GRAPH NEURAL NETWORKS FOR COMPUTATIONAL CHEMISTRY

An Undergraduate Research Scholars Thesis

by

BORA OZTEKIN

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Shuiwang Ji

May 2022

Major: Computer Science

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Bora Oztekin, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Graph Neural Networks for Computational Chemistry

Bora Oztekin
Department of Computer Science & Engineering
Texas A&M University

Research Faculty Advisor: Dr. Shuiwang Ji
Department of Computer Science & Engineering
Texas A&M University

Graph Neural Networks are behind many pharmacological breakthroughs due to their innate ability to learn structural properties of molecules and accelerate high-throughput screening for favorable characteristics that could serve as a treatment or cure to a disease. Much of the world's natural data, such as social networks and molecules, can be represented in the form of graphs. However, advancements in graph-based problems like chemistry have been lacking because graphs are a form of non-Euclidean data, and encoding them into a format that is compatible with deep learning is considerably more challenging. This thesis seeks to understand and benchmark the techniques used to preserve the structure and properties of a graph in the encoded form. Specifically, characteristics of a graph that are distinct in the graph form should be distinct in the encoded form. Preserving both the expressiveness and the distinctness of the encoded graph is a challenging task that has received a lot of attention in geometric deep learning. This work evaluates and compares various graph neural network methods on a large public dataset to quantify the expressive power of more detailed graph neural networks that consider dihedrals and bond information, for example. It becomes evident that simply constructing homogeneous graphs of nodes and edges is insufficient.

ACKNOWLEDGMENTS

Contributors

I would like to thank my faculty advisor, Dr. Shuiwang Ji, and my friends and mentors in the Data Integration, Visualization, and Exploration (DIVE) Lab for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my family for their encouragement and to my friends for their patience and support.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This project was supported in part by National Science Foundation grant DBI-1922969.

NOMENCLATURE

GNN	Graph Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
GCN	Graph Convolutional Network
FCN	Fully-Connected Network
DFT	Density Functional Theory
MAE	Mean Absolute Error

1. INTRODUCTION

Graph Neural Networks (GNNs) are behind many pharmacological breakthroughs due to their innate ability to learn the structural properties of molecules and accelerate high-throughput screening for favorable characteristics that could serve as a treatment or cure to a disease.

Advances in deep learning [1], a technique to extract features from data and make predictions, have catalyzed progress in fields like machine translation and computer vision. However, conventional deep learning methods require highly-structured data with fixed representations, such as an image with each pixel neighbored by eight other pixels. It is challenging to represent less-structured data, such as social networks and molecules, in a compatible form using conventional methods– critical information is almost always lost, and deep learning becomes ineffective. Much of this natural data can be natively represented in the form of graphs without information loss, as shown in Figure 1.1.

Graphs are a collection of entities known as nodes, and they are connected by edges, which indicate some association between the two nodes. A graph could be a social network, with individuals as nodes and friendships denoted by edges. However, there are also more complicated examples. For instance, H_2O, or water, can be represented with three nodes and two edges, each connecting hydrogen to oxygen. This can be seen in Figure 1.2.

Advancements are allowing deep learning to represent graphs, but difficulties remain in preserving both the expressiveness and the distinctness of the encoded graph. This is because graphs are a form of non-Euclidean data, and encoding them into a format compatible with deep learning is considerably more challenging without the assumptions made in Euclidean spaces, such as that of a fixed neighborhood. Solutions to such tasks have been propelled by geometric deep learning [2]. This thesis seeks to understand, compare, and propose the techniques used to preserve the structure and properties of a graph in the encoded form. This includes an analysis of methods, datasets, and training procedures. Further work on GNNs for Computational Chemistry can be

found at https://boraoztekin.com/ug_thesis.

1.1 Graph Neural Networks (GNNs)

Information propagation and aggregation across nodes enables Graph Neural Networks (GNNs) to learn and infer graph properties. Namely, message passing [3], an information-sharing technique that contextualizes nodes by propagating information from its neighbors, is used for its expressiveness. When the locality assumptions of a graph are met, message passing can be used to learn information about unknown nodes using those that have property information.

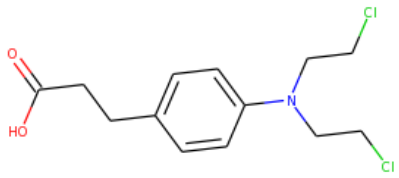

Figure 1.1: *Norchlorambucil*, a molecule from ZINC [4] represented as a structural heterogeneous graph. Non-Euclidean data such as graphs challenge the assumptions and methods of deep learning on Euclidean data.

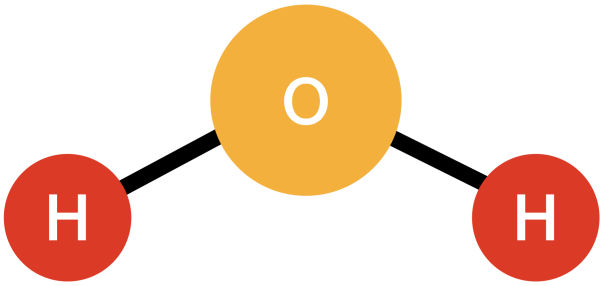

Figure 1.2: A water molecule consisting of two hydrogens and one oxygen, with a bent bond.

