A MOLECULAR HYPER-MESSAGE PASSING NETWORK WITH FUNCTIONAL GROUP INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We proposed the molecular hyper-message passing network (`MolHMPN`[1]) that predicts the molecular properties of a molecule with prior knowledge-guided subgraph. Modeling higher-order connectivities in molecules is necessary as changes in both the pair-wise and higher-order interactions among atoms results in the change of molecular properties. Many approaches have attempted to model the higher-order connectivities. However, those methods relied heavily on data-driven approaches, and it is difficult to determine if the utilized subgraphs contain any properties of interest or have any significance on the molecular properties. Hence, we propose `MolHMPN` to utilize the functional group prior knowledge, which has been defined by chemists, to model the pair-wise and higher-order connectivities among atoms in a molecule. Molecules can contain many types of functional groups, which affect the properties the molecules. For example, the toxicity of a molecule is associated with toxicophores, such as nitroaromatic groups and thiourea. `MolHMPN` uses functional groups to construct hypergraphs, modifies the hypergraph using domain knowledge-guided learning scheme, and embeds the graph and hypergraph inputs using a hypergraph message passing (`HyperMP`) layer. Our model provides a way to utilize prior knowledge in chemistry for molecular properties prediction tasks, and balances between the usage of prior knowledge and data-driven learning adaptively. We show that our model is able to outperform the other baseline methods for most of the dataset, and show that using domain knowledge-guided data-learning is effective.

1 INTRODUCTION

Toxicological screening is vital for the development of new drugs, the evaluation of the therapeutic potential of existing molecules, and the assessment of pharmacological activity and toxicity potential of new molecules on human. Traditionally, toxicity studies of molecules relied on animal testing, which can provide inadequate bases for predicting clinical outcomes on humans (Akhtar, 2015). It has also been estimated that it takes more than eight years to test and study a new drug before its approval to the general public, which includes early laboratory and animal testing (Food & Administration, 2015). Machine learning (ML) methods have therefore been utilized widely to assess the effects that chemicals have on humans and the evironments as it is able to utilize large types and sizes of data while reducing the time and cost it takes for drugs approval, and avoiding costly late-stage failures.

In chemistry, molecules are constructed from a carbon skeleton, onto which functional groups are attached to. The carbon skeleton a chain of carbon atoms and is relatively unreactive. On the other hand, functional group is a group of atoms that are bonded together in a particular fashion, and determines the reactivities and chemical properties of the molecules (Blackman, 2019). Functional groups can therefore be seen as the higher-order interactions between groups of atoms in a molecule. Molecules with the same functional groups often exhibit similar properties while molecules with different functional groups exhibit different properties. Figure 1 shows examples of molecules that have similar structures but with different properties. From figure 1, it can be seen that changes in the pair-wise and higher-order interactions among the atoms can change the properties of the molecules. (Kotera et al., 2008). Hence, accounting for both pair-wise and higher-order interactions

[1]The code is available at will-be-available-after-the-decision.

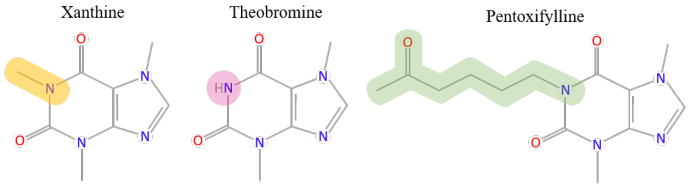


Figure 1: Molecules of similar structures but different properties. xanthine is found in caffeine and temporarily prevents or reduces drowsiness, theobromine is found in cacao and has mood improving effect, and pentoxifylline is a drug used to treat muscle pain in people with peripheral artery disease. The colored parts shows their difference. The yellow and pink parts show that the pair-wise interactions between two atoms can change the properties of the molecules, and the yellow/pink and green parts show that the higher-order interactions between atoms can change molecular properties.

among atoms is important for molecular properties prediction. The current study aims to learn to identify and utilize these higher-order interactions to predict the properties of a target molecule.

In ML, graph-based methods have been used actively for molecule-related tasks for their ability to represent molecules as graphs. Modeling higher-order connectivities is necessary in various graph-related tasks (Jumper et al., 2021; Jin et al., 2018; 2020). Although this can be done by stacking multiple graph convolution layers, it can cause the model to suffer from the oversmoothing problem (Rong et al., 2020). Instead of performing multiple rounds of convolutions, graph pooling methods learns to coarsen some parts of the graph into a single node (Ying et al., 2018; Noutahi et al., 2020). Alternatively, this can also be done by augmenting substructures, such as introducing virtual nodes (Li et al., 2018) or combining multiple nodes (Sun et al., 2019; Jin et al., 2020; Huang & Zitnik, 2020) as the subgraph. Similarly, hypergraphs contains hyperedges that are made up of nodes from a subgraph (Feng et al., 2018; Bai et al., 2021). However, these methods have focused exclusively on data-driven approaches and it is hard to determine if those subgraphs contain any properties of interest or have any significance on the molecular properties like the functional groups. Hence, inspired by the significance of functional groups on the properties of the molecules, the current study aims to utilize the prior knowledge of functional groups to model the higher-order connectivities in a molecule.

In this paper, we propose a molecular hyper-message passing network (`MolHMPN`) that is able to predict the molecular properties of a molecule with prior knowledge-guided subgraphs. Our model (`MolHMPN`) predicts the molecular properties by conducting the following sequential operations:

- **Constructing hypergraphs using functional groups.** Given a graph representation of a molecule that is constructed from its simplified molecular-input line-entry system (SMILES) string, `MolHMPN` constructs the hyperedges according to the chemically-valid functional groups that have been identified by chemists to represent the higher-order connectivities among the atoms. Each hyperedge represents a functional group that is present in a molecule (a molecule can have many functional groups).
- **Embedding the graph and hypergraph using hypergraph message passing layer (`HyperMP`).** The `HyperMP` consists of an atom graph convolution (AtomGC) and a functional group graph convolution (FuncGC) for the graphs and hypergraphs respectively. It performs message passing on the graphs and hypergraphs sequentially.
- **Modifying the hypergraph using the computed embeddings.** `MolHMPN` adjusts the input hypergraph by considering the original graph and hypergraph, and their respective embedded representations. This process updates the prior knowledge (i.e., input hypergraphs) with observations (i.e., embeddings) similar to that of the the Bayesian approaches.
- **Predicting the molecular properties from the modified hypergraph.** `MolHMPN` applies HyperMP again to compute the embedding with the original graph and modified hypergraph, and predict the target label with the updated embeddings.

