Advanced Deep Learning Methods for Chemistry and Material Science

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Zeren Shui

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Professor George Karypis, Advisor

July, 2024

Acknowledgements

First and foremost, I would like to extend my sincerest gratitude to my advisor, George Karypis, for his guidance, patience, and support. With his exceptional research and working experience, George can always provide keen and insightful guidance for my research. When I started my Ph.D. journey, I had very limited research experience. George patiently trained me step-by-step from a novice to a rigorous researcher. He taught me all skills that an awesome researchers need from fundamental ones such as paper writing, presentation, experimental data analysis to advanced ones like how to identify research questions, develop critical thinking skills, and conduct impactful research. George has been extremely supportive for both my research and my life. He always checks if I have the necessary resources for conducting my research and enthusiastically introduces me to researchers and opportunities. In all the hard times such as COVID, Minneapolis riots, and when I faced with personal crisis, George steadily stood by me and provided help to make me feel safe and supported. I am very thankful to have George as my advisor and role model. I am always proud that I am a student of George, and I hope one day George will be proud of me.

I would like to thank Professors Vipin Kumar, Chad Myers, Ellad B. Tadmor for serving on my dissertation committee and Professor Mingyi Hong, Zhi-Li Zhang for serving as my oral preliminary exam committee. Their insightful feedback, rigorous academic standards, and encouragement have significantly shaped my research and professional growth.

I want to thank all the members from Karypis Lab, Costas Mavromatis, Ancy Sarah Tom, Agoritsa Polyzou, Maria Kalantzi, Saurav Manchanda, Athanasios N. Nikolakopoulos, Evangelia Christakopoulou, Shaden Smith, Mohit Sharma, Petros Karypis, and Philipos Karypis. I have learned a lot and received significant support from them.

The time we spent together forms an essential part of my Ph.D. journey.

I would like to thank my amazing collaborators from the KIM-Initiative, especially Professor Ellad B. Tadmor, and Professor Stefano Martiniani for providing support, help, and suggestions during our collaborations. It is my great pleasure to work with them. Our collaborations constitute a significant part of this dissertation. I am glad that I have worked with many wonderful researchers: Mingjian Wen, Daniel Karls, Ilia Nikiforov, Amit Gupta, Eric Fuemmeler, and Brendon Waters.

I want to thank my friends from Digital Technology Center and University of Minnesota, Professor Georgios B. Giannakis, Yilang Zhang, Kostas Polyzous, Bingcong li, Jia Yan, Shijian Gao, Vassilis N. Ioannidis, Yanning Shen, Jason Carpenter, Yingxue Zhou, Burhan Yaman, Bhaskar Sen for their support throughout my Ph.D. journey.

I am very thankful to receive support and mentorship from my Amazon collaborators: Ge Liu, Da Zheng, Yifei Ma, Hao Ding, Bernie Wang, and Anoop Deoras. I have learned invaluable research and industrial experiences from them.

I want to thank my friends, Fan Yang, Hao Sun, Sijia Ai, Pinshu Chen, Sijie He, and Shiao Wang for their friendship. As an only child, they have become like siblings to me, offering both companionship and support throughout my journey.

I want to thank my host family, Jim Goddard, Angela Goddard, Lolo Goddard, and Norah Goddard for for providing me with a family away from home in Minnesota. They kindly welcomed me and helped me adapt to U.S. culture upon my arrival in this new country. Throughout my time in Minnesota, they embraced me as one of their own, truly making me feel at home.

I want to extend my heartfelt thanks to my parents, Yiping Shui and Jinfeng Li, for their unconditional love, support, protection, and sacrifices throughout my life. They have consistently prioritized my needs and my research above their own, striving tirelessly to assist me in every possible way. Their selflessness and dedication have been the cornerstone of this dissertation.

I want to thank my wife, Jiaqi Chen, for going through one of the most challenging journeys imaginable: maintaining a long-distance relationship during a Ph.D. program, with me. We made it through together and turned it into the greatest achievement in our lives. This achievement would not have been possible without her love, companionship, patience, and immense sacrifices.

Dedication

To my grandfather Jisheng Shui, my parents Yiping Shui and Jinfeng Li, and my wife Jiaqi Chen.

Abstract

In chemistry and material science, scientific discovery is usually achieved through a combination of wet-lab experiments and first-principle computational methods. These traditional approaches are often time-consuming and computationally expensive, significantly slowing down the pace of discovery. In recent years, researchers have started exploring deep learning methods to accelerate this process and reduce the cost. While these initial attempts have shown great promise, there remains significant challenges that must be addressed to fully realize the potential of deep learning in this field.

This dissertation advances research on deep learning methods for molecular and material property predictions from three key perspectives: 1) Molecular Representation Learning: We propose an expressive neural network (HMGNN) that can learn better molecule representations and achieves state-of-the-art performance in molecular property prediction tasks. 2) Multi-modal Molecular Learning: we develop a retrieval augmentation method (RTMol) that leverages additional information present in scientific literature to augment molecular structures for accurate property prediction. 3) Label Efficiency: we propose two methods to effectively train neural networks for material and molecular property prediction with limited labeled data. The first method utilizes materials labeled by computationally efficient labeling methods to augment the limited labeled training data. The second method (DSP) selects task specific pre-training subsets to effectively adapt already pre-trained neural networks to downstream tasks.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Scientific discovery in chemistry and material science (e.g., drug discovery, material design) rely heavily on experimental screening and first principle computation. Experimental screening involves synthesizing and characterizing molecular and material candidates in order to identify those with desired properties (e.g., solubility, toxicity). First principle computation (e.g., density functional theory) is usually used to predict quantum properties of molecules and materials, and simulate their behaviours at the atomic level. These approaches are time-consuming and (computationally) expensive, making it challenging to evaluate large number of molecules and materials.

Due to the increasing availability of large datasets and computation resources, deep learning has revolutionized a wide range of application domains such as computer vision, natural language processing, and graph mining. In recent years, researchers have begun exploring deep learning methods to accelerate the scientific discovery process in chemistry and materials science. These methods could allow for efficient prediction, analysis, and screening of large numbers of molecules and materials, helping to guide the selection of those most likely to succeed in further development. Moreover, they can serve as surrogates of first principle computational methods for atomistic modeling tasks.

While deep learning has shown great promise, it still faces three critical challenges that must be addressed to fully realize its potential in chemistry and material science. First, molecules and materials possess intricate geometric structures (e.g., bond angles)

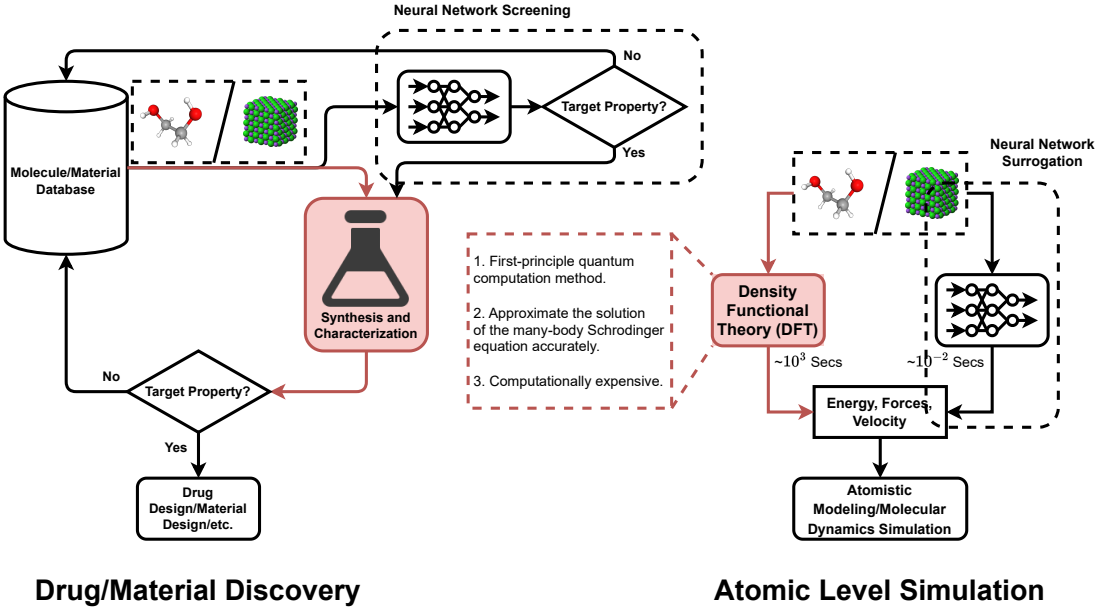


Figure 1.1: Deep learning accelerates scientific discovery in chemistry and material science.

and intrinsic patterns (e.g., many-body interactions). Second, molecules can be represented by data of different modalities (e.g., molecular graphs, scientific documents) that contain different information. Third, deep learning methods require substantial amounts of labeled data for generalization performance, but the process of data labeling is often cost-prohibitive.

This dissertation addresses the aforementioned challenges and advance the research on deep learning methods for molecular and material property predictions. In Chapter 2, we propose a novel graph representation of molecules, heterogeneous molecular graph (HMG), that allows us to explicitly model interactions, representations, and predictions associated with many-bodies as well as complex geometric information (e.g., bond angles and dihedral angles). We design heterogeneous molecular graph neural networks (HMGNN) to leverage the rich information stored in HMG for molecular property prediction tasks. In Chapter 3, we propose a text retrieval augmentation framework to

efficiently and effectively learn to retrieve informative documents from scientific literature to augment molecular property prediction. We address the label scarcity challenge from two perspectives, to use cheap but high-quality supervision signals to train neural networks and to pre-train neural networks using existing large and diverse datasets. In Chapter 4, we discuss label augmentation and multi-task pre-training, two strategies that utilize computationally efficient while relatively accurate labeling methods on unlabeled materials to assist the training of neural networks for material property prediction. In Chapter 5, we propose data selection pre-training (DSP) that adapts already pre-trained neural networks to the domain of specific downstream tasks without the need to label additional domain-specific datasets.

Chapter 2

Heterogeneous Molecular Graph Neural Networks

2.1 Introduction

Predicting quantum mechanical properties of molecules based on their structures is important for molecule screening and drug design. We can compute exact molecular properties by solving the many-body Schrödinger equation. However, closed form solution to this equation is only available for simple systems. Although researchers developed methods such as Density Functional Theory (DFT) [64] to approximate the solution, the computational cost of these methods scales poorly and is worse than $\mathcal{O}(n^3)$ w.r.t. the number of electrons.

Recently, researchers have been developing machine learning methods that are orders of magnitude faster with a moderate compromise in prediction accuracy. Among the machine learning approaches, graph neural network (GNN)-based methods attract a lot of research attention as their ability to model complex interactions among atoms. These methods treat molecules as molecular graphs (e.g., distance graphs [141, 117, 121, 95], chemical graphs [48], K-nearest neighbor graphs [71]) in which atoms are modeled as nodes. They compute an atom's low-dimensional representation as a function of its feature and characteristics of its graph neighbors. The low-dimensional representations are then used to estimate the local contribution of the atoms to the desired property, or to compute a global representation of the molecule for downstream predictions.

The many-body expansion (MBE) [131, 35, 163] is an important scheme that computes the energy of an N-particle system as the sum of the contributions of many-body terms

$$E = \sum_i E_i + \sum_{i<j} E_{ij} + \sum_{i<j<k} E_{ijk} + \cdots + E_{12\cdots N}, \tag{2.1}$$

where E_i is the local energy contribution of a single atom, E_{ij} is the energy contribution of a two-body (a group of two atoms), E_{ijk} is the energy contribution of a three-body, and eventually $E_{12\cdots N}$ is the contribution of the body formed by all the atoms in the molecule. Note that, the local contribution to the total energy decreases fast with the number of atoms in the many-body. As most of the existing GNN-based methods are developed on molecular graphs, they focus mainly on modeling atom-based representations, interactions, and predictions which correspond to the first two terms of the series and do not have an explicit characterization of the higher order terms. This may compromise their accuracy in the chemical prediction problems.

In this dissertation, we introduce a novel graph representation of molecules, *heterogeneous molecular graph* (HMG), to explicitly model many-body interactions. A p-body (the value of p is called the *order* of the many-body) is a group of p atoms that functions as a whole entity. In HMGs, a p-body is modeled as a node of order p. Nodes connect to nodes of the same or different order via different types of edges. This heterogeneous structure allows us to explicitly model interactions, representations, and predictions associated with many-bodies. Moreover, edges between nodes of the same order carry the potential of incorporating complex geometric information (e.g., bond angles and dihedral angles) into node embeddings.

To leverage the rich information stored in HMG for tasks of molecular property predictions, we design heterogeneous molecular graph neural networks (HMGNN) by following a message passing framework. In the message passing framework [48], nodes send and receive messages from their neighbors and update their low-dimensional representations using the received messages. HMGNN is a multi-task learning [115] model whose design is inspired by the MBE of energy surfaces. In HMGNN, each many-body order possesses its own set of parameters and shares computations with other orders. In the prediction phase, HMGNN computes one estimation for each many-body and aggregates them based on their orders. It uses an attention-based model that takes

into account a global representation of the molecule to fuse the prediction of different orders, which correspond to different terms in Eq 2.1. We design a multi-task learning loss that enforces the prediction of each order and the fused prediction to be close to the true target. Experimental results show that the fused prediction is better than any of the standalone predictions. The fusing weight of the predictions are also consistent with the convergence assumption in the many-body expansion.

The main contribution of this work lies in two folds. First, we propose HMG which allows graph learning methods to explicitly model many-body representation, interaction, and prediction. Second, we develop a multi-task learning method HMGNN for the task of molecule property prediction. HMGNN explicitly incorporates many-body interaction and a global molecule representation into the prediction process and achieves state-of-the-art performance on the QM9 dataset [114, 108].

2.2 Related Works

Traditionally, prediction of many important molecular properties such as atomization energies relies on methods that approximate the solution of the many-body Schrödinger equation such as density function theory (DFT) and its variants [101]. This class of methods involves solving complex linear systems and has a computational complexity worse than $\mathcal{O}(n^3)$ where n is the number of atoms.

Recent years have seen a surge in data-driven methods that train machine learning models to learn patterns from molecule databases. The learned patterns are assumed to be general in chemical space and can be used to estimate properties of unknown compounds. These attempts started from [36, 8] which feed hand-crafted molecule descriptors (e.g., Coulomb matrix, bag of bonds) into regression models such as linear regression and random forests. These methods rely heavily on the quality of the crafted descriptors and have limited representation power.

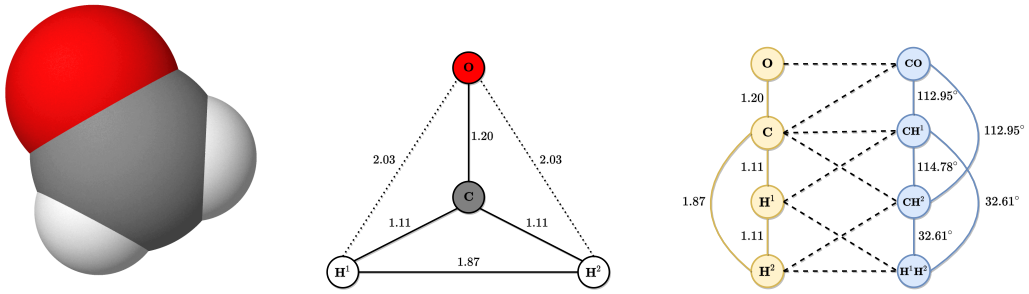
Recently, graph neural networks (GNN) have been achieving a great success in graph-related applications [76, 145, 57, 165]. In chemistry, researchers developed GNN-based method for learning tasks over graph represented molecules. The authors of [48] introduced a generic framework over chemical graphs that models interactions between

atoms in a message passing fashion. In [120, 117, 121], the authors designed neural network structures that have no dependency on hand-crafted features but learn molecule representations from only atom types and coordinates. Since GNNs possess a hierarchical structure, i.e., they iteratively apply GNN layers on graphs to encode each node's multi-hop neighbors into its embedding, GNN-based methods [96] and [141] further decompose atom-wise prediction to layer-wise atom prediction to fit in the MBE framework. Although these methods include many-body contributions into final predictions, they do not have an explicit modeling of many-body representations and interactions. Some recent works have incorporated many-body interactions and representations by updating edge embeddings along message passing [71] or by passing messages on line graphs of the corresponding molecular graphs [78]. However, these methods capture only partial many-body interactions and lack many-body predictions.

Equivariant neural network is another class of neural network methods that has been applied in chemical prediction problems. The notion of group equivariant neural network was first introduced by [24] in the domain of image processing. Later, researchers developed neural network methods that are equivariant to continuous rotations for learning representations for 3D objects, including molecules [3, 139, 80]. These methods achieve rotation invariance by transforming objects from Euclidean space to Fourier space and conducting computations in Fourier space. In these methods, each many-body interacts only with itself but not other many-bodies. Thus, they are not optimal in predicting molecule properties.

2.3 Notations and Definitions

We denote matrices by bold upper-case letters (e.g., $\mathbf{W}$), and vectors by bold lower-case letters (e.g., $\mathbf{x}$). We denote entries of a matrix/vector by lower-case letter with subscripts (e.g., x_{ij}/x_i). We use superscripts to indicate variables at the t-th message passing layer (e.g., $\mathbf{h}^{(t)}$). We denote *molecular graphs* by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes (atoms) and edges, respectively. Two atoms are connected in a molecular graph when the Euclidean distance between them is less than a cutoff threshold $c > 0$. Each edge in the graph is associated with a distance to store the geometric structure of the molecule. We define a p-body in a molecular graph $\mathcal{G}$ as a



((a)) Formaldehyde molecule. ((b)) A molecular graph of the formaldehyde molecule. ((c)) A heterogeneous molecular graph of the molecular graph.

Figure 2.1: An example of heterogeneous molecular graph (HMG). Figure 2.1(a) is a spatial structure of a formaldehyde (CH_2O) molecule. Each atom in the molecule is associated with a three-dimensional coordinates in the Euclidean space. Figure 2.1(b) is the molecular graph of the methanol molecule with a cutoff distance $c = 2$. We convert atom coordinates to pair-wise distances to guarantee translation and rotation invariance of the representation. We denote edges whose distances are less than c using black solid lines, and edges that are broke by the cutoff using black dotted lines. Figure 2.1(c) is a HMG of order two constructed from the molecular graph. There are two types of nodes (1-bodies and 2-bodies denoted by yellow and blue circles, respectively) and three types of edges (1-1 and 2-2 denoted by yellow and blue lines, respectively, 1-2 denoted by black dashed lines) in the HMG. Edges between nodes of the same order are associated with features that depict the geometric relation between the nodes (distance for 1-1 edges, angle for 2-2 edges).

p-clique of the graph. We refer to the value of p as the order of the many-body.

2.4 Heterogeneous Molecular Graph and Many-Body Interactions

In this section, we illustrate the construction of *heterogeneous molecular graphs* (HMG) and how we leverage the heterogeneous structure of HMGs to model many-body representations and interactions.

2.4.1 Heterogeneous Molecular Graph

An HMG is a graph in which nodes are many-bodies and edges are defined by various types of geometric and set relations. HMGs are constructed from molecular graphs. We denote an HMG of order N of a molecular graph $\mathcal{G}$ as $H_N(\mathcal{G}) = (\{\mathcal{V}_p\}, \{\mathcal{E}_{pq}\})$ where $1 \leq p \leq q \leq N$, $\mathcal{V}_p$ is the set of p-bodies in $\mathcal{G}$ (i.e., all p-cliques of $\mathcal{G}$), and $\mathcal{E}_{pq}$ is the set of edges between $\mathcal{V}_p$ and $\mathcal{V}_q$. We denote the order p of p-bodies as the node type and p-q as the type of the edges that connect nodes of order p and nodes of order q. Given two nodes $i \in \mathcal{V}_p$ and $j \in \mathcal{V}_q$, when they are of the same order, i.e., $p = q$, i and j are connected if they share $p - 1$ atoms. A special case is when $p = q = 1$, instead of building a complete graph, we use the edge set $\mathcal{E}$ of the molecular graph to define connections. When the two nodes are of different orders, presumably $p < q$, $(i, j) \in \mathcal{E}_{pq}$ if i is a sub-graph of j. An example HMG is shown in Figure 2.1. With this formulation, we are able to explicitly model up to N-body representations by node embeddings and $N + 1$-body interactions by message passing.

