MolTailor: Tailoring Chemical Molecular Representation to Specific Tasks via Text Prompts

Haoqiang Guo, Sendong Zhao*, Haochun Wang, Yanrui Du, Bing Qin

Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
{hqguo, sdzhao, hchwang, yrdu, bqing}@ir.hit.edu.cn

Abstract

Deep learning is now widely used in drug discovery, providing significant acceleration and cost reduction. As the most fundamental building block, molecular representation is essential for predicting molecular properties to enable various downstream applications. Most existing methods attempt to incorporate more information to learn better representations. However, not all features are equally important for a specific task. Ignoring this would potentially compromise the training efficiency and predictive accuracy. To address this issue, we propose a novel approach, which treats language models as an agent and molecular pretraining models as a knowledge base. The agent accentuates task-relevant features in the molecular representation by understanding the natural language description of the task, just as a tailor customizes clothes for clients. Thus, we call this approach **MolTailor**. Evaluations demonstrate MolTailor's superior performance over baselines, validating the efficacy of enhancing relevance for molecular representation learning. This illustrates the potential of language model guided optimization to better exploit and unleash the capabilities of existing powerful molecular representation methods. Our code and appendix are available at https://github.com/SCIR-HI/MolTailor.

Introduction

In recent years, AI technology has been widely applied to various stages of drug design, such as compound synthesis and screening, etc. (Mamoshina et al. 2016). This has greatly improved the efficiency and reduced the cost of drug development. Molecular representation learning serves as the cornerstone for AI techniques in drug design, empowering a wide range of downstream tasks such as molecular property prediction and drug-drug interaction judgment (Xia et al. 2023). Molecular representations are essentially vector embeddings for molecules, analogous to word embeddings in NLP. The idea of encoding molecules as mathematical vectors dates back to the 1940s (Wiener 1947).

The development of molecular representation learning can be roughly divided into four phases. In the first phase, molecular representations were constructed through expert knowledge, which can be categorized into two types: descriptors (Wiener 1947) and fingerprints (Rogers and Hahn 2010). Descriptors focus on the inherent chemical characteristics of molecules, while fingerprints contain structural information. In the second phase, inspired by deep learning, some works (Gilmer et al. 2017; Yang et al. 2019; Song et al. 2020) started to learn molecular representations from labeled data using supervised learning. However, the scarcity of labeled data, due to the expensive experimental cost and time required, limits further improvements to model performance and generalization.

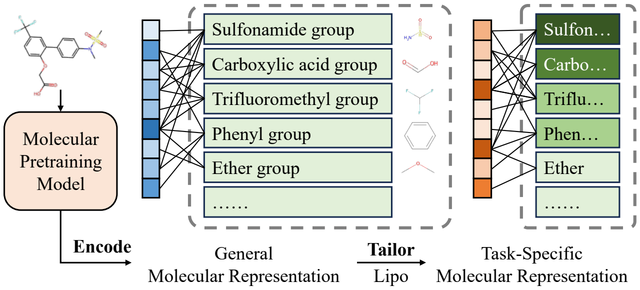

Figure 1: Most existing molecular pretraining models (e.g. Grover, MolCLR) attempt to encode as much molecular information as possible (e.g. various functional groups and molecular weight) into a vector to obtain general molecular representation. However, for specific downstream tasks (e.g. Lipo, predicting lipophilicity of compounds), features are not equally important (e.g. Sulfonamide and Carboxylic acid groups significantly increase the hydrophilicity, being more critical than the remaining groups). By understanding task descriptions, MolTailor adjusts the weights of different features in the representation to obtain task-specific molecular representation.

The success of pretrained language models in NLP (Devlin et al. 2018) has demonstrated the potential of self-supervised learning. Building upon this, in the third phase, some works pretrained sequence-based models to learn molecular representation by mimicking successful NLP pretrained language models. (Chithrananda, Grand, and Ramsundar 2020; Kim et al. 2021; Ross et al. 2022). Concurrently, others pretrained graph-based models (Hu et al. 2019; You et al. 2020). More recently, in the fourth phase, researchers are no longer limited to pretraining solely on the

[*] Corresponding author

molecular data itself. Instead, they are attempting to incorporate additional information, such as knowledge graphs (Fang et al. 2022) and textual information (Zeng et al. 2022; Edwards et al. 2022; Su et al. 2022; Chen et al. 2023). Notably, molecule-text multimodal learning has received increasing attention recently and achieved promising results.

As introduced above, most existing works merely strive to inject more information into the representations without utilizing task information as prior knowledge, which could compromise the training efficiency, as shown in Fig. 1. Inspired by this observation, we propose a new method called **MolTailor**, where a language model acts as a tailor, adapting general molecular representations (ready-made clothing) into task-specific ones (customized clothing) based on user needs.

To achieve this goal, we adopt a dual-tower model structure with one language pretraining module and one molecular pretraining module, joint together through a cross-attention module. Meanwhile, we constructed a new pretraining task: **M**olecule-**T**ext **M**ulti-**T**ask **R**egression (**MT-MTR**). The dataset for this task consists of (molecule, task description, regression labels) triplets. Here, the molecule is represented as a SMILES string (Weininger 1988), the task description is a text prompt describing molecular properties that are most helpful for solving the task, and the regression labels are values of properties mentioned in the task description. The model needs to predict the regression labels based on the SMILES and task description. This pretraining task teaches the model to enhance the weights of task-relevant properties in the molecular representation according to the task description. Our contributions can be summarized as follows:

- We propose MolTailor, the first approach to generate task-specific molecular representations via text prompts, which provides a new perspective on text-molecule multimodal learning: not only injecting the knowledge in texts into molecular representations but also utilizing the reasoning capacity of language models.
- We construct MT-MTR, a new molecule-text multimodal pretraining task. This task teaches the model the capabilities of instruction following and adapting molecular representations.
- We comprehensively evaluate across eight subsets of the MoleculeNet, thereby demonstrating the effectiveness of task-specific molecular representation learning.

