PREDICTING PROPERTIES OF THERMAL FLUIDS USING GRAPH NEURAL NETWORK STRATEGIES

***Hari Om Chadha** hariom.chadha@mail.utoronto.ca

***Bhanu Mamillapalli** b.mamillapalli@mail.utoronto.ca

* indicates equal contribution to the project

ABSTRACT

Thermal fluids are vital for improving the performance and lifespan of technologies like lithium batteries and CPUs, but data on their properties across temperatures is limited. Machine learning, especially Graph Neural Networks (GNNs), offers a solution by generating molecular representations that can generalize across different property predictions. By using a modified version of the Chemprop d-MPNN network, results similar to the existing numerical descriptor model were achieved, with MAE around 0.3 and SRCC around 0.94 on a dynamic viscosity dataset. When tested using transfer learning to predict thermal conductivity, this GNN framework, trained on viscosity data, showed an improvement in performance over training using random initial weights. By implementing the Barlow Twins pretraining technique, significant improvements were made in model performance, especially on small data regimes (≈100 data points). By combining these strategies, GNN-based property prediction may be able to achieve excellent performance on properties which have very little data available.

—-Total Pages: 5

1 INTRODUCTION

Thermal fluids are crucial for cooling technologies like lithium batteries and CPUs, and so choosing the best fluid based on their thermophysical properties is a matter of interest. These properties aren't always readily available, but machine learning offers a way to predict these properties using limited data. Current methods use the Mordred library to compute molecular descriptors and train a feedforward neural network (FFN), but this involves extensive feature engineering and may not generalize well across different properties. Graph neural networks (GNNs) offer a potential solution by training on molecular graphs, enabling them to produce learned representations that better capture structural information and improve transferability with limited data points.

2 METHODOLOGY & RESULTS

This section outlines the methods used during this project and analyzes the results. All the machine learning models mentioned in this section make use of the Arrhenius Equation:

$$Ln(\textbf{label}) = Ln(A) + B \cdot \frac{1}{\textbf{T}} \tag{1}$$

Ln(A): Pre-exponential Factor **B**: Activation Energy **T**: Temperature

Instead of directly predicting the final target values, the models are designed to predict the coefficients ln(A) and B. These coefficients are subsequently used to compute the actual target values through the Arrhenius Equation. This approach not only makes the model temperature-independent but also ensures it is informed by underlying physical principles. By leveraging the Arrhenius Equation, a linear relationship is established between temperature and the predicted values, which in turn enhances the model's performance and accuracy.

2.1 CHEMPROP D-MPNN APPROACH

The main GNN architecture used was adapted from the Chemprop library (1). This model called a directed message-passing neural network (d-MPNN), passes information between directed edges instead of nodes. This helps reduce noise and improve performance, as noted in previous studies (2). This model was modified for this approach.

The model accepts a CSV file containing SMILES strings, normalized temperature, normalized target, split type, and the normalized ln(A) target optionally. Normalized temperature is calculated as $T^* = \frac{T}{T_{max}}$, which is then inverted to fit the $\frac{1}{T}$ form found in equation 1. The target column's mean and standard deviation are used to normalize both the target and ln(A) target columns. All data normalization is done using only the train data as a basis.

The loss is calculated based on the property label's natural log and an auxiliary loss based on the predicted ln(A) value if the targets are provided. The model's hyperparameters were optimized over 1,600 trials, with the best model evaluated on test data from a separate dataset. The results are compared to the best numerical descriptor FFN and a base FFN that does not incorporate the Arrhenius equation, shown in Figure 1.