1.2 Graphs in Chemistry

Unlike in a social network, the arrangement of the nodes (atoms) in chemical graphs plays an important role. For example, the structure of the water molecule shown in Figure 1.2 is 'bent', and there is a specific angle at which the molecule is configured due to the lone electrons on the oxygen. This is a characteristic that would not be represented in a social network graph. Most GNN developments have been focused on nodes and edges, and not characteristics like the distances between nodes (since friendship has no 'distance', for example) or the angles made by any three nodes.

The objective of GNNs on chemical problems is typically to predict the properties of new molecules using labeled training data. The training data is labeled using a procedure known as Density Functional Theory (DFT) [5]. DFT is a simulation tool to approximate chemical processes, and it has vastly accelerated exploration in the chemical space. Despite the growth of computing resources with respect to Moore's law, the efficiency of the DFT calculation has become a bottleneck in searching the chemical space. An approach that has been promising in recent years is to use DFT calculations as training data for machine learning (ML) models. This would allow ML models to make inferences on unlabeled data at a much faster rate than DFT. To study the effectiveness of this method, it is necessary to define error metrics to compare DFT and ML approximations. This can include Mean Absolute Error (MAE) or Root Mean Square Error (RMSE),

among others. These metrics are particularly important in chemistry because unlike in a social network, for example, there are quantifiable ground truths derived via physical approximations. Monitoring the performance of the models relative to the ground truths is a guiding indicator in the development of a GNN.

1.2.1 Graph Convolutions and Message Passing

Graphs use edges to connect nodes with meaningful relationships. Therefore, a center node's neighbors (nodes adjacent to a given node) often contain valuable information about the properties of the center node. This can be considered as a generalization of the fixed-neighborhood image, which can be represented as a graph like the one shown in Figure 1.3. Graph Convolutional Networks (GCN) [6] use weight sharing to learn features from neighboring nodes. Given that the neighboring nodes are connected due to some sort of meaningful relationship to the center node, this convolutional process adds meaningful context, providing locality to the feature vector of the center node. GCNs implicitly assume locality and equal importance of self-connections as compared to edges to neighboring nodes.

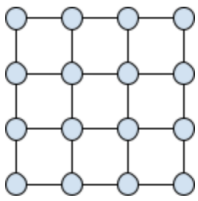

Figure 1.3: A fixed-neighbor graph with Euclidean structure representing the pixels of an image.

Several techniques of message passing include the GCN and Graph Attention Network (GAT) [7]. Furthermore, Neural Message Passing for Quantum Chemistry [3] uses neighborhood aggregation techniques to share information. These concepts have been extended by methods such as Directional Message Passing for Neural Networks (DimeNet) [8].

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right) \quad \text{(Eq. 1.1)}$$

Eq. 1.1 is the **Neighborhood Aggregation Function** used for information sharing in Graph Convolutions.

The aggregation function above is an update function that takes information from neighboring nodes $j \in \mathcal{N}_i$ and uses their feature vectors to update the feature vector $\mathbf{x}_i$. Many methods are built around this framework.

1.3 Related Work

1.3.1 Crystal Graph Convolutional Neural Networks

Crystal Graph Convolutional Neural Networks (CGCNNs) [9] represent crystal structures by constructing a graph that encodes atoms and bonds as nodes and edges, respectively. CGCNN then builds a CNN on top of the graph to interact the nodes and extract properties by training with DFT-calculated data. The convolutional layers iteratively update the atom feature vector by sharing information with surrounding nodes and edges using a non-linear graph convolution function that concatenates information from the convolution weight matrix and the self weight matrices. The entire network of convolutional, pooling, and dense layers allows CGCNN to extract structural differences based on edges to discover relationships between structures and properties.

1.3.2 Continuous-filter Convolutional Neural Network

SchNet [10] builds on top of previous GNNs to use the relative distance between atoms to learn a representation for the prediction of molecular energies. The model also predicts the forces acting on each atom because it is not clear how to obtain equilibrium conformers without optimizing atom positions.

SchNet's significant contribution extends the idea of convolutional layers [11] beyond the Euclidean domain. Unlike image recognition, which is a popular application of CNNs, graph data do not have the same properties, such as a fixed number of neighbors, or neighbors at a fixed distance. For example, a pixel in a 2D image would have eight equidistant neighbors, whereas an

atom may have any number of neighbors at a variety of distances. Thus, using traditional convolutional layers for graphs would cause important information to be lost when discretized by the convolutional filters. To remedy this problem, SchNet introduces the continuous filter convolutional layer in $\mathbb{R}^3$, which is generated by mapping from a position to corresponding filter values. These filter values are applied to a node's neighbors, and then aggregated, providing generalization to a variable number of atoms, as well as scalability due to weight sharing.

In addition to the architectural contribution made by SchNet, the work implements more robust benchmarking by considering datasets beyond QM9 [12, 13], which only contains equilibrium molecules. Specifically, the model is evaluated against MD17 [14], which introduces conformational degrees of freedom, and ISO17, combining chemical and conformational degrees of freedom in a dataset of molecules with a fixed composition of atoms, but different valid arrangements.

SchNet surpasses other models [15] in scalability due to weight sharing, but it does not completely represent the arrangement of a molecule in a way that preserves distinctness after the encoding process. Namely, simply encoding relative distances leaves room for ambiguities due to the large span of arrangements for a molecule with the same nodes and edges, but different angles.

1.3.3 Directional Message Passing Neural Networks (DimeNet)

DimeNet [8] introduces directional message passing, allowing GNNs to model the directions to neighboring atoms instead of just their distances. Previous GNNs do not consider spatial direction, but rather just distance. This can leave behind chemically-significant information such as angles. In particular, DimeNet embeds messages passed between atoms rather than the atoms themselves. This is important because molecules that could not previously be distinguished due to cutoff distances or other constraints can now be encoded to distinct representations.