Related Work

Molecular Pretraining Models. Models can be roughly categorized into the following three types: sequence-based, graph-based, and external knowledge (Xia et al. 2023). **1. Sequence-based**: Representing molecules as SMILES strings or other sequences and using language models as backbones for pretraining. SMILES-BERT (Wang et al. 2019) and ChemBERTa (Chithrananda, Grand, and Ramsundar 2020) use Masked Language Modeling (MLM) for pretraining tasks, while CHEM-BERT (Kim et al. 2021), ChemBERTa-2 (Ahmad et al. 2022) additionally incorporate property prediction tasks. MM-Deacon (Guo et al. 2021) uses contrastive learning with SMILES and INPAC as parallel inputs. **2. Graph-based**: Representing molecules as graphs and using graph neural networks for pretraining. GROVER (Rong et al. 2020) and Mole-BERT (Xia et al. 2022) use Mask Component Modeling (MCM) for pretraining, while GraphCL (You et al. 2020) and MolCLR (Wang et al. 2022) employ contrastive learning. Denoising (Zaidi et al. 2022) and GeoSSL (Liu, Guo, and Tang 2022) are trained with denoising methods inspired by denoising diffusion models. **3. External knowledge**: GraphMVP (Liu et al. 2021) and 3D Infomax (Stärk et al. 2022) incorporate 3D structures as supplementary information. KCL (Fang et al. 2022) uses knowledge graphs to enhance molecular representations.

Image-Text Multimodal Pretraining Models. Models can be divided into dual-tower and single-tower structures based on architectures: **1. Dual-tower structure**: ViLBERT (Lu et al. 2019) introduces the co-attentional transformer (Co-TRM) layer, which effectively integrates information from both modalities. CLIP (Radford et al. 2021) adopts contrastive learning for pretraining and achieves significant performance gains. The CoCa (Yu et al. 2022) model uses dual towers of an image encoder and a text decoder. **2. Single-tower structure**: VisualBERT (Li et al. 2019) takes texts and images as a unified input for learning. ViLT (Kim, Son, and Kim 2021) adopts a patch projection approach, greatly improving the processing speed. BeiT-3 (Wang et al. 2023) uses distinct Feedforward Neural Network (FFN) layers for different modalities while sharing attention modules.

Molecule-Text Multimodal Pretraining Models. Similarly, models can be divided into dual-tower and single-tower structures based on architectures: **1. Single-tower structure**: Using a language model as the backbone for further pretraining. KV-PLM (Zeng et al. 2022) and MolXPT (Liu et al. 2023) inject SMILES into literature text by locating molecule names to obtain mixed corpora and conduct MLM and language pretraining (LM) respectively. MolT5 (Edwards et al. 2022) does replace corrupted spans pretraining on molecular and text data, meanwhile, uses mutual translation between molecules and textual descriptions as downstream tasks. Text+Chem T5 (Christofidellis et al. 2023) and ChatMol (Zeng et al. 2023) use multi-task learning for pretraining. GIMLET (Zhao et al. 2023) takes graphs and texts as a unified input and uses instruction-based supervised property prediction for pretraining. **2. Dual-tower structure**: MoMu (Su et al. 2022) and MoleculeSTM (Liu et al. 2022) do contrastive learning on molecule-description pairs. CLAMP (Seidl et al. 2023) does contrastive learning on molecule-bioassay pairs. Additionally, ChemCrow (Bran et al. 2023) enhances large language models (LLMs) with chemical tools to accomplish real-world chemical tasks.

Methodology

Fig. 2 presents an overview of our proposed approach. In this section, we introduce MolTailor in three aspects: 1. Construction of the MT-MTR corpus. 2. Model architecture of MolTailor. 3. Pretraining of MolTailor and application of MolTailor to downstream tasks.

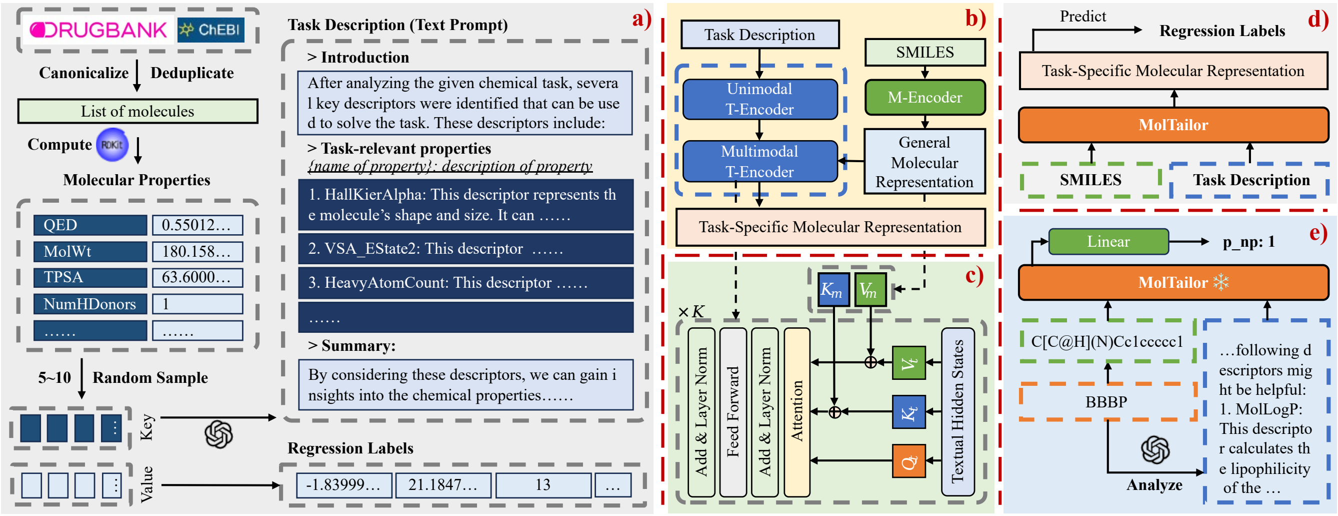
DRUGBANK
ChEBI
Canonicalize
Deduplicate
List of molecules
Compute
RDKit
Molecular Properties
QED
0.55012…
MolWt
180.158…
TPSA
63.6000…
NumHDonors
1
……
……
5~10
Random Sample
Key
Value
Task Description (Text Prompt)
a)
> Introduction
After analyzing the given chemical task, several key descriptors were identified that can be used to solve the task. These descriptors include:
> Task-relevant properties
{name of property}: description of property
1. HallKierAlpha: This descriptor represents the molecule's shape and size. It can ……
2. VSA_EState2: This descriptor ……
3. HeavyAtomCount: This descriptor ……
……
> Summary:
By considering these descriptors, we can gain insights into the chemical properties……
Regression Labels
-1.83999…
21.1847…
13
…
b)
Task Description
SMILES
Unimodal T-Encoder
M-Encoder
Multimodal T-Encoder
General Molecular Representation
Task-Specific Molecular Representation
c)
×K
Add & Layer Norm
Feed Forward
Add & Layer Norm
Attention
K_m
V_m
V_t
K_t
Q
Textual Hidden States
d)
Predict
Regression Labels
Task-Specific Molecular Representation
MolTailor
SMILES
Task Description
e)
Linear
p_np: 1
MolTailor ❄
C[C@H](N)Cc1ccccc1
BBBP
Analyze
…following descriptors might be helpful:
1. MolLogP: This descriptor calculates the lipophilicity of the …

