MULTI-DATASET MULTI-TASK FRAMEWORK FOR LEARNING MOLECULES AND PROTEIN-TARGET INTERACTIONS PROPERTIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular property prediction and protein-target interaction prediction with deep learning are becoming increasingly popular in drug discovery pipelines in recent years. An important factor that limits the development of these two areas is the insufficiency of labeled data. One promising direction to address this problem is to learn shared embedding from multiple prediction tasks within one molecular type, *e.g.,* molecule or protein, because different tasks might actually share similar coarse-grained structural information. Unlike the previous methods, in this paper, we first argue that, due to the possible local structural similarity between molecules and protein-target complexes, coarse-grained latent embeddings can be found across different molecular types. To take advantage of this, we propose a new Multi-Dataset Multi-Task Graph Learning (MDMT-GL) framework, where we are able to make the most use of the labeled data by simultaneously training molecule property prediction and protein-target interaction prediction together. MDMT-GL augments molecular representations with equivariant properties, 2D local structures, and 3D geometric information. MDMT-GL can learn coarse-grained embeddings for molecules and proteins, and also distinguish fine-grained representations in various downstream prediction tasks with unique characteristics. Experimentally, we implement and evaluate MDMT-GL on 2 molecular dynamic datasets and 2 protein-target datasets, consisting of 825 tasks and over 3 million data points. MDMT-GL achieves state-of-the-art performance on several tasks and shows competitive performance on others. These experimental results confirm that molecules and proteins indeed share some coarse-grained structures and that the coarse-grained embedding is trainable, and their fine-grained embeddings are more representative. To the best of our knowledge, this is the first work to train multi-task learning across different molecular types, and to verify the structural similarity between the molecules and the protein-target complexes.

1 INTRODUCTION

The discovery and development of a new drug could take more than a decade and cost billions of dollars Hughes et al. (2011); Sliwoski et al. (2014). Therefore, to reduce costs, predicting the properties of molecules and protein-target complexes (*e.g.,* heat capacity, force field, binding affinity) become an essential component for the early stage of the drug discovery pipeline. Molecules and complexes are always represented as graph-structured data Li et al. (2021); Maziarka et al. (2020); Thölke & De Fabritiis (2022), where atoms and bonds are nodes and edges, respectively, and graph neural networks are in favor of learning representations from relational datasets Kipf & Welling (2016); Luan et al. (2021); Hua et al. (2022). As a result, graph-based deep learning methods that learn molecular graph representations have achieved great success in predicting molecule properties Schütt et al. (2018; 2021); Klicpera et al. (2020); Thölke & De Fabritiis (2022) and protein-target interactions Lim et al. (2019), but the data we have at hand are often insufficient, which will limit model performance Sliwoski et al. (2014); Liu et al. (2022). Thus, reducing the requirement for labeled data needed for the effective prediction of molecular and protein target properties becomes a challenge in drug discovery.

To address the aforementioned issue, multi-task learning for molecular property prediction Tan et al. (2021) and protein-target interaction prediction Lee & Kim (2019); Hu et al. (2021); Liu et al. (2022) is gradually drawing attention from the drug discovery community. Their models always deal with a single molecular type, *i.e.,* a molecule (or complex) is only used for multiple molecule property (or protein-target interaction) prediction tasks. The difficulty stems from the fact that knowledge from different molecular types cannot be easily decomposed and shared. However, we argue that due to the internal geometric and local structural similarities between the molecule and the protein-target complex, they should share similar coarse-grained latent embeddings Jain (2000); Bender & Glen (2004); Löfblom et al. (2010). Hence, we believe that representations of molecules and complexes could be coarse-grained and a coarse-grained latent embedding could be learned together under one learning framework. Embodiments should share internal geometric and local structural information across molecules and complexes from atomic perspectives. Eventually, the learning of protein representations can benefit from the learning of molecule representations, and vice versa.

Therefore, we propose a new learning framework, Multi-Dataset Multi-Task Graph learning (MDMT-GL) for molecular property prediction and protein-target interaction prediction. MDMT-GL aims to make the best use of labeled data by transferring knowledge between molecules and complexes. The cross-dataset paradigm for multi-task learning enables the shared embedding to be more informative representations than the single-dataset paradigm. *To the best of our knowledge, MDMT-GL is the first work to train molecular property prediction and protein-target interaction prediction together and to verify the structural similarities between the molecule and the protein-target complex.* In addition to the major contribution, we also develop the 2D graph transformer proposed by Kim et al. (2021) into a 3D equivariant graph transformer for molecular dynamics, and the model is capable of capturing high-order atom interactions in 3D space. Moreover, unlike multi-task learning within a single dataset, the data imbalance of different datasets will lead to the task imbalance problem which is fatal to multi-task learning. To treat each task equally, we propose a weighted loss to balance the importance of the tasks, which is novel for MDMT-GL. The details of MDMT-GL are discussed in Sec. 3. Furthermore, in Sec. 4, the experimental results support our argument and show that molecules and complexes can share some similar coarse-grained structures, and the geometric and structural similarities can be learned to leverage any molecular prediction task.

2 Related Work

2.1 Molecular Multi-task Learning

Molecular Multi-Task Learning (MTL) is mainly used to address the data insufficiency problem in drug discovery. Liu et al. (2019c) uses a general architecture of a shared representation module and multiple task-specific prediction modules for MTL. Tan et al. (2021) stacks a base regressor and classifier with an additional training stage on the expanded molecular feature space for the prediction of molecular properties. Lee & Kim (2019) finds that similarity within a target group can affect the performance of MTL in the prediction of protein binding. Liu et al. (2022) possesses the knowledge of task relations and constructs a task-relation graph to maximize the performance of MTL in protein targeting. However, the aforementioned methods do not transfer knowledge between molecules and protein-target complexes. Existing models only perform MTL on the same dataset, *i.e.,* molecule or protein, but the MTL between molecule and protein has never been explored. In this work, we aim to make use of the shared information between molecules and proteins across various tasks, so that we can make the most and best use of the labeled data.

2.2 Graph Neural Networks for Property Prediction

In drug discovery, people apply message-passing-based models to predict the properties of molecules and proteins. Schütt et al. (2018) respects essential quantum chemical constraints and models quantum interactions by modeling interactions of atoms at arbitrary positions in a molecule. Satorras et al. (2021) proposes a graph neural network, which is equivariant to rotations, translations, reflections, and permutations in 3D geometry, to model molecular dynamics. Thölke & De Fabritiis (2022) builds on top of the graph transformer and develops an equivariant graph transformer to predict quantum molecule properties. Lim et al. (2019) learns drug-target interactions by extracting the graph features of intermolecular interactions directly from 3D structural information on the protein-ligand binding pose. Li et al. (2021) proposes a structure-aware interactive graph neural network to preserve the

distance and angle information among atoms to learn interactions between proteins and ligands. Overall, our architecture mainly consists of two equivariant graph transformers that focus on long-range atom interactions and featurization of atomic types and coordinates, and a graph neural network to preserve local structure information.