In an HMG, each node i of order p is associated with a discrete feature $Z_{p,i}$ that indicates its atomic composition, and a continuous feature $\mathbf{x}_{p,i}$ that describes aspects of its geometry. Note that, nodes of order 1 do not have continuous features since they are points in the Euclidean space and do not have geometric structure. Each edge (i, j) is associated with an edge feature $\mathbf{e}_{p,ij}$ when i and j are of the same order p. The edge feature characterizes the geometric relation between the two nodes, e.g., distance between atoms, angles between bonds. In this dissertation, we use a hash function to map the set of atomic numbers of the atoms to $Z_{p,i}$. Construction of continuous

node features and edge features requires feature engineering especially when order of the many-bodies are high.

2.4.2 Message Passing on Heterogeneous Molecular Graphs

The message passing framework consists of two phases, message passing and node update. On molecular graphs, each node (atom) i sends/receives messages to/from its neighbors and uses the received messages to update its embedding

$$\begin{aligned} \mathbf{m}_i^{(t)} &= \sum_{j \in \mathcal{N}(i)} f^{(t)} \left(\mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}, \mathbf{e}_{ij} \right) \\ \mathbf{h}_i^{(t+1)} &= g^{(t)} \left(\mathbf{h}_i^{(t)}, \mathbf{m}_i^{(t)} \right). \end{aligned} \tag{2.2}$$

In Eq-2.2, $\mathcal{N}(i)$ is the set of neighbor nodes of i, $\mathbf{h}_i^{(t)}$ is the node (atom) embedding of i, $\mathbf{m}_i^{(t)}$ is the aggregation of messages from i's neighbor nodes, $\mathbf{e}_{ij}$ is the edge feature associated with the edge between i and j, $f(\cdot)$ is a message function that maps embeddings of the sender and the receiver and the corresponding edge feature to a message vector, $g(\cdot)$ is a node update function that combines the incoming message and the old embedding to be the new node embedding. Both $f(\cdot)$ and $g(\cdot)$ are learnable. Message passing on HMGs is different from that on molecular graphs due to the heterogeneous property of HMGs. Nodes in HMGs are of different orders and they pass messages through edges of different types. A message passing framework needs to learn edge type specific message functions and order specific node update functions to capture this heterogeneous structure. Moreover, the framework should allow inter-order message passing such that the node embeddings can capture information from other orders. For example, by passing messages from 2-bodies, 1-bodies can encode edge angle information into their embeddings. Let $i \in \mathcal{V}_p$ be a node of order p in a HMG and $\mathbf{h}_{p,i}^{(t)}$ be its embedding at the t-th layer, we design the message passing framework as

$$\begin{aligned} \mathbf{m}_{q,i}^{(t)} &= \sum_{j \in \mathcal{N}_q(i)} f_{qp}^{(t)} \left(\mathbf{h}_{p,i}^{(t)}, \mathbf{h}_{q,j}^{(t)}, \mathbf{e}_{ij} \right) \\ \mathbf{h}_{p,i}^{(t+1)} &= g_p^{(t)} \left(\mathbf{h}_{p,i}^{(t)}, \mathbf{m}_{1,i}^{(t)}, \mathbf{m}_{2,i}^{(t)}, \cdots, \mathbf{m}_{N,i}^{(t)} \right) \end{aligned} \tag{2.3}$$

where $\mathcal{N}_q(i)$ the set of nodes of order q that are connected to i, $\mathbf{m}_{q,i}^{(t)}$ denotes the aggregated messages from nodes i's neighbor nodes of order q, $\mathbf{e}_{ij}$ denotes the edge

feature between i and j if they are of the same order, $f_{pq}(\cdot)$ and $g_p(\cdot)$ are learnable functions specific to edge type pq and node type (order) p, respectively. Compare to the message passing framework on molecular graphs which has two functions to learn, this framework possesses larger model capacity and is able to model many-body interactions explicitly.

2.5 Heterogeneous Molecular Graph Neural Networks.

We present Heterogeneous Molecular Graph Neural Networks (HMGNN) for the purpose of predicting molecule properties. An HMGNN contains four types of modules, input module, interaction module, output module, and fusion module. All the modules except the fusion module are order specific. HMGNNs learn functions for message passing on heterogeneous molecular graphs to compute local node representations, and uses a readout function to combine the representations to form a global molecule representation. HMGNNs compute node-wise contributions to the target property and aggregates them based on their orders. The final prediction is a weighted combination of the predictions of all orders where the weights are computed by an attention mechanism from the global molecule representation. An HMGNN is learned by optimizing a loss function which forces predictions of each order and the fused prediction to be close to the true target. Since the construction of heterogeneous molecular graphs and associated features rely on atom pairwise distances and atomic numbers but not atom coordinates, HMGNNs are invariant under both translations and rotations. HMGNNs are also permutation invariant to atom indices as the message aggregation function in Eq-2.3 and the readout function are permutation invariant [160]. Figure 2.2 shows an overview of the architecture of HMGNN.

2.5.1 Input Module

The input module of HMGNN converts raw features of nodes to latent embeddings. As we described in Section 2.4.1, each node $i \in \mathcal{V}_p$ in a HMG is associated with a discrete feature $Z_{p,i}$ and a continuous feature $\mathbf{x}_{p,i}$. We use an embedding lookup table to map the discrete feature $Z_{p,i}$ to a real value vector $\mathbf{e}_{Z_{p,i}}$ and apply a fully connected layer to the concatenation of the latent vector $\mathbf{e}_{Z_{p,i}}$ and the continuous feature $\mathbf{x}_{p,i}$ to get the

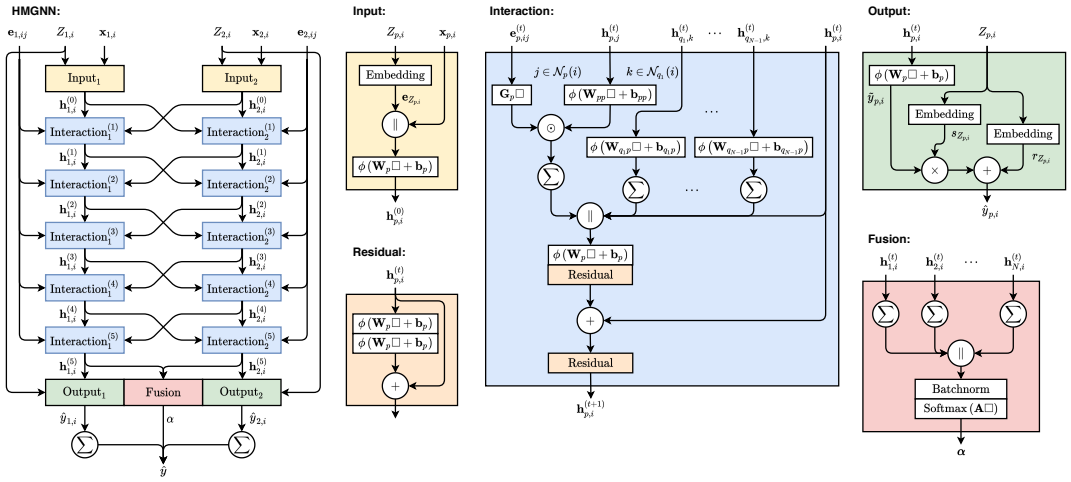


Figure 2.2: Computation flow of heterogeneous molecular graph neural networks (HMGNN) for many-bodies up to order two. We use □ to represent the input to the function. The activation function is set to be the shifted softplus function, i.e., $\phi(x) = \ln(0.5e^x + 0.5)$. Each many-body order p owns its input module, interaction module, and output module. For each node i of order p, an input module converts the discrete and continuous feature of the node to an initial node embedding $\mathbf{h}_{p,i}^{(0)}$. HMGNN passes the initial embeddings through a stack of T interaction modules to encode information from its neighbor nodes of different orders to the node embedding. The outputs of the last interaction module, the final node embedding $\mathbf{h}_{p,i}^{(T+1)}$, are then fed into a fusion module and an output module to compute a weight vector α and prediction $\hat{y}_{p,i}$, respectively. HMGNN sums the predictions per many-body order and computes the final prediction as a weighted sum of these summed predictions.

initial node embedding

$$\mathbf{h}_{p,i}^{(1)} = \phi\left(\mathbf{W}_p^{\text{in}}\left(\mathbf{e}_{Z_{p,i}} \parallel \mathbf{x}_{p,i}\right) + \mathbf{b}_p^{\text{in}}\right)$$

where $\mathbf{W}_p^{\text{in}}$ and $\mathbf{b}_p^{\text{in}}$ are learnable parameters for nodes of order p (p-bodies), $\phi(\cdot)$ is an element-wise activation function, $\parallel$ denotes concatenation of vectors.

2.5.2 Interaction Module

HMGNN stacks T interaction modules to encode information across far reaches of the heterogeneous molecular graph into node embeddings. Each interaction module takes the output embeddings of the previous module and update the embeddings. Note that, edges between nodes of the same orders have features while other edges do not. As a result, we paramatrize the message functions between nodes of the same order as

$$\mathbf{m}_{p,i}^{(t)} = \sum_{j \in \mathcal{N}_p(i)} \mathbf{G}_p^{(t)} \mathbf{e}_{p,ij} \odot \phi\left(\mathbf{W}_{pp}^{(t)} \mathbf{h}_{p,j}^{(t)} + \mathbf{b}_{pp}^{(t)}\right) \tag{2.4}$$

and the message functions along edges without features as

$$\mathbf{m}_{q,i}^{(t)} = \sum_{j \in \mathcal{N}_q(i)} \phi\left(\mathbf{W}_{qp}^{(t)} \mathbf{h}_{q,j}^{(t)} + \mathbf{b}_{qp}^{(t)}\right). \tag{2.5}$$

In Eq-2.4 and Eq-2.5, $\mathcal{N}_p(i)$ and $\mathcal{N}_q(i)$ denotes the set of neighbor nodes of order p and order q of node i, respectively, $\odot$ denotes the Hadamard product, $\mathbf{G}$, $\mathbf{W}$, and $\mathbf{b}$ are learnable parameters. A node embedding $\mathbf{h}_{p,i}^{(t)}$ is then updated as a function of its old embedding and the incoming messages,

$$\mathbf{h}_{p,i}^{(t+1)} = \mathbf{h}_{p,i}^{(t)} + \phi\left(\mathbf{W}_p^{(t)}\left(\mathbf{h}_{p,i}^{(t)} \parallel \mathbf{m}_{1,i}^{(t)} \parallel \cdots \parallel \mathbf{m}_{N,i}^{(t)}\right) + \mathbf{b}_p^{(t)}\right),$$

where $\parallel$ denotes concatenation of vectors. The interaction module then refines the node embeddings with two consecutive fully connected layers with residual connections [59].

2.5.3 Output Module

Each many-body order p possesses a specific output module that passes the output of its interaction module, final node embeddings $\mathbf{h}_{p,i}^{(T+1)}$, through a sequence of linear mappings and a aggregation process to compute the estimated value of the target property.

First, we use a fully connected layer to convert the node embeddings to node predictions

$$\tilde{y}_{p,i} = \mathbf{w}_p^{\text{out}} \mathbf{h}_{p,i}^{(T+1)} + b_p^{\text{out}}.$$

where $\mathbf{w}_p^{\text{out}}$ and b_p^{out} are learnable parameters for nodes of order p. Then we follow [141] and scale the predictions with scaling parameters that are specific to the discrete feature $Z_{p,i}$ of the nodes

$$\hat{y}_{p,i} = s_{Z_{p,i}} \tilde{y}_{p,i} + r_{Z_{p,i}}.$$

where $\mathbf{s}$ and $\mathbf{r}$ are learnable embedding lookup tables that map $Z_{p,i}$ to the corresponding scaling factors and shifts. The goal of the scaling layer is to adapt the magnitude of the predictions to different unit systems of the target property.

2.5.4 Fusion Module

The fusion module computes a global molecule representation out of the final node embeddings and uses the global representation to weigh the prediction of different orders. We sum the final node embeddings $\mathbf{h}_{p,i}^{(T+1)}$ of each p-body to form an order specific representation and concatenate them to be an intermediate representation

$$\tilde{\mathbf{v}} = \sum_{i \in \mathcal{V}_1} \mathbf{h}_{1,i}^{(T+1)} \parallel \sum_{i \in \mathcal{V}_2} \mathbf{h}_{2,i}^{(T+1)} \parallel \cdots \parallel \sum_{i \in \mathcal{V}_N} \mathbf{h}_{N,i}^{(T+1)}.$$

Since node embeddings of different orders are computed by different parameters and the number of nodes of the orders also varies, the distributions of the order specific representations could be dramatically different from each other. In order to unify the distributions of the representations and to accelerate training, we apply batch normalization [70] followed by a fully connected layer on the intermediate representation to obtain the global representation

$$\mathbf{v} = \text{BatchNorm}\left(\tilde{\mathbf{v}}\right)$$
$$\mathbf{z} = \phi\left(\mathbf{W}\mathbf{v} + \mathbf{b}\right).$$

Then we pass the global representation through an attention layer to compute the weight α_p that measures the importance of the predictions of order p

$$\alpha_p = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{z}^T \mathbf{a}_p\right)\right)}{\sum_{q=1}^{N} \exp\left(\text{LeakyReLU}\left(\mathbf{z}^T \mathbf{a}_q\right)\right)}$$

where $\mathbf{a}$ are learnable vectors, and $\sum_p \alpha_p = 1$. We can understand the global representation as a query to the knowledge-base distilled in $\mathbf{a}$ for assigning contributions to predictions of different orders. This gives the model better flexibility and explainability in dealing with different molecules.

2.5.5 Final prediction

Inspired by the many-body expansion, we decompose the final prediction as a weighted sum of the prediction of different orders

$$\hat{y} = \alpha_1 \sum_{i \in \mathcal{V}_1} \hat{y}_{1,i} + \alpha_2 \sum_{i \in \mathcal{V}_2} \hat{y}_{2,i} + \alpha_3 \sum_{i \in \mathcal{V}_3} \hat{y}_{3,i} + \cdots$$

where the weights α_p are computed by the fusion module.

2.5.6 Model Training

Since all the modules in HMGNNs except for the fusion module are order specific, and the final prediction is a weighted average of the predictions per order, training HMGNNs by optimizing objective functions that only depend on the final prediction (the fused prediction) may cause gradient vanishing issues for parameters of some orders so that these parameters do not learn enough and lose their prediction utilities. To avoid this issue, we treat the computation of each order as a separate prediction task and propose a multi-task objective function that forces the prediction of all orders together with the final prediction to be close to the true target

$$\mathcal{L} = \frac{1}{N+1} \left(|\hat{y} - y| + \sum_{p=1}^{N} |\hat{y}_p - y| \right) + \lambda \left\| \Theta \right\|_2^2 \tag{2.6}$$

where $\hat{y}_p = \sum_{i \in \mathcal{V}_p} \hat{y}_{p,i}$ is the node order specific prediction, Θ denotes all trainable parameters of the model, $\lambda \geq 0$ is a hyper-parameter that controls the strength of L_2 normalization to prevent the model overfits. This objective function preserves gradient flow for parameters of each order and gives higher training importance to orders that the fussing module assigning larger weights to.

2.5.7 Complexity Analysis

The time and space complexity of HMGNN depends linearly on the number of nodes and edges in a HMG. The number of nodes determines the complexity of the input module and the output module while the number of edges determines the complexity of message passing.

Let $\mathcal{G}$ be a molecular graph with N atoms and $H_P(\mathcal{G})$ be its HMG that explicitly models up to P-bodies. We assume $\mathcal{G}$ is a complete graph for the worst case scenario. The number of nodes of order p in $H_P(\mathcal{G})$ is $\binom{N}{p}$. Let $i \in \mathcal{V}_p$ be a node of order p (i.e., a q-body), i is connected to nodes that are of order q where $q \in \{1, \cdots, P\}$. When $q < p$, the number of q order neighbors of node i is $\binom{p}{q}$ as i is connected to all q-bodies who are sub-graphs of i; when $q = p$, the number of order p neighbors of i is $p(N-p)$ since i is connected to p-bodies who share $p-1$ atoms with i; When $q > p$, the number of q-body neighbors of i is $\binom{N-p}{q-p}$. As a result, the complexity of message passing is

$$\sum_{p=1}^{P} \binom{N}{p} \left(\sum_{q=1}^{p-1} \binom{p}{q} + \sum_{q=p+1}^{P} \binom{N-p}{q-p} + p(N-p) \right)$$

and the complexity of the input/output module of HMGNN is $\sum_{p=1}^{P} \binom{N}{p}$.

In this dissertation, we experiment with HMGs and HMGNNs for up to 2-bodies, consequently, the time complexity and space complexity of our model are both $\mathcal{O}(N^3)$. Modern computing architectures such as graphics processing unit (GPU) and tensor processing unit (TPU) are optimized to accelerate this computation. Empirically, HMGNNs can generate property predictions for 10000 randomly drawn molecules from the QM9 dataset in 4 seconds.

2.6 Experiments

We conduct experiments to investigate three research problems in regards of many-body modeling and the HMGNN model

- How does HMGNN perform in the molecule property prediction tasks compared against the current state-of-the-art methods?

- How does many-body representation, interaction, and prediction contribute to the prediction?
- What is the utility of the components of HMGNN?

2.6.1 Implementation Details

We experiment with HMGs and HMGNNs for many-bodies up to order two. There are two types of nodes (1-bodies and 2-bodies), two types of edges with edge features (1-1 and 2-2 edges), and one type of edge without edge features (1-2 edges). Since 1-bodies are atoms, they only have discrete features. Each 2-body i is determined by its two end atoms and the distance between them $d_{2,i}$.

There are three types of geometries that we need to model, distance $d_{ij} \in (0, c)$ between 1-bodies i and j, length $l_{2,i} \in (0, c)$ of 2-bodies, and angle $\theta_{ij} \in [0, \pi]$ between 2-bodies i and j. We use a set of K radial basis functions (RBF) to convert the scalar geometries to real valued vector features. Let $x \in [a, b]$ be a scalar input and $\mathbf{x} \in \mathcal{R}^K$ be the real valued output of the RBFs, the k-th entry of $\mathbf{x}$ is computed as

$$x_k = \exp\left(-\beta_k \left(\exp\left(-x\right) - \mu_k\right)^2\right)$$

where μ_k and β_k specify the center and width of x_k. For distance d_{ij} between 1-bodies, we multiply its feature vector by a continuous monotonic decreasing function $\psi(d_{ij})$ that has $\psi(0) = 1$ and $\psi(c) = 0$. With this formulation, an 1-body node will have less influence to/from its distant order 1 neighbors. We follow [141] and set the value of μ_k to be equally spaced between $\exp\left(-a\right)$ and $\exp\left(-b\right)$ while $\beta_k = (2K^{-1}(\exp\left(-a\right) - \exp\left(-b\right)))^{-2}$. The goal of using RBFs is to decorrelate the scalar features to accelerate training [117]. We apply three different sets of RBFs to convert the distance d_{ij}, the length $l_{2,i}$, and the angle θ_{ij} to the corresponding features $\mathbf{e}_{1,ij}$, $\mathbf{x}_{2,i}$, and $\mathbf{e}_{2,ij}$, respectively.

We set the latent dimension to be 128 and use 5 interaction modules for our experiments. We use the shifted softplus function as the activation function. For ZPVE, U, U_0, H, G and C_v, the cutoff distances $c = 3$ while for other targets $c = 5$. We initialize the weights of fully connected layers with random orthogonal matrices scaled by the glorot initialization scheme [51] and the bias to zero. For learning the parameters

Table 2.1: Target properties in the QM9 dataset.

Target	Description
μ	Dipole moment
α	Isotropic polarizability
ϵ_{HOMO}	Energy of Highest occupied molecular orbital (HOMO)
ϵ_{LUMO}	Energy of Lowest occupied molecular orbital (LUMO)
$\Delta\epsilon$	Gap, difference between LUMO and HOMO
$\langle R^2 \rangle$	Electronic spatial extent
ZPVE	Zero point vibrational energy
U_0	Internal energy at 0 K
U	Internal energy at 298.15 K
H	Enthalpy at 298.15 K
G	Free energy at 298.15 K
C_v	Heat capacity at 298.15 K

of HMGNN, we run the AMSGrad algorithm [109] with a batch size of 32 for up to 3000000 steps and set the L_2 regularizer λ to be 1×10^{-6}. We initialize the learning rate to be 1×10^{-3} and multiply it with 0.1 every 2000000 gradient steps. The training algorithm stops if the MAE on the validation set does not decrease for 1000000 steps. We implement HMGNN using the Deep Graph Library (DGL) [172, 146].

