# Deep Interactions for Multimodal Molecular Property Prediction

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Multi-modal learning by means of leveraging both 2D graph and 3D point cloud information has become a prevalent method to improve model performance in molecular property prediction. However, many recent techniques focus on specific pre-training tasks such as contrastive learning, feature blending, and atom/subgraph masking in order to learn multi-modality even though design of model architecture is also impactful for both pre-training and downstream task performance. Relying on pre-training tasks to align 2D and 3D modalities lacks direct interaction which may be more effective in multimodal learning. In this work, we propose MOLINTERACT, which takes a simple yet effective architecture-focused approach to multimodal molecule learning which addresses these challenges. MOLINTERACT leverages an interaction layer for fusing 2D and 3D information and fostering cross-modal alignment, showing strong results using even the simplest pre-training methods such as predicting features of the 3D point cloud and 2D graph. MOLINTERACT exceeds several current state-of-the-art multimodal pre-training techniques and architectures on various downstream 2D and 3D molecule property prediction benchmark tasks.

## 1 Introduction

AI-assisted drug discovery has driven recent research interest in utilizing neural networks for molecule learning. The machine learning community has become especially interested in developing high-quality representations for molecules, which are crucial for predicting molecular properties for a variety of downstream cheminformatic tasks. Self-supervised learning (SSL) on molecular data has emerged as a prevalent research direction to achieve this, leveraging the 2D graph structures of molecules [22, 56, 66]. In parallel, many SSL strategies for 3D point cloud representations of molecules have also been developed [36, 15]. More recent works demonstrate the effectiveness of multimodal SSL techniques which combine 2D and 3D modalities to create better unified representations [54, 34, 36, 77, 60].

Many of these recent and successful multimodal SSL methods make use of SSL techniques from other fields of machine learning. For example, many works leverage attribute and atom/subgraph masking & prediction [22, 66, 75, 24, 34, 35] similar to how masked language-modeling is used to pre-train large transformer-based networks such as BERT [8]. Other works [34, 54, 62] prefer to leverage contrastive learning [5] in order to align the 2D and 3D views of molecules in a unified embedding space similar to how CLIP [45] aligns caption and image embeddings and SimCLR [3] aligns views of images.

These recent approaches typically consider improving molecular SSL via specific pre-training strategies and tasks, but not the underlying architecture. For instance, a common approach [34, 35, 54] is to take two separate models for encoding 3D and 2D structures and then design a pre-training

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

task to align their output embeddings. Alternatively, other works take a single, modality-agnostic model, usually a transformer [58], and task it with predicting multimodal properties such as bond angles [60, 77] or shortest-path distances [73]. Both of these approaches rely on a chosen pre-training task to align 2D and 3D views of molecules.

However, it is not clear to what extent such approaches are able to fully learn cross-modal interactions. For example, contrastive approaches using separate encoders seek to maximize the mutual information between coarse-grained molecule embeddings, and so they may fail to capture key fine-grained relationships. On the other hand, predictive approaches using a single backbone typically accept only atom identities as input, leaving the pre-training task as the only source of multi-modality, potentially missing features which can be extracted by modality-specific encoders. An additional issue is that many pre-training tasks are complex and may require extensive tuning. For example, GraphMVP [34] and MoleculeSDE [35] use a variational autoencoder [27] and diffusion model [19], respectively, to reconstruct the original 2D and 3D structures, which risk mode collapse and are sensitive to hyperparameters such as noise schedules. Other techniques like MoleBlend [73] require both coordinate denoising [15] and prediction of blended multimodal features, where the ratio of noised nodes and 2D & 3D features needs to be tuned.

In order to achieve fine-grained multi-modal information with simpler pre-training tasks, we turn our focus to the role of architectural design for more effective SSL. This work introduces MOLINTERACT, a deep learning architecture designed to fuse 2D and 3D modalities of molecules to better foster multimodal performance. MOLINTERACT uses a series of interaction layers to learn how to combine 2D and 3D embeddings. Specifically, MOLINTERACT consists of two message-passing entrypoints for 2D and 3D data to produce corresponding 2D and 3D embeddings which are then fused and split apart repeatedly in order to exchange unimodal information during pre-training. We pair MOLINTERACT with a set of simple pre-training tasks from the existing literature, such as bond and dihedral angle prediction, which are both sensible in a molecular context and require virtually no tuning. We show that, even with such straightforward pre-training tasks and architecture, MOLINTERACT is able to yield strong multimodal performance, emphasizing the impact of directly fusing 2D and 3D atom embeddings in a model-based approach to improving SSL. We conduct various experiments to demonstrate state-of-the-art performance across various downstream 2D and 3D benchmark tasks.

## 2 Background and Related Work

**SSL for molecules.** Self-supervised learning (SSL) [30, 37] has been adopted in a wide range of domains to obtain high quality representations for downstream tasks. A slew of recent works have emerged attempting to apply the same pre-train-then-finetune paradigm to molecule learning. In particular, research has been aimed at molecular SSL with the primary downstream task of molecule property prediction [4, 75] in mind due to the potential of saving tremendous amounts of time screening new drugs and compounds. However, the success of molecule property prediction requires a comprehensive extraction of molecular features from various modalities, which becomes especially important when only one modality is available for a given real-world molecule. For example, in certain cases, only a compound's 2D structure may be known, but there may be little to no data on its equilibrium conformers. In light of this, it has become important to solve the challenge of learning informative molecular representations using all kinds of modalities, particularly 2D graphs and 3D point clouds.

**Existing work on multimodal SSL.** In order to fuse multimodal representations, works such as GraphMVP [37] and 3DInfomax [54] aim to maximize the mutual information between the 2D and 3D views of molecules, treating 2D graphs with their corresponding 3D conformations as positive samples and all other pairs as negative samples. Alternatively, another line of work proposes to incorporate both modalities via prediction of the original data. MoleculeSDE [35] generates 3D SE(3)-equivariant conformations from the 2D graph, and vice versa, and Zhu et al. [78] use a single model to reconstruct the input 2D graph from the 3D point cloud and vice versa. Similarly, many works [34, 35] task unimodal models with predicting masked sets of 2D and 3D atoms. Other works such as ChemRL-GEM [10] and 3D-PGT [61] propose to predict internal coordinates such as bond length, bond angle, and dihedral angles in order to distill 2D and 3D information. In contrast to these methods, MOLINTERACT seeks to supplement a set of predictive pre-training tasks by leveraging a fusion layer to force interactions between 2D and 3D embeddings to facilitate multimodal learning.

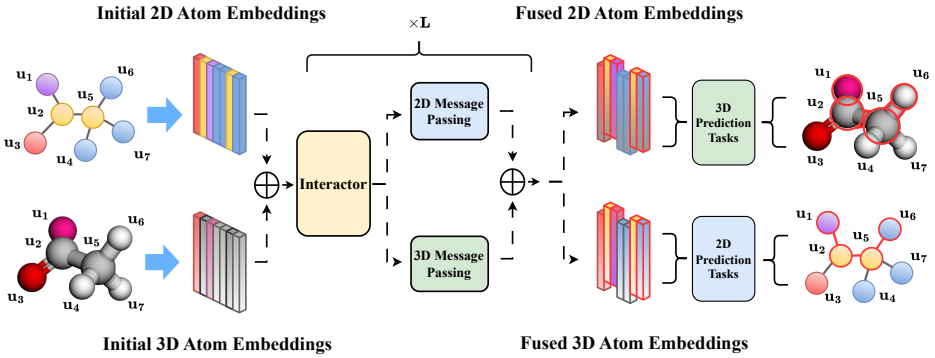



Figure 1: The proposed method's pre-training pipeline. From left-to-right, the input molecule's 2D and 3D graphs are used to derive initial 2D and 3D atom embeddings via message-passing layers. These embeddings are then mixed by an interaction layer before being fed back into the unimodal message-passing branches of the architecture. This process of message-passing followed by interaction repeats $L$ times before the final embeddings from each tower are used for pre-training tasks of the opposite modality, e.g. predicting 3D quantities using the 2D encoder embeddings. Not shown are residual connections between each interaction block to preserve lower-order information.

**Modality interaction.** In order to fully leverage the synergy between different modalities, recent works from other fields propose to learn more fine-grained modality alignment through deep interactive architectures. GreaseLM [74] and Dragon [71] propose to align language models and graph neural networks on knowledge graphs through an interaction token, aiming to integrate text and graph modalities to better identify relevant relations and entities in a given passage. Other works [6, 31] design similar deep interaction layers in various domains such as social networks and recommendation. MOLINTERACT takes inspiration from these methods, proposing to interact 2D and 3D modalities on a fine-grained level in order to better facilitate pre-training and create high-performing multimodal representations.