3 MULTI-DATASET MULTI-TASK FRAMEWORK FOR LEARNING MOLECULES AND PROTEIN-TARGET COMPLEXES

As discussed in Sec. 1, the labeled data for molecules and protein-target complexes are often insufficient. Therefore, we strive to make the most of the available labeled data from molecule and protein datasets for various tasks. In other words, we aim to design an architecture that can learn simultaneously from different molecular and protein datasets, in which learning protein representations can benefit from learning molecule representations and vice versa. *The core technical difficulty is how to identify their coarse-grained similar internal geometry and local structures, and to also differentiate their fine-grained representations for different conformation structures*.

To achieve the goal, we divide our model into four components (1) a coarse-grained module, (2) a fine-grained data-specific module, (3) a task-specific prediction module, and (4) a multi-dataset multi-task loss (see the whole architecture in Fig. 1 and App. A).

The function of each module is as follows: (1) The coarse-grained module is designed to learn a coarse-grained representation of molecules and protein-target complexes. Common geometric and structural information can be obtained in molecules and complexes can be obtained. We will discuss the details in Sec. 3.1. (2) The fine-grained module will process the molecules-specific and complexes-specific representations separately. We will discuss it in Sec. 3.2. (3) Then, the data-type-specific representations are fed into different task-specific prediction modules to make predictions for various tasks, the details are discussed in Sec. 3.3. (4) Finally, weighted losses of all tasks are used to balance the importance of different tasks. We describe how to compute the MDMT loss in Sec. 3.4. The whole framework can be trained in an end-to-end manner. In Sec. 4, we experimentally show that the representations could be coarse-grained between molecules and protein-target complexes.

3.1 COARSE-GRAINED MODULE

Although having different conformation structures and dynamics, molecules and protein-target complexes are made of basic atoms and bonds, and should thus share fundamental internal geometric and local structural information Jain (2000); Bender & Glen (2004); Löfblom et al. (2010). For example, the carbon dioxide molecule **O=C=O** and methanoic acid **H(C=O)OH** have different conformation structures and different force fields, but they share the same carbon atom **C** and similar local structures around the carbon atoms, *e.g.,* double bond with oxygen **O**. Thus, two carbon atoms could potentially share coarse-grained information about their local structures. The coarse-grained module is designed to capture such atomic-level similarities so that generalizable features between molecules and proteins can be learned.

To capture the atomic-level similarities, we give each basic atom a unique learnable embedding Schütt et al. (2018); Klicpera et al. (2020); Thölke & De Fabritiis (2022), which is shared by all compounds in all tasks across different datasets (see the atom-wise embedding layer in Fig. 1).This is the first time that the atomic-level coarse-grained representations are exploited in the MDMT setting for molecules and proteins. To be more specific, an input molecule or complex $\boldsymbol{m} = [a_1, a_2, ..., a_{N_m}]^T \in \mathbb{N}^{N_m \times 1}$ is a 1D vector of the atoms that build m, where N_m is the number of atoms in m, a_i is the number of atoms in the periodic table. The molecular embedding is $\mathbf{z}_m = f_{\text{atom}}(\boldsymbol{m})$, where $f_{\text{atom}} : \mathbb{N}^{N_m \times 1} \rightarrow \mathbb{R}^{N_m \times d}$ projects a 1D molecule vector onto a 2D learnable embedding, where each row of the embedding represents a hidden atom feature, and d is the dimension of the embedding space. Take the carbon dioxide molecule **O=C=O** for example, its input is a 1D vector representation $[8, 6, 8]^T$, where 8 and 6 are the number of atoms of oxygen and carbon in the periodic table, and its embedding follows $\mathbf{z}_{\mathbf{O=C=O}} = [f_{\mathbf{O}}(8), f_{\mathbf{C}}(6), f_{\mathbf{O}}(8)]^T \in \mathbb{R}^{3 \times d}$, where $f_{\mathbf{C}}(6), f_{\mathbf{O}}(8)$ are the learnable embeddings for carbon and oxygen, respectively.

2D molecular local structures, 3D molecular geometric information, and equivariant property are important for coarse-grained representations to preserve physical constraints Löfblom et al. (2010);

Schütt et al. (2021). Therefore, to obtain the above capacities, we augment the coarse-fined representation $\mathbf{z}_m$ by the following augmentation network f_{aug}, which can be an equivariant graph neural network Satorras et al. (2021); Schütt et al. (2021); Thölke & De Fabritiis (2022). The augmentation network f_{aug} takes $\mathbf{z}_m$, edge (bond) indices $\boldsymbol{e}_m \in [0,1]^{N_m \times N_m}$, edge (bond) features $\boldsymbol{f}_m \in \mathbb{R}^{E_m \times f_e}$ and atom positions $\boldsymbol{r}_m \in \mathbb{R}^{N_m \times 3}$ as input, and produces $\hat{\mathbf{z}}_m = f_{\text{aug}}(\mathbf{z}_m, \boldsymbol{r}_m, \mathbf{e}_m, \boldsymbol{f}_m) \in \mathbb{R}^{N_m \times d}$, which is an equivariant coarse-fined representation (see the augmentation network in Fig. 1). This design enables $\hat{\mathbf{z}}_m$ to learn the shared fundamental internal geometric and structural information across different tasks and datasets while preserving equivariant property.

In conclusion, the coarse-grained module consists of two components: (1) an atom-wise embedding layer and (2) an augmentation network. Atom-wise embedding layer is used to obtain an atom-wise coarse-grained representation $\mathbf{z}_m$ for every input molecule or complex $\boldsymbol{m}$, and the augmentation network augments every coarse-grained representation with equivariant property by 2D local structures and 3D geometric information to produce an equivariant coarse-grained representation $\hat{\mathbf{z}}_m$.

In addition to the equivariant coarse-grained representations, different molecular types require fine-grained data-type specific representations to capture differences in conformation structure and geometric information for performing different downstream tasks. In Sec. 3.2, we will introduce the fine-grained data-specific module and discuss the initiative to have it.

