universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Property Prediction with Matched Molecular Pair Networks

verfasst von | submitted by

Rupendra Lal Shrestha

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Assoz. Prof. Dipl.-Inf. Dr. Nils Morten Kriege

Acknowledgements

Undertaking this thesis posed a hard yet interesting challenge for me, and I am grateful for all the tremendous support that helped me achieve it. First and foremost, I would like to express my sincerest gratitude to my supervisor, Nils Morten Kriege, for allowing me to undertake this project as well as for his invaluable guidance, steadfast support, enduring patience, and constructive feedback throughout this project.

I would also like to give my warmest thanks to my co-supervisor, Steffen Hirte, for his crucial suggestions, invaluable assistance, and continuous encouragement.

I want to express my heartfelt gratitude to my family for their endless support, constant motivation, and unyielding belief in me.

I am also deeply thankful to my friends for their encouragement, concern, shared joyful moments, and for always cheering me on.

Abstract

This thesis focuses on the application of Machine Learning (ML) techniques in drug discovery, emphasizing on the Matched Molecular Pair (MMP) methodology alongside Graph Neural Network (GNN) for molecular property prediction. To accelerate the screening of candidate molecules for drug development, computational methods have been employed to predict and filter molecules based on their relevant properties. The change in property between two structurally similar compounds is encapsulated by the theory of MMP, which is defined as a pair of compounds that differ by a well-defined structural change. This concept aids in revealing structural elements essential to property prediction. One approach that employs this concept is Network Balance Scaling (NBS), which integrates the predictive capabilities of ML models and the structural relationships between compounds as an MMP network. However, NBS is primarily utilized as a post-processing step, thereby limiting the potential for further optimization from prediction models such as a GNN.

This thesis presents an approach that incorporates the NBS methodology with the GNN framework through a custom loss function for training. We evaluate our approach on benchmark datasets through extensive experiments, highlighting its advantages and limitations. Additionally, we provide a comprehensive overview of relevant literature regarding the use of MMPs in property prediction. This thesis contributes a robust model aimed at improving performance in molecular property prediction and advancing the role MMPs play in property prediction.

Kurzfassung

Diese Dissertation konzentriert sich auf die Anwendung von Machine-Learning-Techniken (ML) in der Arzneimittelforschung und verwendet die Methode der Matched Molecular Pairs (MMP) neben dem Graph Neural Network (GNN) zur Vorhersage molekularer Eigenschaften. Um das Screening von Kandidatenmolekülen für die Arzneimittelentwicklung zu beschleunigen, wurden rechnergestützte Methoden eingesetzt, um Moleküle basierend auf ihren relevanten Eigenschaften vorherzusagen und zu filtern. Die Veränderung von Eigenschaften zwischen zwei strukturell ähnlichen Verbindungen wird durch die Theorie des MMPs erfasst, das als Paar von Verbindungen definiert ist, die sich durch eine klar definierte strukturelle Veränderung unterscheiden. Dieses Konzept hilft dabei, strukturelle Elemente aufzudecken, die für die Eigenschaftsvorhersage wesentlich sind. Ein Ansatz, der dieses Konzept verwendet, ist das Network Balance Scaling (NBS), das die Vorhersagefähigkeiten von ML-Modellen und die strukturellen Beziehungen zwischen Verbindungen als MMP-Netzwerk integriert. Allerdings wird NBS hauptsächlich als Nachbearbeitungsschritt verwendet, was das Potenzial für weitere Optimierungen aus Vorhersagemodellen wie einem GNN einschränkt.

Diese Dissertation stellt einen Ansatz vor, der die NBS-Methodik mit dem GNN-Framework durch eine neu entwickelte Verlustfunktion für das Training integriert. Wir evaluieren unseren Ansatz anhand von Benchmark-Datensätzen durch umfangreiche Experimente und heben dabei seine Vorteile und Einschränkungen hervor. Zusätzlich bieten wir einen umfassenden Überblick über relevante Literatur zur Verwendung von MMPs bei der Eigenschaftsvorhersage. Diese Dissertation trägt ein robustes Modell bei, das darauf abzielt, die Leistung bei der Vorhersage molekularer Eigenschaften zu verbessern und die Rolle von MMPs bei der Eigenschaftsvorhersage voranzutreiben.

Contents

List of Tables

List of Figures

1. Introduction

In modern drug discovery, the prediction of molecular properties holds utmost importance. The process of creating new drugs is multifaceted, involving the extraction of knowledge from extensive chemical compound databases. Before modifications can be applied to compounds during the optimization process, multiple candidate compounds must undergo screening. Computational methods play a crucial role in accelerating this overall process by accurately predicting compound properties. Various Machine Learning (ML) methods are available for predicting properties such as physicochemical attributes, Absorption, Distribution, Metabolism, Excretion, and Toxicity (ADMET) parameters, potency, among others [MC19; Bag+21; Kun23].

The impact of chemical substituents on the potency or property of a compound is widely acknowledged [HB07; HS08]. Currently, the theory of Matched Molecular Pair (MMP) is of great interest. This concept is founded on the premise that chemical compounds within similar descriptor spaces exhibit similar behaviour in properties. Formally, an MMP represents a pair of compounds that differ by a well-defined structural transformation. This methodology is at the heart of a chemist's intuition, as altering a specific group often leads to a change in the property of interest. MMPs have been extensively utilized for tasks such as lead optimization, compound prioritization, ADMET analysis, prediction of physicochemical properties, rule mining, and more [Kra+17; Lum+20; Pap+10].

In the context of property prediction, MMPs are particularly of interest as they reveal structural elements between pairs of compounds that directly influence their property. In recent years, MMPs have been integrated with various ML models in the prediction of Activity Cliff (AC), which are compound pairs that are structural analogues that exhibit large difference in potency. Furthermore, MMPs have also been used to provide structural explanations to predictions made by ML approaches. Concerning property prediction for single compounds, Network Balance Scaling (NBS) [Hön+22] is an interesting concept that combines MMP methodology, represented by MMP networks, with ML models.

In the NBS methodology, a network is constructed from the MMP relations between compounds alongside a prediction model, which is further utilized to formulate an optimisation problem that optimises the predictions for single compounds generated by the applied prediction model. However, NBS can only be used as a post-processing step, thereby limiting the potential for further optimisation from prediction models such as a Graph Neural Network (GNN). Simultaneously, GNNs have been a popular framework in the field of property prediction, where molecules represented as graphs offer a unique way to capture the intricate structural relationship within molecules [Zha+21; Wie+20].

Building on the NBS approach, we aim to combine the expressive nature of GNNs along with the NBS methodology to investigate whether a GNN model is capable of making better predictions by leveraging the relations provided by an MMP network.

Specifically, a GNN model is trained with a custom loss function that incorporates both the GNN framework and the NBS optimisation. By effectively balancing the contribution of these two approaches i.e. fitting predictions and enforcing MMP rules, we expect to observe an increase in performance compared to using either approach individually. Furthermore, through the analysis of the MMP network, we aim to investigate the factors influencing the improvement or limits of our approach.

2. Fundamentals

This chapter introduces MMP, focusing on its identification methods and common applications in the literature, particularly in drug discovery.

2.1. Matched Molecular Pair

An MMP is formed by two chemical compounds that differ only by a well-defined, structural change. Hence, the compounds belonging to an MMP can be interconverted from one another by interchanging the substructure at this single site [Lea+06; Gri+11]. An example of an MMP is shown in Fig. 2.1. MMP methodology is based on the Structure-Activity Relationship (SAR)/Structure-Property Relationship (SPR), that links the physical or biological properties of a compound directly with its structure. With MMPs, the change in properties between the compound pairs can be directly linked to the specific structural transformation.

MMPs have been used to predict many different physical and biological properties of a chemical compound and also to provide explanations for said predictions. MMPs are particularly of interest in property prediction, as they reveal structural elements between pairs of compounds that directly contribute to said property. Various models utilising different ML methods have been devised to use MMPs diversely [TMB23; Hön+22; ARK20].

2.2. MMP identification

Due to the nature of molecular structures, finding all MMPs can be a computationally exhaustive process. Among the many methods for identify MMPs in the literature, two approaches are prominently used: (1) fragmentation-based and (2) Maximum Common Subgraph [TE17].

2.2.1. Fragmentation-based

Fragmentation-based MMP identification algorithms decompose a molecule into fragments that can then be indexed. The advantage of such an approach is that a molecule needs to be processed once as the fragments are already defined. The Hussain and Rea algorithm [HR10] is a computationally efficient method for identifying MMPs based on the fragmentation method. The algorithm works by first fragmenting and then accordingly indexing the compounds.

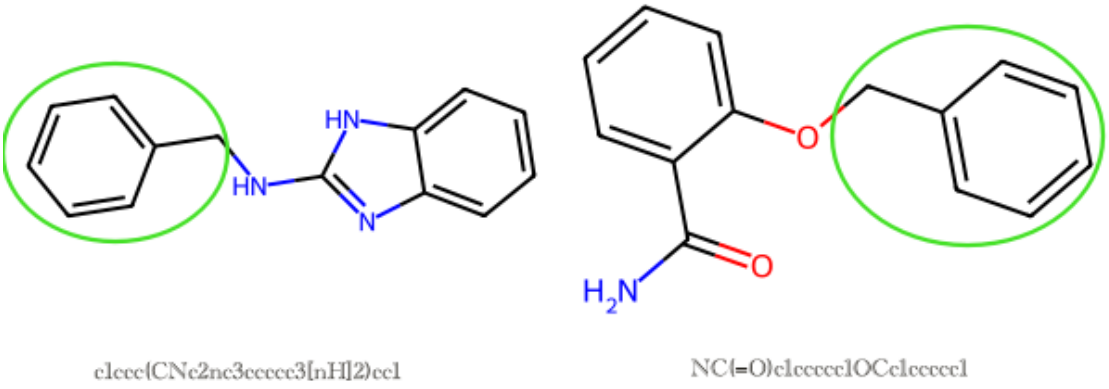


Figure 2.1.: Two compounds that are an MMP. The highlighted parts in both compounds are the common substructure. The identifier below each compound is its canonical SMILES notation [Wei88].

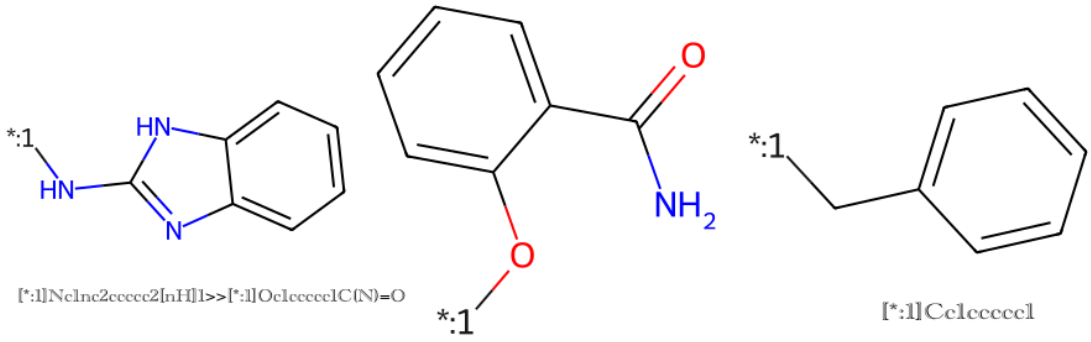


Figure 2.2.: The fragments from the MMP in Fig. 2.1. The fragments on the left and middle are the distinct substructure (variable) in the MMP with its SMIRKS notation below. The fragment on the right is the common substructure (constant) with its SMARTS notation.

All possible fragments for each compound in the data set under analysis are identified. After examining each acyclic bond between two non-hydrogen atoms in a compound, a cut is made at the appropriate point, resulting in two fragments. In the indexing phase, all fragments pairs (X, Y) formed are canonicalised and added to the index. First, fragment X is added as a "Key" into the index with fragment Y as its "Value". The reverse is also performed. An identifier for the compound is also added alongside the fragment. All pairwise combinations of the compounds in the values (variables) part of a key (constant) are valid MMPs. The key and values fragment are presented in Fig. 2.2. The SMIRKS format (below the leftmost figure in Fig. 2.2) is used to specify the change in substructure, and the SMARTS format (below the rightmost figure in Fig. 2.2) is used to specify a fragment substructure.