The key contribution of the current study is on the adaptation of functional groups using prior knowledge and the utilization of the prior knowledge selectively when conducting the molecular prediction tasks. Our novelties are summarized as follows:

- **Providing a way to utilize the prior knowledge.** `MolHMPN` translates functional groups, which are based upon prior knowledge in chemistry, into hyperedges to process higher-order connectivities in molecules effectively.

- **Balancing between prior knowledge and data-driven learning.** Without heavily relying on the functional group prior knowledge, `MolHMPN` learns to use such information adaptively depending on the target input. This can alleviate risk of using faulty information or representations of the target molecule.

We evaluate the effectiveness of `MolHMPN` on several datasets that are used for molecular properties classification and regression tasks, and show that `MolHMPN` is able to outperform the other baseline methods for most of datasets. We also analyze the usage of different types of substructures and the effectiveness of the prior knowledge-guided data-driven learning for the prediction tasks.

2 Related works

In this section, we provide an overview of the applications of graph neural networks (GNNs) in chemistry-related tasks, and methods that utilizes higher-order connectivities and domain knowledge in deep learning.

Applications of GNNs in chemistry. Graph representation of molecules is natural and preferred as the molecular structure is inextricably linked to the molecular properties of the molecules. The atoms and bonds of the molecules are represented by the nodes and edges of the graphs. These methods take the graph as inputs and consider the pair-wise or higher-order connectivities among the graph entities to predict the molecular properties. Message passing neural network (MPNN), a representative GNN architecture, has been devised as a fast simulation method to replace computationally expensive quantum mechanical simulations (Gilmer et al., 2017). Directed MPNN (DMPNN), a variant of MPNN, uses directional message passing based on directions of the edges (Yang et al., 2019). Communicative MPNN (CMPNN) further improves DMPNN by devising a sophisticated updating procedure for the nodes and edges, and strengthens the messages between the nodes and edges using a message booster (Song et al., 2020). Subgraph neural network learns a disentangled subgraph representation and propagates the messages at the subgraph level Alsentzer et al. (2020). It has been shown experimentally that subgraphs contribute significantly to the prediction results (Ying et al., 2019; Pope et al., 2019).

Higher-order connectivities in GNNs. Higher-order connectivities have been utilized in various graph-related tasks. Many approaches attempted to extract meaningful subgraphs for their respective tasks. For instance, frequently-occurring substructures have been utilized for polymer generation and molecule property optimization (Jin et al., 2020), subgraphs that are constructed from their K-hop neighbors for graph meta-learning tasks (Huang & Zitnik, 2020), and residual substructures that are unspecified by the graph adjacency matrix has been utilized for molecular properties prediction tasks (Li et al., 2018). Graph pooling methods has also been used to learn the hierarchical representations of graphs (Ying et al., 2018; Noutahi et al., 2020).

Domain knowledge incorporation to Neural Networks (NNs). Incorporating domain knowledge to ML models often enhances the performance while decreasing the number of training samples that are required to attain a certain performance. For example, when the whole dynamics of the target task is known, the entire (or partial) ML model can be trained to match the dynamics (Raissi et al., 2019; Park & Park, 2019; Long et al., 2018; Yang et al., 2021). However, utilizing the entire domain knowledge in a closed form may not be possible in practice, and may be unfavorable to the models depending on the selection of prior knowledge. In this regard, prior knowledge can be leveraged *partially* to regularize the models via augmented loss functions. It has also be shown that models that leverage the prior knowledge partially are able to outperform their pure data-driven counterparts (Erichson et al., 2019; Seo et al., 2019; Yin et al., 2020). `MolHMPN` is also a method that uses prior knowledge to complement the data-driven (learning) scheme. However, unlike the approaches above that constraint or penalize the models to conform to the prior knowledge, we utilize the prior knowledge to guide the model and also allow the model to overcome it if needed.

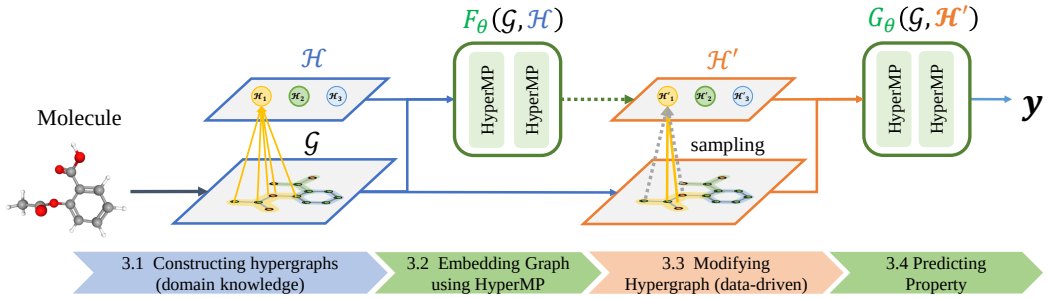


Figure 2: **Overall architecture of `MolHMPN`**

3 METHODOLOGY

This section highlights the methodology of the proposed `MolHMPN`. In `MolHMPN`, the hypergraphs are first constructed using the prior knowledge of functional groups. The graph and constructed hypergraphs are then embedded using the `HyperMP` layer(s) so as to modify the membership of the hyperedges using the computed embeddings. The graph and modified hypergraphs are then embedded again using the `HyperMP` layer(s) to predict the target label with the updated embedding. Figure 2 shows the overall architecture of `MolHMPN`.

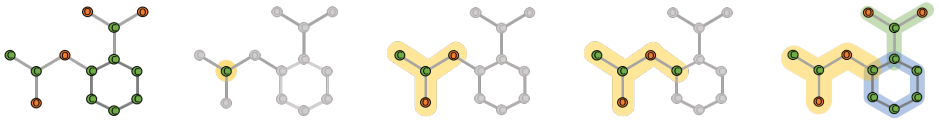
3.1 HYPERGRAPH CONSTRUCTION

Inspired by the significance of functional groups on the molecular properties as discussed in section 1, we utilize the knowledge of functional groups that are defined by chemists to let the model identify the similarities and differences of the molecules more easily. We represent the molecules as conventional pair-wise graphs and hypergraphs. The conventional pair-wise graphs are defined as $\mathcal{G} = \{\mathbb{V}, \mathbb{E}\}$, where $\mathbb{V}$ is a set of nodes (atoms) $v_i \in \mathbb{V}$, and $\mathbb{E}$ is a set of edges (bonds) $e_{ij} \in \mathbb{E}$ if a bond between v_i and v_j exists. The features of v_i and e_{ij} are defined as x_i and x_{ij} respectively. The hypergraph is defined as $\mathcal{H} = \{\mathcal{H}_k | k = 1, ..., n_K\}$, where $\mathcal{H}_k$ is k^{th} hyperedge that has a set of nodes as its members. The features of $\mathcal{H}_k$ are defined as z_k.

