Classifying Spin-Interactions Using Reinforcement Learning

Kate Meuse (Cornell University)
Collaborators: Jessica Opsahl-Ong (Rice University), Paul Kienzle, William Ratcliff

Xufeng Zhang/Yale University

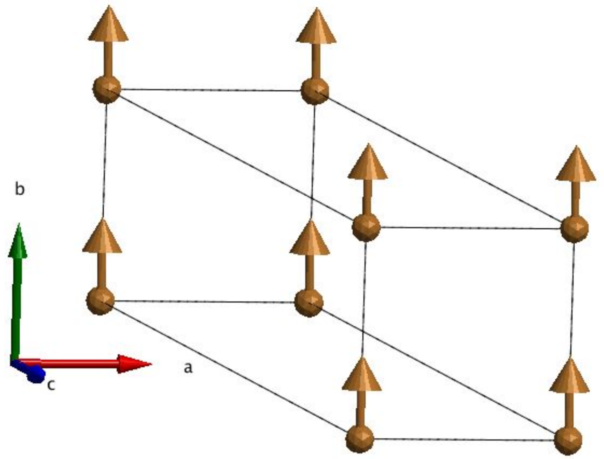
Multitudinous Magnets

- Magnetoresistive RAM (MRAM), hard drives depend on magnets
- Need to examine magnetic structure of new materials to find those with useful properties

Stock Image

Magnet Math

- Ground state orientation of spins in a magnetic structure is derivable
- Don't know which interactions create the ground state

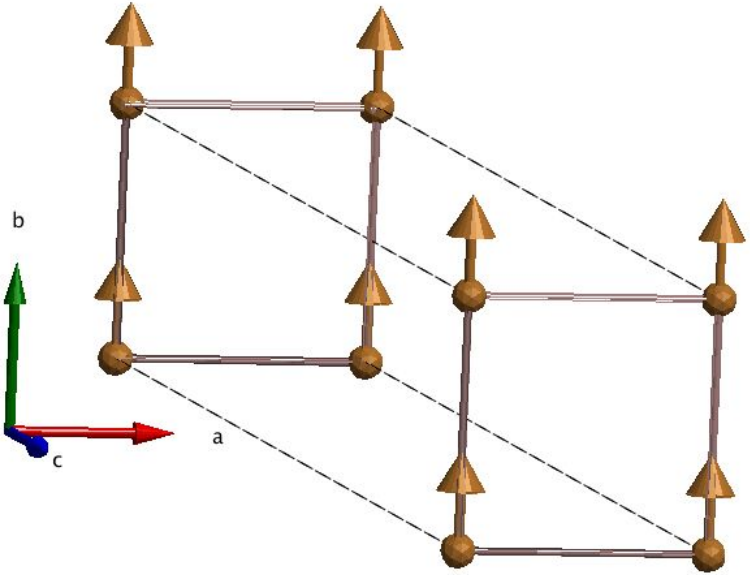


$$\text{Total Energy} = -\frac{1}{2}\sum_{i,j} \mathbf{J}_{i,j}\mathbf{S}_i \cdot \mathbf{S}_j$$

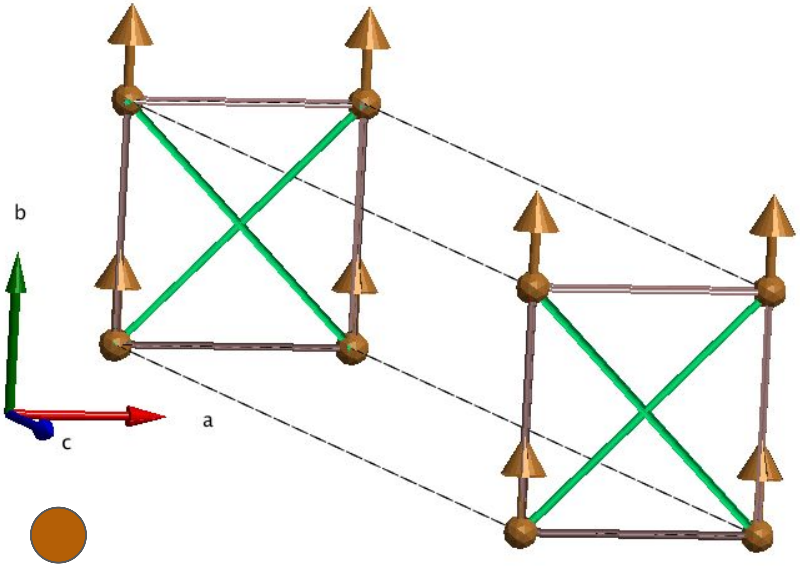
$J_{i,j}$ = strength of interaction
S_i & S_j = spins

