LABORATORY FOR LASER ENERGETICS | University of Rochester

High Pressure Material Phase Classification Using Machine Learning

Cameron Ryan, Shuai Zhang

AI/ML Group Presentation February 20, 2026

Phase Identification

- Task of identifying a material structure, using the coordinates of component atoms
- Coordinates come from DFT or CMD simulations, and this is used as a method for predicting what phase transitions will occur from simulation
- Most methods focus specifically on identifying the phase of a specific local environment of atoms, which is made up from k nearest neighbors to a center atom

Common Existing Methods for Phase Identification

- Many methods already exist for this, which do not necessarily involve machine learning
- Polyhedral Template Matching (Larsen et al. 2016) attempts to compare local environments to the ideal templates for structures
- Voronoi Cell Analysis (Lazar et al. 2015) the Voronoi cell topology for the central atom in an environment, and compares these to known topologies for ideal structures
- The drawback to these methods is that they are highly specialized to a few simple phases (FCC, BCC, HCP, etc). They cannot be used for complex crystal structures, or nonperiodic structures with no specific geometry (Amorphous, Melt)

Machine Learning Phase Identification

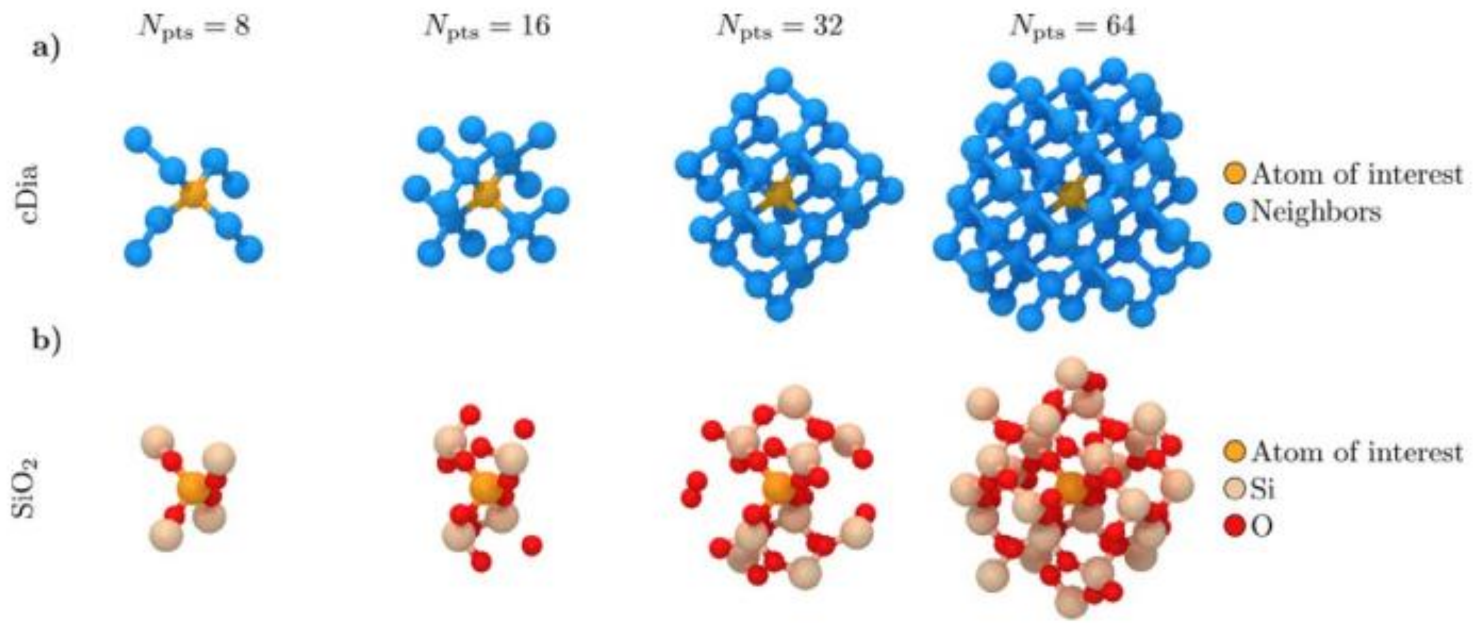
- There are various methods for computing phase predictions with machine learning
- Rather than needing to create a method that is adapted to specific phases, we can extract some sort of descriptors from the local environment, and use machine learning to learn to identify phases
- This is much more efficient for performing the task with many highly complex, or non periodic phases

Erhard’s method

- Erhard et al. propose a method for computing phase predictions from atomic coordinates
- Take ideal structure pure phase samples
- run simulations of the pure phase samples being heated
- extract ‘local environments’ centered around specific atoms, which contain the relative coordinates of each of the closest atoms
- Generate a dataset containing the local environments
- Use this dataset to train a model to try to predict the correct structure from the atomic positions

Local environment dataset

- Using Erhard’s method, we have a dataset of local environments for pure phase samples
- We generated datasets for 16, 32, 64, and 128 atom environments



Adapted from Linus C Erhard *et al Modelling Simul. Mater. Sci. Eng.* 32 (2024)

Dynamic Graph Convolutional Neural Networks

- Erhard proposed using this architecture for phase identification
- Introduced in “Dynamic Graph CNN for Learning on Point Clouds” (Wang et al.)
- Provides a method for performing an operation similar to image convolution, but for point graphs

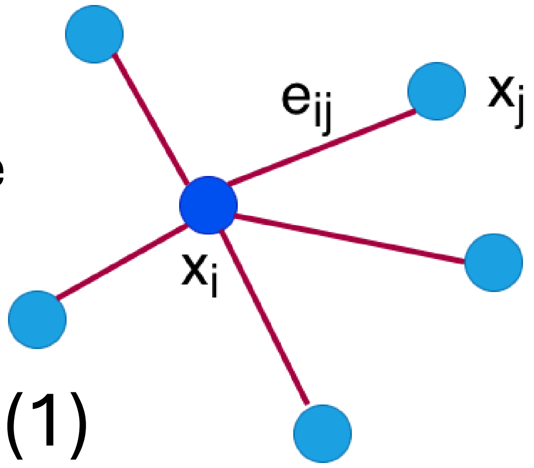
Dynamic Graph Convolutional Neural Networks: Edge Convolution

- Given a set of points
- Construct a k-nearest neighbor graph on the set of points
- For each edge, feed the relative position between the two points into a single layer neural network. The output of this neural network is in an arbitrary dimensional latent space, and called the ‘edge feature’
- For each point, and for each dimension, take the maximum edge feature in that dimension

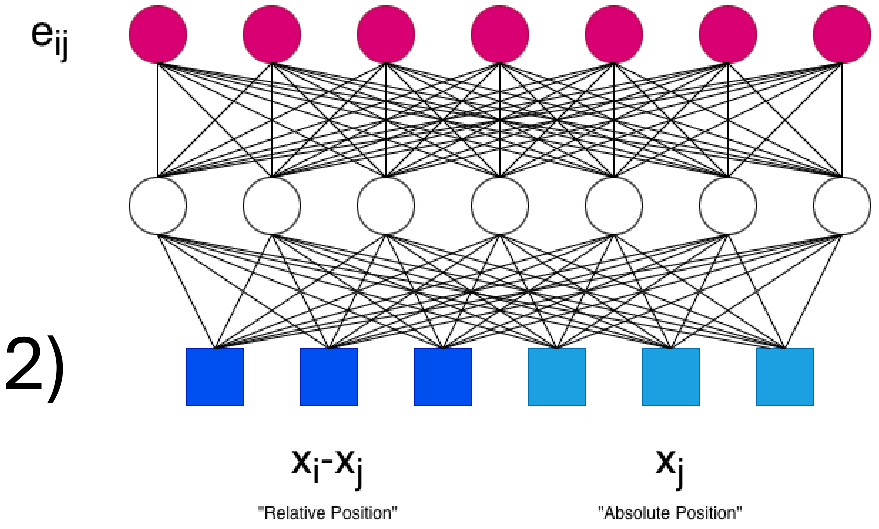
Dynamic Graph Convolutional Neural Networks: Edge Convolution Layer

- Given set of n points
- (1) Construct a graph on the point set
- (2) For each edge in the graph, create edge feature e_{ij}
- (3) For each point, aggregate all the edge features connected to that point
- Note that the output dimension will not always be the same as the input dimension

(1)