# 3 Method - Deep Interactions

## 3.1 Notation and Preliminaries

We consider molecules in terms of their 2D graph and 3D point cloud modalities. For simplicity, we will use the term "graph" to refer to both the typical 2D node-edge formalism as well as a molecule's 3D point cloud. We denote the 2D graph of a molecule with $n$ atoms by $G_{\text{2D}} = (V, E, \boldsymbol{X}, \boldsymbol{B})$ where $V$ is a set of its atoms (nodes), $E$ is a set of its bonds (edges), $\boldsymbol{X} \in \mathbb{R}^{n \times d_V}$ is a 2D feature matrix corresponding to the atoms of the molecule with features specific to the 2D graph, such as membership in a ring [21], and $\boldsymbol{B} \in \mathbb{R}^{|E| \times d_E}$ is an edge feature matrix corresponding to edge information such as bond type. For simplicity, we let $d = d_V = d_E$. We also define the 3D graph of a molecule by $G_{\text{3D}} = (\boldsymbol{R}, \boldsymbol{X})$ where $\boldsymbol{R} \in \mathbb{R}^{n \times 3}$ is the molecule's position matrix where rows are $(x, y, z)$ coordinates in 3D space. Unless otherwise specified, we use $z_{\text{2D}}^{(\ell,i)}$ and $z_{\text{3D}}^{(\ell,i)}$ to refer to the $i$th 2D and 3D atom embedding resulting from the $\ell$th layer of a neural network. For simplicity, we assume that all embeddings are of dimension $d$, and we use $\boldsymbol{W}_{(\cdot)}$ to refer to a linear layer with the bias omitted. All classification-based loss functions use CE to stand for Cross-Entropy.

MOLINTERACT is comprised of two central components: (1) an architecture which fosters deep multimodal interactions, and (2) a pre-training scheme which leverages this architecture to enforce multimodal understanding similar to previous works. We introduce each component one-by-one in the following sections.

## 3.2 Model Architecture

During pre-training and multi-modal fine-tuning, MOLINTERACT receives a molecule's 2D and 3D views $G_{\text{2D}}$ and $G_{\text{3D}}$. These views are then passed through a two-tower architecture which alternates

between phases of message-passing and interaction. Each tower is a 2D and 3D modality-specific stack of encoders periodically conjoined by interactor layers as visualized in Figure 1.

**2D and 3D atom encoders.** In order to compute 2D atom embeddings, we follow previous work on multimodal pre-training [34, 54, 36, 77, 60] and use message-passing graph neural network [13] (MPNN) layers as 2D encoders. Given a molecular graph $G_{\text{2D}}$ and one of its nodes $i$, its 2D node embedding $h_i$ at the $(\ell+1)$th layer of an MPNN is given by

$$h_i^{(\ell+1)} = \text{Update}\left(h_i^{(\ell)}, \underset{j \in \mathcal{N}(i)}{\text{Agg}}\ \psi\left(h_i^{(\ell)}, h_j^{(\ell)}, e_{ij}\right)\right) \tag{1}$$

where Update is a function which updates the node embedding, Agg is a permutation-invariant function on the neighbors of $i$, and $\psi$ is a function which computes "messages", or interactions, between the node $i$ and its neighbor $j$ with the edge between them as context. In our case, we use layers from the GINE [22] architecture, which is a variant of GIN [68] that incorporates edge features during message-passing. We choose GINE due to its simplicity and 1-WL-expressivity [64], although we note that MOLINTERACT places no restrictions on its 2D backbone, and one can easily replace the MPNN with a more powerful 2D model such as a graph transformer [9, 48, 40] as other multimodal works [60, 73] do to improve performance.

To compute 3D atom embeddings, we opt to use the continuous convolutional layers from SchNet [52]. These layers conduct message-passing according to relative distances between atoms, incorporating both geometric and atom information into the resulting embeddings. Similar to the 2D encoder, MOLINTERACT is agnostic to the choice of 3D encoder, and so one may choose to opt for more expressive 3D backbones [12, 28, 38, 59, 53]. Despite the limited expressivity of SchNet and GIN, we find that MOLINTERACT is able to outperform several state-of-the-art methods, which we show in Section 4.

### 3.3 Multimodal Interaction Layer

At the $\ell$th layer of unimodal message-passing, we take the 2D and 3D atom embeddings $z_{\text{2D}}^{(\ell,i)}$ and $z_{\text{3D}}^{(\ell,i)}$ and pass them to an interaction layer $\phi^{(\ell)}$. Then, the updated atom embeddings $z_{\text{2D}}^{(\ell+1,i)}$ and $z_{\text{3D}}^{(\ell+1,i)}$ are decoded from the output of $\phi^{(\ell)}$ and fed back into their respective unimodal message-passing towers. There are a variety of options for choosing $\phi^{(\ell)}$, such as an attention-based aggregation between the embeddings, or the aggregation of representative nodes [74], such as a virtual node for the 2D graph [13, 44, 25] or a center-of-mass node for the 3D graph. However, for simplicity, we use a 2-layer MLP of dimension $2d$ with Swish activation [18] for $\phi^{(\ell)}$, and feed it the concatenated unimodal embeddings. We run ablations testing different functions for each $\phi^{(\ell)}$ in Table 2. For decoding, we simply split the output of the MLP in half along the channel dimension to retrieve the updated 2D and 3D embeddings. With this, we do not risk information loss via pooling or choosing a representative token for the whole molecule, attaining more granular, node-level interactions. Formally, the multimodal embeddings at layer $\ell+1$ are given by

$$w^{(\ell+1,i)} = \phi^{(\ell)}(z_{\text{2D}}^{(\ell,i)}, z_{\text{3D}}^{(\ell,i)}) = \text{MLP}^{(\ell)}\left(z_{\text{2D}}^{(\ell,i)} \middle|\middle| z_{\text{3D}}^{(\ell,i)}\right) \tag{2}$$

where $||$ denotes concatenation in column-major order. Then, our updated atom embeddings can be written in the following implementation-friendly way:

$$z_{\text{2D}}^{(\ell+1,i)}, z_{\text{3D}}^{(\ell+1,i)} = w_{:,:d}^{(\ell+1,i)}, w_{:,d:}^{(\ell+1,i)} \tag{3}$$

where the subscripts of $w^{(\ell+1,i)}$ denote Python-like indices. this way, the 2D embeddings are a fusion of both 2D and 3D features, and similarly for the 3D embeddings with each subsequent iteration of message-passing and interaction, encoding higher-order multimodal features. Unlike molecule-level approaches like 3D Infomax [54], GraphMVP [34], and MoleculeSDE [35], and unlike modality-agnostic backbone-based methods like MoleBlend [73], MOLINTERACT benefits from both fine-grained, atom-level interactions as well as modality-specific encoders to create more powerful multimodal representations. See Appendix F to see a UMAP visualization of the molecule embeddings from MOLINTERACT compared with MoleculeSDE.

The pre-training tasks for MOLINTERACT are designed to be heterogeneous, meaning 3D quantities are predicted using $z_{\text{2D}}^{(\ell,i)}$, and 2D quantities are predicted using $z_{\text{3D}}^{(\ell,i)}$. By using embeddings of one modality to predict features of the opposite modality, we maximize a lower bound of the mutual information between the modalities. As Liu et al. [34] note, if $X_{3D}, X_{2D}$ denote random variables from 2D and 3D spaces, then their mutual information (MI) $I(X_{3D}, X_{2D})$ is bounded below by $-\frac{1}{2}\left(H(X_{3D}|X_{2D}) + H(X_{2D}|X_{3D})\right)$ where $H$ denotes entropy. Visibly, this bound is maximized when $p(x_{3D}|x_{2D})$ and $p(x_{2D}|x_{3D})$ are also maximized. This motivates the pre-training pipeline in this work where predicting 2D quantities from 3D embeddings and 3D quantities from 2D embeddings maximizes the MI between 2D and 3D information in MOLINTERACT.

Intuitively, during pre-training, the interaction layer $\phi$ serves as a exchange pathway between the unimodal towers. As the only point of contact between the 2D and 3D message-passing layers, $\phi$ must effectively route the most important cross-modal information relevant to the pre-training task. During fine-tuning, each $\phi$ serves as an aggregator of multimodal features, learning to fuse 2D and 3D information effectively for the given downstream task.

### 3.4 Pre-training Tasks

MOLINTERACT's architecture is complemented by a set of simple pre-training tasks in order to facilitate multimodal learning using its fused atom embeddings. Specifically, for our 3D tasks, we choose to predict interatomic distances, bond angles, and dihedral angles. We then predict edge type, shortest-path distance, and centrality ranking as our 2D tasks. We choose these specific pre-training tasks due to (1) their hierarchical relationships with each other in their respective modalities and (2) their ease of computation. The intuition behind (1) is that, in order to learn a comprehensive multimodal representation, both lower-order and higher-order geometric/topological features must be learned and infused in the post-interaction atom embeddings. We value (2) for efficiency's sake, noting that computing these quantities is fairly straightforward and require little to no tuning to be effective. Such prediction tasks are among the simplest in the molecular SSL literature compared to diffusion models and substructure masking, which highlights the effectiveness of the interaction layers during pre-training.

**3D Tasks.** For our 3D tasks, similar to [60] and [10], we choose to predict interatomic distances, bond angles, and dihedral angles: three of the some of the basic internal coordinates in 3D molecular graphs. Not only are such 3D graphs uniquely identified by these primitives [38, 59], but they are also crucially linked to energy-based properties of molecules, making them important geometric priors to encode in 2D embeddings. Furthermore, these angles are prime examples of hierarchical quantities since interatomic distances can be used to compute bond angles, and bond angles can be used to compute dihedral angles, ascending from 2-tuple atom information to 4-tuple atom information.