When constructing $\mathcal{H}$, we consider atoms in cyclic and acyclic (open-chain) groups separately. The minimal collection of cycles in the molecules are extracted as $\mathcal{H}_k$. For the acyclic groups, the vicinity of the functional group is considered when extracting the hyperedge representation, which is defined as the central atom and the atoms that are attached to it (Kotera et al., 2008). The main atoms that are used are carbon (C), nitrogen (N), oxygen (O), phosphorus (P) and sulfur (S), and the main bond types that are used are the single ($-$), double ($=$) and triple bonds ($\equiv$). The extraction process of the acyclic groups can be described as follows:

1. Find a central atom (e.g., C, N, O, P or S) from $\mathcal{G}$ and set it as v_c.
2. Find the 1-hop neighborhood set $\mathbb{F}_1(v_c)$ of v_c, which is given as $\mathbb{F}_1(v_c) = \{v_j \in \mathcal{N}(v_c) \,|\, t(v_j) \in \mathbb{A}_t, t(e_{ij}) \in \mathbb{B}_t\}$, where $\mathcal{N}(v_c)$ is the neighborhood of v_c, $t(\cdot)$ denotes the types of atom/bond, and $\mathbb{A}_t, \mathbb{B}_t$ are the sets of target atom and bond respectively that are based accordingly to the target functional group.
3. Find the 2-hop neighborhood set $\mathbb{F}_2(v_c)$ of v_c, which is given as $\mathbb{F}_2(v_c) = \{v_k \in \bigcup_{v_j \in \mathcal{N}(v_i)} \mathcal{N}(v_j) \,|\, t(v_j) \neq \text{C} \vee t(e_{ij}) \neq -\}$.
4. The extracted hyperedge is hence $\mathcal{H}_k = \{v_c\} \cup \mathbb{F}_1(v_c) \cup \mathbb{F}_2(v_c)$.

Different combinations of the central atoms, and $\mathbb{A}_t$, $\mathbb{B}_t$ are used to match each functional group. Here, the prior knowledge of functional groups is applied in $\mathbb{A}_t$ and $\mathbb{B}_t$. The remaining atoms that do not belong to any of the specified functional groups are put into the same hyperedge if they are connected by an edge. Figure 3 shows an example of the hyperedge construction for the carboxyl group in aspirin. The list of functional groups used in this paper are given in Appendix A.1. $\mathcal{G}$ and $\mathcal{H}$ will then be fed into the `HyperMP` layer to compute the embedding that are needed to adjust the members of $\mathcal{H}_k$.



(a) $\mathcal{G}$ (b) Set v_c (c) Find $\mathbb{F}_1(v_c)$ (d) Find $\mathbb{F}_2(v_c)$ (e) Extracted $\mathcal{H}$

Figure 3: **Hypergraph construction for aspirin.** a) $\mathcal{G}$ of aspirin. b) Set carbon as v_c. c) To find $\mathbb{F}_1(v_c)$, set v_c- O, $v_c =$ O and v_c- C, where $\{\text{O}, \text{C} \in \mathbb{A}_t\}$ and $\{-, = \in \mathbb{B}_t\}$. d) To find $\mathbb{F}_2(v_c)$, find v_j that is not C and e_{ij} that is not a single bond. e) All the extracted $\mathcal{H}_k$ of $\mathcal{G}$.

3.2 Graph and hypergrph embedding with Hypergraph message passing

Modeling both the pair-wise (atom/bond) and higher-order (functional group) connectivities is crucial for conducting the molecule property predictions. Hence, we introduce the hypergraph message passing `HyperMP` layer to integrate the information from both the atoms and functional groups. `HyperMP` updates the input graphs via two steps: atom graph convolution (AtomGC) and functional group graph convolution (FuncGC). The general equation of the `HyperMP` can be defined as:

$$\mathcal{G}', \mathcal{H}' = \text{HyperMP}(\mathcal{G}, \mathcal{H}) \tag{1}$$

where $\mathcal{G}'$ and $\mathcal{H}'$ are the updated graph and hypergraph respectively.

AtomGC. AtomGC is designed to model the pair-wise connectivities between atoms that are bonded together. It involves updating the edge features using the features of the edges and nodes that it connects, and updating the node features using the updated edge features. The edge update step is given as:

$$x'_{ij} = f_{\text{bond}}(x_i, x_j, x_{ij}) \tag{2}$$

where $f_{\text{bond}}(\cdot)$ is the edge multi-layer perceptron (MLP). It is noteworthy that, for the target tasks, the edge information is essential as the chemical bonds contains crucial information about the molecular properties. In the node update step, the x'_{ij} is aggregated to produce the x'_i as follows:

$$\alpha_{ij} = f_{\text{attn}}(x_i, x_j, x_{ij}) \tag{3}$$

$$x'_i = f_{\text{atom}}\Big(x_i, \sum_{j=\mathcal{N}(i)} \alpha_{ij} x'_{ij}\Big) \tag{4}$$

where α_{ij} is the attention coefficient of e_{ij}, $f_{\text{attn}}(\cdot)$ is the attention multi-layer perceptron (MLP) whose output activation is the sigmoid activation function, and $f_{\text{atom}}(\cdot)$ is the node MLP and $\mathcal{N}(i)$ is the neighborhood set of v_i. Here, unlike many attention modules that normalizes the attention scores so that the summantion of the scores becomes 1.0, we normalize each attention score to be between 0.0 and 1.0. We empirically confirmed that this selection results in better prediction performance than the conventional attention scheme.