Additionally, double and triple cuts can also be performed. For these scenarios, the fragmentation results in a core and two fragments. The core is stored as a value, and the two fragments are added in a singular form with the canonicalised dot-disconnected SMILES. Furthermore, to find MMPs with transformations involving the substitution of hydrogen atoms, a hydrogen atom is added to all the single-cut fragments in the index key. If the resulting canonicalised SMILES for a key is present in the input structures of the data set then a "$*[H]$" is added to the value part of that key. This algorithm can efficiently identify all possible MMPs within a data set irrespective of the size of the structural moiety that is modified. User-specific conditions can be used to filter the substructures before adding them to the index. Such conditions can include the number of non-hydrogen atoms in the substructure, attachment order, etc. One such widely used implementation of the Hussain and Rea algorithm is MMPDB [DHK18] which makes it accessible to create, store and query MMP rules.

2.2.2. Maximum Common Subgraph

Maximum Common Subgraph (MCS) algorithms work by performing a pairwise comparison within a compound data set to find the MCS between each pair of compounds. These algorithms have the advantage that they identify only the absolute minimum substructure change between compound pairs. A general MCS algorithm works by first initializing a common substructure. This substructure is then iteratively expanded by adding atoms and bonds that match specific criteria. The goal is to maximise the overlap of the common substructure while considering factors such as aromaticity, formal charge, and ring membership. This iterative process continues until no further expansion is possible or until a stopping criterion is met. Flexible Maximum Common Substructure (FMCS) [DH13] is one such algorithm to find MCS. The identified MCS represents the core structural motif that is present in compound pairs. To identify MMPs, the atoms within compound pairs that are not part of the MCS are then analysed to determine if they constitute a single-point change.

WizePairs [WGS10] is one widely known implementation of the MCS method to find MMPs along with archiving and applying medicinal chemistry. To identify MMPs with this implementation, a matrix is created with all possible combinations of compounds in the data set and then applying the MCS algorithm on each pair. The matches with the most mapped atoms are favoured, with ties resolved by penalising unmatched ring bonds. In this context, an MMP is defined as having no more than a 10% difference in structure, with a minimum of 90% of atoms in the largest molecule considered a match. In a second iteration, atoms with differing atomic properties are identified, and their radii are adjusted accordingly. Atoms with a bond count greater than four are deleted, resulting in an extended version called Remainder after Elimination of Common Substructure (RECS). Valid transformations are those where the RECS for each molecule consists of a single fragment, indicating a modification at a single site. Transformations with multiple fragments are considered invalid and not processed. A SMIRKS string is encoded for each transformation, starting with the outer shell of atoms (three bonds away from the modification site) and gradually reducing the number of bonds until either no remaining atoms are part of the common substructure or the number of fragments in the RECS is no longer one, indicating an invalid transformation. Up to four SMIRKS strings for each MMP are identified, each encoding different levels of the local structural environment of the modification. The SMIRKS are encoded by iterating over the RECS where the same atom mapping is applied between the template and corresponding target graph atoms. Finally, to group all similar transformations, the atomic mappings were removed from RECS and the resulting strings were concatenated with a dot separator. For example, the transformations [c:7]([H])»[c:7][F] and [c:18]([H])»[c:18][F] results in cH.cF. MCS-based algorithms tend to be slower as they require a full $n \times n$ comparison of all the compounds in a data set. Hence, heuristics such as bond elimination, seed pruning, etc., are required to make it efficient.

2.3. MMP Analysis

Building on MMP algorithms, MMP Analysis (MMPA) is a computational approach used to extract valuable information from chemical assays and databases. Different statistical and data mining methods are utilized in MMPA to extract critical and interesting information such as transformation rules, activity characteristics, predictions, etc., from a group of molecules. MMPA is widely applied in the literature to obtain critical knowledge about the compounds present in and between chemical assays [Dua+23; Kra+14; War+12; Lon+22; Van+23; Fu+19].

2.4. Machine Learning in Drug Discovery

ML models such as Support Vector Machines (SVM), Random Forests (RF) and Deep Learning (DL) models have played a crucial role in drug discovery. Numerous studies have employed ML models for a variety of tasks, including compound screening, molecule optimisation, ADMET prediction, bioactivity and physical property prediction, and Drug-Drug Interaction assessment, among others [He+21; Wie+20; Yan+23; Zha+21; RB22; Gri23]. Concerning property prediction, ML models are commonly applied to molecules by first using a form of feature representation that conveys the structural components of the compounds are conveyed by that representation. MMP methodology can be further combined with the ML model to extract insightful knowledge regarding specific structural elements.

3. Related Work

This chapter provides a comprehensive literature review on the utilisation of MMPs in property prediction, exploring various studies and methodologies involving MMPs in diverse prediction tasks related to molecular property. Initially, we discuss the application of MMPs in predicting the properties of individual compounds. We then examine how MMP-based methods are employed to predict the change in property. Additionally, we review methodologies where MMPs are utilised to predict ACs. Finally, we discuss the role of MMPs in advancing explainable property prediction.

3.1. Prediction for Single Compounds

A recent model termed Network Balance Scaling (NBS) has been devised by Hönig et al. [Hön+22] to incorporate the predicted properties of new compounds into a network of MMPs. Here, the core idea is that the molecular property prediction is improved by utilising the network of MMP relations where the compound of interest benefits from all known MMP transformations in the training set. The paper presents a fully integrated approach to performing a supervised ML together with MMPA as a post-processing step utilising convex optimisation. The model can predict a compound's property by itself and in case another prediction model (RF, GNN, etc.) is applied beforehand, NBS can improve upon this predicted property. The model is separated into two parts. First, an MMP network is generated based on the identified MMP relations and their properties from a data set, then an optimisation step is applied to this MMP network utilising a convex optimisation.

An MMP network is a graph where the nodes are compounds and the edges are the transformation relationship between the compounds. Additionally, the nodes are divided into green and red nodes where the green nodes represent the compounds in training set (compounds with experimental property) and the red nodes represent the compounds in test set (compounds without a property or with a predicted property). The edges also describe the difference in properties between the two compounds in addition to the structural changes. As there can be numerous transformations between two nodes, only one transformation is chosen based on its average activity difference (maximum value) to represent an edge between two nodes. All edges include a deviation value to account for the error in predicted property for the compounds. Furthermore, for each edge there is also a reverse edge that expresses the reverse transformation. Also, a deviation value is associated with red nodes that either predicts the property for that node or accounts for the prediction error. A focused part of an MMP network that outlines its general structure is shown in Fig. 3.1.

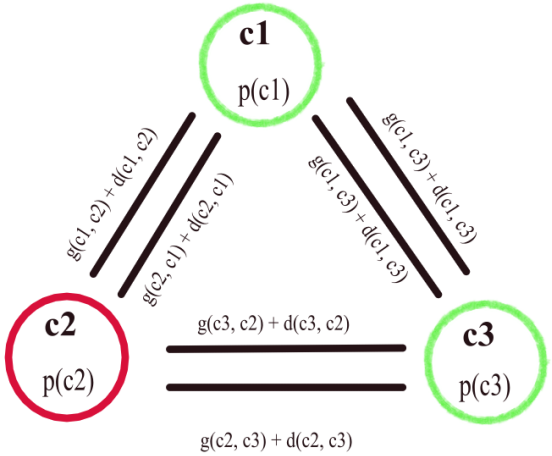


Figure 3.1.: The figure shows a small part of the MMP network as described in the NBS approach. Here, the green nodes are in the training set and the red nodes are in the test set. $g(a, b)$ is the difference in property between two nodes and $d(a, b)$ is the deviation value associated with the edge. $p(a)$ gives the property of the compound. In the case of red nodes $p(a)$ is either the predicted property and node deviation or only the node deviation.

To optimise the defined MMP network, we create an optimisation problem where the objective is to minimise the sum of squares of the edge deviation values. The constraints for the optimisation problem are formulated with the experimental/predicted property, the change in property, the node deviations of red nodes, and edge deviations for all edges in the network. A detailed description is provided in section (4.4.1). After optimisation, the node deviation values provide the predicted/optimised values for the test set compounds. The entire workflow is presented in Fig. 3.2.

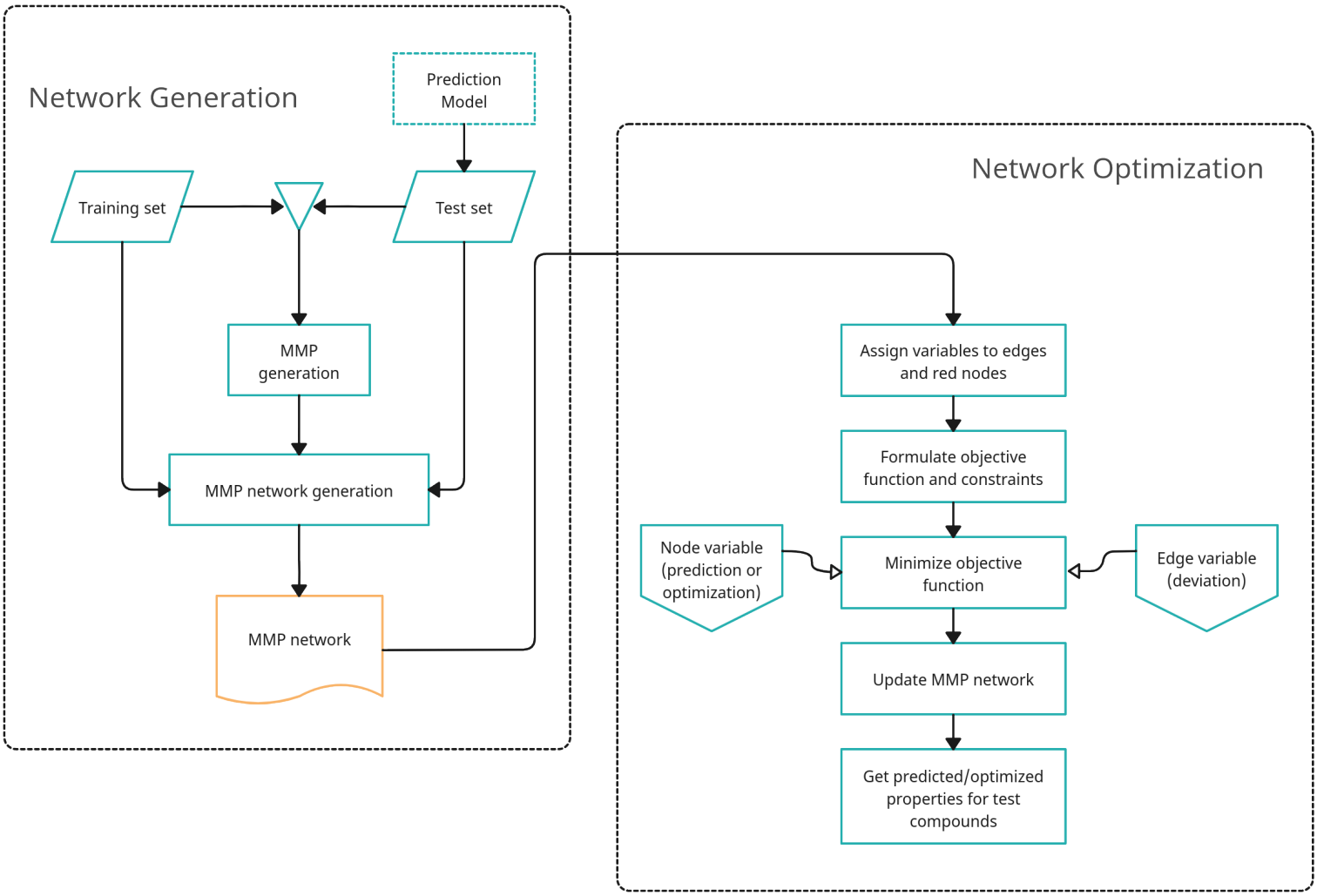


Figure 3.2.: NBS workflow.