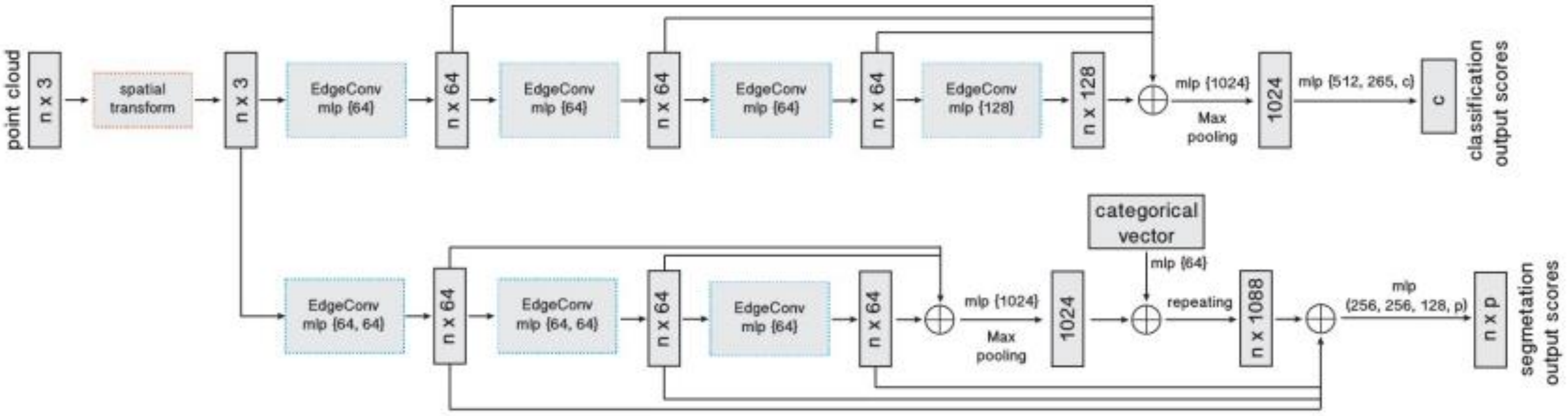
(2)

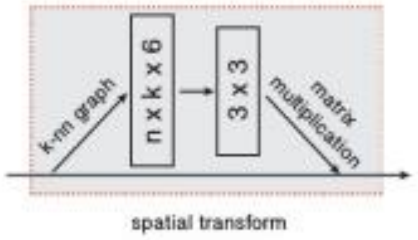


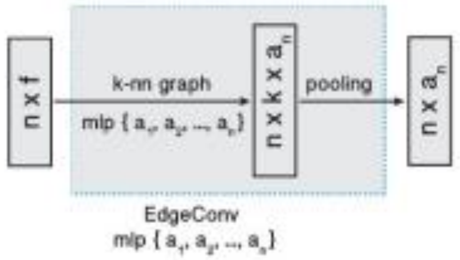
(3)

$$x'_{im} = \max_j e_{ij_m}$$

Dynamic Graph Convolutional Neural Networks: Model Architecture



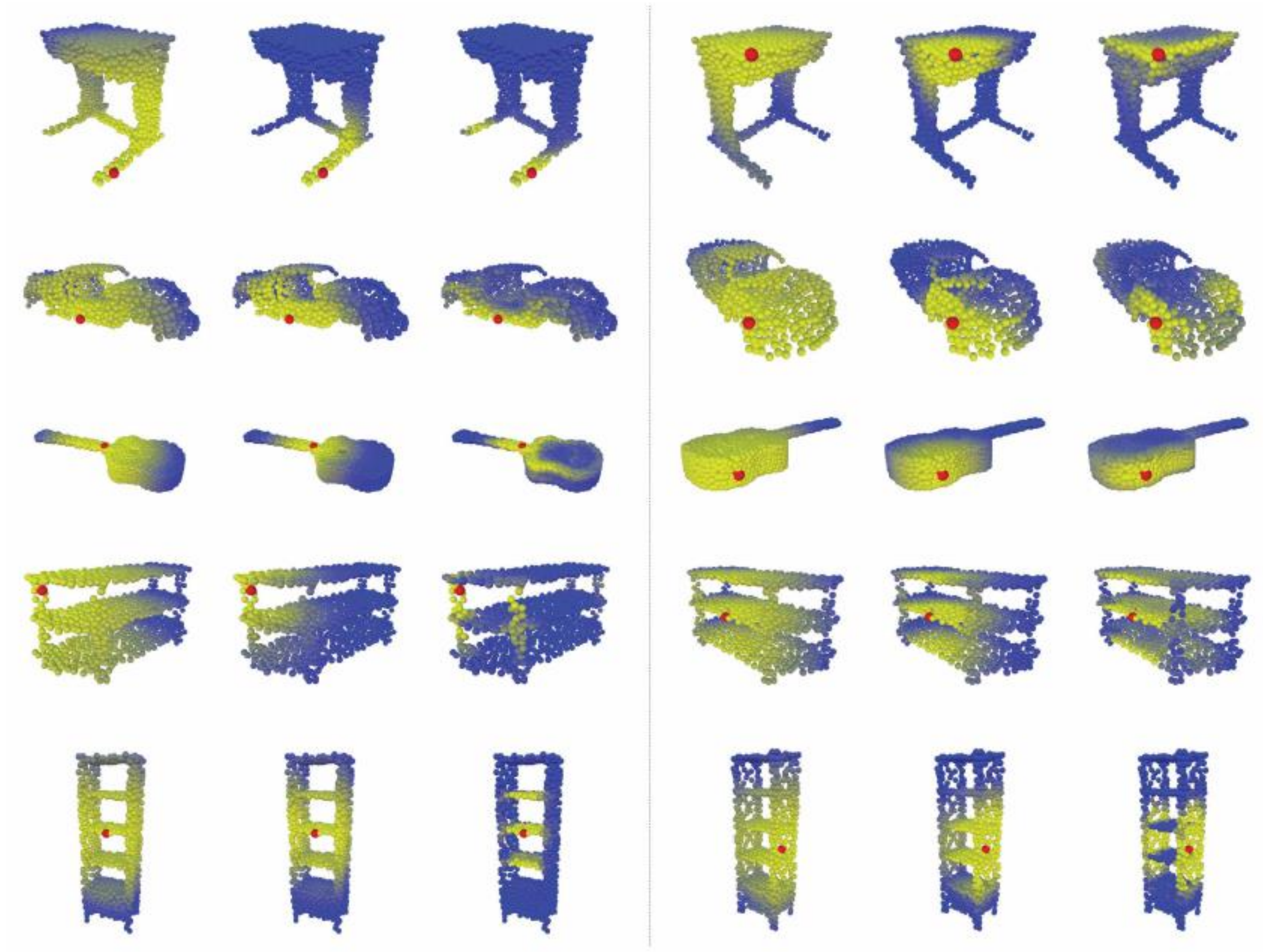




Adapted from Wang et al, ACM Transactions on Graphics Vol 38 (2019)

Dynamic Graph Convolutional Neural Networks

Further right represents activations in select dimensions at increasingly deeper layers



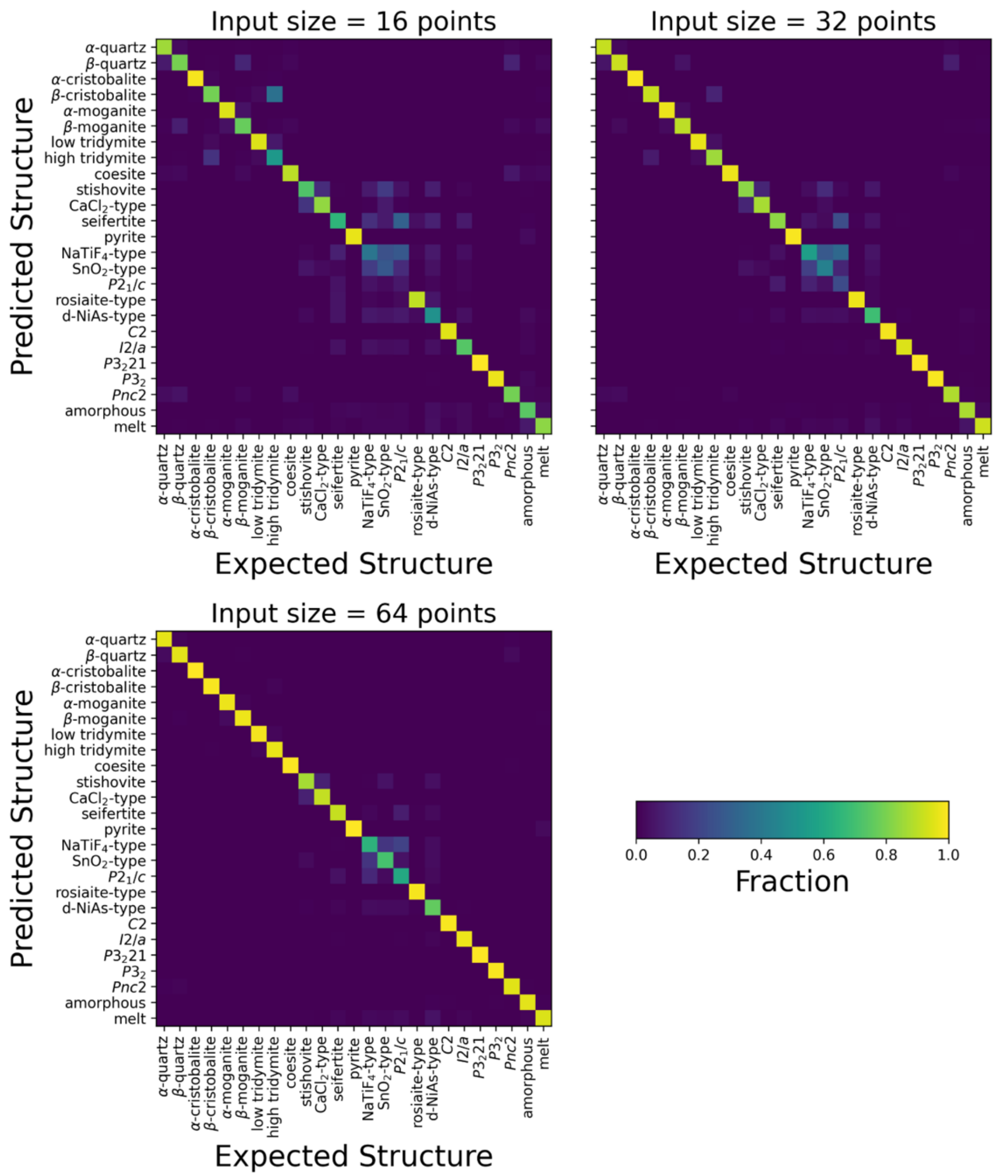
Adapted from Wang et al, ACM Transactions on Graphics Vol 38 (2019)

