Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Text Mining

MolLM: A Unified Language Model to Integrate Biomedical Text with 2D and 3D Molecular Representations

Xiangru Tang [1], Andrew Tran [1], Jeffrey Tan [1] and Mark B. Gerstein [1,2,3,4,†]

[1]Department of Computer Science, Yale University, New Haven, CT 06520, USA
[2]Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06520, USA
[3]Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT 06520, USA
[4]Department of Statistics & Data Science, Yale University, New Haven, CT 06520, USA

[†]To whom correspondence should be addressed.

Abstract

Motivation: The present paradigm of deep learning models for molecular representation relies mostly on 1D or 2D formats, neglecting significant 3D structural information that offers valuable physical insight. This narrow focus inhibits the model's versatility and adaptability across a wide range of modalities. Conversely, the smaller amount of research that focuses on explicit 3D representation tends to overlook textual data within the biomedical domain.
Results: We present a unified pre-trained language model that concurrently captures biomedical text, 2D, and 3D molecular information. Our model, the three-modality molecular language model MolLM, consists of a text Transformer encoder and a molecular Transformer encoder, which encodes both 2D and 3D molecular structures. For MolLM, we construct 168K molecule-text pairings for training. We employ contrastive learning as a supervisory signal for cross-modal information learning. MolLM demonstrates robust molecular representation capabilities in numerous downstream tasks, including cross-modality molecule and text matching, property prediction, captioning, and text-prompted editing. Through ablation, we demonstrate that the inclusion of explicit 3D representations improves performance on downstream tasks.
Availability and implementation: Our code, data, and pre-trained model weights are all available at https://github.com/gersteinlab/MolLM.

1 Introduction

Pre-training techniques in Natural Language Processing (NLP) have caused significant shifts in numerous domains due to their ability to interpret complex patterns from large-scale datasets (Radford *et al.*, 2018; Vaswani *et al.*, 2017; Devlin *et al.*, 2019; Raffel *et al.*, 2020; Wei *et al.*, 2021). This success has led to promising outcomes in the biomedical field, especially in tasks like molecular property prediction and chemical structure generation, indicating a strong AI potential in biomolecular studies (Gu *et al.*, 2021; Lee *et al.*, 2019; Xia *et al.*, 2022; Luo *et al.*, 2022a; Singhal *et al.*, 2023). The capabilities of pre-training techniques allow a heightened understanding of biomedical data, thereby potentially enhancing the drug discovery processes, specifically for novel small chemical molecules (Zeng *et al.*, 2022; Liu *et al.*, 2022; Su *et al.*, 2022; Seidl *et al.*, 2023; Liu *et al.*, 2023b; Lewis *et al.*, 2020; Xue *et al.*, 2020; Yu *et al.*, 2021; Wang *et al.*, 2019; Chithrananda *et al.*, 2020).

Molecular structures encapsulate crucial chemical information, and the positions of functional groups within them serve as potent predictors of molecular properties and interactions. Current machine learning (ML) methodologies primarily focus on modeling singular modalities of molecular chemical structures, such as one-dimensional descriptors or two-dimensional molecular graphs. While there has been a small amount of research done on incorporating three-dimensional geometric structures, these works have their limitations (Yu *et al.*, 2021; Miao *et al.*, 2023; Jiang *et al.*, 2021; Devinyak *et al.*, 2014). Past endeavors have sought to learn representations of molecular chemical properties through supervised tasks with labeled data (Hwang *et al.*, 2020; Zuranski

© The Author 2015. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Model	Text	2D	3D	# Molecules	Source
MoMu (Su *et al.*, 2022)	SMILES + S2orc	✓	✗	15K	PubChem, S2orc
MoleculeSTM (Liu *et al.*, 2022)	PubChem Descriptions	✓	✓	281K	PubChem
KV-PLM (Zeng *et al.*, 2022)	SMILES + S2orc	✓	✗	10K	PubChem, S2orc
MolLM	SMILES + PubChem Descriptions + S2orc	✓	✓	160K	PubChem, S2orc

Table 1. Comparison of datasets used in our study (MolLM) and in the baseline approaches. Each dataset is characterized by the source of text, molecular formula representation, availability of 2D / 3D structural information, total molecule count, main application areas, and the original data source.

et al., 2021). Nevertheless, these efforts hinge heavily on the availability of large-scale datasets, prompting an increased focus on self-supervised learning. Recent investigations have leveraged pre-trained, self-supervised methods to derive representations of molecular structures (Zeng *et al.*, 2022; Rong *et al.*, 2020; Zhang *et al.*, 2021), primarily through a masking approach to learning the geometric structure, consequently obtaining the corresponding chemical structure representations. However, these endeavors have predominantly concentrated on two-dimensional molecular structures (Su *et al.*, 2022; Pogány *et al.*, 2018; Coley *et al.*, 2017; Liu *et al.*, 2023a), thereby neglecting the pivotal role of three-dimensional structures in molecular modeling. In practical terms, three-dimensional structures offer a representation of the spatial organization of atoms and functional groups, proving indispensable for understanding molecular interactions and properties.

Additionally, the natural language modality encompasses a vast volume of biochemical knowledge that should aid model comprehension of molecular structures. When generalizing to downstream tasks, if models can comprehend natural language, it would enable better-conditioned control of molecule generation and editing, using natural language as the mode of interaction. Even in state-of-the-art multimodal models for these tasks, they still do not incorporate valuable 3D information alongside natural language (Su *et al.*, 2022). Additionally, these multimodal models have been limited by reliance on small-scale molecular graph-text data In contrast to existing models that primarily focus on either one-dimensional descriptions (SMILES) or two-dimensional molecular graphs, human expertise is honed by engaging with both molecular structures and biomedical text. Hence, a pressing requirement exists for the development of models that can seamlessly integrate both two-dimensional and three-dimensional molecular structures with biomedical text to propel novel scientific discovery. Inspired by this human proficiency, we introduce the molecular multimodal language model (MolLM) that binds two-dimensional and three-dimensional molecular structures with natural language, a tri-modal approach not explored in the current research space.

In this work, we first create a multimodal dataset, from molecules in PubChem (Kim *et al.*, 2023) and related publicly available academic texts, with 160k molecule graph-text pairs to incorporate natural language. At the core of MolLM is a unified pre-training framework, integrating a text Transformer encoder and a molecular Transformer encoder, pre-trained on molecular graphs and related textual data jointly. The text Transformer encoder is pre-trained on biomedical text. To accommodate the multimodal molecular structure information, we then adopt a molecular Transformer encoder. Using the graph Transformer architecture, we incorporate pair-wise encodings for 2D graphs and 3D geometric structures. We integrate positional encoding with the standard graph encoding, allowing the model to include atom-wise structural information. Similar to Transformer-M (Luo *et al.*, 2022b), we effectively leverage both 2D and 3D molecular structures for better performance in downstream tasks. Crucially, our model aligns these cross-modal representations during pre-training through the paradigm of contrastive learning, a technique that allows for efficient self-supervised learning by associating similar molecules closer together and dissimilar molecules away in the representation space.

We further emphasize the significance of capturing 3D molecular structures. Our model MolLM overcomes the limitations of existing pre-trained molecular models, which can only process different data types separately, by offering a comprehensive and in-depth understanding of molecular entities. We conduct extensive experiments to showcase the model's capacity to capture molecular expertise, demonstrating promising performance in downstream tasks such as **cross-modal retrieval** (retrieving the related molecule/text with the query of the opposite modality), **molecule captioning** (providing natural language description of molecules), **property prediction** (classification task of predicting molecule properties), and **molecule editing** (editing molecule based on natural language prompt) (Zeng *et al.*, 2022; Edwards *et al.*, 2022; Wu *et al.*, 2017a; Liu *et al.*, 2022). We emphasize how our promising results demonstrate that 3D information is beneficial to be integrated. For example, in the cross-modality matching task, while our multimodal architecture outperforms all baselines in a variety of zero-shot and supervised settings, the direct incorporation of 3D data further improves each score. Ultimately, our main contribution is in two areas. First, we show that multimodal pre-trained models are effective in molecular analysis, especially when all three modalities of text, 2D, and 3D are incorporated. Second, we create a novel dataset, much larger than baselines with comparable text complexity. We demonstrate that datasets require both in-depth textual information and a large size to maximize effectiveness.

2 Related work

2.1 Molecular Representations

Historically, many works have focused on 1D molecular representations. The most common method for 1D representation is the Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988), which many papers have leveraged for a variety of molecular tasks (Chen and Zhang, 2021; Kuenzi *et al.*, 2020; An *et al.*, 2022). While simple and robust, SMILES strings are non-unique and have no standard generation method, which causes generative models to sometimes create invalid molecules. IUPAC International Chemical Identifier (InChI) strings are similarly accessible to SMILES in online chemical databases but are guaranteed to be unique. Thus, there has been an exploration of using InChI on tasks with a data integrity emphasis (An *et al.*, 2022; Handsel *et al.*, 2021). Nonetheless, InChI representations still suffer from having less domain support and higher computational complexity than SMILES. Self-Referencing Embedded Strings (SELFIES) (Krenn *et al.*, 2019) is another molecular string representation explicitly created for molecular analysis and deep learning. Though not as ubiquitous as SMILES nor InChI, recent works have demonstrated superior results with SELFIES (Yüksel *et al.*, 2023; Nigam *et al.*, 2021; Kaneko, 2023). However, more domain studies and supporting resources may be necessary before SELFIES becomes viable as a baseline for broader research. While SMILES strings and similar string-based representations are only 1D representations of molecules, 2D structures can still be inferred from them. Methods such

as convolutional neural networks (CNNs), which are typically used on 2D data to project onto a higher dimensional space, can also be applied to 1D strings representing molecules to infer 2D features (Hirohara *et al.*, 2018).

