1 Additional Experiment Settings

In the finetuning process, we consider 8 benchmark binary classification datasets from MoleculeNet. The detailed statistics of datasets are listed in Table 1. The detailed information of these benchmark datasets are listed below:

- **muv** is a subset of PubChem BioAssay by applying a refined nearest neighbor analysis. It is designed for the validation of virtual screening techniques.
- **clintox** compares drugs approved through FDA and drugs eliminated due to the toxicity during clinical trials.
- **sider** records marketed drugs along with its adverse drug reactions, also known as the side effect resource.
- **hiv** records the experimentally measured abilities to inhibit HIV replication.
- **tox21** is a public database measuring the toxicity of compounds, which has been used in the 2014 Tox21 Data Challenge.
- **bace** is collected for recording compounds which could act as the inhibitors of (BACE-1) in the past few years.
- **toxcast** contains multiple toxicity labels over thousands of compounds by running high-throughput screening tests on thousands of chemicals.
- **bbbp** involves records of whether a compound carries the permeability property of penetrating the blood-brain barrier.

Table 1: Dataset statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>numv</th>
<th>clintox</th>
<th>sider</th>
<th>hiv</th>
<th>tox21</th>
<th>bace</th>
<th>toxcast</th>
<th>bbbp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of molecules</td>
<td>93087</td>
<td>1478</td>
<td>1427</td>
<td>41127</td>
<td>7831</td>
<td>1513</td>
<td>8575</td>
<td>2039</td>
</tr>
<tr>
<td>Number of tasks</td>
<td>17</td>
<td>2</td>
<td>27</td>
<td>1</td>
<td>12</td>
<td>1</td>
<td>617</td>
<td>1</td>
</tr>
</tbody>
</table>

2 Multi-level Self-supervised Pre-training

Here, we show the pseudo code of the multi-level self-supervised pre-training (Algorithm 1). We adopt the MGDA-UB algorithm from multi-task learning to efficiently solve the optimization problem. Since MGDA-UB calculates the weights $\lambda_i$ by Frank-Wolfe algorithm in each training step, we do not have to provide the weights explicitly.

The input of Multi-level self-supervised pre-training is the molecule datasets, a list of self-supervised tasks. The output is the model parameters of pre-trained GNNs. At the beginning of pre-training, we first randomly initialized the task-specific parameters and the parameters of the pre-trained GNN model. In each training iteration, we update the parameters of all the pretext models. Then we...
Algorithm 1 Multi-level Self-supervised Pre-training

**Input**: Set of input molecules $X$, set of self-supervised tasks $T$, learning rate $\alpha$  

**Output**: Pre-trained parameter $\theta$;  

1: $T = \{\text{atom}, \text{bond}, \text{motif}\}$  
2: Initialize self-supervised task-specific parameters, $\phi_{\text{atom}}, \phi_{\text{bond}}$ and $\phi_{\text{motif}}$.  
3: Initialize the parameters of pre-trained GNN model, $\theta$.  
4: **while** not convergence **do**  
5: Randomly sample and pre-process input molecules $X$, generate self-supervised labels $Y_{\text{atom}}, Y_{\text{bond}}, Y_{\text{motif}}$.  
6: **for** $i \in T$ **do**  
7: $\phi_i \leftarrow \phi_i - \alpha \nabla \phi_i \mathcal{L}_i(X, Y_i, \theta, \phi_i)$  
8: **end for**  
9: $\lambda_{\text{atom}}, \lambda_{\text{bond}}, \lambda_{\text{motif}} \leftarrow \text{FRANKWOLFE} (\theta, \phi_{\text{atom}}, \phi_{\text{bond}}, \phi_{\text{motif}})$  
10: $\theta \leftarrow \theta - \sum_{i \in T} \lambda_i \nabla Z \mathcal{L}_i(X, Y_i, \theta, \phi_i), Z = \text{GNN}(X; \theta)$  
11: **end while**

apply the Frank-Wolfe algorithm [1] to choose the weights, which solves the following optimization problem:

$$
\min_{\lambda_i, i \in T} \left\{ \left\| \sum_{i \in T} \lambda_i \nabla Z \mathcal{L}_i(X, Y_i, \theta, \phi_i) \right\|_2^2, \sum_{i \in T} \lambda_i = 1, \lambda_i \geq 0 \right\},
$$

(1)

where $Z = \text{GNN}(X; \theta)$ is the representation of molecules. $\nabla Z \mathcal{L}_i(X, Y_i, \theta, \phi_i)$ can be computed in a single backward pass for all tasks. With the calculated weights, we can update the parameters of pre-trained GNN model. More details can be found in the original paper of MGDA-UB [2].

References