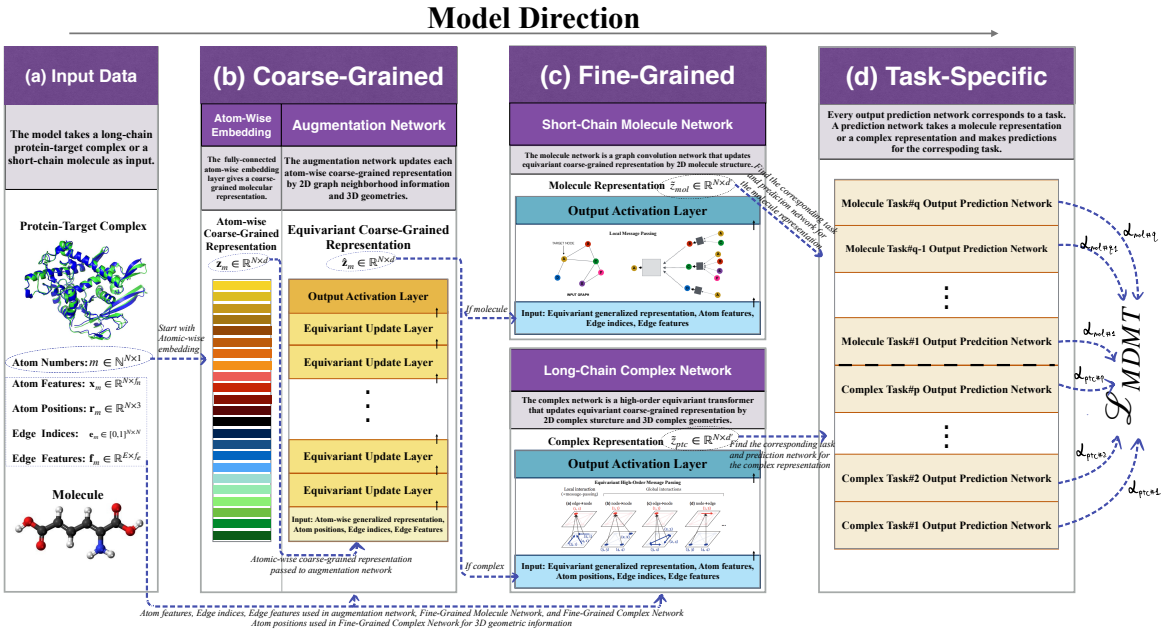

Figure 1: Overview of the Multi-Dataset Multi-Task Graph Learning framework (MDMT-GL) for concurrently learning representations of molecules and protein-target complexes. The figure demonstrates the direction of the MDMT-GL framework from left to right. MDMT-GL can be divided into five components which include (a) input data, (b) coarse-grained module, (c) fine-grained data-specific module, (d) task-specific prediction module, and (f) multi-dataset multi-task loss. The input data are molecules and protein-target complexes $\mathbf{m}$ with their 2D local structures (atom features $\mathbf{x}_m$, edge indices $\mathbf{e}_m$, edges features $\mathbf{f}_m$) and 3D geometry ($\mathbf{r}_m$). In the coarse-grained module, an input object will be embedded by an atom-wise embedding layer and augmented by the augmentation network. Molecules and protein-target complexes will share principal features, position, and structure information in coarse-grained embeddings. Then, we distinguish the difference between molecules and complexes in fine-grained data-specific modules. If the object is originally a molecule, it will get processed by the short-chain molecule network; or if the object is originally a complex, it will get processed by the long-chain complex network. Then, the processed item will be fed into its corresponding prediction network for a task prediction. All the task losses will be weighted and aggregated to a multi-dataset multi-task loss to balance all datasets and tasks for optimization. The architecture details are discussed in Sec. 3.

3.2 FINE-GRAINED DATA-SPECIFIC MODULE

Previously in Sec. 3.1, we discuss how the coarse-grained module can learn atom-wise atom-wise coarse-grained representations to utilize the use of labeled molecules and complexes. And we discuss the initiative and reason to make coarse-grained representations fine-grained for downstream uses.

The chain of a protein-target complex (normally from 100 to more than 1000 atoms) is always significantly longer than the chain of a molecule (normally from 1 to 60 atoms), thus making atom-wise interactions highly different, *i.e.,* two atoms might be farther away in a long chain. They could potentially interact, and the high-order long-range interactions always exist, which should be captured between atoms in a complex but are not solid and required for molecules Luan et al. (2019); Morris et al. (2019). For example, oxidoreductase $\mathbf{C}_{879}\mathbf{H}_{1426}\mathbf{N}_{250}\mathbf{O}_{260}\mathbf{S}_3$ is a protein of 2818 atoms while carbon dioxide $\mathbf{CO}_2$ is a molecule that has only 3 atoms.

With this in mind, to distinguish the different conformation structures resulting from the chain-size difference between molecules and complexes, in the fine-grained data-specific module, we process coarse-grained representations $\hat{\mathbf{z}}_m$ of molecules and complexes in different ways. To be more specific, we use high-order graph networks for large graphs Morris et al. (2019) like complexes to capture high-order interactions, and shallow graph networks for small graphs like molecules where high-order interactions are not solid Luan et al. (2019).

Therefore, we divide our fine-grained module into two data-specific networks, (1) a fine-grained complex network f_{ptc} that has the ability to capture high-order long-range interactions for atoms in complexes (see the long-chain complex network in Fig. 1), and (2) a shallow fine-grained molecule network f_{mol} for molecules (see the short-chain molecule network in Fig. 1).

The fine-grained complex network f_{ptc} can be any high-order graph neural network Li et al. (2021); Kim et al. (2021); Thölke & De Fabritiis (2022). We adopt and develop the 2D high-order transformer Kim et al. (2021) to a 3D equivariant transformer (see App. A), our fine-grained complex network f_{ptc} is capable of capturing **any-order** atom interactions and preserving equivariant property, which is novel. The fine-grained protein-target complex embedding follows $\tilde{\mathbf{z}}_{\text{ptc}} = f_{\text{ptc}}(\hat{\mathbf{z}}_m, \boldsymbol{r}_m, \mathbf{e}_m, \boldsymbol{f}_m, \boldsymbol{x}_m) \in \mathbb{R}^{N_m \times d'}$, where $\boldsymbol{x}_m \in \mathbb{R}^{N_m \times f_n}$ is atom features and d' denotes the dimension of the embedding.

For the fine-grained molecule network f_{mol}, the idea is fairly easy. Since equivariant property is closely related to high-order long-range interactions in 3D space Satorras et al. (2021), which is not required in small molecule graphs, we only need a shallow graph neural network as the fine-grained molecule network f_{mol} to model local message passing in short-chain molecules Kipf & Welling (2016); Luan et al. (2020); Hua et al. (2022). And considering the computational cost, we choose the simplest graph convolutional network Kipf & Welling (2016) for f_{mol}. The fine-grained molecule embedding follows $\tilde{\mathbf{z}}_{\text{mol}} = f_{\text{mol}}(\hat{\mathbf{z}}_m, \mathbf{e}_m, \boldsymbol{f}_m, \boldsymbol{x}_m) \in \mathbb{R}^{N_m \times d'}$.