Figure 2: Overview of the MolTailor framework. a) The construction process of the MT-MTR dataset. We obtain representative molecules from DrugBank (Wishart et al. 2018) and ChEBI (Hastings et al. 2016), and then use RDKit to calculate 209 properties for each molecule. For each molecule, we randomly sample 5-10 properties from the property set, use the property names to generate virtual task descriptions via GPT-3.5, and use the property values as regression labels. b) Model architecture of MolTailor. MolTailor consists of a language pretraining model (T-Encoder) and a molecular pretraining model (M-Encoder). The T-Encoder is divided into a unimodal part (for understanding task descriptions) and a multimodal part (for adjusting molecular representations). c) Internal structure of the Multimodal T-Encoder. It modifies the original Transformer Encoder Block to perform self-attention and cross-attention operations simultaneously: mapping the general molecular representation to obtain K_m and V_m vectors which are then concatenated with textual vectors K_t and V_t. d) Pretraining task of MolTailor. The model needs to predict properties mentioned in the task description based on the molecule and text prompt. e) Downstream tasks of MolTailor. For a specific downstream task, we first generate the task description in the same format as pretraining via GPT-4 analysis, then take the SMILES and task description as input to predict labels for the corresponding task.

Construction of MT-MTR

The construction process of the MT-MTR (Molecule-Text Multi-Task Regression) dataset consists of 3 main steps as shown in Fig. 2a:

Step 1: Obtain representative molecules. We obtain molecules from DrugBank (Wishart et al. 2018) and ChEBI (Hastings et al. 2016), then use RDKit[1] to deduplicate and canonicalize, resulting in 55,759 valid SMILES.

Step 2: Calculate molecular properties. For each molecule obtained in the previous step, we use RDKit to calculate 209 properties[2]. Among these properties, some are continuous values while others are discrete values. We unify them into regression tasks.

Step 3: Obtain task descriptions and regression labels. For each molecule, we randomly sample 5-10 properties from the property set obtained in the previous step. The sampled property names are filled into the prompt template in Tab. 1 (Prompt for Pretraining) to obtain text input fed into GPT for generating virtual task description. Since GPT-4 and GPT-3.5 achieve similar performance on this task, we use the more cost-effective GPT-3.5 for generation in our experiments. The values of the sampled properties are used as the regression labels.

The resulting MT-MTR corpus contains (molecule, task description, regression labels) triplets. It is also worth noting that MT-MTR does not ask the model to predict all properties of the molecules; it only needs to predict properties mentioned in the text prompt. This aims to teach the model to generate molecule representations tailored to the text for improved predictions.

Model Architecture of MolTailor

As illustrated in Fig. 2b&c, MolTailor consists of a language pretraining model (T-Encoder) and a molecular pretraining model (M-Encoder). The language model is divided into a Unimodal Text Encoder (UT-Encoder) and a Multimodal Text Encoder (MT-Encoder) by introducing a few parameters. We treat the M-Encoder as a knowledge base and the T-Encoder as an agent. The agent obtains the final desired representation by understanding natural language and adjusting the well-initialized molecular representation from the knowledge base. The UT-Encoder captures semantic information and the MT-Encoder then tailors molecular representations based on understanding task descriptions.

Since any molecular pretraining model can serve as the M-Encoder in our method, in this section we mainly describe the Transformer encoder block (TEB), the fundamental component composing mainstream language models, as

[1] https://www.rdkit.org

[2] RDKit's "Descriptor" module provides 209 descriptors, which we uniformly refer to as "properties" in this paper for clarity.

	Template
Prompt for Pre-training	As a seasoned expert in the field of chemistry, your task is to analyse a chemical task. And you found following properties of chemical compounds can help solve this task. Please summarize your analysis. The length should be less than 300 tokens. **Properties**: {*Sampled Properties*}. **Task analysis results**:
Prompt for Down-stream	**Example**: {*Example of task description from MT-MTR*} **Property Names**: {*209 property names from RDKit*} Please analyze the Task Name. When discussing the properties related to this task, list any properties that you think may be helpful for solving the task. Simply provide the analysis results directly in less than 400 tokens, referring to the example for guidance.

Table 1: Prompt templates for interacting with GPT to generate task descriptions.

well as the internal structure of the MT-Encoder. Additionally, in our experiments, we tested two types of M-Encoder: CHEM-BERT (Kim et al. 2021) and ChemBERTa-2 (Ahmad et al. 2022).