FuncGC. FuncGC is designed to model the higher-order connectivities that are defined by the chemically-valid functional groups. Although the same functional groups can be present in many molecules, the effects that they have on the molecular properties may differ depending on their neighboring functional groups (or atoms). To account for such differences, we utilize the updated node feature that contains local information from the molecular graphs when generating the localized functional group features. We start the FuncGC by updating z_k using x'_i as follows:

$$\tilde{z}_k = g_{\text{atom}\to\text{fg}}\Big(z_k, \sum_{i \in \mathcal{H}_k} x'_i\Big) \tag{5}$$

where $\tilde{z}_k$ is the localized feature that receives localized information from AtomGC, and $g_{\text{atom}\to\text{fg}}(\cdot)$ is the localizing MLP. Unlike $\mathcal{G}$, $\mathcal{H}$ has no naturally defined edges as the functional groups are concepts rather than physically exist. Hence, we learn the edges among the hyperedges as follows:

$$z'_{km} = g_{\text{edge}}(\tilde{z}_k, \tilde{z}_m) \tag{6}$$

where z'_{km} is the learnt edge feature between $\mathcal{H}_k$ and $\mathcal{H}_m$, and g_{edge} is the edge MLP. z'_{km} thus captures the interaction between $\mathcal{H}_k$ and $\mathcal{H}_m$. Lastly, we perform the hyperedge update with z'_{km} as follows:

$$\beta_{km} = g_{\text{attn}}(\tilde{z}_k, \tilde{z}_m) \tag{7}$$

$$z'_k = g_{\text{fg}}\Big(z_k, \sum_{m \in \mathcal{H}} \beta_{km} z'_{km}\Big) \tag{8}$$

where β_{km} is the attention coefficient between the k^{th} and the m^{th} hyperedge, $g_{\text{attn}}(\cdot)$ is the attention MLP whose output activation is sigmoid activation function as in AtomGC, and $g_{\text{fg}}(\cdot)$ is the hyperedge update function. The `HyperMP` layer is then used to modify the membership of the hyperedges and predict the molecular properties of the molecules using their respective computed embeddings.

Note that we did not design a path that propagate z'_k back to the members (atoms) of $\mathcal{H}_k$. This design works similar to the uninterrupted gradient path of LSTM (Hochreiter & Schmidhuber, 1997) or the latent arrays of Perciever models (Jaegle et al., 2021). We also experimentally confirmed that this design shows better prediction results.

3.3 Learning the prior-guided subgraph structures

$\mathcal{H}$ is constructed using the functional groups of the molecules. However, molecular properties from the understanding of functional groups may be more straightforward for chemists, but may not be so for GNNs as we have discussed in section 1. Hence, we allow models to adjust the members of $\mathcal{H}_k$, which is built upon the prior knowledge of functional groups, while predicting the molecular property. The general equation of the membership adjustment function $F_\theta(\mathcal{G}, \mathcal{H})$ can be defined as:

$$F_\theta(\mathcal{G}, \mathcal{H}) = \tilde{\mathcal{H}} \tag{9}$$

where $\tilde{\mathcal{H}}$ is the membership-adjusted hypergraph. It first uses the membership encoder $f_\theta(\cdot)$ to produce the membership-encoded features as follows:

$$\{\hat{x}_i\}, \{\hat{z}_k\} = f_\theta(\mathcal{G}, \mathcal{H}) \tag{10}$$

where $\hat{x}_i$ and $\hat{z}_k$ are the membership-encoded node and hyperedge features respectively, and $f_\theta(\cdot)$ is a stack of the `HyperMP` layer(s). As the memberships can be interpreted as a virtual "edge" between an atom v_i and its functional group $\mathcal{H}_k$, we employ a graph structure learning method to adjust the membership. In the adjustment procedure, we consider the random discrete methods (i.e., the adjusted memberships are binary) which share a common philosophy with the Bayesian approaches. The membership adjustment procedure then starts by using the membership-encoded features to produce $\tilde{\mathcal{H}}$ as follows:

$$m_{ik} = g_\theta(\hat{x}_i, \hat{z}_k) \qquad \forall v_i \in \mathcal{H}_k \tag{11}$$

$$\tilde{m}_{ik} = \text{sigmoid}\big(\big(\log\big(\frac{m_{ik}}{1 - m_{ik}}\big) + \epsilon_0 - \epsilon_1\big)/s\big) \qquad \forall v_i \in \mathcal{H}_k \tag{12}$$

where m_{ik} is the bernoulli parameter for v_i to become a member of $\mathcal{H}_k$, $\tilde{m}_{ik}$ is the sampled membership, $g_\theta(\cdot)$ is the MLP whose output activation is the sigmoid function, ϵ_0 and ϵ_1 are the samples of Gumbel(0,1), and $s > 0$ is the temperature parameter. This procedure reparameterize the Bernoulli distribution via Gumbel reparameterization such that the (sampled) binary $\tilde{m}_{ik}$ are differentiable (Jang et al., 2016). By annealing $s \to 0$, we can recover $\tilde{m}_{ik} \sim \text{Ber}(m_{ik})$. We define the k^{th} adjusted hyperedge $\tilde{\mathcal{H}}_k = \{v_i \in \mathcal{H}_k \,|\, \tilde{m}_{ik} = 1\}$. $\tilde{\mathcal{H}}$ will then be used to produce the final predictions.

A similar approach is investigated in the context of pair-wise graph structure learning (Shang et al., 2021), where they assume that the edges of a complete graph is subjective to edge learning. On the other hand, we utilize this idea only to the members of hyperedges so as to provide a balance between the usage of prior knowledge and the data-driven scheme.

Extending hyperedges As we allow the model to adjust the given hyperedges, it naturally provokes us to use extended hyperedges as it may provide more efficient representations for molecular properties predictions. In that regard, we extend the hyperedges as their K-local subgraph as follows:

$$\mathcal{H}_k = \bigcup_{v_i \in \mathcal{H}_k} \mathcal{N}_K(v_i) \tag{13}$$

Table 1: **Benchmark results.** Comparing between different methods for molecular properties prediction. All results are taken from the original papers except CMPNN. Results in bold are the best-performing results for their respective datasets. (↑ means that higher result is better and ↓ means that lower result is better.)