2.6.2 Experimental Setting

We evaluate the performance of the proposed model on the QM9 dataset [114, 108]. QM9 is a widely used benchmark for evaluating models that predict molecule properties. It consists of around 130K equilibrium molecules associated with 12 geometric, energetic, electronic, and thermodynamic properties. The properties are described in Table 2.1. These molecules contain up to nine heavy atoms (C, O, N, and F). We randomly select 110000 molecules for training, 10000 molecules for validation, and 10831 molecules as the test set. We conduct model selection for different targets on the validation set and

Table 2.2: Mean absolute error on QM9 with 110K training molecules. In each row, we use boldface for the best performance method. Column HMGNN-1 and HMGNN-2 correspond to the performance of summing over predictions of 1-bodies and 2-bodies, respectively.

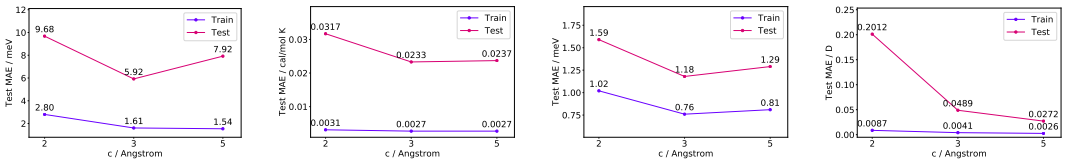
Target	Unit	enn-s2s	SchNet	NMP-edge	Cormorant	PhysNet	DimeNet	HMGNN-1	HMGNN-2	HMGNN
μ	D	0.030	0.033	0.029	0.038	0.0529	0.0286	0.0276	0.0283	**0.0272**
α	a_0^3	0.092	0.235	0.077	0.085	0.0615	**0.0469**	0.0571	0.0647	0.0561
ϵ_{HOMO}	meV	43	41	36.7	34	32.9	27.8	24.94	26.31	**24.78**
ϵ_{LUMO}	meV	37	34	30.8	38	27.4	**19.7**	20.72	21.42	20.61
$\Delta\epsilon$	meV	69	63	58.0	61	42.5	34.8	33.44	35.02	**33.31**
$\langle R^2 \rangle$	a_0^2	0.180	0.073	**0.072**	0.961	0.765	0.331	0.43	0.6	0.416
ZPVE	meV	1.5	1.7	1.49	2.03	1.39	1.29	1.24	1.34	**1.18**
U_0	meV	19	14	10.5	22	8.15	8.02	6.19	9.06	**5.92**
U	meV	19	19	10.6	21	8.34	7.89	7.22	11	**6.85**
H	meV	17	14	11.3	21	8.42	8.11	6.35	8.37	**6.08**
G	meV	19	14	12.2	20	9.40	8.98	7.95	11.06	**7.61**
c_v	$\frac{\text{cal}}{\text{mol K}}$	0.040	0.033	0.032	0.026	0.0280	0.0249	0.0241	0.025	**0.0233**

report the mean absolute error (MAE) of the best performing models. For properties with atomic reference values (U_0, U, H, G, C_v), we subtract the original value by the per-atom-type reference values to be the target. Since $\Delta\epsilon$ is defined as the gap between ϵ_{LUMO} and ϵ_{HOMO}, we predict it as $\Delta\epsilon = \epsilon_{\text{LUMO}} - \epsilon_{\text{HOMO}}$. In our experiments, we convert the units of ϵ_{HOMO}, ϵ_{LUMO}, $\Delta\epsilon$, ZPVE, U_0, U, H, G to eV.

We compare the performance of HMGNN with six state-of-the-art methods, enn-s2s [48], SchNet [117], neural message passing with edge updates (NMP-edge) [71], Cormorant [3], PhysNet [141], and directional message passing neural network (DimeNet) [78]. Results of enn-s2s, SchNet, NMP-edge, Cormorant, and DimeNet are from the corresponding papers. We take the results of PhysNet from [78].

2.6.3 Prediction Performance

We show the prediction performance of HMGNN and the competing methods on the 12 properties of QM9 in Table 2.2. Our proposed method sets the new state-of-the-art on 9 out of the 12 target properties. HMGNN's performance aligns with the best



(a) U_0. (b) C_v. (c) ZPVE. (d) μ.

Figure 2.3: Effect of the cutoff distance c on prediction performance on four target properties.

results on the remaining targets with an exception of $\langle R^2 \rangle$. We also present the performance of summing over predictions over 1-bodies (HMGNN-1) and 2-bodies (HMGNN-2), respectively. Although the performance of HMGNN-2 is consistently worse than HMGNN-1, their weighted combination outperforms any of the standalone prediction. This demonstrates the effectiveness of the fusion module driven by the global molecule representations and the attention mechanism, and that explicitly modeling and computing predictions of many-bodies can be beneficial for chemical prediction tasks.

We analyze the effect of a critical hyper-parameter, the cutoff distance c, on prediction performances of four types of properties. We choose U_0 to represent properties related to atomization energies (U_0, U, H, G), C_v to represent thermodynamic properties (C_v), ZPVE to represent properties related to fundamental vibrations of the molecule (ZPVE), and μ to represent electronic properties (μ, α, ϵ_{HOMO}, ϵ_{LUMO}, $\Delta\epsilon$, $\langle R^2 \rangle$) [48]. We present the training and test mean absolute error (MAE) of HMGNNs on HMGs constructed with $c \in \{2, 3, 5\}$ in Figure 2.3.

When constructing molecular graphs as well as HMGs, the larger the cutoff distance we choose, the less geometric information about the molecules that we lose. However, a large cutoff value does not always lead to better performance. In Figure 2.3, despite the training error decreases across all the four targets as the cutoff value increases, the test error shows an increasing trend for three properties. This is a signal that the model over-fits the training set on the three properties. This is because of the large model capacity of HMGNNs as they have one set of parameters for each many-body order. An HMGNN of order N possesses N times the number of parameters of a normal GNN-based model.

Table 2.3: Ablation study of various components of HMGNN on U_0 and C_v.

Target	Architecture	HMGNN-1	HMGNN-2	HMGNN
U_0	Default	6.19	9.06	5.92
	Remove MTL	8.22	9716.95	8.22
	Remove IOMP	10.26	8.18	7.88
	Remove HO	10.08	-	-
C_v	Default	0.0241	0.0250	0.0233
	Remove MTL	0.0247	1.4022	0.0247
	Remove IOMP	0.0297	0.0275	0.0244
	Remove HO	0.0289	-	-

2.6.4 Ablation Study

In this section, we conduct ablation study on two targets (i.e., U_0, C_v) to demonstrate the importance of the multi-task learning loss, inter-order message passing, and explicit modeling of high-order bodies in improving the performance of molecular property prediction. We propose three variants of the HMGNN model and show their results in Table 2.3.

Remove MTL (Multi-Task Learning loss)

This variant has the same specification with the default model. It differs with the default model in that it is trained by minimizing the naive loss $|\hat{y} - y|$ instead of the multi-task learning loss that we proposed in Eq-2.6. As shown in Table 2.3, the 2-bodies of this variant lose their prediction power while the fusion module gives all attention weights to the 1-bodies, and as a result, the performance of this variant is worse than the default HMGNN. Furthermore, the prediction of 1-bodies (i.e., HMGNN-1) is also less accurate than the default model.

Remove IOMP (Inter-Order Message Passing)

This variant removes edges/messages between 1-bodies and 2-bodies, as a result, information of the two orders are not shared. We can see that the performance of HMGNN-1 and HMGNN drops in the prediction of both U_0 and C_v. This demonstrates the importance of inter-order message passing. However, the prediction accuracy of HMGNN-2 on U_0 is better than models with inter-order message passing. This might because 2-bodies (both distance and angle) contain more geometric information than 1-bodies (only distance).

Remove HO (High-Order modeling)

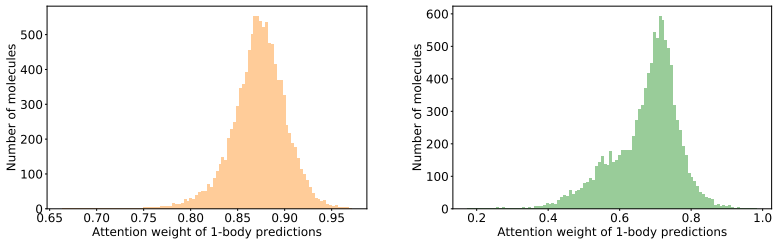
This variant removes high-order related modeling (2-body interaction, representation, and prediction) and is similar to existing GNN-based prediction methods (i.e., PhysNet). As shown in Table 2.3, this method performs worse than HMGNN-1 of the variant that removes multi-task learning loss. This shows another evidence of the effectiveness of inter-order message passing.

2.6.5 Visualization of Attention weights

In Figure 2.4, we show the attention scores of the 1-body predictions generated by the fusion module for predicting U_0, C_v, μ, and ZPVE on the test set. Since we only experiment with many-bodies up to the second order, the attention weights of the 2-bodies is one minus that of the 1-bodies. On the four types of chemical properties, 1-body contribution dominates the prediction of most of the molecules. However, 2-body predictions also take a considerable amount of attention.

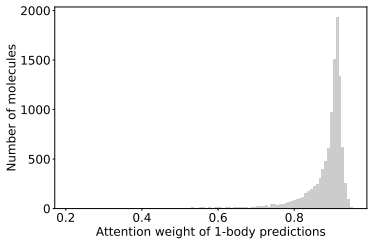
2.7 Conclusion

We propose a novel heterogeneous graph based molecule representation, heterogeneous molecular graph (HMG), to model many-body representations and interactions. Inspired by the many-body expansion of energy surfaces, we design a heterogeneous molecular graph neural network (HMGNN) to leverage the rich information stored in HMGs

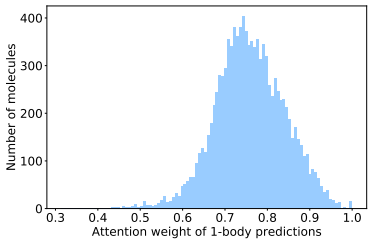


(a) U_0.

(b) C_v.



(c) ZPVE.



(d) μ.

Figure 2.4: Attention weights generate by the fusion module for predicting the four properties.

for molecular prediction tasks. HMGNN follows a message passing paradigm and leverages global molecule representations using an attention mechanism. We propose to train HMGNNs by optimizing a multi-task learning loss. HMGNN achieves state-of-the-art performance on 9 out of 12 properties on the QM9 dataset. Experiments also show that the multi-task learning loss improves the generalization of the model. In this dissertation, we only model many-bodies up to the second order, future works should aim to model many-bodies of higher than third orders and also to enable HMGNNs for another important chemical prediction tasks, molecular dynamics simulations.

Chapter 3

Text Retrieval-Augmented Molecular Property Prediction

3.1 Introduction

In the field of drug discovery and material design, the accurate prediction of molecular properties is crucial. Machine learning methods, particularly those based on neural networks, have demonstrated effectiveness in these tasks. These approaches learn to predict the properties of molecules based on their structures, which are usually represented using Simplified Molecular Line Entry System (SMILES) strings or molecular graphs.

Human knowledge of chemistry, including descriptions of chemical reactions and processes, the properties and behavior of chemical compounds, and records of chemical and clinical testing on molecules, is usually documented in the form of natural language. Such knowledge is a very important source of information for predicting molecular properties. Recently have witnessed a surge in computational chemistry that uses multimodal learning approaches to jointly learn from molecular structures and textual data. These models acquire both structural information and textual expert annotation of molecules and show superior MPP performance than models trained on structural data alone (e.g., graph neural networks).

Existing multimodal MPP models can be categorized into two classes, contrastive

models [132, 90] and predictive models [167]. Contrastive model have a molecule structure encoder and a text encoder to project both modalities into a shared low-dimensional space, respectively. These models are trained via contrastive objectives such that the molecules and their corresponding textual descriptions are closer in the shared representation space. Predictive models use one text encoder to encode both SMILES strings and documents. Such models are (pre)trained via language modeling objectives such as masked language modeling [31] on scientific literature in which compounds are replaced by their corresponding SMILES strings. In these models, textual knowledge is encoded implicitly in their parameters. It is hard to determine what knowledge is stored and used for the predictions.

In this dissertation, we explore an alternative multimodal learning paradigm, retrieval augmentation, to quickly incorporates external literature for MPP tasks. Retrieval-augmented model is a class of models that combines a neural network with a retrieval system. The retrieval system retrieves relevant information from an external knowledge source, which is then used to inform the model's predictions. Retrieval-augmented models do not store knowledge in their parameters, but learn to explicitly decide what knowledge to use and how to use it for predictions. Such models have shown great success in knowledge-intensive tasks such as open-domain question answering [55, 83, 75] but remain under-explored in the space of chemistry.

We propose a two-stage retrieval system to efficiently and effectively learn to retrieve informative documents from a knowledge corpus to augment molecular property prediction. The two-stage retrieval system consists of two components, a molecular text-retriever, and a learnable scoring function. The molecular text-retriever uses the International Union of Pure and Applied Chemistry (IUPAC) names [37] of the molecules as queries and the BM25 algorithm [111] to reduce the size of the knowledge corpus by retrieving few hundred molecule-related documents. The scoring function $\alpha(x, z)$ is then trained to find the top k documents to be used towards the final prediction. The two-stage retrieval system is a generic method that can be used with any structure-based MPP models such as graph neural networks [65, 124] and pre-trained language models [15, 167]. We term our method RTMol.

The contributions of this work are two-fold: 1) We propose RTMol, an effective and efficient framework, that retrieves documents from a knowledge corpus for molecular

property predictions. To the best of our knowledge, RTMol is the first text-retrieval method for MPP tasks; 2) We evaluate RTMol by augmenting it with four widely used structural-based MPP models and two knowledge corpora. We conduct experiments on eight classification benchmarks and four regression benchmarks and compare against existing multimodal learning MPP models that also use textual knowledge for molecular predictions. Experimental results show that incorporating textual knowledge using RTMol improves the performance of the baseline models and provides explainable predictions. Moreover, after training on one corpus, RTMol can easily adapt to a different corpus without retraining.

3.2 Related Works

3.2.1 Molecular Property Prediction

Most of the machine learning models for predicting molecule properties fall into two categories, descriptor-based models and neural network-based models, which differ in their way of representing molecules. Descriptor-based models use physics-motivated algorithms to convert molecules into descriptors (i.e., low-dimensional vectors) and apply machine learning models on the descriptors to predict molecular properties [36, 8]. Neural network-based models jointly train a representation model and a predictive model in an end-to-end fashion. Graph neural networks [49, 118, 78, 124] and equivariant neural networks [119, 45] are popular choices for representation learning because of their ability to encode geometries and symmetries. Researchers also use language models [85, 167] to learn molecule representations directly from SMILES strings.

Recently, self-supervised learning (SSL) has attracted a lot of research attention in molecule representation learning [166, 65, 112]. SSL methods pre-train the representation models on a large bulk of unlabeled molecules by optimizing physics-informed objectives. The pre-trained models are then finetuned on labeled molecules for different downstream tasks.

3.2.2 Multimodal Molecular Learning

Molecules have different representations that contain different types of information. Researchers have been exploring multi-modal learning that connects different modalities to benefit molecule representation learning. Contrastive learning methods have been applied to bridge SMILES with IUPAC [53], SMILES to molecular graphs [175], 2D molecular graphs to 3D geometries [89, 130], and molecular graphs to textual annotations [33, 132, 90]. Researchers also developed generative methods that learn to generate one modality from another [89, 176]. Large language models (LLMs), a.k.a., foundation models, are trained on large corpora of literature to comprehensively inject chemistry knowledge into the models' parameters for various scientific tasks [167, 34, 138, 2].

3.2.3 Retrieval-Augmented Models

Retrieval-augmented models are gaining success in natural language processing tasks such as language modeling [17, 75] and question answering [83, 55]. These models learn to retrieve relevant documents from external knowledge corpora to assist prediction so as to reduce the number of parameters needed to store the knowledge. Cross-modal methods are developed to retrieve text for image inputs [94, 164]. In chemistry, RetMol [148] learns to retrieve from exemplar molecules for controllable molecule generation.

3.3 Methodology

3.3.1 Problem Setup

The goal of molecular property prediction (MPP) is to learn a model $\mathcal{M}(x) : \mathcal{X} \mapsto \mathcal{Y}$ that takes a molecule $x \in \mathcal{X}$ as input and outputs the molecule's property $y \in \mathcal{Y}$ where $\mathcal{X}$ denotes the universe of molecules and $\mathcal{Y}$ denotes the range of the desired property. Since a molecule's structure is more determinant to its properties, x usually refers to the molecule's SMILES string or molecular graph.

3.3.2 Text Retrieval-Augmented Molecular Property Prediction

We propose Text Retrieval-Augmented Molecular Property Prediction RTMol that leverages textual chemistry knowledge in literature corpora for MPP. In concrete, RTMol trains a model that takes a molecule's structure and a text corpus $\mathcal{C}$ as input to predict the desired property, i.e.,

$$\hat{y} = \mathcal{M}(x, \mathcal{C}) = \sum_{z \in \mathcal{C}} \alpha(x, z) \mathcal{M}'(x, z) \tag{3.1}$$

where z is a document from the corpus, $\alpha(x, z) \geq 0$ is a learnable scoring function that weighs documents by their contribution towards the prediction, and $\mathcal{M}'(x, z)$ is a text-augmented predictor that jointly make use of the molecular structure and the document to make predictions.

Training a model with the formulation of Eq. 3.1 could be computationally prohibitive as it requires conducting forward and backward propagation over the corpus with more than millions of documents. A common solution to this challenge is to approximate Eq. 3.1 by considering only the top k documents with the highest scores under $\alpha(x, z)$ [55]. The process of scoring documents and fetching the top k documents is retrieval. However, this solution causes an exploration dilemma. In each gradient step, parameters of $\mathcal{M}$ are only updated according to the top k input documents. This causes the training process to be inefficient in finding the informative documents. For example, suppose the corpus has one million documents, and $k = 100$, the model needs to be trained for at least ten thousand gradient steps for one molecule (ten thousand training epochs for the dataset) to explore the corpus once.

In the field of information retrieval and recommendation systems, the utilization of two-stage systems has become a prevalent approach for achieving a balance between efficiency and ranking accuracy [91, 25]. The method consists of two stages: in the first stage, coarse but efficient rule-based methods are employed to significantly reduce the size of the corpus by identifying a subset of several hundred relevant items that are most likely to be relevant to the query. In the second stage, a more fine-grained model is applied to rank these items and present the most relevant ones to the user.

Inspired by the two-stage systems, we propose a two-stage document retrieval system to resolve the exploration dilemma. For each molecule, we first use a molecular text-retriever $\mathcal{R}(x, \mathcal{C})$ to find $K \ll |\mathcal{C}|$ relevant documents, then train the scoring function

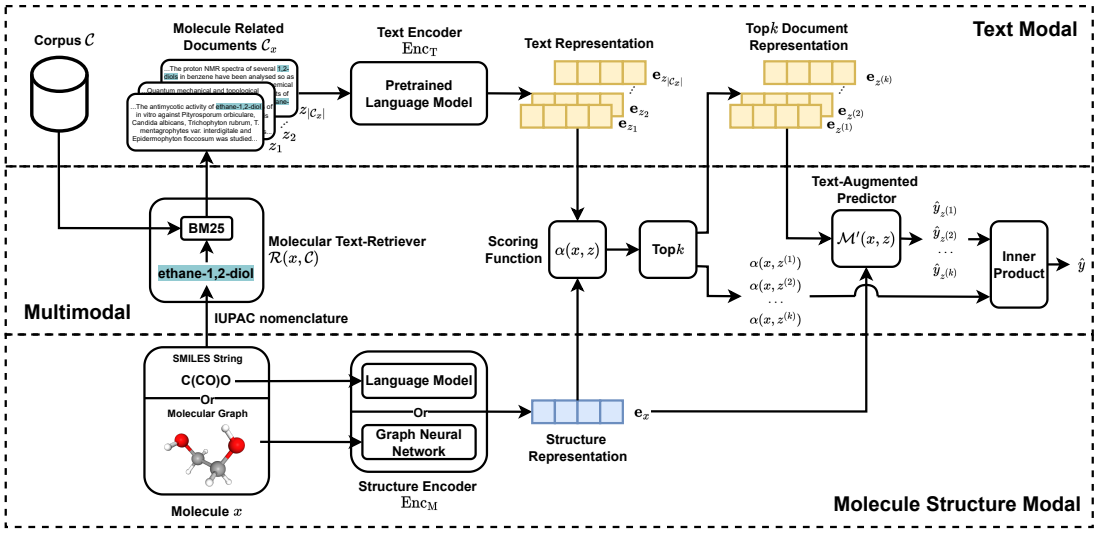


Figure 3.1: Overall architecture of RTMol.