Given an $L$-layer instance of MOLINTERACT, We introduce our interatomic distance loss and bond angle losses as follows:

$$\mathcal{L}_{\text{Inter}} = \frac{1}{|I|} \sum_{(i,j)\in I} \left( \boldsymbol{W}_{\text{Inter}} \left( z_{\text{2D}}^{(L,i)} + z_{\text{2D}}^{(L,j)} \right) - \alpha_{ij} \right)^2 \tag{4}$$

where $I$ is a set of sampled atom index pairs, and $\alpha_{ij}$ is the ground-truth interatomic distance between atoms $i$ and $j$. Note that we add the 2D embeddings in this loss formulation due to the requirement that distances between atoms are symmetric ($\alpha_{ij} = \alpha_{ji}$), and therefore their encodings should be permutation-invariant. Our loss function for predicting bond angles is similarly defined:

$$\mathcal{L}_{\text{B}} = \frac{1}{|B|} \sum_{(i,j,k)\in B} \left( \boldsymbol{W}_{\text{B}} \left( z_{\text{2D}}^{(L,i)} || z_{\text{2D}}^{(L,j)} || z_{\text{2D}}^{(L,k)} \right) - \beta_{ijk} \right)^2 \tag{5}$$

where $B$ is a set of computed bond angles of adjacent atoms, and $i, j, k$ denote indices of the respective anchor, source, and destination atoms. Note that we concatenate the indexed atom embeddings rather than sum them since bond angles differ depending on which atoms are chosen as anchors.

Finally, while our dihedral angle prediction loss could be analogously defined, we found that our model had difficulty predicting dihedral angles directly, with MSE barely reducing from 0.475 in the first epoch to 0.45 in the 50th when pre-training on PCQM4Mv2, indicating that no useful angle information was being learned. To improve learning, we replace the angle regression task with an

angle classification task, where the task becomes to categorize quadruplets of atom embeddings according to what bin the corresponding dihedral angle belongs to. This is a relatively easier task than direct prediction, and we subsequently saw an increase in performance. Formally, our loss is the cross-entropy term

$$\mathcal{L}_{\text{D}} = -\frac{1}{|D|} \sum_{(i,j,k,l)\in D} \text{CE}\left(\mathcal{D}_b, \boldsymbol{W}_{\text{D}}\left(z_{2D}^{(L,*)}\right)\right) \tag{6}$$

where $D$ is a set of dihedral angles, $z_{2D}^{(L,*)}$ is short for $z_{\text{2D}}^{(L,i)}||z_{\text{2D}}^{(L,j)}||z_{\text{2D}}^{(L,k)}||z_{\text{2D}}^{(L,\ell)}$ where $i, j, k, l$ are indices of the atoms which form a dihedral angle from $D$, and each $\mathcal{D}_b$ indexes the bin to which each angle belongs. In our case, we use $|\mathcal{D}| = 20$. We note that Guha et al. [17] take a similar approach to turning a regression problem into a classification problem, although they do this to aid in conformer prediction rather than angle prediction.

**2D Tasks.** The ways in which 2D topological quantities are relevant for molecular property prediction are more subtle than in the 3D case. Given that $G_{3D}$ is missing key information regarding atom-atom relationships such as bond types, we first task MOLINTERACT with classifying edges $f$rom $G_{2D}$ according to the cross-entropy loss term

$$\mathcal{L}_{\text{Edge}} = -\frac{1}{|E|} \sum_{(i,j)\in E} \text{CE}\left(\mathbf{B}_{ij}, \boldsymbol{W}_{\text{Edge}}\left(z_{\text{3D}}^{(L,i)} + z_{\text{3D}}^{(L,j)}\right)\right) \tag{7}$$

where $\boldsymbol{B}_{ij}$ indexes the label of the corresponding edge $(i, j)$. With this loss, we aim to instill precise atom relational information in the 3D embeddings. Next, a logically higher-order task is to determine the shortest-path distances (SPDs) between atoms, similar to Transformer-M [40] and MOLEBLEND [73], which encodes a global characterization of the molecule's topology. Further, bond type information may serve as a useful preliminary task given that edge classification implicitly informs the modeling of existing edges, meaning that SPD prediction becomes a task of counting which of the said edges are incident. Formally, our SPD loss is formulated as

$$\mathcal{L}_{\text{SPD}} = -\frac{1}{|C|} \sum_{(i,j)\in C} \text{CE}\left(\boldsymbol{D}_{ij}, \boldsymbol{W}_{\text{SPD}}\left(z_{\text{3D}}^{(L,i)} + z_{\text{3D}}^{(L,j)}\right)\right) \tag{8}$$

where $C \subseteq V \times V$ is a set of sampled node pairs, and $\boldsymbol{D}_{ij}$ corresponds to the SPD between atoms $i$ and $j$.

Our final 2D pre-training task is centrality ranking, which aims to use SPD information from the preceding pre-training task to capture global structure. Centrality [1, 2] is a concept from network science which quantifies node importance. In the molecular case, centrality might be used as an indicator of structural importance such as acting as a bridge between a ring or functional group [43]. Furthermore, centrality may act as a proxy for structure/subgraph membership since atoms which participate in chemically relevant substructures are likely to have similar centrality measures. This may serve as informative signal for the 3D tower of MOLINTERACT which is ignorant of the 2D graph topology. In this way, learning to rank nodes by centrality may be thought of as a proxy task to more complex structural pre-training tasks such as subgraph masking, replacing subgraph sampling steps with the simple cross-entropy loss term

$$\mathcal{L}_{\text{Cent}} = -\frac{1}{|C|} \sum_{(i,j)\in C} \text{CE}\left(\mathcal{C}_{i,j}\boldsymbol{W}_{\text{Cent}}\left(z_{\text{3D}}^{(L,i)} + z_{\text{3D}}^{(L,j)}\right)\right) \tag{9}$$

where $\mathcal{C}_{i,j} = 1$ if node $i$ has a higher centrality than node $j$ and 0 otherwise. Among the various definitions of centrality, we experiment with betweenness centrality and eigenvector centrality [63, 11]. Betweenness centrality of a node $v$ is defined as $\sum_{s,t\in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$, where $\sigma(s, t|v)$ stands for the number of shortest paths between $s$ and $t$ which pass through $v$, which would appear to reuse information from $\mathcal{L}_{\text{SPD}}$. However, in practice, we observe superior performance using the eigenvector centrality of each node $u$, defined as the $u$th entry of the eigenvector corresponding to the largest eigenvalue of the 2D graph's adjacency matrix. Intuitively, information learned during SPD prediction may still be used since a node's eigenvector centrality is proportional to the number of infinite random walks passing through that node. We provide some visualizations of betweenness and eigenvector centrality on molecular graphs in Appendix B.

| Method | $\alpha$ | $\Delta\mathcal{E}$ | $\mathcal{E}_{\text{HOMO}}$ | $\mathcal{E}_{\text{LUMO}}$ | $\mu$ | $C_v$ | $G$ | $H$ | $R^2$ | $U$ | $U_0$ | ZPVE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Stock SchNet (Schütt et al. [52]), 8 layers | 0.076 | 51.28 | 33.17 | 26.53 | 0.032 | 0.031 | 17.86 | 15.77 | 0.146 | 17.88 | 18.24 | 1.605 |
| Distance Prediction (Liu et al. [36]) | 0.065 | 45.87 | 27.61 | 23.34 | 0.031 | 0.033 | 14.83 | 15.81 | 0.248 | 15.07 | 15.01 | 1.837 |
| 3D InfoGraph (Liu et al. [36]) | 0.062 | 45.96 | 29.29 | 24.60 | 0.028 | 0.030 | 13.93 | 13.97 | 0.133 | 13.55 | 13.47 | 1.644 |
| 3D InfoMax (Stärk et al. [54]) | 0.057 | 42.09 | 25.90 | 21.60 | 0.028 | 0.030 | 13.73 | 13.62 | 0.141 | 13.81 | 13.30 | 1.670 |
| GraphMVP (Liu et al. [34]) | 0.056 | 41.99 | 25.75 | 21.58 | 0.027 | 0.029 | 13.43 | 13.31 | 0.136 | 13.03 | 13.07 | 1.609 |
| MoleculeSDE (Liu et al. [35]) | 0.054 | 41.77 | 25.74 | 21.41 | 0.026 | 0.028 | 13.07 | 12.05 | 0.151 | 12.54 | 12.04 | 1.587 |
| MOLEBLEND (Yu et al. [73]) | 0.060 | **34.75** | <u>21.47</u> | <u>19.23</u> | 0.037 | 0.031 | 12.44 | 11.97 | 0.417 | 12.02 | 11.82 | 1.580 |
| MOLINTERACT (no pre-training) | <u>0.048</u> | 37.66 | 21.87 | 19.45 | <u>0.022</u> | <u>0.026</u> | 9.54 | 8.84 | <u>0.119</u> | 8.77 | 8.421 | 1.396 |
| MOLINTERACT ($\mathcal{L}_{\text{Simple}}$) | **0.047** | 35.92 | 21.54 | <u>18.34</u> | **0.021** | **0.025** | <u>9.13</u> | <u>8.26</u> | **0.097** | <u>8.16</u> | <u>8.17</u> | <u>1.365</u> |
| MOLINTERACT ($\mathcal{L}_{\text{All}}$) | **0.047** | <u>35.58</u> | **20.60** | **17.88** | **0.021** | **0.025** | **8.56** | **8.24** | 0.136 | **7.92** | **7.72** | **1.327** |

Table 1: Performance on QM9 measured in MAE. Lower is better.