Understanding how the graphs that are used as inputs to GNNs are constructed uncovers a limitation that must be overcome before using directional information. Namely, if there exist two different molecules that, for all atoms in each molecule, their neighborhoods are the same, a distance-based GNN would be unable to distinguish between the two. As shown in the DimeNet paper, a Cyclohexane and two Cyclopropane molecules with stretched bonds will bear the same

encoding. From a local neighborhood perspective, other models do not use angular information, and they cannot see that these are in fact two different structures.

Although modeling the directions rather than the distances would typically violate the rules of symmetry and invariances because rotations would change directions, the method uses group equivariance [16] to stay invariant to a transformation group.

DimeNet has since been succeeded by DimeNet++ [17], which models interactions using a Hadamard product and linear layers instead of a bilinear layer. This significantly reduces a combinatorial representation explosion.

1.3.4 Spherical Message Passing for 3D Graph Neural Networks (SphereNet)

SphereNet [18] proposes spherical message passing (SMP) as a scheme to realize the 3D graph network (3DGN). It accomplishes this by using the spherical coordinate system (SCS) to encode 3D information in a relative manner that yields predictions that are invariant to translation and rotation of input graphs.

Despite computing an additional angle between planes, SphereNet is comparable to other models in complexity due to the redundancies eliminated by deviating from the Cartesian coordinate system. It is easy to convert positions from the (x, y, z) plane to the (ρ, θ, ϕ), and SphereNet is able to switch to the SCS and preserve translational and rotational invariances because it uses aggregation functions that can be adapted to generate relative information that moves with the molecule as it is shifted. For example, when updating node features, the aggregation function can be adapted to a radial basis function that computes distances between a node and the nodes that point to it. The model architecture in Figure 1.4 shows the interaction block that updates the messages.

The additional angle computed in SphereNet is known as the torsion angle, which can be described as the angle between two planes. There are many instances, such as in H_2O_2, where two molecules can be represented with identical bond distances and angles, but with variability on a planar angle. Being able to represent the torsion angle would empower GNNs to recognize chirality, for example. As expressed in the objective of GNNs for computational chemistry, a successful

model must be able to distinguish different representations in their encodings. SphereNet accomplishes this in SMP by computing the azimuthal angle between the neighbors of sender nodes, and using a time-independent Schrödinger equation that depends on the electron state being expressed as a function of locations. Specifically, a spherical harmonic function in the Schrödinger equation takes the bond angle and torsion angle into account.

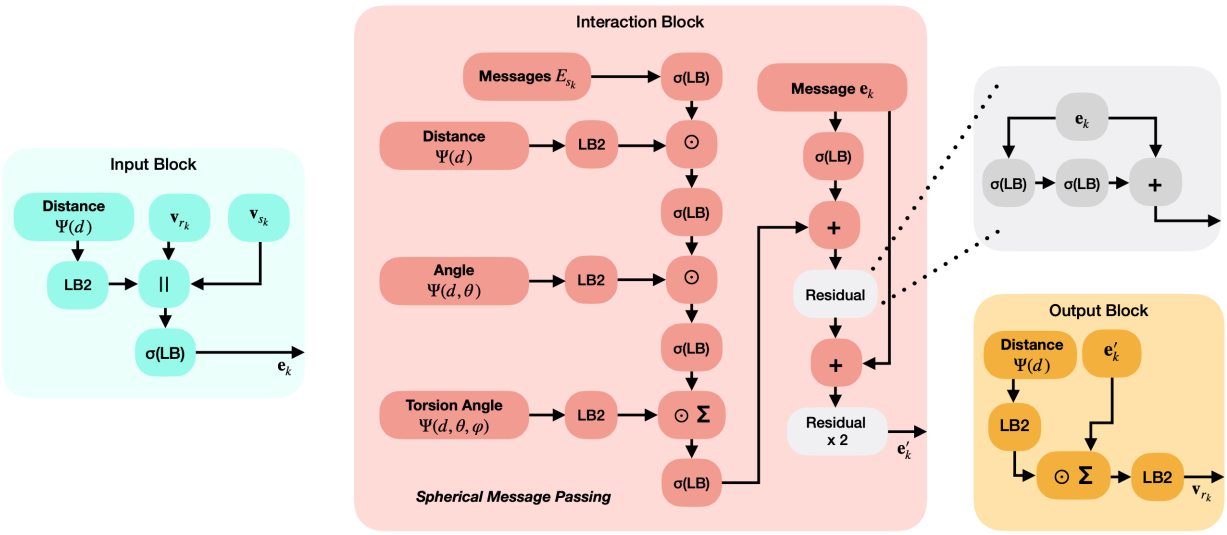

Figure 1.4: The architecture of SphereNet consists of an interaction block that considers distance, angle, and torsion angle.

Although there are other methods that can be used to fully describe the structure of a 3D molecule, they require accessing the neighbors of the neighbors, or the 2-hop neighborhood, of a reference node. This is computationally expensive at $O(nk^3)$. By using SMP, SphereNet stays within the 1-hop neighborhood, at a time complexity of $O(nk^2)$.