Interaction Models

Nearest neighbor



Next nearest neighbor



J

1

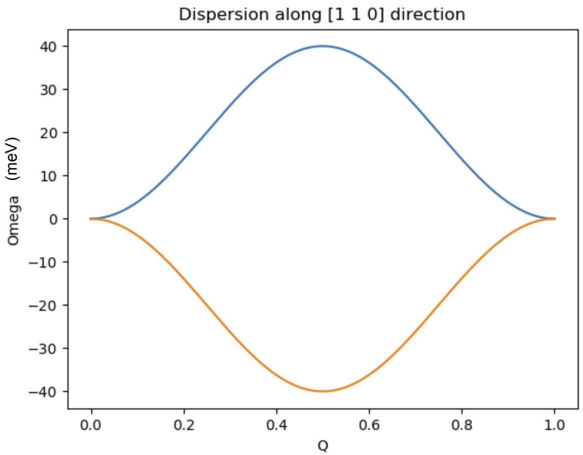
J

Magnet Math

- Inelastic neutron scattering excites the system to create a spinwave that is dependent on the types and strengths of the interactions
- Want to measure the energy of the spinwave along directions of the structure

SpinW/bumps

- MATLAB bound to Python
- Generates dispersion graph given structure, interactions
- Bumps can fit dispersion to get J values (strengths of interactions)
- Compare goodness of fit to pick model



[1 1 0]

➡ $J_1 = -5$

Certain commercial products are mentioned for information purposes, but are not endorsed by NIST

The Problem

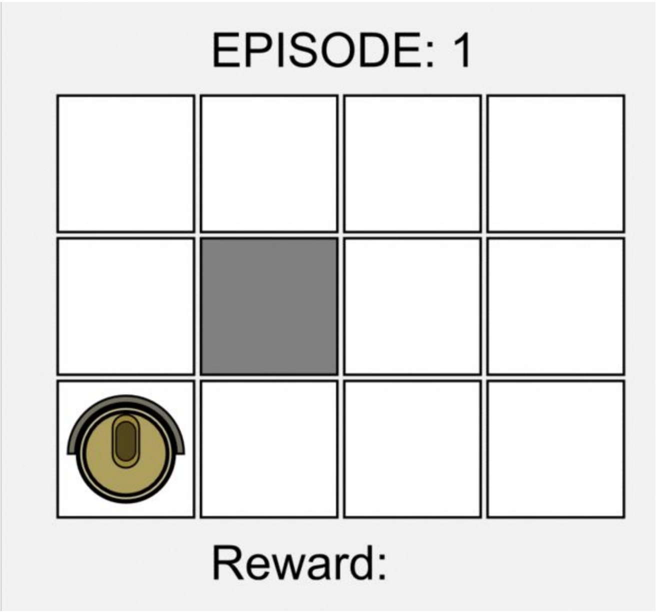
- Beam time is valuable – limited access
- Select more efficient measurements as not all are required
- Determine how to quickly select measurements that describe both type and strength of interactions