	Metric	AUROC					RMSE		
	Dataset	Tox21 (↑)	ClinTox (↑)	SIDER (↑)	BBBP (↑)	BACE (↑)	ESOL (↓)	FreeSolv (↓)	Lipophilicity (↓)
PAIR	● MPNN (atom only)	0.845	0.896	0.644	0.908	0.864	0.719	1.243	0.625
	⋆ MPNN	0.844	0.881	0.641	0.910	0.850	0.702	1.242	0.645
	× DMPNN	0.845	0.894	0.646	0.913	0.878	0.665	1.167	0.596
	▲ CMPNN	**0.854**	0.908	0.656	**0.958**	0.887	0.567	0.901	0.582
SUB	● AGCN	0.802	0.868	0.592	–	–	0.306	1.33	0.736
	⋆ GAAN	0.839	0.888	0.658	–	–	**0.294**	1.057	0.605
	× ML-MPNN	0.852	0.892	**0.689**	–	–	0.571	1.052	0.560
	●MolHMPN	0.837	**0.924**	0.620	0.928	**0.894**	0.392	**0.815**	**0.511**

where $\mathcal{N}_K(v_i)$ is the K-hop neighborhood set of v_i. This extension allows the membership adjustment to consider a much higher-order interactions while restricting the scope of the edge (or membership) learning to the extented $\mathcal{H}_k$ so that the learned memberships are guided by the chemically-valid prior knowledge.

3.4 Molecular properties prediction

From the aforementioned methods, $\mathcal{H}$ is first constructed using the prior knowledge of the functional groups for a given $\mathcal{G}$. Then the memberships of $\mathcal{H}$ are adjusted using $F_\theta(\mathcal{G}, \mathcal{H})$ to produce $\tilde{\mathcal{H}}$. Hence, in the final step of `MolHMPN`, we predict the target label y of a given molecule by updating $\mathcal{G}$ and $\tilde{\mathcal{H}}$ using the `HyperMP` layer as follows:

$$y = G_\theta(\mathcal{G}, \tilde{\mathcal{H}}) \tag{14}$$

where $G_\theta(\mathcal{G}, \tilde{\mathcal{H}})$ is the property prediction function, which consists of a stack of the `HyperMP` layer(s), a readout function, and a MLP.

4 Benchmark results

This section highlights the performance of `MolHMPN` as compared to other baseline methods. The training details can be found in Appendix A.2.

4.1 Results

We evaluate the performance of `MolHMPN` with baselines that make use of the pair-wise connectivities (*PAIR*) and subgraphs (*SUB*). This is done to analyze the effectiveness of the usage of pair-wise and higher-order connectivites. For the *PAIR* baselines, we analyze the usage of atom (MPNN (atom only)) (Yang et al., 2019), atom and bonds (MPNN) (Yang et al., 2019), directed bonds (DMPNN) (Yang et al., 2019), and atoms and bonds with enhanced interactions (CMPNN) (Song et al., 2020). For the *SUB* baselines, we compare with baselines that have utilized substructures with nodes that are not connected by an edge (AGCN) (Li et al., 2018), substructure with marginal nodes (GAAN) (Sun et al., 2019), and substructures constructed by junction tree (ML-MPNN) (Wang et al., 2021). The benchmark datasets for the performance evaluation includes Tox21, ClinTox, SIDER, BBBP, BACE, ESOL, FreeSolv and Lipophilicity. The results of the baselines are taken directly from their respective papers, except for CMPNN[2]).

Table 1 shows the overall results of `MolHMPN` on graph classification and regression tasks. From Table 1, we can see that `MolHMPN` has outperformed the other baselines for four out of eight datasets. This shows the efficacy of using both pair-wise and higher-order connectivities, as well as the prior knowledge-guided data-driven scheme. From the *PAIR* results, we can see that the usage of atoms, directed and undirected bond information do not have a significant impact on the performance.

[2]We rerun their condes for all datasets as a mistake was found in their results as stated in their official code https://github.com/SY575/CMPNN.git

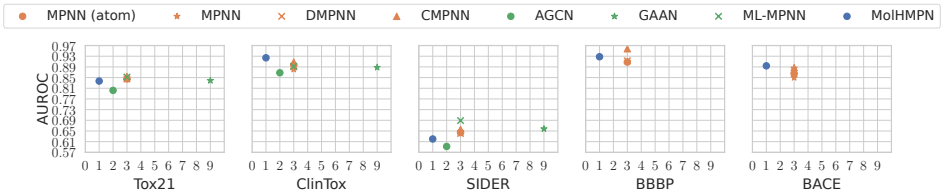


Figure 4: **Number of graph convolutions vs. classification performances**

Table 2: **Increasing K for hyperedge learning.** Comparison between the different K used. Results in bold are the best-performing results for their respective datasets. (↑ means that higher result is better and ↓ means that lower result is better.)

Metric	AUROC					RMSE		
Dataset	Tox21 (↑)	ClinTox (↑)	SIDER (↑)	BBBP (↑)	BACE (↑)	ESOL (↓)	FreeSolv (↓)	Lipophilicity (↓)
MolHMPN-0	**0.838** (± 0.0146)	0.918 (± 0.0426)	0.605 (± 0.0227)	0.927 (± 0.0299)	0.873 (± 0.0232)	0.450 (± 0.0339)	**0.815** (± 0.3606)	0.519 (± 0.0391)
MolHMPN-1	0.837 (± 0.0042)	**0.924** (± 0.0452)	0.614 (± 0.0105)	**0.928** (± 0.0388)	0.888 (± 0.0106)	0.436 (± 0.1324)	0.980 (± 0.4127)	**0.511** (± 0.0672)
MolHMPN-2	0.837 (± 0.0072)	0.912 (± 0.0485)	**0.620** (± 0.0160)	0.903 (± 0.0416)	**0.894** (± 0.0173)	**0.392** (± 0.0917)	1.012 (± 0.5553)	0.533 (± 0.0744)
MolHMPN-3	0.833 (± 0.0717)	0.909 (± 0.0465)	0.608 (± 0.0144)	0.921 (± 0.0202)	0.885 (± 0.0258)	0.406 (± 0.0872)	1.100 (± 0.4243)	0.556 (± 0.0928)

Instead, increasing the interactions between the atoms and bonds in CMPNN gives better results, especially for BBBP, ESOL and FreeSolv. Comparing MolHMPN with the *PAIR* models, we can see that the inclusion of higher-order connectivities is indeed beneficial for the tasks as MolHMPN has outperformed the models for five out of eight datasets. From the *SUB* results, we can see that MolHMPN outperforms the other baselines for three out of six datasets. Also, although ML-MPNN has integrated information from the nodes, edges, subgraphs and graphs, MolHMPN has outperformed it for four out of six datasets. This shows the efficacy of employing chemically-useful representations when conducting the benchmark tasks. Although the *PAIR* models can capture higher-order connectivities by using multiple layers, MolHMPN has outperformed the baselines with only one HyperMP layer as shown in figure 4. The other results can be found in Appendix A.3.