2. METHODS

According to The Chemical Space Project [19], the "drug-like" chemical space has been estimated at 10^{60}, which is clearly far too large for practical application. Domain experts have developed heuristics to reduce the practical span of the chemical space based on structural features and chemical properties, but this is a lengthy process that requires human intervention, and sometimes even the handcrafting of features. Furthermore, it is not guaranteed to be accurate. Rather, some of the latest advances in graph neural networks have eliminated the need for this type of intervention, thanks to models that are more expressive and able to distinguish molecules.

This process is made possible through a variety of techniques that leverage the physical rules of molecules and the mathematical rules of graphs. For example, models must account for chirality, since a chiral representation of a certain molecule may be toxic while another may be nontoxic.

Novel GNN architectures are constantly being developed to address challenges and limitations in chemistry applications. However, the performance of a technique is not limited solely to the model. In fact, the training datasets and techniques play a critical role in a GNN's ability to properly fit a model and make solid inferences. The overarching goal of making chemical representations more expressive and distinct is aided by these other factors. Specifically, a cherry-picked dataset that includes a small subset of a larger dataset may hinder a model's ability to classify molecules with higher novelty. Also, simple considerations like shuffling training data can have an impact on the learned model.

2.1 Model Architecture

Keeping in mind that the objective is to increase the expressiveness of the encoding of the graph, the model itself achieves the goal by including distinguishing information about the properties and structure of a graph. This includes taking steps like message passing to perform neighborhood aggregation.

$$m_v^{t+1} = \sum_{w \in N(v)} h_w^t \qquad \text{(Eq. 2.1)}$$

The message consists of a sum of the feature vectors of the neighbors $w \in \mathcal{N}(v)$.

$$h_v^{t+1} = \bigoplus (h_v, m_v^{t+1}) \qquad \text{(Eq. 2.2)}$$

The hidden state is updated using an aggregation function consisting of its prior state and the new message from Eq. 2.1.

Considering the existence of edges between nodes is typically sufficient to describe a social network of friends, for example, but in the context of chemistry, this would be insufficient. This is because it is not expressive enough to just know which atoms are connected by a bond, since this excludes information such as the atomic number or the type of bond. Furthermore, it would also be more expressive to include information about the angle of the bond. In other words, chemistry introduces the need for graphs whose nodes are not all the same type: heterogeneous graphs. Clearly, there is plenty of information contained in natural graphs, which are known as networks, that must be considered to create an encoding that sufficiently distinguishes characteristics.

Although 3D graph neural networks are expected to be more performant than their 2D counterparts, they present their own set of logistical and theoretical challenges. Namely, 3D GNNs are computationally intensive, especially with larger molecules where there are many angles. This issue compounds when working with dynamic graphs, which change over time with respect to a state (an example of this is a quantum state relaxation). To keep a focused direction, this thesis will not examine molecular dynamics in-depth.

2.1.1 Graph Invariance and Equivariance

Since it is the goal of a GNN to expressively and distinctly represent the natural graph, or network, that it is encoding, it is paramount that GNNs for chemistry be able to learn domain-unique features. However, a new challenge is posed when considering the distances or angles, for example. For the weights to be properly and efficiently learned in a GNN, there needs to be some

kind of procedure that canonically represents molecules that are viewed from a different frame of reference. For example, looking at water from the side and the front would express the absolute positions in different places, but it is in fact the same molecule. A solid GNN would encode both frames of reference to the same representation. This phenomena in particular is known as rotational invariance. Intuitively, we also seek translational invariance, as shown in Eq. 2.3.

$$f(\tau x) = f(x) \tag{Eq. 2.3}$$

$$f(\tau x) = \tau' f(x) \tag{Eq. 2.4}$$

Equivariance describes any operation applied to x being accordingly applied in the encoded representation. So, if there is a 180 degree rotation in the input, the encoding will vary equivalently, thus establishing rotational equivariance, as in Eq. 2.4.

There are striking similarities between Convolutional Neural Networks (CNNs) and GNNs in terms of how they propagate information. Unlike a fully-connected network (FCN), a CNN has far fewer connections, and utilizes weight sharing. This is helpful in CNNs because weight sharing introduces translational equivariance: if an object is shifted elsewhere on the grid of its image, it will be detected regardless of its position. Similar concepts apply when extending the idea of weight sharing and locality to graph neural networks, although there are fundamental differences in the architectures due to the varying number of neighbors and distances between nodes. Furthermore, without weight sharing, the number of parameters is linearly related to the number of nodes, which can be quite inefficient in larger graphs.

Overall, the design of 3D GNNs evolved from considering the distances between atoms to considering the angle produced by three different atoms, to evaluating the torsion angle produced by the planes of four atoms. Clearly, 3D GNNs consider information from more atoms to create a more localized and distinct mapping. A future direction for 3D GNNs is to consider a cloud of nodes and embed information about the cloud globally. Works in other non-3D specific GNN

areas have attempted to address this by identifying and distinguishing functional groups, which are groups of atoms that serve a specific purpose within a molecule.

2.2 Data Selection

Current models are typically benchmarked using datasets that contain under one million molecules. Considering the highly expressive nature of chemical graphs and the number of parameters in the latest 3D GNN models, it is hypothesized that model performance will improve considerably using a larger dataset. This thesis benchmarks current state-of-the-art models against PubChem PM6 (PM6) [20], a consolidation of datasets from institutions around the world. It contains over 90 million molecular compounds, and is maintained by the National Institutes of Health (NIH).

