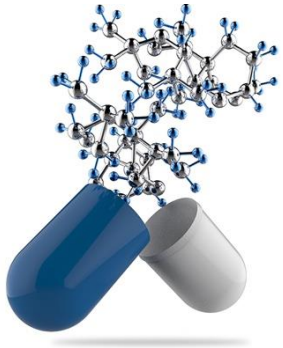
Hike at Valles Calderas

Proud holder of a Los Alamos public library card

Motivation

- **How can we accurately simulate molecules?**
  - Given a set of atoms, want to predict how they will act

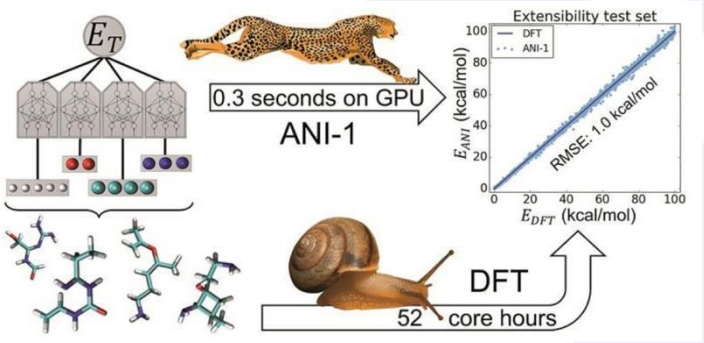
- **Applications:**
  1. Improved drug design
  2. Discovering new materials
  3. Study chemical reactions



Challenges

1. **High quality quantum simulations are VERY slow**
   - Neural networks can approximate quality for orders of magnitude lower cost!

2. **Neural networks don't know physics**
   - If we give models good priors about the physical world, training them becomes much more efficient

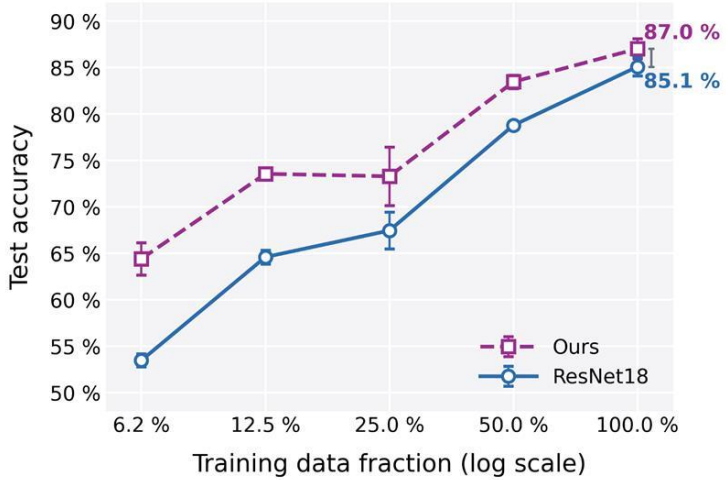


ANAKIN-ME: Accurate NeurAl networK engINe for Molecular Energies

Choosing a Good Physical Prior

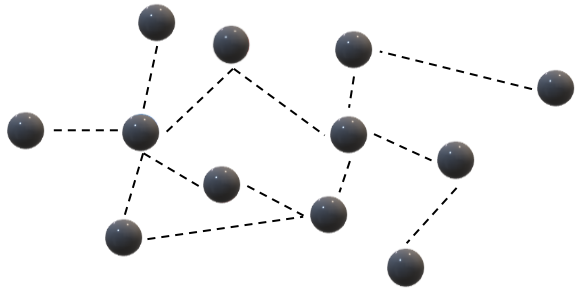
- **By default, neural networks don't understand 3D rotation**
  - If you rotate a molecule, the model will not recognize it
  - Want the model to understand that the global rotation is unimportant

- **Rotation Invariance**
  - Representation does NOT change with rotation
  - Improves training efficiency and accuracy
  - $f(g \cdot x) = f(x)$

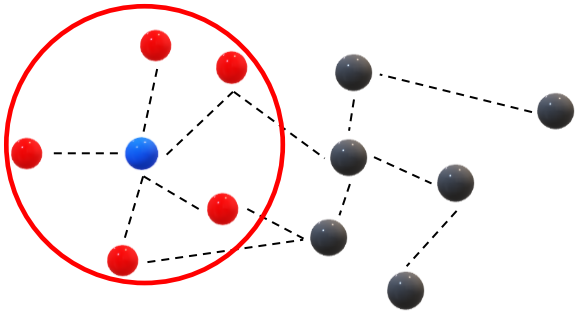


Invariant model image classification is better than the baseline model even with 50x less parameters