Transformer Encoder Block. As the basic building block of the Transformer (Vaswani et al. 2017) encoder, it consists of four main components: Multi-Head Self-Attention (MHA), Feed Forward Network (FFN), and Residual Connection with Layer Normalization (LN). For MHA, given hidden states $\boldsymbol{x} \in \mathbb{R}^{(bs \times n \times d)}$ where bs indicates batch size, n indicates sequence length, and d indicates embedding dimension, it is first mapped to vectors $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ by matrices $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{(d \times d)}$, keeping dimensions unchanged. Then, with h heads, d is split into h parts. After that, vectors are transposed and reshaped to get $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{(bs \times h \times n \times d/h)}$. Finally, attention is computed:

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{V}^{\boldsymbol{T}}}{\sqrt{d_k}}\right) \tag{1}$$

$$\mathrm{MHA}(\boldsymbol{x}) = \mathrm{Attention}(f_Q^r(\boldsymbol{x}), f_K^r(\boldsymbol{x}), f_V^r(\boldsymbol{x})) \tag{2}$$

where f_*^r means $f^{reshape}(\boldsymbol{x}\boldsymbol{W}_*)$. The output $\boldsymbol{x}^* \in \mathbb{R}^{(bs \times n \times d)}$ of the MHA goes through the FFN, residual connection and LN to obtain the final output $\boldsymbol{x} \in \mathbb{R}^{(bs \times n \times d)}$.

$$\mathrm{FFN}(\boldsymbol{x}) = \mathrm{ReLU}(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{3}$$

$$\boldsymbol{x} = \mathrm{LN}(\boldsymbol{x} + \mathrm{FFN}(\mathrm{MHA}(\boldsymbol{x}))) \tag{4}$$

Multimodal T-Encoder. Since the task descriptions contain biomedical terminology, we chose PubMedBERT (Gu et al. 2021) as the backbone for the T-Encoder. Additionally, we also present in the appendix the results of using BioLinkBERT (Yasunaga, Leskovec, and Liang 2022). PubMedBERT contains 12 TEBs. We designate the first 9 layers as the UT-Encoder, and the remaining 3 layers as the MT-Encoder.

To build the MT-Encoder, we draw on previous work (Chen et al. 2022) to modify the original MHA to MHA^* that can simultaneously perform Self-Attention and Cross-Attention operations, by introducing very few parameters. Specifically, the MT-Encoder takes $\boldsymbol{x}_t$ and $\boldsymbol{x}_m$ as inputs, where $\boldsymbol{x}_t \in \mathbb{R}^{(bs \times n_t \times d_t)}$ denotes the hidden states from the UT-Encoder, and $\boldsymbol{x}_m \in \mathbb{R}^{(bs \times n_m \times d_m)}$ denotes the general molecular representation (i.e., the last hidden states from the M-Encoder). MHA^* computes as follows:

$$\begin{aligned} \boldsymbol{Q}^* &= f^r(\boldsymbol{x}_t \boldsymbol{W}_Q^t) \\ \boldsymbol{K}^* &= f^r([\boldsymbol{x}_t \boldsymbol{W}_K^t, \boldsymbol{x}_m \boldsymbol{W}_K^m]) \\ \boldsymbol{V}^* &= f^r([\boldsymbol{x}_t \boldsymbol{W}_V^t, \boldsymbol{x}_m \boldsymbol{W}_V^m]) \\ \mathrm{MHA}^*(\boldsymbol{x}_t, \boldsymbol{x}_m) &= \mathrm{Attention}(\boldsymbol{Q}^*, \boldsymbol{K}^*, \boldsymbol{V}^*) \end{aligned} \tag{5}$$

where f^r denotes reshape function, $[*]$ denotes concatenation along the dimension corresponding to the sequence length n. The output of the MT-Encoder Block is computed:

$$\boldsymbol{x}^* = \mathrm{LN}(\boldsymbol{x}_t + \mathrm{FFN}(\mathrm{MHA}^*(\boldsymbol{x}_t, \boldsymbol{x}_m))) \tag{6}$$

Chen et al. (2022) formally proves that $\mathrm{MHA}^*(\boldsymbol{x})$ is equivalent to a weighted average of self-attention and cross-attention:

$$\begin{aligned} \mathrm{MHA}^*(\boldsymbol{x}_t, \boldsymbol{x}_m) = {} & (1 - \lambda(\boldsymbol{x}_t))\mathrm{MHA}(\boldsymbol{x}_t) \\ & + \lambda(\boldsymbol{x}_t)\mathrm{MHA}(\boldsymbol{Q}_t, \boldsymbol{K}_m, \boldsymbol{V}_m) \end{aligned} \tag{7}$$

Pretraining and Downstream Application

Pretraining on MT-MTR. We use MT-MTR as the pretraining objective . In detail, given SMILES ($\boldsymbol{z}_m$) and task description ($\boldsymbol{z}_t$) as input, the model predicts regression labels $\boldsymbol{y} \in \mathbb{R}^{(1 \times 209)}$, as displayed in Fig. 2d. The pretraining loss uses MSE and can be formulated as:

$$\mathcal{L} = \frac{1}{N} \sum_{j=1}^{N} \left(\frac{1}{\mathrm{count}(\boldsymbol{m})} \sum_{i=1}^{M} m_{ij} (y_{ij} - \hat{y}_{ij})^2 \right) \tag{8}$$

where N is the number of samples, $\boldsymbol{m} \in \mathbb{R}^{(1 \times 209)}$ is a 0-1 vector, in which the value 1 indicates the existence of the corresponding regression label, while 0 indicates its absence, $\mathrm{count}(\boldsymbol{m})$ means the number of valid labels, M is the number of all properties (here $M = 209$), y_{ij} is the ground truth regression label, and $\hat{y}_{ij}$ is the predicted value.