According to the authors, the PubChemQC PM6 datasets are the largest datasets developed by semiempirical quantum chemical calculations. It is important to note that while DFT is more accurate than semiempirical calculations, it is considerably more expensive to compute, and it has therefore been used mostly for subset datasets. The authors remark that the PM6 method gives fairly good geometries for molecules, which include proteins. The results are also valuable for estimating electron structures. When evaluating the results of models trained using PM6 and other methods, it is important to reference how these datasets were developed since all performance is relative to the ground truth, which is indeed an approximation in all of these datasets.

2.3 Data Preprocessing

Given the variety of benchmark models and dataset sizes, data preprocessing is not a trivial task, and it can be vastly different across models. Since this work seeks to unify and compare several of such models, it is important to develop a standard from which the input data are generated. Specifically, the PubChem PM6 dataset is much larger than QM9. The Linux filesystem, Python, and input/output parallelism limits all pose challenges for such a large dataset with over 100 million molecules.

To address these limitations, this work utilizes PyTorch Geometric Dataset objects. This is in contrast to InMemoryDataset objects, which, as the name implies, are stored in memory rather

than read from a file. Each block of up to 25,000 molecules is independently processed and the relevant tensors are stored as a PyTorch serialized file object. In fact, the original file hierarchy structure of the PM6 raw dataset is quite slow to process due to each molecule being in its own folder. Due to how the Linux filesystem treats directories, this causes a considerable slowdown, so molecules are extracted in groups of 25,000 to a temporary folder, where the relevant data are obtained.

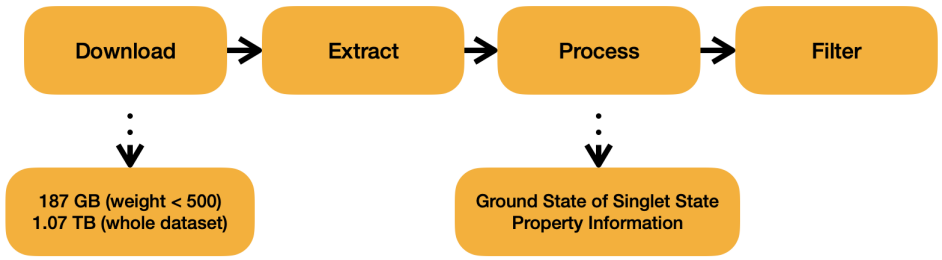

Figure 2.1: Data handling pipeline from raw files to molecular graphs.

As shown in Figure 2.1, the data pipeline extends beyond simply organizing the data into a database because there are several models to benchmark, and each has slightly different input requirements. However, SchNet, DimeNet, and SphereNet have been reimplemented within Dive Into Graphs (DIG) [21], which unifies the model's inputs and outputs so benchmarking across models can be done easily with the same datasets.

First, the datasets are downloaded from the PubChemQC PM6 Datasets webpage. Then, a subset of 120,000 molecules is extracted. The reason for selecting this size is parity with the QM9 dataset. Experimental trials in this study use 100,000 molecules for training, and 10,000 each for validation and testing. Then, the processing step is invoked for each molecule.

The process involves extracting a list of atoms and their coordinates, followed by the molecular properties, such as total dipole moment, highest-energy occupied molecular orbital (HOMO), lowest-energy unoccupied orbital (LUMO), gap (HOMO-LUMO), total energy, and enthalpy. Fi-

nally, molecules that are missing property files or have single nodes (and therefore no message passing update) are filtered.

Another considered alternative to processing many XYZ files was to use Spatial Data Files (SDFs), which are a collection of XYZ files compressed into one. Unfortunately, this method poses two new challenges: it is not standardized across many of the benchmarked models, and indexing errors are much more likely to occur in SDF files due to the implicit sequential ordering of molecules within the file. With tens of millions of molecules in the dataset being chemically invalid or undesirable for experimentation due to missing information or a very small structure, this could pose serious inconsistencies.

We have established that there is a plethora of information to be considered when encoding a graph, especially those involving chemistry. Beyond a strong encoding, it is important to distinguish the techniques that can be used to propagate information amongst nodes and edges. In particular, graph neural networks use various forms of message passing to share information along the nodes and edges of a GNN. This technique works because of the assumption that links connect entities that are supposed to be related. This explains why it is so important to pick criteria like cutoff distances systematically when synthetically adding edges to a set of atoms that do not have determined edges, for example.

2.4 Data Target Parity

This work uses the Mean Absolute Error (MAE) as the error metric to compare the performance of a model against the ground truth.

$$\frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \qquad \text{(Eq 2.5)}$$

The predicted value is represented by $\hat{y}_i$, and y_i denotes the ground truth. The number of samples is n.

An important part of ensuring the correctness of the MAEs is making sure that the targets in the dataset have similar distributions. As shown in Figure 2.2, the HOMO-LUMO (Gap) property

in both PM6 and QM9 have similar distributions and ranges of values.