The Solution: Reinforcement Learning

AlphaGo Deepmind

Teaching a computer to make optimal decisions using rewards

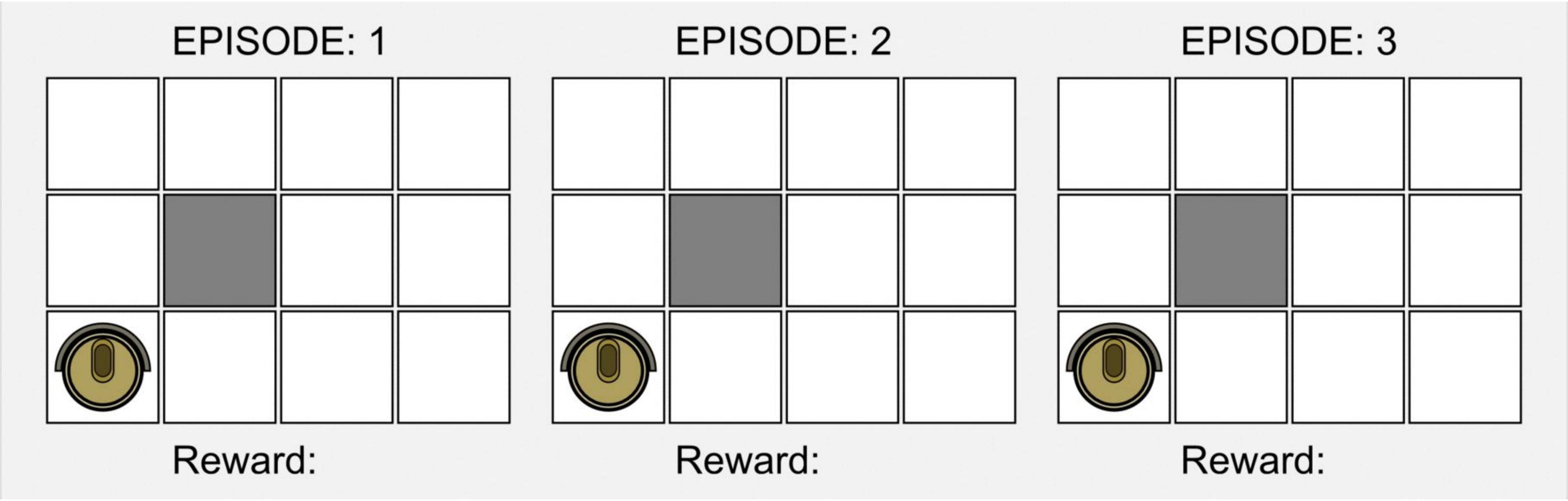
Reinforcement Learning



Massimiliano Patacchiola

- Observes environment to make decisions
- Each step it receives a reward
- Next decision is based on that reward and new environment information