A different approach using Matched Molecular Series (MMS) pairs and MMS Analysis (MMSA), is on par with ML models and MMPA [ARK20]. MMS are a set of structurally related compounds that have been systematically modified at specific positions, where each compound in the series differs from the others by a single modification [WB11]. An MMS pair is defined as any two MMS that share at least five identical R-groups. SAR information from medical data is exploited using MMSA to enable the compound design by transferring information from one MMS to another. This approach consists of two components: A similarity metric to identify MMS pairs with significant similarities and a prediction method that transfers the SAR from one MMS to another which enables the prediction of the property of the compound with the new R-group. The best model used central Root Mean Square Deviation (cRMSD) as the similarity metric and nearest neighbour model (nnModel) as the prediction method.

3.2. Prediction of Change in Property

Using MMPA in the context of change in property prediction, Leach et al. [Lea+06] were able to successfully predict the magnitude of change in physical property when substituting one substructure with another. Here, the authors have used statistical mean for an MMP rule, as a measure for the magnitude of property change among MMPs.

Physical and biological activities cannot always be predicted in the same manner due to the nature of the bioactive elements involved. One such limitation in predicting biological activity is the averaging of rules from diverse binding sites with different SAR characteristics. O'Boyle et al. [OBo+14] have used the concept of MMS and developed "Matsy", that tries to overcome the aforementioned limitation. For reference, An MMS of length two is equivalent to an MMP. The "Matsy" method predicts R groups that are likely to improve an activity given an observed activity order for a matched series.

Combining MMPA and Support Vector Regression (SVR), Vega de León and Bajorath [VB14] have predicted the potency difference values for MMPs using kernel functions. The model was applied using various kernel functions and fingerprint descriptors for the representation of molecule fragments. All the kernel functions are based on the Tanimoto similarity function but use various fingerprints based on different structural aspects of the molecule pair.

The MMP/ML method successfully combine MMP methodology with RF, Gradient Boosting Machines (GBM) and Deep Neural Networks (DNN) to predict ΔpIC_{50} values [Tur+17]. pIC_{50} is a bioactivity property that provides information on the potency of a compound in inhibiting a biological target. Two molecule optimisation scenarios are covered with this method- (1) For new fragments and (2) For new static core and transformation. In both cases, a ΔpIC_{50} is predicted. For training, the shared static core and the differing fragments between the MMP were individually encoded as hashed Morgen fingerprints and were trained with RF, GBM and DNN, with DNN showing the best result for the change in pIC_{50} values.

3.3. Prediction of Activity Cliffs

Another paradigm of interest is Activity Cliff (AC) [Stu+14; Zha+23; Dab+23; TMB23] are pairs or groups of structurally similar compounds or structural analogues that exhibit a large difference in activity values (potency). These compound pairs are of interest as they show SAR discontinuity, i.e., a small structural modification leads to a large potency change. Such substructural changes could also cause problems for medicinal chemists later in the optimization stage due to the aforementioned properties [WB10].

For the prediction of ACs, MMPs along with the common core were used in the form of images to predict MMP-cliffs [IVB21b]. An MMP-cliff is an MMP formed by two compounds that are active against the same target and have a statistically significant difference in potency [Hu+12]. Images of the three structures constituting an MMP (the static core and the two fragments) were concatenated horizontally to obtain a single image. These images were then trained using a Convolutional Neural Network (CNN) to distinguish MMP-cliffs from non-ACs. The CNN model was shown to be successful in learning chemistry from MMP images to make AC predictions. Following this proof-of-concept, Bajorath et. al. have combined CNN with transfer learning to distinguish ACs from non-ACs [IVB21a]. Here, the CNN model was trained on the image data set and further fine-tuning was carried out using transfer learning, where the last fully connected layer of the CNN is replaced with three fully connected layers. The compound pair images were in the form of a Condensed Graph of Reaction (CGR). In CGR, the shared core of an MMP and the two exchanged substructures are represented as a single pseudo-molecule.

A recent study focused on Drug-Target Interactions (DTIs) used MMP along with Graph Convolutional Network (GCN) for the prediction of ACs [Par+22]. Park et al. [Par+22] have proposed two GCN architectures that predict AC relationships in MMPs based on graph representation. The two architectures named ACGCN-mmp and ACGCN-sub work similarly, except for the inputs to the model. ACGCN-mmp takes two whole molecular graphs (MMP) as input, whereas ACGCN-sub takes three substituents comprising an MMP i.e. the core and the two differing substructures as input. First, the input graphs are converted into one-dimensional vectors using graph convolution layers and readout functions. These vectors then pass through fully connected layers and finally through an output layer to classify the molecules as AC or non-AC.

3.4. Explainable Property Prediction

A study was done to disentangle the structural patterns exploited by GNN in compound property predictions [ARJ23]. A message-passing GNN model was trained to make predictions on compound pairs using a substructure-aware loss, a modification to the regression objective. This loss function takes into account the uncommon structural motifs between compound pairs. The GNN model trained with this procedure is further used to test the performance of feature attribution methods. In the context of property prediction, the aspect of interest are the application of the substructure aware loss to

make a GNN learn the uncommon latent spaces between compounds and the use of this prediction in feature attribution method. Although the formalism of MCS is used, it is analogous to the concept of MMP.

Another study on the ligand-based AC prediction has used SVM and MMP kernels for the interpretation of the predictions made by SVMs [Tam+21]. This method decomposes the cross-term contribution of the MMP kernel into core and substituents evenly which enables the approximation of the kernel into a sum of feature weights.

MMPA has also been employed to provide the structural reasoning behind predictions. A study on hematotoxicity was conducted where Long et al. [Lon+22] utilised various ML algorithms and molecular representation to predict Hematotoxic compounds. Here, MMPA was employed to find, analyze and confirm non-hematotoxic substructures. Another study by Vangala et al. [Van+23] has employed MMPA as an additional method alongside the Gradient-weighted Class Activation Mapping (Grad-CAM) model for explainability of the compounds classified by the pBRICs method. Fu et al. [Fu+19] have used MMPA and descriptor importance analysis to explain predictions made by numerous QSPR models used for the prediction of $logD_{7.4}$. The logarithm of the distribution coefficient at pH 7.4 ($logD_{7.4}$) is a measure of the lipophilicity of a molecule and describes how it is likely to distribute between lipophilic and hydrophilic environments.

4. Methodology

Building on the idea of employing an MMP network with ML models, in this chapter, we propose a method that tries to force a GNN to be consistent with MMP rules for property prediction. Our goal is to investigate whether a model can be formulated such that the model improves predictions with the aid of transformations with every iteration. The enforcement of MMP rules is facilitated through an MMP network, based on the NBS methodology [Hön+22]. The GNN architecture used, the MMP network structure and the loss function for training are detailed in this chapter.

4.1. GNN Architecture

Predictions on molecular graph representations are made using a GNN. In a molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, nodes $\mathcal{V} = \{1, ..., n\}$ represent atoms and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are defined by a predefined structure. Given a molecular graph with atom features $X \in \mathbb{R}^{|\mathcal{V}| \times d}$ and edge features $\mathcal{E} \in \mathbb{R}^{|\mathcal{E}| \times k}$, where d and k are the dimensionality of the node and edge features respectively, GNNs work by iteratively updating the node embeddings x_i^l with each layer by aggregating the localised information using the following functions:

$$m_v^l = AGGREGATE\left(\left\{\left\{x_u^{l-1} \mid u \in N(v)\right\}\right\}\right)$$

$$x_v^l = UPDATE\left(x_v^{l-1}, m_v^l\right)$$

Here, $\{\{\dots\}\}$ denotes a multi-set and $N(v)$ are the neighbouring nodes to $v \in V$ and $l \in L$ is one layer within the GNN architecture. After these L layers, a graph representation X^L can be obtained using global aggregation [Gil+17].

The Graph Isomorphism Network (GIN) [Xu+19] is a powerful form of the above scheme which can be defined as:

$$x_v^l = MLP\left((1 + \epsilon) \cdot x_v^{l-1} + \sum_{w \in N(v)} x_w^{l-1}\right)$$

GINs use sum as an aggregation function on all the neighbouring nodes of $v \in V$ and apply a Multi-Layer Perceptron (MLP) to update the node embeddings x_v^l of v. Here, ϵ is a learnable scalar parameter used to control the self-information propagation.

4.2. MMP Network

An MMP network describes the MMP relations between molecules within a dataset. Formally, an MMP network for a dataset is given by an undirected multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V} = \{1, \ldots, n\}$ represent individual molecules within the dataset, and the edges $\mathcal{E}$ are a multiset of unordered pairs of vertices from $\mathcal{V} \times \mathcal{V}$, indicating structural transformations between molecules. A transformation, represented by an edge between two molecules, is defined as any function $T : \mathcal{V} \to 2^{\mathcal{V}}$, where T maps molecules from set $\mathcal{V}$ to a set of other molecules. With this, multiple possible transformations between any two molecules are taken into account. The nodes are associated with the property of interest $p(v)$ for all $v \in \mathcal{V}$.

Additionally, the nodes in the MMP network can be sub-divided into green nodes $\mathcal{V}_{\text{green}} \subset \mathcal{V}$ and red nodes $\mathcal{V}_{\text{red}} \subset \mathcal{V}$. $\mathcal{V}_{\text{green}}$ represents molecules whose properties are known and $\mathcal{V}_{\text{red}}$ represent molecules with unknown property.

Furthermore, symmetric transformations are taken into account; hence, the edges are undirected, as the direction of a transformation can be arbitrarily interpreted. Each edge is also associated with a μ_T value, which represents the average property difference for the associated transformation T. This value can be calculated as follows:

$$\mu_T = \frac{1}{N} \sum_{(u,v) \in \gamma_T} \Big(p(v) - p(u) \Big) \tag{4.1}$$

Here, γ_T represents all the molecular pairs (u, v) associated with a transformation T and N is the total number of molecular pairs in γ_T. In the case that at least one endpoint of an edge is in $\mathcal{V}_{\text{red}}$, the difference associated with that transformation is assigned zero. An overview diagram of an MMP network is provided in Fig. 4.1. In Fig. 4.1, out network molecules are those not associated with any transformations.