For more explicit representation of molecules, representing molecules as graphs is a popular method of integrating 2D structures. The most common and straightforward graph representation has vertices representing atoms and their respective features and edges representing bonds and their respective features. For an example of what such atom and bond features could be, we examine Open Graph Benchmark (OGB)'s molecule utility library, which is used by many (Hu *et al.*, 2020) state-of-the-art works. Under this framework, a molecule could have atom features such as atomic number, valence electrons, and formal charge, with bond features such as bond type and bond stereo. Utilizing this framework, various works have applied common graph model architectures to perform molecule analysis, such as CNNs, Graph Convolutional Networks (GCNs), and GraphSAGE (Pogány *et al.*, 2018; Coley *et al.*, 2017; Liu *et al.*, 2023a). These models achieve good results, however, they fail to account for nuances within the 3D molecular structure, such as stereochemistry, interaction energies, and spatial binding affinities. More recent works include 3D molecular graph data by providing 3D coordinates of each molecule within the atom representation. Since 3D data is not universally provided for many molecules, most works generate 3D conformations using molecular analysis libraries such as RDKit (Landrum, 2023). By focusing on 3D contexts, these models are able to achieve better results on certain tasks where 3D structure matters significantly, such as property prediction (Kuzminykh *et al.*, 2018; Thomas *et al.*, 2018; Kajita *et al.*, 2017; Stärk *et al.*, 2022; Li *et al.*, 2022b; Yang *et al.*, 2019). Certain models use contrastive learning techniques to accept multiple modalities of molecule inputs, including to unify 2D and 3D structural information (Liu *et al.*, 2021; Zhu *et al.*, 2022; Li *et al.*, 2022a).

2.2 Molecular Pre-trained Language Models

With the popularity of pre-trained language models, prior work has been done on utilizing pre-trained language models on molecules represented via textual representations such as SMILES. SMILES-BERT is one of the first models to explore inputting SMILES strings into language models (Wang *et al.*, 2019), training with masking tasks where the model predicts missing tokens in sequences that represent text, with SMILES strings included alongside regular text. BARTSmiles does similar with a BART-like model and more downstream tasks (Chilingaryan *et al.*, 2022). However, as these models use inherently 1D textual representations of molecules, they lose a significant amount of information about geometric and stereoelectric effects.

In terms of multimodality works, there is prior research on multimodal text and molecule models to represent molecules and related textual descriptions in a joint latent space. KV-PLM (Zeng *et al.*, 2022) utilizes the SMILES representation of molecules before projecting to such a joint latent space while MoMu (Su *et al.*, 2022) utilizes a graph representation of molecules before projection. They, alongside other recent works (Wang *et al.*, 2022a,b), utilize a contrastive loss function to align molecular representations and textual representations. To improve accuracy and robustness during training, MoMu employs augmentations of the molecular graphs to create semantically similar graphs (Liu *et al.*, 2022). However, with both MoMu and KV-PLM's molecular representation inputs, these models only consider a limited 2D view of molecular geometry.

In terms of training data, KV-PLM and MoMu each collected their datasets of 15k molecule-text pairs. KV-PLM sourced the textual data from PubChem while MoMu sourced a different set of 15k pairs from the S2ORC (Lo *et al.*, 2020) corpus dataset. As S2ORC is a dataset without clear pairing between molecules and related text, Su *et al.* expressed concerns with how naively selecting texts simply referencing molecules does not guarantee strongly related text. Therefore, limitations in the datasets of prior works extend beyond size concerns as these datasets may also have weak correlations within their molecule and text pairings.

2.3 Models for Specific Molecular Tasks

Molecular Property Prediction is one of the most widely researched molecular tasks, as property prediction may expedite traditionally expensive and time-consuming wet-lab experiments. Previous works like (Wang *et al.*, 2022b; Liu *et al.*, 2023c; Wen *et al.*, 2022; Ross *et al.*, 2022) aim to create models that perform well on this prediction task. MolCLR (Wang *et al.*, 2022b) is one of these models, and was trained on over 10 million unique molecules. MolCLR performs contrastive learning via graph neural networks and data augmentation. However, while MolCLR's large dataset allows for effective pretraining, MolCLR's dataset is not annotated with any text descriptions, thus removing MolCLR's ability to perform any molecular tasks involving text.

In terms of other models that have the downstream capability of property prediction, the representation learning method described in (Zhu *et al.*, 2022) combines 2D and 3D graph data. This method encodes atomic coordinates and interatomic distances, then merges them with their atom representations through graph neural networks. The method is then tested on property prediction downstream tasks, showing a significant improvement over 2D-only methods. This work shows us that 3D representations are important, and proposes a way to incorporate 3D information into the pretraining process. However, without textual data as an input into this model with molecules as its only modality, the method lacks flexibility for more generalized tasks, many of which involve text data.

Although molecular property prediction does not necessitate the use of text, other works have proposed molecular tasks that demand a combined understanding of both molecules and text. MolT5 introduced the molecule captioning task which involves generating meaningful textual descriptions of molecules that capture their most important properties, structural elements, and applications (Edwards *et al.*, 2022). Furthermore, MoleculeSTM introduced a molecule editing task that involves an input of a molecule and a prompt such as "This molecule is more soluble." with a desired task output of a molecule similar to the original that aligns with the prompt (Liu *et al.*, 2022).

3 Methods

3.1 Data

We collect our dataset from two sources. First, for collecting the molecular data and directly related descriptions, we access the PubChem molecular database (Kim *et al.*, 2023) which hosts information for over 100 million molecular compounds. Of these, we used the PubChem Classification Browser to identify 160k molecules with substantial physical descriptions, with emphasis on the comprehensiveness of the "Description" field. For example, we may get the molecule "Acetylcarnitine" from the PubChem Classification Browser. Then, using the PubChem REST API, we query for 2D graphical data, as well as the basic physical descriptions available from PubChem. Then, we compute 3D atomic positions to add to the graphs via RDKit's MMFF94 implementation. Second, for collecting additional textual data, we follow the methodology of MoMu to extract excerpts from papers in the S2ORC corpus database that relate to a certain molecule. We then take these text descriptions and pair them with the 2D and 3D graphical representations of each molecule to create a molecular graph-text pairs. Using the same aforementioned example, we then look into the S2ORC database to find papers related to "Acetylcarnitine", then extract text containing the molecule, or its synonyms, to add to our textual data

from the Abstract, Introduction, and Conclusion sections of each paper. Our dataset is the largest dataset in the domain currently that includes academic text, as well as full multimodal molecule information. For example, MoleculeSTM (Liu *et al.*, 2022) has a large dataset with 2D and 3D data, but does not incorporate academic text data, thus overlooking a significant data source. Refer to Table 1 for a comparison of our dataset to similar datasets.

3.1.1 Text Encoder

For the text encoder, we initialize with KV-PLM (Zeng *et al.*, 2022). KV-PLM enhances BERT (Bidirectional Encoder Representations from Transformers) (Devlin *et al.*, 2019), a widely used general text encoder, by fine-tuning its performance on academic texts related to molecules that include SMILES strings (Weininger, 1988). KV-PLM's model weights are initialized with those of SciBERT as well, a powerful pre-trained model that has been tailored specifically for scientific text. SciBERT differs from the general-purpose BERT model in that it is pre-trained on a large corpus of scientific text. This specialization allows SciBERT to better understand and generate representations for scientific language, making it highly valuable in the scientific and biomedical fields.

In our methodology, we employ KV-PLM's fine-tuned BERT checkpoint to initialize our text encoder, aiming to generate meaningful embeddings, emphasizing scientific context, for the extracted sentences from academic literature related to the selected molecules in our dataset. We also double the tokenizer output length limit within KV-PLM's and MoMu's use of the BERT tokenizer from 128 tokens to 256 tokens to avoid unnecessary truncating and loss of detail in processing text sequences, allowing for the inclusion of more detailed text descriptions. Thus, our approach enables capturing the core meaning and context of the full sentences and paragraphs efficiently.

3.1.2 Molecular Structure Encoder

MolLM is designed to obtain representations of molecular data via pathways that process both 2D and 3D structural information. The 2D pathway harnesses information extracted from the 2D graph structure of the molecule, including degrees, shortest path distances, and edges. These data elements represent the spatial relationships between atoms in the molecule, enabling the model to understand its two-dimensional architecture. Conversely, the 3D pathway focuses on the three-dimensional geometrical structure of the molecule. Here, we compute spatial distances between atoms within the 3D structure. This geometric perspective allows the model to gain insights into the three-dimensional arrangement of atoms and their complex interactions, which often play a pivotal role in molecular properties and interactions. Together, these three pathways allow the MolLM to generate a comprehensive understanding of the molecule, spanning its sequence and both its two-dimensional and three-dimensional structures, thereby producing a rich, multimodal representation.

Inspired by Transformer-M (Luo *et al.*, 2022b), MolLM employs a similar strategy to process both 2D graph structures and 3D geometric structures in molecular representations by a unified model. Using a graph Transformer as its base, it encodes structural data such as edge features, such as bond type, and 3D spatial relation as bias values within the attention mechanism, which the Transformer uses to weigh and combine different parts of the input. We utilize this modification to the attention mechanism as it allows the relationship between atoms in the model to be directly influenced by their 2D and 3D spatial relations, aligning with the physical significance of these geometric relations.

For a summary of the implementation, we recap the encodings of these structural features that we use from Transformer-M. In terms of the graph structure, the graph has vertices that represent atoms and edges that represent the bonds between these atoms.