| Pre-training method | $\alpha$ | $\Delta\mathcal{E}$ | $\mathcal{E}_{\text{HOMO}}$ | $\mathcal{E}_{\text{LUMO}}$ | $\mu$ | $C_v$ | $G$ | $H$ | $R^2$ | $U$ | $U_0$ | ZPVE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Only 3D tasks | 0.048 | 36.58 | 21.12 | 18.54 | 0.024 | 0.026 | 9.45 | 8.80 | 0.152 | 8.62 | 8.70 | 1.448 |
| Betweenness | 0.050 | 37.78 | 22.18 | 18.90 | 0.025 | 0.027 | 10.08 | 10.30 | 0.171 | 9.71 | 10.50 | 1.508 |
| Mean Interactor | 0.061 | 46.82 | 30.89 | 24.18 | 0.035 | 0.030 | 11.93 | 12.06 | 0.126 | 11.67 | 11.99 | 1.543 |
| Self-Attention Interactor | 0.074 | 52.73 | 32.27 | 27.74 | 0.042 | 0.034 | 14.80 | 14.09 | 0.116 | 14.41 | 13.98 | 1.749 |
| $\mathcal{L}_{\text{Simple}}$, 3D structures only | 0.057 | 42.00 | 24.78 | 20.77 | 0.022 | 0.028 | 13.47 | 12.87 | 0.163 | 12.75 | 12.36 | 1.480 |

Table 2: Ablations of MOLINTERACT on QM9.

Finally, the total loss formulation during pre-training is $\mathcal{L}_{\text{All}} = \mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{B}} + \mathcal{L}_{\text{D}} + \mathcal{L}_{\text{Edge}} + \mathcal{L}_{\text{SPD}} + \mathcal{L}_{\text{Cent}}$. In practice, we see that each loss term exhibits varying influence since terms like $\mathcal{L}_{\text{edge}}$ are naturally easier to minimize than more complex terms like $\mathcal{L}_{\text{D}}$. Therefore, in our experiments, we compare both MOLINTERACT using $\mathcal{L}_{\text{All}}$ and $\mathcal{L}_{\text{Simple}} = \mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{SPD}}$, and find comparable performance. We find that $\mathcal{L}_{\text{Inter}}$ and $\mathcal{L}_{\text{SPD}}$ work best together, achieving the best overall performance among all the tasks considered. Intuitively, interatomic distances with shortest-path distances give the minimum description of the topology of the 3D and 2D graphs. For a more detailed analysis of the behavior of these losses, see Appendix E. In summary, each of these 3D and 2D pre-training tasks play a role in forming a unified molecular representation in terms of geometric and topological quantities in increasing levels of complexity.

# 4 Experiments

## 4.1 Datasets and Experimental Setup

We pre-train an 8-layer version of MOLINTERACT with 9M parameters for 50 epochs on PCQM4Mv2 [21], which contains over 3.3M molecules with their DFT-computed 3D structures from the PubChemQC [42] project. For evaluation, we evaluate MOLINTERACTon 12 tasks from QM9 [46] and 8 datasets from MoleculeNet [65] in order to compare our method with works in the multimodal molecular SSL literature. We compare MOLINTERACT on QM9 and MoleculeNet with the same baselines as reported by the comprehensive study by Yu et al. [73]. All best metrics are **bolded**, and second-best metrics are <u>underlined</u>. Results for QM9 are measured in mean absolute error (MAE), and results for MoleculeNet are measured in ROC AUC. All metrics reported are from each dataset's test split using the weights which perform best on a validation set. We also include results on QM8 [47, 51] in Appendix D due to space limitations.

In datasets where both 2D and 3D information are available such as QM9 and QM8, we provide both 2D and 3D structures to MOLINTERACT, aggregate the resulting embeddings with mean pooling, and then input their concatenation to a 2-layer MLP head. Otherwise, when only one modality is available, as in MoleculeNet, only the corresponding unimodal branch of our method is activated while the frozen atom embeddings of the other modality are used as input to the interaction layers in place of the embeddings produced by the disabled complementary branch.

| Method | BBBP | Tox21 | ToxCast | SIDER | ClinTox | MUV | HIV | BACE | Average |
|---|---|---|---|---|---|---|---|---|---|
| AttrMask (Hu et al. [22]) | 65.0±2.3 | 74.8±0.2 | 62.9±0.1 | 61.2±0.1 | 87.7±1.1 | 73.4±2.0 | 76.8±0.5 | 79.7±0.3 | 72.68 |
| ContextPred (Hu et al. [22]) | 65.7±0.6 | 74.2±0.0 | 62.5±0.3 | 62.2±0.5 | 77.2±0.8 | 75.3±1.5 | 77.1±0.8 | 76.0±2.0 | 71.28 |
| GraphCL (You et al. [72]) | 69.7±0.6 | 73.9±0.6 | 62.4±0.5 | 60.5±0.8 | 76.0±2.6 | 69.8±2.6 | 78.5±1.2 | 75.4±1.4 | 70.78 |
| InfoGraph (Sun et al. [56]) | 67.5±0.1 | 73.2±0.4 | 63.7±0.5 | 59.9±0.3 | 76.5±1.0 | 74.1±0.7 | 75.1±0.9 | 77.8±0.8 | 70.98 |
| GROVER (Rong et al. [50]) | 70.0±0.10 | 74.3±0.1 | 65.4±0.4 | 64.8±0.6 | 81.2±3.0 | 67.3±1.8 | 62.5±0.9 | 82.6±0.7 | 71.01 |
| MolCLR (Wang et al. [62]) | 66.6±1.8 | 73.0±0.1 | 62.9±0.3 | 57.5±1.7 | 86.1±0.9 | 72.5±2.3 | 76.2±1.5 | 71.5±3.1 | 70.79 |
| GraphLoG (Xu et al. [69]) | 72.5±0.8 | 75.7±0.5 | 63.5±0.7 | 61.2±1.1 | 76.7±3.3 | 76.0±1.1 | 77.8±0.8 | 83.5±1.2 | 73.40 |
| MGSSL (Zhang et al. [75]) | 69.7±0.9 | 76.5±0.3 | 64.1±0.7 | 61.8±0.8 | 80.7±2.1 | 78.7±1.5 | 78.8±1.2 | 79.1±0.9 | 73.70 |
| GraphMAE (Hou et al. [20]) | 72.0±0.6 | 75.5±0.6 | 64.1±0.3 | 60.3±1.1 | 82.3±1.2 | 76.3±2.4 | 77.2±1.0 | 83.1±0.9 | 73.85 |
| Mole-BERT (Xia et al. [66]) | 71.9±1.6 | 76.8±0.5 | 64.3±0.2 | 62.8±1.1 | 78.9±3.0 | 78.6±1.8 | 78.2±0.8 | 80.8±1.4 | 74.04 |
| 3D InfoMax (Stärk et al. [54]) | 69.1±1.0 | 74.5±0.7 | 64.4±0.8 | 60.6±0.7 | 79.9±3.4 | 74.4±2.4 | 76.1±1.3 | 79.7±1.5 | 72.34 |
| GraphMVP (Liu et al. [34]) | 68.5±0.2 | 74.5±0.4 | 62.7±0.1 | 62.3±1.6 | 79.0±2.5 | 75.0±1.4 | 74.8±1.4 | 76.8±1.1 | 71.69 |
| MoleculeSDE (Liu et al. [35]) | 71.8±0.7 | 76.8±0.3 | 65.0±0.2 | 60.8±0.3 | 87.0±0.5 | **80.9±0.3** | 78.8±0.9 | 79.5±2.1 | 75.07 |
| MOLEBLEND (Yu et al. [73]) | **73.0±0.8** | **77.8±0.8** | **66.1±0.0** | **64.9±0.3** | 87.6±0.7 | 77.2±2.3 | 79.0±0.8 | **83.7±1.4** | **76.16** |
| MOLINTERACT ($\mathcal{L}_{\text{Simple}}$) | 67.2±3.9 | 76.4±0.2 | 64.5±0.2 | 62.5±0.4 | 86.1±0.4 | 78.6±0.4 | 78.6±0.8 | 82.4±1.7 | 74.52 |
| MOLINTERACT ($\mathcal{L}_{\text{All}}$) | 68.5±1.3 | 77.3±0.5 | 65.4±0.2 | 62.9±0.4 | **88.4±1.0** | 77.1±3.1 | **79.5±0.4** | 79.1±0.03 | 74.77 |

Table 3: Performance on MoleculeNet measured in ROC AUC. Higher is better.

### 4.2 3D Datasets - QM9

In QM9, we follow Thölke and Fabritiis [57] and finetune on 110K random molecules and use the remaining 10K and 10.8K molecules as validation and test sets, respectively. QM9 is a dataset of small molecules designed to test models' abilities to predicting various quantum and thermodynamic properties, which crucially depend on 3D information. Performance is measured in terms of MAE in order to determine how closely models can match DFT-level approximations of various quantum properties of small molecules. Per Table 1, both versions of MOLINTERACT exhibit a substantial lead in performance compared to baseline 3D pre-training methods, even without pre-training, demonstrating the effectiveness of the deep multimodal interaction layers. Including the 3D and 2D pre-training tasks, we see that MOLINTERACT's performance improves across the board, exceeding the most recent state-of-the-art by as much as 34% ($U$), further validating the use of deep interactions for improving the quality of learned features even with simple predictive SSL tasks.