Erhard’s Results

- Erhard trained three models on 25 SiO2 phases

N points	Erhard Reported Test Accuracy
16	75%
32	85%
64	92%

Erhard’s Results for SiO_2 phases

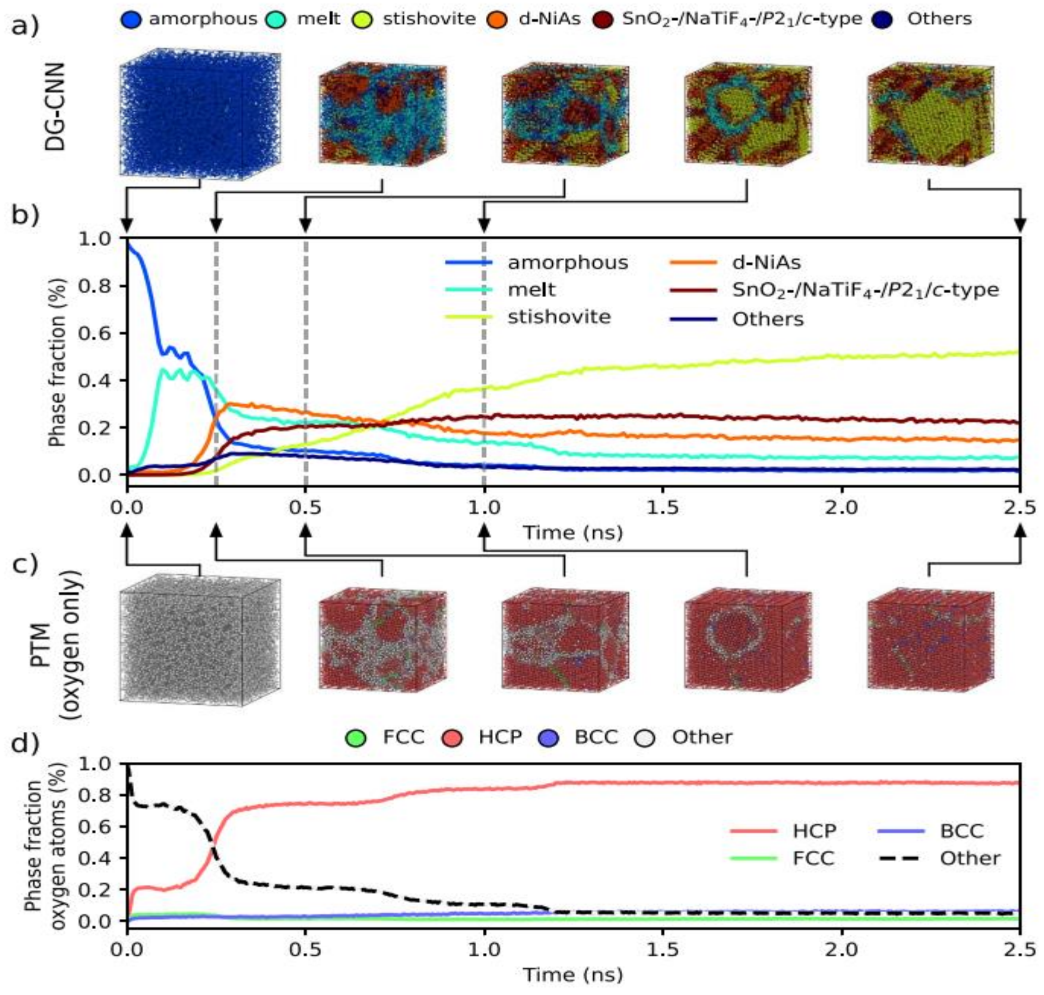


Adapted from Linus C Erhard *et al Modelling Simul. Mater. Sci. Eng*. 32 (2024)

Erhard’s predictions: shock loading simulations of amorphous structures

- They start with an amorphous SiO2 sample, and subject it to shock. Experimentally, this is known to cause the silica to crystallize into stishovite
- The DGCNN was used to classify the structure of each atom at each snapshot during the shock simulation
- This was used to calculate the fraction of each structure type in the sample at each time
- Erhard’s model predicts a gradual transition to stishovite being the dominant phase

Erhard’s Predictions: Shock loading simulations of amorphous structures



Adapted from Linus C Erhard *et al Modelling Simul. Mater. Sci. Eng.* (2024)

LABORATORY FOR LASER ENERGETICS | MELIORA University of Rochester

My Research Goals

- Replicate Erhard’s results using their code
- Look into ways that I can improve the architecture
- Create a model that will work for more phases, specifically high- pressure phases which are relevant to research at LLE

Replicating Erhard’s Code

- Cloned their Github repo
- Created datasets for 16, 32, 64, and 128 atom environments using their simulations
- Using their code, trained models:

N points	Erhard	Recreation
16	75%	
32	85%	81%
64	92%	89%

Rotation Invariance

- Attempted to train model to be rotation invariant by combining standard deviation of rotated predictions with crossentropy loss
- Interestingly, this method causes the crossentropy loss to decrease slightly faster at the beginning of training. Despite this, it was not pursued, because this method would often double the memory footprint of model training, and the model eventually learns rotation invariance with or without this

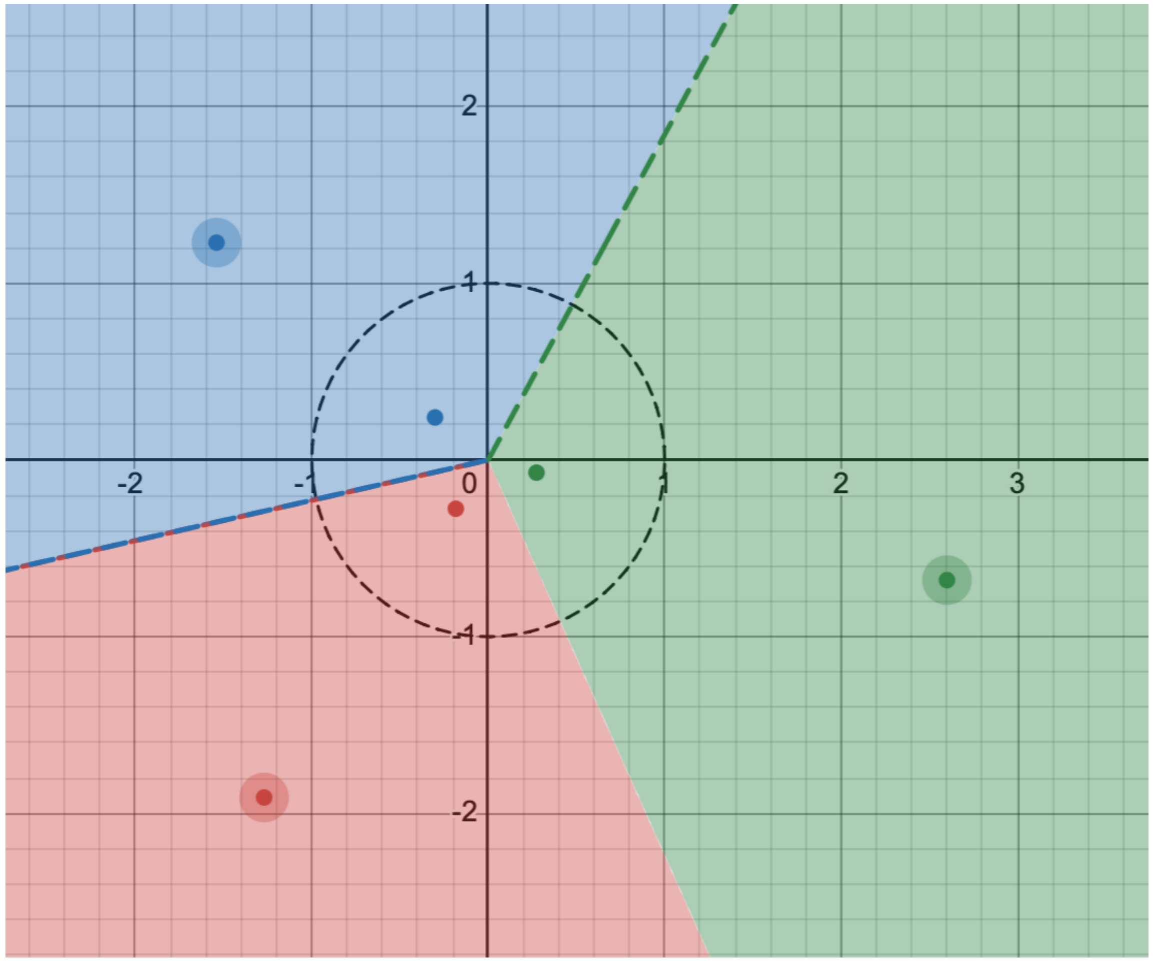
$$\mathcal{L}_{\text{combined}}(\theta) = -\phi \sum_{i=0}^{N} \log \mathcal{F}_\theta(x_i) + (1-\phi)\alpha \sum_{i=0}^{N} \sqrt{\frac{1}{C} \sum_{j=0}^{C} (\mathcal{F}_\theta(x_{ij}) - \mu_i)^2}$$