Downstream Application. When applying MolTailor to downstream tasks, we use GPT-4 to analyze the specific task and then generate the corresponding task description, as shown in Fig. 2e. We use the template in Tab. 1 to prompt GPT to generate descriptions that mimic the format of those in the pretraining corpus. Notably, we restrict it such that it can only select the names of properties from the set supported by RDKit. Then, we feed the generated analysis and SMILES into MolTailor to obtain the task-specific molecular representation:

$$\boldsymbol{z} = \mathrm{MolTailor}\left(\boldsymbol{z}_m | \boldsymbol{z}_t\right) \tag{9}$$

where z denotes the task-specific molecular representation. Finally, z goes through a prediction head to predict the labels corresponding to the task.

Experiments

In this section, we conduct comprehensive experiments to demonstrate the efficacy of our proposed method. The experiments are designed to analyze the method by addressing the following six key questions:

Q1: Are the task-specific molecular representations generated by MolTailor better than general representations? **Q2**: Can MolTailor achieve performance improvements on different M-Encoders? **Q3**: How do the task descriptions influence the effectiveness of pretraining? **Q4**: How do different text prompts affect MolTailor? **Q5**: When MolTailor achieves better performance, are the task-relevant properties in the molecular representations enhanced? **Q6**: Does MolTailor pay attention to the key information in both the molecules and text prompts?

Experimental Setup

Pretraining Corpus. We use MT-MTR corpus for pretraining, which contains 55,759 triples. Moreover, we present the data overlap analysis results in the appendix.

Downstream Datasets. We select 8 representative tasks from MoleculeNet (Wu et al. 2018) for experiments, which consist of 4 classification tasks (BBBP, ClinTox, HIV, Tox21) and 4 regression tasks (ESOL, FreeSolv, Lipophilicity, QM8), covering physiology, biophysics, physical chemistry, and quantum mechanics. Following Wu et al. (2018), each task uses the recommended splitting method to divide data into training/validation/test sets with a ratio of 8:1:1.

Baselines. We adopt the following four types of baselines:

- **Molecular Fingerprints**: MACCSFP (Durant et al. 2002) encodes molecules based on substructures, RDKitFP (O'Boyle and Sayle 2016) encodes molecules based on topology or path, and MorganFP (Rogers and Hahn 2010) encodes molecular environment and structure starting from atoms within a radius.
- **Sequence-based Methods**: BioLinkBERT(Yasunaga, Leskovec, and Liang 2022), PubMedBERT (Gu et al. 2021); ChemBERTa-2 (Ahmad et al. 2022), and CHEM-BERT (Kim et al. 2021).
- **Graph-based Methods**: Grover (Rong et al. 2020), MolCLR (Wang et al. 2022), Mole-BERT (Xia et al. 2022), and Uni-Mol (Zhou et al. 2023).
- **Multimodal Methods**: KCL (Fang et al. 2022), KV-PLM (Zeng et al. 2022), MolT5 (Edwards et al. 2022), and MoMu (Su et al. 2022).

We additionally construct Random and RDKit-DP as baselines for comparison, where Random refers to random vectors, and RDKit-DP consists of the 209 molecular properties calculated by RDKit.

Evaluation Methodology. To better evaluate the molecular representations learned by different models, we conduct experiments in linear probe setting. As a result, the baseline experimental results reported in this paper may differ from the original papers. We freeze the model parameters to extract representations for downstream tasks. These extracted representations are then mapped to labels through a learnable linear layer.

Following the recommendation of Wu et al. (2018), we use ROC-AUC as the evaluation metric for classification tasks. For the regression task qm8, we use MAE, and for other regression tasks, we use RMSE. To ensure fairness, we use Optuna (Akiba et al. 2019) to search 10 learning rates (LRs) for each model. We repeat each task 3 times and report the mean and standard deviation. Due to space limitations, the standard deviations are included in the appendix.

Implementation Details. For the pretraining phase, we employ the AdamW optimizer, complemented by a linear learning rate scheduler. We set the LR at 5.5e-5 and use a warmup ratio of 0.1. The training is conducted 50 epochs with a batch size of 64, utilizing two A100-SXM4-80GB GPUs. For the downstream tasks, we opt for the Adam optimizer and leverage Optuna for hyperparameter tuning, conducting 10 trials to identify the optimal LR within the range of 1e-5 to 1e-2 for each model on every task. The optimal LR is determined based on the performance of the validation set. We train our model using a batch size of 64 on a single GeForce RTX 2080 Ti GPU, employing an early stop mechanism with a patience setting of 3 and limiting the training to a maximum of 50 epochs.

Performance Comparison (Q1 & Q2)

Q1: We evaluate whether molecular representations enhanced by task descriptions could improve performance across 8 tasks. Tab. 2 shows that MolTailor achieves performance gains over the backbone model on the 4 regression tasks, and notably attains state-of-the-art (SOTA) results on ESOL, FreeSolv, and Lipophilicity datasets. On the 4 classification tasks, MolTailor's performance is inconsistent, with gains on HIV and Tox21 but losses on the other 2 datasets when using CHEM-BERT as backbone. We hypothesize that the difference in model performance on classification and regression tasks is due to the pretraining data benefiting the regression task more. Meanwhile, we experimented with converting the regression pretraining task into a classification form. The results were still similar, eliminating the influence of the form of the pretraining task on downstream tasks.

Q2: We experiment on two backbones, ChemBERTa-2 and CHEM-BERT. MolTailor achieves similar gains on top of both backbones. Notably, the performance differences between the backbones are also mirrored in the respective MolTailor variants. This demonstrates MolTailor's transferability and ability to inherit strengths of different backbones. The results support that further gains may be achievable by transferring MolTailor to new state-of-the-art single-modality models.