5 Ablation Studies

In this section, we analyze the effects of the hyperedge expansion and different subgraphs usage.

5.1 Hyperedge learning with extended hyperedges

We analyze the performance of MolHMPN with increased K as in Section 3.3. When K increases, the size of each hyperedge increases as it gets further away from the original functional group information. MolHMPN is then able to modify the membership of the nodes in each hyperedge. We refer to MolHMPN with the K-hop extension as MolHMPN-K. For MolHMPN-0, the original functional group hyperedges were used without the additional hyperedge learning scheme.

Table 2 shows the results of the effects of increasing K. From Table 2, we can see that the extended learning strategy has mostly improved the performance of MolHMPN. MolHMPN-1 has generally improved the performance from MolHMPN-0 for six out of eight datasets, where MolHMPN-2 has further improved the performance for two of those datasets (SIDER and ESOL). However, when K is too large (e.g., $K = 3$), performance degradation is observed for most of the datasets. This is because the extended hyperedges has deviated too far away from the original functional groups and often cover all the atoms in $\mathcal{G}$, thus making the hyperedges indistinguishable. From this results, we can see that the domain knowledge-guided hyperedge learning can play a crucial role when modeling higher-order connectivities robustly.

Table 3: **Subgraph Comparison.** Comparison between different types of subgraphs. Results in bold are the best-performing results for their respective datasets. (↑ means that higher result is better and ↓ means that lower result is better.)

Metric	AUROC					RMSE		
Dataset	Tox21 (↑)	ClinTox (↑)	SIDER (↑)	BBBP (↑)	BACE (↑)	ESOL (↓)	FreeSolv (↓)	Lipophilicity (↓)
Ring & C. Bond	0.834 (± 0.0142)	0.904 (± 0.0401)	0.577 (± 0.0339)	0.919 (± 0.0124)	0.884 (± 0.0106)	0.509 (± 0.0547)	1.468 (± 0.5970)	**0.513** (± 0.0475)
2-hop ngh.	0.836 (± 0.0135)	0.902 (± 0.0448)	0.582 (± 0.0271)	0.926 (± 0.0314)	**0.894** (± 0.0240)	**0.431** (± 0.0709)	0.995 (± 0.3844)	0.526 (± 0.0475)
3-hop ngh.	0.830 (± 0.0147)	0.881 (± 0.0363)	0.597 (± 0.0307)	0.918 (± 0.0278)	0.871 (± 0.0136)	0.508 (± 0.1032)	0.984 (± 0.3094)	0.524 (± 0.0889)
MolHMPN-0	**0.838** (± 0.0146)	**0.918** (± 0.0426)	**0.605** (± 0.0227)	**0.927** (± 0.0299)	0.873 (± 0.0232)	0.450 (± 0.0339)	**0.815** (± 0.3606)	0.519 (± 0.0391)

5.2 SUBGRAPH COMPARISONS

We evaluate the performance of MolHMPN with other methods that employs other kinds of substructures. We do this by assessing the effectiveness of employment of the functional group information as compared to the baseline methods, which are known to effective in solving molecule generation and graph meta-learning tasks. Since we are making comparison based on the substructure types only, we analyze the results using MolHMPN-0. The baseline methods are (1) "Ring & Chemical Bond" which utilizes the ring structure and chemical bonds as the subgraph[3] (Jin et al., 2020) and (2) "K-hop neighbors" which utilizes the K-hop neighbors as substructures (Huang & Zitnik, 2020). In the following experiments, we replace the hyperedge construction rules with those of the baseline methods, and assess their performances with our benchmark datasets. Other than the hyperedge constructions, we use the same experiment setups as in Appendix A.2.

Table 3 shows the results where different types of subgraphs are used. From Table 3, MolHMPN-0 has outperformed the other methods for five out of eight datasets, especially for ClinTox and FreeSolv. For SIDER, MolHMPN has outperformed the other methods and has the smallest standard deviation. For BBBP, although 2-hop neighbor is comparable with MolHMPN-0, MolHMPN-0 has a smaller standard deviation. This is also the case for ESOL, where MolHMPN-0 has a smaller standard deviation even though it is comparable with 2-hop neighbor. One notable trend is that the 3-hop neighbor underperforms as compared to the 2-hop neighbor even though it can model higher-order connectivities. However, this is not observed in MolHMPN-0 even though we also employ up to 3-hop neighbors for the functional groups as we used chemically meaningful substructures. Hence, this shows the efficacy of employing chemically meaningful and valid substructures (functional groups in our case) when conducting molecular properties prediction tasks.

6 CONCLUSION

We propose a molecular hyper-message passing network (MolHMPN) to integrate pair-wise and higher-order connectivities for molecular properties prediction using domain knowledge-guided learnt substructures. We construct the hypergraph representation of the molecules using chemically-valid functional groups, update the nodes and hyperedge features in the HyperMP layer, and learn the substructures from the constructed substructures. We evaluate the performance of our model with several baseline methods, and show that our model is able to achieve outstanding results with only one HyperMP layer. In our ablation study, we show that using domain knowledge-guided learnt substructure improves the performance of the benchmark tasks. We also compare the usage of different types of substructures using the same model architecture and show the efficacy of employing chemically meaningful and valid substructures.

[3]In original paper, the frequently-occurring chemical substructures are also considered. However, in our benchmark datasets, none of the dataset satisfies the proposed value for occurrence frequency.

Ethics statements Although the proposed method has shown its potential in molecular property prediction tasks, overreliance on such methods might lead to the neglection of the possible side effects that these molecules have on humans and their potential negative impact when released to the environment since these information are not given in the datasets.

Reproducibility As machine learning researchers, we consider the reproducibility of numerical results as one of the top priorities. Thus, we put a significant amount of effort into pursuing the reproducibility of our experimental results. As such, we set and tracked the random seed used for our experiments and confirmed the experiments were reproducible.