For edge features encoding, we utilize the 2D graph structure by considering edge feature row vectors, $\mathbf{e}_1, \mathbf{e}_2, \ldots$, along the shortest path between any two vertices. These edge feature vectors represent features of the chemical bonds, including bond type and stereochemistry. Suppose the edge feature vectors are of dimension k_{edge}, as in $e_i \in \mathbb{R}^{k_{edge}}$ for each edge. Then, denote the element-wise average of them as $avg(\mathbf{e})$. We also have a learnable vector $\mathbf{w} \in \mathbb{R}^{k_{edge}}$ to emphasize the importance of certain bond features over others. Our edge features encoding for any two vertices indexed i and j, which is a scalar quantity, is given as

$$\phi_{ij}^{\text{Edge}} = avg(\mathbf{e}) \cdot \mathbf{w}. \tag{1}$$

Additionally, we want to include 2D distance, specifically the shortest path between atoms via following their bonds, between any two given atoms in our structural encoding. Thus, for the relation between atoms in the 2D, we define an encoding to represent the shortest path distance between two atoms. And we have a learnable scalar w to allow weighting the importance of considering shortest path distances in our structural encoding. Define $spd(i, j)$ to be the shortest path distance between atoms i and j in the 2D graph counted by edges, so it is a scalar quantity. So, our encoding for shortest path distances, a scalar quantity as well, is

$$\phi_{ij}^{\text{SPD}} = spd(i, j) \cdot w. \tag{2}$$

Similarly, we consider 3D spatial relations. Instead of utilizing standard Euclidean distance, we utilize the Gaussian Basis Kernel as it provides a non-linear function of distance which can give a more nuanced representation of the 3D spatial relationship. Suppose that we utilize K Gaussian Basis Kernels to represent 3D distance with the vector $\psi(\mathbf{i}, \mathbf{j}) = [\psi(i, j, 1), \ldots, \psi(i, j, K)] \in \mathbb{R}^K$. We also have learnable matrix $\mathbf{W}_1 \in \mathbb{R}^{K \times K}$ and vector $\mathbf{w}_2 \in \mathbb{R}^K$ for a multi-layer perceptron (MLP) to obtain the final encoding by processing the distance representation given by the Gaussian Basis Kernel. The MLP allows for a richer representation with higher-level features that can be inferred from the 3D distance along with the ability to selectively emphasize certain aspects, such as certain directions, within the 3D distance. We utilize ReLU as the activation function, act, within this MLP. The 3D distance encoding is processed between vertices, indexed i and j, as a scalar quantity, after applying the Gaussian Basis Kernels as

$$\phi_{ij}^{\text{3D Distance}} = act(\psi(i, j) \times \mathbf{W}_1) \cdot \mathbf{w}_2. \tag{3}$$

Specifically, the Gaussian Basis Kernel has a hyperparameter σ that controls the width of the kernel, which adjusts the sensitivity of the kernel to distance and the smoothness of its output. These aspects of the kernel align with the scales of distances at the molecular level and the potential noise of these distances respectively. The k-th Gaussian Basis Kernel (Scholkopf *et al.*, 1997) within $\psi(i, j)$ is given as

$$\psi(i, j, k) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^k}{\sigma^k}\right). \tag{4}$$

These 2D and 3D positional encodings from the graph structure of the molecule and the 3D geometric structures are combined linearly to consider all of them in aggregate. We incorporate this combined encoding into the attention mechanism of the graph Transformer layers. This linear combination, considering all atoms in the molecular graph, is given as

$$\phi_{ij}^{\text{2D/3D Position}} = \phi_{ij}^{\text{Edge}} + \phi_{ij}^{\text{SPD}} + \phi_{ij}^{\text{3D Distance}}. \tag{5}$$

Let a be the count of atoms in the molecular graph. Define the matrix $\phi^{\text{2D/3D Position}} \in \mathbb{R}^{a \times a}$ by Equation 5. We add this matrix as a bias directly to the standard Transformer attention calculation before the final application of Softmax. Recall that the motivation behind adding this to the

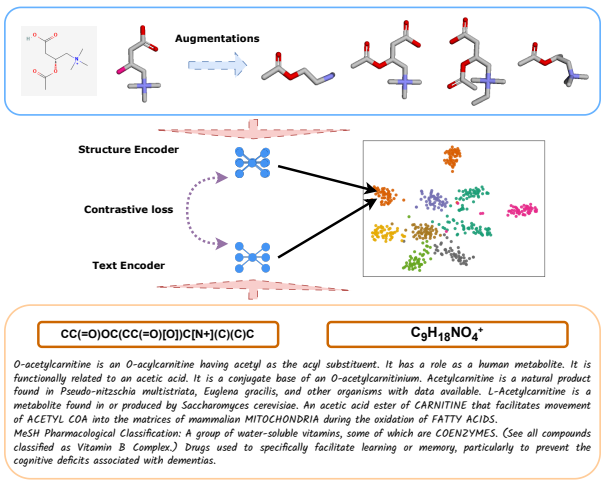

Fig. 1. Overview of the pretraining pipeline. MolLM encodes both molecular graph data (top) and textual data (bottom). For each molecule, we generate up to 4 molecule augmentations before encoding the 2D and 3D structural data. On the text side, we encode each molecule's SMILES string and molecular formula, as well as related passages from scientific journals. These encoded representations are then used during the contrastive learning step.

Transformer attention mechanism is that the purpose of this mechanism is to capture contextural relationships between our graph's vertices, and atoms, while also weighing the importance of each in the final embedding. By incorporating our bias which is based upon 2D and 3D spatial data, the model directly considers this rich spatial data in the attention calculation, thus considering spatial relationships between atoms within the molecular structure.

3.2 Contrastive Learning

Contrastive learning aims to learn meaningful representations by comparing positive and negative sample pairs. We add two losses: (1) *a contrastive loss* comparing the representations of different modalities of the same data sample, and (2) *a self loss* comparing different augmentations of the same modality for the same data sample.

3.2.1 Data Augmentation

By using data augmentation, we can increase the size of our training data, while also taking advantage of MolLM's contrastive learning framework. We expand upon the data augmentations proposed in MoMu (Su *et al.*, 2022) by adding 2 augmentations that alter each molecule's chemical features, such as molecular motifs and functional groups, on top of the 2 augmentations used in MoMu, which alter graph features like nodes and edges. In total, we apply four data augmentations to each 2D molecular graph: node dropping, subgraph sampling, chemical transformation, and substructure removal. We then compute and insert 3D atomic positions into these augmented graphs, aiming to create semantically consistent molecular structures suitable for pretraining. Node dropping randomly discards vertices from the original graph, while subgraph sampling is performed by traversing random walks. Chemical transformation-based augmentations generate new structures by applying various chemical transformations alongside other graph augmentations. They are implemented by adding and removing functional groups and subsequently optimizing 3D structures. Substructure removal uses BRICS decomposition to remove certain substructures in a molecular viable manner. A random viable molecular motif is then chosen out of the possible substructure generations. A visualization of these augmentations can be seen in Fig 1. Refer to Appendix 3 for more examples of augmentations.

For the aforementioned augmentations, we perform manipulations and optimizations on molecular structures using the open-source RDKit library. We repeat augmentations, which purposely have randomization in their augmentation process until we obtain an augmented molecule that measures a difference of below 3.0 root-mean-square deviation (RMSD) on RDKit's molecular distance, or inverse similarity, calculation to avoid augmenting molecules to be too semantically different from the original. See Appendix 2 for more motivation behind our use of RDKit. In terms of implementation details, we utilize the PyTorch Geometric library to represent our 3D molecular graphs, including all atomic features including 3D coordinates and bond features. Prior to pretraining, we employed this augmentation method efficiently on many CPU-optimized machines as CPU computing is limited on our GPU-optimized machines used for model pretraining.

Given a mini-batch of N molecular graphs, we generate four different augmentations for each graph. Let $\boldsymbol{z}_i^{G_1}$, $\boldsymbol{z}_i^{G_2}$, $\boldsymbol{z}_i^{G_3}$, and $\boldsymbol{z}_i^{G_4}$ each denote the representations of the augmented versios of the i-th graph $\boldsymbol{\mathcal{G}}_i$. Let $\boldsymbol{z}_i^{T_1}$, $\boldsymbol{z}_i^{T_2}$, and $\boldsymbol{z}_i^{T_3}$ denote the representations of three different sentences describing the i-th graph $\boldsymbol{\mathcal{G}}_i$. The objective of contrastive learning is to minimize the contrastive loss for each graph in the mini-batch, considering both cross-modal and self-contrastive losses.

3.2.2 Contrastive Learning Objective

The purpose of the cross-modality contrastive loss is to align representations of molecules and text in the latent space, while the self loss is to align augmented representations within the same modality. We define the cross-modality contrastive loss $\mathcal{L}_{\text{cross}}$ and self contrastive loss $\mathcal{L}_{\text{self}}$ for the i-th graph with a mini-batch of N molecular graphs and three different augmentations for each graph as follows. The final contrastive learning objective combines cross-modal and self-contrastive losses: $\mathcal{L}_{\text{MolLM}} = \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{self}}$, where $\ell_i^{(\cdot)}$ calculates the contrastive loss for the i-th graph $\boldsymbol{\mathcal{G}}_i$ considering the respective pairs of multimodal representations.

For an explanation of contrastive loss, $\mathcal{L}_{\text{cross}}$, recall that for the i-th graph of N molecular graphs, we have $\boldsymbol{z}_i^{G_1}$, $\boldsymbol{z}_i^{G_2}$, $\boldsymbol{z}_i^{G_3}$, and $\boldsymbol{z}_i^{G_4}$ as four representations of augmented graphs and also $\boldsymbol{z}_i^{T_1}$, $\boldsymbol{z}_i^{T_2}$, and $\boldsymbol{z}_i^{T_3}$ for related text. Equation 6 describes the contrastive loss as a measure of similarity between the graph representations ($\boldsymbol{z}_i^{G_a}$) and the corresponding text representations ($\boldsymbol{z}_i^{T}$) for a given molecular graph. This loss function is formulated as a negative logarithm of the ratio of the exponential similarity between the matching graph and text representations (for a specific graph-text pair) to the sum of exponential similarities across all possible graph-text pairs within the batch of N molecular graphs. The term $sim(\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_i^{T})$ denotes the cosine similarity function between a graph representation and a text representation, and τ is a temperature parameter that scales the similarity scores. The loss encourages maximizing the numerator which is the similarity between like matching pairs and minimizing the denominator which is the similarity between non-matching pairs. The overall effect of this is to encourage the model to produce graph and text representations that are closer in the embedding space for matching pairs, while pushing non-matching pairs apart.

$$\ell_{cross_i} = -\log \frac{\exp\left(sim(\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_i^{T})/\tau\right)}{\sum_{j=1}^{N} \exp\left(sim(\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_j^{T})/\tau\right)}, \tag{6}$$

Similarly, for the self loss $\mathcal{L}_{\text{self}}$, recall the same notation for the augmented graphs and related text. This loss function in Equation 7 is defined as the negative logarithm between the ratio of the sum of exponential similarities between all pairs of augmented graph representations ($\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_i^{G_b}$) to the sum of all exponential similarities of

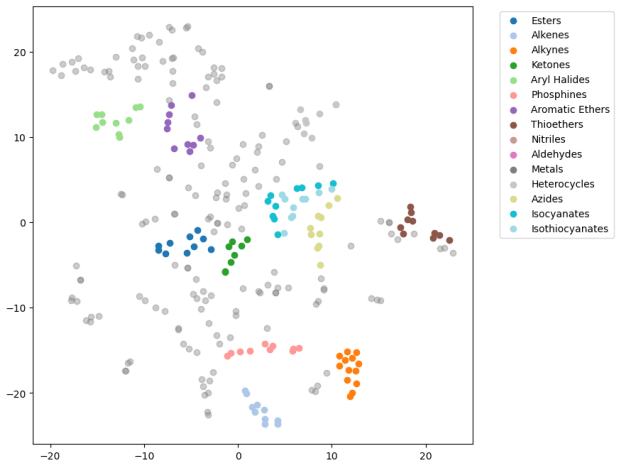

Fig. 2. t-SNE visualization of molecule embeddings in MolLM's joint latent space for select groups of molecules. Colored points represent molecules belonging to the select groups while the gray points represent a variety of other molecules.