Building Invariant Representations

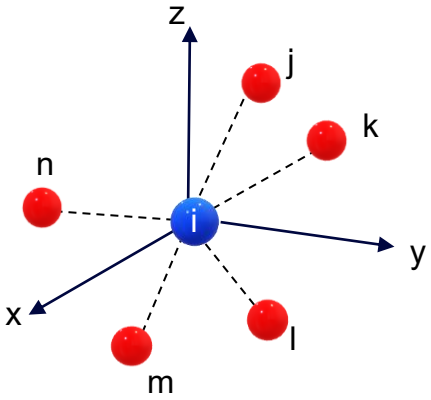


Building Invariant Representations

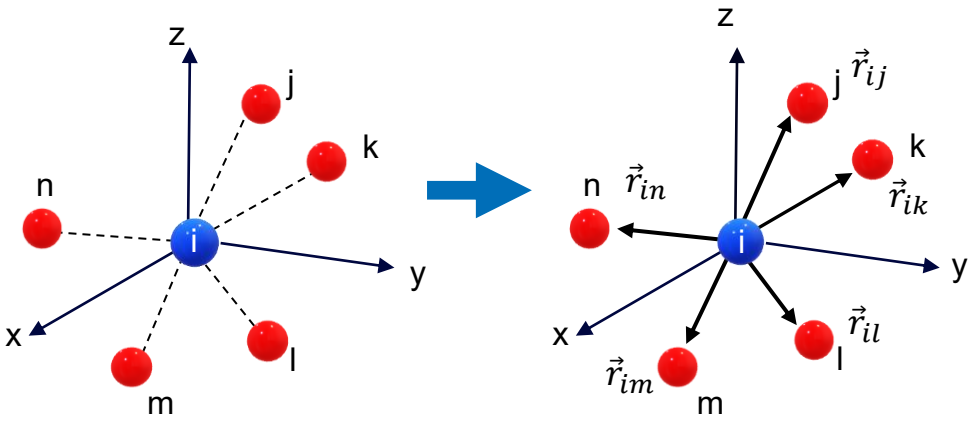


Find Local Neighbors

Building Invariant Representations

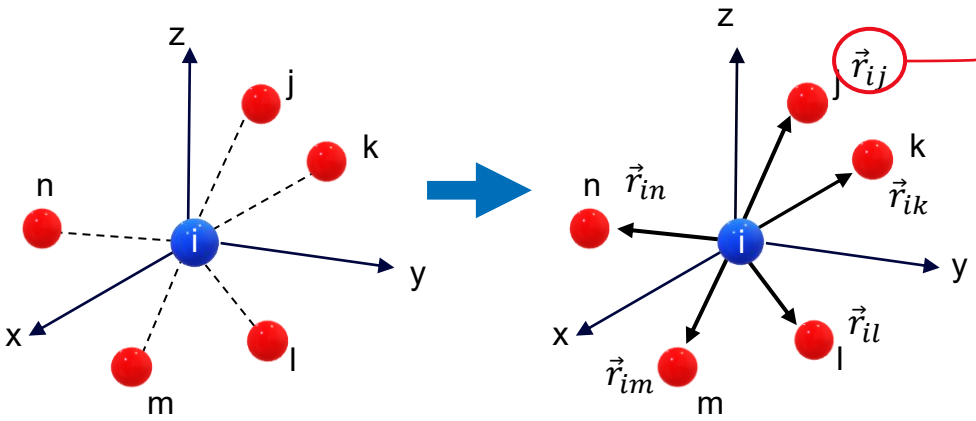


Building Invariant Representations



Get direction information of neighbors

Building Invariant Representations