$$N = \text{Number of batches}$$
$$C = \text{Possible phases}$$
$$\mathcal{F}_\theta(x_i) = \text{Model predictions}$$
$$\phi \in [0,1]$$
$$\mu_i = \text{Mean predictions}$$

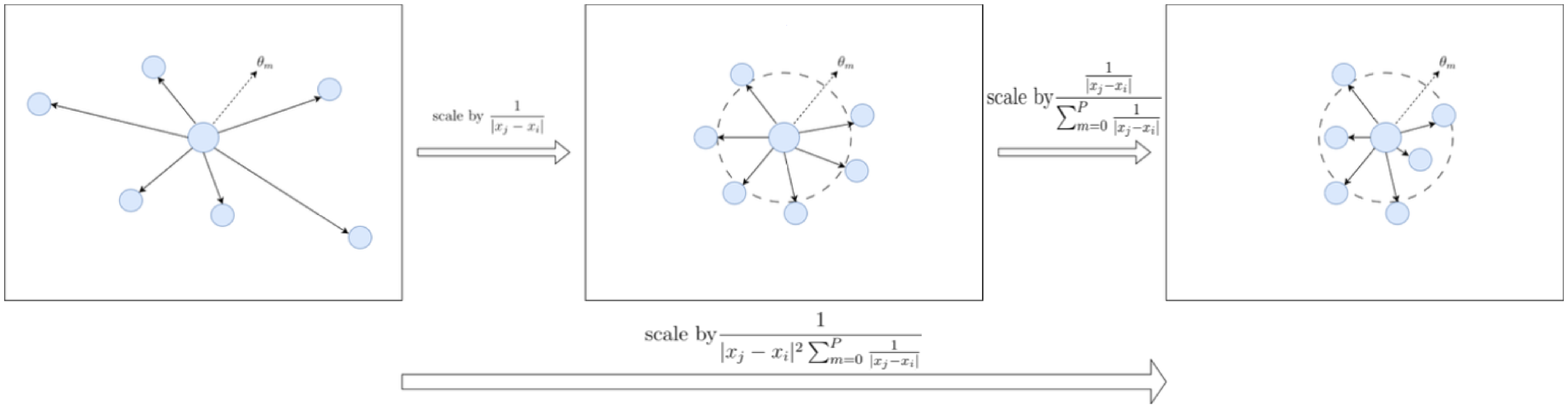
LABORATORY FOR LASER ENERGETICS | University of Rochester

Spatial Weighting

- Because of the nature of dot product, atoms that are farther away will have a greater impact on the feature generated by the first edge convolution layer
- This is not good, because in phase identification, usually the closest atoms are the most important

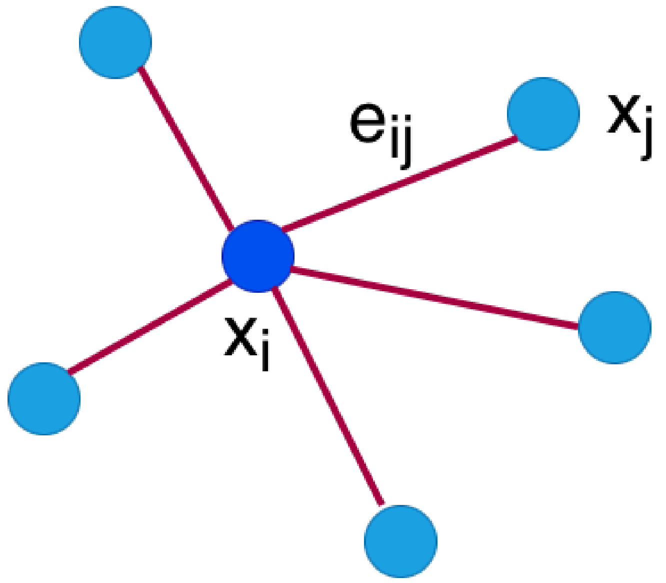


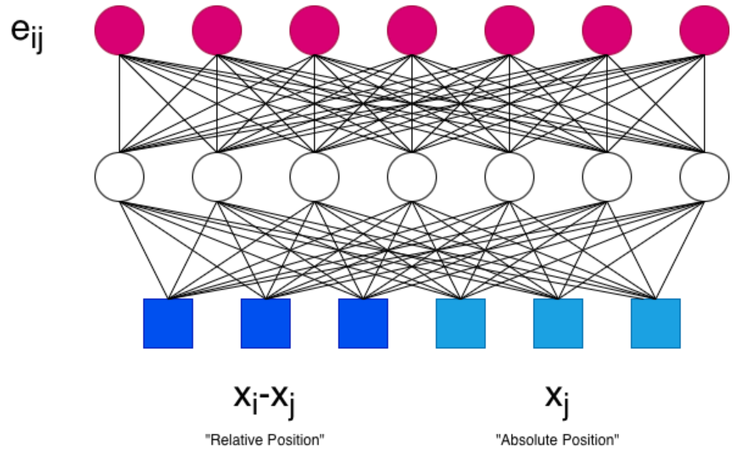
Spatial Weighting


θ_m
θ_m
θ_m
scale by $\frac{1}{|x_j - x_i|}$
scale by $\frac{\frac{1}{|x_j - x_i|}}{\sum_{m=0}^{P} \frac{1}{|x_j - x_i|}}$
scale by $\frac{1}{|x_j - x_i|^2 \sum_{m=0}^{P} \frac{1}{|x_j - x_i|}}$

Nearest Neighbors Graph

- In Erhard's original implementation, a new nearest neighbor graph is generated at every edge convolution layer based on distances in the latent space
- The latent space represents the 'qualities' of points, rather than their actual location
- Our experiments showed that using the nearest neighbor graph from the input space improves model performance