Reinforcement Learning

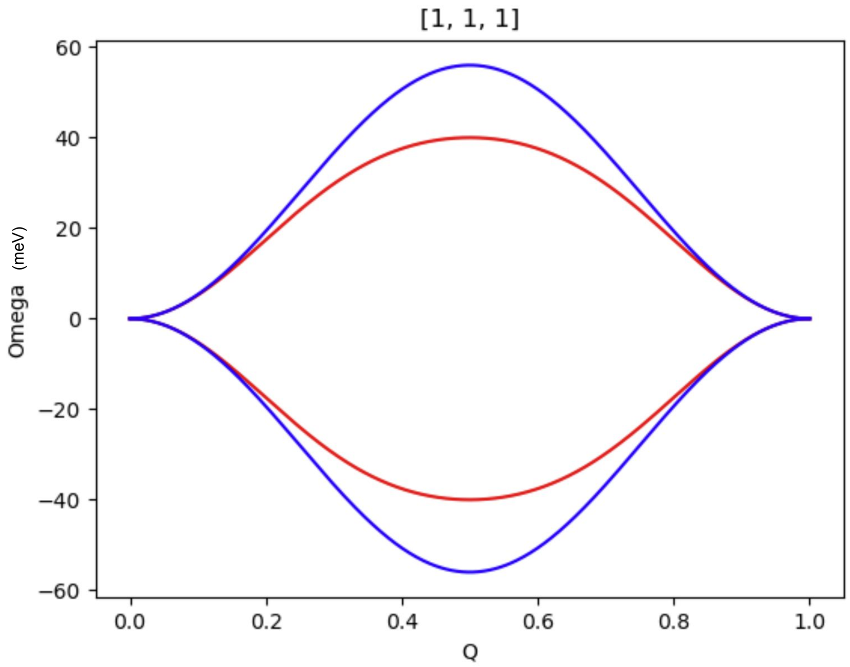


Massimiliano Patacchiola

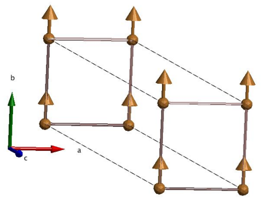
Learns each step and each episode to create the fastest navigation

Applying Reinforcement Learning

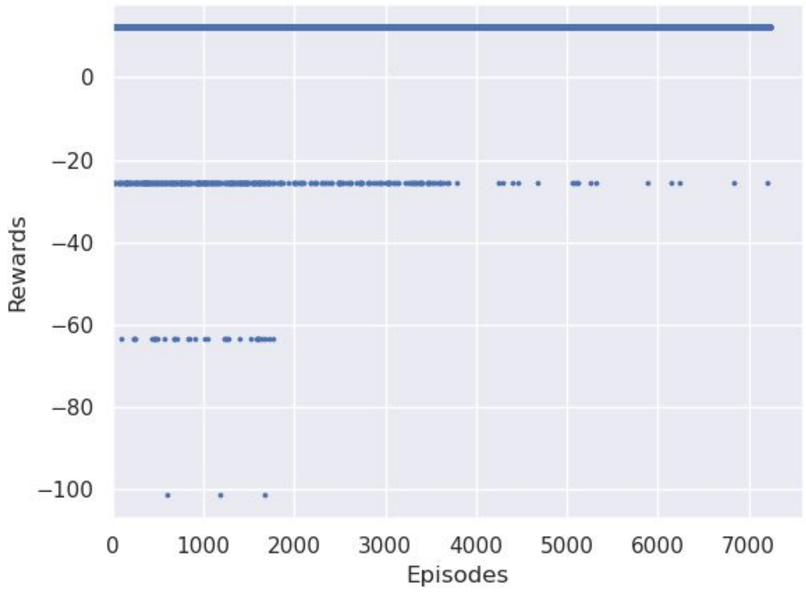
- Action: select a direction to calculate the dispersion
- State: all previously measured directions
- Reward: low chi squared, low uncertainty, and high difference between models
- Ends episode with low chi squared and low uncertainty