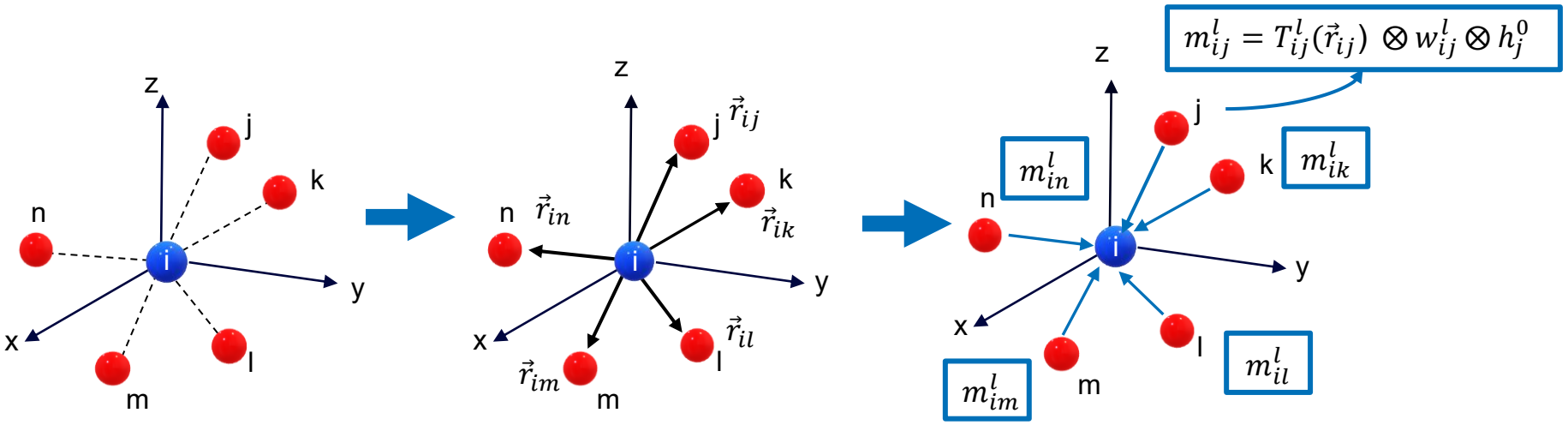
$T_{ij}^{0}(\vec{r}_{ij})$,$T_{ij}^{1}(\vec{r}_{ij})$, $T_{ij}^{2}(\vec{r}_{ij})$, …

Cartesian Tensors

How much geometric information do you want to keep?

Parameter Value	Information
$L_{\max} = 0$	Only keep distance information
$L_{\max} = 1$	Keep shape information
$L_{\max} = 2$	Keep *more* shape information - spread of neighbors