Overall, we have a fine-grained complex network f_{ptc} which is a high-order equivariant graph network, and a fine-grained molecule network f_{mol} which is a shallow graph network. We treat molecules and protein-target complexes differently in fine-grained data-specific networks because complexes are always significantly longer than molecules and the high-order long-range interactions need to be captured among them. For a coarse-grained representation $\hat{\mathbf{z}}_m$, if it is originally a protein-target complex, it will be embedded by the complex network f_{ptc}, or if it is originally a molecule, it will be embedded by the molecule network f_{mol}.

3.3 TASK-SPECIFIC PREDICTION MODULE

The task-specific prediction module will distinguish representations $\hat{\mathbf{z}}_{\text{ptc}}$, $\hat{\mathbf{z}}_{\text{mol}}$, and generate the outputs for each task $\hat{\mathbf{y}}_{\text{task}}$. In the multi-task learning setting, each task should have its own specific prediction network f_{task} Collobert & Weston (2008); Liu et al. (2019c); Aribandi et al. (2021) (see the task-specific prediction module in Fig. 1). In practice, our task-specific prediction module consists of 825 output networks corresponding to 825 prediction tasks from the following 4 datasets.

QM9 (12 prediction networks) QM9 is a dataset of molecules consisting of 12 tasks Ramakrishnan et al. (2014). We use the specialized output networks in Thölke & De Fabritiis (2022) for the prediction of molecular dipole moment μ and the prediction of electronic spatial extent $\langle R^2 \rangle$. The gated equivariant blocks Weiler et al. (2018); Schütt et al. (2021) are used for the remaining 10 tasks.

MD17 (14 prediction networks) MD17 is a dataset of molecules consisting of 7 sub-datasets (Aspirin, Ethanol, Malondialdehyde, Naphthalene, Salicylic Acid, Toluene, Uracil) Chmiela et al. (2017). There are 14 tasks in total, where each sub-dataset has 2 prediction tasks for molecular energy E and forces $\vec{F}$. We use the gated equivariant blocks proposed in Weiler et al. (2018); Schütt et al. (2021) to predict E, and $\vec{F}$ are calculated using the negative gradient of E with respect to the atomic coordinates $\vec{F} = -\partial E/\partial \vec{r}$ Thölke & De Fabritiis (2022).

ChEMBL (798 prediction networks) ChEMBL is a protein-target dataset originally proposed in Mendez et al. (2019). Furthermore, 3 sub-datasets ChEMBL10, ChEMBL50, ChEMBL100 are developed by Mayr et al. (2018); Liu et al. (2022) for multi-task learning, and each sub-dataset contains 406, 263, 129 regression tasks, accordingly. We apply a linear function over $\tilde{\mathbf{z}}_{ptc}$ and apply sum pooling to get an output for each regression task.

PDBbind (1 prediction network) PDBbind Wang et al. (2005) is a protein-target dataset consisting of 1 regression task for protein-ligand binding affinity prediction. We apply a linear function over $\tilde{\mathbf{z}}_{ptc}$ and apply sum pooling to predict protein-ligand binding affinity.

The loss $\mathcal{L}_i$ for each task will be calculated based on the outputs $\hat{\boldsymbol{y}}_i$ from each task and the ground truth labels $\boldsymbol{y}_i$, where i is the task number. All $\mathcal{L}_i$ will be weighted and sum up to a multi-dataset multi-task loss $\mathcal{L}_{\text{MDMT}}$ for optimization. One principle for the $\mathcal{L}_{\text{MDMT}}$ design is to treat each task equally important. This principle is naturally held in conventional multi-task learning Mayr et al. (2018). But when it comes to the multi-dataset setting, the data imbalance between different molecular datasets will break this principle. In Sec. 3.4, we will discuss this problem and how to address it by the design of the weighted loss $\mathcal{L}_{\text{MDMT}}$.

3.4 Multi-Dataset Multi-Task Loss

In MDMT-GL, we will face the data imbalance problem. The problem only occurs when we train our model on different datasets simultaneously, *e.g.,* molecules and protein-target complexes, because the number of labeled molecules is always greater than the number of labeled protein-target complexes, and the model will focus more on molecule datasets than protein datasets. This problem is special for multi-dataset setting and does not exist in previous works on multi-task learning with a single molecular type Tan et al. (2021); Lee & Kim (2019); Hu et al. (2021); Liu et al. (2022).

To address this issue, we propose a weighted loss, specific to MDMT-GL, to address the data imbalance problem between different molecular datasets. We are motivated to design the loss so that all tasks are treated equally regardless of the size of labeled training data.

Suppose that we have U tasks and originally $n_1, n_2, \ldots, n_U$ labeled training data for each task, we obtain predictions $\hat{\mathbf{y}}_{i,1}, \hat{\mathbf{y}}_{i,2}, \ldots, \hat{\mathbf{y}}_{i,n_i}$ for task i, and compare them with ground-truth labels $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,n_i}$ for the loss of i-th task $\mathcal{L}_i = \sum_{j=1}^{n_i} l_i(\mathbf{y}_{i,j}, \hat{\mathbf{y}}_{i,j})$. The multi-dataset multi-task loss $\mathcal{L}_{\text{MDMT}} = \sum_{i=1}^{U} c_i \mathcal{L}_i$ is a weighted sum of $\mathcal{L}_i$. To balance the weights of $\mathcal{L}_i$, we want $\sum_{i=1}^{n_1} c_1 = \sum_{i=1}^{n_2} c_2 = \cdots = \sum_{i=1}^{n_U} c_U$, which leads to $c_1 n_1 = c_2 n_2 = \cdots = c_U n_U$. In practice, suppose $n_{\text{min}} = \text{MIN}(n_1, n_2, \ldots, n_U) = n_k$, then we set $c_k = 1$ and for any $i \neq k$, we have $c_i = \frac{n_{\text{min}}}{n_i}$. We will discuss the implementation in Sec. 4.

4 Experiments

In this section, we evaluate the Multi-Dataset Multi-Task Graph Learning framework (MDMT-GL) on real-world molecule and protein-target complex datasets, and show that our proposed learning method can be used to better learn molecule and complex representations. We briefly introduce our datasets in Sec. 3.3. We conduct experiments across 2 molecule datasets and 2 complex datasets, consisting of 825 tasks and 3,139,011 labeled molecular graphs. We divide the experiment section into two subsections, including discussions of molecule datasets in Sec. 4.1 in and discussions on protein datasets in Sec. 4.2. In more detail, we discuss the performance of the model on QM9 in Sec. 4.1.1, on MD17 in Sec. 4.1.1, on ChEMBL in Sec. 4.2.1, and on PDBbind in Sec. 4.2.2.

4.1 MOLECULE DATASETS

In this section, we discuss our model performance on molecule datasets including QM9 Ramakrishnan et al. (2014) and MD17 Chmiela et al. (2017). We compare our MDMT-GL with several classic baselines and the state-of-the-art models in Tab. 1&2. The experimental results show that learning molecule representations can benefit from learning protein representations.