Models	**Classification (ROC-AUC)**				**Regression (RMSE / MAE)**				
Dataset	BBBP	ClinTox	HIV	Tox21	ESOL	FreeSolv	Lip	QM8	**Params**
#Molecules	2039	1478	41127	7831	1128	642	4200	21786	
#Split	Scaffold	Random	Scaffold	Random	Random	Random	Random	Random	
#Tasks	1	2	1	12	1	1	2	16	
Random	48.38	56.01	49.54	51.11	3.3358	5.4831	1.3813	0.0320	-
RDKit-DP	78.25	67.36	70.85	65.61	4.8940	2.8068	0.9963	0.0202	-
RDKit-FP	87.65	57.13	78.66	76.14	1.0830	2.0725	0.9007	**0.0181**	-
MACCS-FP	81.64	83.05	77.53	77.27	1.0833	1.9086	0.9886	0.0196	-
Morgan-FP	82.73	72.61	**82.65**	75.29	1.2413	2.1896	0.8196	0.0200	-
Grover	79.83	87.75	77.47	79.61	0.8977	1.9041	0.8301	0.0184	107M
MolCLR	81.27	78.15	71.48	75.61	1.3421	3.0436	1.0448	0.0219	2M
Mole-BERT	82.70	81.82	79.35	**84.20**	1.1379	2.3626	0.8316	0.0221	2M
Uni-Mol	79.52	88.65	74.18	78.08	1.0509	2.6913	1.0363	0.0219	48M
BioLinkBERT	83.81	87.75	71.24	73.81	1.1739	3.1350	1.0589	0.0234	108M
PubMedBERT	**89.10**	84.29	72.30	73.77	1.0715	2.5999	1.0851	0.0232	108M
ChemBERTa-2	84.70	84.21	78.88	80.75	0.8103	1.8439	0.7948	0.0191	3M
CHEM-BERT	84.10	**93.80**	76.99	80.54	0.7973	2.0214	0.8571	0.0215	51M
KCL	76.86	60.80	68.48	74.98	0.8728	2.7615	0.9868	0.0225	1M
KV-PLM	86.36	81.20	73.52	74.62	1.1785	2.8840	1.1004	0.0233	109M
MoMu-ME	80.41	67.99	71.91	74.75	1.4135	2.3229	0.9835	0.0222	2M
MoMu-TE	82.24	81.94	67.88	73.07	1.2562	3.1480	1.0885	0.0250	109M
MolTailor*	84.65	85.95	76.42	80.32	**0.7128**	**1.7826**	**0.7848**	0.0185	112M
MolTailor	81.15	92.37	77.42	80.67	0.7234	1.7881	0.8107	0.0196	161M

Table 2: Evaluation results of MolTailor and baselines under the linear probe setting on MoleculeNet. Here, ME stands for Molecule Encoder and TE stands for Text Encoder in MoMu-ME/TE. MolTailor* denotes using ChemBERTa-2 as the M-Encoder, while MolTailor indicates using CHEM-BERT. The table shows the average performance over 3 runs. Standard deviations are omitted due to space limitations but can be viewed in the appendix. Results in bold indicate state-of-the-art (SOTA) performance, while the underlined results show that the model outperforms the backbone used in the M-Encoder.

Ablation Study (Q3 & Q4 & Q5)

Q3. To evaluate the influence of task descriptions on pretraining in MT-MTR, we first remove the task descriptions to obtain the dataset MT-MTR*, then pretrain CHEM-BERT and PubMedBERT on MT-MTR* to get CHEM-BERT* and PubMedBERT* models. Next, we concatenate the task descriptions after the SMILES strings to construct dataset MT-MTR†, and pretrain PubMedBERT on MT-MTR† to obtain PubMedBERT†. As shown in Tab. 3, CHEM-BERT* improves over untrained CHEM-BERT on regression tasks but drops on classification, indicating MTR as a pretraining task benefits downstream regression but negatively impacts classification. Also, PubMedBERT† over PubMedBERT* and MolTailor over CHEM-BERT both demonstrate further regression performance gains but classification performance drops. This shows introducing task descriptions helps models better learn from the data. That is, if constructing new labels can improve model performance on classification tasks, then supplementing task descriptions can further amplify such gains.

Q4. We replace the task-specific descriptions generated via GPT-4 with the irrelevant text *"to be or not to be, this is the question."* as noise prompts. Results in Tab. 4 show degraded performance with noise prompts, indicating information in the task descriptions does help the model generate better molecular representations. However, it should be noted that the performance drop is slight, suggesting textual information is not as important as expected in the process of generating representations. Further investigation into this phenomenon is needed.

Models	**Classification↑**	**Regression↓**
CHEM-BERT	83.86	0.9243
CHEM-BERT*	83.11	0.9209
PubMedBERT	79.87	1.1949
PubMedBERT*	81.67	0.9148
PubMedBERT†	81.08	0.9131
MolTailor	82.09	0.8475

Table 3: Experimental results for Q3. CHEM-BERT and PubMedBERT use their original weights when evaluated on downstream tasks. The remaining methods are pretrained on MT-MTR and its variants, with overlapping molecules between pretraining and downstream data removed. Additionally, all methods are pretrained for only 20 epochs.

Q5. The experiments in Tab. 2 show that the molecule representations generated by MolTailor attained better performance. However, are the obtained representations more task-specific, meaning the task-relevant properties in the representations are enhanced? To validate this, we designed

Prompt Types	Classification↑	Regression↓
Origin	82.92	0.8348
Noise	82.77	0.8541

Table 4: Experimental results for Q4.

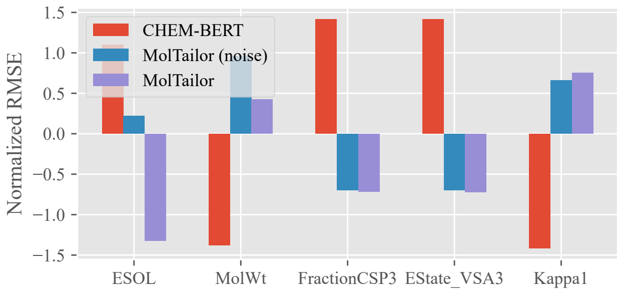

Figure 3: Performance of three methods on ESOL and molecular properties prediction tasks for Q5. The x-axis shows the task names, the y-axis shows the normalized RMSE, with lower values indicating better performance. Of the four molecular properties, the first three are related to the ESOL task, while the last one is opposite.