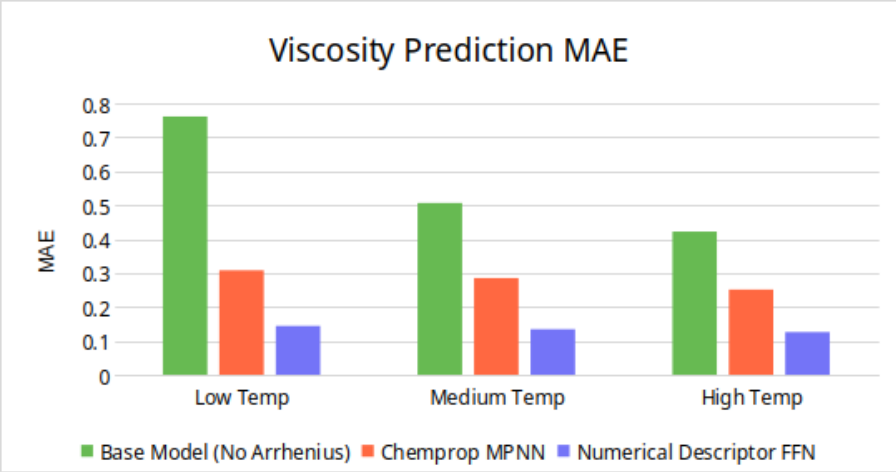


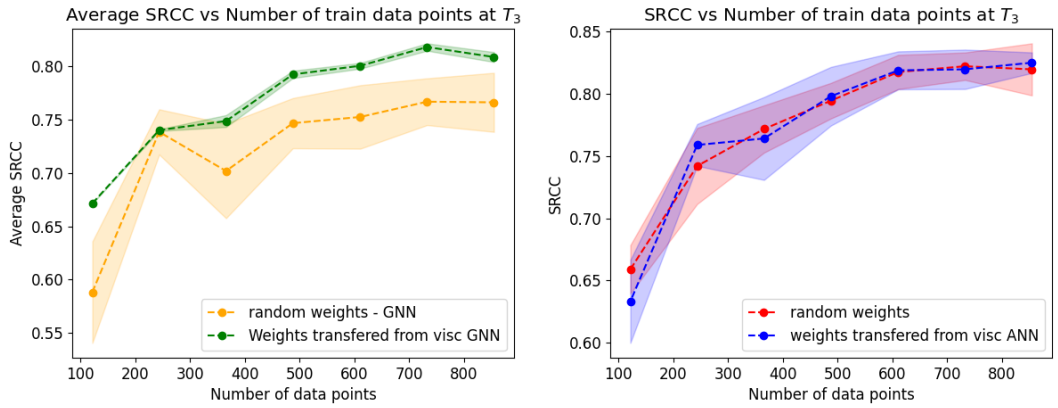
Figure 1: Comparison between GNN and FNN performance: GNN achieves underperforms compared to the FFN with the Arrhenius equation. This is as expected because the FFN (with Arrhenius Equation) uses an extensive feature selection processes to find features that directly relate to Dynamic Viscosity.

2.2 TRANSFER LEARNING APPLICATIONS

GNN training creates a model able to generate embeddings that capture general information that can be applied to predicting properties other than the initially trained one. By freezing the GNN weights and adding a new regression head, the model should be able to quickly adapt to predicting a new property, even with smaller datasets. This was tested by taking the GNN from section 2.1, which was trained on dynamic viscosity data, freezing its weights, and retraining the regression head to predict thermal conductivity.

This process was repeated 10 times across various training set sizes, with the mean and standard deviation of the SRCCs calculated for each size. The same procedure was applied to a simplified numerical descriptor FFN model without the extensive feature selection to compare the results.

As shown in Figure 2a, transfer learning significantly improves the GNN's performance compared to scratch learning, but it has no impact on the FFN model (Figure 2b). This was expected since the FFN relies on numerical descriptors, meaning the weights learned for one property aren't always transferable to another. In addition, this shows that the GNN is capable of learning generalizable representations that can make it easier to predict properties with smaller datasets.



(a) SRCC comparison between Chemprop GNN trained from scratch, and using transfer learning from viscosity data.

(b) SRCC comparison between simple numerical descriptor model trained from scratch, and using transfer learning from viscosity data.

Figure 2

2.3 Barlow Twins Pretraining

While transfer learning has demonstrated promising results, its applicability is limited to scenarios where the target domains are closely related. When targets differ significantly, the model's learned representations struggle to generalize, leading to sub-optimal performance on diverse properties. To combat this issue, we applied an approach called Barlow Twins (3).