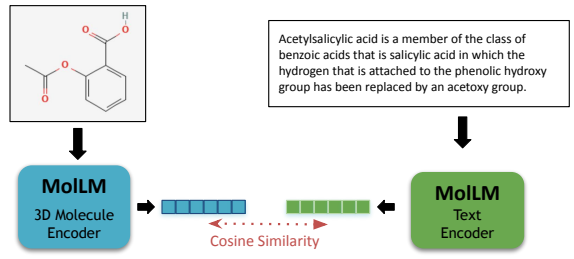

Fig. 3. Visualization of the cross-modality matching text for a given pair of molecule and text. Embeddings are generated for each molecule and text using the respective encoders of MolLM. Then, cosine similarity is used to find the most similar pair for top-1 matching, or the 20 most similar for R@20, for the matching task.

all such pairs in the graph. The effect of this loss is to enforce that different augmented representations of the same molecular graph are similar to each other in the embedding space. The τ is also a temperature parameter that scales the similarity scores. This loss encourages the model to push different augmented views of the same graph closer together, while pushing those of different graphs further apart.

$$\ell_{self_i} = -\log \frac{\sum\limits_{G_a,G_b \in \binom{\{G_1,G_2,G_3,G_4\}}{2}} \exp\left(sim(\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_i^{G_b})/\tau\right)}{\sum\limits_{j=1}^{N} \sum\limits_{G_a,G_b \in \binom{\{G_1,G_2,G_3,G_4\}}{2}} \exp\left(sim(\boldsymbol{z}_i^{G_a}, \boldsymbol{z}_j^{G_b})/\tau\right)}, \tag{7}$$

* Here, the notation $\binom{\{G_1,G_2,G_3,G_4\}}{2}$ represents the selection of any combination of two different graphs from the set of four.

$$\mathcal{L}_{\text{cross}} = \frac{1}{N}\sum_{i=1}^{N} \ell_{cross_i}^{(\boldsymbol{z}_i^{G_1},\boldsymbol{z}_i^{G_2},\boldsymbol{z}_i^{G_3},\boldsymbol{z}_i^{G_4},\boldsymbol{z}_i^{T_1},\boldsymbol{z}_i^{T_2},\boldsymbol{z}_i^{T_3})} \tag{8}$$

$$\mathcal{L}_{\text{self}} = \frac{1}{N}\sum_{i=1}^{N} \ell_{self_i}^{(\boldsymbol{z}_i^{G_1},\boldsymbol{z}_i^{G_2},\boldsymbol{z}_i^{G_3},\boldsymbol{z}_i^{G_4})} \tag{9}$$

In brief, the contrastive learning objective consists of two components: cross-modal contrastive loss and self-contrastive loss. The cross-modality contrastive loss minimizes the distance between the different modalities (i.e., molecular graphs and related textual descriptions) of the same molecule while maximizing the distance between different molecules. The self-contrastive loss minimizes the distance between different augmentations of the same molecule while maximizing the distance between augmentations of different molecules. Thus, the model learns to generate more robust and semantically meaningful representations for both molecules and text in a joint latent space.

3.3 Fine-tuning Downstream Tasks

To showcase the adaptability and practicality of our model across a wide variety of downstream tasks that necessitate both molecular and textual inputs, we fine-tune our model on four such tasks. To provide succinct overviews of these tasks, Cross-modality Matching involves retrieving the correct molecular graph/text from a list of data in the opposite modality. Property Prediction is a classification task for various experimental molecular properties, many of which are related to biological activity such as Blood-Brain Barrier Penetration (BBBP). Molecule Captioning involves generating a useful text description of a molecule. Molecule Editing involves generating a new molecule given an original molecule and a prompt with a desired property in text description form. See Table 2 for a summary of these tasks.

3.3.1 Cross-modality Matching

To implement the Cross-modality Matching task using our model, we begin by using the two encoders in our model to produce embeddings for all of the text and molecules respectively. There is a "key" text or molecule that is used to query for a match in the opposite modality. We generate embeddings for the "key" text or molecule being used for matching, as well as for the set of texts or molecules that it will be queried against. We then compare the "key" text or molecule to the set of the opposite modality by selecting those with maximal cosine similarity between their embeddings to find the best match (or the best matches for the recall metric). See Figure 3 for a visualization of how the model considers a specific molecule and text pair by computing cosine similarity for the task. For fine-tuning on this task, similar to (Su *et al.*, 2022), we split the data into a training set of $10,500$ molecules, a validation set of $1,500$ molecules, and a test set of $3,000$ molecules while generating their 2D graphs.

3.3.2 Property Prediction

In contrast, for the Property Prediction task, which is a classification task, using our model, we employ only the 2D/3D molecular encoder of our model to produce an embedding for the molecule. Then, we attach to our molecular encoder (which provides a 768-dimensional molecule embedding) a graph prediction head, described in Appendix 7.1.

3.3.3 Molecule Captioning

For the implementation of the Molecule Captioning task, we utilize the 2D/3D molecular encoder of our model to produce an embedding for the molecules, then forward this embedding through the MolT5l model's decoder (Edwards *et al.*, 2022). We utilize this decoder to generate the descriptions as our model does not have a decoder for text. Refer to Figure 6 for a visualization of this technique for generating captions.

3.3.4 Molecule Editing

Finally, for the Molecule Editing task, we have an input molecule and a text prompt. The desired output is a generated molecule that retains similarity to the input molecule while also having properties desired in the text prompt. For the generation, as MolLM can not directly reverse its embeddings into molecular graphs without a decoder, we utilize MoFlow (Zang and Wang, 2020), an invertible flow-based molecular model. We

Task	Definition	Dataset	Modalities
Cross-modality Matching	Given a key molecule or text, find the associated item of the opposite modality.	PCdes	Text, 2D, 3D
Property Prediction	Given a molecule, predict properties with classification tasks.	MoleculeNet	2D, 3D
Molecule Caption	Given a molecule, generate a descriptive text.	MolT5	Text, 2D, 3D
Molecule Editing	Given a molecule and a text prompt, generate an edited molecule accordingly.	MoleculeSTM	Text, 2D, 3D

Table 2. Description of downstream tasks tested along with their associated dataset and modalities.

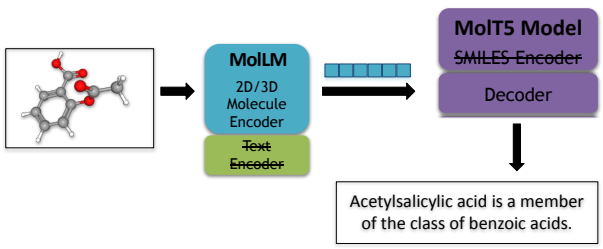

Fig. 4. Visualization of the molecule captioning task. We begin with a 2D/3D molecular graph of Acetylsalicylic Acid, which is then encoded through MolLM's molecule encoder. This encoded representation is then decoded through MolT5's decoder, which produces a molecule-specific caption.

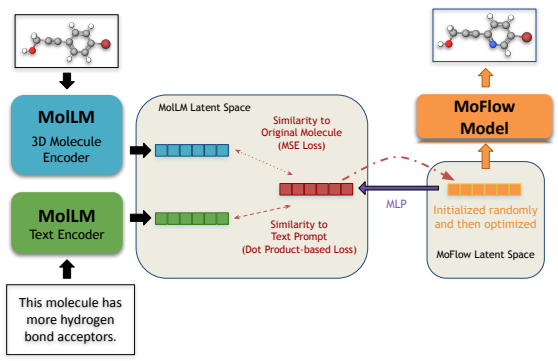

Fig. 5. Visualization of molecule editing task. We start with a molecular graph and an instruction on how to edit the molecule. The graph and text are sent through MolLM's molecule and text encoders, respectively. We optimize on an embedding in MoFlow's latent space but with two losses, to retain similarity to the original molecule while also following the text prompt, computed on a translated embedding within MolLM's latent space. The optimized embedding within MoFlow's latent space is then reversed through MoFlow to obtain the final molecule.

optimize the embedding in MoFlow's latent space so that we can reverse it into the final edited molecule, but with losses computed within the latent space of MolLM. Refer to Figure 5 for a diagram of this process. We pass the input molecule and text prompt through the molecule and text encoders of MolLM respectively to obtain embeddings of each. In the diagram, these are represented as blue and green rectangles respectively.