4.3. Custom Loss Function

The proposed loss function combines the use of a GNN and the MMP network. We formulate an optimisation problem from the MMP network and the resulting objective function constitutes a part of the custom loss. The custom loss is formulated as:

$$\mathcal{L} = \lambda \cdot \overbrace{\frac{1}{N} \sum_{i=1}^{N} \Big(f(x^{(i)}) - y_{\text{true}}^{(i)} \Big)^2}^{\text{Fit prediction}} + (1 - \lambda) \cdot \overbrace{\frac{1}{M} \sum_{T \in \tau} \sum_{(i,j) \in \mathcal{E}_T} \Big(h(x^{(j)}) - h(x^{(i)}) - \mu_T \Big)^2}^{\text{Enforce transformation rules}} \tag{4.2}$$

$$h(x^{(i)}) = \begin{cases} y_{\text{true}}^{(i)}, & \text{if } x^{(i)} \in \mathcal{V}_{\text{green}} \\ f(x^{(i)}), & \text{if } x^{(i)} \in \mathcal{V}_{\text{red}}. \end{cases} \tag{4.3}$$

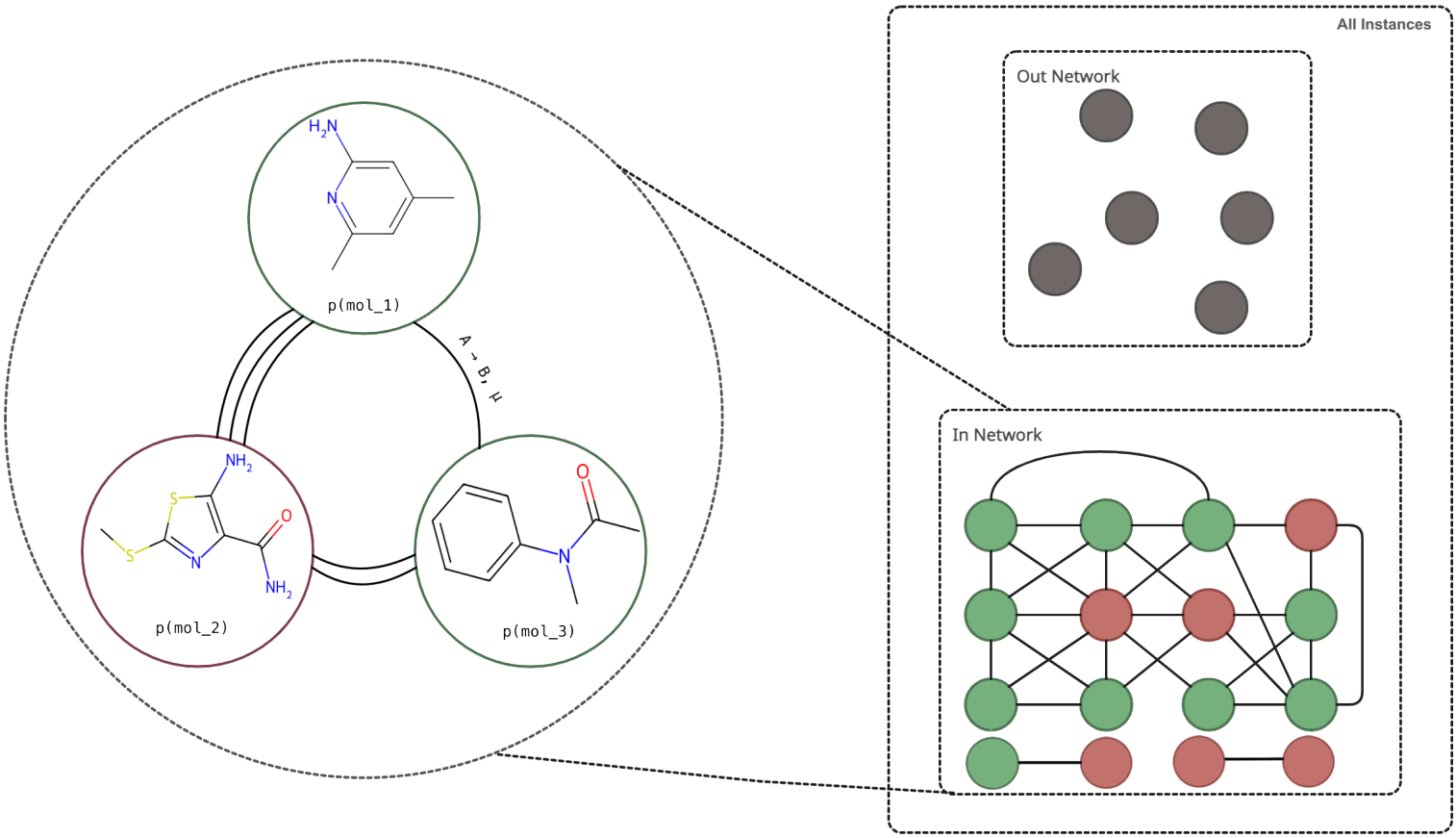


Figure 4.1.: The figure shows a redefined MMP network with the instances of both training (green) and test set (red) molecules. A dataset is differentiated into in-network and out-network (gray) instances. The out-network instances include both training and test molecules that do not have MMP relations. On the focused portion on the left side, nodes are molecules and the edges as transformations. Each transformation $A \rightarrow B$ also has a reverse transformation given by $B \rightarrow A$ and its μ value.

Here, $f(x^{(i)})$ is the prediction made by a GNN model on a molecular graph $x^{(i)}$ and $y_{\text{true}}^{(i)}$ is the true label of $x^{(i)}$. τ is the collection of all the unique transformations and M is the total number of transformations taken into consideration, that is associated with a data set and $\mathcal{E}_T$ are the edges associated with a transformation T. N is the total number of instances in the training set and λ is a hyperparameter whose value ranges between zero to one.

The custom loss Eq. (4.2) consists of two loss terms where the first term fits the predictions made by the GNN model to the true labels and the second term enforces the activity differences of the transformation rules, thereby using the MMP relations in the network. To take full advantage of the MMP relations, the model is trained in full batch mode without randomized shuffling. The normalization terms N and M are introduced so that the loss terms are independent of the number of available instances and their magnitude are comparable. Also, the coefficient λ controls the contribution of the loss terms to the overall loss $\mathcal{L}$. The second term is based on the optimisation problem described in the NBS methodology. A detailed explanation on the optimisation problem given in the paper by Hönig et al. [Hön+22] and the difference between their approach and ours can be found in the following sections.

4.4. Comparison with the NBS Approach

4.4.1. Optimisation Problem Formulation in NBS

The optimisation problem is based on the MMP network for the NBS method, where the goal is to minimizes the edge deviations. The objective function and the constraints are described in this section.

Objective Function

The objective function minimizes the sum of squares over the edge deviations. A free variable x_i is assigned all edges and to nodes in $\mathcal{V}_{\text{red}}$. The objective function is shown in Eq. (4.4).

$$f_0(x) = \sum_{i=1}^{n} \delta_i x_i^2 \tag{4.4}$$

$$\delta_i = \begin{cases} 1, & \text{if } x_i \text{ is an edge deviation} \\ 0, & \text{if } x_i \text{ is a node variable.} \end{cases} \tag{4.5}$$

Here, δ_i is a decision variable associated with each free variable x_i. This variable indicates that only free variables that belong to edges are minimised, while those associated with red nodes are not affected by the objective function as given in Eq. (4.5). The free

variables for red nodes are used to either predict the property or show the error in the predicted property. Such distinction is made so that the optimiser does not also minimise the free variables for red nodes.

Constraints

The constraints are based on the edge deviations and for each edge in an MMP network, a constraint is created. Hence, the number of constraints depends on the number of edges in the network. The basic structure of a constraint is given as:

$$\zeta(v) - \zeta(u) = \mu_{(u,v)} + x_{(u,v)} \tag{4.6}$$

Here, $\mu_{(u,v)}$ represents the average activity difference, and $x_{(u,v)}$ denotes the edge deviation variable associated with an edge $(u, v) \in \mathcal{E}$. It is crucial to note that $\mu_{(u,v)}$ for $(u, v) \in \mathcal{E}$ differs from μ_T as given in section 4.2. The specific formulation of the constraint depends on whether an edge involves a red node. The replacement conditions of $\zeta(a)$ for the exact formulation of a constraint are shown in Eq. (4.7).

$$\zeta(a) = \begin{cases} p(a), & \text{if } a \in \mathcal{V}_{\text{green}} \\ \\ h_{(a)} + x_{(a)}, & \text{if } a \in \mathcal{V}_{\text{red}} \end{cases} \tag{4.7}$$

Here, $\zeta(a)$ represents either an experimental property or a predicted property plus a free variable. When model predictions are available, the free variable measures prediction error (prediction model with NBS), termed node deviation. When predictions are not available, $h_{(a)}$ is set to zero, and the free variable is used for prediction (NBS standalone).

4.4.2. Reformulation of the Optimisation Problem

The original formulation in the NBS approach can be translated into the custom loss function as follows: We have,

$$f_0(x) = \sum_{i=1}^{n} \delta_i x_i^2 \tag{4.8}$$

As δ_i removes variables associated with V_{red} from the objective function, we can remove this term and only include variables associated with edge deviations.

$$f_0(x) = \sum_{\forall (u,v) \in \mathcal{E}} x_{(u,v)}^2 \tag{4.9}$$

Now taking the constraints formulation structure:

$$\zeta(v) - \zeta(u) = \mu_{(u,v)} + x_{(u,v)} \tag{4.10}$$

$$\zeta(v) - \zeta(u) - \mu_{(u,v)} = x_{(u,v)} \tag{4.11}$$

Substituting Eq. (4.11) in Eq. (4.9):

$$\mathcal{L} = \sum_{\forall (u,v) \in \mathcal{E}} \Big(\zeta(v) - \zeta(u) - \mu_{(u,v)}\Big)^2 \tag{4.12}$$

4.4.3. Comparison of the Two Formulations

We have the second term of the custom loss function Eq. (4.2):

$$\left[\sum_{T \in \tau} \sum_{(i,j) \in \mathcal{E}_T} \Big(h(x^{(j)}) - h(x^{(i)}) - \mu_T\Big)^2\right] \tag{4.13}$$

$$h(x^{(i)}) = \begin{cases} y^{(i)}_{\text{true}}, & \text{if } x^{(i)} \in \mathcal{V}_{\text{green}} \\ \\ f(x^{(i)}), & \text{if } x^{(i)} \in \mathcal{V}_{\text{red}}. \end{cases} \tag{4.14}$$

The similarities between Eq. (4.11) and Eq. (4.13) are apparent. In both equations, we have the sum of squared differences for the edges in the network, taking the true label for nodes in $\mathcal{V}_{\text{green}}$. Concerning the nodes in $\mathcal{V}_{\text{red}}$, either a free variable or the predicted property are employed. The differences, are in the terms μ_T and $\mu_{(u,v)}$. The term μ_T represents the average activity difference for a transformation T, obtained by computing the mean of all available activity differences for MMP rules in V_{green}. On the other hand, $\mu_{(u,v)}$ signifies the average activity difference associated with an edge $(u, v) \in \mathcal{E}$, derived by selecting the maximum value among the transformations (μ_T) available for that edge. Consequently, Eq. (4.11) sums only over the edges whereas, Eq. (4.13) involves the sum over all transformations and MMP rules associated with those transformations. Another important aspect is, when optimising the predictions of a model, $\mu_{(u,v)}$ also takes into account the predicted values of nodes in V_{red}. However, in the context of μ_T used in Eq. (4.13), such predictions cannot be considered, as they are calculated before any predictions are applied.

5. Experimental Setup

This chapter outlines the setup for our experiments aimed at thoroughly evaluating the custom loss model introduced in chapter 4. The subsequent sections provide comprehensive details regarding the benchmark dataset collections employed and the state-of-the-art models utilized for comparison.

5.1. Datasets

To validate our model, MoleculeNet [Wu+18] was utilised. Various data sets from different public sources were collected in MoleculeNet and specially designed for benchmarking machine learning methods. Altogether, over 700,000 compounds are contained in this collection along with their respective properties. Furthermore, the datasets contained within MoleculeNet vary in size, ranging from 600 to 437,000 training examples. Various types of molecular properties, such as physicochemical properties and bioactivity, are contained in this collection. Depending on the dataset, different classification or regression metrics are also provided. For our experiments, we mainly focus on the following data sets.

Lipophilicity set (LIPO) contains molecules labeled with dimensionless experimental results of the octanol/water distribution coefficient $logD$ measured at pH 7.4.

Freesolv set are compounds extracted from the Free Solvation Database and the property used is the $expt$, that is the experimental hydration free energies of small molecules in water specified in $kcal/mol$.

ESOL data set are associated with water solubility data for common organic small molecules and the property of interest is the logarithmized, experimentally measured aqueous solubility (sol).

BACE set provides quantitative binding results as pIC_{50} values for a set of inhibitors of human β-secretase 1 (BACE-1).

Another benchmark collection, MoleculeACE [TAG22] was utilised. This collection contains carefully curated bioactivity data on thirty macromolecular targets, which were specifically collected to evaluate the performance of ML models on activity cliffs. Three data sets were chosen for experiments.

CHEMBL218 set comprises of molecules associated with the Cannabinoid receptor 1 (CB1), with reported EC_{50} values. These EC_{50} values indicate the concentration of a compound required to produce half of the maximum possible effect on the CB1 receptor.

CHEMBL219 and **CHEMBL234** are data sets containing collections of molecules with reported K_i values for the Dopamine D3 receptor (D3R) and Dopamine D4 receptor

Table 5.1.: Descriptive statistics of the MoleculeNet data sets.