4.1.1 QM9

Data QM9 dataset reports computed geometric, thermodynamic, energetic, and electronic properties for locally optimized geometries. We use the same data split as in Schütt et al. (2018); Klicpera et al. (2020); Thölke & De Fabritiis (2022), where the labeled molecules are divided into 110,000 / 10,000 / 10,831 for training / validation / testing.

Comparison We compare MDMT-GL with several popular baselines and state-of-the-art models, including SchNet Schütt et al. (2018), EGNN Satorras et al. (2021), PhysNet Unke & Meuwly (2019), DimeNet++ Klicpera et al. (2020), Cormorant Anderson et al. (2019), PaiNN Schütt et al. (2021), and Equivariant Transformer (ET) Thölke & De Fabritiis (2022), and report the results in Tab. 1. The results of baselines are obtained from Thölke & De Fabritiis (2022), and the MDMT-GL results are averaged over three runs.

Table 1: Results on all QM9 targets and comparison to previous literature. Scores are reported as mean absolute errors (MAE). Results of MDMT-GL are averaged over three runs.

Target	Unit	SchNet	EGNN	PhysNet	DimeNet++	Cormorant	PaiNN	ET	MDMT-GL
μ	D	0.033	0.029	0.0529	0.041	0.0297	0.012	**0.011**	0.024
α	a_0^3	0.235	0.071	0.0615	**0.0435**	0.085	0.045	0.059	0.061
ϵ_{HOMO}	meV	41	29	32.9	24.6	34	27.6	20.3	**19.2**
ϵ_{LUMO}	meV	34	25	24.7	19.5	38	20.4	17.5	**16.5**
$\Delta\epsilon$	meV	63	48	42.5	32.6	61	45.7	36.1	**31.7**
$\langle R^2 \rangle$	a_0^2	0.073	0.106	0.765	0.331	0.961	0.066	**0.033**	0.047
$ZPVE$	meV	1.7	1.55	1.39	**1.21**	2.027	1.28	1.84	1.35
$\mathbf{U}_0$	meV	14	11	8.15	6.32	22	5.85	6.15	**5.74**
$\mathbf{U}$	meV	19	12	8.34	6.28	21	5.83	6.38	**5.75**
$\mathbf{H}$	meV	14	12	8.42	6.53	21	**5.98**	6.16	6.06
$\mathbf{G}$	meV	14	12	9.4	7.56	20	7.35	7.62	**7.23**
$\mathbf{C}_v$	$\frac{cal}{mol\ K}$	0.033	0.031	0.028	**0.023**	0.026	0.024	0.026	0.026

From Tab. 1, we can observe that MDMT-GL outperforms most popular baselines with significant improvements on 6 out of 12 QM9 targets, including $\epsilon_{\text{HOMO}}, \epsilon_{\text{LUMO}}, \Delta\epsilon, \mathbf{U}_0, \mathbf{U}, \mathbf{G}$. MDMT-GL shows very competitive performance and delivers significant improvements in the challenging molecular chemical property prediction problem via multi-dataset learning.

4.1.2 MD17

Data consists of molecular dynamics trajectories of small organic molecules, including both energies and forces. We use the same data split as in previous works Schütt et al. (2018); Klicpera et al. (2020); Thölke & De Fabritiis (2022). For each sub-dataset, we split the data into a training set with 950 molecules and a validation set with 50 molecules, leaving the remaining molecules for testing.

Comparison We compare MDMT-GL with several popular baselines and state-of-the-art models, including SchNet Schütt et al. (2018), PhysNet Unke & Meuwly (2019), DimeNet Klicpera et al. (2020), PaiNN Schütt et al. (2021), and Equivariant Transformer (ET) Thölke & De Fabritiis (2022), and report the results in Tab. 2. The baseline results are obtained from Thölke & De Fabritiis (2022), and the MDMT-GL results are averaged over three runs.

From Tab. 2, we can observe that MDMT-GL outperforms the most popular baselines with significant improvements on 8 out of 14 MD17 sub-datasets, except energy and forces for naphthalene, forces for salicylic acid, energy and forces for toluene, and forces for uracil. MDMT-GL shows very competitive performance and delivers significant improvements in the challenging molecular dynamics trajectory prediction problem via multi-dataset learning.

Table 2: Results on MD trajectories from the MD17 dataset and comparison to previous literature. The scores are given by the MAE of the energy predictions $(kcal/mol)$ and forces $(kcal/mol/\mathring{A})$. The results of MDMT-GL are averaged over three runs.

Molecule		SchNet	PhysNet	DimeNet	PaiNN	ET	MDMT-GL
Aspirin	*energy*	0.37	0.230	0.204	0.167	0.123	**0.122**
	forces	1.35	0.605	0.499	0.338	0.253	**0.233**
Ethanol	*energy*	0.08	0.059	0.064	0.064	0.052	**0.063**
	forces	0.39	0.160	0.230	0.224	0.109	**0.107**
Malondialdehyde	*energy*	0.13	0.094	0.104	0.091	**0.077**	**0.077**
	forces	0.66	0.319	0.383	0.319	0.169	**0.165**
Naphthalene	*energy*	0.16	0.142	0.122	0.116	**0.085**	0.093
	forces	0.58	0.310	0.215	0.077	**0.061**	0.085
Salicylic Acid	*energy*	0.20	0.126	0.134	0.116	0.093	**0.090**
	forces	0.85	0.337	0.374	0.195	**0.129**	0.137
Toluene	*energy*	0.12	0.100	0.102	0.095	**0.074**	0.079
	forces	0.57	0.191	0.216	0.094	**0.067**	0.071
Uracil	*energy*	0.14	0.108	0.115	0.106	0.095	**0.087**
	forces	0.56	0.218	0.301	0.139	**0.095**	0.101

4.2 PROTEIN-TARGET DATASETS

In this section, we discuss our model performance on protein-target complex datasets including ChEMBL Mendez et al. (2019) and PDBbind Wang et al. (2005). We compare our MDMT-GL with several classic baselines and the state-of-the-art model in Tab. 3&4. The experimental results show that learning protein representations can benefit from learning molecule representations.

4.2.1 CHEMBL

Data The ChEMBL dataset is originally proposed by Mendez et al. (2019) for protein-targeting, but authors in Liu et al. (2022) modify the original dataset and provide three sub-datasets ChEMBL10, ChEMBL50, ChEMBL100 for multi-task learning. In Liu et al. (2022), they claim the tasks numbers are 382/ 152/ 132 (666 tasks in total) for ChEMBL10/ ChEMBL50/ ChEMBL100, but we actually get 406/ 263/ 129 (798 tasks in total) when running their data generation steps. So, we run and test baselines and MDMT-GL on 406/ 263/ 129 tasks, and report the averaged results over three runs. We use the same data split in Liu et al. (2022), splitting the labeled data into the ratio of 80%/ 10%/ 10% for training/ validation/ testing.