To begin the optimization process, we initialize an embedding in the MoFlow latent space, represented by the orange rectangle in the diagram, to a random embedding from Gaussian noise. Then, we translate this embedding into an embedding within MolLM latent space, in which we denote the intermediate embedding by a multilayer perceptron (MLP), represented by the purple arrow in Figure 5. Refer to Appendix 7.5 for details of this MLP. Next, for the indirect optimization of the embedding within the MoFlow latent space, we compute two losses within the MolLM latent space (shown in Figure 5 with the double-headed red dashed arrows). The first loss is between the embedding of the original molecule to retain similarity to it. The second loss is between the embedding of the original molecule and the embedding of the text prompt to achieve the desired molecular properties. See Appendix 7.4.4 for details of these losses. After the losses within the MolLM latent space are computed, the gradients with respect to the embedding within the MoFlow latent space, represented by orange in the diagram, are backpropagated, represented with a curved red arrow in the diagram. Then, these gradients are used to optimize the embedding in MoFlow latent space to minimize both of the aforementioned losses. Thus, the optimized embedding represents a molecule with similarity to the input molecule but with properties described in the text prompt. After 600 steps of optimization, we obtain the final state of the embedding. Finally, we reverse the embedding, represented by orange in the diagram, through the invertible MoFlow model to output the final edited molecule, shown in the diagram as the orange arrow.

4 Results

To evaluate MolLM, we test four different multimodality downstream tasks, including molecule-to-text retrieval (M-T), text-to-molecule retrieval (T-M), molecule-to-text caption, and molecule editing with a text prompt.

4.1 Cross-modality Matching

M-T retrieval begins with a 3D molecular graph and aims to select the most relevant textual descriptions associated with the given molecule. Conversely, T-M retrieval starts with a text description and attempts to identify the molecular graph that best corresponds to the description. Both supervised tasks utilize the PCDes dataset (Zeng *et al.*, 2022), which is comprised of SMILES strings and corresponding textual descriptions. The dataset used for zero-shot is consistent with that composed by MoMu. We generate 2D graphs for each example and then convert these SMILES strings into PubChem CIDs. To incorporate 3D conformation data into our model, we modify each graph by adding 3D information using RDKit's MMFF94 implementation. As in (Su *et al.*, 2022), we perform retrieval in randomly selected batches of 64, assessing the average accuracy of top-1 retrieval results and the recall of top-20 (R @ 20) retrieval results. Specifically, the recall of top-1 retrieval measures how often the model is able to retrieve the correct result as its best match while recall of top-20 retrieval measures how often the best match is within what it deems to be the top 20 most similar results. For each task, we create both sentence-level and paragraph-level tasks. In the sentence-level setting, following (Su *et al.*, 2022), we use a randomly chosen sentence from the molecule's description, while in the paragraph-level setting, we utilize the entire description. We adopt Sci-BERT, KV-PLM (Zeng *et al.*, 2022), and MoMu (Su *et al.*, 2022) as baselines.

From Table 3, it is clear that MolLM-3D achieves notable results as it outperforms all baselines for each task. This performance is evident in both zero-shot and supervised settings, across sentence-level and paragraph-level tasks. Notably, the M-T R@20 in the supervised sentence-level task for MolLM-3D is an impressive **92.05±1.03**, indicating a high recall rate and suggesting that the model is highly effective in retrieving relevant textual descriptions for given molecular structures. Similarly, T-M accuracy and recall at the sentence level and paragraph level in the supervised setting show a marked improvement over other models,

Model	M-T Acc	M-T R@20	T-M Acc	T-M R@20
Sentence-level & zero-shot				
Sci-BERT	$1.38^{\pm 0.09}$	$0.38^{\pm 0.06}$	$1.57^{\pm 0.05}$	$0.28^{\pm 0.03}$
KV-PLM	$1.42^{\pm 0.13}$	$0.64^{\pm 0.35}$	$1.43^{\pm 0.06}$	$0.31^{\pm 0.08}$
MoMu	$39.07^{\pm 1.11}$	$37.36^{\pm 0.83}$	$38.51^{\pm 1.04}$	$31.18^{\pm 1.04}$
MolLM-2D	$45.46^{\pm 0.99}$	$46.60^{\pm 0.81}$	$45.25^{\pm 0.90}$	$41.17^{\pm 0.92}$
MolLM-3D	$\mathbf{45.82}^{\pm 0.87}$	$\mathbf{47.29}^{\pm 0.85}$	$\mathbf{46.02}^{\pm 0.90}$	$\mathbf{41.99}^{\pm 0.94}$
Sentence-level & supervised				
Sci-BERT	$50.39^{\pm 1.39}$	$62.11^{\pm 1.39}$	$50.12^{\pm 1.67}$	$68.02^{\pm 1.87}$
KV-PLM	$53.79^{\pm 1.42}$	$66.63^{\pm 1.51}$	$55.22^{\pm 0.94}$	$71.80^{\pm 1.56}$
MoMu	$58.74^{\pm 1.61}$	$81.29^{\pm 0.99}$	$54.94^{\pm 0.73}$	$78.29^{\pm 0.56}$
MolLM-2D	$64.59^{\pm 1.20}$	$91.73^{\pm 2.03}$	$64.83^{\pm 1.89}$	$91.20^{\pm 1.82}$
MolLM-3D	$\mathbf{66.11}^{\pm 0.84}$	$\mathbf{92.05}^{\pm 1.03}$	$\mathbf{66.10}^{\pm 1.05}$	$\mathbf{91.88}^{\pm 1.79}$
Paragraph-level & zero-shot				
Sci-BERT	$1.38^{\pm 0.01}$	$0.34^{\pm 0.05}$	$1.56^{\pm 0.05}$	$0.32^{\pm 0.03}$
KV-PLM	$1.94^{\pm 0.04}$	$0.40^{\pm 0.03}$	$1.82^{\pm 0.12}$	$0.52^{\pm 0.05}$
MoMu	$46.66^{\pm 0.31}$	$43.66^{\pm 0.28}$	$45.66^{\pm 0.12}$	$43.52^{\pm 0.14}$
MolLM-2D	$54.81^{\pm 0.82}$	$54.24^{\pm 0.87}$	$54.95^{\pm 0.85}$	$54.64^{\pm 0.90}$
MolLM-3D	$\mathbf{55.90}^{\pm 0.88}$	$\mathbf{55.44}^{\pm 0.80}$	$\mathbf{55.06}^{\pm 0.83}$	$\mathbf{54.88}^{\pm 0.86}$
Paragraph-level & supervised				
Sci-BERT	$62.57^{\pm 1.33}$	$60.67^{\pm 0.21}$	$61.75^{\pm 0.74}$	$60.77^{\pm 0.68}$
KV-PLM	$64.81^{\pm 0.81}$	$63.87^{\pm 1.14}$	$64.95^{\pm 1.47}$	$64.27^{\pm 1.05}$
MoMu	$81.09^{\pm 0.21}$	$80.15^{\pm 1.06}$	$81.45^{\pm 0.51}$	$80.62^{\pm 0.44}$
MolLM-2D	$86.33^{\pm 1.80}$	$90.67^{\pm 2.11}$	$87.00^{\pm 1.83}$	$91.06^{\pm 1.95}$
MolLM-3D	$\mathbf{87.05}^{\pm 1.94}$	$\mathbf{90.98}^{\pm 1.79}$	$\mathbf{87.85}^{\pm 1.97}$	$\mathbf{91.75}^{\pm 1.26}$

Table 3. Molecule-to-text (M-T) and text-to-molecule (T-M) retrieval results, presented across four different settings: sentence-level zero-shot, sentence-level supervised, paragraph-level zero-shot, and paragraph-level supervised.

with MolLM-3D scoring 66.11 and 91.75, respectively. As MolLM-3D also consistently outperforms MolLM-2D, these results demonstrate the efficacy of incorporating 3D structural information into the retrieval process.

4.2 Property Prediction

Property prediction is a downstream graph task widely used in prior work such as (Su *et al.*, 2022) and (Liu *et al.*, 2022). Similar to these works, we use the MoleculeNet (Wu *et al.*, 2017b) benchmark, whose purpose is to compare how well a pre-trained graph encoder can be adapted to a variety of disparate classification tasks. For example, the TOX21 and HIV datasets might ask a model to classify new molecular graphs for toxicity markers or anti-HIV activity, respectively. We directly compare our model to relevant models including MoMu and KV-PLM that use a similar multimodal strategy to our model. As seen in Table 4, our model excels in property prediction tasks over the relevant baselines. In zero-shot scenarios, MolLM outperforms MoMu by up to 2% on average, demonstrating enhanced generalization capabilities. Upon fine-tuning, MolLM's performance surpasses that of the baselines with significant margins, evident in datasets such as TOX21, with an increase from 75.6% to 80.0%, and BACE, where it improves from 77.8% to 84.1%. These enhancements underscore the efficacy of our more expressive molecular encoder which also captures important 3D information, which can capture structure that is useful for some of these classification tasks where considering molecular structure is important.

4.3 Molecule Caption

We utilize the molecule captioning task as illustrated in (Edwards *et al.*, 2022). In this task, our model generates text relating to a molecule, given its SMILES string, 2D, and 3D graphs. We choose MolT5 (Edwards *et al.*, 2022) as one of our baselines, which generates captions by inputting SMILES strings through their encoder-decoder transformer architecture. The other baseline we choose is MoMu (Su *et al.*, 2022), which expands upon this process by appending 2D structural data to the SMILES embedding inputs. We further introduce 3D data into the input to improve the model's understanding of the molecular structure, and thus generate more accurate descriptions based on the new 3D context. Using the CheBI-20 dataset as in (Edwards *et al.*, 2022), we compare the performance of MolLM, MoMu, and MolT5 on four different captioning metrics. The results can be seen in Fig 6. We see that MolLM significantly outperforms both MolT5 and MoMu on the BLEU, METEOR, and Text2Mol metrics, and performs essentially equivalent to MolT5 on the ROUGE-L metric. In summary, MolLM performs equivalent or slightly better than every single baseline. The fact that MolLM's 3D models do not exhibit better results than its 2D variant could potentially be explained by the importance of 3D information for this specific task, or lack thereof, and also potentially the nature of the expected captions which may not emphasize the need for 3D information. These results indicate that MolLM is at least as performant as every baseline but its added features and expressivity are not useful for this captioning task specifically.