### 4.3 2D Datasets - MoleculeNet

MoleculeNet is a set of 2D-only datasets with property prediction tasks ranging from toxicity prediction to drug reactivity. ROC-AUC is used in order to evaluate each model's ability to correctly determine these properties. Following [73], we report the mean and standard deviation across three random seeds and use the Bemis-Murcko scaffolds recommended in DeepChem [49]. Per Table 3, MOLINTERACT's performance on MoleculeNet is competitive with its multimodal and unimodal GNN-based peers, among which it ranks only behind MoleculeSDE [35] in average ROC AUC and even exceeds MoleculeSDE in 5/8 datasets. A possible explanation is that MoleculeSDE is tasked directly with reconstructing the original equilibrium state 3D conformer during pre-training, granting it highly detailed 3D knowledge which is especially useful in determining properties which may benefit from a precise understanding of the 3D geometry of a molecule like blood-brain barrier permeability [55]. It is also possible that the kind of 3D information learned by MOLINTERACT is not immediately useful for MoleculeNet tasks unlike quantum metrics, as shown in Table 4. Regarding other baselines, MOLINTERACT primarily falls behind non-GNN-based methods like GROVER [50] and MOLEBLEND [73] which use more powerful transformer backbones. We also see that MOLINTERACT with $\mathcal{L}_{\text{Simple}}$ performs worse than with $\mathcal{L}_{\text{All}}$, which is expected given its smaller ensemble of loss functions.

### 4.4 Ablation Studies

**3D Transfer Performance in QM9.** Table 4 shows performance on QM9 where only 2D graphs are provided to MOLINTERACT in order to test the degree of information transfer. Such an application may be valuable in cases where 3D structures are not consistently available, such as high-throughput

| Method | $\alpha$ | $\Delta\mathcal{E}$ | $\mathcal{E}_{\text{HOMO}}$ | $\mathcal{E}_{\text{LUMO}}$ | $\mu$ | $C_v$ | $R^2$ | ZPVE |
|---|---|---|---|---|---|---|---|---|
| PNA Corso et al. [7] | 0.3972 | 123.08 | 82.10 | 85.72 | 0.4133 | 0.1670 | 22.14 | 15.08 |
| GraphCL (You et al. [72]) | 0.3295 | 120.08 | 79.57 | 80.81 | 0.3937 | 0.1422 | 21.84 | 12.39 |
| AttrMask (You et al. [72]) | 0.3570 | 116.21 | 80.58 | 84.93 | 0.4626 | 0.1587 | 29.23 | 25.91 |
| GPT-GNN (Hu et al. [23]) | 0.3732 | 131.99 | 93.11 | 99.84 | 0.3975 | 0.1795 | 29.21 | 11.17 |
| GraphMVP (Liu et al. [34]) | 0.3227 | 101.84 | 68.62 | 70.23 | 0.3489 | 0.1287 | 17.03 | 7.96 |
| 3D InfoMax (Stärk et al. [54]) | 0.3268 | 101.71 | 68.96 | 69.51 | 0.3507 | 0.1306 | 17.39 | 7.96 |
| 3D-PGT (Wang et al. [60]) | 0.3121 | 101.53 | 68.24 | 69.73 | 0.3409 | 0.1217 | 16.89 | 7.92 |
| MOLINTERACT (2D) | **0.2145** | **86.49** | **59.77** | **57.55** | **0.3055** | **0.0830** | **12.03** | **5.11** |

Table 4: Performance on QM9 using 2D-only models to study the degree of 3D information transfer.

preliminary drug screening [54, 16]. Under this restriction, MOLINTERACT outperforms several 2D baselines benchmarked by [54], demonstrating substantially stronger 3D performance given only 2D graphs, suggesting a high degree of 3D-to-2D information transfer despite such a simple suite of pre-training tasks. This emphasizes the importance of the architecture of MOLINTERACT, showing the strength of the interaction layers even when one modality is missing.

**Impact of Pre-training Tasks and Interactor Types.** In Table 2, we investigate the impact of the pre-training tasks in the architecture of MOLINTERACT on performance in the QM9 dataset. "Only 3D tasks" refers to the method pre-trained only on interatomic distance, bond angle, and dihedral angle prediction. "Betweenness" refers to the $\mathcal{L}_{\text{All}}$ setting swapping centrality ranking loss with betweenness ranking loss. "Mean" and "Self-Attention Interactor" refer to the $\mathcal{L}_{\text{Simple}}$ setting except with the averaging operation and a separate single-head self-attention modules for $\phi$ layer, respectively. Finally, "3D structures only" refers to the setting where only 3D graphs are supplied to MOLINTERACT.

In these ablations, we observe a noticeable decline in performance across the board compared to the final version of MOLINTERACT. First, the "Only 3D tasks" ablation confirms that the 2D pre-training tasks indeed play a role in enhancing multimodal performance on downstream tasks even when they are not as directly related to properties such as $\Delta\mathcal{E}$, which are primarily reliant on 3D features. Next, the worse performance of betweenness centrality compared to eigenvector centrality suggests that the latter is more chemically meaningful. This is expected since eigenvector centrality is directly related to Laplacian eigenvector positional encodings [9, 29], which have been shown to enhance performance on molecular graphs by breaking the symmetries of WL-indistinguishable nodes [29]. The "Mean" and "Self-Attention" ablations show the superiority of a simple MLP-based interactor as a balance between a parameter-free and complex interactor. In our training runs, the self-attention-based interactor exhibited extensive over-fitting. Finally, given only 3D structures, we see that MOLINTERACT is competitive with contemporary methods, exceeds the current state-of-the-art in $\mu, C_v$, and ZPVE, and vastly outperforms a stock SchNet on all metrics by as much as 18% ($\Delta\mathcal{E}$), suggesting that multimodal information is successfully utilized during interaction.

# 5 Conclusion

In this work, we introduce MOLINTERACT an architectural approach to improving multimodal self-supervised learning that leverages deep interactions to fuse 2D and 3D representations of molecules. With this deep interaction mechanism, our method is able to access fine-grained cross-modal information without sacrificing rich embeddings from modality-specific backbones, allowing for more effective interplay between 2D and 3D information when paired with even a simple set of predictive pre-training tasks, achieving new state-of-the-art performance on benchmark datasets as a result and contributing to the growing field of multimodal property prediction for small molecules.

# References


[1] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987. ISSN 00029602, 15375390.

[2] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005. ISSN 0378-8733. doi: https://doi.org/10.1016/j.socnet.2004.11.008.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[4] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202.

[6] Nurendra Choudhary, Edward W Huang, Karthik Subbian, and Chandan K Reddy. An interpretable ensemble of graph and language models for improving search relevance in e-commerce. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 206–215, 2024.

[7] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems*, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[9] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[10] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, pages 1–8, 2022. doi: 10.1038/s42256-021-00438-4.

[11] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1): 35–41, 1977. ISSN 00380431.

[12] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

[13] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.

[14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[15] Jonathan Godwin, Michael Schaarschmidt, Alexander L Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple GNN regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations*, 2022.

[16] Christoph Gorgulla, AkshatKumar Nigam, Matt Koop, Süleyman Selim Çınaroğlu, Christopher Secker, Mohammad Haddadnia, Abhishek Kumar, Yehor Malets, Alexander Hasson, Minkai Li, Ming Tang, Roni Levin-Konigsberg, Dmitry Radchenko, Aditya Kumar, Minko Gehev, Pierre-Yves Aquilanti, Henry Gabb, Amr Alhossary, Gerhard Wagner, Alán Aspuru-Guzik,

Yurii S. Moroz, Konstantin Fackeldey, and Haribabu Arthanari. Virtualflow 2.0 - the next generation drug discovery platform enabling adaptive screens of 69 billion molecules. *bioRxiv*, 2023. doi: 10.1101/2023.04.25.537981.

[17] Etash Kumar Guha, Shlok Natarajan, Thomas Möllenhoff, Mohammad Emtiyaz Khan, and Eugene Ndiaye. Conformal prediction via regression-as-classification. In *International Conference on Learning Representations*, 2024.

[18] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[20] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022.

[21] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *CoRR*, abs/2005.00687, 2020.

[22] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

[23] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[24] Eric Inae, Gang Liu, and Meng Jiang. Motif-aware attribute masking for molecular graph pre-training, 2023.

[25] Xiaofei He Junying Li, Deng Cai. Learning graph-level representation for drug discoveryk. *arXiv preprint arXiv:1709.03741*, 2017.

[26] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, San Diega, CA, USA, 2015.

[27] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

[28] Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[29] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21618–21629. Curran Associates, Inc., 2021.

[30] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.

[31] Zhenyu Lei, Herun Wan, Wenqian Zhang, Shangbin Feng, Zilong Chen, Jundong Li, Qinghua Zheng, and Minnan Luo. BIC: Twitter bot detection with text-graph interaction and semantic consistency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[32] John W. Liebeschuetz, Jana Hennemann, Tjelvar S. G. Olsson, and Colin R. Groom. The good, the bad and the twisted: a survey of ligand geometry in protein crystal structures. *Journal of Computer-Aided Molecular Design*, 26:169 – 183, 2012.

[33] Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2022.