Comparison We compare MDMT-GL with several classic multi-task learning baselines and state-of-the-art models, including Multi-Task Learning (MTL) Mayr et al. (2018), Uncertainty Weighing (UW) Kendall et al. (2018), GradNorm Chen et al. (2018), Dynamic Weight Average (DWA) Liu et al. (2019b), Loss-Balanced Task Weighting (LBTW) Liu et al. (2019a), State Graph Neural Network (SGNN) Liu et al. (2022), and Energy-Based State Graph Neural Network (SGNN-EBM) Liu et al. (2022), and report the averaged results in Tab. 3.

Table 3: Results on ChEMBL10, ChEMBL50, ChEMBL100 and comparison to previous literatures. We follow Liu et al. (2022) evaluation metrics on multi-task learning for drug discovery, *i.e.,* the mean of ROC-AUC over all tasks. Results of baselines and MDMT-GL are averaged over three runs.

Method	MTL	UW	GradNorm	DWA	LBTW	SGNN	SGNN-EBM	MGMT-GL
ChEMBL10	0.567	0.552	0.579	0.550	0.583	0.592	0.611	**0.637**
ChEMBL50	0.531	0.549	0.588	0.569	0.571	0.597	0.613	**0.621**
ChEMBL100	0.552	0.571	0.567	0.537	0.568	0.605	0.623	**0.649**

From Tab. 3, we can observe that MDMT-GL outperforms all popular baselines with marginal improvements of AUC-ROC score on ChEMBL10, ChEMBL50, ChEMBL100. By simultaneously learning other molecular datasets and tasks, the MDMT-GL framework can make the best use of the data and leverage the results of predictions for protein-targeting.

4.2.2 PDBBIND

Data PDBbind dataset provides 3D binding structures of protein-ligand complexes with experimentally determined binding affinities. In our experiment, we use the PDBbind2016 dataset, which is the most used PDBbind dataset in previous works Lim et al. (2019); Li et al. (2021). We use the same data split in Li et al. (2021).

Comparison We compare MDMT-GL with several classic baselines and state-of-the-art models, including Spatial Graph Convolution Network (SGCN) Danel et al. (2020), GNN-DTI Lim et al. (2019), DMPNN Yang et al. (2019), Molecule Attention Transformer (MAT) Maziarka et al. (2020), DimeNet Klicpera et al. (2020), CMPNN Song et al. (2020), and Structure-aware Interactive Graph Network (SIGN) Liu et al. (2022). The baseline results are obtained from Li et al. (2021), and MDMT-GL results are averaged over three runs.

Table 4: Results on protein-ligand binding affinity of PDBbind and comparison to previous literature. Scores are reported as root mean square error (RMSE), mean absolute errors (MAE), Pearson's correlation coefficient (R) and standard deviation (SD) in regression to measure the prediction error as in Liu et al. (2022). The results of MDMT-GL are averaged over three runs.

Method	GraphDTA	SGCN	GNN-DTI	DMPNN	MAT	DimeNet	CMPNN	SIGN	MGMT-GL
RMSE ↓	1.562	1.583	1.492	1.493	1.457	1.453	1.408	1.316	**1.172**
MAE ↓	1.191	1.250	1.192	1.188	1.154	1.138	1.117	1.027	**0.923**
SD ↓	1.558	1.582	1.471	1.489	1.445	1.434	1.399	1.312	**1.201**
R ↑	0.697	0.686	0.736	0.729	0.747	0.752	0.765	0.797	**0.866**

From Tab. 4, we can observe that MDMT-GL outperforms all popular baselines with significant improvements in RMSE, MAE, SD, and R scores on the PDBbind dataset. MDMT-GL shows very competitive performance and delivers significant improvements on the challenging protein-binding affinity prediction problem via multi-dataset learning.

Overall, we can see that the Multi-Dataset Multi-Task Graph Learning framework (MDMT-GL) is very competitive in all tasks. We can conclude that MDMT-GL enables the learning of protein representations to benefit the learning of molecule representations, and vice versa. The strong experimental results show that our proposed learning method can utilize the use of labeled training data, and can make the most and best use of them. And this learning framework can mitigate the lack of labeled data in drug discovery.

5 CONCLUSION AND FUTURE WORK

In conclusion, our proposed Multi-Dataset Multi-Task Graph Learning (MDMT-GL) framework is able to address the data insufficiency problem by concurrently training the representations of molecules and protein-target complexes for multiple prediction tasks. The strong experimental results show that there does exist transferable information between molecules and protein-target complexes and it is learnable. We can also say that the learning of protein representations can facilitate the learning of molecule representations, and vice versa. Furthermore, in the future, we could discover some quantum chemical constraints and prior knowledge and add them to the coarse-grained network to capture more informative coarse-grained embeddings.

REFERENCES

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*, 2021.

Andreas Bender and Robert C Glen. Molecular similarity: a key technique in molecular informatics. *Organic & biomolecular chemistry*, 2(22):3204–3218, 2004.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, and Łukasz Maziarka. Spatial graph convolutional networks. In *International Conference on Neural Information Processing*, pp. 668–675. Springer, 2020.

Fan Hu, Jiaxin Jiang, Dongqi Wang, Muchun Zhu, and Peng Yin. Multi-pli: interpretable multi-task deep learning model for unifying protein–ligand interaction datasets. *Journal of cheminformatics*, 13(1):1–14, 2021.

Chenqing Hua, Guillaume Rabusseau, and Jian Tang. High-order pooling for graph neural networks with tensor decomposition. *arXiv preprint arXiv:2205.11691*, 2022.

James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

Ajay N Jain. Morphological similarity: a 3d molecular similarity method correlated with protein-ligand recognition. *Journal of computer-aided molecular design*, 14(2):199–213, 2000.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems*, 34: 28016–28028, 2021.

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

Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

Kyoungyeul Lee and Dongsup Kim. In-silico molecular binding prediction for human drug targets using deep neural multi-task learning. *Genes*, 10(11):906, 2019.

Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 975–985, 2021.

Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

Shengchao Liu, Yingyu Liang, and Anthony Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9977–9978, 2019a.

Shengchao Liu, Meng Qu, Zuobai Zhang, Huiyu Cai, and Jian Tang. Structured multi-task learning for molecular property prediction. In *International Conference on Artificial Intelligence and Statistics*, pp. 8906–8920. PMLR, 2022.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1871–1880, 2019b.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019c.

John Löfblom, Joachim Feldwisch, Vladimir Tolmachev, Jörgen Carlsson, Stefan Ståhl, and Fredrik Y Frejd. Affibody molecules: engineered proteins for therapeutic, diagnostic and biotechnological applications. *FEBS letters*, 584(12):2670–2680, 2010.

Sitao Luan, Mingde Zhao, Xiao-Wen Chang, and Doina Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1906.02174*, 2019.