$\alpha(x, z)$ to find the top $k < K$ documents to be used towards the final prediction. The final prediction function of RTMol is,

$$\mathcal{M}(x, \mathcal{C}_x) = \sum_{i=1}^{k} \alpha(x, z^{(i)}) \mathcal{M}'(x, z^{(i)}) \tag{3.2}$$

where $\mathcal{C}_x = \mathcal{R}(x, \mathcal{C})$ is the documents retrieved by the molecular text-retriever, and $z^{(i)}$ is the document with the i-the highest α score.

In the following sections, we give a detailed description of the molecular text-retriever $\mathcal{R}(x, \mathcal{C})$, RTMol's architecture, the scoring function $\alpha(x, z)$, and the text-augmented predictor $\mathcal{M}'(x, z)$. Lastly, we describe our strategies of training RTMol. The overall architecture of RTMol is shown in Fig. 3.1.

3.3.3 Molecular Text-Retriever

Text retrieval for molecules requires ranking documents based on their relevance scores to the molecules. Since molecules are usually represented in their structural forms such as SMILES strings or molecular graphs, there exists a modality gap between molecule structures and natural language documents. It is challenging to define relevance between molecules and documents.

In this dissertation, we propose to use the International Union of Pure and Applied Chemistry (IUPAC) names of molecules as queries and a sparse text retrieval algorithm, BM25 [111], to retrieve from the knowledge corpus. IUPAC nomenclature serves the purpose of systematically naming organic [37] or inorganic [26] molecules by basic words that indicate the structure of the molecule and prioritize functional groups to facilitate communication. An example of IUPAC nomenclature can be found in Fig. 3.1. The BM25 algorithm is a bag-of-words type text retrieval algorithm that assigns a relevance score to each document based on its relevance to a given query (IUPAC name). In general, the BM25 algorithm would assign a higher score to a document that contains a higher frequency of the query terms (functional groups).

There are three advantages to the molecular text-retriever. Firstly, IUPAC nomenclature is a standardized method for representing molecules, and every molecule has a corresponding IUPAC name. This allows for a universal method of querying for documents related to a specific molecule. Secondly, in literature, molecules are commonly referred to by their IUPAC names, making it a useful method for identifying documents that pertain to a specific molecule. Thirdly, IUPAC nomenclature decomposes a molecule into its functional groups, which play a crucial role in determining the molecule's reactivity and properties. These functional groups serve as query terms in the retrieval process, allowing for the identification of not only documents pertaining to the specific molecule, but also those related to molecules with similar functional group compositions.

3.3.4 Model Architecture

RTMol has two encoders to encode molecule structures and documents to their low-dimensional representations, respectively. We denote the structure representation as $\mathbf{e}_x = \text{Enc}_\text{M}(x)$ and the document representation as $\mathbf{e}_z = \text{Enc}_\text{T}(z)$. The structure encoder $\text{Enc}_\text{M}(x)$ could be any encoder that is used for structure alone MPP. For example, graph neural networks (GNNs) are usually used to encode molecular graphs, (pre-trained) transformers are used when the inputs are SMILES strings. The text encoder $\text{Enc}_\text{T}(z)$ is a pre-trained language model that converts documents to contextualized embeddings.

We pass the structure and the text representations, $\mathbf{e}_x$ and $\mathbf{e}_z$, into the scoring function $\alpha(x, z)$ to compute the scores that weigh the contribution of documents towards

the prediction,

$$\alpha(x,z) = \begin{cases} \frac{\exp f(\mathbf{e}_x, \mathbf{e}_z)}{\sum_{z_j \in \mathcal{C}_x} \exp f(\mathbf{e}_x, \mathbf{e}_{z_j})}, & z \in \mathcal{C}_x \\ 0, & z \notin \mathcal{C}_x \end{cases}$$

where $f(\mathbf{e}_x, \mathbf{e}_z) = \text{MLP}_m(\mathbf{e}_x)^T \text{MLP}_t(\mathbf{e}_z)$ is a learnable inner product. We use the α scores to choose the top k informative documents and weigh their predictions computed by the text-augmented predictor.

The goal of the text-augmented predictor $\mathcal{M}'(x,z)$ is to augment structure alone predictors with the retrieved textual information. Given that the structure of a molecule plays a crucial role in determining its properties, it would be expected that the structure would be the primary determinant of molecular property predictions. The retrieved documents may or may not be needed to predict the properties. As demonstrated in [60], learning a zero function ($f(x) = 0$) is easier than learning an identity function ($f(x) = x$). We define the text-augmented predictor $\mathcal{M}'(x,z)$ as

$$\mathcal{M}'(x,z) = \text{MLP}(\mathbf{e}_x + \text{MLP}(\mathbf{e}_z)).$$

With this formulation, the text-augmented predictor is endowed the ability to selectively leveraging the retrieved documents for the final prediction.

3.3.5 Training

As shown in Eq. 3.2, the final prediction $\hat{y}$ of RTMol is essentially a average of the text-augmented predictions $\mathcal{M}'(x,z)$. Suppose RTMol is trained by optimizing an objective that only enforces the final prediction $\hat{y}$ to approximate its ground truth label y. In a gradient backpropagation step, the scoring function $\alpha(x,z)$ is updated such that a document z gets a higher score if it performs better than expected [55]. The parameters of the text-augmented predictor $\mathcal{M}'(x,z)$ and the text encoder $\text{Enc}_\text{T}(z)$ are updated according to documents with higher weights (i.e., $\alpha(x,z)$ scores) so that their predictions are more accurate. This may cause a "rich gets richer" issue that prevents RTMol from exploring the corpus for more relevant documents.

We propose to optimize a multitask objective to encourage exploration. Besides the final prediction $\hat{y}$, the multitask objective also enforces each text-augmented prediction

to be accurate, i.e.,

$$\mathcal{L} = l(y, \hat{y}) + \frac{\lambda}{k} \sum_{i=1}^{k} l(y, \hat{y}_{z^{(i)}})$$

where $\lambda \geq 0$ is a hyper-parameter that controls exploration vs. exploitation, $\hat{y}_{z^{(i)}} = \mathcal{M}'(x, z^{(i)})$ is the prediction augmented by the document with the i-th highest α scores.

3.4 Experiments

3.4.1 Datasets

MPP Benchmarks. We evaluate the MPP performance of RTMol on a diverse set of eight classification and four regression datasets. These datasets cover a wide range of properties, including pharmacology, physical chemistry, and biophysics of molecules. Most of the datasets are taken from the MoleculeNet benchmark [153] with the exception of two regression datasets, Malaria [41] and CEP [56]. Detailed descriptions of the datasets are in Tab. **??**. Following [65], we split the datasets into train/validation/test (0.8/0.1/0.1) sets according to the scaffold of the molecules to evaluate the models' generalizability. IUPAC names of the molecules are obtained from PubChem[1] . For molecules without PubChem records, we use STOUT [106] to generate the IUPAC names from their canonical SMILES strings.

Knowledge Corpora. In our experiments, we utilize two knowledge corpora that contain extensive information on chemistry and molecules, namely S2ORC and PubMed Abstract (PMA). S2ORC[2] [93] is a corpus of scientific papers spanning a wide range of academic disciplines. We use a filtered version of S2ORC [132] with 17M paragraphs focusing on Medicine, Biology, Chemistry, and Computer Science. PMA [42] consists of 15.5M abstracts from publications in PubMed[3] , a database of biomedical literature.

3.4.2 Baselines and Competing Methods

In our evaluation of RTMol, we consider two classes of structure-based MPP models that reflect the majority of MPP methods currently in use, graph neural networks (GNNs)

[1] https://pubchem.ncbi.nlm.nih.gov/
[2] https://github.com/allenai/s2orc
[3] https://pubmed.ncbi.nlm.nih.gov/

and language models (LMs). For GNNs, we use graph isomorphism network (GIN) [159] as the backbone network with three different parameter initialization strategies, random initialization, pre-trained via node attribution masking (GIN-AM) [65], and pre-trained via graph contrastive learning (GIN-CL) [166]. For LMs, we use SciBERT [15], a BERT-like language model pre-trained on a large corpus of scientific literature. We augment these models with RTMol to investigate its effectiveness in improving the performance of structure-based MPP models.

We compare the performance of RTMol with two state-of-the-art multimodal MPP models, MoMu [132] and KVPLM [167]. MoMu is a model that uses a graph encoder to encode molecular graphs and a text encoder to encode documents. It is pre-trained via contrastive learning on a dataset that pairs molecules with documents from the S2ORC corpus. The authors of MoMu have released two sets of pre-trained parameters, MoMu-S and MoMu-K, in which the parameters of the text encoders are initialized with SciBERT and KVPLM, respectively. The graph encoder of MoMu is a GIN whose parameters are initialized via graph contrastive learning. KVPLM is a language model that encodes molecules by their SMILES strings. Its parameters are initialized using SciBERT and it is pre-trained on scientific literature in which molecules are replaced by their corresponding SMILES strings. Note that, the scientific literature also come from the S2ORC corpus. In our experiments, we finetune the GIN encoders of MoMu-S and MoMu-K and the text encoder of KVPLM on the MPP benchmarks.

3.4.3 Experimental Settings

In the experiments, we follow the setup in [65, 166, 132] to train GIN-based models for 100 epochs with a learning rate of 1e-3 using the Adam optimizer. We follow the setup in [167] to finetune SciBERT and KVPLM for 20 epochs with a learning rate of 5e-6 using the BertAdam optimizer. The batch size is set to 128 for GIN-based models and 64 for language models. We report the test performance of the epoch with the best validation performance. For language models with RTMol, we first freeze the parameters of the language model and only conduct gradient descent on the scoring function and the predictor for 20 epochs with a learning rate of 1e-4 to warm up their parameters. We set the text encoder of all RTMol models to be SciBERT. For efficiency purposes, we freeze its parameters throughout the whole training process. We conduct

Baseline	Text Method	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	Avg.
GIN	-	67.41 (2.84)	73.73 (0.96)	62.81 (0.65)	57.82 (0.81)	**63.43** (3.83)	72.42 (1.02)	74.15 (1.38)	69.84 (2.67)	67.70
	RTMol	**68.04** (2.53)	**75.53** (0.61)	**63.99** (0.35)	**61.20** (1.43)	61.92 (2.41)	**73.39** (0.18)	**77.51** (0.33)	**71.20** (3.85)	**69.10**
GIN-AM	-	65.16 (1.16)	**75.74** (0.59)	63.28 (0.28)	59.61 (0.81)	64.24 (2.32)	76.67 (1.44)	74.96 (0.45)	**80.15** (0.38)	69.98
	RTMol	**70.52** (0.20)	75.63 (0.32)	**64.05** (0.73)	**63.44** (0.16)	**75.73** (4.17)	**78.66** (0.91)	**76.16** (0.23)	75.02 (8.14)	**72.40**
GIN-CL	-	68.79 (0.63)	**74.93** (0.61)	62.53 (0.37)	60.83 (0.29)	55.03 (7.70)	**74.19** (0.90)	74.58 (0.46)	75.78 (0.94)	68.33
	MoMu-S	69.55 (0.51)	74.00 (0.23)	62.76 (0.27)	61.96 (0.25)	62.65 (9.61)	73.18 (1.15)	75.75 (0.28)	**76.42** (0.89)	69.53
	MoMu-K	69.32 (0.54)	73.94 (0.77)	62.62 (0.79)	60.93 (0.84)	57.65 (2.18)	72.81 (2.14)	74.25 (2.00)	76.14 (0.86)	68.46
	RTMol	**70.01** (0.10)	**74.90** (0.42)	**64.75** (0.22)	**63.23** (1.27)	**64.61** (5.82)	74.09 (1.06)	**77.16** (1.57)	74.81 (0.40)	**70.45**
SciBERT	-	68.58 (2.25)	73.42 (0.65)	57.62 (2.31)	59.02 (1.55)	**90.58** (1.49)	46.31 (2.81)	74.82 (0.78)	**78.83** (3.07)	68.65
	KVPLM	68.09 (1.56)	72.82 (1.33)	59.42 (0.16)	57.53 (1.47)	85.39 (5.60)	52.55 (3.68)	**77.34** (0.85)	73.92 (2.53)	68.38
	RTMol	**71.39** (0.71)	**75.30** (0.11)	**62.44** (0.72)	**61.19** (0.69)	90.34 (0.69)	**71.81** (1.98)	74.29 (0.34)	72.71 (4.72)	**72.43**

Table 3.1: Mean (and standard deviation) ROC-AUC (higher is better) of RTMol and baseline methods on classification benchmarks. The best performance in each group is shown in bold. Results in each group use the same backbone neural network with the same initialization.

hyper-parameter searches for RTMol for the value of K and k from [100, 500] and [5, 10], respectively. We run each experiment with three random seeds and report the mean and standard deviation of the metrics.

3.4.4 Molecular Property Prediction Performance

We show the MPP performance of RTMol on classification and regression tasks in Tab. 4.1 and 3.2, respectively. We aggregate the results into four groups. Results in each group use the same backbone neural network with the same initialization. In these tables, RTMol are trained and evaluated on the S2ORC corpus. According to the tables, the GIN-based baselines' performance is improved in at least 9 out of the 12 benchmarks by augmenting with RTMol to leverage chemistry literature. RTMol improves the LM-based baseline in 7 out of the 12 benchmarks and aligns with the best performance in the rest. In general, multimodal models augmented with textual information outperform structure-only models. Moreover, RTMol outperforms the state-of-the-art multimodal MPP models, MoMu and KVPLM, in most of the benchmarks.

Baseline	Text Method	Malaria	CEP	ESOL	Lipophilicity	Avg.
GIN	-	1.11 (0.01)	1.31 (0.00)	1.50 (0.04)	0.81 (0.01)	1.18
	RTMol	**1.10** (0.00)	**1.04** (0.01)	**1.44** (0.02)	**0.80** (0.00)	**1.09**
GIN-AM	-	1.14 (0.02)	1.37 (0.02)	1.43 (0.01)	0.83 (0.01)	1.19
	RTMol	**1.09** (0.01)	**1.07** (0.01)	**1.29** (0.02)	**0.82** (0.02)	**1.07**
GIN-CL	-	1.12 (0.00)	1.39 (0.01)	**1.25** (0.01)	0.79 (0.02)	1.14
	MoMu-S	1.11 (0.00)	1.43 (0.02)	1.34 (0.02)	0.79 (0.01)	1.16
	MoMu-K	1.13 (0.01)	1.41 (0.01)	1.33 (0.02)	0.79 (0.00)	1.17
	RTMol	**1.08** (0.01)	**1.04** (0.01)	1.28 (0.06)	**0.76** (0.02)	**1.04**
SciBERT	-	**1.12** (0.00)	1.34 (0.01)	**0.90** (0.03)	0.90 (0.01)	1.06
	KVPLM	1.13 (0.01)	**1.28** (0.02)	0.91 (0.29)	**0.86** (0.01)	**1.05**
	RTMol	**1.12** (0.02)	**1.28** (0.03)	0.98 (0.03)	0.90 (0.01)	1.07

Table 3.2: Mean (and standard deviation) RMSE (lower is better) of RTMol and baseline methods on regression benchmarks. The best performance in each group is shown in bold. Results in each group use the same backbone neural network with the same initialization.

Baseline	Test Corpus	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE
GIN	S2ORC	68.04 (2.53)	75.53 (0.61)	63.99 (0.35)	61.20 (1.43)	61.92 (2.41)	73.39 (0.18)	77.51 (0.33)	71.20 (3.85)
	PMA	67.88 (2.30)	74.96 (1.04)	63.48 (0.39)	60.14 (1.35)	62.21 (3.83)	73.16 (0.05)	77.52 (0.83)	71.40 (4.05)
GIN-AM	S2ORC	70.52 (0.20)	75.63 (0.32)	64.05 (0.73)	63.44 (0.16)	75.73 (4.17)	78.66 (0.91)	76.16 (0.23)	75.02 (8.14)
	PMA	70.70 (0.89)	74.79 (0.55)	62.52 (0.50)	60.67 (1.20)	75.91 (6.92)	78.64 (0.79)	76.17 (0.34)	75.20 (7.85)
GIN-CL	S2ORC	70.01 (0.10)	74.90 (0.42)	64.75 (0.22)	63.23 (1.27)	64.61 (5.82)	74.09 (1.06)	77.16 (1.57)	74.81 (0.40)
	PMA	70.06 (0.33)	74.73 (0.19)	63.81 (0.63)	62.52 (1.96)	65.51 (7.60)	74.17 (1.00)	77.06 (1.68)	74.60 (0.51)
SciBERT	S2ORC	71.39 (0.71)	75.30 (0.11)	62.44 (0.72)	61.19 (0.69)	90.34 (0.69)	71.81 (1.98)	74.29 (0.34)	72.71 (4.72)
	PMA	70.88 (1.14)	75.26 (0.20)	61.65 (0.74)	61.10 (0.61)	90.64 (0.81)	72.00 (1.57)	74.05 (0.77)	72.89 (4.03)

Table 3.3: Cross corpus performance of RTMol on classification benchmarks.

3.4.5 Cross Corpus Generalization

We demonstrate the cross corpus generalization capability of RTMol in Tab. 3.3. In this experiment, we train RTMol on the S2ORC corpus and evaluate it with the PMA corpus without any retraining. In the tables, we observe that RTMol performs equally well on both the training corpus S2ORC and the PMA corpus that it did not see before. This indicates RTMol generalizes well across corpora.

3.4.6 Case Study

We investigate what documents are retrieved by RTMol and how they are helpful in accurately predicting the target properties via a case study. We select three molecules that are predicted wrong by random initialized GINs but are correctly predicted by RTMol and show their top1 retrieved documents in Fig. 3.2. We observe that all the retrieved documents contain two chunks of information. First the IUPAC name of the target molecules or molecules that share functional groups with the target molecules. This demonstrates the correctness of the sparse retrieval step. Second, information that is implicitly or explicitly related to the target property thanks to the learnable scoring function. As a conclusion, RTMol can find documents that are informative to the prediction tasks and use them to improve the predictions. More importantly, RTMol can provide interpretability to its predictions.

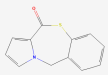
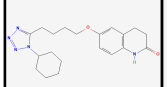
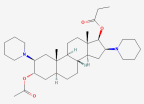
Molecule	IUPAC Name	Target Property	Top1 Document
	11H-pyrrolo[2,1-c] [1,4]benzothiazepin-6-one	Ability to inhibit HIV replication	...pyrrolo [2,1-c][1,4] benzodiazepine (PBD) were studied...long terminal repeat (LTR) of the human immunodeficiency type 1 virus (HIV-1) were analysed....PCR) were performed on HIV-1 LTR gene sequences....compared to antiproliferative effects of the natural product distamycin A 1 and pyrrolo [2,1-c][1,4] benzodiazepine (PBD 6)..... With respect to inhibition of HIV-1 LTR driven transcription, it was found that the hybrid 5 containing the four pyrroles distamycin analogs, is more active than 2, 3 and 4....
	6-[4-(1-cyclohexyltetrazol-5-yl)butoxy]-3,4-dihydro-1H-quinolin-2-one	FDA approval status	..cilostazol (6-[4-(1-cyclohexyl-1 H-tetrazol-5-yl)butoxy]-3,4-dihydro-2(1H)-quinolinone, OPC-13013) was studied for its inhibitory effect on platelet aggregation in vitro in various experimental animals and man and in dogs ex vivo... Cilostazol produced a potent inhibition of platelet aggregation.... The drug potently prevented death...cilostazol prevented both collagen- and ADP-induced platelet aggregation. cilostazol is a promising antithrombotic drug.
	[(2S,3S,5S,8R,9S,10S,13S, 14S,16S,17R)-3 acetyloxy-10,13-dimethyl-2,16 di(piperidin-1yl)-2,3,4,5,6,7,8,9,11,12, 14,15,16,17-tetradecahydro-1Hcyclopenta[a]phenanthren-17-yl] propanoate	Toxity on Androgen Receptor	Drospirenone (DRS), chemically (6R,7R,8R,9S,10R,13S,14S, 15S,16S,17S) 1,3 ,4 ,6,6a,7,8,9,10,11,12,13,14,15,15a,16-hexadecahydro-10, 13-dimethylspiro-[17H-dicyclopropa [6,7:15,16] cyclopenta[a]phenanthrene-17, 2 (5 H)-furan]-3,5 (2H)-dione (Figure 1), is used in contraception and hormone replacement therapy after menopause.