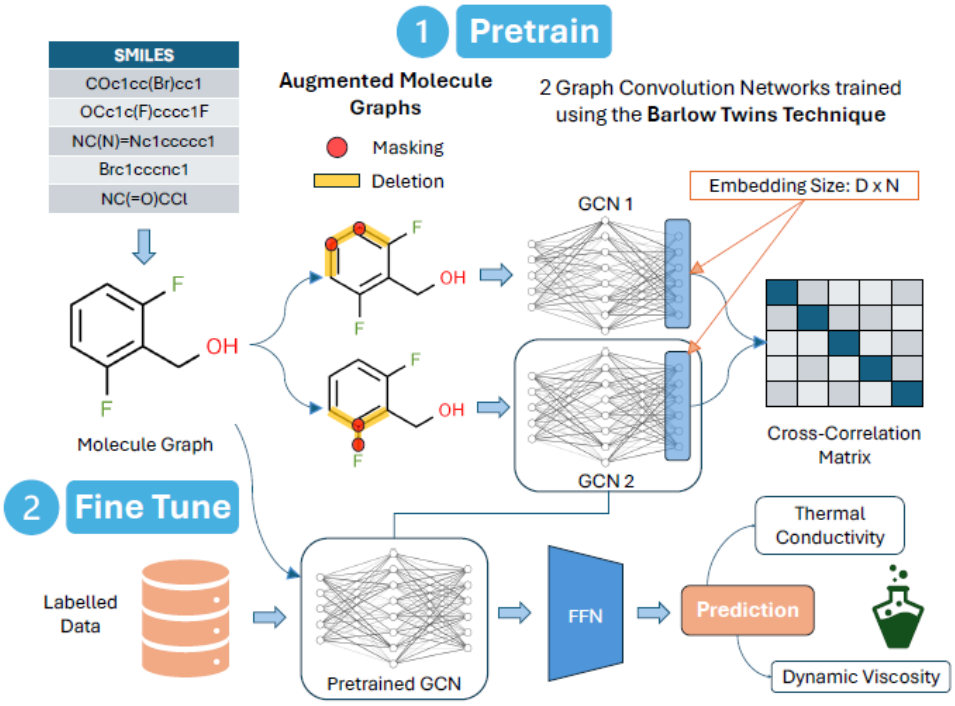
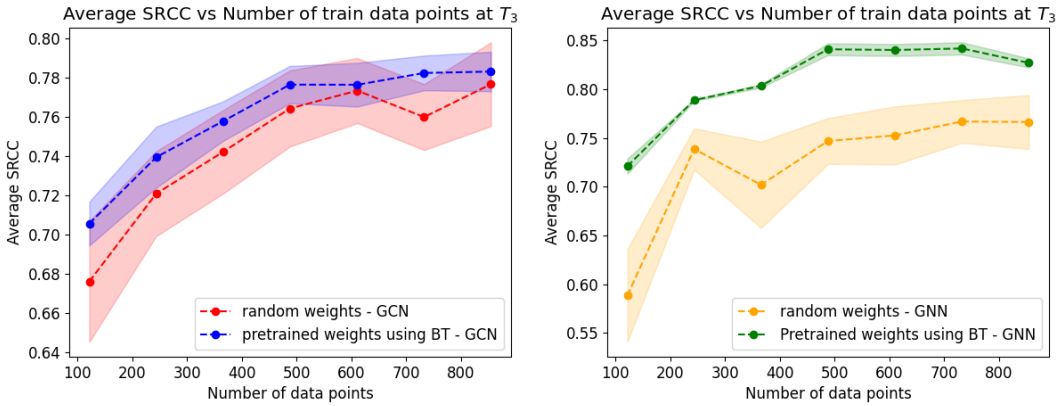


Figure 3: Barlow Twins Workflow: The GCNs in this framework can be substituted with other types of GNNs. However, it's important to recognize that not all model combinations are compatible or able to learn effectively from each other. For instance, when we attempted to train an FFN and a GNN together, the models were unable to learn collaboratively.

Using the Barlow Twins technique (Figure 3), two models are pretrained simultaneously and then fine-tuned on small datasets tailored to specific tasks. Each model receives a different augmentation of the same molecular graph, generated from a SMILES string (4). The training process involves aligning the embeddings of these augmentations by minimizing the difference between the cross-correlation matrix and the identity matrix, which serves as the loss function. This approach enables the models to learn generalized molecular representations, which can later be fine-tuned to predict the physical properties of fluids.



(a) Thermal Conductivity: SRCC comparison between pre-trained GCN and GCN trained from scratch.

(b) Thermal Conductivity: SRCC comparison between pre-trained GNN and GNN trained from scratch.

Figure 4

The Barlow Twins method was first implemented using Graph Convolutional Networks (GCNs), leading to significant improvements over training the model from scratch (Figure 4a). These promising results motivated us to apply this approach to Chemprop, a more advanced GNN architecture.

As shown in Figure 4b, this approach increased the SRCC by at least 0.05 over training from scratch, even surpassing the benefits of transfer learning. The standard deviations also decreased notably after pretraining, likely because the model required only minor adjustments to its weights from a better initialization. The substantial increase in performance with small dataset sizes is particularly noteworthy, suggesting that pretraining could be highly advantageous in situations with limited data.