REFERENCES

Aysha Akhtar. The flaws and human harms of animal experimentation. *Cambridge quarterly of healthcare ethics : CQ : the international journal of healthcare ethics committees*, 24:407–19, 09 2015. doi: 10.1017/S0963180115000079.

Emily Alsentzer, Samuel Finlayson, Michelle Li, and Marinka Zitnik. Subgraph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8017–8029. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/5bca8566db79f3788be9efd96c9ed70d-Paper.pdf.

Song Bai, Feihu Zhang, and Philip H. S. Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognit.*, 110:107637, 2021. doi: 10.1016/j.patcog.2020.107637. URL https://doi.org/10.1016/j.patcog.2020.107637.

A.G. Blackman. *Chemistry, 4th Edition*. John Wiley & Sons, Incorporated, 2019. ISBN 9780730355038. URL https://books.google.co.kr/books?id=AssQvwEACAAJ.

N Benjamin Erichson, Michael Muehlebach, and Michael W Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. *AAAI 2019*, 2018.

U.S. Food and Drug Administration. The beginnings: Laboratory and animal studies, 2015. URL https://www.fda.gov/drugs/information-consumers-and-patients-drugs/beginnings-laboratory-and-animal-studies.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *International Conference on Learning Representations (ICLR)*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5862–5874. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/412604be30f701b1b1e3124c252065e6-Paper.pdf.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2323–2332. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/jin18a.html.

Wengong Jin, Dr.Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4839–4848. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jin20a.html.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Masaaki Kotera, Andrew G McDonald, Sinéad Boyce, and Keith F Tipton. Functional group and substructure searching as a tool in metabolomics. *PLoS One*, 3(2):e1537, 2008.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. *CoRR*, abs/1801.03226, 2018. URL http://arxiv.org/abs/1801.03226.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pp. 3208–3216. PMLR, 2018.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Emmanuel Noutahi, Dominique Beaini, Julien Horwood, Sébastien Giguère, and Prudencio Tossou. Towards interpretable sparse graph representation learning with laplacian pooling, 2020.

Junyoung Park and Jinkyoo Park. Physics-induced graph neural network: An application to wind-farm power estimation. *Energy*, 187:115883, 2019.

Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10764–10773, 2019. doi: 10.1109/CVPR.2019.01103.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification, 2020.

Sungyong Seo, Chuizheng Meng, and Yan Liu. Physics-aware difference graph networks for sparsely-observed dynamics. In *International Conference on Learning Representations*, 2019.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

Ying Song, Shuangjia Zheng, Zhangming Niu, Zhang-hua Fu, Yutong Lu, and Yuedong Yang. Communicative representation learning on attributed molecular graphs. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2831–2838. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/392. URL https://doi.org/10.24963/ijcai.2020/392. Main track.

Penghui Sun, Jingwei Qu, Xiaoqing Lyu, Haibin Ling, and Zhi Tang. Graph attribute aggregation network with progressive margin folding, 2019.

Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, and Shuiwang Ji. Advanced graph and sequence neural networks for molecular property prediction and drug discovery, 2021.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237. URL https://doi.org/10.1021/acs.jcim.9b00237. PMID: 31361484.

Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation, 2021.

Yuan Yin, Vincent LE GUEN, DONA Jérémie, Emmanuel de Bezenac, Ibrahim Ayed, Nicolas THOME, et al. Augmenting physical models with deep networks for complex dynamics forecasting. In *International Conference on Learning Representations*, 2020.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/e77dbaf6759253c7c6d0efc5690369c7-Paper.pdf.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d80b7040b773199015de6d3b4293c8ff-Paper.pdf.

A APPENDIX

A.1 HYPERGRAPH CONSTRUCTION

In this section, we provide the list of functional groups that have been utilized in our current study based on their central atoms. We highlight the central atoms and their respective first- and second-hop neighbors with circles of different colors.

Table A.1: **Functional groups with nitrogen as the central atom.** The red circles represent the central atoms, and the blue and green circles represent the 1-hop and 2-hop neighbors from the central atom respectively.

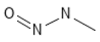
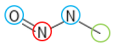
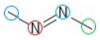
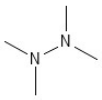
Functional group	Structure	Hyperedge	Functional group	Structure	Hyperedge
Amine			Nitro		
Nitrate			C nitroso		
N nitroso			Azo		
Hydrazine			Hydroxylamine		
Nitrile					

Table A.2: **Functional groups with carbon as the central atom.** The red circles represent the central atoms, and the blue and green circles represent the 1-hop and 2-hop neighbors from the central atom respectively.

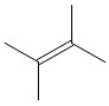
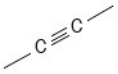
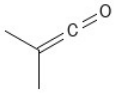
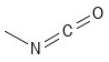
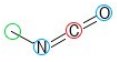
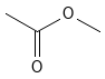
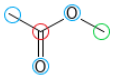
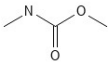
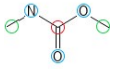
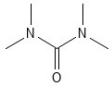
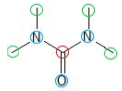
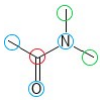
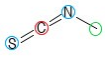
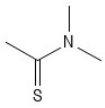
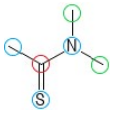
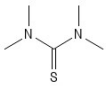
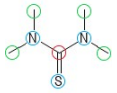
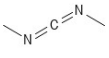
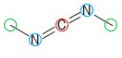
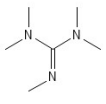
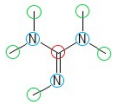
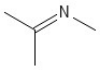
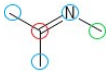
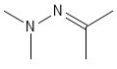
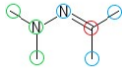
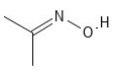
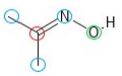
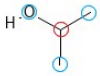
Functional group	Structure	Hyperedge	Functional group	Structure	Hyperedge
Alkene			Alkyne		
Aldehyde			Ketene		
Isocynate			Carboxyl		
Carbamate			Carbamide		
Amide			Ketone		
Isothiocynate			Thione		
Thioamide			Thiourea		
Carbodiimide			Carboximidamide		
Imine			Hydrazone		
Oxime			Alcohol		
Thiol			Allene		

Table A.3: **Functional groups with oxygen as the central atom.** The red circles represent the central atoms, and the blue and green circles represent the 1-hop and 2-hop neighbors from the central atom respectively.