Building Invariant Representations



Weighted Contribution

Atom features

Atom type (carbon, hydrogen, etc.). and other existing information

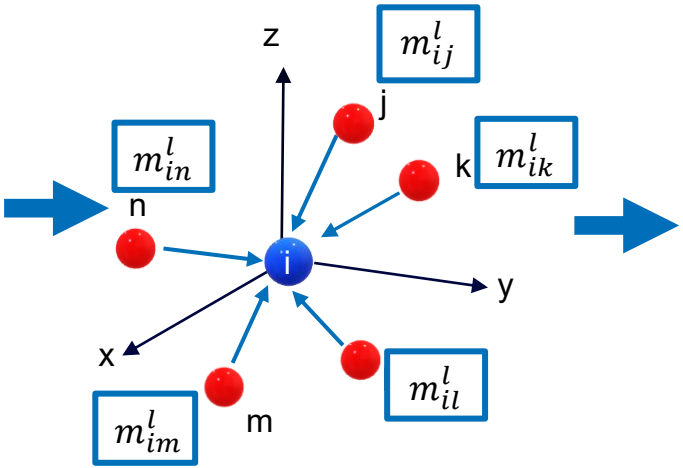
Spatial Information

Angles, distances and other high dimensional geometric information

Learnable Weights

w_{ij}^{l} - learnable weights per atom and tensor type

Building Invariant Representations

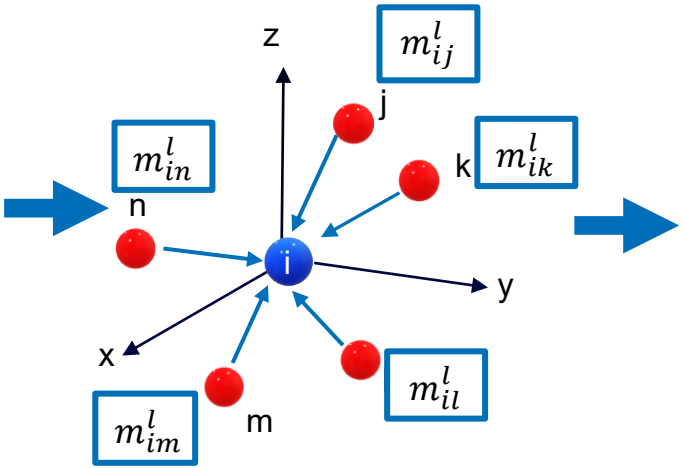


Combine neighbor information into center node

Not Invariant

$[M^{(0)}, M^{(1)}, M^{(2)}, M^{(3)}]$

Building Invariant Representations



$$\left[M^{(0)}, M^{(1)}, M^{(2)}, M^{(3)}\right]$$

Contract tensors:
$$M^{(k)} \oplus M^{(j)} = Scalar$$

Parameter 2: $n_{\max}$
Determines how many tensors contribute to the contraction

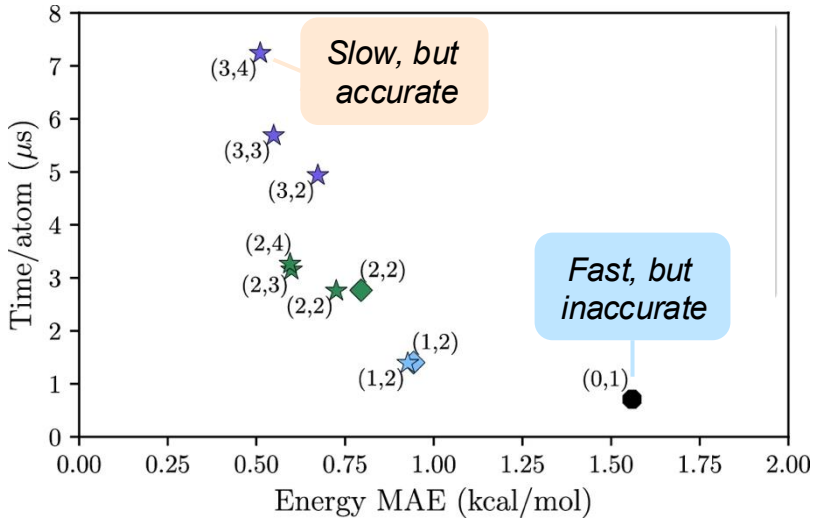
Invariant Features!

$$\left[I^{(0)}, I^{(1)}, I^{(2)}, I^{(3)}, I^{(4)}, I^{(5)}, \ldots\right]$$

Tradeoffs in Representation Quality

What set (basis) of polynomials is best?:

1. **Overly simplified representation**
   - *Ex: model can see relative distance but not angles*
   - Very fast, but high prediction error
2. **Extremely detailed representation**
   - High values for $L_{\max}$ and $n_{\max}$
   - Will take months to run

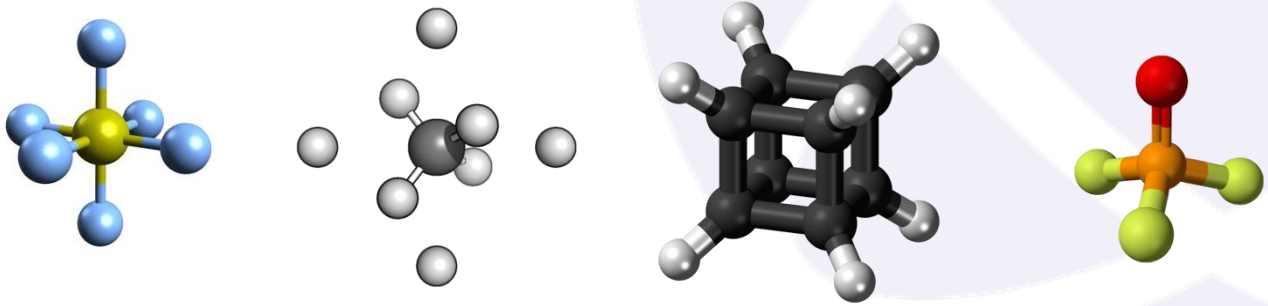


Tradeoffs in Representation Quality

- **Can we get speed AND accuracy?**
  - Basis that gives high amount of detail, but is not slow
- **Where and why do bases fail?**
  - Can we fix those issues with minor changes to latency?