LABORATORY FOR LASER ENERGETICS | MELIORA University of Rochester

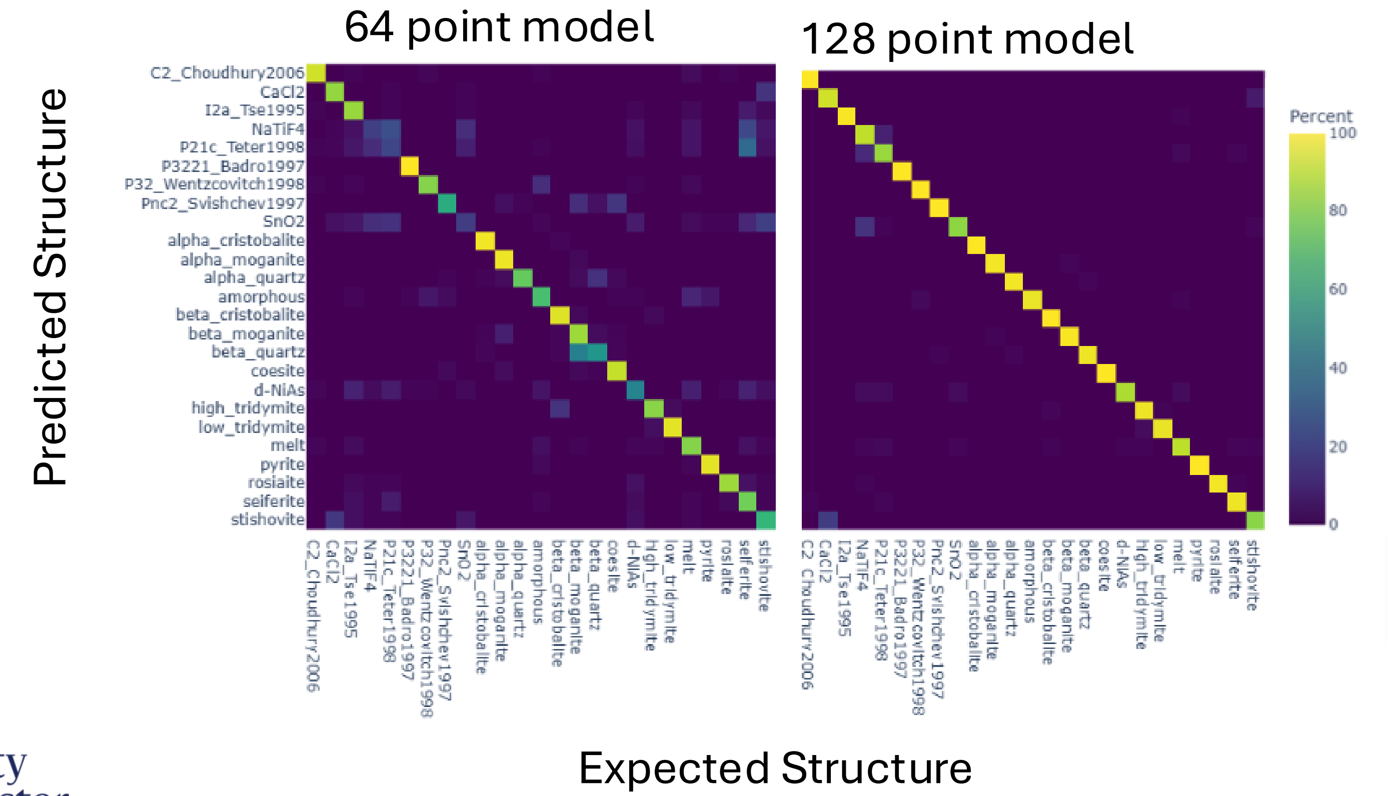
DGCNN Experiments Overview

Rotation Optimizatio n	Nearest Neighbors Graph	Spatial Weighting Mechanism	Included Relative Positions (Input space)	Included Absolute Positions (Input Space)	Included Relative Positions (Embeddin g Space)	Included Absolute Positions (Embeddin g Space)	Cross entropy after 100 batches	
No	Embedding Space	No	Yes	Yes	Yes	Yes	1.93	Erhard
No	Embedding Space	No	Yes	No	Yes	No	2.33	
No	Embedding Space	No	No	Yes	No	Yes	2.13	
No	Embedding Space	Yes	Yes	No	Yes	Yes	1.86	
Yes	Embedding Space	Yes	Yes	No	Yes	Yes	1.69	
Yes	Input Space	Yes	Yes	No	Yes	Yes	1.63	Modified DGCNN

LABORATORY FOR LASER ENERGETICS | University of Rochester

128 point model (Modified DGCNN)

- Trained a model on 128 point environments, using these new architecture modifications
- This model obtained an accuracy of 95%, which is the highest of any model



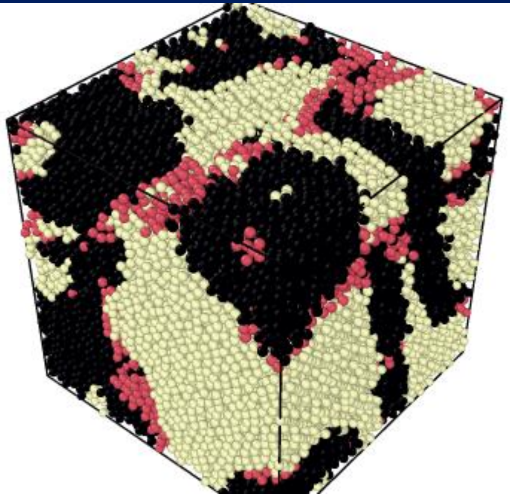
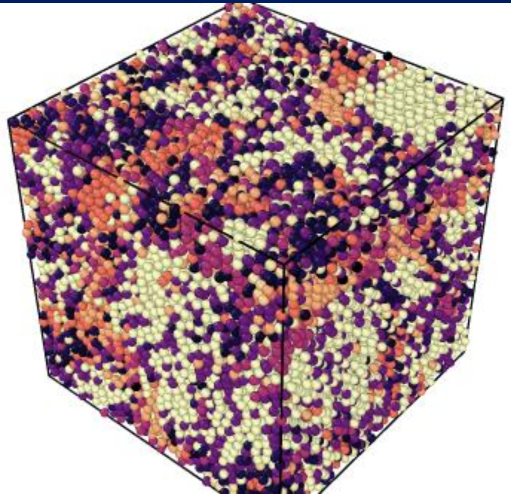
Modified DGCNN Model Comparisons

Model	Test Accuracy After 10 Epochs
DGCNN (Erhard), 32 point environments	0.81
Modified DGCNN, 32 point environments	0.85
DGCNN (Erhard), 64 point environments	0.89
Modified DGCNN, 64 point environments	0.93
Modified DGCNN, 128 point environments	0.95

Full Sample Inference Idea

- Attempted to train a model on the largest possible environments
- Theoretically, such a model would 'think' that it is in an infinitely sized environment, and could be used on an entire sample, generating predictions for each atom in one forward pass
- This method would create a great speed boost for calculating phase predictions, because it would cut out the need to extract local environments for every atom, and generate predictions for every environment
- Unfortunately, after much experimentation, this method was unable to generate reliable results

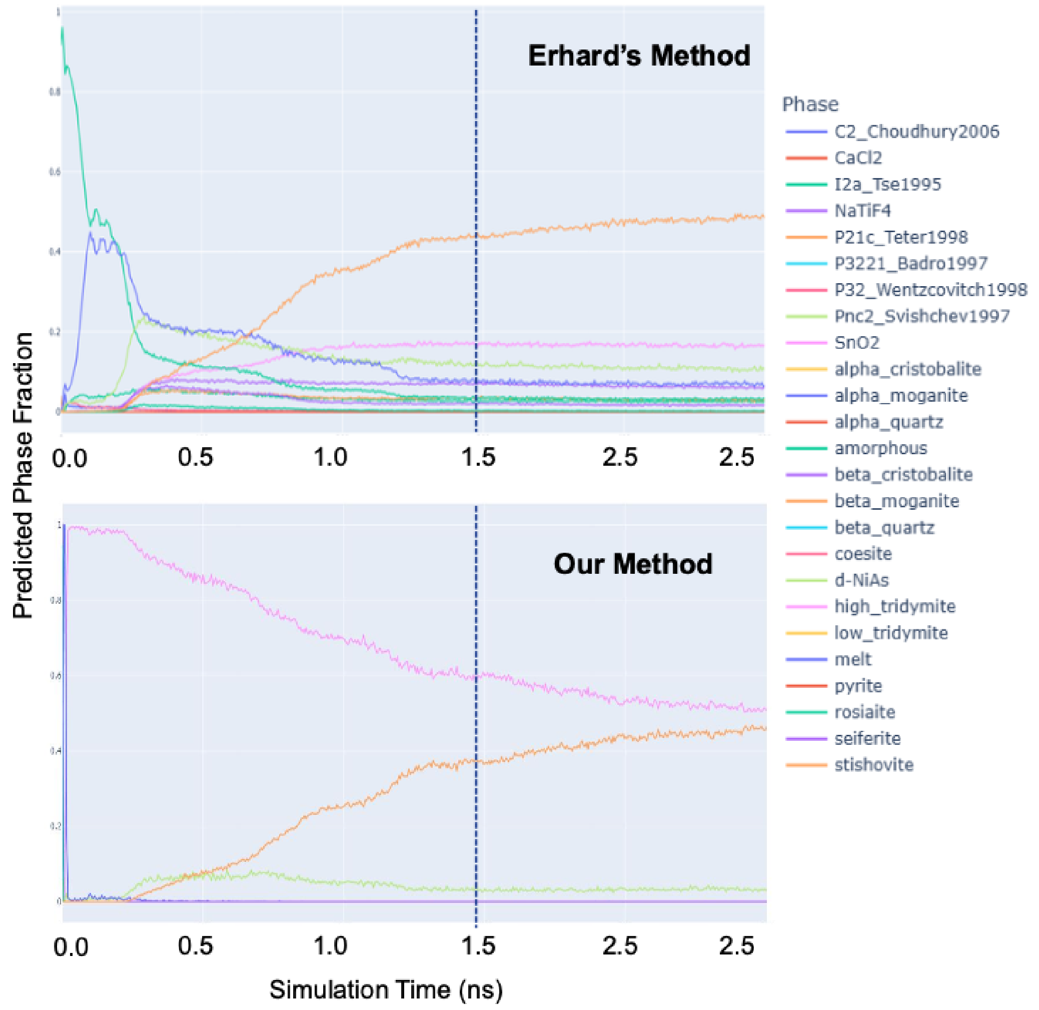
Full Sample Inference Comparison on Amorphous SiO2 shock simulation

Full Sample Inference	Environment Inference (Erhard Model)
	