[35] Shengchao Liu, Weitao Du, Zhi-Ming Ma, Hongyu Guo, and Jian Tang. A group symmetric stochastic differential equation model for molecule multi-modal pretraining. In *International Conference on Machine Learning*, pages 21497–21526. PMLR, 2023.

[36] Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In *International Conference on Learning Representations*, 2023.

[37] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021.

[38] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2022.

[39] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[40] Shengjie Luo, Tianlang Chen, Yixian Xu, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. One transformer can understand both 2d & 3d molecular data. *arXiv preprint arXiv:2210.01765*, 2022.

[41] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap.

[42] Maho Nakata and Tomomi Shimazaki. Pubchemqc project: A large-scale first-principles electronic structure database for data-driven chemistry. *Journal of Chemical Information and Modeling*, 57(6):1300–1308, 2017. doi: 10.1021/acs.jcim.7b00083. PMID: 28481528.

[43] Nirmala Parisutham. How do centrality measures help to predict similarity patterns in molecular chemical structural graphs? *Artificial Intelligence Chemistry*, 1(2):100007, 2023. ISSN 2949-7477. doi: https://doi.org/10.1016/j.aichem.2023.100007.

[44] Trang Pham, Truyen Tran, Khanh Hoa Dam, and Svetha Venkatesh. Graph classification via deep learning with virtual nodes. *CoRR*, abs/1708.04357, 2017.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[46] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Sci Data*, 1:140022, 2014.

[47] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O. Anatole von Lilienfeld. Electronic spectra from TDDFT and machine learning in chemical space. *The Journal of Chemical Physics*, 143(8):084111, 08 2015. ISSN 0021-9606. doi: 10.1063/1.4928757.

[48] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35, 2022.

[49] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

[50] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying WEI, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12559–12571. Curran Associates, Inc., 2020.

[51] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. doi: 10.1021/ci300415d. PMID: 23088335.

[52] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[53] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9377–9388. PMLR, 18–24 Jul 2021.

[54] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Lió. 3D infomax improves GNNs for molecular property prediction. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20479–20502. PMLR, 17–23 Jul 2022.

[55] Claudia Suenderhauf, Felix Hammann, and Jörg Huwyler. Computational prediction of blood-brain barrier permeability using decision tree induction. *Molecules*, 17(9):10429–10445, 2012. ISSN 1420-3049. doi: 10.3390/molecules170910429.

[56] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations*, 2019.

[57] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[59] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3d molecular graphs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[60] Xu Wang, Huan Zhao, Wei-wei Tu, and Quanming Yao. Automated 3d pre-training for molecular property prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2419–2430, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599252.

[61] Xu Wang, Huan Zhao, Wei-wei Tu, and Quanming Yao. Automated 3d pre-training for molecular property prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2419–2430, 2023.

[62] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, pages 1–9, 2022. doi: 10.1038/s42256-022-00447-x.

[63] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[64] B.Yu. Weisfeiler and A.A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968. Translation from Russian to English by Grigory Ryabov.

[65] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9:513 – 530, 2017.

[66] Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z. Li. Mole-BERT: Rethinking pre-training graph neural networks for molecules. In *International Conference on Learning Representations*, 2023.

[67] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2020. doi: 10.1021/acs.jmedchem.9b00959. PMID: 31408336.

[68] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[69] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11548–11558. PMLR, 18–24 Jul 2021.

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

[71] Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems*, 2022.

[72] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc., 2020.

[73] Qiying Yu, Yudi Zhang, Yuyan Ni, Shikun Feng, Yanyan Lan, Hao Zhou, and Jingjing Liu. Multimodal molecular pretraining via modality blending. In *International Conference on Learning Representations*, 2024.

[74] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations*, 2021.

[75] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021.

[76] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *International Conference on Learning Representations*, 2023.

[77] Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Unified 2d and 3d pre-training of molecular representations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2626–2636, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539368.

[78] Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Unified 2d and 3d pre-training of molecular representations. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2626–2636, 2022.

# A Broader Impact

This work proposes a more effective method for developing multimodal representations of molecules for molecular property prediction. As such, it follows a line of work that has the potential to accelerate the drug and compound discovery process, making the development of new therapeutics easier and more cost-efficient. At the same time, there is potential for this work to be misused in order to aid in the development of compounds which negatively impact humanity in the form of harmful drugs, for example. We support the extensive usage of expert-guided control and regulation in order to steer the use of this technology and similar AI-assisted drug discovery techniques for social good.

# B Limitations

There are generally two major limitations of MOLINTERACT. First, like other multimodal approaches like MOLEBLEND [73], MOLINTERACT only takes into account the geometry of single 3D molecular conformations, such as those in QM9 [46] and QM8 [51] which are in equilibrium state. In this way, our method may not learn a comprehensive 3D representation of molecules and the wide spectrum of possible conformers which make up a valid 3D geometry for a given compound. However, other works [34, 35] tackle this problem by also tasking their architecture with generating 3D conformations directly, which is an SSL task which may be adopted for our architecture as well. Second, MOLINTERACT is limited in that even though the multimodal representations it learns are effective, it still finds optimal performance when both modalities are available to the model, suggesting a slight dependence on both modalities being provided to its interaction layers for downstream task performance. However, this may be remedied by experimenting with tasking the architecture with reproducing 2D/3D topology/geometry similar to MoleculeSDE [35] in order to make use of the deactivated unimodal branch during finetuning. Further, in the 2D-only case, MOLINTERACT already demonstrates an advance in the state-of-the-art when only 2D information is provide, and when 3D structures are involved, 2D structure is generally easily recoverable.

Figure 2 plots eigenvector centrality versus betweenness centrality on two different molecules from PCQM4Mv2. While the two centrality measures are similar, eigenvector centrality is able to highlight nodes which are not only towards the middle of the molecule but also parts of certain substructures, such as the central ring in the top molecule or the top ring in the bottom molecule. In other words, it appears eigenvector centrality is a better measure of "communities" in the molecular graph, assigning nodes in substructures more similar centralities. Being able to discern which atoms have higher centrality may be a useful proxy for learning higher-order structure in molecular graphs.

# C Hyperparameters and implementation details

Hyperparameters for pre-training on PCQM4Mv2 and finetuning on QM9, QM8, and MoleculeNet are in Table 5.

# D Pre-training computational cost

Table 6 shows wall times to pre-train various baseline pre-training methods as included in the appendix of MoleculeSDE [35]. Unfortunately, MOLEBLEND [73] does not yet have code available, and so we could not include it in this benchmark. The wall-times for all methods besides MOLINTERACT are reported from a machine using a single Nvidia V100 GPU. Due to access issues, we could not attain a V100 GPU, and so our reported time is from a SLURM cluster node equipped with an Nvidia A100 SXM4 80GB GPU. We recognize the generational gap in hardware, and so we hypothesize that MOLINTERACT will almost certainly train slower on a V100. Reducing the number of pre-training tasks will likely reduce pre-training wall time. Regrading memory requirements, pre-training MOLINTERACT under our settings required at least 30GB VRAM.

# E Performance on QM8

We also evaluate MOLINTERACT on 12 tasks from QM8 [47, 51]. QM8 is a smaller dataset than QM9 (20K vs 134K) with the task of predicting the electronic spectra of small organic molecules. Both 2D