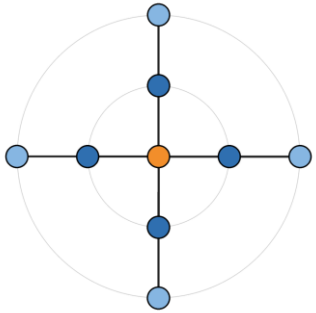
Stress Testing Representation Failures

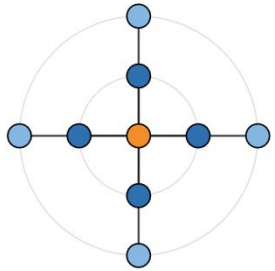
- **Theory tells us that representation should fail to represent symmetrical configurations**
- **Many molecules in nature have high amounts of symmetry**
  - Ex: methane, cubane, sulfur hexafluoride and phosphoryl fluoride
- **Want to find out where bases fail with highly symmetrical molecules**
  - Need simple benchmark



Stress Testing Representation Failures

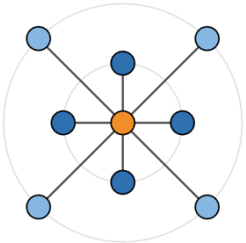
- **Dataset:**
  - Create n-fold symmetries by rotating two sets of ring around a central node
  - Rings are either close together or far apart
- **Task:**
  - Binary classification
  - Assign the correct close or far labels to the graph





Class 1: Close

Rotated outer ring

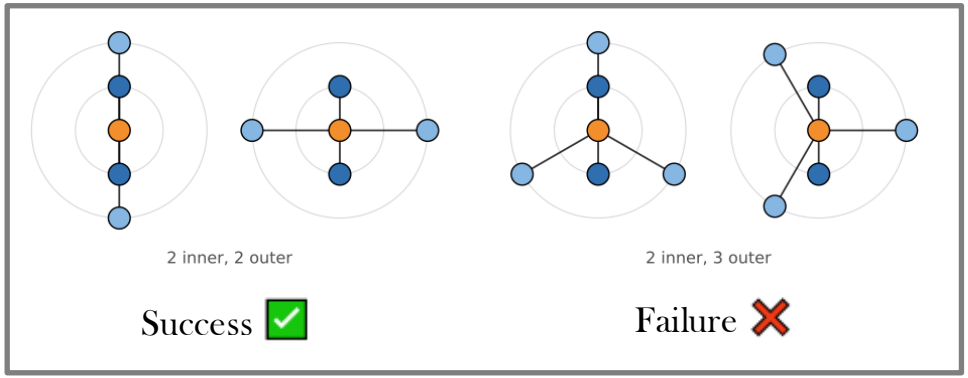


Class 2: Far

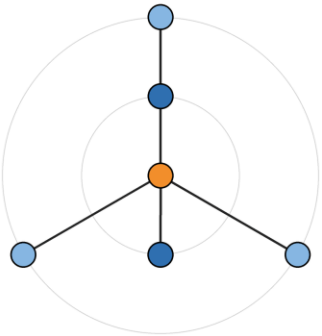
Best basis fails on the majority of the graph pairs

	(1, 3)	(1, 4)	(2, 2)	(2, 3)	(2, 4)	(3, 3)	(3, 4)	(4, 4)
EquiformerV3	1.0	0.5	1.0	0.5	0.5	1.0	0.5	0.5
HIP-HOP ($n_{\max} = 4$)	1.0	0.5	1.0	0.5	0.5	1.0	0.5	0.5

Graph configuration (inner-ring nodes, outer-ring nodes)