🟨	Stishovite
⬛	SnO2
🟥	d-NiAs

LABORATORY FOR LASER ENERGETICS | University of Rochester

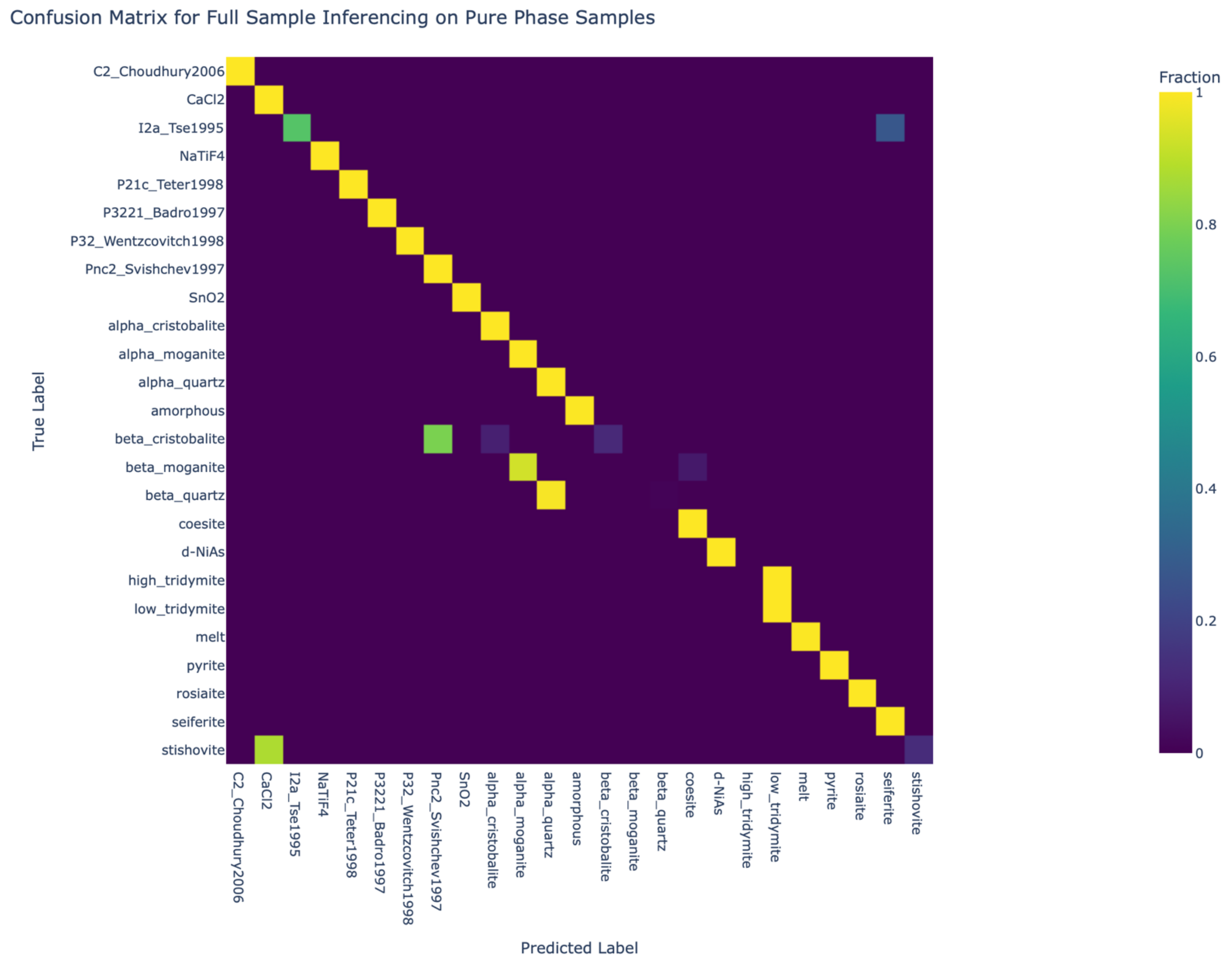
Full Sample Inference Comparison on Amorphous SiO2 shock simulation



LABORATORY FOR LASER ENERGETICS | University of Rochester

Full Sample Inference Confusion Matrix

- Confusion Matrix generated by calculating model predictions on entire sample at once
- Creates very clean, consistent predictions for most samples, however small mistakes lead to confident wrong predictions
- This makes the model less useful for scientific applications



High Pressure Phases

- Used transfer learning to train models on the new phases
  - Fe2P
  - P21m
  - R-3
  - Cotunnite
- Data was generated from smaller, DFT simulations
- Models were able to learn to identify these structures with a high level of accuracy

Experimental Architectures

- Experimented with creating newer, more simple architectures
- Hypersphere model
- Gaussian Kernel Model

Gaussian Kernel Model

- For each atom in a local environment, create an isotropic Gaussian centered at that coordinate
- Create an arbitrary number of ‘atom detectors’, which will calculate the sum of gaussians at a specific coordinate
- Use the sum of gaussians value for each ’atom detector’ as the input to a neural network

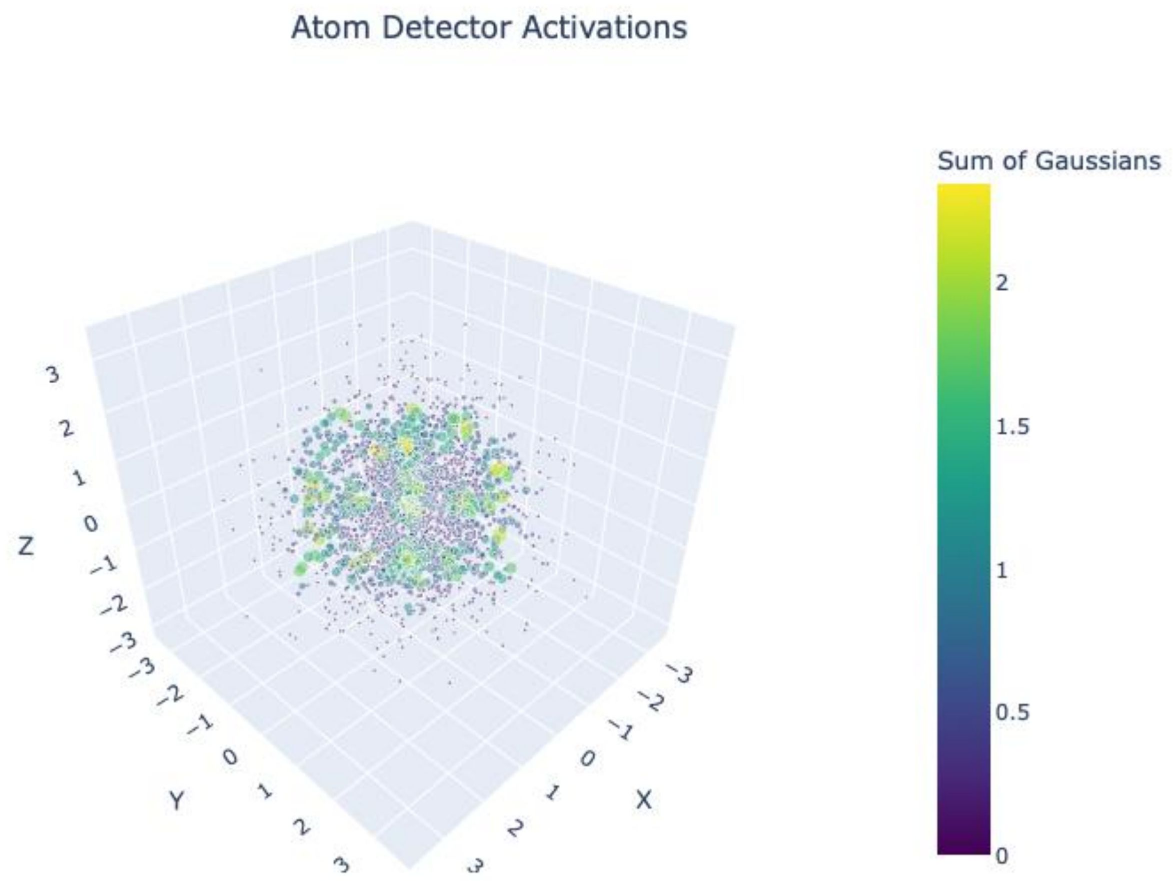
$$g_x(z) = \sum_{i=1}^{n} e^{-\frac{|x_i - z|^2}{2\sigma^2}}$$

$x = \{x_1, ..., x_n\}$ (set of atom coordinates)