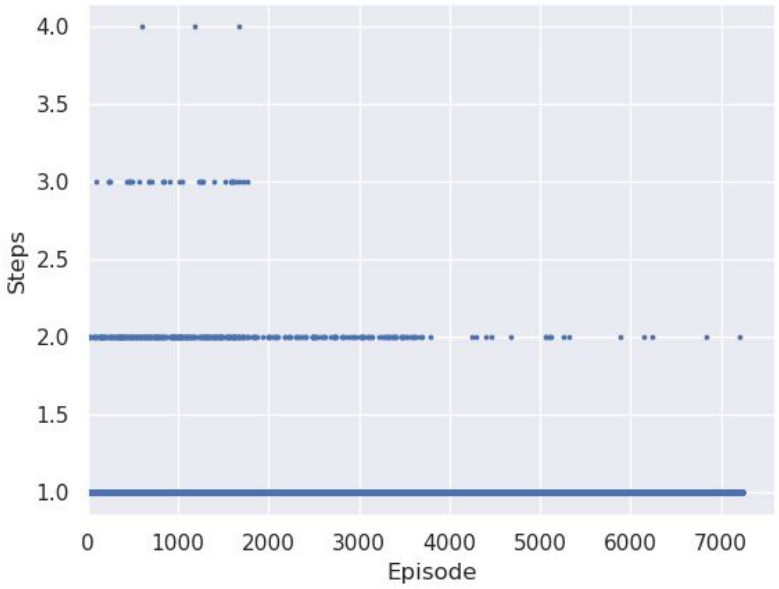
Results: Nearest Neighbor

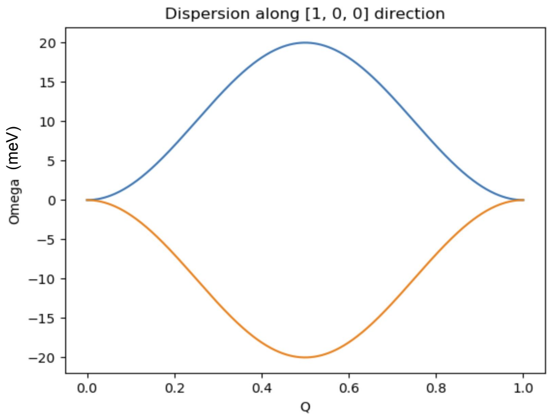


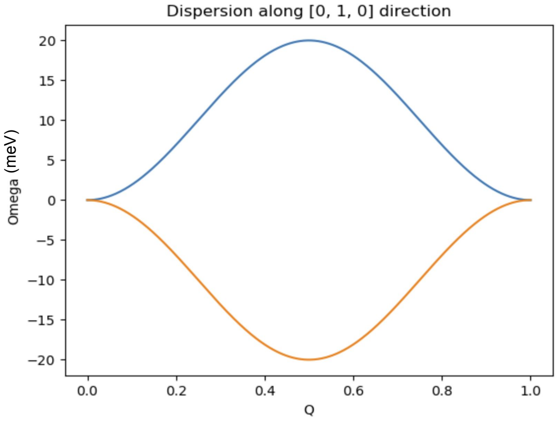
Reward Per Episode

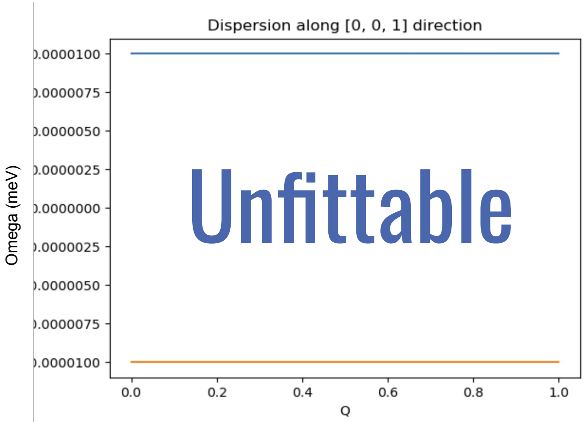


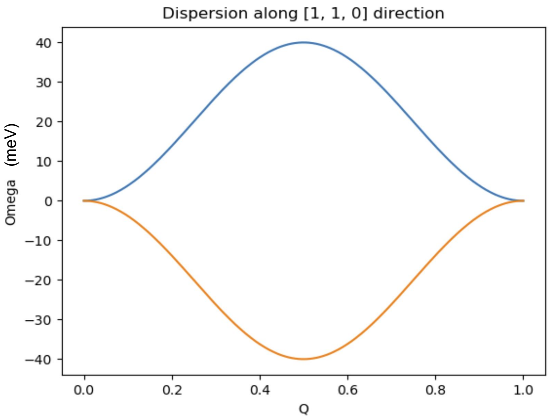
Steps Per Episode

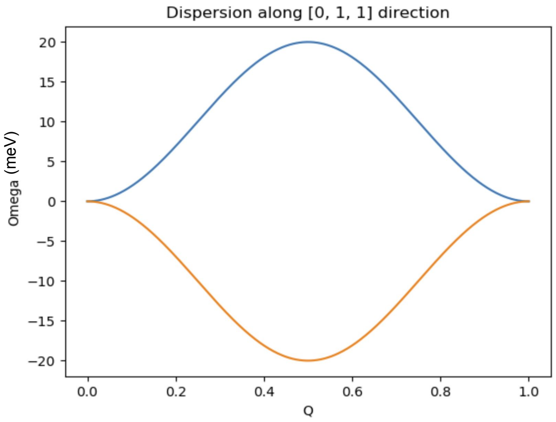


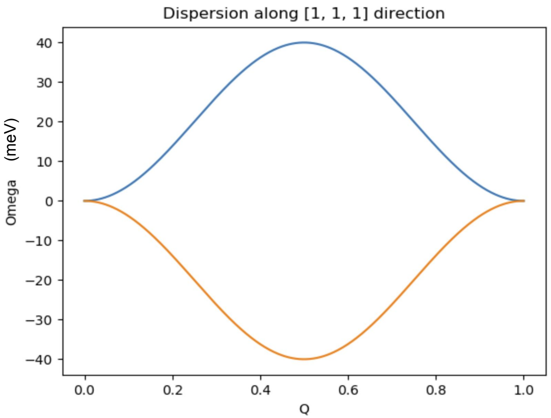

Dispersion along [1, 0, 0] direction
Omega (meV)
Q


Dispersion along [0, 1, 0] direction
Omega (meV)
Q


Dispersion along [0, 0, 1] direction
Omega (meV)
Q
Unfittable

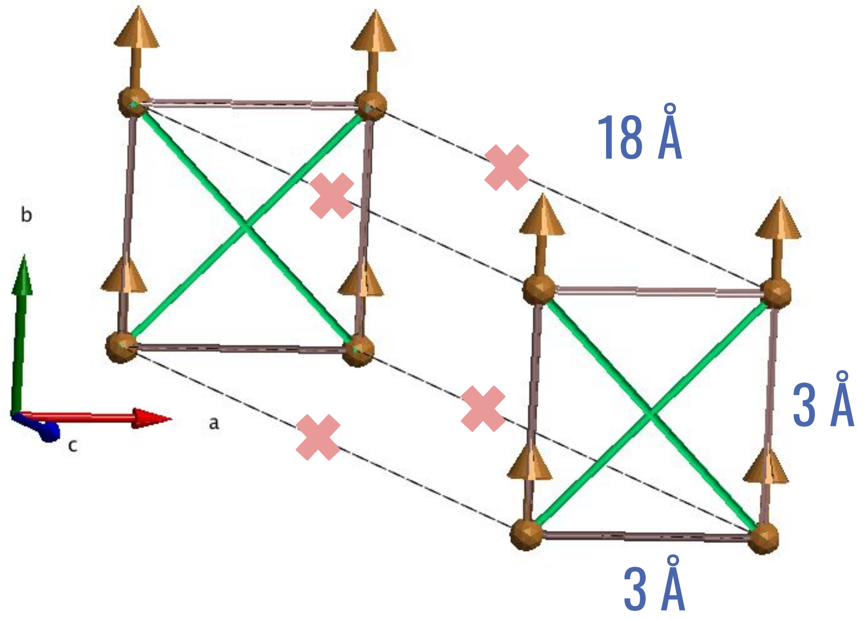

Dispersion along [1, 1, 0] direction
Omega (meV)