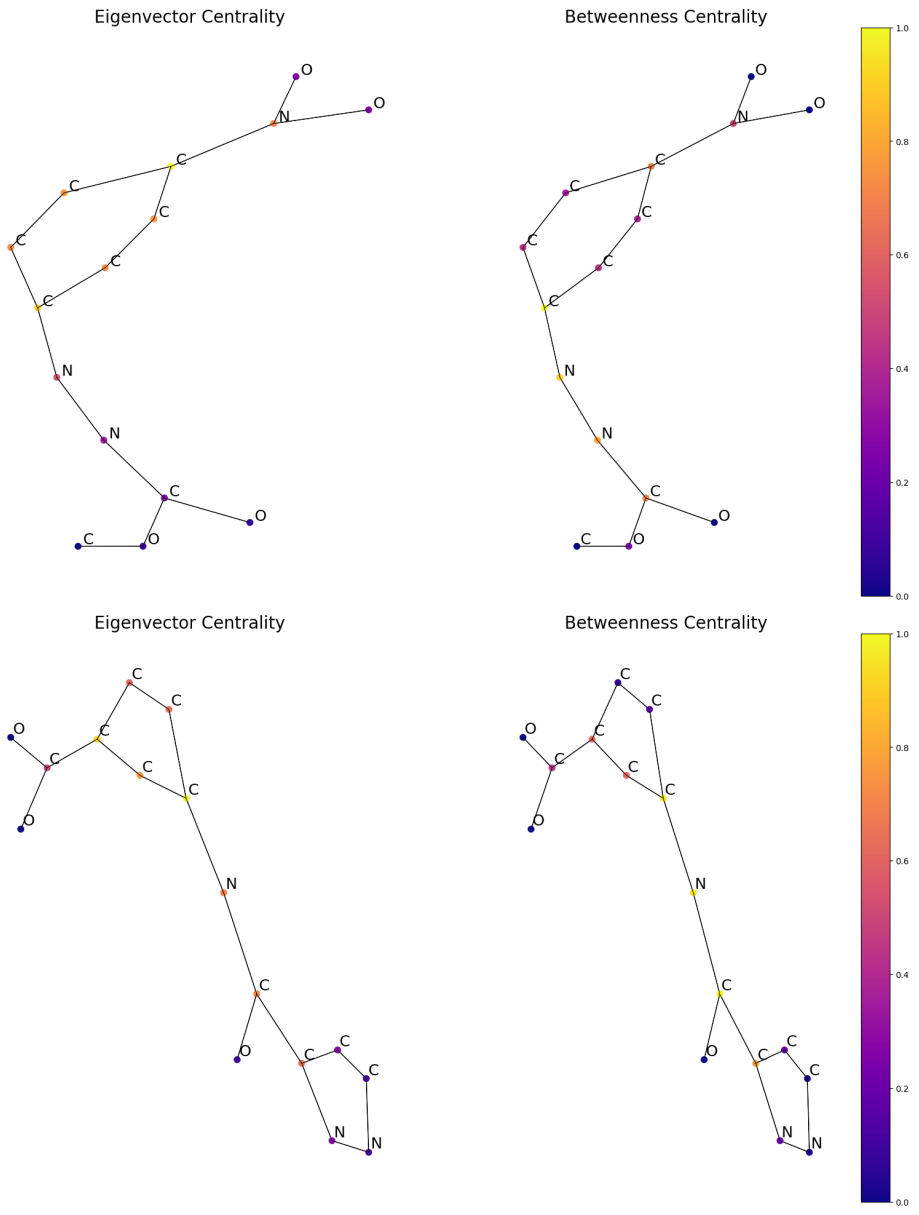



Figure 2: Comparing eigenvector and betweenness centrality on a molecule from PCQM4Mv2.

| Hyperparameter | PCQM4Mv2 | QM9 | QM8 | MoleculeNet |
|---|---|---|---|---|
| Optimizer | Adam [26] | Adam [26] | Adam [26] | Adam [26] |
| Initialization | Glorot uniform [14] | - | - | - |
| Learning rate scheduler | Cosine annealing [39] | Cosine annealing [39] | Cosine annealing [39] | Cosine annealing [39] |
| Adam betas | (0.9, 0.999) | (0.9, 0.999) | (0.9, 0.999) | (0.9, 0.999) |
| Batch size | 1024 | 128 | 128 | {32, 64, 128, 256} |
| Max learning rate | 1e-4 | 1e-4 | 1e-4 | {1e-3, 3e-4, 5e-4, 1e-5} |
| Min learning rate | 0 | 0 | 0 | 0 |
| Epochs | 50 | 1000 | 40 | {40, 60, 80, 100} |
| Weight decay | 0.0 | 0.0 | 0.0 | 0.0 |
| All embedding dimensions | 300 | 300 | 300 | 300 |
| Number of layers | 8 | 8 | 8 | 8 |
| Interactor activation | Swish | Swish | Swish | Swish |
| Interactor Batch norm | None | None | None | None |
| Interactor Layer norm | None | None | None | None |
| Number of SchNet filters | 128 | 128 | 128 | 128 |
| Number of SchNet Gaussians | 51 | 51 | 51 | 51 |
| GIN learnable $\epsilon$ | True | True | True | True |
| GIN Jumping knowledge | Last | Last | Last | {Last, Mean, Sum} |
| Dropout | 0.0 | 0.0 | 0.0 | {0.0, 0.1, 0.15} |

Table 5: Hyperparameters for pre-training (PCQM4Mv2) and finetuning (QM9, QM8, MoleculeNet)

| Pre-training algorithm | Min/epoch | GPU |
|---|---|---|
| AttrMask | 5.5 | Nvidia V100 32GB |
| ContextPred | 14 | Nvidia V100 32GB |
| InfoGraph | 6 | Nvidia V100 32GB |
| MolCLR | 10 | Nvidia V100 32GB |
| Distance Prediction | 6.7 | Nvidia V100 32GB |
| 3D InfoGraph | 7.5 | Nvidia V100 32GB |
| 3D InfoMax | 8.6 | Nvidia V100 32GB |
| GraphMVP | 11 | Nvidia V100 32GB |
| MoleculeSDE | 30 | Nvidia V100 32GB |
| MolInteract ($\mathcal{L}_{\text{All}}$) | 17.8 | Nvidia A100 80GB |

Table 6: Wall time to pre-train MolInteract compared to other pre-training algorithms.

| Pre-training method | $\alpha$ | $\Delta\mathcal{E}$ | $\mathcal{E}_{\text{HOMO}}$ | $\mathcal{E}_{\text{LUMO}}$ | $\mu$ | $C_v$ | $G$ | $H$ | $R^2$ | $U$ | $U_0$ | ZPVE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{\text{B}} + \mathcal{L}_{\text{Cent}}$ | 0.048 | 36.98 | 21.59 | 18.97 | 0.022 | 0.026 | 9.16 | 8.31 | 0.109 | 8.58 | 8.31 | 1.399 |
| $\mathcal{L}_{\text{D}} + \mathcal{L}_{\text{Cent}}$ | **0.046** | 36.54 | 20.85 | 18.33 | 0.023 | 0.026 | 9.42 | 8.64 | 0.160 | 8.35 | 8.59 | 1.400 |
| $\mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{Cent}}$ | 0.048 | 36.36 | 21.50 | 18.52 | 0.022 | 0.026 | 9.40 | 8.79 | 0.130 | 8.41 | 8.19 | 1.418 |
| $\mathcal{L}_{\text{B}} + \mathcal{L}_{\text{Edge}}$ | 0.047 | 36.34 | 21.28 | 18.52 | 0.023 | 0.027 | 9.55 | 8.87 | 0.134 | 9.15 | 8.81 | 1.441 |
| $\mathcal{L}_{\text{D}} + \mathcal{L}_{\text{Edge}}$ | 0.048 | 36.65 | 21.02 | 18.16 | 0.024 | 0.026 | 9.75 | 9.16 | 0.166 | 8.64 | 8.70 | 1.414 |
| $\mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{Edge}}$ | 0.048 | 36.96 | 21.48 | 18.50 | 0.022 | 0.026 | 9.26 | 8.72 | 0.102 | 8.46 | 8.33 | 1.401 |
| $\mathcal{L}_{\text{B}} + \mathcal{L}_{\text{SPD}}$ | 0.047 | 36.32 | 21.40 | 17.96 | **0.021** | **0.025** | 9.13 | 8.37 | 0.111 | 8.36 | **8.12** | 1.387 |
| $\mathcal{L}_{\text{D}} + \mathcal{L}_{\text{SPD}}$ | 0.047 | 36.48 | **20.83** | **18.00** | 0.022 | **0.025** | **9.00** | 8.55 | 0.161 | 8.54 | 8.44 | **1.307** |
| $\mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{SPD}}$ ($\mathcal{L}_{\text{Simple}}$) | 0.047 | **35.92** | 21.54 | 18.34 | **0.021** | **0.025** | 9.13 | 8.26 | **0.097** | **8.16** | 8.17 | 1.365 |

Table 7: Ablations of MolInteract on QM9 with different combinations of 2D and 3D loss terms.

| Method | Average MAE |
|---|---|
| D-MPNN (Yang et al. [70]) | 0.0190 ± 0.0001 |
| Attentive FP (Xiong et al. [67]) | 0.0179 ± 0.001 |
| N-Gram$_{RF}$ (Liu et al. [33]) | 0.0236±0.0006 |
| N-Gram$_{XGB}$ (Liu et al. [33]) | 0.0215±0.0005 |
| Pretrained GNN (Hu et al. [22]) | 0.0200±0.0001 |
| GROVER$_{base}$ (Rong et al. [50]) | 0.0218±0.0004 |
| GROVER$_{large}$ (Rong et al. [50]) | 0.0224±0.0003 |
| MolCLR (Wang et al. [62]) | 0.0178±0.0003 |
| ChemRL-GEM (Fang et al. [10]) | 0.0171±0.0001 |
| UniMol (Zhou et al. [76]) | **0.0156±0.0001** |
| MolInteract (base) | 0.0161 ± 0.0005 |
| MolInteract ($\mathcal{L}_{\text{Simple}}$) | 0.0158±0.0002 |
| MolInteract ($\mathcal{L}_{\text{All}}$) | <u>0.0157±0.0002</u> |

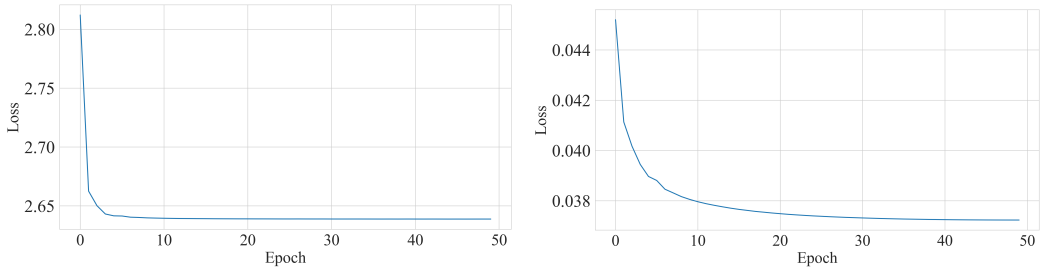
Table 8: Multi-task performance on QM8 measured in average MAE across 12 tasks. Lower is better.

and 3D structures are provided. Following Zhou et al. [76], we use an 80%/10%/10% scaffold split, and train for only 40 epochs. We compare with baselines reported by Zhou et al. [76] and report the average MAE of 12 tasks in a multi-task setting across three random seeds. Table 8 demonstrates the effectiveness of MolInteract, which not only outperforms pre-trained methods that leverage angle information such as Fang et al. [10], but also competes with Uni-Mol, a large 3D model pre-trained on over 200M molecular conformations, a dataset which is around 60 times larger and more diverse than PCQM4Mv2. This shows that MolInteract is able to use significantly less pre-training data, which may be attributable to its utilization of both 2D and 3D information from modality-specific encoders. Even when only using $\mathcal{L}_{\text{Simple}}$, MolInteract achieves comparable results.