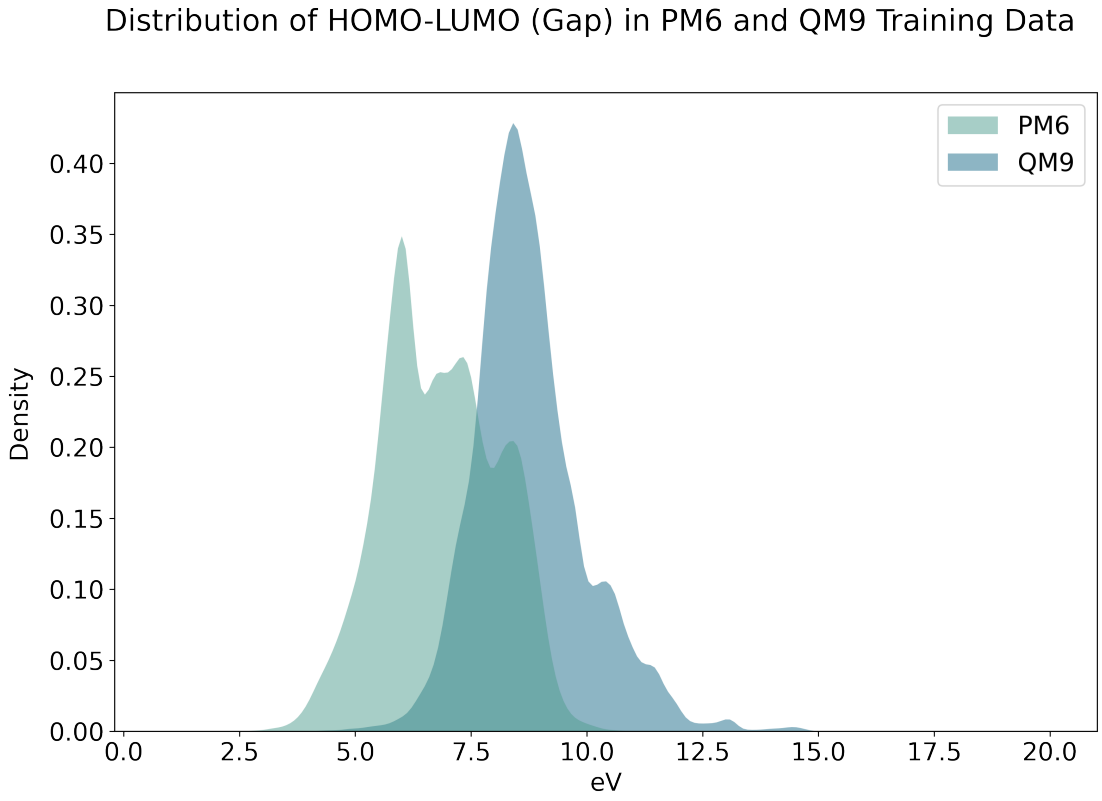

Figure 2.2: Distribution of HOMO-LUMO (Gap) in PM6 and QM9 Training Data

2.5 Hyperparameter Tuning

There are two kinds of graphs: structural and nonstructural. Structural graphs represent phenomena that naturally occur in graphs. This would include molecules or social networks. Non-structural graphs take data that are not ordinarily represented by networks and construct a graph, linking nodes based on some criteria like distance or similarity, such as in Figure 2.3.

Choosing the correct cutoff distance for when two neighboring atoms are considered linked is a key influencer of model performance. Therefore, hyperparameter tuning in the form of a grid search is conducted to find the values that optimize model performance. This is done for the cutoff

distance, number of layers, hidden channels, interaction embedding size, and output embedding size.

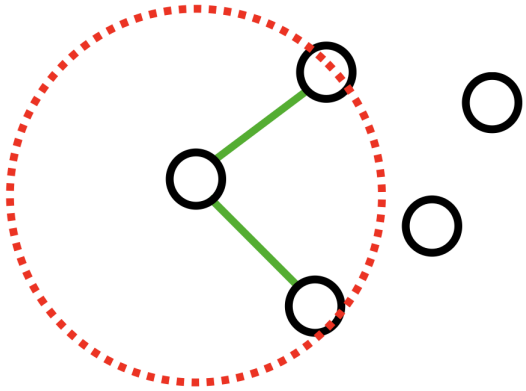

Figure 2.3: Using distance as a criterion to create edges between nodes.

2.6 Dataset Comparison

This thesis seeks to highlight that in addition to the expressiveness of the models that are used, it is paramount to consider that GNN development is in part guided and motivated by the performance of models on various benchmark datasets. Therefore, using datasets that are representative of real problems is critical, especially for these methods to break through to industry applications.

Although smaller datasets have their practical advantages, namely through being easy to work with and oftentimes highly constrained, this does not represent the actual problem of drug discovery. Instead, these datasets should be used as milestones in the design process, with final testing being on a larger dataset like PM6, which spans a broader section of the chemical space than smaller datasets.

The common 3D GNN benchmark datasets can be seen in Table 2.1, and Hyperparameter settings for SphereNet are found in Table 2.2.

Table 2.1: 3D GNN Datasets

Name	**Size**	**Ground-Truth**	**Specific Subset**
PM6	91M	Semi-Empirical	No
QM9	134K	DFT	Up to 9 heavy atoms

Table 2.2: SphereNet PM6 and QM9 Hyperparameters

Hyperparameter	**QM9 Value**	**PM6 Value**
Cutoff	5.0	5.0
Layers	4	4
Hidden Channels	128	256
Interaction Embedding Size	64	128
Output Embedding Channels	256	256

3. RESULTS

3.1 Influential Factors

The results indicate that the PM6 dataset is quite different from QM9, not only in terms of the number of molecules, but also the types. This is to be expected since QM9 is a subset dataset that contains molecules sharing similar properties that make them easier for chemical property prediction. Furthermore, PM6 ground truth values were calculated using a semi-empirical process, rather than Density Functional Theory (DFT).