Stats	BACE	LIPO	ESOL	FREESOLV
`Molecule count`	1513	4200	1128	642
`Property average`	6.522	2.186	-3.05	-3.803
`Min. property`	2.545	-1.5	-11.6	-25.47
`Max. property`	10.523	4.5	1.58	3.43
`Standard deviation`	1.342	1.203	2.096	3.848
`Transformation count`	33890	55118	345166	258438
`Unique transformation count`	29291	45689	306275	239581

(D4R). These K_i values represent the concentration of a molecule required to achieve half-maximal inhibition of the respective receptors.

Table 5.2.: Descriptive statistics of the MoleculeACE data sets.

Stats	CHEMBL218	CHEMBL219	CHEMBL234
`Molecule count`	1031	1865	3657
`Property average`	-2.251	-1.774	-1.537
`Min. property`	-4.991	-4.954	-4.862
`Max. property`	1.523	1.745	1.699
`Standard deviation`	1.051	1.066	1.178
`Transformation count`	14350	27236	96192
`Unique transformation count`	11276	17104	46992

The MMPs required to generate the MMP network were obtained through mmpdb [DHK18] with default settings, including enabling the symmetric option.

5.2. Metrics & Dataset Splitting

The splits required for the experiments were obtained using the randomised nested k-fold cross-validation scheme on all datasets. The nested k-fold scheme utilised a k value of ten for the outer loop and five for the inner loop. In this setup, each of the inner five folds served as a validation set used to determine the optimal hyperparameter configuration for the corresponding outer fold. Subsequently, the outer folds were utilised as hold-out test sets for the final predictions. This validation approach was selected to facilitate hyperparameter optimization during GNN training while utilising the entire dataset for MMP network generation.

Further, as the model primarily focuses on molecules within an MMP network, the hold-out set ($\mathcal{D}_{\text{test}}$) is differentiated into in-network ($\mathcal{V}_{\text{red}} \subseteq \mathcal{D}_{\text{test}}$) and out-network molecules ($m_1, m_2, ..., m_n \subseteq \mathcal{D}_{\text{test}} \setminus \mathcal{V}_{\text{red}}$). Alternatively, out-network molecules are not associated with any transformations and hence cannot have any relation in an MMP network.

In all experiments, Root Mean Square Deviation (RMSD) was used as a comparison metric which can be calculated as follows:

$$RMSD(\mathcal{W}) = \sqrt{\frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} (\hat{p}(w) - p(w))^2} \tag{5.1}$$

Here, $\mathcal{W}$ is the set of chosen molecules within the test set and $|\mathcal{W}|$ is the total number of instances in that set. $\hat{p}(w)$ is the prediction value made by a model on w and $p(w)$ is the true label of w.

5.3. Models

The following models were used as a benchmark comparison to our methodology:

- NBS standalone
- GNN trained with RMSD
- GNN with NBS optimisation
- Training augmentation
- Test-time augmentation.

A GNN model is trained on the training set using RMSD as a loss function, with the resulting model used in making predictions on the test set. These predictions can be further optimised using NBS. To achieve this, an MMP network is generated using the training and test sets, alongside the MMPs generated for the dataset. The predictions for the test molecules by the GNN model are utilised in the generation of the MMP network. NBS optimisation is then performed on the network to obtain the optimised predictions. NBS is also used as a standalone method to obtain predictions following a similar procedure, except the labels for the test set are excluded in generating the MMP network. NBS optimisation is then applied to obtain the predictions.

Training augmentation is achieved by applying various possible transformations to each molecule in the training set, thereby converting them into augmented molecules. Specifically, only the observed transformations from the training set, i.e., those with known activity differences were considered. The labels corresponding to these molecules are augmented by incorporating the average activity difference associated with the applied transformations. Subsequently, the augmented training set is utilised to train a GNN model with RMSD. Similarly, test-time augmentation involves the transformation of each molecule in the test set, resulting in the creation of n augmented test sets. A GNN model trained using the original training set is subsequently employed to predict properties for all these augmented test sets. Following this, the average activity difference for the applied transformation is added to the model's prediction for each test molecule. Lastly, the actual prediction for a test molecule is determined by taking the average of all n

augmentations of that molecule including, the prediction for the molecule itself. Further information is provided in the appendix A.1.

The custom loss function that employs the MMP network is used to train a GNN model. As the MMP network contains both the training set and the test set molecules, careful consideration must be taken. In this regard, only the true labels for the molecules in the training set are taken into account in the calculation of the loss, as specified in Eq. (4.3). Only the predictions for the molecules in the test set are used for the calculation of the second loss term. Moreover, Eq. (4.1) ensures that the molecules in the test set are excluded from the calculation of μ_T. This configuration was chosen such that the second loss term (Eq. (4.13)) closely emulates the NBS approach in both the generation and optimisation of the MMP network.

All GNN based models were implemented using PyTorch [Pas+12], PyTorch Geometric [FL19] and RDkit [Lan]. As for the NBS based models, the implemention provided by Hönig et al. [Hön+22] was utilised.

6. Results and Discussion

Experiments were conducted with various models, and their performance was evaluated. In this chapter, we present and discuss the results of these experiments, with a particular focus on the performance of our proposed model, comparing it with other state-of-the-art models mentioned in chapter 5. For all datasets, we provide a comprehensive analysis aimed at highlighting the strengths and limitations of our model. Additionally, we demonstrate the effectiveness of our model in addressing the research objectives.

The RMSD values shown are a mean of all the outer 10 folds. Additionally, the RMSD values are differentiated into three categories - *all* indicates all instances in a test set, *in* indicates instances in $\mathcal{V}_{\text{red}}$ and *out* indicates instances that are not in $\mathcal{V}_{\text{red}}$ but are in the test set. Here, lower RMSD values imply better performance. The *out* values are omitted for GNN with NBS optimisation as the approach only works on in-network molecules and the RMSD value for the *out* instances remain the same as for the model GNN trained with RMSD. Furthermore, considering that each training set for every dataset underwent augmentation to expand its size by 100-500 %, only the optimal RMSD value from the augmented datasets was selected. Hereafter, the models are abbreviated for clarity: GNN trained with RMSD is denoted as *GNN with RMSD*, GNN with NBS optimisation is referred to as *GNN with NBS*, and the GNN with custom loss is simply labelled as *custom loss*.

6.1. Performance for MoleculeNet Dataset

The comparison of the results for datasets in MoleculeNet is presented in Table 6.1. The performance of the *custom loss* model for *all* instances varies across the datasets in the MoleculeNet benchmark, with the LIPO and BACE datasets exhibiting better RMSD than the other models. The RMSD for the ESOL dataset is worse compared to the *GNN with RMSD*, *GNN with NBS* and *train augmentation* models. As for the FREESOLV dataset, the RMSD for the *GNN with RMSD* and *GNN with NBS* models show better results than the *custom loss* model.

On closer examination of the RMSD values for the *in* instances across the datasets, it is observed that the performance of the *custom loss* model is favourable for the LIPO and BACE datasets compared to the *GNN with RMSD* model. However, for the ESOL and FREESOLV datasets, the performance of the *custom loss* model is comparatively inferior. Specifically, the *custom loss* model demonstrates a 1.44 % improvement over the *GNN with NBS* model for LIPO. Conversely, the difference in performance between the *GNN with NBS* model and the *custom loss* model for BACE is minimal, with only a 0.15 % disparity. In contrast, for the ESOL and FREESOLV datasets, the performance of the

Table 6.1.: Average RMSD values from the experiments on MolecueNet dataset.

Model	BACE (pIC_{50})	LIPO ($logD$)	ESOL ($esol$)	FREESOLV ($expt$)
NBS standalone				
in	0.76 ± 0.2	0.83 ± 0.04	1.102 ± 0.05	2.76 ± 0.44
GNN trained with RMSD				
all	0.74 ± 0.08	0.693 ± 0.07	0.73 ± 0.07	1.51 ± 0.27
in	0.66 ± 0.09	0.635 ± 0.08	0.648 ± 0.08	1.407 ± 0.27
out	0.88 ± 0.09	**0.816 ± 0.06**	**0.97 ± 0.1**	2.19 ± 1.1
GNN with NBS optimisation				
all	0.719 ± 0.07	0.684 ± 0.07	**0.72 ± 0.07**	**1.39 ± 0.21**
in	**0.626 ± 0.08**	0.621 ± 0.08	**0.64 ± 0.09**	**1.26 ± 0.19**
GNN with custom loss				
all	**0.696 ± 0.08**	**0.681 ± 0.04**	0.77 ± 0.05	1.84 ± 0.5
in	0.627 ± 0.07	**0.612 ± 0.05**	0.71 ± 0.07	1.69 ± 0.4
out	**0.819 ± 0.09**	0.825 ± 0.04	0.98 ± 0.1	2.36 ± 1.9
Training augmentation				
all	0.72 ± 0.06	0.73 ± 0.04	0.75 ± 0.05	1.87 ± 0.4
in	0.65 ± 0.08	0.66 ± 0.04	0.64 ± 0.09	1.8 ± 0.4
out	0.84 ± 0.09	0.88 ± 0.07	1.07 ± 0.2	2.1 ± 0.7
Test-time augmentation				
all	0.76 ± 0.09	0.71 ± 0.05	1.24 ± 0.1	2.18 ± 0.7
in	0.7 ± 0.09	0.65 ± 0.06	1.27 ± 0.1	2.16 ± 0.7
out	0.88 ± 0.1	0.85 ± 0.05	1.09 ± 0.1	**2.08 ± 1.1**

custom loss model is notably lower, exhibiting a 9.56 % and 20 % decrease, compared to the *GNN with RMSD* model and a 10.9 % and 34 % decrease, compared to the *GNN with NBS* model. Moreover, in conjunction with the preceding observations, the *custom loss* model outperforms all other models for *in* instances, except for ESOL, where it exhibits a decrease of 9.56 % against the *train augmentation* model.

Discussion. The lower RMSD observed for in-network instances of the BACE and LIPO datasets with the *custom loss* model, in contrast to the *GNN with RMSD* model, implies that, for these datasets, the GNN model is able to successfully incorporate the relations in an MMP network. Furthermore, we observe from the chosen λ values for the different folds from BACE and LIPO shown in Table 6.3, that the best prediction are obtained when the MMP network contributes to the loss function. Conversely, for ESOL and FREESOLV, the optimal results for the *custom loss* relies heavily on the first loss term, that indicates the contribution of the MMP network actually lowers prediction accuracy. These exceptions can be explained by the difference between the μ_T values of the edges for the different folds and the true μ_T values for the MMP network. In Table 6.2, the standard deviation of the average μ_T for all edges across folds in ESOL and FREESOLV datasets are lower compared to standard deviation of the true μ_T of all edges, suggesting ample dissimilarity between the calculated μ_T for an edge and its

true μ_T in the MMP network. Here, average μ_T is calculated by taking the mean of the varying μ_T values of a transformation in different folds. Also, the difference in μ_T values arise due to the exclusion of transformations involved with test set instances in the calculation of μ_T. In contrast, BACE and LIPO have a lower gap between these standard deviations.

Furthermore, in Fig. 6.1c and Fig. 6.1d, numerous edges are observed to have considerably different μ_T values compared to their true values. Conversely, the BACE and LIPO datasets exhibit a lower number of such edges. This could be a contributing factor to the lower prediction accuracy, as the effect of a transformation on a property relies heavily on the information that can be obtained for that transformation from the molecule pairs within a dataset [Kra+17].

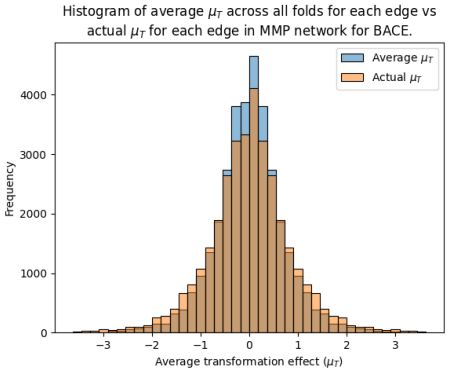
The *GNN with NBS* model performs well for all datasets experimented, compared with the *custom loss* model. One major difference between the two approaches is that, in the *GNN with NBS* model, the predictions for test instances are used in the calculation of μ_T. This reduces the difference between the μ_T values for the different folds and the true μ_T values of the MMP network. Moreover, the MMP networks for ESOL and FREESOLV benefits from these μ_T values, which further helps to optimise the predictions.