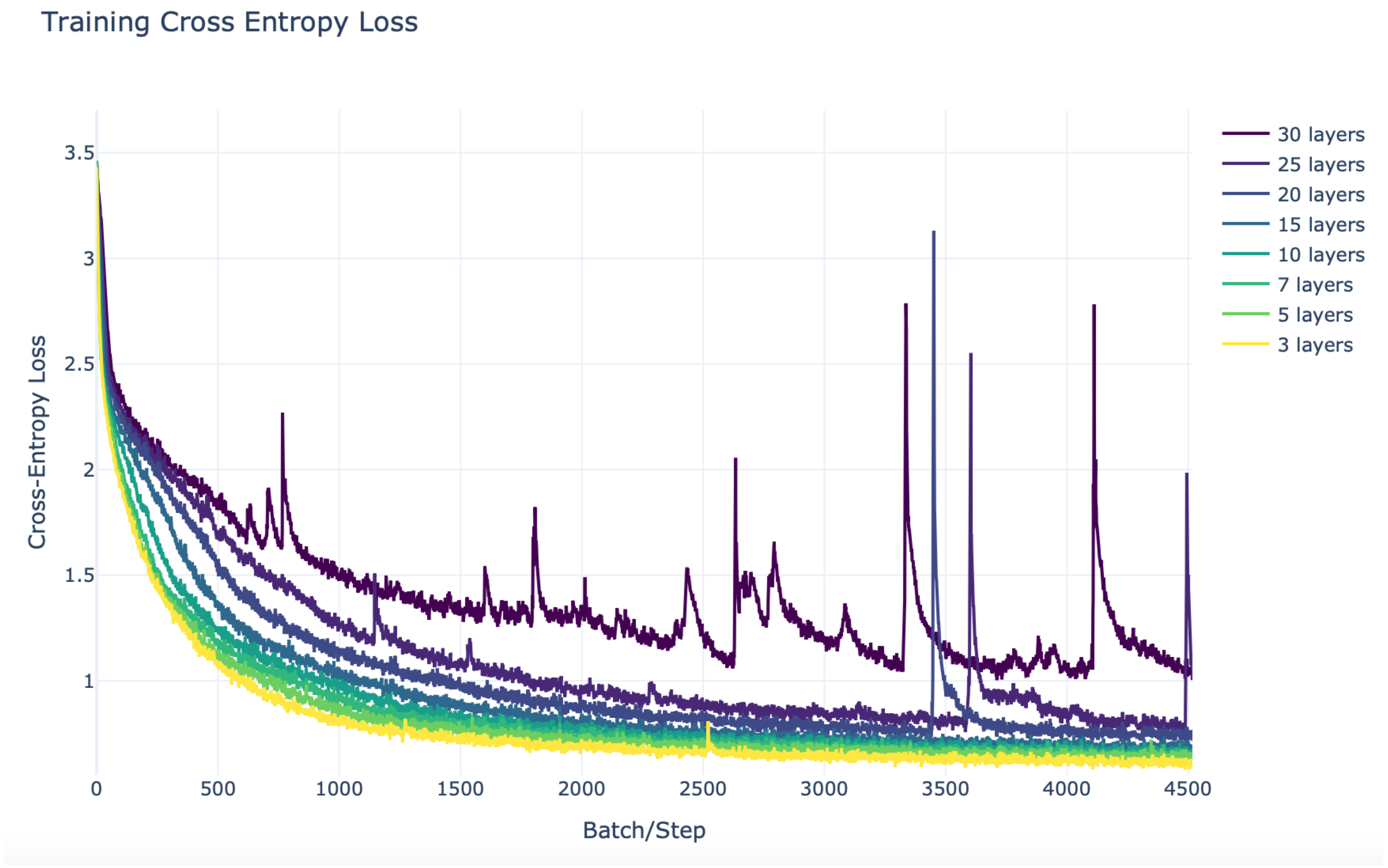
z (coordinate of atom detector)

σ (parameter learned during training)