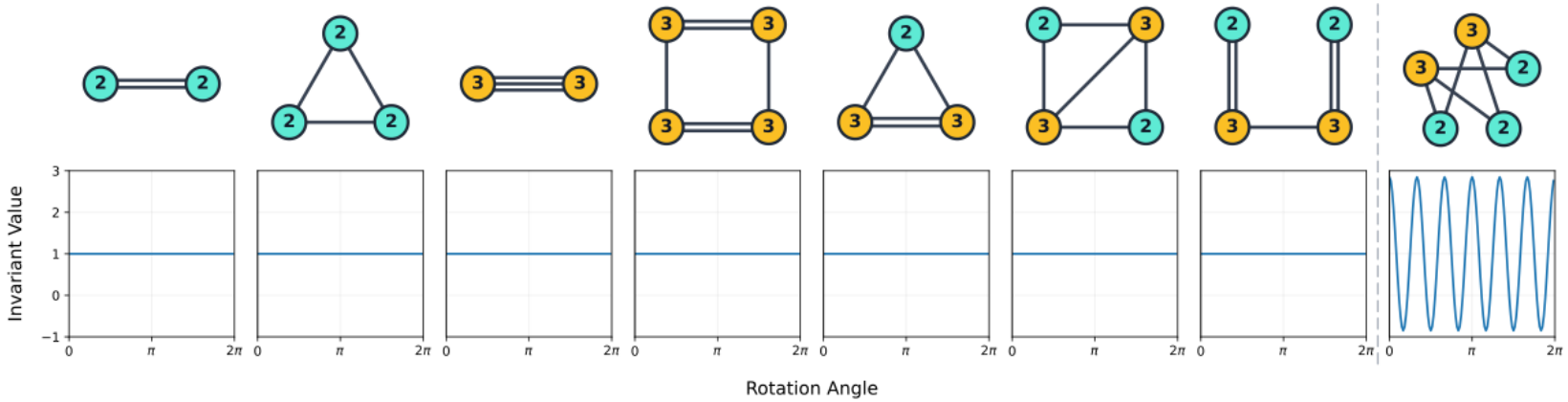
How to Improve the Representation?



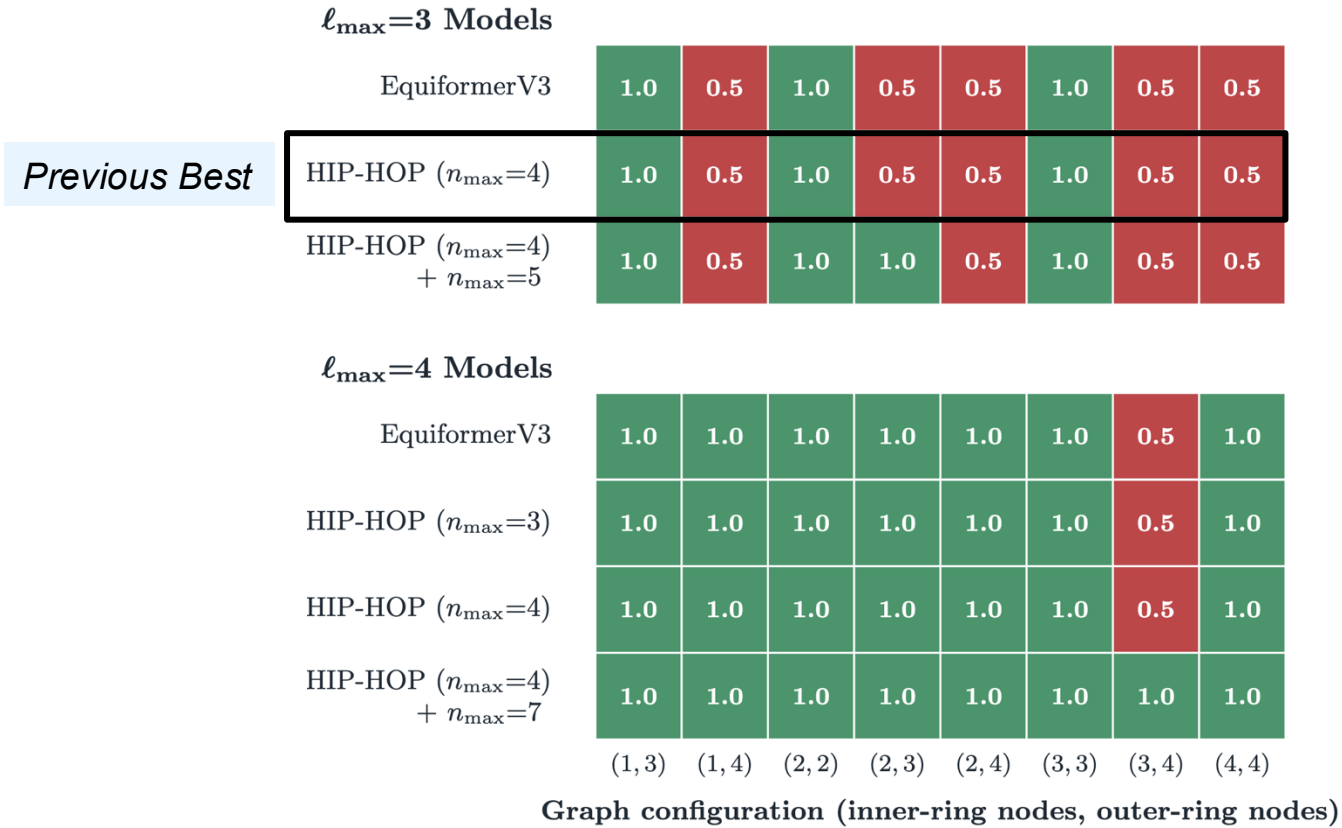
- **Current model internals don't change with rotation**
  - Fixed by adding additional contraction that reacts to rotation