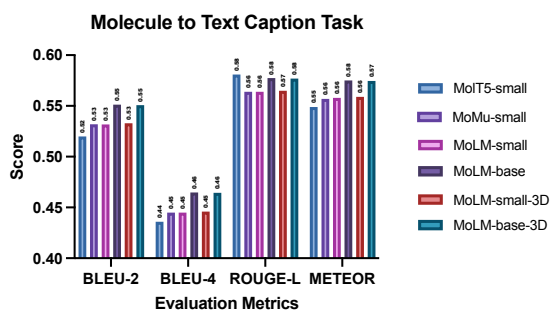

Fig. 6. Our results for molecule captioning task. We utilize the ChEBI dataset (Edwards et al., 2022) with 33,010 molecules. We use BLEU-2, BLEU-4, ROUGE-L, and METEOR metrics to compare performance with MolT5 and MoMu, as our baselines.

4.4 Molecule Editing

Following the procedure and settings within (Liu *et al.*, 2022), we begin with the 200 randomly sampled molecules from ZINC (Sterling and Irwin, 2015) alongside a short text editing prompt as inputs. The task is to perform single-objective editing with prompts, given by Liu *et al.*, including "this molecule has high permeability," which prompts the model to edit a molecule toward a certain molecular property. We evaluate each of these metrics by their satisfactory "hit" ratio, which measures the ratio of the 200 molecules that the model can modify following the prompt, where a successful modification is defined as a "hit." Calculated molecular metrics are used to determine hits, with the metric depending on the prompt. For example, a satisfactory hit for the "high permeability" prompt would be the model outputting a molecule with a higher permeability, measured by a lower calculated topological polar surface area (TPSA). For the other properties, hydrogen bond acceptors, hydrogen bond donors, drug-likeness, and solubility, the respective calculated properties to determine hits are calculated hydrogen bond acceptors, calculated hydrogen bond donors, QED drug-likeness (Bickerton *et al.*, 2012), and Wildman-Crippen LogP value (Wildman and Crippen, 1999). The satisfactory hit ratio results for our model's performance on this task for the selected 200 molecules are scaled to a range of 0.0-100.0.

Looking at the results for molecule editing, distinct hit ratios were observed for various properties. The model achieved moderate success in increasing hydrogen bond acceptors and donors with hit ratios of 46.0% and 54.5% respectively. However, it showed a striking contrast in

Dataset	BBBP	TOX21	ToxCast	SIDER	ClinTox	MUX	HIV	BACE	Average
zero-shot									
MoMu	65.8±4.5	74.0±0.8	63.4±0.6	57.3±1.6	58.0±4.4	71.8±2.5	75.3±1.9	70.1±5.4	66.96
MolLM	**70.6±1.2**	**77.5±0.7**	**65.7±1.3**	**67.1±0.9**	**70.8±1.1**	**74.5±0.6**	**76.1±1.4**	**76.9±0.8**	**72.40**
fine-tune									
RandomForest	71.4±0.0	76.9±1.5	–	68.4±0.9	71.3±5.6	63.2±2.3	78.1±0.6	86.7±0.8	–
RXNFP	68.5 ± 0.7	65.5 ± 0.5	–	54.0 ± 1.6	–	–	73.5 ± 1.0	–	–
BERT	68.4 ± 0.5	68.8 ± 0.7	–	60.2 ± 1.2	–	–	69.4 ± 1.0	–	–
SMI-BERT	71.1 ± 2.2	69.6 ± 0.4	–	59.9 ± 0.9	–	–	73.9 ± 1.5	–	–
GCN	71.8 ± 0.9	70.9 ± 2.6	–	53.6 ± 3.2	62.5 ± 2.8	71.6 ± 4.0	74.0 ± 3.0	71.6 ± 2.0	–
GIN	65.8 ± 4.5	74.0 ± 0.8	–	57.3 ± 1.6	58.0 ± 4.4	71.8 ± 2.5	75.3 ± 1.9	70.1 ± 5.4	–
D-MPNN	71.2±3.8	68.9±1.3	–	63.2±2.3	90.5±5.3	76.2±2.8	75.0±2.1	85.3±5.3	–
MolCLR GCN	73.8±0.2	74.7±0.8	–	66.9± 1.2	86.7 ±1.0	84.0± 1.8	77.8 ± 0.5	78.8 ±0.5	–
MolCLR GIN	73.6±0.5	79.8±0.7	78.9±1.2	68.0±1.1	93.2±1.7	88.6±2.2	80.6 ± 1.1	89.0±0.3	81.46
KV-PLM	72.0±0.9	70.0±0.5	55.0±1.7	59.8±1.5	89.2±2.7	54.6±4.8	71.8±1.4	78.5±2.7	68.73
MoMu-S	70.5±2.0	75.6±0.3	63.4±0.5	60.5±0.9	79.9±4.1	70.5±1.4	75.9±0.8	76.7±2.1	71.63
MoMu-K	70.1±1.4	75.6±0.5	63.0±0.4	60.4±0.8	77.4±4.1	71.1±2.7	76.2±0.9	77.1±1.4	71.36
MolLM	**75.7±1.7**	**80.0±1.7**	**68.2±0.4**	**71±0.8**	**91.1±1.0**	**79±1.2**	**80.2±0.7**	**84.1±0.9**	**76.16**

Table 4. Results of the property prediction task compared to other relevant baselines that also utilize a multimodal biomedical text and molecule model. We also include other fine-tuned models for comparison, but only compare directly against the aforementioned relevant baselines.

Model	Prompt	Hit Ratio
MolLM-3D	This molecule has more hydrogen bond acceptors	46.0
MolLM-3D	This molecule has more hydrogen bond donors.	54.5
MolLM-3D	This molecule is like a drug.	5.00
MolLM-3D	This molecule is not like a drug.	81.5
MolLM-3D	This molecule has high permeability.	33.5
MolLM-3D	This molecule has low permeability.	56.5
MolLM-3D	This molecule is soluble in water.	76.5
MolLM-3D	This molecule is insoluble in water.	21.5

Table 5. Results of molecule editing task presented on different prompts.

modifying drug-likeness, with a low hit ratio of 5.0% for making molecules drug-like and a high of 81.5% for the converse. The ability to edit for permeability was less successful, especially for increasing it (33.5%), compared to decreasing it (56.5%). Notably, the model performed well in enhancing solubility in water with a hit ratio of 76.5%, but only managed a 21.5% success rate for rendering molecules insoluble in water. These results indicate that MolLM shows promising results for certain molecular editing tasks, but there is still room for improvement to make it more robust across various differing prompts.

5 Discussion

We introduce MolLM, a transformer-based multimodal model, to demonstrate the benefit of incorporating biomedical text, 2D molecular, and 3D molecular data into molecular analysis. This section provides an analysis of our choice of model architecture, each downstream task, and the experiments reveal about MolLM's performance.

5.1 Model Architecture

Our choice of model architecture was based on our selection of molecular representations and the advantages offered by Transformers in encoding molecules. For molecular representation, we examined prior works to inform our approach. While SMILES and SELFIES strings are simple and leverage natural language pipelines directly, literature has demonstrated that these representations show limitations. SMILES strings, for instance, can represent the same molecule in multiple ways, leading to unnecessary data redundancy. Additionally, both SMILES and SELFIES representations lack the ability to encode 2D/3D spatial information, making them perform worse in certain tasks.

Thus, we choose to focus on graph-based molecular representations. Our reasoning for continuing to use Transformers to encode molecules stems from the architecture's superior capability to handle sequences. In contrast to traditional GNN or GIN models, which primarily focus on the graph structure of molecules, transformers allow for more flexible encoding. The attention mechanism within Transformers captures molecular patterns, and the encoder-decoder structure allows us to easily leverage multimodal inputs for pretraining. This means that we can also leverage text data for our inputs, giving our model the ability to understand domain knowledge for a given molecule.

5.2 Downstream Tasks

We choose a wide range of tasks, showing MolLM's flexibility in multiple contexts. For certain tasks, such as property prediction, our model doesn't explicitly use the text encoder. However, even without a direct usage of text, the embeddings generated through the molecule encoding are still influenced by the contrastive learning process, thereby indirectly benefiting from the biomedical text knowledge. This concept is evident in our utilization of fixed encoders for tasks, like MolT5's usage in molecule captioning and MoFlow's inclusion in molecule generation. In these tasks, we fix our encoder during fine-tuning, which serves to give us a better understanding of our fine-tuning effects, prevent overfitting, and reduce computational costs. These tasks are able to perform well because the encoders are already pre-trained and our multimodal approach provides enough context prior to fine-tuning.

We achieve results significantly surpassing current state-of-the-art baselines on cross-modality matching and property prediction tasks. In the matching task, we see that including 3D data leads to better results than its 2D counterpart, and both outperform all baselines. These results show not only the importance of including multimodal contexts for model pretraining but also the specific significance of 3D encoding. Similarly with the property prediction task, MolLM's performance in both zero-shot and fine-tuned environments supports the validity of our multimodal approach.

In molecule captioning, our model also achieves competitive results to baselines. We attribute this primarily to the size of our dataset, which while relatively small, is larger than other text-based datasets. Interestingly, we see that the inclusion of 3D data does not significantly improve results

on this task. This could be because of the heavily text-focused nature of the task, with text that may implicitly not require a lot of molecule structure data. In other words, the captions themselves may not be highly correlated to molecular structure data. For an example of the opposite, captions such as "the molecule is symmetrical" may be more conducive to having molecular structural data.

However, in molecule editing, our results were subpar. The model's performance varied significantly depending on the editing task. This variation could be due to the difficulty of specific prompts as some prompts have successful hits for a much narrower range of similar molecules to the original. However, our model only edits molecules to a certain extent to avoid over-editing a molecule. For example, over-editing for the solubility prompt could always be giving a very soluble molecule that has no relation to the original molecule. Additionally, the variation between the prompts may be due to inherent limitations in the training data of how often text related to the prompt occurs.