# F Pre-training loss function behavior

In this section, we show loss curves for each loss function term in $\mathcal{L}_{\text{All}}$ during pre-training on PCQM4Mv2 for MolInteract. In Figures 3a, 5a, and 5b, we see that lower-order quantities such as interatomic distances and edge types, low loss and high accuracy are easily achieved by epoch 10 and begin to plateau thereafter. More complex quantities, such as bond angles and SPDs, exhibit similar elbow-shaped curves but saturate more slowly as shown in Figures 3b, 6a, and 6b. Finally, dihedral angle and eigenvector centrality classification are the hardest quantities to predict during pre-training, with both losses and accuracies improving much more slowly per Figures 4a, 4b, 7a, and 7b. This is expected given that the dihedral angle distribution in each molecule are complex in comparison [32], and learning to rank nodes by eigenvector centrality distills global structural patterns.

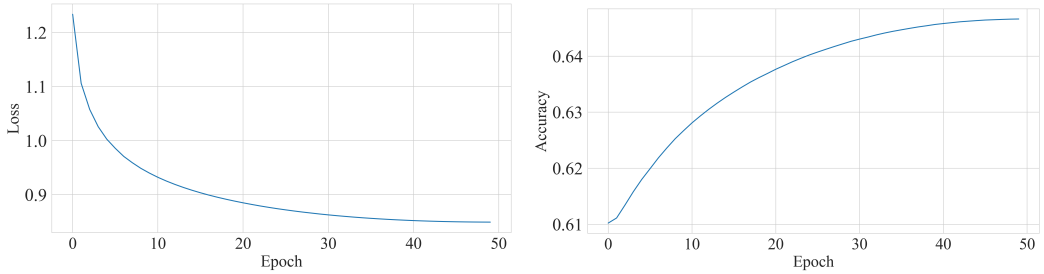
We also show comprehensive ablations for each combination of individual 2D and 3D pre-training tasks in Table 7. We see that $\mathcal{L}_{\text{Simple}}$ performs the best overall out of each combination of tasks, with $\mathcal{L}_{\text{D}}$ + $\mathcal{L}_{\text{SPD}}$ following closely. Notably, the highest metrics usually occur for losses which include $\mathcal{L}_{\text{SPD}}$, lending to the idea that shortest-path distances may contain the most useful 2D graph feature information. This is somewhat surprising since SPDs do not include edge types, missing important features such as whether an edge is a single or double bond, for example. A plausible explanation is that edge information is already incorporated into the node embeddings during 2D message-passing due to GINE's edge feature-aware convolution. Meanwhile, interatomic distance and dihedral angle prediction take turns as the most effective 3D tasks with bond angle regression lagging behind. While all three quantities are related to the overall equilibrium state of a molecule, a possible explanation for their performance difference is that interatomic distances give a more complete description of the overall 3D structure of a molecule, and dihedral angles may offer more fine-grained information than bond angles with more complex distributions.




(a) Interatomic distance regression loss.

(b) Bond angle regression loss.

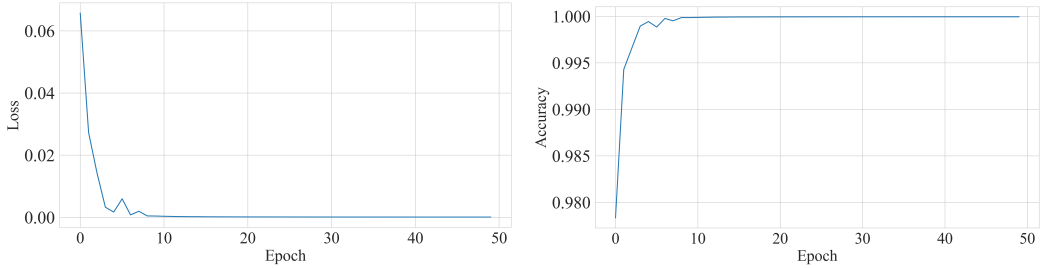
Figure 3: Interatomic distance and bond angle regression loss.




(a) Classification loss.

(b) Classification accuracy.

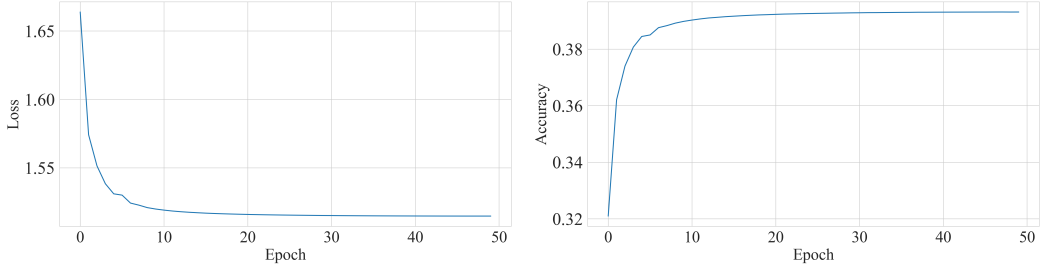
Figure 4: Dihedral angle classification loss and accuracy.




(a) Classification loss.

(b) Classification accuracy.

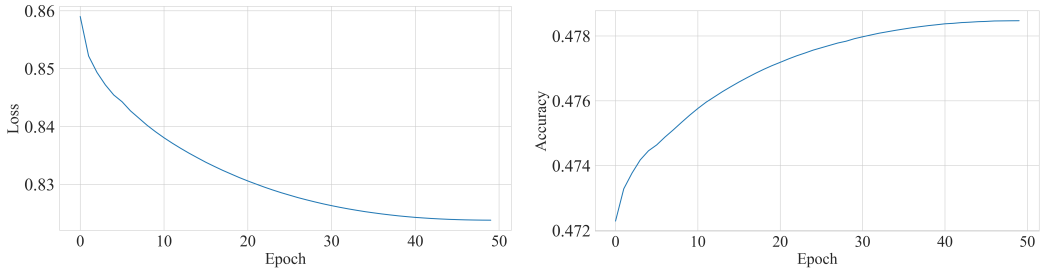
Figure 5: Edge type classification loss and accuracy.




(a) Classification loss.

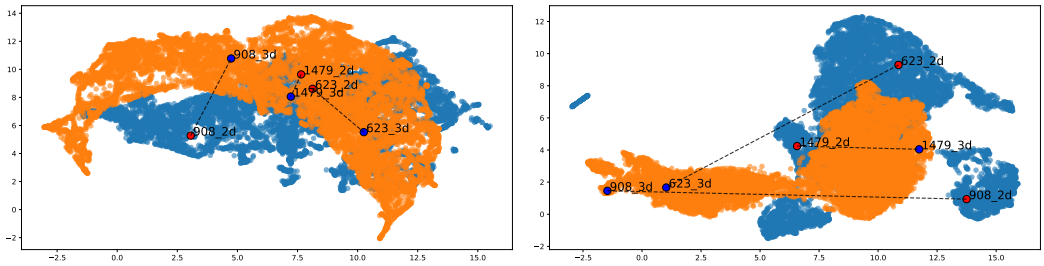
(b) Classification accuracy.

Figure 6: SPD classification loss and accuracy.




(a) Classification loss. (b) Classification accuracy.

Figure 7: Centrality ranking loss and accuracy.




(a) Embeddings from MOLINTERACT. (b) Embeddings from MoleculeSDE [35].

Figure 8: UMAP projection of QM9 molecule embeddings.

## G UMAP visualization

In Figure 8a and Figure 8b, we select 3 random test molecules from QM9 and plot them on their respective UMAP [41] projections. We see that MOLINTERACT exhibits more faithful multimodal molecule representations with 2D and 3D embeddings being more closely co-located than in the embedding space for MoleculeSDE. The 2D and 3D latent spaces of MOLINTERACT are therefore more well-aligned, contributing to its effectiveness in downstream tasks.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our work is focused on providing a new approach to multimodal molecular self-supervised learning, which is reflected in the main content of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Appendix B for a discussion of limitations of our method.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We report all the necessary experimental details of our method in Section 3.4 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include all code and data and instructions to reproduce our results in our supplementary material as a zipped archive.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We lit all training and test details needed to understand the results in Section 3.4 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We follow the existing literature on multimodal molecular SSL and report the mean and standard deviation for ROC AUC and MAE performance on MoleculeNet and QM8 from three random seeds. For QM9, we also follow the codebases from existing literature and report MAE from a single random seed (42).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include information on the amount of compute needed to run our experiments in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work did not involve human subjects, use compromising datasets, or engage in behavior that breaches the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in Appenxdix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not pose any risks of the kind described.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit the original authors of QM9, QM8, MoleculeNet, and all competing baselines.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include documented code and details for our model.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper did not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper did not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.