Table 6.2.: Descriptive statistics of MMP networks for MoleculeNet datasets

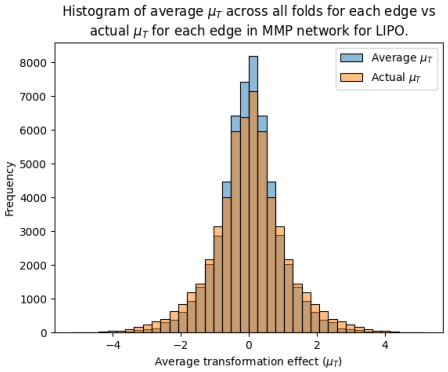
	BACE	LIPO	ESOL	FREESOLV
Nodes	1037	3009	920	594
Edges	33890	55118	345166	258438
Average red nodes	103.7	300.9	92.0	59.4
Average green nodes	933.3	2708.1	828.0	534.6
STD of true μ_T of all edges	0.854	1.103	1.973	4.381
STD of average μ_T of all edges across folds	0.706	0.91	1.635	3.605

Table 6.3.: Optimal λ values for all folds of each dataset in MolecueNet. Here, λ is the coefficient for the first loss term (to fit prediction) and $(1-\lambda)$ is for the second term (to enforce transformation rules) as given in Eq. (4.2).

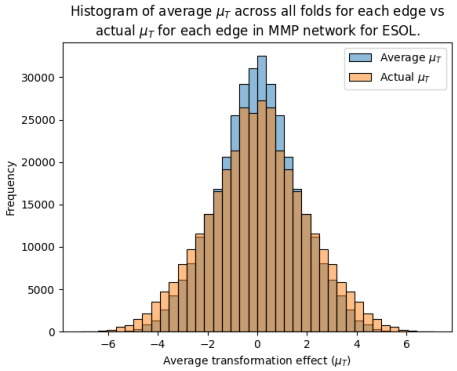
Dataset \ Fold	1	2	3	4	5	6	7	8	9	10
BACE	0.7	0.6	0.7	0.6	0.7	0.6	0.7	0.8	0.6	0.6
LIPO	0.8	0.8	0.8	0.7	0.7	0.8	0.8	0.8	0.8	0.8
ESOL	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
FREESOLV	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9



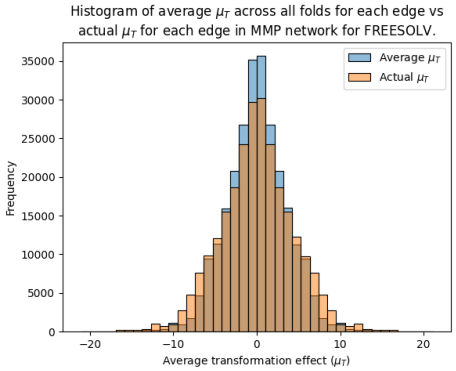
(a) Histogram of average μ_T across folds vs true μ_T for BACE MMP network.



(b) Histogram of average μ_T across folds vs true μ_T for LIPO MMP network.



(c) Histogram of average μ_T across folds vs true μ_T for ESOL MMP network.



(d) Histogram of average μ_T across folds vs true μ_T for FREESOLV MMP network.

Figure 6.1.: Histograms illustrating the discrepancy between the calculated μ_T values of edges averaged across all folds and the true μ_T values of the edges in the MMP network for MoleculeNet datasets.

6.2. Performance for MoleculeACE Dataset

Table 6.4.: Average RMSD values from the experiments on MoleculeACE dataset.

Model	CHEMBL218 (EC_{50})	CHEMBL219 (Ki)	CHEMBL234 (EC_{50})
NBS standalone			
in	0.74 ± 0.1	0.72 ± 0.05	0.62 ± 0.04
GNN trained with RMSD			
all	0.71 ± 0.04	0.74 ± 0.07	0.67 ± 0.05
in	0.68 ± 0.06	0.69 ± 0.06	0.61 ± 0.05
out	0.83 ± 0.1	0.9 ± 0.1	1.01 ± 0.1
GNN with NBS optimisation			
all	0.699 ± 0.05	0.73 ± 0.07	0.65 ± 0.04
in	0.665 ± 0.06	0.68 ± 0.06	0.59 ± 0.04
GNN with custom loss			
all	**0.693 ± 0.04**	**0.71 ± 0.06**	**0.64 ± 0.03**
in	**0.661 ± 0.08**	**0.65 ± 0.06**	**0.58 ± 0.04**
out	0.82 ± 0.1	**0.9 ± 0.1**	1.0 ± 0.09
Training augmentation			
all	0.72 ± 0.05	0.73 ± 0.07	1.0 ± 0.06
in	0.7 ± 0.06	0.68 ± 0.08	0.98 ± 0.07
out	**0.79 ± 0.2**	0.9 ± 0.2	1.14 ± 0.1
Test-time augmentation			
all	0.72 ± 0.06	0.74 ± 0.09	0.71 ± 0.05
in	0.69 ± 0.06	0.69 ± 0.09	0.67 ± 0.05
out	0.82 ± 0.1	0.9 ± 0.1	**0.98 ± 0.1**

The results for datasets in MoleculeACE are compared in Table 6.4. Across the selected datasets in MoleculeACE, the *custom loss* model demonstrates an improved performance on *all* instances compared to all other models, with CHEMBL219 exhibiting a 2.7 % performance boost against the next best model *GNN with NBS*. Analysing the RMSD values for the in-network instances, the *custom loss* model performs 4.4 % better than the *GNN with NBS* model and 5.8 % better than the *GNN with RMSD* model for the CHEMBL219 dataset. For CHEMBL234, predictions could be improved by 1.7 % and 4.9 % against the *GNN with NBS* and *GNN with RMSD* models, respectively. In the case of CHEMBL218, the performance improvement against the *GNN with NBS* and *GNN with RMSD* models are 0.6 % and 2.8 %, respectively.

Discussion. We observe that, for the *all* and *in* instances, the *custom loss* model outperforms all other models for the selected datasets from MoleculeACE. Furthermore, the optimal λ values shown in Table 6.6 suggest that higher prediction accuracy is achieved when the MMP network is involved. Additionally, as observed in Table 6.5, for the datasets in MoleculeACE, the difference in standard deviations between the μ_T values for different folds and the true μ_T values is minimal, implying that the μ_T values for the

edges are closer to their actual counterparts. Moreover, the number of edges with differing μ_T values is also lower, as illustrated in Fig. 6.2. These factors collectively indicate a greater contribution of the MMP network towards achieving improved prediction accuracy.

Table 6.5.: Descriptive statistics of MMP networks for MoleculeACE datasets

	CHEMBL218	CHEMBL219	CHEMBL234
Nodes	864	1610	3210
Edges	14350	27236	96192
Average red nodes	86.4	161.0	321.0
Average green nodes	777.6	1449.0	2889.0
STD of true μ_T of all edges	0.823	0.809	0.601
STD of average μ_T of all edges across folds	0.693	0.690	0.534

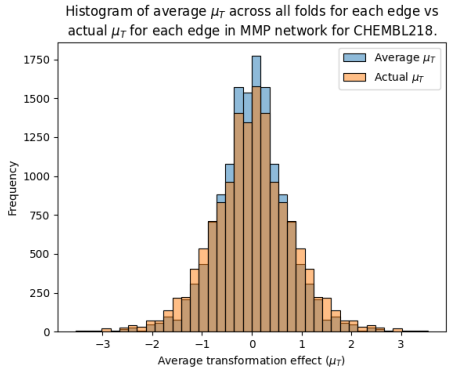
Table 6.6.: Optimal λ values for all folds of each dataset in MoleculeACE. Here, λ is the coefficient for the first loss term (to fit prediction) and $(1 - \lambda)$ is for the second term (to enforce transformation rules) as given in Eq. (4.2).

Dataset \ Fold	1	2	3	4	5	6	7	8	9	10
CHEMBL218	0.8	0.7	0.9	0.6	0.6	0.9	0.8	0.6	0.9	0.6
CHEMBL219	0.7	0.7	0.6	0.6	0.7	0.6	0.8	0.6	0.7	0.7
CHEMBL234	0.7	0.6	0.6	0.8	0.7	0.8	0.6	0.8	0.7	0.8

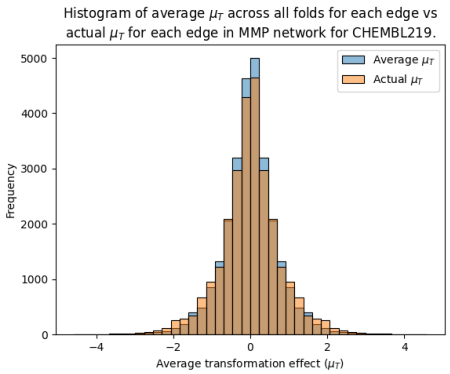
6.3. Performance Analysis and Discussion

Analysing the results from the experiments, we observe that the *custom loss* model successfully encapsulates the NBS methodology together with a GNN model, demonstrating on par or improved prediction accuracy on *all* instances across five datasets from both benchmark collections compared to *GNN with RMSD* and *GNN with NBS*. For these datasets, the *custom loss* model is able to take advantage of the relations in the MMP network to further improve prediction. Namely, the designed custom loss function incorporates the information from MMP transformations together with a GNN model, resulting in better predictions. However, it encounters challenges in performance on two datasets, namely ESOL and FREESOLV.

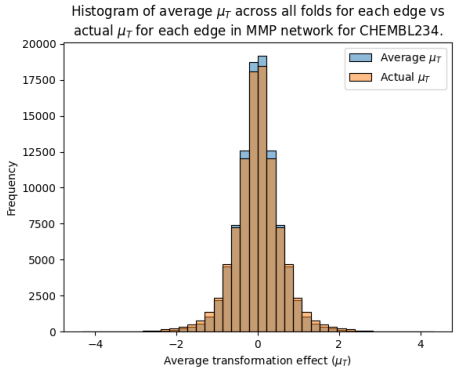
The design of the custom loss function facilitates the balance between the contributions of the two loss terms. The first loss term primarily fits predictions, typical of any GNN model, while the second loss term effectively enforces the transformation rules. Moreover, the second loss term allows, to a certain extent, discrepancies between the μ_T values in the MMP network to enforce the transformation rules. However, the substantial number of edges with μ_T values differing from the true μ_T values negatively impacts the contribution



(a) Histogram of average μ_T across folds vs true μ_T for CHEMBL218 MMP network.



(b) Histogram of average μ_T across folds vs true μ_T for CHEMBL219 MMP network.

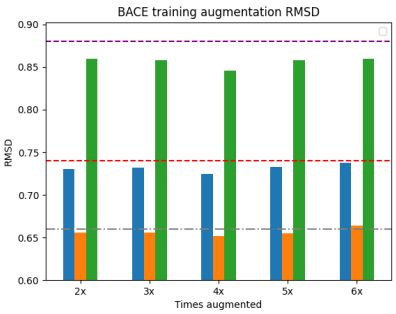


(c) Histogram of average μ_T across folds vs true μ_T for CHEMBL234 MMP network.

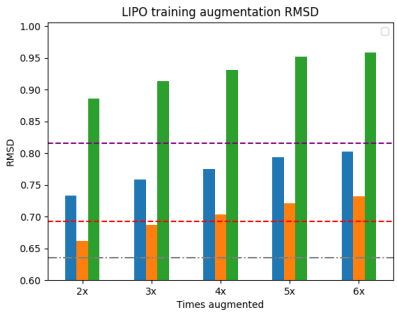
Figure 6.2.: Histograms illustrating the discrepancy between the calculated μ_T values of edges averaged across all folds and the true μ_T values of the edges in the MMP network for MoleculeACE datasets.

of the second term, as observed in the ESOL and FREESOLV datasets. Nonetheless, the *custom loss* model demonstrates improved prediction performance compared to the *GNN with NBS*, where predictions from the *GNN with RMSD* model are required to calculate the μ_T values. The μ_T values utilized for the *custom loss* model precisely match those used for *NBS standalone*, and in comparison, the *custom loss* model significantly outperforms it for the in-network instances. It is important to note that *NBS standalone* only works for in-network instances, as the approach relies on MMP relations.