Sitao Luan, Mingde Zhao, Chenqing Hua, Xiao-Wen Chang, and Doina Precup. Complete the missing half: Augmenting aggregation filtering with diversification for graph convolutional networks. *arXiv preprint arXiv:2008.08844*, 2020.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.

Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzębski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. Chembl: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

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

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

Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe. Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395, 2014.

Ying Song, Shuangjia Zheng, Zhangming Niu, Zhang-Hua Fu, Yutong Lu, and Yuedong Yang. Communicative representation learning on attributed molecular graphs. In *IJCAI*, volume 2020, pp. 2831–2838, 2020.

Zheng Tan, Yan Li, Weimei Shi, and Shiqing Yang. A multitask approach to learn molecular properties. *Journal of Chemical Information and Modeling*, 61(8):3824–3834, 2021.

Philipp Thölke and Gianni De Fabritiis. Torchmd-net: Equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

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

Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370–3388, 2019.

A MODEL ARCHITECTURE

We introduce the full MDMT-GL architecture. Suppose we are given an input molecular data of N_m atoms and E_m edges, its atom numbers $m \in \mathbb{N}^{N_m \times 1}$, atom features $\mathbf{x}_m \in \mathbb{R}^{N_m \times d}$, atom positions $\mathbf{r}_m \in \mathbb{R}^{N_m \times 3}$ in 3D space, edge indices $\mathbf{e}_m \in [0,1]^{N_m \times N_m}$, edge features $\mathbf{f}_m \in \mathbb{R}^{E_m \times f_e}$, where f_n, f_e denote numbers of node features and edge features, respectively.

First, we embed the atom numbers m to an atom-wise coarse-grained representation $\mathbf{z}_m$ by an atom-embedding transformation:

$$\mathbf{z}_m = W_{\text{atom}}\, \boldsymbol{m} \in \mathbb{R}^{N_m \times d}$$

where d is the hidden feature dimension.

Then the coarse-grained representation $\mathbf{z}_m$ will be augmented to an equivariant coarse-grained representation $\hat{\mathbf{z}}_m$ by an augmentation network. There are many equivariant graph neural network options Satorras et al. (2021); Schütt et al. (2021); Thölke & De Fabritiis (2022), and our choice is the equivariant transformer proposed in Thölke & De Fabritiis (2022).

Before the coarse-grained representation gets augmented, there is an exponential normal radial basis function that resembles a continuous filter convolution to filter the neighborhood of an atom Schütt et al. (2018). The distance d_{ij} between atoms i, j is defined as:

$$e_k^{\text{RBF}} = \phi(d_{ij})\text{exp}(-\beta_k(\text{exp}(-d_{ij}) - \mu_k)^2, \quad \phi(d_{ij}) = \begin{cases} \frac{1}{2}(\cos(\frac{\pi d_{ij}}{d_{\text{cut}}}) + 1), & \text{if } d_{ij} \leq d_{\text{cut}} \\ 0, & \text{if } d_{ij} > d_{\text{cut}} \end{cases}$$

where β_k, μ_k are fixed parameters specifying the center and width of the radial basis function k. β is initialized as $(2K^{-1}(1 - \text{exp}(d_{\text{cut}})))^{-2}$, μ is initialized with values equally spaced between $\text{exp}(-d_{\text{cut}})$ and 1 for all k proposed by Unke & Meuwly (2019). And the cosine cutoff $\phi(d_{\text{cut}})$ is used to ensure a smooth transition to 0 as d_{ij} approaches to d_{cut}.

The neighborhood embedding $\mathbf{n}_m$ for m is then defined as:

$$\mathbf{n}_m \in \mathbb{R}^{N_m \times d},\ n_{m,i} = \sum_{j=1}^{N} \mathbf{z}_{m,j} \odot W_{\text{Filter}}\, e^{\text{RBF}}(d_{ij}) \in \mathbb{R}^d,$$

where each row i corresponds to the neighbor embedding of atom i of m. We update the coarse-grained representations $\mathbf{z}_m$ with the neighbor embedding $\mathbf{n}_m$:

$$\mathbf{z}_m = \text{LayerNorm}(W_{\text{Transform}}[\mathbf{z}_m, \mathbf{n}_m] + b_{\text{Transform}}).$$

Then the coarse-grained representation $\mathbf{z}_m$ is augmented by an equivariant transformer layer proposed in Thölke & De Fabritiis (2022). The interatomic distances are projected into two multidimensional filters D^K, D^V:

$$D^K = \sigma(W_{D_K}\, e^{\text{RBF}}(\mathbf{r}_{m,ij}) + b_{D_K}), \quad D^V = \sigma(W_{D_V}\, e^{\text{RBF}}(\mathbf{r}_{m,ij}) + b_{D_V}).$$

And attention is weighted by a cosine cutoff to ensure that atoms with a distance greater than d_{cut} do not interact:

$$A = \text{Activation}(\sum_{k}^{F} Q_k \odot K_k \odot D_k^K) \cdot \phi(d_{ij}), \quad Q = W^{Q_1}\boldsymbol{z}_m \text{ and } K = W^{K_1}\boldsymbol{z}_m.$$

The attention mechanism's value is also split into three vectors of equal dimension:

$$\mathbf{s}_{m,ij}^1, \mathbf{s}_{m,ij}^2, \mathbf{s}_{m,ij}^3 = \text{split}(V_j \odot D_{ij}^V) \in \mathbb{R}^d, \quad V = W^{V_1}\mathbf{z}_m,$$

and

$$\mathbf{y}_m \in \mathbb{R}^{N_m \times 3d}, \quad \mathbf{y}_{m,i} = W_{O_1}(\sum_{j}^{N} A_{ij} \cdot \mathbf{s}_{ij}^3),$$

where $\mathbf{y}_{m,i}, \mathbf{s}_{m,ij}^1, \mathbf{s}_{m,ij}^2$ correspond to features, and two filters. Then the features $\mathbf{y}_m$ are split into three features of equal size:

$$\mathbf{q}_m^1, \mathbf{q}_m^2, \mathbf{q}_m^3 \in \mathbb{R}^{N_m \times d'}$$

$$\Delta \mathbf{z}_m = \mathbf{q}_m^1 + \mathbf{q}_m^2 \odot <W_{\text{Linear1}}\mathbf{v}, W_{\text{Linear2}}\mathbf{v}> \in \mathbb{R}^{N_m \times d},$$

notice that $\mathbf{v}_m \in \mathbb{R}^{N_m \times 3}$ is set to 0 in the beginning, *i.e.,* initially $\mathbf{v}_m = 0^{N_m \times 3}$. And for $\mathbf{v}$,

$$\Delta \mathbf{v}_m = \mathbf{w}_m + \mathbf{q}_m^3 \odot W_{\text{Linear3}}\mathbf{v}_m, \quad \mathbf{w}_{m,i} = \sum_j^N \mathbf{s}_{m,ij}^1 \odot \mathbf{v}_{m,j} + \mathbf{s}_{m,ij}^2 \odot \frac{\mathbf{r}_{m,i} - \mathbf{r}_{m,j}}{||\mathbf{r}_{m,i} - \mathbf{r}_{m,j}||},$$

and

$$\mathbf{z}_m = \mathbf{z}_m + \Delta \mathbf{z}_m, \quad \mathbf{v}_m = \mathbf{v}_m + \Delta \mathbf{v}_m.$$