Investigating further, we found that many of the edits create molecules that are very different from the original, suggesting that a high hit ratio may also come from overly editing a molecule. This indicates issues with the metric itself, suggesting a potential need for developing a more robust metric for evaluation. There is a large downside with how the metric solely looks at changes in related calculated properties and ignores similarity to the original molecule. A large part of the value of molecule editing is that the generated molecule remains highly similar to the original, but the metric encourages editing that strongly favors the prompt without consideration for the original. A better evaluation metric could be some combination of both similarity to the original and how well the prompt was followed. However, the designer of such a metric could just adjust the weighting between these two factors to favor their model. Similar to MoleculeSTM, within the implementation of our model, we have a hyperparameter that controls the linear combination of two losses which each correlate to one of these factors.

The value of integrating 3D information cannot be overstated. By employing a multimodal approach, we capture a richer representation of molecules, highlighting the interplay between their structural and spatial features. Additionally, the quality and size of the datasets used play a pivotal role in the performance of such models. Moving forward, molecular models should consider a multimodal approach that incorporates 3D data. as we have demonstrated the effectiveness of this architecture.

5.3 Future Direction

In terms of future directions, there's an evident need for more extensive and better-curated datasets. Larger datasets with a tighter correlation between text and molecular representations would allow the model to make clearer distinctions between molecules. Automatically finding such text in large text corpora is difficult as molecules being mentioned in certain texts do not necessarily mean the text is relevant. Thus, an improvement to the current text sampling method is necessary. For example, future works could leverage language models to filter this academic text or include more quantitative data from other sources.

Exploring improvements to downstream tasks is also important. This could be in the form of novel downstream tasks, which would allow us to validate the model's applicability across a broader spectrum of challenges in molecular biology and chemistry. However, there are also numerous possible directions for improvement of the downstream tasks that MolLM utilized. For example, a major weakness of the molecule generation task lies in the fact that it sometimes generates infeasible molecules. Future works could aim to validate these generations for chemical validity, chemical stability, or feasibility of synthesis. To do this, Reinforcement Learning from Human Feedback (Ziegler *et al.*, 2019; Stiennon *et al.*, 2020) could be used, creating a reward function based on human professional knowledge to fine-tune the generation model. Transfer learning from models like GPT-4 could also aid in this improvement.

For our molecule editing task, future work could explore evaluating how well MoleculeSTM, our model, and other similar models allow for dynamic adjustment of weighting between emphasizing retaining original molecular features and chemical changes that favor the given prompt. The motivation behind this is that there is likely often the case where a molecule already has many desirable properties and minimal editing is wanted to achieve a prompt. Moreover, a better metric could even look at the ratio of change in the molecule in favor of the prompt to how much chemically the molecule is actually changed. Our experiments also shed light on the lack of reliable baselines, especially in de novo generation, where there is no gold standard. This poses questions about the veracity and stability of generated molecules and presents the opportunity for better baselines to be created and adopted.

Additionally, the editing task highlights how useful having natural language prompts as input into the model is for powerful biomedical-related tasks. Exploring how to make this editing task more robust, experimenting with the prompts used, and proposing new tasks that also have natural language prompts could lead to more powerful tools that can take advantage of the utility and ease of use of having natural language prompts.

Furthermore, the pretraining technique could be expanded upon in a manner more advanced than just curating a larger or higher-quality dataset. There is also an avenue to explore where a much larger language model such as ChatGPT (Jahan *et al.*, 2023) or LLaMA could be used as an agent concerning biomedical tasks.

Finally, explorations into different molecular encoding methods could see promising results. For example, Transformer-M combines 2D/3D data in a linear fashion, but future works could experiment with non-linear combinations of 2D and 3D data. This could allow for increased expressiveness, allowing the model to better understand the nuances between 2D and 3D representations.

6 Conclusion

MolLM, with its emphasis on including 3D positional data, shows strong results across a variety of downstream tasks including cross-modality matching, property prediction, captioning, and editing. This transformative approach underscores the importance of incorporating a richer molecular representation, highlighting the nuanced interplay between structural and spatial features. Additionally, it shows the important role of incorporating biomedical knowledge via text into this interplay. Our model demonstrates a promising future for utilizing computational methods that incorporate the wealth of existing biomedical literature to advance complex tasks such as drug discovery that require a complex understanding that transcends multiple modalities.

References

An, X. *et al.* (2022). Representation of molecules for drug response prediction. *Briefings in Bioinformatics*, **23**(1), bbab393.

Bickerton, G. R. *et al.* (2012). Quantifying the chemical beauty of drugs. *Nature Chemistry*, **4**(2), 90–98.

Chen, J. and Zhang, L. (2021). A survey and systematic assessment of computational methods for drug response prediction. *Briefings in bioinformatics*, **22**(1), 232–246.

Chilingaryan, G. *et al.* (2022). Bartsmiles: Generative masked language models for molecular representations.

Chithrananda, S. *et al.* (2020). Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*.

Coley, C. W. *et al.* (2017). Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, **57**(8), 1757–1772.

Devinyak, O. *et al.* (2014). 3d-morse descriptors explained. *Journal of Molecular Graphics and Modelling*, **54**, 194–203.

Devlin, J. *et al.* (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Edwards, C. *et al.* (2022). Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*.

Gu, Y. *et al.* (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, **3**(1), 1–23.

Handsel, J. *et al.* (2021). Translating the inchi: adapting neural machine translation to predict iupac names from a chemical identifier. *Journal of cheminformatics*, **13**(1), 1–11.

Hirohara, M. *et al.* (2018). Convolutional neural network based on smiles representation of compounds for detecting chemical motif. *BMC Bioinformatics*, **19**(19), 526.

Hu, W. *et al.* (2020). Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, **33**, 22118–22133.

Hwang, D. *et al.* (2020). Comprehensive study on molecular supervised learning with graph neural networks. *Journal of Chemical Information and Modeling*, **60**(12), 5936–5945.

Jahan, I. *et al.* (2023). Evaluation of chatgpt on biomedical tasks: A zero-shot comparison with fine-tuned generative transformers.

Jiang, J. *et al.* (2021). Ggl-tox: geometric graph learning for toxicity prediction. *Journal of chemical information and modeling*, **61**(4), 1691–1700.

Kajita, S. *et al.* (2017). A universal 3d voxel descriptor for solid-state material informatics with deep convolutional neural networks. *Scientific reports*, **7**(1), 16991.

Kaneko, H. (2023). Molecular descriptors, structure generation, and inverse qsar/qspr based on selfies. *ACS omega*.

Kim, S. *et al.* (2023). Pubchem 2023 update. *Nucleic acids research doi:10.1093/nar/gkac956*.

Krenn, M. *et al.* (2019). Selfies: a robust representation of semantically constrained graphs with an example application in chemistry. *arXiv preprint arXiv:1905.13741*, **1**(3).

Kuenzi, B. M. *et al.* (2020). Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer cell*, **38**(5), 672–684.

Kuzminykh, D. *et al.* (2018). 3d molecular representations based on the wave transform for convolutional neural networks. *Molecular pharmaceutics*, **15**(10), 4378–4385.

Landrum, G. (2023). Rdkit: Open-source cheminformatics.

Lee, J. *et al.* (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240.

Lewis, M. *et al.* (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, S. *et al.* (2022a). Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4541–4549.

Li, Z. *et al.* (2022b). Deep learning methods for molecular representation and property prediction. *Drug Discovery Today*, page 103373.

Liu, J. *et al.* (2023a). The prediction of molecular toxicity based on bigru and graphsage. *Computers in Biology and Medicine*, **153**, 106524.

Liu, P. *et al.* (2023b). Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *arXiv preprint arXiv:2308.06911*.

Liu, S. *et al.* (2021). Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*.

Liu, S. *et al.* (2022). Multi-modal molecule structure-text model for text-based retrieval and editing.

Liu, Y. *et al.* (2023c). Molrope-bert: An enhanced molecular representation with rotary position embedding for molecular property prediction. *Journal of Molecular Graphics and Modelling*, **118**, 108344.

Liu, Z. *et al.* (2023d). Dropout reduces underfitting.

Lo, K. *et al.* (2020). S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Luo, R. *et al.* (2022a). Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, **23**(6), bbac409.

Luo, S. *et al.* (2022b). One transformer can understand both 2d & 3d molecular data. *arXiv preprint arXiv:2210.01765*.

Miao, Y. *et al.* (2023). Recent advances in toxicity prediction: Applications of deep graph learning. *Chemical Research in Toxicology*, **36**(8), 1206–1226.

Nigam, A. *et al.* (2021). Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chemical science*, **12**(20), 7079–7090.

Pogány, P. *et al.* (2018). De novo molecule design by translating from reduced graphs to smiles. *Journal of chemical information and modeling*, **59**(3), 1136–1146.

Radford, A. *et al.* (2018). Improving language understanding by generative pre-training. *OpenAI blog*.

Raffel, C. *et al.* (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, **21**, 1–67.

Rong, Y. *et al.* (2020). Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, **33**, 12559–12571.

Ross, J. *et al.* (2022). Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, **4**(12), 1256–1264.

Scholkopf, B. *et al.* (1997). Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, **45**(11), 2758–2765.

Seidl, P. *et al.* (2023). Enhancing activity prediction models in drug discovery with the ability to understand human language. *arXiv preprint arXiv:2303.03363*.

Singhal, K. *et al.* (2023). Large language models encode clinical knowledge. *Nature*, pages 1–9.

Stärk, H. *et al.* (2022). 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pages 20479–20502. PMLR.

Sterling, T. and Irwin, J. J. (2015). Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, **55**(11), 2324–2337.

Stiennon, N. *et al.* (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, **33**, 3008–3021.

Su, B. *et al.* (2022). A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*.

Thomas, N. *et al.* (2018). Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint*

arXiv:1802.08219.

Vaswani, A. *et al.* (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

Wang, S. *et al.* (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*, BCB '19, page 429–436, New York, NY, USA. Association for Computing Machinery.

Wang, Y. *et al.* (2022a). Improving molecular contrastive learning via faulty negative mitigation and decomposed fragment contrast. *Journal of Chemical Information and Modeling*, **62**(11), 2713–2725.

Wang, Y. *et al.* (2022b). Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*.