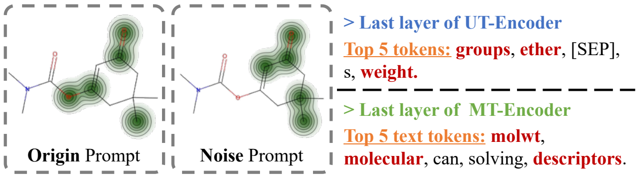
Origin Prompt
Noise Prompt
> Last layer of UT-Encoder
Top 5 tokens: groups, ether, [SEP], s, weight.
> Last layer of MT-Encoder
Top 5 text tokens: molwt, molecular, can, solving, descriptors.

Figure 4: Visualization of the attention, answering Q6. The two molecular graphs on the left show MolTailor's attention over the molecules under different prompts. The text on the right shows which input tokens from the original prompt MolTailor pays most attention to.

the following experiment: we used the representations generated by different methods to predict task-related and task-unrelated properties. If the expectations are met, then the representations enhanced by MolTailor should have better performance in predicting task-related molecular properties.

We conduct experiments on the ESOL dataset. For the models, we use MolTailor and its corresponding backbone CHEM-BERT. We also test the performance of MolTailor using irrelevant text as the prompt, denoted as MolTailor(noise). In detail, we first use RDKit to compute the properties of the molecules in the ESOL dataset, obtaining the ESOL-MTR dataset. Then, we randomly split the dataset into 8:1:1 for training, validation, and testing.

The results are shown in Fig. 3. We present the performance of the three methods on the ESOL task and in predicting four molecular properties. Among the four selected properties, two have names mentioned in the prompt and are relevant to the ESOL task (MolWt and FractionCSP3), one is not mentioned but is relevant (EState_VSA3), and one is neither mentioned nor relevant (Kappa1). We use RMSE as the evaluation metric. For better visualization, we normalize the RMSE under each task using the formula: Normalized RMSE $= \frac{\text{RMSE}-\text{mean}(\text{RMSE})}{\text{std}(\text{RMSE})}$.

The results show that: a) Compared to CHEM-BERT, MolTailor does focus more on the properties mentioned in the prompt, validating that the obtained representations are indeed task-specific. b) MolTailor not only pays more attention to the relevant properties explicitly stated in the prompt, but also to the unstated yet still relevant properties. Meanwhile, it decreases attention on the unrelated, unmentioned properties. This demonstrates the method's generalization capability. c) The declined performance of MolTailor(noise) in property prediction indicates that the prompt does help the model attend to critical information.

Case Study (Q6)

Q6. We analyze the attention matrices from the last layers of the UT/MT-Encoder to investigate whether MolTailor pays attention to key information. We select the ESOL dataset, which is related to solubility, to conduct experiments. If the model attends to information such as molecular weight or polar functional groups that are critical determinants of solubility, it suggests key information is captured.

In detail, We pass the SMILES from the test set through MolTailor trained on ESOL to obtain the required attention matrices $\boldsymbol{M}_{UT}$ and $\boldsymbol{M}_{MT}$. Then, we take the values of the matrix between the "[CLS]" token and the other tokens as the required attention weights.

The analysis results in Fig. 4 show that MolTailor does pay attention to solubility-related information, such as "groups", "ethers", and other highlighted tokens. Notably, the tokens here are merged to form complete words. Additionally, the model also pays attention, as seen in the left molecule attention graph, to polar functional groups within the SMILES. Furthermore, variations between the two molecule attention graphs under different prompts indicate effective prompts help the model better focus on key information that assist in solving the task

Conclusion and Future Work

Overall, in this work we propose a new perspective on molecular representation learning. Instead of trying to incorporate more information into the representations, we obtain more task-specific representations by combining contextual information. At the same time, we not only utilize the knowledge contained in the text modality, but also try to leverage the reasoning ability of language models, which has huge potential in the era of large language models.

In the future, we first plan to explore new pretraining tasks that can stably improve model performance on both classification and regression tasks. Secondly, we will explore how to build molecular-text multimodal models based on large language models. Finally, we look forward to in-depth collaborations with domain experts to apply the molecular-text multimodal methods to actual production problems.

Acknowledgements

We thank the reviewers for their detailed and constructive suggestions and for the affirmation of our paper. Meanwhile, we gratefully acknowledge the support of the National Key R&D Program of China (2021ZD0113302), the National Natural Science Foundation of China Youth Fund (62206079), and the Heilongjiang Provincial Natural Science Foundation of China (YQ2022F006). We also appreciate the support from Du Xiaoman (Beijing) Science Technology Co., Ltd. @ on our research.

References

Ahmad, W.; Simon, E.; Chithrananda, S.; Grand, G.; and Ramsundar, B. 2022. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*.

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Bran, A. M.; Cox, S.; Schilter, O.; Baldassari, C.; White, A.; and Schwaller, P. 2023. Augmenting large language models with chemistry tools. In *NeurIPS 2023 AI for Science Workshop*.

Chen, X.; Zhang, N.; Li, L.; Deng, S.; Tan, C.; Xu, C.; Huang, F.; Si, L.; and Chen, H. 2022. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 904–915.

Chen, Y.; Xi, N.; Du, Y.; Wang, H.; Jianyu, C.; Zhao, S.; and Qin, B. 2023. From Artificially Real to Real: Leveraging Pseudo Data from Large Language Models for Low-Resource Molecule Discovery. *arXiv preprint arXiv:2309.05203*.

Chithrananda, S.; Grand, G.; and Ramsundar, B. 2020. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*.