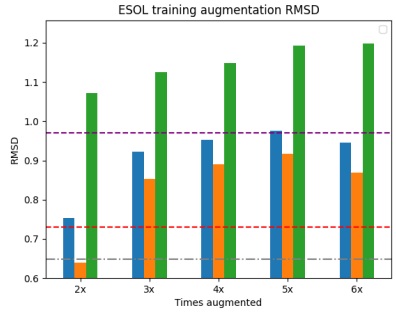
The performance of the two augmentation approaches varies for different datasets. We observe improvements only for the BACE and CHEMBL219 datasets with the *train augmentation* model as compared to the *GNN with RMSD* model. The augmentation approaches perform better only against the *NBS standalone* model for the in-network instances, except for CHEMBL219 dataset. Fig 6.3 illustrates the RMSD values for the different augmented training sets, where we observe that the prediction performance does not improve regardless of the increase in dataset size. The poor performance of both the augmentation models can be attributed to the random application of transformations to the molecules, coupled with the inaccuracy of the augmented labels with μ_T values. Finding available transformations that can augment all molecules within a training set can be challenging and commonly, the same set of transformations was possible and therefore used for augmentation. Additionally, only transformations with an occurrence of greater than two were used in augmentation to ensure that μ_T values were close to their true values.



(a) BACE

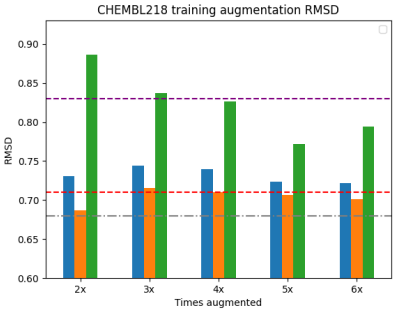


(b) LIPO

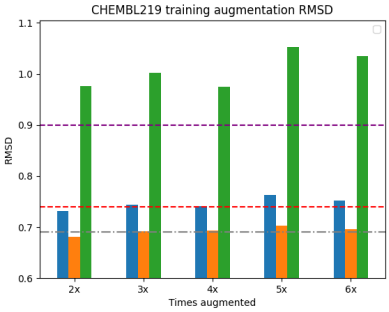


(c) ESOL

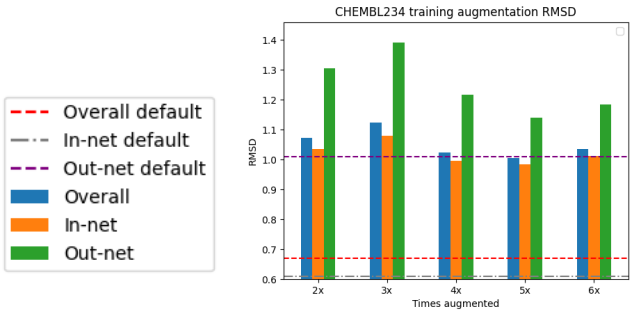
(d) FREESOLV



(e) CHEMBL218



(f) CHEMBL219



(g) CHEMBL234

Figure 6.3.: The bar charts illustrate the comparison of the RMSD values between the different augmented training data for the datasets in MoleculeNet and MoleculeACE collections. For each dataset, the training data was augmented from two to six times its size.

7. Conclusion

This thesis introduced an approach that incorporates the NBS methodology through MMP network optimisation, coupled with the GNN framework, to devise a custom loss function. We explored the concept of combining a GNN model with MMP relations with the aim of enhancing predictions. Furthermore, we presented a comprehensive review of the current relevant literature regarding the application of MMP methodology in property prediction. In this overview, we summarised how MMPs have been employed in various forms alongside ML methods across various prediction tasks, as well as to provide explanations for such predictions.

Drawing inspiration from the NBS model, we introduced our approach, that balances both fitting predictions and enforces transformation rules. We highlight the differences between the NBS methodology and our approach, particularly concerning the structure of the MMP network and the resulting optimisation problem.

We conducted experiments on seven datasets from two benchmark collections using a total of six models, employing the nested k-fold cross validation scheme. Our experimental results demonstrate that our approach performs on par or better for five datasets compared to all other models, particularly against the NBS approach. However, for the ESOL and FREESOLV datasets, our approach falls short. Subsequently, upon analysis of the MMP networks for all datasets, we observed that significant number of edges exhibiting a discrepancy in the μ_T values adversely affect performance.

In conclusion, our findings demonstrate that training a GNN using a custom loss function that leverages information from an MMP network successfully enhances prediction performance.

7.1. Future research

Several interesting possibilities emerge for future research based on the findings of this thesis. Three possible directions could be subjected to further investigation.

Foremost, during the initial phase of this thesis, the idea of a dynamic μ_T was discussed and investigated. However, initial experiments revealed that the GNN model encountered difficulties in convergence, therefore a static version was selected. From the analysis of the results for ESOL and FREESOLV, we have identified differing μ_T values for the edges as a key indicator of adverse performance. Following additional experiments shown in Table A.1, we can speculate that improvement in μ_T values towards its true value could lead to further improvement in predictions.

Another intriguing idea for further research could involve exploring the integration of mini-batching together with MMP networks. To capture the full potential of the

MMP networks, full batch mode had to be used requiring additional space and run-time. Mini-batching could aid in mitigating this factor, accelerating convergence. Although care must be taken as smaller batches could result in lack of or smaller MMP networks due to limited MMPs between instances within a batch.

Lastly, considering MMP networks themselves are graph structures, an ensemble architecture could be employed. In such an approach, one GNN model could focus on optimising the MMP network, while another focuses on fitting predictions.

Bibliography

[ARJ23] Kenza Amara, Raquel Rodríguez-Pérez and José Jiménez-Luna. "Explaining compound activity predictions with a substructure-aware loss for graph neural networks". In: *Journal of cheminformatics* 15.1 (2023), p. 67.

[ARK20] Mahendra Awale, Sereina Riniker and Christian Kramer. "Matched molecular series analysis for ADME property prediction". In: *Journal of chemical information and modeling* 60.6 (2020), pp. 2903–2914.

[Bag+21] Maryam Bagherian et al. "Machine learning approaches and databases for prediction of drug–target interaction: a survey paper". In: *Briefings in bioinformatics* 22.1 (2021), pp. 247–269.

[Dab+23] Markus Dablander et al. "Exploring QSAR models for activity-cliff prediction". In: *Journal of Cheminformatics* 15.1 (2023), p. 47.

[DH13] Andrew Dalke and Janna Hastings. "FMCS: a novel algorithm for the multiple MCS problem". In: *Journal of cheminformatics* 5.Suppl 1 (2013), O6.

[DHK18] Andrew Dalke, Jerome Hert and Christian Kramer. "mmpdb: An open-source matched molecular pair platform for large multiproperty data sets". In: *Journal of chemical information and modeling* 58.5 (2018), pp. 902–910.

[Dua+23] Yan-Jing Duan et al. "Improved GNNs for Log D 7.4 Prediction by Transferring Knowledge from Low-Fidelity Data". In: *Journal of Chemical Information and Modeling* 63.8 (2023), pp. 2345–2359.

[FL19] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[Fu+19] Li Fu et al. "Systematic Modeling of log D 7.4 Based on Ensemble Machine Learning, Group Contribution, and Matched Molecular Pair Analysis". In: *Journal of Chemical Information and Modeling* 60.1 (2019), pp. 63–76.

[Gil+17] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272.

[Gri+11] Ed Griffen et al. "Matched molecular pairs as a medicinal chemistry tool: miniperspective". In: *Journal of medicinal chemistry* 54.22 (2011), pp. 7739–7750.

[Gri23] Francesca Grisoni. "Chemical language models for de novo drug design: Challenges and opportunities". In: *Current Opinion in Structural Biology* 79 (2023), p. 102527.

[HB07] David Y Haubertin and Pierre Bruneau. “A database of historically-observed chemical replacements”. In: *Journal of chemical information and modeling* 47.4 (2007), pp. 1294–1302.

[He+21] Jiazhen He et al. “Molecular optimization by capturing chemist’s intuition using deep neural networks”. In: *Journal of cheminformatics* 13.1 (2021), pp. 1–17.

[Hön+22] Sophia Hönig et al. “Network Balance Scaling Improves Performance and Explainability of Machine Learning Models in Early Drug Discovery”. In: (2022).

[HR10] Jameed Hussain and Ceara Rea. “Computationally efficient algorithm to identify matched molecular pairs (MMPs) in large data sets”. In: *Journal of chemical information and modeling* 50.3 (2010), pp. 339–348.

[HS08] Philip J Hajduk and Daryl R Sauer. “Statistical analysis of the effects of common chemical substituents on ligand potency”. In: *Journal of medicinal chemistry* 51.3 (2008), pp. 553–564.

[Hu+12] Xiaoying Hu et al. “MMP-cliffs: systematic identification of activity cliffs on the basis of matched molecular pairs”. In: *Journal of chemical information and modeling* 52.5 (2012), pp. 1138–1145.

[IVB21a] Javed Iqbal, Martin Vogt and Jürgen Bajorath. “Learning functional group chemistry from molecular images leads to accurate prediction of activity cliffs”. In: *Artificial Intelligence in the Life Sciences* 1 (2021), p. 100022.

[IVB21b] Javed Iqbal, Martin Vogt and Jürgen Bajorath. “Prediction of activity cliffs on the basis of images using convolutional neural networks”. In: *Journal of Computer-Aided Molecular Design* (2021), pp. 1–8.

[Kra+14] Christian Kramer et al. “Matched molecular pair analysis: significance and the impact of experimental uncertainty”. In: *Journal of medicinal chemistry* 57.9 (2014), pp. 3786–3802.

[Kra+17] Christian Kramer et al. “Learning medicinal chemistry absorption, distribution, metabolism, excretion, and toxicity (ADMET) rules from cross-company matched molecular pairs analysis (MMPA) miniperspective”. In: *Journal of medicinal chemistry* 61.8 (2017), pp. 3277–3292.

[Kun23] Arjun Reddy Kunduru. “Machine Learning in Drug Discovery: A Comprehensive Analysis of Applications, Challenges, and Future Directions”. In: *International Journal on Orange Technologies* 5.8 (2023), pp. 29–37.

[Lan] G. Landrum. *RDKit: Open-source cheminformatics.* URL: `https://www.rdkit.org/` (visited on 2024).

[Lea+06] Andrew G Leach et al. “Matched molecular pairs as a guide in the optimization of pharmaceutical properties; a study of aqueous solubility, plasma protein binding and oral exposure”. In: *Journal of medicinal chemistry* 49.23 (2006), pp. 6672–6682.

[Lon+22] Teng-Zhi Long et al. “Structural Analysis and Prediction of Hematotoxicity Using Deep Learning Approaches”. In: *Journal of Chemical Information and Modeling* 63.1 (2022), pp. 111–125.

[Lum+20] James A Lumley et al. “The derivation of a matched molecular pairs based ADME/Tox knowledge base for compound optimization”. In: *Journal of Chemical Information and Modeling* 60.10 (2020), pp. 4757–4771.

[MC19] Adam C Mater and Michelle L Coote. “Deep learning in chemistry”. In: *Journal of chemical information and modeling* 59.6 (2019), pp. 2545–2559.

[OBo+14] Noel M O’Boyle et al. “Using matched molecular series as a predictive tool to optimize biological activity”. In: *Journal of Medicinal Chemistry* 57.6 (2014), pp. 2704–2713.

[Pap+10] George Papadatos et al. “Lead optimization using matched molecular pairs: inclusion of contextual information for enhanced prediction of HERG inhibition, solubility, and lipophilicity”. In: *Journal of chemical information and modeling* 50.10 (2010), pp. 1872–1886.