3.2 Comparative Tables

The **Validation** Mean Absolute Error (MAE) on **PM6** can be seen within Table 3.1:

Table 3.1: Validation MAE for PM6 Dataset

Property (Units)	**SchNet**	**DimeNet++**	**SphereNet**
Total Dipole Moment (D)	848	473	**418**
HOMO (meV)	177	141	**121**
LUMO (meV)	350	319	**274**
Gap (meV)	243	209	**170**
Enthalpy (H)	10	**3**	**3**

The **Validation** Mean Absolute Error (MAE) on **QM9** can be seen within Table 3.2:

Table 3.2: Validation MAE for QM9 Dataset

Property (Units)	**SchNet**	**DimeNet++**	**SphereNet**
Total Dipole Moment (D)	98	89	**62**
HOMO (meV)	73	77	**49**
LUMO (meV)	60	42	**40**
Gap (meV)	109	**79**	**79**
Enthalpy (H)	61	38	**33**

3.3 Analysis

The results demonstrate that QM9 outperforms PM6; however, rather than solely comparing the performance on the same model across datasets, it is important to compare different models using the same dataset as well. This is because a model's expressive powers are uniquely assessed by each dataset against which it is benchmarked.

Hyperparameter tuning made a measurable contribution to the model performance on PM6, with larger values generally yielding lower validation and testing MAEs. By adding parameters to the model, it is expected that the training MAE would be lower, but we confirm its benefit by noting that this effect is observed in validation and testing.

The training process over time provides valuable insights about the convergence of a model. In Figure 3.1, we observe that the validation and testing MAE are 0.156 and 0.154, respectively. By the last steps, there is plateauing that indicates that the errors are unlikely to improve with more steps. It is also important to note that the training MAE relative to the validation and testing MAE is in a place such that we can be confident that overfitting did not occur. We can make this claim because there is no situation where the training MAE is significantly smaller than the validation and testing MAE.

One noted benefit of PM6 is that in some properties, the margin between the value of the best model and the other ones is greater than in QM9. This observation is beneficial because it indicates that PM6 is able to better discern a more expressive model, which is in alignment with the key goal of increasing expressiveness and distinctiveness.

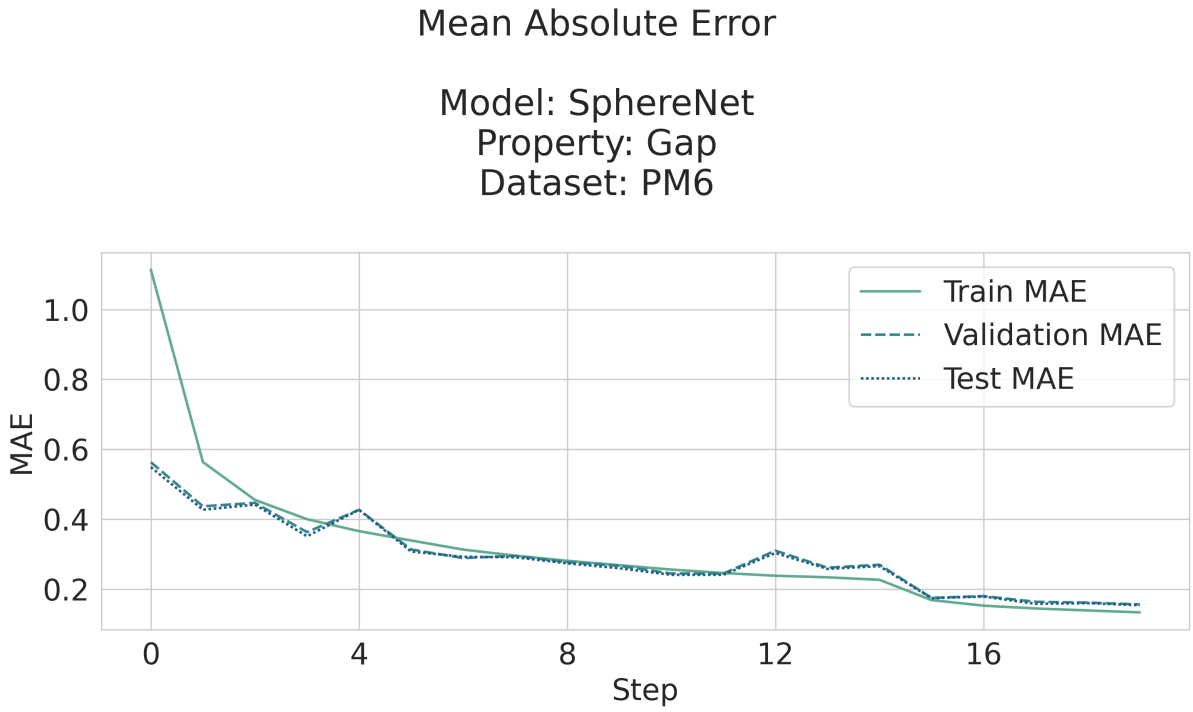

Figure 3.1: Training process over time for SphereNet using PM6 data.

4. CONCLUSION

4.1 Perspective

Until recently, deep learning has largely been advanced on data in the Euclidean space, such as image classification and natural language understanding.

Advances in graph neural networks would stand to share the breakthroughs in Euclidean-data problems to graphs, which would immensely support research in chemistry, social phenomena, etc. The gain to be made here is that research is highly limited by being constricted to one space. Expanding beyond the Euclidean space removes many of the favorable assumptions that can be made to simplify problems; however, they dilute the representational quality of the encoding because the problems in the world are not necessarily Euclidean.

Clearly, Graph Neural Networks feature exciting prospects for the field of drug discovery, which is in more critical need of advancements than ever before. However, as new models are architected to advance the world's toolbox, it is important to bear in mind the implications of benchmarking.