Christofidellis, D.; Giannone, G.; Born, J.; Winther, O.; Laino, T.; and Manica, M. 2023. Unifying molecular and textual representations via multi-task language modelling. *arXiv preprint arXiv:2301.12586*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Durant, J. L.; Leland, B. A.; Henry, D. R.; and Nourse, J. G. 2002. Reoptimization of MDL keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6): 1273–1280.

Edwards, C.; Lai, T.; Ros, K.; Honke, G.; Cho, K.; and Ji, H. 2022. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*.

Fang, Y.; Zhang, Q.; Yang, H.; Zhuang, X.; Deng, S.; Zhang, W.; Qin, M.; Chen, Z.; Fan, X.; and Chen, H. 2022. Molecular contrastive learning with chemical element knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3968–3976.

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

Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; and Poon, H. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23.

Guo, Z.; Sharma, P.; Martinez, A.; Du, L.; and Abraham, R. 2021. Multilingual molecular representation learning via contrastive pre-training. *arXiv preprint arXiv:2109.08830*.

Hastings, J.; Owen, G.; Dekker, A.; Ennis, M.; Kale, N.; Muthukrishnan, V.; Turner, S.; Swainston, N.; Mendes, P.; and Steinbeck, C. 2016. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic acids research*, 44(D1): D1214–D1219.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Kim, H.; Lee, J.; Ahn, S.; and Lee, J. R. 2021. A merged molecular representation learning for molecular properties prediction with a web-based service. *Scientific Reports*, 11(1): 11028.

Kim, W.; Son, B.; and Kim, I. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, 5583–5594. PMLR.

Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; and Chang, K.-W. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Liu, S.; Guo, H.; and Tang, J. 2022. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*.

Liu, S.; Nie, W.; Wang, C.; Lu, J.; Qiao, Z.; Liu, L.; Tang, J.; Xiao, C.; and Anandkumar, A. 2022. Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789*.

Liu, S.; Wang, H.; Liu, W.; Lasenby, J.; Guo, H.; and Tang, J. 2021. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*.

Liu, Z.; Zhang, W.; Xia, Y.; Wu, L.; Xie, S.; Qin, T.; Zhang, M.; and Liu, T.-Y. 2023. MolXPT: Wrapping Molecules with Text for Generative Pre-training. *arXiv preprint arXiv:2305.10688*.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Mamoshina, P.; Vieira, A.; Putin, E.; and Zhavoronkov, A. 2016. Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5): 1445–1454.

O'Boyle, N. M.; and Sayle, R. A. 2016. Comparing structural fingerprints using a literature-based similarity benchmark. *Journal of cheminformatics*, 8(1): 1–14.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rogers, D.; and Hahn, M. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5): 742–754.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; and Huang, J. 2020. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33: 12559–12571.

Ross, J.; Belgodere, B.; Chenthamarakshan, V.; Padhi, I.; Mroueh, Y.; and Das, P. 2022. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12): 1256–1264.

Seidl, P.; Vall, A.; Hochreiter, S.; and Klambauer, G. 2023. Enhancing activity prediction models in drug discovery with the ability to understand human language. *arXiv preprint arXiv:2303.03363*.

Song, Y.; Zheng, S.; Niu, Z.; Fu, Z.-H.; Lu, Y.; and Yang, Y. 2020. Communicative Representation Learning on Attributed Molecular Graphs. In *IJCAI*, volume 2020, 2831–2838.

Stärk, H.; Beaini, D.; Corso, G.; Tossou, P.; Dallago, C.; Günnemann, S.; and Liò, P. 2022. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, 20479–20502. PMLR.

Su, B.; Du, D.; Yang, Z.; Zhou, Y.; Li, J.; Rao, A.; Sun, H.; Lu, Z.; and Wen, J.-R. 2022. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, S.; Guo, Y.; Wang, Y.; Sun, H.; and Huang, J. 2019. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, 429–436.

Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; et al. 2023. Image as a Foreign Language: BEiT Pretraining for Vision and Vision-Language Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19175–19186.

Wang, Y.; Wang, J.; Cao, Z.; and Barati Farimani, A. 2022. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3): 279–287.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.

Wiener, H. 1947. Structural determination of paraffin boiling points. *Journal of the American chemical society*, 69(1): 17–20.

Wishart, D. S.; Feunang, Y. D.; Guo, A. C.; Lo, E. J.; Marcu, A.; Grant, J. R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. 2018. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research*, 46(D1): D1074–D1082.

Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; and Pande, V. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2): 513–530.

Xia, J.; Zhao, C.; Hu, B.; Gao, Z.; Tan, C.; Liu, Y.; Li, S.; and Li, S. Z. 2022. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*.

Xia, J.; Zhu, Y.; Du, Y.; Liu, Y.; and Li, S. 2023. A Systematic Survey of Chemical Pre-trained Models. IJCAI.

Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. 2019. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8): 3370–3388.

Yasunaga, M.; Leskovec, J.; and Liang, P. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823.

Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; and Wu, Y. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Zaidi, S.; Schaarschmidt, M.; Martens, J.; Kim, H.; Teh, Y. W.; Sanchez-Gonzalez, A.; Battaglia, P.; Pascanu, R.; and Godwin, J. 2022. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*.

Zeng, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2022. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1): 862.

Zeng, Z.; Yin, B.; Wang, S.; Liu, J.; Yang, C.; Yao, H.; Sun, X.; Sun, M.; Xie, G.; and Liu, Z. 2023. Interactive Molecular Discovery with Natural Language. *arXiv preprint arXiv:2306.11976*.

Zhao, H.; Liu, S.; Ma, C.; Xu, H.; Fu, J.; Deng, Z.-H.; Kong, L.; and Liu, Q. 2023. GIMLET: A Unified Graph-Text Model for Instruction-Based Molecule Zero-Shot Learning. *bioRxiv*, 2023–05.

Zhou, G.; Gao, Z.; Ding, Q.; Zheng, H.; Xu, H.; Wei, Z.; Zhang, L.; and Ke, G. 2023. Uni-Mol: a universal 3D molecular representation learning framework.