Q


Dispersion along [0, 1, 1] direction
Omega (meV)
Q

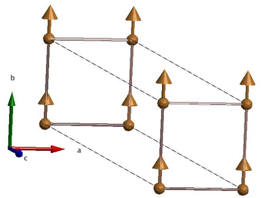

Dispersion along [1, 1, 1] direction
Omega (meV)
Q

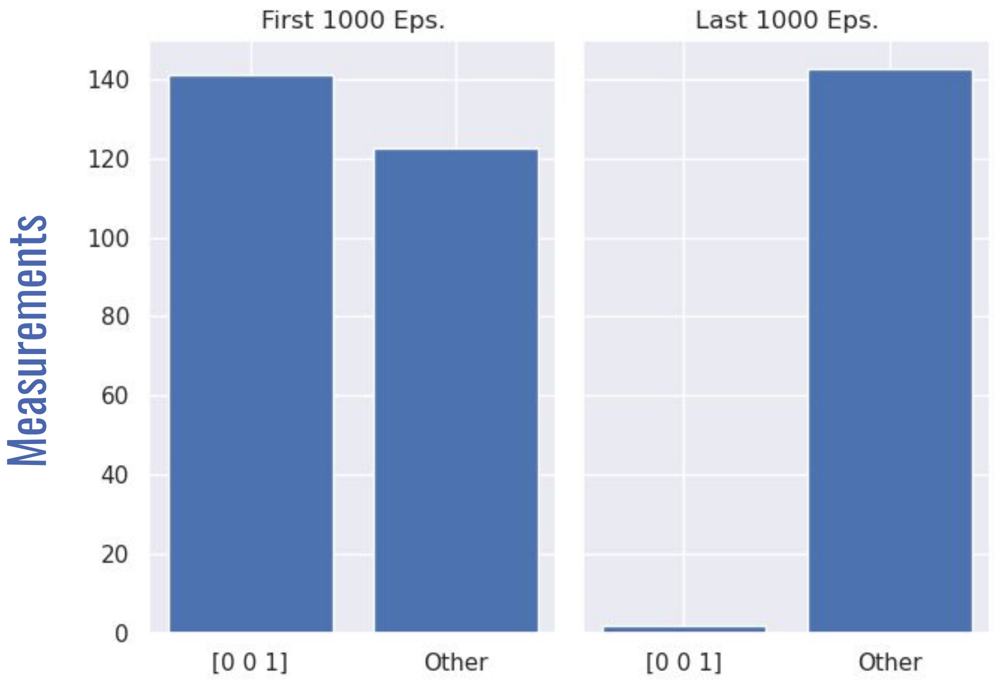
It's Essentially A Square

- Out of the plane of the square, not much is happening
- Valuable information only found in directions with some a or b component



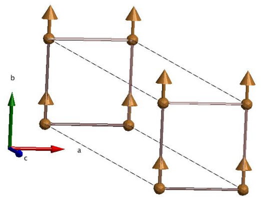
Correctness: Nearest Neighbor





- Slightly higher at first, probably not significant
- Likelihood of picking bad measurement compared to any other option much decreased

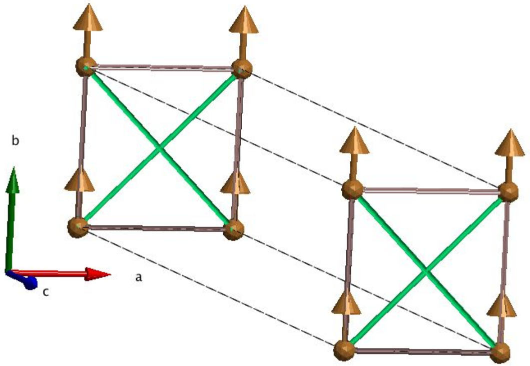
Correctness: Nearest Neighbor



- Always chose the correct model
- Due to the process of model selection, this is unsurprising
- Model with fewer parameters and same fit will be chosen

~100%

Next Step: Next-Nearest Neighbor



Future Steps

- Calculate and fit neutron intensities instead of merely dispersion
- Add finite resolution of the instrument
- Attempt same problem with Gaussian processes
- Publish!

Acknowledgements

Thanks to the SURF program directors, interns and mentors.

Special thanks to William Ratcliff, mentor.

We are grateful for funding from the Center for High Resolution Neutron Scattering.

Questions