More details on the transformer can be found in Thölke & De Fabritiis (2022). After iterative updates, we will receive our equivariant coarse-grained representation $\hat{\mathbf{z}}_m$,

$$\hat{\mathbf{z}}_m = \text{LayerNorm}(\mathbf{z}_m + \sum_l \Delta \mathbf{z}_m) \in \mathbb{R}^{N \times d}.$$

Then the equivariant coarse-grained representation is cooperated with node and edge features

$$\hat{\boldsymbol{z}}_m = \text{LayerNorm}(W_C[\hat{\boldsymbol{z}}_m, \boldsymbol{x}_m, W_E \boldsymbol{f}_m]) \in \mathbb{R}^{N \times d}$$

If $\hat{\boldsymbol{z}}_m$ is originally a protein-target complex then will be encoded by an equivariant high-order graph neural network Satorras et al. (2021); Schütt et al. (2021); Thölke & De Fabritiis (2022). And our choice is to develop Kim et al. (2021) to an equivariant graph transformer for complex network, it follows

$$\hat{\boldsymbol{z}}_m = \text{Enc}_{k\to l}(\hat{\boldsymbol{z}}_m) = \text{Attn}_{k\to l}(\hat{\boldsymbol{z}}_m) + L_{l\to l}^2(\text{Activation}(L_{l\to l}^1(\text{Attn}_{k\to l}(\hat{\boldsymbol{z}}_m)))) \in \mathbb{R}^{N_m^l \times d'},$$

$$\text{Attn}_{k\to l}(\hat{\boldsymbol{z}}_m)_j = \sum_{h=1}^H \sum_\mu \sum_i \alpha_{i,j}^{h,\mu} \hat{\boldsymbol{z}}_{m,i} W_{h,\mu}^{V_2} W_{h,\mu}^O,$$

where in the first layer $k = 1$, H is the number of heads, $L_{l\to l}^1 : \mathbb{R}^{N_m^l \times d} \to \mathbb{R}^{N_m^l \times d'}, L_{l\to l}^2 : \mathbb{R}^{N_m^l \times d'} \to \mathbb{R}^{N_m^l \times d}$. And to compute each attention $\alpha^{h,\mu} \in \mathbb{R}^{n^{k+l}}$ from $\hat{\boldsymbol{z}}_m \in \mathbb{R}^{n^k \times d}$,

$$a_{i,j}^{h,\mu} = \begin{cases} \dfrac{\sigma(Q_j^\mu, K_i^\mu)}{\sum_{i|(i,j)\in\mu} \sigma(Q_j^\mu, K_i^\mu)}, & (i,j) \in \mu \\ 0, & \text{otherwise} \end{cases}, \quad Q^\mu = L_{k\to l}^\mu(\hat{\boldsymbol{z}}_m) \text{ and } K^\mu = L_{k\to k}^\mu(\hat{\boldsymbol{z}}_m).$$

More details can be found in Kim et al. (2021), we augment $\hat{\boldsymbol{z}}_m \in \mathbb{R}^{N_m^l \times d'}$ to an equivariant form by

$$\mathbf{s}_{m,ij}^1, \mathbf{s}_{m,ij}^2, \mathbf{s}_{m,ij}^3 = \text{split}(V_j) \in \mathbb{R}^{l \times d'}, \quad V = W^{V_2}\hat{\mathbf{z}}_m,$$

and

$$\mathbf{y}_m \in \mathbb{R}^{N_m^l \times 3d'}, \quad \mathbf{y}_{m,i} = W_{O_2}(\sum_j^N a_{i,j} \cdot \mathbf{s}_{ij}^3).$$

Then the features $\mathbf{y}_m$ are split into three features of equal size:

$$\mathbf{q}_m^1, \mathbf{q}_m^2, \mathbf{q}_m^3 \in \mathbb{R}^{N_m^l \times d'}$$

$$\Delta \hat{\boldsymbol{z}}_m = \mathbf{q}_m^1 + \mathbf{q}_m^2 \odot <W_{\text{Linear1'}}\mathbf{v}, W_{\text{Linear2'}}\mathbf{v}> \in \mathbb{R}^{N_m^l \times d'},$$

notice that $\mathbf{v}_m \in \mathbb{R}^{N_m^l \times 3}$ is set to 0 in the beginning, *i.e.,* initially $\mathbf{v}_m = 0^{N_m^l \times 3}$. And for $\mathbf{v}_m$,

$$\Delta \mathbf{v}_m = \mathbf{w}_m + \mathbf{q}_m^3 \odot W_{\text{Linear3'}}\mathbf{v}_m, \quad \mathbf{w}_{m,i} = \sum_j^N \mathbf{s}_{m,ij}^1 \odot \mathbf{v}_{m,j} + \mathbf{s}_{m,ij}^2 \odot \frac{\mathbf{r}_{m,i} - \mathbf{r}_{m,j}}{||\mathbf{r}_{m,i} - \mathbf{r}_{m,j}||},$$

and

$$\tilde{\mathbf{z}}_{\text{mol}} = \mathbf{z}_m + \Delta \mathbf{z}_m \in \mathbb{R}^{N_m^l \times d'}, \quad \mathbf{v}_m = \mathbf{v}_m + \Delta \mathbf{v}_m \in \mathbb{R}^{N_m^l \times 3}.$$

In the last layer, we set $l = 1$ and receive receive our equivariant fine-grained complex representation $\hat{\mathbf{z}}_m$,

$$\tilde{\mathbf{z}}_{\text{ptc}} = \text{LayerNorm}(\tilde{\mathbf{z}}_{\text{ptc}}) \in \mathbb{R}^{N_m \times d'}.$$

We have the fine-gained representations for protein-target complex.

Or if $\hat{\boldsymbol{z}}_m$ is originally a molecule then will be encoded by a shallow graph neural network Kipf & Welling (2016); Luan et al. (2020); Hua et al. (2022). And our choice is the simplest graph convolution network Kipf & Welling (2016),

$$\tilde{\mathbf{z}}_{\text{mol}} = \text{LayerNorm}(\text{Activation}(\boldsymbol{e}_m \boldsymbol{z}_m W_m)) \in \mathbb{R}^{N_m \times d'}.$$

Then it will be fed into downstream task-specific module.