3 Conclusion & Recommendations

This research showcases the efficacy of pretrained graph neural networks in predicting the thermophysical properties of thermal fluids. In Figure 1 it is shown that using a GNN to get a learned representation of molecules, followed by a regression head to predict Arrhenius parameters, is a competitive way to predict these properties. It is similar in performance to a numerical descriptor approach in terms of MAE. However, it specializes in its capability to utilize transfer learning and pretraining to achieve improved performance on small datasets.

Figure 2a shows the benefit of transfer learning to the performance of a GNN, whereas figure 2b shows the lack of improvement in a FFN strategy using the same method. This result highlights that the GNN network is capable of learning more nuanced representations, which contain information transferable across properties.

Figures 4a and 4b show the large improvement found in GCN and Chemprop GNN performance by utilizing pretraining. This difference is especially pronounced on lower dataset sizes, highlighting the possibility for this method to be used for properties where little data is available.

Overall, these results showcase that using a GNN-based approach for thermophysical property prediction can improve transferability and performance on small dataset sizes. Improvements may be found by optimizing the augmentation steps during pretraining, or by using an equation more accurate than the Arrhenius equation for properties of higher complexity. Further steps should focus on the pretraining strategies using different augmentations and testing on various properties to ensure versatility.

4 GitHub Repository

All relevant code corresponding to this project, including the datasets, processed data, neural networks, and other files can be found on the following GitHub Repositories:
Transfer Learning FFN: `https://github.com/HariOmChadha/FluidNet_TL`
Barlow Twins Using GCNs: `https://github.com/HariOmChadha/MolBTR`
Modified Chemprop: `https://github.com/bhanum1/chemprop`
Chemprop Results: `https://github.com/bhanum1/GNN_TF_Pred`

5 Acknowledgements

All work on this project was done with the help of Mahyar Rajabi and Sartaaj Khan. They provided expertise, advice, and resources that enabled this research. In addition, the project was completed under the supervision of Professor Mohammad Moosavi, who provided us with the technology and support necessary to complete the project. Furthermore, Hari Om Chadha is supported by a summer research studentship from the University of Toronto's Acceleration Consortium initiative, which receives funding from the Canada First Research Excellence Fund (CFREF) and Bhanu Mamillapalli received funding from the NSERC USRA grant.

References

[1] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay, "Analyzing learned molecular representations for property prediction," *Journal of Chemical Information and Modeling*, vol. 59, no. 8, pp. 3370–3388, 2019, pMID: 31361484. [Online]. Available: https://doi.org/10.1021/acs.jcim.9b00237

[2] E. Heid, K. P. Greenman, Y. Chung, S.-C. Li, D. E. Graff, F. H. Vermeire, H. Wu, W. H. Green, and C. J. McGill, "Chemprop: A machine learning package for chemical property prediction," *Journal of Chemical Information and Modeling*, vol. 64, no. 1, pp. 9–17, 2024, pMID: 38147829. [Online]. Available: https://doi.org/10.1021/acs.jcim.3c01250

[3] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 12 310–12 320. [Online]. Available: https://proceedings.mlr.press/v139/zbontar21a.html

[4] Y. Wang, J. Wang, Z. Cao, and A. B. Farimani, "Molclr: Molecular contrastive learning of representations via graph neural networks," *CoRR*, vol. abs/2102.10056, 2021. [Online]. Available: https://arxiv.org/abs/2102.10056

6 Supplemental Information

Additional information and figures that can help provide context to this research are provided below.

6.1 Distribution of Conductivity Data

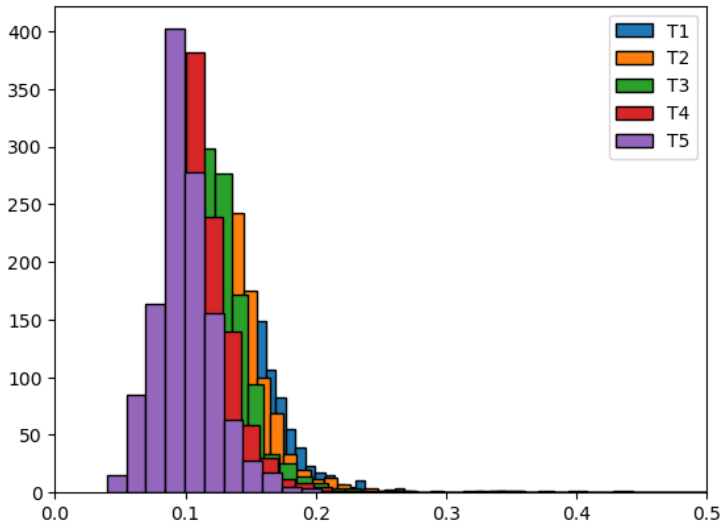


Figure 5: Histogram showcasing the spread of labels in the conductivity data. Due to the normal distribution, there is much less data at high and low thermal conductivities, which indicates that stratified sampling would be beneficial for training.