Figure 3.2: Top1 retrieved documents of molecules that are incorrectly predicted by randomly initialized Graph Isomorphism Networks (GINs), but are correctly identified by our proposed method, RTMol. Within the retrieved documents, functional group information is highlighted in green, while text pertaining to the target properties is highlighted in yellow.

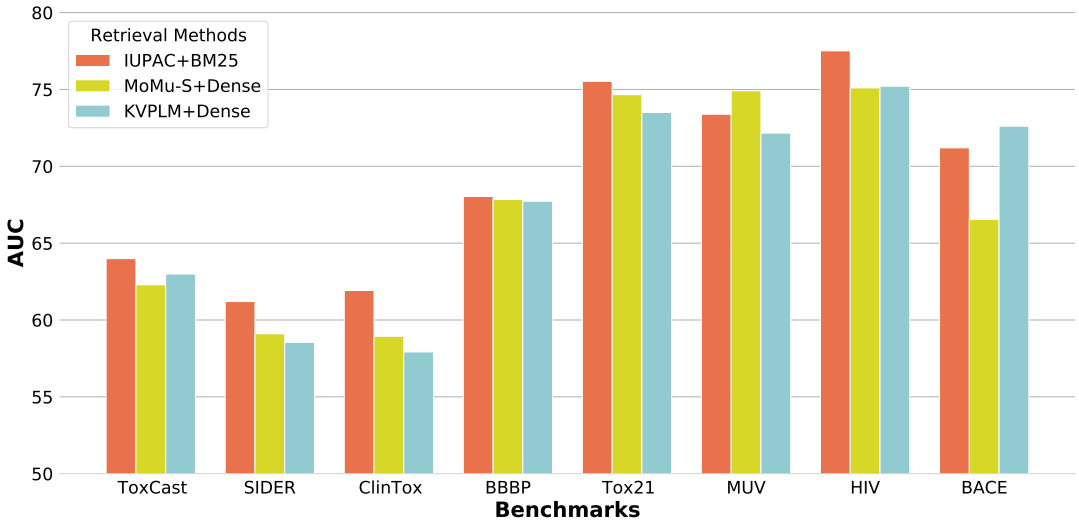


Figure 3.3: Performance of RTMol with different text-retrieval methods.

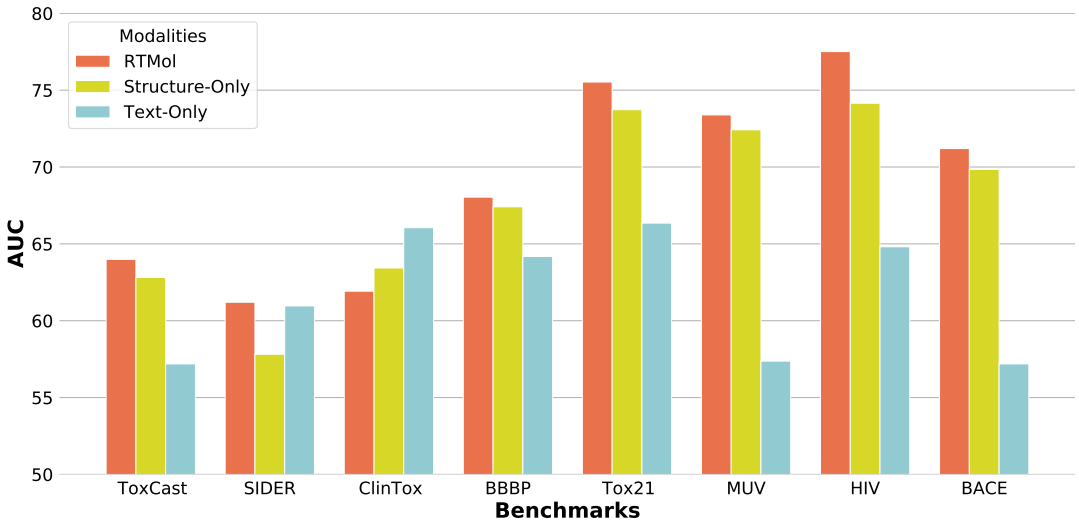


Figure 3.4: Performance of the text-augmented predictor trained/tested with inputs of different modalities.

3.4.7 Ablation Study

Molecule Dependent Text Retrieval. We justify the necessity of our proposed molecule dependent text retrieval method that uses the IUPAC names as queries and BM25 as the sparse retrieval algorithm. We replace the sparse retrieval in RTMol with two alternative cross-modal retrieval methods, MoMu and KVPLM, that can project molecules and documents into a shared low-dimensional space. For each molecule, we compute its cosine similarity with all the documents in the low-dimensional space and choose the top$K$ documents as the Molecule Dependent Documents. The results are shown in Fig. 3.3. RTMol with our proposed retrieval method outperforms the alternative methods in most of the benchmarks. More interestingly, we also found that our proposed method retrieves much more diverse documents than the two alternatives. We provide more detailed discussions and diversity results in Appendix **??**.

Performance of Different Modalities. We compare the performance of the text-augmented predictors trained/evaluated with structure-only, document-only, and multimodal input. Results shown in Fig. 3.4 indicate that the multimodal predictor outperforms single-modal predictors.

3.5 Conclusion

In this dissertation, we present RTMol, a novel multimodal learning method that utilizes both molecular structures and textual knowledge for molecular property predictions (MPP). In contrast to existing multimodal MPP models that implicitly store the textual knowledge into their parameters, RTMol learns to retrieve documents from a knowledge corpus and use the retrieved information for its predictions. In RTMol, we propose a chemically motivated molecular text-retriever to improve the exploration efficiency of the learnable retrieval system. The molecular text-retriever greatly reduces the search space and assists the learnable retrieval system to find documents that are informative for the predictions. We conduct experiments on twelve MPP benchmarks and demonstrate that RTMol is an effective multimodal MPP model that improves structure-only MPP models and it outperforms existing MPP models. Moreover, RTMol offers explainable predictions and can easily adapt to different corpora.

Chapter 4

Label Efficient Learning for Material Science

4.1 Introduction

Predictive modeling of materials is a field with manifold applications that has been the subject of many cross-disciplinary studies. While modeling is conducted at different length scales, all material behavior ultimately has its origins at the nano scale, where the interactions between individual atoms must be understood. The most accurate methods at this scale are based on quantum mechanics theory, requiring explicit consideration of the electronic degrees of freedom described by the Schrödinger equation. However, these methods are presently limited to systems containing at most several thousand atoms, precluding their use in investigation of important microstructural phenomena such as crack propagation. To overcome this limitation, practitioners have long relied upon heuristic models known as *empirical interatomic potentials* (EIPs), which consist of physically motivated analytical functional forms that strive to model the complex electronic interactions between atoms using only the nuclear coordinates of the atoms and their elemental species as input [134]. In the past several years, there has been a surge of interest in the development of machine learning EIPs, particularly those based on neural networks (NNs), as a more accurate alternative to traditional EIPs [13]. In contrast to traditional EIPs, NN potentials contain little inductive bias and, accordingly, require large volumes of training data labeled using first-principles quantum mechanical

methods. The first-principles method most commonly used is density functional theory (DFT), which scales as $\mathcal{O}(n_{\mathrm{e}}^3)$ with the number of valence electrons n_{e} in a given configuration of atoms [74]. As a result of this high computational cost, it is difficult to acquire a sufficient number of labeled training instances to create an NN potential that performs accurately over a wide range of applications.

One potential solution to this problem is to seek additional supervision signals. Traditional EIPs are attractive sources of such additional supervision for two reasons. First, their functional forms incorporate prior physical information that allows them to correlate with DFT, and, in regions of relevance to common applications, are often quite accurate. That is, they contain domain knowledge that could benefit the training of NN potentials. Second, EIPs scale linearly with the number of atoms and are thus orders of magnitude faster than DFT, permitting the labeling of massive datasets. Despite the advantages, no research has focused on using EIP supervision signals in training NN potentials.

In this dissertation, we leverage physically motivated EIPs and unlabeled configurations to tackle the label scarcity challenge for training NN potentials.We propose two generic strategies, weakly supervised learning and transfer learning, for exploiting this additional source of information. In the first strategy, we expand the DFT-labeled training set with unlabeled configurations and their EIP energies. To achieve this goal, we train an auxiliary classifier on the original DFT-labeled training configurations that predicts which one of a selected set of EIPs is likely to produce the most accurate estimate of the DFT energy for each of a large set of unlabeled configurations. The unlabeled configurations are then labeled by their corresponding predicted best-performing EIPs and appended to the training set. We train NN potentials on the expanded training set by optimizing a robust regression loss to mitigate the influence of noise and outliers introduced by the EIP energies. In the second strategy, we adopt a transfer learning approach by way of multi-task pretraining. We first pretrain the representation module of an NN potential to reproduce the energies predicted by the EIPs. During the subsequent fine-tuning stage, the representation module of the NN is paired with a prediction head and trained on the DFT-labeled configurations. These two strategies can be flexibly used and coupled to train any NN potential.

The contributions of this work are three-fold: 1) We demonstrate that EIPs are

capable of providing high quality supervision signals for training NN potentials, which opens a new direction for future development of NN potentials; 2) We propose two effective and generic strategies that take advantage of EIPs and unlabeled configurations to tackle the label scarcity challenge for training NN potentials; 3) We conduct comprehensive experiments on three benchmark datasets and four representative NN potentials that cover most of the NN potential forms currently in use. Experimental results show that the proposed strategies successfully inject domain knowledge from EIPs to NN potentials and improve the performance of the NN potentials by up to 55%.

4.2 Preliminaries

4.2.1 Atomic Configurations

The fundamental input in atomistic modeling is an *atomic configuration*. An atomic configuration is a spatial arrangement of atoms $C = \{(Z_i, \mathbf{r}_i)\}_{i=1}^{N}$ where Z_i and $\mathbf{r}_i$ are the atomic number and the three-dimensional Euclidean coordinates of atom i, respectively. In the simplest scenario, an atomic configuration corresponds to an isolated cluster of atoms that comprise a molecule. However, it may more generally describe a bulk system such as a crystal, which contains an infinite number of atoms distributed over space. These systems are modeled using a small collection of atoms in a finite simulation cell that is effectively repeated across all of space with the aid of periodic boundary conditions (PBCs) [134].

4.2.2 Physics-based Potentials

Mathematically, an EIP is a function, $E = \mathcal{V}(C; \theta)$, that takes an atomic configuration C as input and returns its total potential energy E; here, θ denotes a set of fitting parameters to be determined. The functional form $\mathcal{V}$ of a physics-based EIP is made up of carefully designed analytic expressions that strive to capture the underlying physics in the material it models [150, 151]. Because the functional form itself is intended to capture most of the relevant physics, such models need relatively few parameters (typically on the order of ten) and are usually fitted to a set of material properties deemed most relevant to real-world applications. It is expected that physics-based

EIPs will approximate the first-principles energy surface well in the vicinity of atomic configurations corresponding to the material properties to which they were fit. However, their generalizability can be inconsistent, as shown in Fig. 4.1.

4.2.3 Machine Learning Potentials

In contrast to physics-based EIPs, machine learning EIPs employ general-purpose regression algorithms as the functional form $\mathcal{V}$ that do not encode any knowledge of the material it models. Therefore, machine learning EIPs are almost exclusively trained on large sets of DFT data so as to include as much physical knowledge as possible.

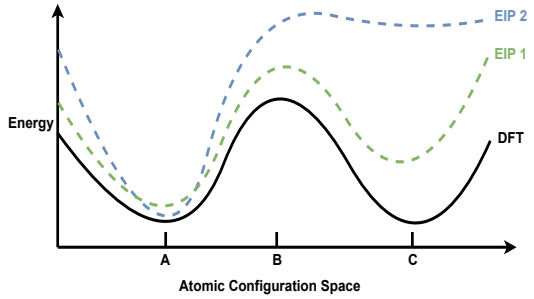


Figure 4.1: Schematic illustration of the energy landscape defined over atomic configuration space by two physics-based EIPs and by DFT. Both EIPs are fitted to reproduce properties of the DFT energy landscape near atomic configuration A. Away from configuration A, the relative accuracy of the EIPs compared to DFT varies: at point B, EIP 1 is a fair approximation while EIP 2 is less accurate; at point C, neither EIP is accurate.

One important class of machine learning EIPs are neural network (NN)-based potentials. In this dissertation, we define an NN potential $\mathcal{M} : \mathcal{C} \mapsto \mathcal{E}$ as

$$\mathcal{M}(C) = f_{\text{pred}} \circ f_{\text{rep}}(C),$$

where $f_{\text{rep}} : \mathcal{C} \mapsto \mathcal{R}^{n \times d}$ is the representation learning module that maps each of the n atoms of C to a feature vector of length d based on its local environment, and $f_{\text{pred}} : \mathcal{R}^{n \times d} \mapsto \mathcal{E}$ is the prediction module that maps the feature vectors of the atoms to the total potential energy.

A concrete instance of this type of model are those that use a graph neural network (GNN) [77, 58, 144, 123, 72] as the representation learning module and a multilayer perceptron (MLP) as the prediction module. Before being passed to the representation learning module f_{rep}, an atomic configuration C is first converted to a graph $\mathcal{G} = (V, E)$ with nodes V and edges E. One node is defined for each atom in the simulation cell, as well as for additional padding atoms representing PBCs if present. An edge is created

between any two nodes with a distance smaller than a prescribed cutoff radius. Next, for each atom i, its atomic number Z_i is one-hot encoded into an initial feature vector $h_i^{(0)}$ of the node corresponding to the atom. The GNN representation learning module f_{rep} then updates the feature vectors using a message passing paradigm [50], i.e., the nodes iteratively aggregate information from their neighbors. Formally, the feature vector of node i at the $l+1$-th layer $\mathbf{h}_i^{(l+1)}$ is updated as a function of the feature vectors of its neighbors $\mathcal{N}(i)$ and itself at the previous layer,

$$\mathbf{m}_i^{(l)} = \text{Agg}^{(l)}\left(\left\{f^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}^{(l)}\right) \mid j \in \mathcal{N}(i)\right\}\right),$$
$$\mathbf{h}_i^{(l+1)} = \text{Update}^{(l)}\left(\mathbf{h}_i^{(l)}, \mathbf{m}_i^{(l)}\right),$$

where $f^{(l)}$ and $\text{Update}^{(l)}$ are learnable functions, $\text{Agg}^{(l)}$ is a permutation-invariant function that operates on sets of feature vectors, and $\mathbf{e}_{ij}^{(l)}$ denotes the feature vector associated with the edge connecting node i and node j at the l-th layer. Finally, the prediction module f_{pred} maps the feature vector of each atom at the last layer $\mathbf{h}_i$ to a corresponding energy contribution and sums them to arrive at the total energy of the configuration, i.e., $E = \sum_i \text{MLP}(\mathbf{h}_i)$.

4.3 Related Works

4.3.1 Neural Network Potentials

The first modern NN potential was proposed by Behler and Parrinello [12]. In their formulation, an atomic descriptor (i.e., basis functions that transform an atomic configuration into a fixed-length fingerprint vector) based on the bond lengths and bond angles is passed to an MLP. On top of this formulation, a Monte Carlo dropout technique can be applied to the MLP to equip the potential with the ability to quantify its predictive uncertainty [149]. The DeePMD method is a similar approach to that of Behler and Parrinello, except that a novel atomic environment descriptor is used [169]. More recently, researchers have developed GNN potentials based on a message passing paradigm [118, 158]. Another class of GNN potentials such as NequIP [10] and GemNet [46] pass equivariant messages rather than invariant ones based on the formulation of tensor field networks [139] and achieve state-of-the-art performance. All of these models

are trained with supervised learning, without exploring the possibilities of leveraging weakly supervised learning or transfer learning to take advantage of unlabeled data.

4.3.2 Weakly Supervised Learning

Weakly supervised learning refers to techniques that attempt to train machine learning models from incomplete (only a portion of training instances are labeled) or inaccurate (noisy labels) supervision signals [174]. Solutions for incomplete supervision usually fall into the category of semi-supervised learning, which assumes that nearby instances have similar labels [81, 128, 110]. For inaccurate supervision, a model either learns directly from noisy labels with noise-robust algorithms [39, 171, 43] or resorts to a small portion of clean labeled data to reduce the noise [143, 156].

4.3.3 Transfer Learning

Transfer learning [177] refers to a machine learning paradigm that transfers knowledge a model learns from one or more relevant tasks to benefit a target task. Transfer learning has enjoyed great success, especially in the low-data regime, as demonstrated by the rise of pretrained neural networks. This recent trend began with natural language processing when BERT [30] and successive large pretrained language models [92, 18] were released and quickly gained popularity in other domains such as computer vision [61, 22, 32] and graph learning [166, 68, 98]. These methods pretrain large neural networks on self-supervised tasks in order to encode common contextual knowledge in the structured input. Another approach imparts domain-specific knowledge by pretraining models on tasks related to the target task but for which abundant labeled data is available [66, 97]. The pretrained model is then fine-tuned on the limited training data of the target task.