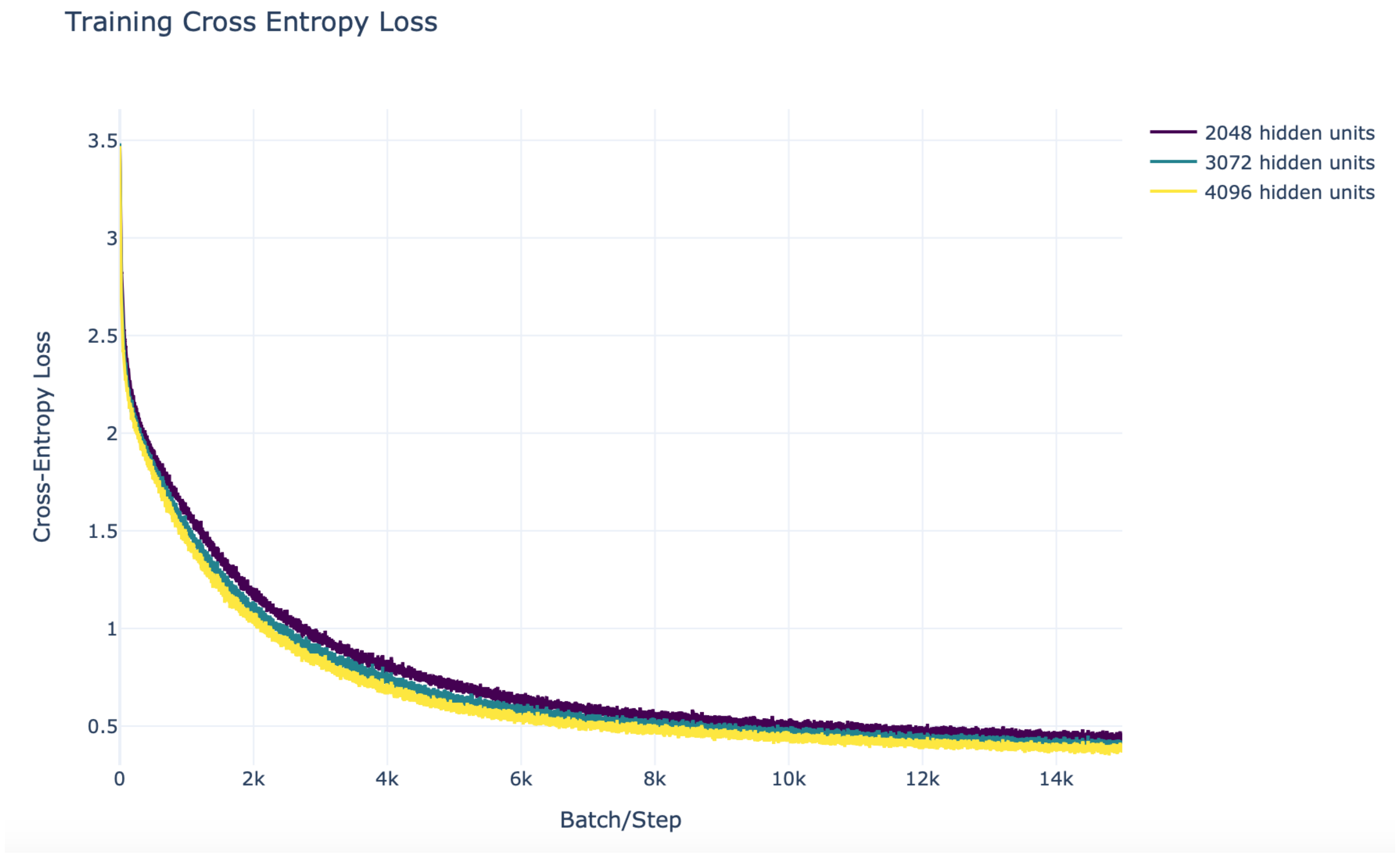
Gaussian Kernel Atom Detectors



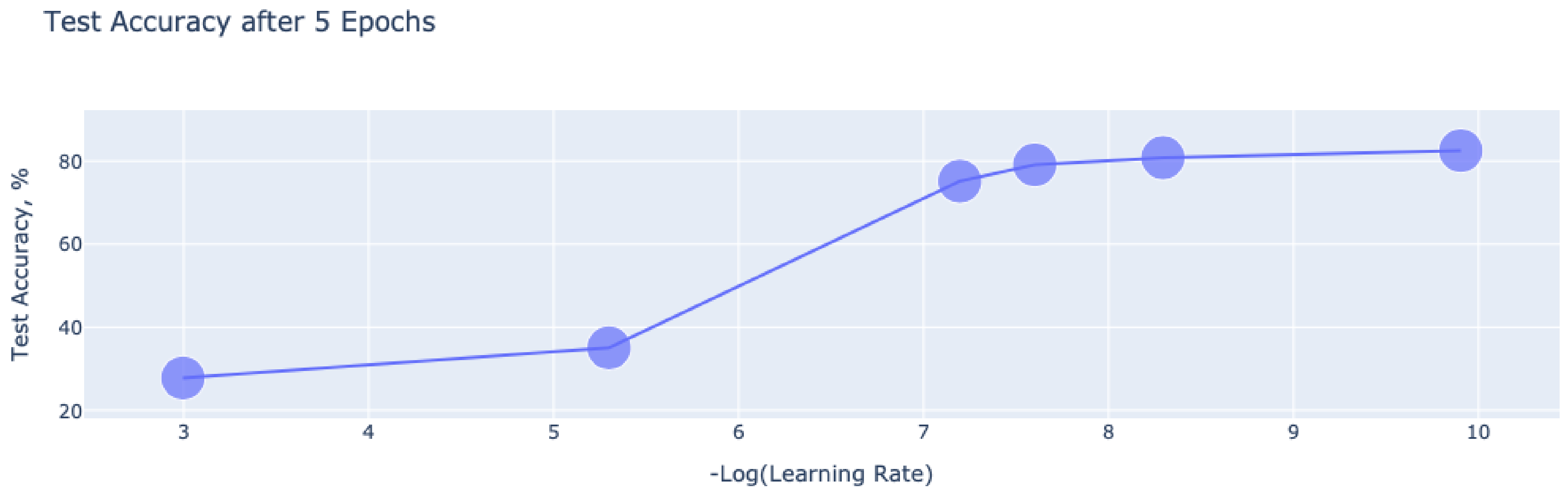
Gaussian Kernel Layer Scaling



Gaussian Kernel Hidden Unit Scaling



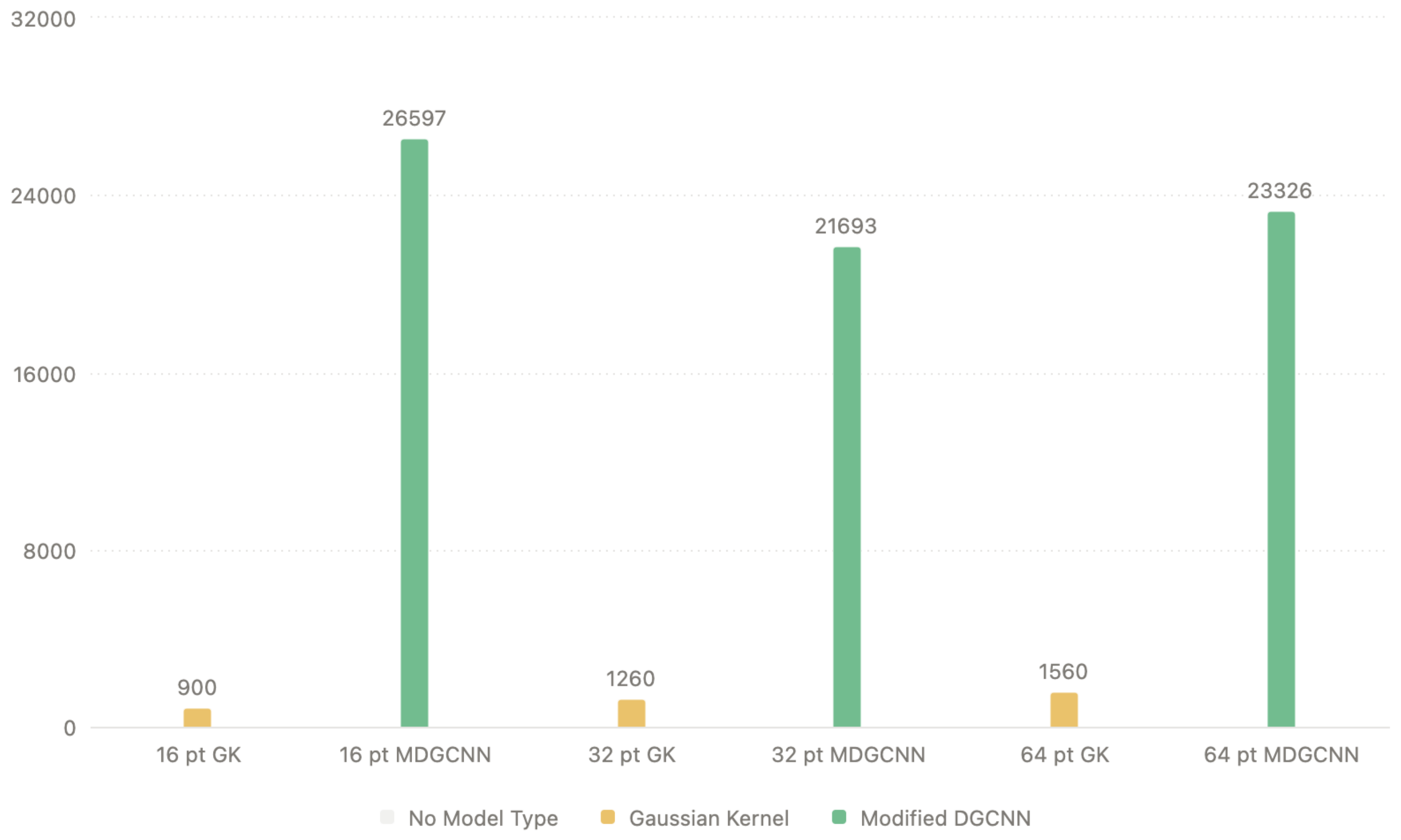
Gaussian Kernel Learning Rate Scaling



- Suggests that the loss landscape is highly convex, because the models do not get trapped at undesirable local minima
- Lower learning rate increases training time; however, it reliably increases the model performance

Model Inference Speed Comparisons

- Number of seconds taken to compute predictions on stsh.ramp simulation

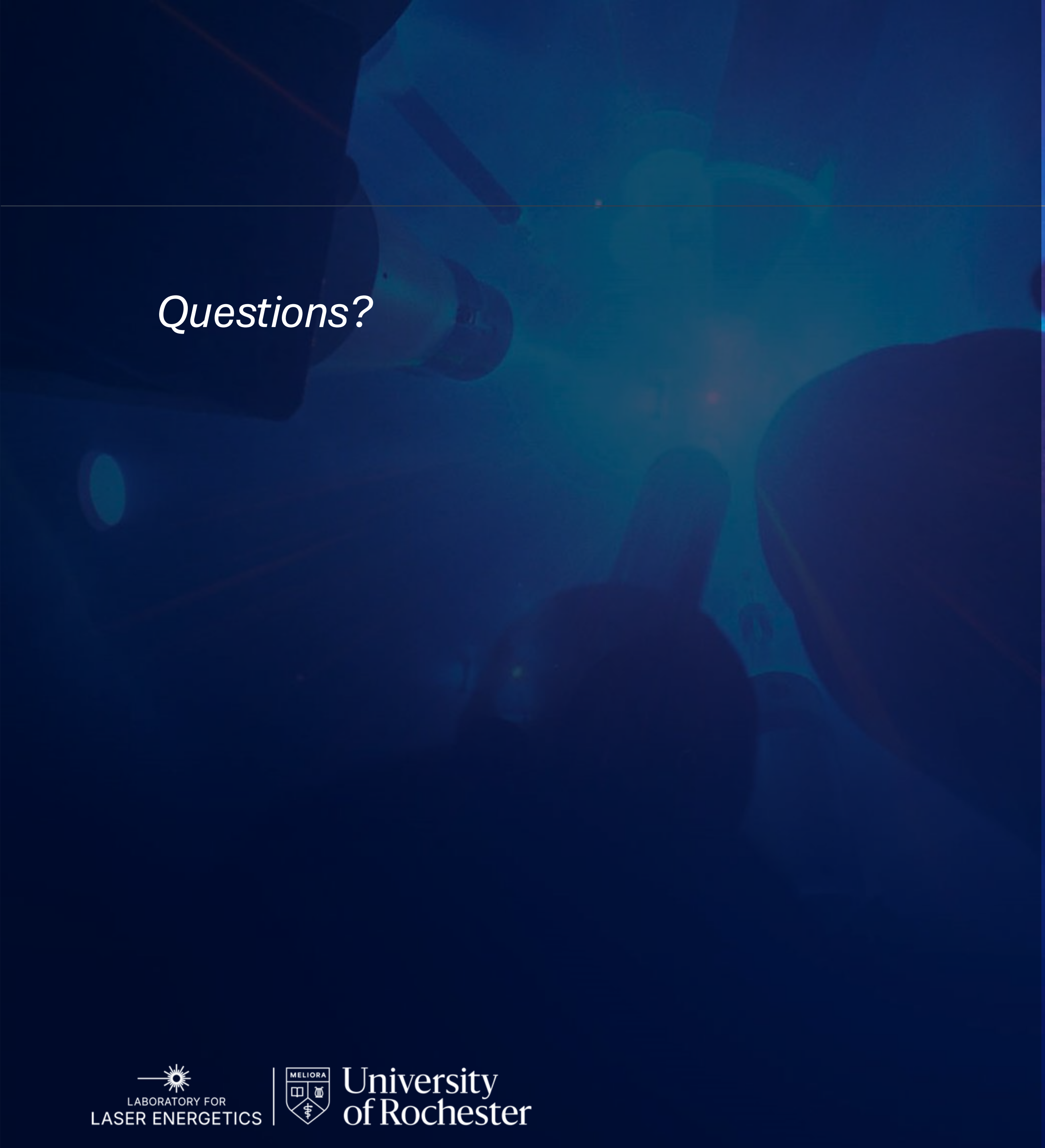


Model Performance Comparisons

N points	Erhard	Modified DGCNN	Gaussian Kernel
16	75%	79%	78%
32	85%	88%	86%
64	92%	93%	90%

Gaussian Kernel Model: Future Work

- Our research has shown that the model’s test accuracy will scale positively with the number of hidden units and negatively with the learning rate
- Develop models which have more hidden units, and a smaller learning rate
- Investigate theoretical reasoning for why this method is effective
- Publish paper explaining method



Questions?

LABORATORY for LASER ENERGETICS
UNIVERSITY of ROCHESTER
UR LLE