6.2 Mean Absolute Error Plots

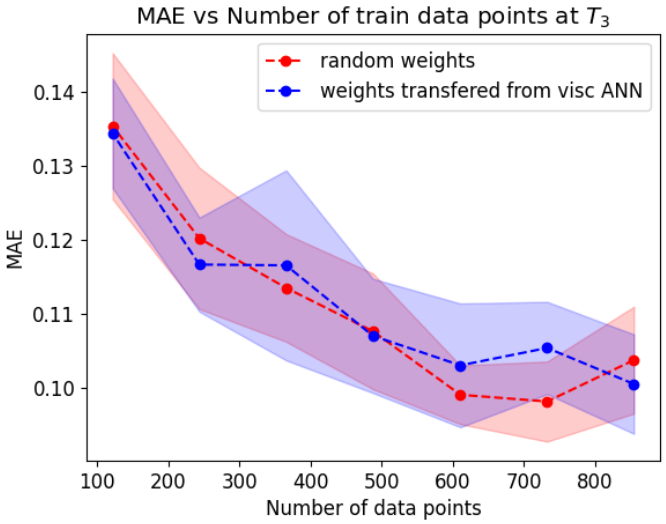


Figure 6: Learning curve with MAE values for the training of the FFN on conductivity data using transfer learning from dynamic viscosity data. No significant improvement can be seen over the scratch model.

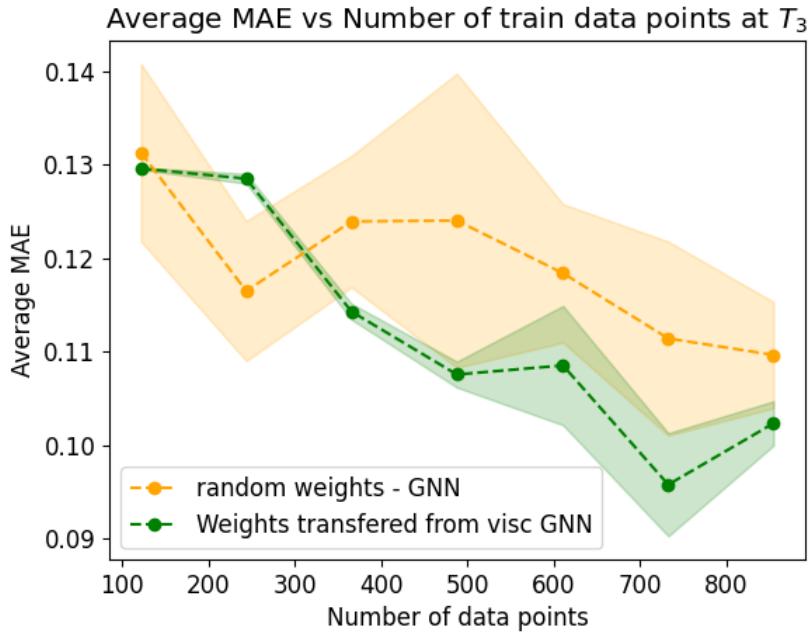


Figure 7: Learning curve with MAE values for the training of the Chemprop GNN on conductivity data using transfer learning from dynamic viscosity data. Significant improvement over the scratch model can be seen on larger data, but no improvement at the small data section.