4.4 Methodology

4.4.1 Problem Definition

Let $\mathcal{P}$ be a set of physics-based EIPs, $\mathcal{C}$ be the space of all possible atomic configurations, $\mathcal{C}_{\text{DFT}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}, E_i^{\text{DFT}}\}_{i=1}^{m}$ be a set of m configurations with corresponding DFT and EIP energies, and $\mathcal{C}_{\text{EIP}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}\}_{i=m+1}^{m+n}$ be a set of n configurations with

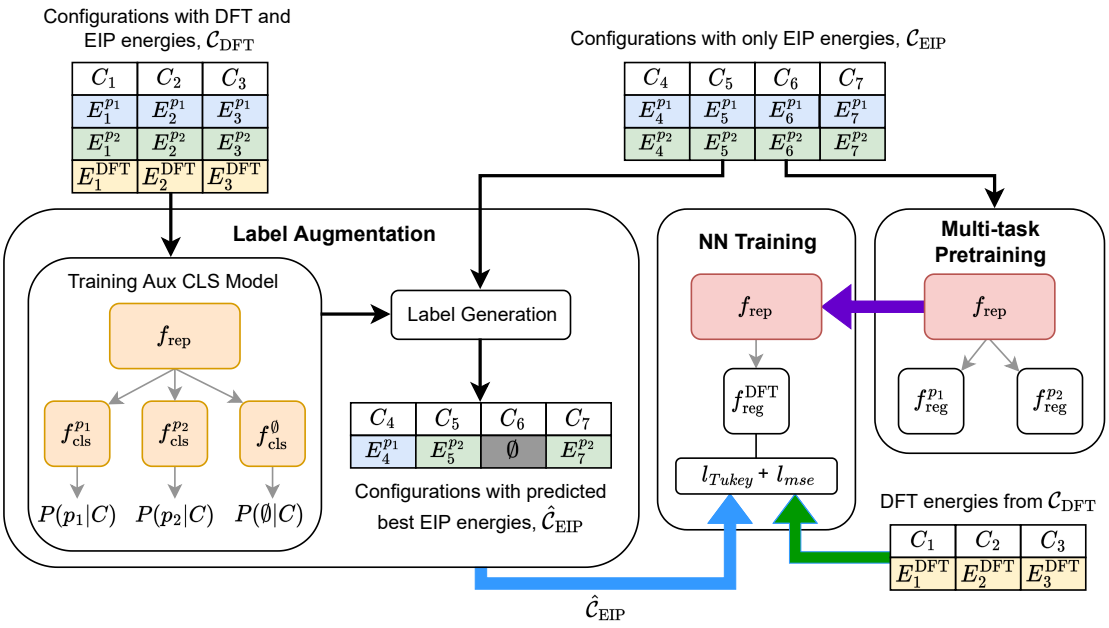


Figure 4.2: Illustration of the Label Augmentation (LA) and Multi-task Pretraining (MP) strategies and their usage in the training of NN-based potentials for the case of two physics-based EIPs, p_1 and p_2. In LA, a classifier is trained to predict the most accurate EIP energies $\hat{\mathcal{C}}_{\text{EIP}}$ for the unlabeled training instances, which are combined with the DFT energies from $\mathcal{C}_{\text{DFT}}$ in the loss function when training the final NN (blue arrow + green arrow). In MP, the representation module f_{rep} is pretrained by simultaneously fitting the energies of each physics-based EIP before it is inherited as the initial state for the representation module in the final NN training, where the DFT-labeled instances are again used (purple arrow + green arrow). The two strategies can be combined by incorporating $\hat{\mathcal{C}}_{\text{EIP}}$ into the loss while also using multi-task pretraining to initialize f_{rep} (blue arrow + purple arrow + green arrow).

only physics-based EIP energies. Here, E_i^{DFT} and E_i^p denote the energies predicted for configuration C_i by DFT and the p-th physics-based EIP, respectively. In practice, physics-based EIPs are much less expensive than DFT, and so the size of $\mathcal{C}_{\mathrm{EIP}}$ is much larger than $\mathcal{C}_{\mathrm{DFT}}$, i.e., $n \gg m$. Our goal is to use this data from physics-based EIPs to train an NN-based EIP $\mathcal{M}$ to closely approximate the DFT energy surface over the space of all atomic configurations, i.e.,

$$\mathcal{M}(C) \approx E^{\mathrm{DFT}}, \forall C \in \mathcal{C}.$$

4.4.2 Label Augmentation

Because physics-based EIPs are developed as approximations to DFT, it is intuitive to use their predictions as surrogate labels for configurations without DFT energies in order to resolve the label scarcity issue of training neural networks. However, there are two fundamental challenges. First, as discussed in Sec. 4.2.2, while physics-based EIPs are designed to be accurate in specific regions of the configuration space and generalize better than those based on machine learning, they may still be inaccurate in other regions. Given an arbitrary configuration for which no DFT energy is available, it is unknown which EIP from a given set will yield the most accurate prediction and how large its error will be. Second, surrogating DFT energies with EIP energies inevitably introduces noise, and potentially outliers, into the training set that may have pathological effects.

EIP prediction using an auxiliary classification model. In order to augment training sets with unlabeled configurations and their corresponding EIP-approximated energies, we use an auxiliary classification model to predict the best-performing EIP for a given configuration. For a configuration C, the classification model predicts a discrete distribution over the EIP set $\mathcal{P}' = \mathcal{P} \bigcup \{p_\emptyset\}$ that indicates their probability of being the most accurate EIP for C, i.e., $\mathbb{P}(\mathcal{P}' \mid C)$. We introduce a dummy EIP, $p_\emptyset$, to represent the case where none of the physics-based EIPs in $\mathcal{P}$ is predicted to approximate DFT to an accuracy level c, i.e., $\frac{1}{N_i}||E_i^p - E_i^{\mathrm{DFT}}||_1 > c, \forall p \in \mathcal{P}$, where N_i is the number of atoms in configuration i; throughout this work, c is set to 0.1. By excluding configurations that are labeled with the dummy class from the training set, c acts as a confidence threshold to control the noise and outliers introduced by using the surrogate EIP energies.

The classification model consists of a representation learning module that converts an atomic configuration to fixed-length feature vectors (one for each atom in the configuration), a permutation-invariant readout function that aggregates them to form a feature vector describing the entire configuration, and a prediction module that maps the configuration representation to a set of probabilities. We train the classification model on $\mathcal{C}_{\text{DFT}}$ by optimizing a cross-entropy loss and apply it to $\mathcal{C}_{\text{EIP}}$. Configurations with a predicted EIP other than $p_\emptyset$ are assigned the corresponding EIP energy and are merged with the configurations that have DFT energies to arrive at the final training set. We denote the set of EIP-labeled configurations as $\hat{\mathcal{C}}_{\text{EIP}} = \{C_i, \{E_i^p\}_{p\in\mathcal{P}}, E_i^{\hat{p}_i}\}_{i=m+1}^{m+s}$ where $1 \leq s \leq n$ is the number of selected configurations not labelled by DFT, and $\hat{p}_i$ and $E_i^{\hat{p}_i}$ are the predicted best-performing EIP and its prediction on C_i.

Regression with robust loss functions. We train the NN potential using configurations with ground-truth DFT energies and configurations with EIP energies selected by the classification model. In regression problems, models are usually trained by optimizing the mean square error (MSE) loss. The MSE loss is sensitive to outliers, as the magnitude of its gradient is linearly proportional to the difference between the predicted value and the ground truth value. To lessen the impact of outliers, we optimize the MSE loss on DFT-labeled configurations, while on EIP labeled-configurations, we optimize the Tukey biweight (bisquare) loss [16, 38, 14], i.e.,

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}(E_i^{\text{DFT}} - \hat{E}_i)^2 + \frac{\alpha}{s}\sum_{i=m+1}^{m+s} l_{\text{Tukey}}(E_i^{\hat{p}_i} - \hat{E}_i) \tag{4.1}$$

where $\hat{E}_i$ is the model prediction of the energy for configuration C_i, and $\alpha > 0$ is a hyper-parameter that controls the contribution of EIP-labeled configurations to the loss and its gradient.

The Tukey biweight loss falls under the M-estimation method [69] and is intended to screen outliers using robust statistics of the regression residuals such as median absolute residuals (MAR) and suppress their influence on the gradient. The Tukey loss function is computed as

$$l_{\text{Tukey}}(r_i) = \begin{cases} \frac{k^2}{6}\left[1 - \left(1 - \left(\frac{r_i}{k}\right)^2\right)^3\right], & |r_i| \leq k \\ \frac{k^2}{6}, & |r_i| > k \end{cases} \tag{4.2}$$

where $r_i = E_i^{\hat{p}_i} - \hat{E}_i$ is the residual and k is a tuning constant that is commonly set to 4.685σ to produce 95% efficiency when the errors are normally distributed with standard deviation σ. To set the value of k, we estimate the standard deviation as $\hat{\sigma} = \text{MAR}/0.6745$. From Eq. 4.2, residuals with absolute values greater than k are considered outliers and are rejected for gradient computation. In each gradient step, we sample from both DFT- and EIP-labeled configurations to form a batch for gradient back-propagation, and the MAR is estimated on residuals of both DFT- and EIP-labeled configurations in the batch. Since the MAR and $\hat{\sigma}$ are dynamically estimated during training, the Tukey loss does not introduce additional hyperparameters.

4.4.3 Multi-task Pretraining

We propose a multi-task pretraining strategy to encode the domain knowledge in physics-based EIPs to the parameters of the representation learning module by jointly predicting the set of EIP calculations for configurations with only EIP predictions, $\mathcal{C}_{\text{EIP}}$, and optimizing the following multi-task regression loss:

$$\mathcal{L} = \frac{1}{n|\mathcal{P}|} \sum_{p \in \mathcal{P}} \sum_{i=m+1}^{m+n} (\hat{E}_i^p - E_i^p)^2.$$

During pretraining, we couple the representation learning module with $|\mathcal{P}|$ prediction modules (MLPs) to generate predictions corresponding to different physics-based EIPs, i.e., $\hat{E}_i^p = f_{\text{pred}}^p \circ f_{\text{rep}}(C_i)$. The pretrained representation module is then fine-tuned with a randomly initialized prediction module for the downstream DFT prediction task. Note that the representation module could either be naively fine-tuned on configurations with DFT energies by optimizing an MSE loss, or on the training set generated by our proposed label augmentation method by optimizing Eq. 4.1.

Although transfer learning has been successful in various application domains, it could easily hinder model performance on the target task if the pretraining tasks are unrelated to the target task (negative transfer). We argue that predicting the output of physics-based EIPs is relevant and beneficial to the target task of predicting DFT energies. Although physics-based EIPs are not perfectly accurate across the space of all possible spatial arrangements of atoms, their functional forms incorporate prior physical information that allow them to correlate with DFT over this space. In Sec. 3.4, we

empirically demonstrate that the multi-task pretraining strategy successfully encodes domain knowledge into configuration representations and creates a smoother DFT energy surface.

4.4.4 Combining Label Augmentation and Multi-task Pretraining

The label augmentation and multi-task pretraining methods outlined above can be combined with relative ease. The procedure is similar to the ordinary label augmentation strategy, but rather than using a randomly initialized representation module f_{rep} for the final NN training, the representation module produced by the multi-task pretraining method is used during fine-tuning. Fig. 4.2 provides a schematic overview of both strategies, how they relate to one another, and how they can be combined.

To test our methodology, we experiment with three datasets: the ANI-Al [127] dataset and the KIM-Si [73] dataset each with a single species, as well as a multispecies AgAu dataset [147]. For each dataset, we generate three splits by randomly assigning 20% of the DFT-labeled configurations as test sets and the other 80% as training sets. During training, we use 20% of the training set as a validation set for model selection. All of the reported experimental results are averaged over three different splits to avoid over-fitting to a specific split. We release our code [1] and the KIM-Si dataset [2] for reproducing our experimental results and continuous works.

4.4.5 Experimental Setting

Neural Network-based Potentials

We evaluate our proposed strategies on two classes of neural network potentials that reflect the majority of machine learning potentials currently in use. The first represents atomic environments using pre-computed descriptors and learns non-linear transformations (MLPs) to map the descriptors to atomic embeddings, while the second uses GNNs to learn atomic representations from configuration graphs. We select one representative potential from each class. For our MLP-based potential, we use the Smooth Overlap

[1] `https://github.com/shuix007/EIP4NNPotentials`
[2] `https://doi.org/10.6084/m9.figshare.21266064`

of Atomic Positions (SOAP) [6] atomic environment descriptor together with a representation module and prediction module consisting of MLPs; we term this potential SOAPNet in later discussions. For our GNN-based potential, we select SchNet [118], CGCNN [158], and GemNet [46]. In our label augmentation experiments, we set the representation module of the auxiliary classification model to be the same kind as the corresponding NN potential, e.g., the classification model used for training the SchNet potential has a SchNet GNN as its representation module.

Selection of Physics-based EIPs

The physics-based EIPs used in our experiments were selected to encompass differing levels of functional complexity. Because physics-based EIPs are designed for specific elemental species (in this case, aluminum, silicon, and gold and silver systems), a different set of physics-based EIPs had to be chosen for each dataset. A total of ten physics-based EIPs were used for aluminum, eight for silicon, and two for the gold-silver system.They were taken mainly from the Open Knowledgebase of Interatomic Models (OpenKIM)[3] repository. [135, 136].

4.4.6 Experimental Results

Performance of Label Augmentation and Multi-task Pretraining

As shown in Tab. 4.1, our proposed strategies improve the performance of the four baseline NNs on the three benchmark datasets. In particular, the label augmentation strategy improves the baseline NNs by 5% to 51%, while the multi-task pretraining strategy improves the baselines by 2% to 55%. Combining the two strategies gives further improvement.

EIP Energies as High-Quality Supervision Signals for Training NN Potentials

Recall that in the label augmentation strategy, an auxiliary classification model selects unlabeled configurations and predicts their corresponding best-performing physics-based EIPs, which are subsequently used to label them for the training of the NN potential. These configurations are then labeled by the predicted best-performing physics-based

[3] `https://openkim.org/`

Table 4.1: Performance of the two proposed strategies on DFT energy prediction tasks. We report the configuration-level and atom-level mean absolute error (MAE, lower is better) in eV and eV/atom, respectively. We denote the label augmentation strategy by LA and the multi-task pretraining strategy by MP. Best performance is shown in bold. Cases where the training procedure failed due to running out of memory are marked OOM.

	KIM-Si			ANI-Al			AgAu		
	Config	Atom	Improv.	Config	Atom	Improv.	Config	Atom	Improv.
Best EIP	1.6326	0.2524	-	46.4869	0.3561	-	4.4587	0.2063	-
SOAPNet	0.7706	0.0975	-	0.2153	0.0017	-	0.5422	0.0226	-
+LA	0.5595	0.0704	27.58%	0.1786	0.0014	18.14%	0.5067	0.0205	07.92%
+MP	0.5717	0.0733	25.28%	0.1744	0.0014	19.12%	0.3962	0.0154	29.34%
+MP+LA	**0.5307**	**0.0657**	**31.88%**	**0.1697**	**0.0013**	**22.13%**	**0.3858**	**0.0154**	**30.23%**
SchNet	0.4805	0.0718	-	0.1693	0.0014	-	0.7290	0.0290	-
+LA	0.4015	0.0549	19.99%	0.0845	0.0007	51.24%	0.6815	0.0266	07.33%
+MP	0.4034	0.0569	18.40%	0.1296	0.0010	26.00%	**0.3353**	**0.0130**	**54.65%**
+MP+LA	**0.3719**	**0.0490**	**27.17%**	**0.0816**	**0.0006**	**53.27%**	0.3496	0.0135	52.80%
CGCNN	0.9314	0.1410	-	0.2410	0.0019	-	1.6683	0.0625	-
+LA	0.7476	0.1050	22.61%	0.1786	0.0014	25.44%	1.6065	0.0589	04.71%
+MP	0.8457	0.1253	10.16%	0.2206	0.0017	07.80%	1.4377	0.0532	14.38%
+MP+LA	**0.7435**	**0.1005**	**24.44%**	**0.1392**	**0.0011**	**41.65%**	**1.3857**	**0.0499**	**18.55%**
GemNet	0.5138	0.0546	-	OOM	OOM	OOM	0.9257	0.0342	-
+LA	0.4691	0.0511	07.55%	OOM	OOM	OOM	0.8381	0.0300	10.87%
+MP	0.5024	0.0531	02.48%	OOM	OOM	OOM	**0.5057**	**0.0185**	**45.71%**
+MP+LA	**0.4651**	**0.0476**	**11.12%**	OOM	OOM	OOM	0.6074	0.0218	35.30%

Table 4.2: Average number of configurations and outliers selected by the classification models on the ANI-Al dataset.

	#Mild	#Normal	#Severe	#Selected	#Unlabeled
SOAPNet	226	42	5	1211	5081
SchNet	271	50	9	1300	5081

EIPs for NN potential training. We investigate the quality of EIP labels and the auxiliary classification model by training NN potentials on three augmented training sets in which the selected unlabeled configurations are labeled by three different sources: DFT-based energies, ground-truth best-performing-EIP energies, and predicted best-performing EIP energies. The DFT-labeled configurations (0.8K) are the same for the three training sets. We only conduct this experiment on the ANI-Al dataset, as all of its configurations have DFT energies available. The number of selected unlabeled configurations is shown in Tab 4.2.

Fig. 4.3(a) shows the performance of the NN potentials trained on the three augmented training sets and the original training set (where only DFT-labeled configurations are used). As shown in the figure, expanding the training set with ground truth DFT calculations (blue) greatly improves the baseline MAE (yellow bar, model trained on the original training set). Labeling configurations with the ground-truth best-performing physics-based EIPs (red bar) performs slightly worse than with DFT energies (blue bar) but still much better than the baseline (yellow bar). This demonstrates that physics-based EIPs are valuable sources of supervision signals for training NN-based potentials. Using the predicted best-performing physics-based EIPs for labeling (green bar) performs on par with the ground-truth best-performing physics-based EIP labeling (red), revealing the utility of the auxiliary classification model.

Importance of the Robust Tukey Loss

We next conduct experiments to investigate the importance of the Tukey loss and its ability to reject outliers during training. As before, we only conduct this experiment on the ANI-Al dataset, using DFT energies for unlabeled configurations to determine noise

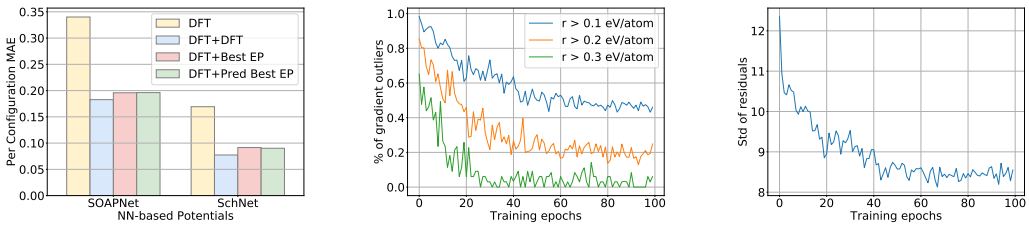
Table 4.3: Configuration-level MAE (eV) with and without the Tukey loss.

	KIM-Si		ANI-Al	
	SOAPNet	SchNet	SOAPNet	SchNet
w/o Tukey	0.5556	0.4374	0.1906	0.1031
w/ Tukey	0.5595	0.4015	0.1786	0.0845

and outliers introduced by the predicted best-performing physics-based EIPs. We define an unlabeled configuration with a predicted best-performing physics-based EIP to be an outlier if the absolute difference between its physics-based EIP energy and its DFT energy is larger than a threshold, i.e., $|r_i| = \frac{1}{N_i}|E_i^{\text{DFT}} - E_i^{\hat{p}_i}| > c$ where N_i is the number of atoms in configuration i. We categorize the outliers as mild, normal, and severe by setting c to $\{0.1, 0.2, 0.3\}$. The initial number of outliers introduced by the unlabeled configurations can be found in Tab. 6 (in Appendix). Figs. 4.3(b) and 4.3(c) show that the number of outliers of all kinds included for computing gradients decreases as the training proceeds, demonstrating that as the NN potential gets progressively more accurate, the Tukey loss can effectively eliminate outliers from the training set. We also conduct an ablation study by replacing the Tukey loss in Eq. 4.1 with the MSE loss. The results in Tab 4.3 show that the models' performance degrades without the Tukey loss.

Visualization of Pretrained Configuration Representations

Fig. 5.1 plots the t-SNE [142] 2D projections of the training silicon configuration representations colored by their per-atom DFT energies. The left-hand figure plots representations generated by a SchNet with random weights and the right-hand figure plots representations generated by a SchNet pretrained by the multi-task strategy. The representations generated by the randomly initialized SchNet do not exhibit any clear patterns and the energy surface is rough. In the right-hand figure, representations of the atomic cluster configurations (i.e., isolated groups of atoms) and the bulk configurations (crystals) are clearly separated and form clusters in the t-SNE 2D space. The



(a) (b) (c)

Figure 4.3: (a) Performance of NN potentials trained on the original training sets (yellow bar, DFT-labeled configurations only) and three augmented training sets whose unlabeled configurations are labeled by DFT energies (blue bar), ground-truth best-performing physics-based EIP energies (red bar), and predicted best-performing physics-based EIP energies (green bar). (b) Percentage of outliers used for computing gradient during training. (c) Standard deviation of residuals during training.

per-atom DFT energy surface of the right-hand figure is much smoother, i.e., configurations with similar energies are close to one another after pretraining. This verifies our previous statement that, although physics-based EIPs lack complete generalizability, they nonetheless correlate reasonably well with DFT over atomic configuration space, and demonstrates that our proposed multi-task pretraining strategy successfully encodes domain knowledge into the NN-based potentials.