After all, a model's performance metrics are all relative to the datasets on which they are evaluated. Oftentimes, smaller datasets that are more manageable to work with tend to have the side effect of being unrepresentative of the chemical space.

Importantly, while this is an issue in prediction tasks, there stands to be even more of a gain for incorporating larger datasets into generation tasks, which is a future research direction that has been receiving attention lately. This is because predictive tasks are constrained to their datasets, whereas generation tasks need novelty.

4.2 Outcomes

Performance on PM6 is not indicated to be as strong as on QM9; however, it is important to note that QM9 is widely considered as a benchmark dataset for chemistry GNN problems. Therefore, many models are designed with such datasets in mind. PM6 is not to be discarded

due to a higher error rate, but rather, it should be considered as an additional benchmark that can signal the performance of the model toward a larger subset of the chemical space that more closely represents reality.

PM6 is shown as being capable of assessing models relative to each other because it indicates mostly the same pattern by QM9. Nevertheless, there are some differences that point out opportunities for improvement using a larger dataset. Interestingly, it is not always the case that a more expressive model produces a lower validation and testing MAE. This could indicate that the effectiveness of the additional expression may not translate to a larger dataset like PM6.

Overall, this study has made it clear that QM9 and PM6 need not be compared to each other to provide valuable insights. Instead, this work encourages comparing models with several datasets, since a model that has the best metrics on one dataset may not exhibit that same performance on another. Since different datasets, and especially subsets, focus on different kinds of molecules, a model's versatility is difficult to assess using only one source.

4.3 Direction

GNNs in computational chemistry have been lifted by many groundbreaking works in the last several years, but there is still progress to be made before this research area can have more widespread applications in industry. One of the greatest disparities between GNN research and industry needs is found in datasets. This is likely because the two tracks are optimizing for different outcomes; in industry, the performance of a model is important, but the discovery of a novel drug is the ultimate goal. Because of this, industry datasets tend to be much larger, containing molecules that have few similarities amongst them. While designing models for these datasets may be considerably more challenging, the outcome would be a method that is more resilient to the variety in the chemical space.

Although many industry datasets are proprietary, PubChem PM6 is one of the largest molecular compound datasets, with over 90 million valid molecules. Data have been contributed to PM6 by many institutions and organizations, producing a diverse source. Benchmarking models against larger and broader datasets like PM6 could be the next step needed to raise future GNNs as

a stronger candidate for applications.

The case for more distinct and expressive GNNs is clear, but model development should be guided by performance metrics across multiple datasets. More thorough benchmarking would make models more resilient to the types of molecules that may be less commonly seen in research, but are omnipresent in industry and practical applications. Novel model architectures have driven the capabilities of GNNs, but the efforts to develop them could be more guided and targeted with stronger analytics that are provided by more representative samples.

GNNs are playing a rising role in all aspects of our lives, and especially chemistry. Developing practices that encourage real-world feasibility will further accelerate this growth, and yield fruitful applications.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[3] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, pp. 1263–1272, PMLR, 2017.

[4] T. Sterling and J. J. Irwin, "Zinc 15 – ligand discovery for everyone," *Journal of Chemical Information and Modeling*, vol. 55, no. 11, pp. 2324–2337, 2015. PMID: 26479676.

[5] D. Sholl and J. A. Steckel, *Density functional theory: a practical introduction*. John Wiley & Sons, 2011.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[8] J. Klicpera, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," *arXiv preprint arXiv:2003.03123*, 2020.

[9] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical review letters*, vol. 120, no. 14, p. 145301, 2018.

[10] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," *Advances in neural information processing systems*, vol. 30, 2017.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[12] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific data*, vol. 1, no. 1, pp. 1–7, 2014.

[13] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, 2012. PMID: 23088335.

[14] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Science Advances*, vol. 3, no. 5, p. e1603015, 2017.

[15] H. E. Sauceda, M. Gastegger, S. Chmiela, K.-R. Müller, and A. Tkatchenko, "Molecular force fields with gradient-domain machine learning (gdml): Comparison and synergies with classical force fields," *The Journal of Chemical Physics*, vol. 153, no. 12, p. 124109, 2020.

[16] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*, pp. 2990–2999, PMLR, 2016.

[17] J. Klicpera, S. Giri, J. T. Margraf, and S. Günnemann, "Fast and uncertainty-aware directional message passing for non-equilibrium molecules," *arXiv preprint arXiv:2011.14115*, 2020.

[18] Y. Liu, L. Wang, M. Liu, X. Zhang, B. Oztekin, and S. Ji, "Spherical message passing for 3d graph networks," *arXiv preprint arXiv:2102.05013*, 2021.

[19] J.-L. Reymond, "The chemical space project," *Accounts of Chemical Research*, vol. 48, no. 3, pp. 722–730, 2015. PMID: 25687211.

[20] M. Nakata, T. Shimazaki, M. Hashimoto, and T. Maeda, "Pubchemqc pm6: Data sets of 221 million molecules with optimized molecular geometries and electronic properties," *Journal of Chemical Information and Modeling*, vol. 60, no. 12, pp. 5891–5899, 2020. PMID: 33104339.

[21] M. Liu, Y. Luo, L. Wang, Y. Xie, H. Yuan, S. Gui, H. Yu, Z. Xu, J. Zhang, Y. Liu, *et al.*, "Dig: a turnkey library for diving into graph deep learning research," *Journal of Machine Learning Research*, vol. 22, no. 240, pp. 1–9, 2021.