Wei, J. *et al.* (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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.

Wen, N. *et al.* (2022). A fingerprints based molecular property prediction method using the bert model. *Journal of Cheminformatics*, **14**(1), 1–13.

Wildman, S. and Crippen, G. (1999). Prediction of physicochemical parameters by atomic contributions. *Journal of Chemical Information and Computer Sciences*, **39**, 868–873.

Wu, Z. *et al.* (2017a). Moleculenet: A benchmark for molecular machine learning.

Wu, Z. *et al.* (2017b). Moleculenet: A benchmark for molecular machine learning.

Xia, J. *et al.* (2022). A systematic survey of molecular pre-trained models. *arXiv preprint arXiv:2210.16484*.

Xue, D. *et al.* (2020). X-mol: large-scale pre-training for molecular understanding and diverse molecular analysis. *bioRxiv*, pages 2020–12.

Yang, K. *et al.* (2019). Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, **59**(8), 3370–3388.

Yu, L. *et al.* (2021). Review of unsupervised pretraining strategies for molecules representation. *Briefings in Functional Genomics*, **20**(5), 323–332.

Yüksel, A. *et al.* (2023). Selformer: Molecular representation learning via selfies language models. *Machine Learning: Science and Technology*.

Zang, C. and Wang, F. (2020). Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 617–626.

Zeng, Z. *et al.* (2022). A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, **13**(1), 862.

Zhang, Z. *et al.* (2021). Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, **34**, 15870–15882.

Zhu, J. *et al.* (2022). Unified 2d and 3d pre-training of molecular representations.

Ziegler, D. M. *et al.* (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Zuranski, A. M. *et al.* (2021). Predicting reaction yields via supervised learning. *Accounts of chemical research*, **54**(8), 1856–1865.

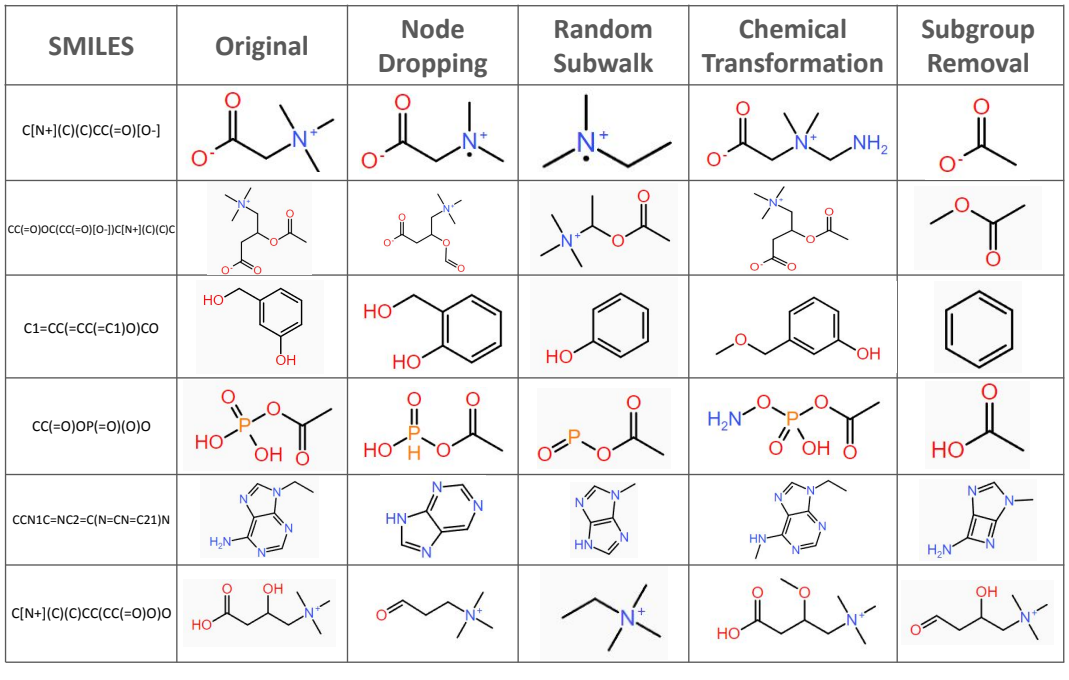

Fig. 7. Examples of different molecule augmentations. For each molecule, their SMILES string, original molecular graph, and the molecular graph of each of the four augmentations are shown.

7 Appendix

7.1 Property Prediction Head

Below are the details of the prediction head we append to the base molecular encoder within our model to adapt to the property prediction task.

The main motivations behind the prediction head layer involve reducing features to learn the most important features for property prediction in the linear layers, ensuring uniform distribution of outputs with the normalization layers for training stability, and dropout to avoid both underfitting and overfitting (Liu *et al.*, 2023d). We perform a grid search to obtain the final prediction head layer dimensions and hyperparameters.

- Linear layer: 768×512
- 1D Batch Normalization: 512 features
- Dropout: rate of 0.2
- Activation: ReLU
- Linear layer: 512×300
- Activation: ReLU
- Dropout: rate of 0.2
- Linear layer: $300 \times C$, where C represents the number of categories for the specific property prediction task.

7.2 Use of RDKit

We use RDKit to represent molecules from SMILES strings and obtain their graph structure prior to transforming their structures to our specific graph format with PyTorch Geometric. The motivation behind having an upper threshold of RMSD after performing molecular augmentation is to ensure that the augmentation does not result in a molecule that is too semantically different for use in contrastive learning.

7.3 Molecule Augmentation Examples

We utilize 4 different molecule augmentations during our pretraining process: Node Dropping, Random Subwalk, Chemical Transformation, and Subgroup Removal. We choose these augmentations as they edit the molecules in a manner in which we believe the semantic meaning of the molecule is kept but is still augmented enough for the model to be more robust and explore the latent space. A more detailed explanation of each of these processes can be found in section 3.2.1. Figure 7 depicts 5 examples of how our augmentations would take an original molecule and convert it into their augmented versions. Let's look at a few examples. The first row shows the molecule Betaine. We can see the effect of node dropping, as one of the carbons has been dropped from the nitrogen. The random subwalk has started on the carbon atom which was connected to the oxygen atoms, and walked through the graph. The chemical transformation has added an amine group to one of the carbon atoms. The subgroup removal has left only one subgroup of the Betaine, based on BRICS decomposition. Another molecule we can examine is in the fourth row: acetylphosphate. We can see that node dropping has caused an oxygen to be dropped from the phosphorus atom. The random subwalk has taken a walk of the molecular graph, starting from the oxygen atom that is double-bonded to the phosphorus atom. The chemical transformation has added an amine group to an oxygen atom. The subgraph removal has left us with a random subgroup of the graph, as dictated by BRICS decomposition.

7.4 Training Implementation Details

7.4.1 Cross-modality Matching

We utilize a contrastive loss between the graph representations and text representations for fine-tuning. See Equation 10 where m is the margin to control how far apart to embeddings of negative pairs should be relative to positive pairs and δ_{ij} is the Kronecker delta function.

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\max\left(0, m+\cos(x_{\text{graph},i}, x_{\text{text},j}) - \delta_{ij}\cos(x_{\text{graph},i}, x_{\text{text},i})\right) \tag{10}$$

Specifically, we utilize $m = 0.2$ throughout all of our fine-tuning of this task. We fine-tune for 60 epochs with a batch size of 64 at a learning rate of $5 \cdot 10^{-5}$ for each subtask.

7.4.2 Property Prediction

For the fine-tuned tasks, which are all classification tasks, we utilize a Binary Cross-Entropy loss. See Equation 11 where N is number of samples, $\hat{y}_i$ is the predicted value, and y_i is the target label.

$$-\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot\log(1-\hat{y}_i)\right] \tag{11}$$

We fine-tune for 200 epochs with a batch size of 32 with learning rate of 10^{-4}. See Appendix 7.1 for details on the linear prediction head for the classification task.

7.4.3 Molecule Caption

We utilize a Cross-Entropy Loss between the predicted captions and target captions. See Equation 12 for detail on this loss where $p(y_{ij}|x_i)$ is the model's probability for a token given the input concatenated SMILES string embedding and molecule embedding, x_i.

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\log(p(y_{ij}|x_i)) \tag{12}$$

We fine-tune for 10 epochs with a batch size of 16 with a learning rate of 10^{-4}.

7.4.4 Molecule Editing

Finally, for the molecule editing task, we do not fine-tune our model as the task is de novo generation. Instead, we optimize the predicted molecule such that its embedding through MolLM is more aligned with the text prompt's embedding within MolLM latent space. We utilize a Mean Squared Error (MSE) between the predicted molecule embedding and the original molecule embedding to maintain similarity to the original molecule. See Equation 13 for the MSE loss, where z_{pred} is the predicted molecule embedding, z_{orig} is the original molecule embedding, and N is the number of samples.

$$\frac{1}{N}\sum_{i=1}^{N}\left(z_{\text{pred},i} - z_{\text{orig},i}\right)^2 \tag{13}$$

Additionally, we incorporate a Dot Product-based loss to ensure alignment between the predicted molecule's embedding and the text prompt's embedding. See Equation 14 for the Dot Product-based loss, where z_{text} is the text prompt's embedding.

$$-\frac{1}{N}\sum_{i=1}^{N}\left(z_{\text{pred},i}\cdot z_{\text{text},i}\right) \tag{14}$$

This dual approach with a linear combination of these two losses for similarity retention and for alignment to the prompt ensures that the edited molecules are related to the original while being edited in a manner relevant to the prompt.

7.5 Embedding Translation MLP

The weights of this MLP for projection from MoFlow latent space to MolLM latent space were trained to minimize cosine similarity between the translated embeddings and the actual MolLM embedding for molecules within the ZINC (Sterling and Irwin, 2015) dataset. The cosine similarity loss for N pairs, where z_{pred} is the predicted translation and z_{true} is the true embedding through MolLM's molecule encoder, is

$$-\frac{1}{N}\sum_{i=1}^{N}\frac{z_{pred}\cdot z_{true}}{\|z_{pred}\|\,\|z_{true}\|} \tag{15}$$