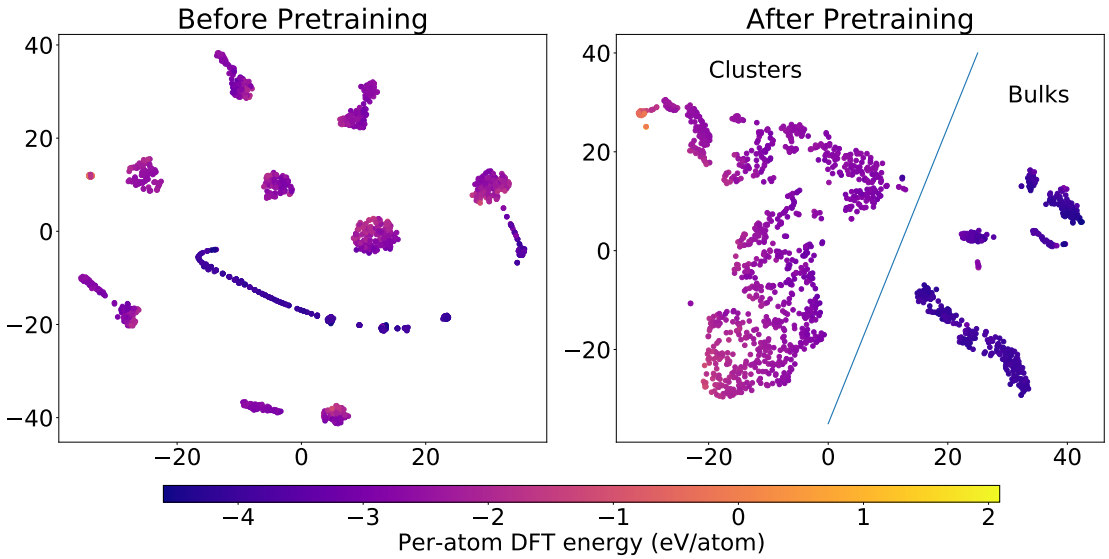


Figure 4.4: T-SNE plots of silicon configuration representations generated by a randomly initialized SchNet (left) and a SchNet pretrained with our proposed multi-task pretraining strategy (right). Configurations are colored by their per-atom DFT energies. Representations generated by the pretrained SchNet naturally form two clusters that correspond to the atomic cluster configurations (i.e., isolated groups of atoms) and the bulk configurations (i.e., crystals).

Chapter 5

Data Selection for Pre-training Machine Learning Force Fields

5.1 Introduction

Accurately predicting the properties and modeling the behaviours of atomic configurations (i.e., a set of atoms and their spatial arrangements) are fundamental to drug discovery and material design tasks. Machine learning force fields (MLFF) are machine learning models that predict potential energies and atom-wise forces for atomic configurations. These models are trained to approximate and surrogate computationally expensive first-principle methods, such as density functional theory (DFT), which are widely utilized in atomistic modeling tasks such as molecular property prediction and molecular dynamics simulations. Recent years, researchers have been making efforts in pre-training MLFFs to encode the underlying physics into them as an upfront investment [122]. The pre-trained MLFFs can be fine-tuned with improved accuracy and enhanced data efficiency across a broad spectrum of low-resource chemical systems.

Although pre-trained MLFFs have shown great performance in atomistic modeling tasks, they often lack the capability to capture domain-specific features and nuances required for accurately modeling diverse atomic systems. In natural language processing (NLP), Domain Adaptive Pre-Training (DAPT) [54] has emerged as a promising method for adapting pre-trained language models to specific domains by further pre-training the already pre-trained model on additional datasets in the target domain. This adaptation

improves the model's ability to understand and process domain-specific language and concepts, thereby improving its performance on the target tasks. However, applying DAPT for training MLFFs presents unique challenges. Unlike datasets used for pre-training language models, which do not require labels, datasets for training MLFFs necessitate energy and force labels generated by expensive DFTs. Consequently, further pre-training with DAPT in force field learning incurs significant labeling costs to curate domain-specific datasets.

To ensure a pre-trained Machine Learning Force Field (MLFF) captures the complexity of the underlying chemical space, its pre-training dataset is curated to be large and diverse, covering various domains. This raises the research question: *Can we improve the performance of an MLFF on a specific task by identifying and pre-training on a domain-specific subset of this dataset?* In this dissertation, we propose Data Selection Pre-training (DSP), a novel method that adapts a pre-trained MLFF to the domain of a target task without requiring additional domain-specific datasets. DSP achieves this goal by selecting a task-specific subset from the pre-training dataset for domain-adaptive pre-training. We formulate the data selection problem as a budget-constrained linear programming (LP) [27] problem, which maximizes the relevance of the configurations included in the subset with respect to the target task. We prove that this LP problem can be efficiently solved by sorting the task relevance scores of the pre-training configurations.

The contributions of this work are three-fold: First, we introduce Data Selection Pre-training (DSP), a novel method that enhances the performance of pre-trained MLFFs by selecting and pre-training on domain-specific subsets from a diverse pre-training dataset, eliminating the cost for labeling additional domain-specific data. Second, we formulate the data selection problem as a budget-constrained linear programming (LP) problem, which efficiently maximizes the relevance of the configurations for the target task. Third, we conduct experiments on 18 force field benchmarks to demonstrate that DSP significantly improves the accuracy of MLFFs in domain-specific applications.

5.2 Related Works

5.2.1 Machine Learning Force Field

Machine learning force fields (MLFF) are machine learning models that predict potential energies and atom-wise forces for atomic configurations. Earlier MLFFs featurize atomic configurations using physics motivated descriptors and feed them as input to machine learning models to predict the energies and the forces [170, 149]. Recently have seen a trend in developing graph neural network (GNN)-based force fields given their potential in modeling complex interactions among atoms. These models resort to the message passing mechanism [49] to encode geometries including atomic distance [118, 158], bond angle [21, 124, 78, 44], torsion angle (dihedral angle) [45, 47] into latent atomic representations which are used for property prediction. Equivariant graph neural networks introduce spherical harmonics and tensor products to the message passing mechanism such that the intermediate representations are rotationally equivariant [11, 9, 86, 87, 178].

5.2.2 Pre-training

The paradigm of training machine learning models, especially deep neural networks, has shifted from end-to-end training from scratch to pre-training then fine-tuning. The trend started from natural language processing (NLP) [31, 92, 105, 104], then quickly revolutionized other fields including computer vision (CV) [62, 5, 19] and graph learning [68, 161, 166, 65, 125]. In chemistry, various pre-training strategies have been developed to pre-train graph neural networks on 2D topology and 3D geometry of stable molecules using self-supervised objectives [113, 133, 88, 130]. As of force field learning, researchers also explored pre-training MLFFs using large scale DFT datasets [122, 79, 162].

In NLP, researchers found that further pre-training language models (LMs) on domain relevant corpus effectively improves the LMs performance on tasks in the specific domain [54, 15]. However, unlike datasets used for pre-training language models, which do not require labels, datasets for training MLFFs necessitate energy and force labels generated by expensive DFTs. This incurs significant labeling cost to curate domain relevant datasets.

5.2.3 Data Selection

Earlier data selection (a.k.a., data pruning) methods aim to reduce the training cost of machine learning models. They select the most influential / representative data (coreset) from the entire dataset so that models trained on the coreset perform as well as if they were trained on the entire dataset [4, 99, 137, 155, 173, 103, 107]. In the era of large language models, training data have shifted from human labeled datasets to massive web-crawls which contain noise and duplicates that impairs the learning of LMs [63]. Research in data selection focus on creating high quality datasets by deduplication [1, 82, 140] and quality filtering [116, 152]. Researchers also demonstrate that training on filtered high-quality datasets improves neural scaling laws [52, 129, 100]. In this line of research, there are two works that have similar tastes with ours. One that select data for specific downstream instruction tuning tasks [154]. Another that select from a large pool of language modeling data for further pre-training LMs for specific downstream tasks [157]. In the field of material science, researchers also noticed the redundancy in material datasets [84, 102, 162]. However, there hasn't been data selection methods developed for pre-training MLFFs. Moreover, unlike our proposed DSP, these existing methods are agnostic to budgets.

5.3 Machine Learning Force Field

Let $C = \{z_i, \mathbf{r}_i\}_{i=1}^{N} \in \mathcal{C}$ be an **atomic configuration** from a configuration space $\mathcal{C}$, where $z_i \in \mathcal{N}$ and $\mathbf{r}_i \in \mathcal{R}^3$ represent the atomic number and the three-dimensional Euclidean coordinates of atom i in the configuration, respectively. A first-principle force field (e.g., density functional theory) $\mathcal{M}(\cdot)$ computes the energy $E(C) \in \mathcal{R}$ for the configuration, i.e., $E(C) = \mathcal{M}(C)$. By definition, the force $\mathbf{F}_i(C) \in \mathcal{R}^3$ on an atom i is the partial derivative of the energy w.r.t. its Euclidean coordinates, i.e., $\mathbf{F}_i(C) = -\frac{\partial \mathcal{M}(C)}{\partial \mathbf{r}_i}$. Note that, there are numerous first-principle force fields, their computation can vary significantly from one another. Let $T \in \mathcal{T}$ be a task of interest in a task space $\mathcal{T}$. A machine learning force field (MLFF) with parameters Θ, denoted as $\mathcal{M}(\cdot|\Theta)$, is a machine learning model trained on a dataset $\mathcal{D}_T = \{C_i, E(C_i), \mathbf{F}(C_i)\}_{i=1}^{|\mathcal{D}_T|}$ that

represents T by optimizing

$$\mathcal{L}(\Theta) = \frac{1}{|\mathcal{D}_T|} \sum_{i=1}^{|\mathcal{D}_T|} \left[\lambda_E \mathcal{L}_E \left(E(C_i), \hat{E}(C_i|\Theta) \right) + \lambda_F \mathcal{L}_F \left(\mathbf{F}(C_i), \hat{\mathbf{F}}(C_i|\Theta) \right) \right],$$

where $\hat{E}(C_i|\Theta)$ and $\hat{\mathbf{F}}(C_i|\Theta)$ are the energy and forces predicted by $\mathcal{M}(\cdot|\Theta)$, respectively. There are two classes of MLFFs that compute the forces differently. The first class computes the forces as the partial derivatives of the predicted energy while the second class trains a separate prediction head to directly predict the forces. We refer readers to the relevant references for further details [45, 47, 67, 40].

5.4 Data Selection Pre-training

5.4.1 Problem Formulation

Let $\tilde{\mathcal{D}} = \{\tilde{C}_i, \tilde{E}(\tilde{C}_i), \tilde{\mathbf{F}}(\tilde{C}_i)\}_{i=1}^{|\tilde{\mathcal{D}}|}$ be a large and diverse pre-training dataset, and $\mathcal{D}_T = \{C_i, E(C_i), \mathbf{F}(C_i)\}_{i=1}^{|\mathcal{D}_T|}$ represents a dataset for a task of interest T. We assume a general MLFF $\mathcal{M}(\cdot|\tilde{\Theta})$ that has already been pre-trained on $\tilde{\mathcal{D}}$. Our goal is to adapt $\mathcal{M}(\cdot|\tilde{\Theta})$ to the domain of T by selecting a task specific subset $\tilde{\mathcal{D}}_T \subset \tilde{\mathcal{D}}$ and further pre-training $\mathcal{M}(\cdot|\tilde{\Theta})$ on the subset to obtain a domain specific MLFF $\mathcal{M}(\cdot|\tilde{\Theta}_T)$ which is finally fine-tuned on $\mathcal{D}_T$. We refer this domain adaptive process as data selection pre-training (DSP). We posit a budget B for DSP that is measured as the total number of configurations that $\mathcal{M}(\cdot|\tilde{\Theta})$ will be further pre-trained on. The budget B can be larger than the size of $\tilde{\mathcal{D}}$ as one configuration can be trained on for multiple times.

5.4.2 Budget Aware Data Selection

At the core of DSP is an algorithm $\mathcal{A}$ that selects $\tilde{\mathcal{D}}_T$ from $\tilde{\mathcal{D}}$ and identifies how many times each configuration appears during the DSP training. The algorithm is dependent on the task T and the budget B. To adapt $\mathcal{M}(\cdot|\tilde{\Theta})$ to the target task T, DSP selects configurations that are relevant to T for the further pre-training. We formulate the data

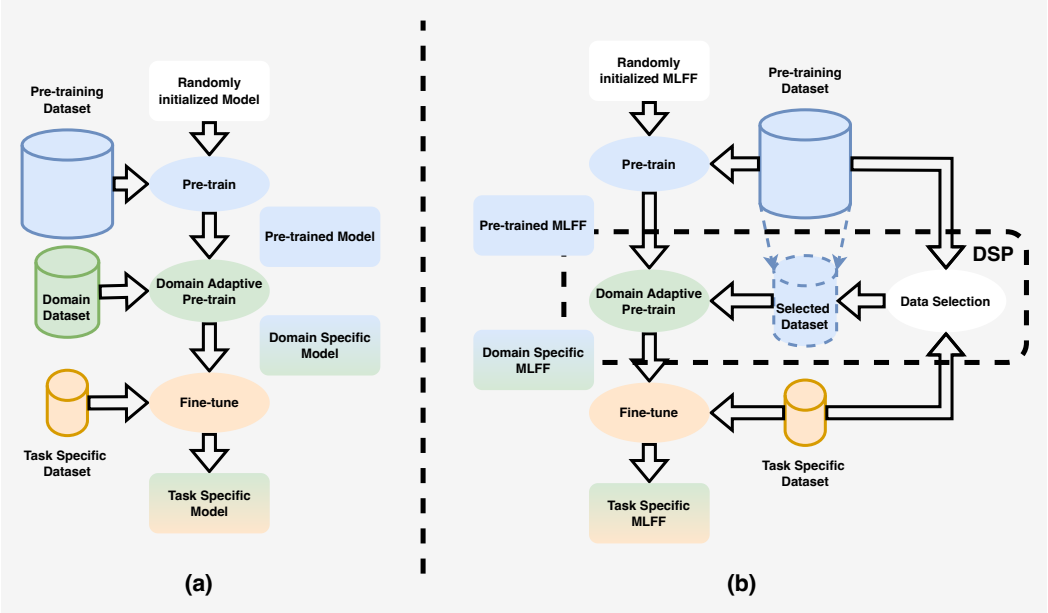


Figure 5.1: (a) Data Adaptive Pre-Training (DAPT) adapts a pre-trained model to a specific domain by further training it on an additional domain specific dataset; (b) Our proposed Data Selection Pre-training (DSP) adapts a MLFF to the domain of a target task by selecting a task specific subset from the pre-training dataset then training the MLFF on the subset.

selection problem as a linear programming (LP) problem [27], i.e.,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{i=1}^{|\tilde{\mathcal{D}}|} a_i^T x_i \\ \text{s.t.} \quad & m \geq x_i \geq 0, \forall i \in \{1, \ldots, |\tilde{\mathcal{D}}|\} \qquad (5.1) \\ & \sum_{i=1}^{|\tilde{\mathcal{D}}|} x_i \leq B \qquad (5.2) \end{aligned}$$

where $\mathbf{x} = [x_1, \ldots, x_{|\tilde{\mathcal{D}}|}]$ are the variables to be optimized, x_i indicates the number of appearance of $\tilde{C}_i \in \tilde{\mathcal{D}}$ in DSP, $a_i^T > 0$ measures the relevance of $\tilde{C}_i$ to the task T. We posit that after training the MLFF on a configuration $\tilde{C}_i$ for m epochs, the magnitude of the loss of $\tilde{C}_i$ will be negligible to lead to meaningful gradient updates to the the MLFF's parameters. Further training on the configuration will cause a waste of budget. In Eq 5.1, we bound x_i by m. In our experiments, we treat m as a hyper-parameter and tune it using validation sets. Eq 5.2 constraints the total number of DSP training instances to be under the budget.

Proposition. *The solution to the linear programming problem is*

$$x_i = \begin{cases} m & |\{j \mid a_i < a_j\}| \leq \frac{B}{m} \\ 0 & otherwise. \end{cases}$$

This Proposition states that by solving the LP problem, DSP selects the configurations with top-k task relevance scores, where $k = B/m$. It also implies that, if given a sufficiently large budget, DSP will select all configurations for the further pre-training.

5.4.3 Task Relevance

To measure the relevance of pre-training configurations to the task T represented by the task-specific dataset $\mathcal{D}_T$, we use the Smooth Overlap of Atomic Positions (SOAP) [7, 28] descriptor to convert all the configurations into a shared embedding space $\mathbf{h}(C)$. The SOAP descriptor is a physics-based embedding method that converts atomic configurations into a vector space while ensuring invariance to the translation and rotation of atomic coordinates. Moreover, SOAP is a universal embedding method that applies to

any configurations. We represent the target task T as the average embedding of all configurations in $\mathcal{D}_T$, i.e.,

$$\mathbf{h}_T = \frac{1}{|\mathcal{D}_T|} \sum_{i=1}^{|\mathcal{D}_T|} \mathbf{h}(C_i).$$

For each pre-training configuration $\tilde{C}_i$, we compute its task relevance score $a_i^T = \text{Cosine}(\mathbf{h}(\tilde{C}_i), \mathbf{h}_T) + 1$ where $\text{Cosine}(\cdot, \cdot)$ refers to the cosine similarity.

5.5 Experiments

5.5.1 Datasets

Pre-train: We use **ANI-1x** [126] as the pre-training dataset in our experiments. The ANI-1x dataset contains DFT calculated energy and forces for 5M diverse atomic configurations of organic molecules. We randomly sample 200K configurations from the ANI-1x dataset for pre-training the MLFFs.

Fine-tune: In our experiments, we utilize the **rMD17**, **SPICE**, and **MD22** datasets, which collectively encompass **18** target fine-tuning datasets. rMD17 contains molecular dynamic simulation trajectories (MDS) of ten small organic molecules. Each trajectory contains different configurations of one molecule. We split the trajectory of each molecule into train/validation/test sets of size 950/50/1000. SPICE contains configurations of drug-like molecules and proteins. We experiment with the dipeptides and the solvated amino acids subsets that contain 34K configurations of 676 dipeptides (formed by different combinations of 26 amino acids) and 1300 configurations of 26 amino acids, respectively. We randomly leave out five amino acids from the solvated amino acids and the 25 dipeptides they form from the dipeptides subset to evaluate the out-of-distribution performance of the MLFFs. We randomly split the remaining in-distribution configurations in a 8/1/1 ratio to be the training/validation/test sets. MD22 contains MDS trajectories of large molecules such as carbohydrates and nucleic acids. We evaluate DSP on six trajectories using the train/validation/test split proposed in the original paper [23].

Table 5.1: Performance (Force MAE, unit meV/A, lower ↓ the better) of MLFFs. "Baseline" refers to results of randomly initialized MLFFs. "Default" is the fine-tuning performance of the generally pre-trained MLFF. "Full" further pre-trains the general MLFF using the entire pre-training dataset. All the experiments are repeated three times with different seeds. We use **bold** to denote the method with the best performance, and ↓ to denote the further pre-training method improves the generally pre-trained MLFFs.

Dataset	Molecule	Baseline	5 Epochs			20 Epochs		
			Default	Full	DSP	Default	Full	DSP
rMD17	Aspirin	21.87	15.86	15.30	**15.13** ↓	13.30	13.07	**13.03** ↓
	Azobenzene	13.97	10.96	10.80	**10.73** ↓	10.10	9.97	**9.93** ↓
	Benzene	4.83	3.48	3.53	3.63	3.46	3.30	**3.30** ↓
	Ethanol	12.73	9.58	10.27	9.63	8.34	8.00	8.23 ↓
	Malonaldehyde	17.87	14.82	14.67	**14.57** ↓	13.04	13.17	13.47
	Naphthalene	9.27	7.40	7.33	**7.23** ↓	6.82	6.83	**6.73** ↓
	Paracetamol	18.93	13.18	12.70	**12.63** ↓	11.50	11.57	**11.33** ↓
	Salicylic	16.73	12.10	11.97	**11.70** ↓	10.94	10.97	**10.50** ↓
	Toluene	9.50	7.22	7.13	7.20 ↓	6.46	6.70	6.60
	Uracil	14.40	11.08	10.93	**10.70** ↓	9.92	10.03	**9.60** ↓
SPICE	Dipeptides	25.37	23.34	22.83	**22.80** ↓	21.88	21.90	**21.83** ↓
	Amino acids	50.50	40.74	38.27	**33.10** ↓	31.94	31.70	**29.53** ↓
MD22	AT-AT	33.93	21.00	20.20	**19.97** ↓	16.80	16.57	**16.37** ↓
	AT-AT-CG-CG	45.23	27.65	26.53	**26.20** ↓	21.93	21.73	**21.50** ↓
	Ac-Ala3-NHMe	21.97	15.15	14.67	**14.50** ↓	12.50	12.33	**12.17** ↓
	DHA	17.30	12.98	12.57	**12.53** ↓	10.87	10.70	**10.63** ↓
	Buckyball-catcher	134.97	66.03	65.17	**63.73** ↓	63.53	62.53	**61.60** ↓
	Stachyose	22.57	17.33	16.83	**16.77** ↓	14.90	14.70	**14.63** ↓

5.5.2 Machine Learning Force Field

In our experiments, we use GemNet-dT [45] as the backbone architecture for MLFFs. It represents a broad class of message passing neural network-based force fields. GemNet-dT is also widely employed in various atomistic modeling tasks such as structural relaxation [20], molecular dynamic simulations [40], and material generation tasks [168]. MLFFs are usually pre-trained for a few epochs to ensure their generality. We pre-train two general MLFFs on the ANI-1x dataset for 5 epochs and 20 epochs, respectively, to evaluate the performance of DSP on MLFFs at different training stages.