[Par+22] Junhui Park et al. “ACGCN: graph convolutional networks for activity cliff prediction between matched molecular pairs”. In: *Journal of Chemical Information and Modeling* 62.10 (2022), pp. 2341–2351.

[Pas+12] A Paszke et al. “An imperative style, high-performance deep learning library”. In: *Adv. Neural Inf. Process. Syst* 32 (1912), p. 8026.

[RB22] Raquel Rodríguez-Pérez and Jürgen Bajorath. “Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery”. In: *Journal of Computer-Aided Molecular Design* 36.5 (2022), pp. 355–362.

[Stu+14] Dagmar Stumpfe et al. “Recent progress in understanding activity cliffs and their utility in medicinal chemistry: miniperspective”. In: *Journal of medicinal chemistry* 57.1 (2014), pp. 18–28.

[TAG22] Derek van Tilborg, Alisa Alenicheva and Francesca Grisoni. “Exposing the limitations of molecular machine learning with activity cliffs”. In: *Journal of Chemical Information and Modeling* 62.23 (2022), pp. 5938–5951.

[Tam+21] Shunsuke Tamura et al. “Interpretation of ligand-based activity cliff prediction models using the matched molecular pair kernel”. In: *Molecules* 26.16 (2021), p. 4916.

[TE17] Christian Tyrchan and Emma Evertsson. “Matched molecular pair analysis in short: algorithms, applications and limitations”. In: *Computational and structural biotechnology journal* 15 (2017), pp. 86–90.

[TMB23] Shunsuke Tamura, Tomoyuki Miyao and Jürgen Bajorath. “Large-scale prediction of activity cliffs using machine and deep learning methods of increasing complexity”. In: *Journal of Cheminformatics* 15.1 (2023), p. 4.

[Tur+17] Samo Turk et al. “Coupling matched molecular pairs with machine learning for virtual compound optimization”. In: *Journal of Chemical Information and Modeling* 57.12 (2017), pp. 3079–3085.

[Van+23] Sarveswara Rao Vangala et al. “pbrics: A novel fragmentation method for explainable property prediction of drug-like small molecules”. In: *Journal of Chemical Information and Modeling* 63.16 (2023), pp. 5066–5076.

[VB14] Antonio de la Vega de León and Jürgen Bajorath. “Prediction of compound potency changes in matched molecular pairs using support vector regression”. In: *Journal of Chemical Information and Modeling* 54.10 (2014), pp. 2654–2663.

[War+12] Daniel J Warner et al. “Prospective prediction of antitarget activity by matched molecular pairs analysis”. In: *Molecular Informatics* 31.5 (2012), pp. 365–368.

[WB10] Anne Mai Wassermann and Jürgen Bajorath. “Chemical substitutions that introduce activity cliffs across different compound classes and biological targets”. In: *Journal of chemical information and modeling* 50.7 (2010), pp. 1248–1256.

[WB11] Mathias Wawer and Jürgen Bajorath. “Local structural changes, global data views: graphical substructure- activity relationship trailing”. In: *Journal of medicinal chemistry* 54.8 (2011), pp. 2944–2951.

[Wei88] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[WGS10] Daniel J Warner, Edward J Griffen and Stephen A St-Gallay. “WizePairZ: a novel algorithm to identify, encode, and exploit matched molecular pairs with unspecified cores in medicinal chemistry”. In: *Journal of chemical information and modeling* 50.8 (2010), pp. 1350–1357.

[Wie+20] Oliver Wieder et al. “A compact review of molecular property prediction with graph neural networks”. In: *Drug Discovery Today: Technologies* 37 (2020), pp. 1–12.

[Wu+18] Zhenqin Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. In: *Chemical science* 9.2 (2018), pp. 513–530.

[Xu+19] Keyulu Xu et al. “How Powerful are Graph Neural Networks?” In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Yan+23] Ziyi Yang et al. “Matched Molecular Pair Analysis in Drug Discovery: Methods and Recent Applications”. In: *Journal of Medicinal Chemistry* 66.7 (2023), pp. 4361–4377.

[Zha+21] Xiao-Meng Zhang et al. “Graph neural networks and their current applications in bioinformatics”. In: *Frontiers in genetics* 12 (2021), p. 690049.

[Zha+23] Ziqiao Zhang et al. “Activity Cliff Prediction: Dataset and Benchmark”. In: *arXiv preprint arXiv:2302.07541* (2023).

Acronyms

AC Activity Cliff. 1, 9, 13, 14

ADMET Absorption, Distribution, Metabolism, Excretion, and Toxicity. 1, 7

CGR Condensed Graph of Reaction. 13

CNN Convolutional Neural Network. 13

DL Deep Learning. 7

DNN Deep Neural Networks. 12

DTIs Drug-Target Interactions. 13

FMCS Flexible Maximum Common Substructure. 5

GBM Gradient Boosting Machines. 12

GCN Graph Convolutional Network. 13

GIN Graph Isomorphism Network. 15

GNN Graph Neural Network. iii, vii, 1, 2, 9, 13–16, 18, 22–26, 30, 35, 36, 45–47

MCS Maximum Common Subgraph. vii, 3, 5, 6, 14

ML Machine Learning. iii, vii, 1, 3, 7, 9, 12, 14, 15, 35

MLP Multi-Layer Perceptron. 15

MMP Matched Molecular Pair. iii, vii, ix, xi, 1–7, 9, 10, 12–20, 22–24, 26–32, 35, 36, 46

MMPA MMP Analysis. vii, 6, 9, 12, 14

MMS Matched Molecular Series. 12

MMSA MMS Analysis. 12

NBS Network Balance Scaling. iii, vii, xi, 1, 2, 9–11, 18, 19, 23–25, 30, 35

RECS Remainder after Elimination of Common Substructure. 6

Acronyms

RF Random Forests. 7, 9, 12

RMSD Root Mean Square Deviation. ix, 23, 25, 26, 29, 32, 33, 45

SAR Structure-Activity Relationship. 3, 12, 13

SMARTS SMILES arbitrary target Specification. 4, 5

SMILES Simplified Molecular-Input Line-Entry System. 4, 5, 45

SMIRKS SMILES arbitrary target specification for reaction kernels. 4–6, 45

SPR Structure-Property Relationship. 3

SVM Support Vector Machines. 7, 14

SVR Support Vector Regression. 12

A. Appendix

A.1. Training and Test-time Augmentation

To generate an augmented molecule, a molecule is transformed using a transformation rule, wherein a matching substructure from the left-hand side of a SMIRKS string is substituted by the corresponding substructure from the right-hand side. This process is accomplished using various functions available in the RDKit library [Lan].

To construct augmented training and test set, transformations from a dataset with an occurrence greater than two were utilised. These transformations were randomly chosen and applied to each molecule until a viable augmented molecule was generated. Since multiple viable augmented molecules can stem from a single transformation, all viable augmented molecules were included in the new augmented training set. However, for the test set, only the first viable augmented molecule was selected and added to one of the five augmented test sets. Additional transformations were then applied to generate augmented molecules for the other test sets. If a molecule could not be augmented, the same molecule was added to all five test sets. Augmented molecules with the same SMILES as any original molecules were discarded. The labels for these augmented molecules were also adjusted, with the μ_T values simply added to their original labels. It should be noted that transformations and their activity differences from molecules in the test set were not taken into account when calculating μ_T. Furthermore, five augmented training sets were created, with each set increased in size between two to six times. This procedure was done for all folds of a dataset.

A.2. Additional experiments

Table A.1.: Average RMSD values for experiments on ESOL, FREESOLV and CHEMBL219 datasets using the true μ_T values of the edges.

`Model`	`ESOL (esol)`	`FREESOLV (expt)`	`CHEMBL219 (`K_i`)`
`GNN with custom loss`	all - 0.61 ± 0.07 in - 0.45 ± 0.04 out - 1.04 ± 0.1	all - 0.87± 0.2 in - 0.65 ± 0.1 out - 2.03 ± 1.05	all - 0.54 ± 0.05 in - 0.40 ± 0.05 out - 1.03 ± 0.1

Table A.1 presents the RMSD values obtained by averaging the results of the 10-fold split. This experiment was conducted to assess the performance of the GNN model when utilizing the true μ_T values of the edges in the custom loss function. Here, the true μ_T values also consider the transformations from the test set instances ($\mathcal{V}_{\text{red}}$), thereby

Table A.2.: Optimal λ values for all folds for datasets when using the true μ_T values of the edges.

Dataset \ Fold	1	2	3	4	5	6	7	8	9	10
ESOL	0.7	0.9	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
FREESOLV	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
CHEMBL219	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6

providing complete information about all transformations in a dataset. Additionally, Table A.2 indicates that the MMP network is utilized more, as the λ values shift more towards the second term.

A.3. Discard Approaches

A.3.1. Version 1

$$\mathcal{L} = \overbrace{\sum_{i=1}^{N} \left(f(x^{(i)}) - y_{\text{true}}^{(i)} \right)^2}^{\text{Fit prediction}} + \overbrace{\sum_{T \in \tau} \sum_{(i,j) \in \mathcal{E}_T} \left(\left| f(x^{(j)}) - f(x^{(i)}) - d_T \right| \right)}^{\text{Enforce transformation rules}} \tag{A.1}$$

Here, $f(x^{(i)})$ is the prediction made by a GNN model on a molecular graph $x^{(i)}$ and $y_{\text{true}}^{(i)}$ is the true label of $x^{(i)}$. τ is the collection of all the unique transformations and $\mathcal{E}_T$ are the edges associated with a transformation T. N is the total number of instances in the training set and d_T is a scalar for each T that is optimised alongside the parameter for the GNN model.

The idea behind the second term for Eq. (A.1) was that, given a transformation T and an edge $(i, j) \in \mathcal{E}_T$, the activity difference $f(x^{(j)}) - f(x^{(i)})$ should be close to the parameter d_T. However, through experiments, we observed that the loss function converged primarily due to the influence of the first loss term. When only the second loss term was utilized, the GNN model encountered challenges in convergence, despite extensive training over two thousand epochs and adjustments to various hyperparameters, including the learning rate.

A.3.2. Version 2

$$\mathcal{L} = \overbrace{\sum_{i=1}^{N} \left(f(x^{(i)}) - y_{\text{true}}^{(i)} \right)^2}^{\text{Fit prediction}} + \overbrace{\sum_{T \in \tau} \sum_{(i,j) \in \mathcal{E}_T} \left(h(x^{(j)}) - h(x^{(i)}) - \mu_T \right)^2}^{\text{Enforce transformation rules}} \tag{A.2}$$

$$h(x^{(i)}) = \begin{cases} y_{\text{true}}^{(i)}, & \text{if } x^{(i)} \in \mathcal{V}_{\text{green}} \\ \\ f(x^{(i)}), & \text{if } x^{(i)} \in \mathcal{V}_{\text{red}}. \end{cases} \tag{A.3}$$

Here, $f(x^{(i)})$ is the prediction made by a GNN model on a molecular graph $x^{(i)}$ and $y_{\text{true}}^{(i)}$ is the true label of $x^{(i)}$. τ is the collection of all the unique transformations and $\mathcal{E}_T$ are the edges associated with a transformation T. N is the total number of instances in the training set and μ_T is the average activity difference calculated for a transformation T.

After replacing the scalar term d_T in Eq. (A.1) with μ_T in Eq. (A.2), and using a sum of squares instead of absolutes, we conducted further experiments using only the second loss term. We found that the GNN model was able to converge, attributed to the inclusion of the μ_T term for transformations. However, despite this convergence, the prediction performance remained subpar, with prediction accuracy falling significantly short to those of the *GNN with RMSD* model for in-network instances. Following analysis, we identified two problematic areas. The first issue was the difference in magnitude between the two loss terms, with the second loss term having considerably larger magnitude than the first. The second issue was that the two individual loss terms contributed equally to the overall loss. In light of these findings, we made improvements and formulated our final loss function as presented in Eq. (4.2).