Current HIP-HOP basis (L_{max}=3 and n_{max}=4)

New n_{max}= 5 contraction



New Basis Reaches Perfect Performance



Increasing Efficiency

- **Why aren't complex bases currently used?**
  - Exponentially more terms to compute → very slow to run
- **Optimized performance around large basis representations**
  - Converted native Python to multi threaded PyTorch
  - Parallelized low level Triton kernels

Increasing Efficiency

Invariant Construction Speed

Basis		Total Construction Time		
$\ell_{\max}$	$n_{\max}$	Baseline	Optimized	Speedup
3	4	0.899 s	0.079 s	**11.40×**
3	5	1.791 s	0.084 s	**21.35×**
4	3	0.500 s	0.075 s	**6.69×**
4	4	4.028 s	0.138 s	**29.29×**
4	7	267.811 s	0.862 s	**310.70×**

One time cost during training and inference

Inference Speed

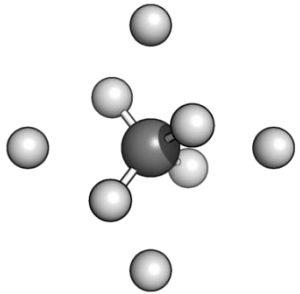
Basis		Time per Atom (μs)		
$\ell_{\max}$	$n_{\max}$	Baseline	Optimized	Speedup
3	4	7.2	6.9	**1.04×**
3	5	10.1	10.5	**0.96×**
4	3	11.9	11.2	**1.06×**
4	4	28.9	16.5	**1.75×**
4	7	239.1	113.2	**2.11×**

How quickly the model can make a prediction

How well do new bases do on real world tasks?

- **Want to accurately predict properties of each molecule**
  1. *Energy*: quantifies the stability of the configuration
  2. *Forces*: direction and strength each atom moves towards lower energy

- **Measure performance on methane dataset**
  - Highly symmetric molecule



Example methane molecule

Methane Results

- **Largest basis significantly improves performance**

- **Complexity of basis does not always improve performance**
  - Surprising result!
  - Found cases where theory ≠ real life

	Basis		Energy		Forces	
	$\ell_{\max}$	$n_{\max}$	MAE	RMSE	MAE	RMSE
	HIP-HOP-NN					
	3	4	0.213 ± 0.016	0.500 ± 0.347	0.607 ± 0.032	2.631 ± 2.557
New bases	3	5	0.225 ± 0.031	0.389 ± 0.046	0.642 ± 0.046	3.842 ± 4.498
New bases	4	3	0.213 ± 0.004	0.378 ± 0.024	0.623 ± 0.013	1.537 ± 0.333
New bases	4	4	$\mathbf{0.184 \pm 0.003}$	$\mathbf{0.322 \pm 0.006}$	$\mathbf{0.551 \pm 0.011}$	$\mathbf{1.244 \pm 0.026}$

Conclusion

1. **Problem:**
   - 3D representations can fail to represent highly symmetrical configurations
2. **Created a benchmark to stress test failures**
   - Found new bases that succeed on benchmark
3. **Optimized model speed and efficiency**
   - Up to 2x faster
4. **Created new model with high accuracy on real world dataset**
   - 35% lower energy RMSE

What I learned

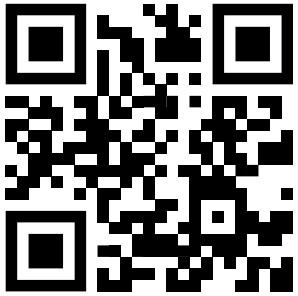
1. **Learned lots of new math**
   - Group theory, spherical harmonics, etc.

2. **Gained experience with low-level optimizations**
   - Had never done Triton kernel speedups before
   - Extremely desirable skill

3. **Gained interdisciplinary knowledge**
   - Only took chemistry 101 in undergrad, learned about computational chemistry and quantum physics

Thank You!

Questions?



Personal Website

HIP-HOP-NN