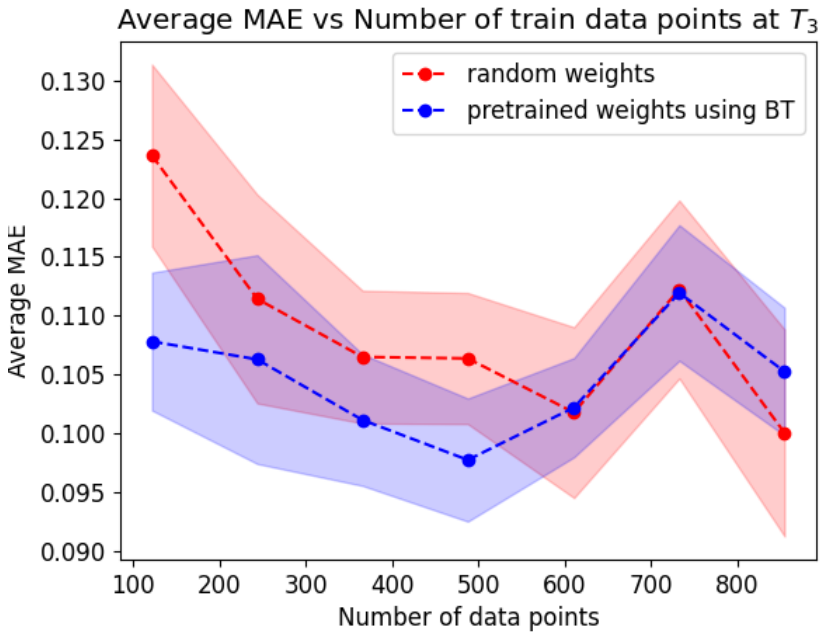


Figure 8: Learning curve with MAE values for the training of the GCN version of the GNN on conductivity data using BT pretraining. A small improvement over the scratch model can be seen on smaller data, but it is identical or worse at larger data sections.

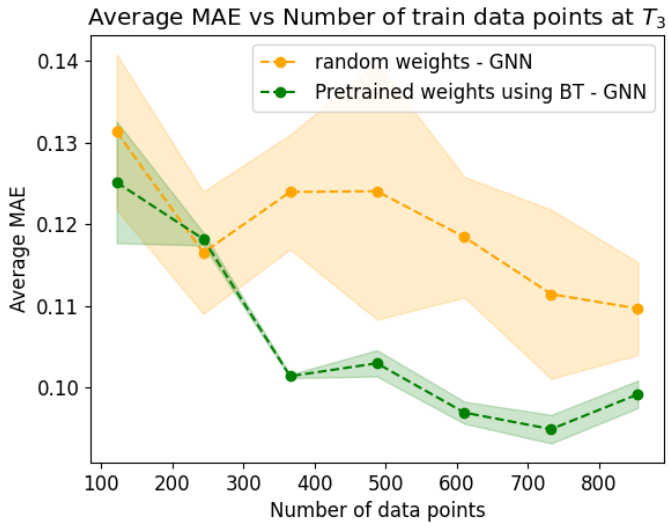


Figure 9: Learning curve with MAE values for the training of the Chemprop GNN on conductivity data using BT pretraining. No improvement over the scratch model can be seen on smaller data, but it is significantly better at larger data sections.

From figures 6 and 7, the same trend is shown in section 2.2. The FFN does not improve via transfer learning, but the GNN does. In figures 8 and 10, the MAE learning curves are shown for Barlow twins training, and showcase the same trends as in section 2.3. Interestingly, for the Chemprop GNN, the MAE values are not better at lower data sizes, although the SRCCs were improved.

6.3 Parity Plot

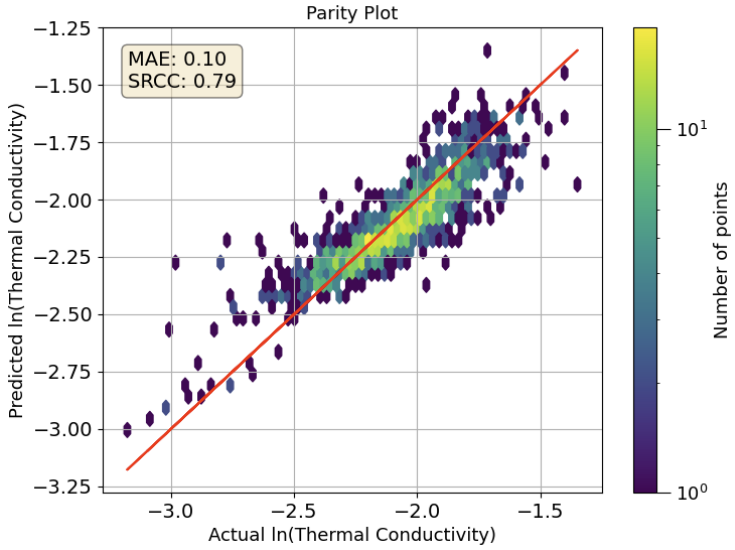


Figure 10: Example parity plot for the Barlow Twins pretrained GCN network for thermal conductivity data. Finetuning was done on approximately 350 datapoints.