5.5.3 Implementation and Evaluation

We pre-train two general MLFFs on the ANI-1x dataset for 5 epochs and 20 epochs, respectively, to evaluate the performance of DSP on MLFFs at different training stages. In all our experiments, we (further) pre-train the MLFFs using different methods, fine-tune them separately on the downstream tasks, then evaluate them on the corresponding test sets. Since force field learning is a regression problem, we use force mean absolute error (MAE, unit meV/A) as the main metric to evaluate the accuracy of the MLFFs.

5.5.4 Experimental Results

Main Results

Table 5.1 shows the performance of DSP on the 18 benchmark datasets with a budget of $B = 2|\tilde{\mathcal{D}}|$. For both base MLFFs pre-trained with 5 and 20 epochs, further pre-training using DSP improves their accuracy on 16 out of the 18 benchmarks. This result demonstrates that DSP effectively adapts the already pre-trained MLFFs to the specific tasks, enhancing their performance on those tasks. Moreover, DSP enhances the performance of MLFFs irrespective of the number of epochs they have already been trained on the pre-training datasets. With the same budget, DSP outperforms further pre-training with the entire pre-training dataset on 14/15 out of the 18 benchmarks on the based MLFFs pre-trained for 5/20 epochs. This indicates that, given a fixed budget for further pre-training, repeated training on a small set of relevant configurations has more utility than spending the budget on the entire pre-training dataset. This finding highlights the necessity of the data selection algorithm.

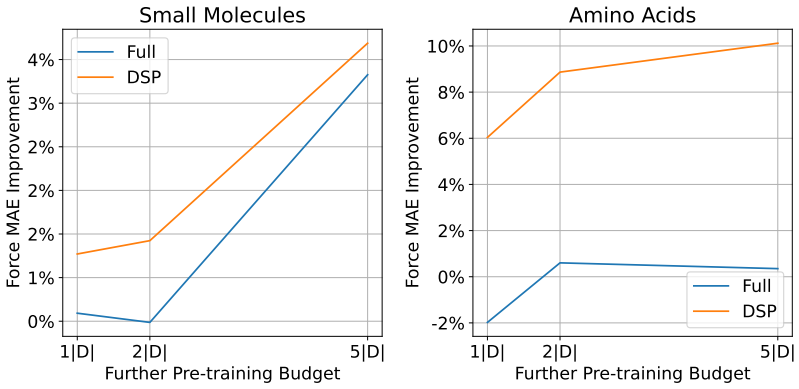


Figure 5.2: Performance of DSP under different budgets. The y-axis is the relative improvement compared to the base MLFF pre-trained for 20 epochs.

Performance Under Different Budgets

We investigate the effectiveness of DSP under different budgets and show the results in Figure 5.2. The improvement obtained through DSP increases with the budget. Moreover, across various budgets, further pre-training the base MLFF with DSP consistently outperforms pre-training with the full dataset.

Ablation Study on the SOAP Descriptor

In this study, we compare the embedding method that we used for computing task relevance, SOAP, with another embedding method that relies on a pre-trained neural network, CHGNet [29]. CHGNet is a neural network force field pre-trained on the DFT labeled materials trajectory dataset [29]. It has shown been shown effective in a wide range of force field applications. Table 5.2 shows that, the performance of SOAP outperforms CHGNet in both the aspirin and the solvated amino acids datasets across different budgets. This is because that SOAP is a physics-based embedding method that does not require training. It is uniformly applicable to any configurations. While CHGNet is a neural network trained on a specific dataset. Although its developed to be universal, its performance is still biased by its training set.

Table 5.2: Performance (Force MAE, unit meV/A, lower ↓ the better) of DSP with different embedding methods for computing the task relevance.

Dataset	Embedding	Budget		
		$\lvert\tilde{\mathcal{D}}\rvert$	$2\lvert\tilde{\mathcal{D}}\rvert$	$5\lvert\tilde{\mathcal{D}}\rvert$
Aspirin	CHGNet	13.33	13.10	12.70
	SOAP	13.33	13.03	12.63
Amino acids	CHGNet	32.80	31.33	32.10
	SOAP	29.53	28.40	27.90

5.6 Conclusion

In this dissertation, we show that further pre-training MLFFs using task specific subsets effectively adapts them to the domain of the task and improves their accuracy on the task. We introduce data selection pre-training (DSP), a novel domain adaptive pre-training method that does not require additional domain datasets. DSP automatically selects a task specific subset from the pre-training dataset for adaptive training. We formulate the data selection problem as a budget constrained linear programming (LP) problem. We prove that the LP problem can be efficiently solved by selecting the configurations with the topK relevance scores. We evaluate DSP on 12 benchmark tasks. Experimental results show that DSP improves the pre-trained MLFFs and outperforms further pre-training on the entire pre-training dataset.

Chapter 6

Conclusion

This dissertation presents four advanced deep learning methods designed to accelerate scientific discovery in the fields of chemistry and materials science. These methods address three critical challenges in applying deep learning to these fields: 1) the structural complexity of molecules and materials, 2) the multi-modality of molecular data, and 3) the high cost of labeling.

In Chapter 2, we introduce a novel graph representation of molecules, heterogeneous molecular graph (HMG) in which nodes and edges are of various types, to model many-body interactions. HMGs have the potential to carry complex geometric information. To leverage the rich information stored in HMGs for chemical prediction problems, we build heterogeneous molecular graph neural networks (HMGNN) on the basis of a neural message passing scheme. HMGNN incorporates global molecule representations and an attention mechanism into the prediction process.

In Chapter 3, we propose a two-stage text-retrieval system to retrieve text from scientific literature to augment molecular property prediction (RTMol). To bridge the modality gap between molecule structures and text, RTMol uses the International Union of Pure and Applied Chemistry (IUPAC) name of molecules as the index and the BM25 algorithm to retrieve molecule related documents from the knowledge corpus. It then learns to re-rank and augment structure-based prediction with the retrieved documents.

In Chapter 4, we propose two generic strategies that take advantage of unlabeled training instances and computationally efficient labeling methods, empirical interatomic potentials (EIP), to increase the generalizability of deep neural networks. The first

strategy, based on weakly supervised learning, trains an auxiliary classifier on EIPs and selects the best-performing EIP to generate energies to supplement the ground-truth DFT energies in training the neural network. The second strategy, based on transfer learning, first pre-trains the neural network on a large set of easily obtainable EIP energies, and then fine-tunes it on ground-truth DFT energies.

In Chapter 5, we improve the performance of pre-trained machine learning force fields (MLFF) on downstream tasks by further pre-training them on domain adaptive datasets. In order to avoid the labeling cost of curating additional domain datasets, we propose data selection pre-training (DSP) that selects a task specific subset from the pre-training dataset and further pre-trains the MLFF on the subset. We formulate the data selection problem as a budget constrained linear programming problem and prove that its solution can be obtained efficiently via ranking the relevance scores.

In the past few years, deep learning has revolutionized many fields such as computer vision and natural language processing. However, its development in chemistry and materials science has lagged behind. Deep learning holds immense potential to transform scientific discovery in these fields, enabling us to understand the universe more rapidly, develop superior materials, and swiftly discover new drugs for previously incurable diseases. This dissertation advances the frontier by making deep learning more accessible to the fields of chemistry and materials science, thereby accelerating the pace of innovation and discovery.

References

[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

[2] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

[3] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in Neural Information Processing Systems*, pages 14510–14519, 2019.

[4] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[6] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87:184115, May 2013. doi: 10.1103/PhysRevB.87.184115. URL `https://link.aps.org/doi/10.1103/PhysRevB.87.184115`.

[7] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B—Condensed Matter and Materials Physics*, 87 (18):184115, 2013.

[8] Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816, 2017.

[9] Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=YPpSngE-ZU`.

[10] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):1–11, 2022.

[11] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

[12] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[13] Jörg Behler. Four generations of high-dimensional neural network potentials. *Chemical Reviews*, 121(16):10037–10072, 2021.

[14] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2838, 2015.

[15] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, 2019.

[16] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision*, 19(1):57–91, 1996.

[17] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[18] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[19] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[20] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021.

[21] Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[23] Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T Unke, Adil Kabylda, Huziel E

Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023.

[24] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[25] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[26] Ture Damhus, RM Hartshorn, and AT Hutton. Nomenclature of inorganic chemistry: Iupac recommendations 2005. *Chemistry International*, 2005.

[27] George B Dantzig. Linear programming. *Operations research*, 50(1):42–47, 2002.

[28] James P Darby, James R Kermode, and Gábor Csányi. Compressing local atomic neighbourhood descriptors. *npj Computational Materials*, 8(1):166, 2022.

[29] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[33] Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

[34] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

[35] Matthew J Elrod and Richard J Saykally. Many-body effects in intermolecular forces. *Chemical reviews*, 94(7):1975–1997, 1994.

[36] Felix A Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S Schoenholz, George E Dahl, Oriol Vinyals, Steven Kearnes, Patrick F Riley, and O Anatole von Lilienfeld. Machine learning prediction errors better than dft accuracy. *arXiv preprint arXiv:1702.05532*, 2017.

[37] Henri A Favre and Warren H Powell. *Nomenclature of organic chemistry: IUPAC recommendations and preferred names 2013*. Royal Society of Chemistry, 2013.

[38] John Fox and Sanford Weisberg. Robust regression. *An R and S-Plus companion to applied regression*, 91, 2002.

[39] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5): 845–869, 2013.

[40] Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *arXiv preprint arXiv:2210.07237*, 2022.

[41] Francisco-Javier Gamo, Laura M Sanz, Jaume Vidal, Cristina De Cozar, Emilio Alvarez, Jose-Luis Lavandera, Dana E Vanderwall, Darren VS Green, Vinod Kumar, Samiul Hasan, et al. Thousands of chemical starting points for antimalarial lead identification. *Nature*, 465(7296):305–310, 2010.

[42] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[43] Wei Gao, Lu Wang, Zhi-Hua Zhou, et al. Risk minimization in the presence of label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[44] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In *Machine Learning for Molecules Workshop, NeurIPS*, 2020.

[45] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

[46] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6790–6802. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/35cf8659cfcb13224cbd47863a34fc58-Paper.pdf`.

[47] Johannes Gasteiger, Muhammed Shuaibi, Anuroop Sriram, Stephan Günnemann, Zachary Ward Ulissi, C. Lawrence Zitnick, and Abhishek Das. Gemnet-OC: Developing graph neural networks for large and diverse molecular simulation datasets. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=u8tvSxm4Bs`.

[48] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[49] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[50] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[51] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[52] Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. *arXiv preprint arXiv:2404.07177*, 2024.

[53] Zhihui Guo, Pramod Sharma, Andy Martinez, Liang Du, and Robin Abraham. Multilingual molecular representation learning via contrastive pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3441–3453, 2022.

[54] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[55] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020.

[56] Johannes Hachmann, Roberto Olivares-Amaya, Sule Atahan-Evrenk, Carlos Amador-Bedolla, Roel S Sánchez-Carrera, Aryeh Gold-Parker, Leslie Vogt,

Anna M Brockway, and Alán Aspuru-Guzik. The harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011.

[57] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[58] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[61] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[62] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[63] Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

[64] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

[65] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

[66] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

[67] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations. *arXiv preprint arXiv:2103.01436*, 2021.

[68] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[69] Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

[70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[71] Peter Bjørn Jørgensen, Karsten Wedel Jacobsen, and Mikkel N Schmidt. Neural message passing with edge updates for predicting properties of molecules and materials. *arXiv preprint arXiv:1806.03146*, 2018.

[72] Maria Kalantzi and George Karypis. Position-based hash embeddings for scaling graph neural networks. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 779–789. IEEE, 2021.

[73] Daniel S Karls. *Transferability of empirical potentials and the Knowledgebase of Interatomic Models (KIM)*. PhD thesis, University of Minnesota, 2016.

[74] Efthimios Kaxiras. *Atomic and Electronic Structure of Solids*. Cambridge University Press, 2003. doi: 10.1017/CBO9780511755545.

[75] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

[76] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[77] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[78] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=B1eWbxStPH`.

[79] Adeesh Kolluru, Nima Shoghi, Muhammed Shuaibi, Siddharth Goyal, Abhishek Das, C Lawrence Zitnick, and Zachary Ulissi. Transfer learning using attentions across atomic systems with graph neural networks (taag). *The Journal of Chemical Physics*, 156(18), 2022.

[80] Risi Kondor, Truong Son Hy, Horace Pan, Brandon M. Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs, 2018. URL `https://openreview.net/forum?id=S1TgE7WR-`.

[81] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[82] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

[83] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, 2019.

[84] Kangming Li, Daniel Persaud, Kamal Choudhary, Brian DeCost, Michael Greenwood, and Jason Hattrick-Simpers. On the redundancy in large material datasets: efficient and robust learning with less data. *arXiv preprint arXiv:2304.13076*, 2023.

[85] Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in Bioinformatics*, 22(6):bbab109, 2021.

[86] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2022.

[87] Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

[88] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

[89] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2021.

[90] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789*, 2022.

[91] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[92] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A

robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[93] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.447. URL `https://www.aclweb.org/anthology/2020.acl-main.447`.

[94] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6959–6969, 2022.

[95] Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1052–1060, 2019.

[96] Nicholas Lubbers, Justin S Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *The Journal of chemical physics*, 148(24):241715, 2018.

[97] Saurav Manchanda, Mohit Sharma, and George Karypis. Distant-supervised slot-filling for e-commerce queries. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 677–686, 2021. doi: 10.1109/BigData52589.2021.9671825.

[98] Costas Mavromatis and George Karypis. Graph infoclust: Maximizing coarse-grain mutual information in graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 541–553. Springer, 2021.

[99] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

[100] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[101] Robert G Parr. Density functional theory of atoms and molecules. In *Horizons of Quantum Chemistry*, pages 5–15. Springer, 1980.

[102] Ji Qi, Tsz Wai Ko, Brandon C Wood, Tuan Anh Pham, and Shyue Ping Ong. Robust training of machine learning interatomic potentials with dimensionality reduction and stratified sampling. *npj Computational Materials*, 10(1):43, 2024.

[103] Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, xu Zhao Pan, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=C61sk5LsK6`.

[104] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[105] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[106] Kohulan Rajan, Achim Zielesny, and Christoph Steinbeck. Stout: Smiles to iupac names using neural machine translation. *Journal of Cheminformatics*, 13(1):1–14, 2021.

[107] Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021.

[108] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1:140022, 2014.

[109] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ryQu7f-RZ`.

[110] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2020.

[111] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389, 2009.

[112] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

[113] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

[114] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

[115] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[116] Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient llms. *arXiv preprint arXiv:2402.09668*, 2024.

[117] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pages 991–1001, 2017.

[118] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[119] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[120] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1):1–8, 2017.

[121] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[122] Nima Shoghi, Adeesh Kolluru, John R Kitchin, Zachary Ward Ulissi, C Lawrence Zitnick, and Brandon M Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction. In *The Twelfth International Conference on Learning Representations*, 2023.

[123] Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 492–500. IEEE, 2020.

[124] Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 492–500. IEEE, 2020.

[125] Zeren Shui, Daniel Karls, Mingjian Wen, Ellad Tadmor, George Karypis, et al. Injecting domain knowledge from empirical interatomic potentials to neural networks for predicting material properties. *Advances in Neural Information Processing Systems*, 35:14839–14851, 2022.

[126] Justin S Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx

and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific data*, 7(1):134, 2020.

[127] Justin S Smith, Benjamin Nebgen, Nithin Mathew, Jie Chen, Nicholas Lubbers, Leonid Burakovsky, Sergei Tretiak, Hai Ah Nam, Timothy Germann, Saryu Fensin, et al. Automated discovery of a robust interatomic potential for aluminum. *Nature communications*, 12(1):1–13, 2021. doi: 10.1038/s41467-021-21376-0.

[128] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[129] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[130] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pages 20479–20502. PMLR, 2022.

[131] Frank H Stillinger and Thomas A Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.

[132] Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*, 2022.

[133] Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. Mocl: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3585–3594, 2021.

[134] Ellad B Tadmor and Ronald E Miller. *Modeling materials: continuum, atomistic and multiscale techniques*. Cambridge University Press, 2011.

[135] Ellad B. Tadmor, Ryan S. Elliott, James P. Sethna, Ronald E. Miller, and Chandler A. Becker. The potential of atomistic simulations and the Knowledgebase of Interatomic Models. *JOM*, 63(7):17, July 2011. doi: 10.1007/s11837-011-0102-6. URL `http://dx.doi.org/10.1007/s11837-011-0102-6`.

[136] Ellad B. Tadmor, Ryan S. Elliott, Simon R. Phillpot, and Susan B. Sinnott. NSF cyberinfrastructures: A new paradigm for advancing materials simulation. *COSSMS*, 17(6):298–304, December 2013. doi: 10.1016/j.cossms.2013.10.004. URL `http://dx.doi.org/10.1016/j.cossms.2013.10.004`.

[137] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36, 2024.

[138] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[139] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[140] Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36, 2024.

[141] Oliver T Unke and Markus Meuwly. Physnet: a neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6):3678–3693, 2019.

[142] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[143] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 839–847, 2017.

[144] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[145] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

[146] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. *arXiv preprint arXiv:1909.01315*, 2019.

[147] YiNan Wang, LinFeng Zhang, Ben Xu, XiaoYang Wang, and Han Wang. A generalizable machine learning potential of ag–au nanoalloys and its application to surface reconstruction, segregation and diffusion. *Modelling and Simulation in Materials Science and Engineering*, 30(2):025003, 2021.

[148] Zichao Wang, Weili Nie, Zhuoran Qiao, Chaowei Xiao, Richard Baraniuk, and Anima Anandkumar. Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126*, 2022.

[149] Mingjian Wen and Ellad B Tadmor. Uncertainty quantification in molecular simulations with dropout neural network potentials. *npj Comput. Mater.*, 6(1):124, 2020. doi: 10.1038/s41524-020-00390-8.

[150] Mingjian Wen, Stephen Carr, Shiang Fang, Efthimios Kaxiras, and Ellad B. Tadmor. Dihedral-angle-corrected registry-dependent interlayer potential for multilayer graphene structures. *Phys. Rev. B*, 98(23):235404, dec 2018. doi: 10.1103/physrevb.98.235404.

[151] Mingjian Wen, Yaser Afshar, Ryan S Elliott, and Ellad B Tadmor. KLIFF: A framework to develop physics-based and machine learning interatomic potentials. *Computer Physics Communications*, 272:108218, 2022.

[152] Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*, 2024.

[153] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[154] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

[155] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

[156] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[157] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

[158] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.120.145301`.

[159] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[160] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

[161] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558. PMLR, 2021.

[162] Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

[163] Kun Yao, John E Herr, and John Parkhill. The many-body expansion combined with neural networks. *The Journal of chemical physics*, 146(1):014106, 2017.

[164] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. *arXiv preprint arXiv:2211.12561*, 2022.

[165] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

[166] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[167] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):1–11, 2022.

[168] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*, 2023.

[169] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.

[170] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.

[171] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[172] Da Zheng, Minjie Wang, Quan Gan, Zheng Zhang, and George Karypis. Learning graph neural networks with deep graph library. In *Companion Proceedings of the Web Conference 2020*, pages 305–306, 2020.

[173] Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=QwKvL6wC8Yi`.

[174] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

[175] Jinhua Zhu, Yingce Xia, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Dual-view molecule pre-training. *arXiv preprint arXiv:2106.10234*, 2021.

[176] Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Unified 2d and 3d pre-training of molecular representations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2626–2636, 2022.

[177] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[178] Larry Zitnick, Abhishek Das, Adeesh Kolluru, Janice Lan, Muhammed Shuaibi, Anuroop Sriram, Zachary Ulissi, and Brandon Wood. Spherical channels for modeling atomic interactions. *Advances in Neural Information Processing Systems*, 35: 8054–8067, 2022.