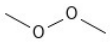
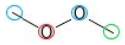
Functional group	Structure	Hyperedge	Functional group	Structure	Hyperedge
Ether			Peroxide		

Table A.4: **Functional groups with phosphorus as the central atom.** The red circles represent the central atoms, and the blue and green circles represent the 1-hop and 2-hop neighbors from the central atom respectively.

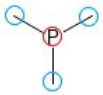
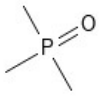
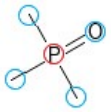
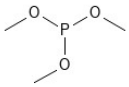
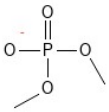
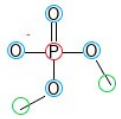
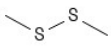
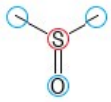
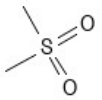
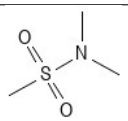
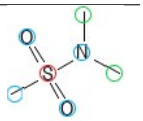
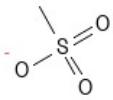
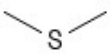
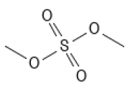
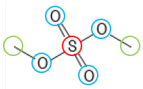
Functional group	Structure	Hyperedge	Functional group	Structure	Hyperedge
Phosphanyl			Phosphine oxide		
Phosphite ester			Phosphodiester		

Table A.5: **Functional groups with sulfur as the central atom.** The red circles represent the central atoms, and the blue and green circles represent the 1-hop and 2-hop neighbors from the central atom respectively.

Functional group	Structure	Hyperedge	Functional group	Structure	Hyperedge
Disulfide			Sulfoxide		
Sulfone			Sulfonamide		
Sulfonate			Thioether		
Sulfate					

A.2 TRAINING DETAILS

In this section, we provide the data and training details.

Data details. The dataset information are given in Table A.6. The atom and bond features that are used as the initial node and edge features are given in Tables A.7 and A.8 respectively. We use the BaseAtomFeaturizer and BaseBondFeaturizer of DGL-LifeSci to extract features from the initial atom and bond features. The hypergraphs were constructed using DGL and Networkx.

Table A.6: Datasets types, number of tasks, performance metric and split type

Dataset	Task	Number of tasks	Metric	Split
Tox21	Classification	12	AUROC	Random
ClinTox	Classification	2	AUROC	Random
SIDER	Classification	27	AUROC	Random
BBBP	Classification	1	AUROC	Random
BACE	Classification	1	AUROC	Random
ESOL	Regression	1	RMSE	Random
FreeSolv	Regression	1	RMSE	Random
Lipophilicity	Regression	1	RMSE	Random

Table A.7: Atom features used to featurize the node features

Atom Features	Number of Features
atom type one hot	43
atomic number	1
atom mass	1
atom degree one hot	11
atom explicit valence one hot	6
atom implicit valence one hot	7
atom total num H one hot	5
atom formal charge one hot	5
atom hybridisation one hot	5
atom num radical electrons one hot	5
atom is aromiatic one hot	2
atom is in ring one hot	2
atom chiral tag one hot	4
atom chirality type one hot	2
atom is chiral center	1

Table A.8: Bond features used to featurize the edge features

Bond Features	Number of Features
bond type one hot	4
bond is in ring	1
bond is conjugated one hot	2

Training details. For our tasks, we randomly split the datasets into 80:10:10 ratio as the training, validation and test sets and take the average of the results from different 5 random seeds (0 to 4). We use the AdamP optimizer (Heo et al., 2021) whose learning rate is scheduled by the CosineAnnealing scheduler(Loshchilov & Hutter, 2016). The loss functions for the classification and regression tasks are the binary cross-entropy (BCE) loss and mean squared error (MSE) respectively. We give extra weights to the minority class in the loss functions for the classification datasets based on the ratio of the minority to majority class of each task to handle the class imbalance problems. The attentive sum and max function are used as the readout function of $G_\theta(\cdot)$. We use a batch size of 512, run the models for 500 epochs and initialized the learning rate as 0.001. For $F_\theta(\cdot)$ and $G_\theta(\cdot)$, we use only one `HyperMP` layer each. The training details can be found in Table A.9 and A.10.

Table A.9: Hyperparameters for `MolHMPN`-0

Dataset	x_k	Cycles	GNN dropout	Regressor dropout	MLP neurons	Latent dimensions
Tox21	ZERO	FALSE	0.2	0.2	[64]	128
ClinTox	ZERO	FALSE	0.3	0.3	[64, 32]	128
SIDER	MEAN	FALSE	0.0	0.1	[64]	128
BBBP	MEAN	FALSE	0.0	0.0	[128]	256
BACE	MEAN	TRUE	0.2	0.0	[64, 32]	128
ESOL	MEAN	TRUE	0.0	0.0	[128]	256
FreeSolv	MEAN	FALSE	0.4	0.4	–	128
Lipophilicity	MEAN	FALSE	0.2	0.2	–	128

Table A.10: Hyperparameters for `MolHMPN`-1,2,3

Dataset	x_k	Cycles	GNN dropout	Theta dropout	Regressor dropout	MLP neurons	Latent dimensions
Tox21	ZERO	FALSE	0.2	0.2	0.2	[64]	128
ClinTox	ZERO	FALSE	0.3	0.3	0.3	[128]	256
SIDER	MEAN	FALSE	0.0	0.0	0.1	[64]	128
BBBP	MEAN	FALSE	0.0	0.0	0.0	[128, 64]	256
BACE	MEAN	TRUE	0.0	0.0	0.0	[128]	256
ESOL	MEAN	TRUE	0.0	0.0	0.0	[128]	256
FreeSolv	MEAN	FALSE	0.4	0.4	0.4	–	128
Lipophilicity	MEAN	FALSE	0.2	0.2	0.2	–	128

A.3 ADDITIONAL RESULTS

In this section, we provide the extended plots for showing the benchmark results of baseline models with their number of graph convolutions.

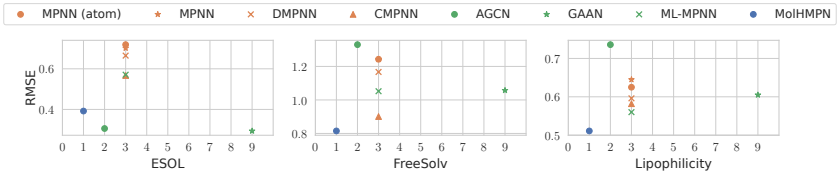


Figure A.1: **Number of graph convolutions vs. classification performances**