# 3D Interaction Geometric Pre-training for Molecular Relational Learning

**Namkyeong Lee**[1], **Yunhak Oh**[1], **Heewoong Noh**[1], **Gyoung S. Na**[1,2],
**Minkai Xu**[3], **Hanchen Wang**[3,4] , **Tianfan Fu**[5], **Chanyoung Park**[1*]
[1] KAIST [2] KRICT [3] Stanford University [4] Genentech [5] Nanjing University

## Abstract

Molecular Relational Learning (MRL) is a rapidly growing field that focuses on understanding the interaction dynamics between molecules, which is crucial for applications ranging from catalyst engineering to drug discovery. Despite recent progress, earlier MRL approaches are limited to using only the 2D topological structure of molecules, as obtaining the 3D interaction geometry remains prohibitively expensive. This paper introduces a novel 3D geometric pre-training strategy for MRL (3DMRL) that incorporates a 3D virtual interaction environment, overcoming the limitations of costly traditional quantum mechanical calculation methods. With the constructed 3D virtual interaction environment, 3DMRL trains 2D MRL model to learn the global and local 3D geometric information of molecular interaction. Extensive experiments on various tasks using real-world datasets, including out-of-distribution and extrapolation scenarios, demonstrate the effectiveness of 3DMRL, showing up to a 24.93% improvement in performance across 40 tasks. Our code is publicly available at https://github.com/Namkyeong/3DMRL.

## 1 Introduction

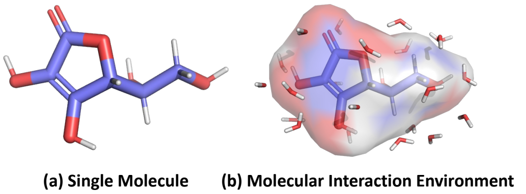


Figure 1: 3D geometry of (a) an individual molecule and (b) the molecular interaction environment.

Molecular relational learning (MRL) focuses on understanding the interaction dynamics between molecules and has gained significant attention from researchers thanks to its diverse applications [21, 32]. Despite recent advancements in MRL, previous works tend to ignore molecules' 3D geometric information and instead focus solely on their 2D topological structures. However, in molecular science, the 3D geometric information of molecules (Figure 1 (a)) is crucial for understanding and predicting molecular behavior across various contexts, ranging from physical properties [1] to biological functions [10]. This is particularly important in MRL, as geometric information plays a key role in molecular interactions by determining how molecules recognize, interact, and bind with one another in their interaction environment [35]. In traditional molecular dynamics simulations, explicit solvent models, which directly consider the detailed environment of molecular interaction, have demonstrated superior performance compared to implicit solvent models, which simplify the solvent as a continuous medium, highlighting the significance of modeling the complex geometries of interaction environments [47].

However, acquiring stereochemical structures of molecules is often very costly, resulting in limited availability of such 3D geometric information for downstream tasks [24]. Consequently, in the

[*] Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

domain of molecular property prediction (MPP), there has been substantial progress in injecting 3D geometric information to 2D molecular graph encoders during the pre-training phase, while utilizing only the 2D molecular graph encoder for downstream tasks [36, 25]. In contrast, compared to the MPP, pre-training strategies for MRL have been surprisingly underexplored, primarily due to the following two distinct challenges in modeling complex molecular interaction environments.

Firstly, interactions between molecules occur through complex geometry as they are chaotically distributed in space as shown in Figure 1 (b). Therefore, it is essential to consider not only each molecule's independent geometry but also their relative positions and orientations in space. This requirement further complicates the acquisition of geometric information, making it more challenging to obtain detailed 3D geometry of molecular interaction environments. Consequently, it is essential to model an interaction environment that can simulate molecular interactions based solely on the 3D geometry of the individual molecules.

Secondly, even after constructing the interaction environment, how to inject the geometry between molecules during interactions are not trivial. More specifically, while the global geometry of the interaction environment is essential for understanding overall interactions and system stability, the local geometry is also critical for examining localized interactions and precise molecular behaviors. Therefore, developing pre-training strategies that effectively capture the complementary global and local geometries between molecules and their interaction environment is essential.

To address these challenges, we introduce a novel 3D geometric pre-training strategy that is applicable to various MRL models by incorporating the 3D geometry of the interaction environment for molecules (3DMRL). Specifically, instead of relying on costly traditional quantum mechanical calculation methods to obtain interaction environments, we first propose a virtual interaction environment involving multiple molecules designed to simulate real molecular interactions. Then, during the pre-training stage, a 2D MRL model is trained to produce representations that are globally aligned with those of the 3D virtual interaction environment via contrastive learning. Additionally, the 2D MRL model is trained to predict the localized relative geometry between molecules within this virtual interaction environment, allowing the model to effectively learn fine-grained atom-level interactions between molecules. These two pre-training strategies enable the 2D MRL model to be pre-trained to understand the nature of molecular interactions, facilitating positive transfer to a wide range of downstream MRL tasks. In this paper, we make the following contributions:

- Rather than relying on costly traditional quantum mechanical calculation methods to obtain interaction geometry, we propose a virtual interaction geometry made up of multiple molecules to mimic the molecular interaction environment observed in real-world conditions.
- We propose pre-training strategies that allow the 2D MRL model to learn representations of the 3D interaction environment, capturing both its global and local geometries.
- We conduct extensive experiments across various MRL models pre-trained with 3DMRL on a range of MRL tasks, including *out-of-distribution* and *extrapolation* scenarios. These experiments demonstrate improvements of up to 24.93% compared to MRL methods trained from scratch, underscoring the versatility of 3DMRL (Section 5).

To the best of our knowledge, this is the first paper proposing pre-training strategies specifically designed for molecular relational learning.

## 2 Related Works

**Molecular Relational Learning.** Molecular Relational Learning (MRL) focuses on understanding the interaction dynamics between paired molecules. Delfos [23] employs recurrent neural networks combined with attention mechanisms to predict solvation-free energy, a key factor influencing the solubility of chemical substances, using SMILES string as input. Similarly, CIGIN [32] utilizes message-passing neural networks [11] along with a cross-attention mechanism to capture atomic representations for solvation-free energy prediction. In a different context, Joung et al. [17] use graph convolutional networks [18] to generate representations of chromophores and solvents, which are then used to predict various optical and photophysical properties of chromophores, essential for developing new materials with vibrant colors. Meanwhile, MHCADDI [4] introduces a co-attentive message passing network [38] designed for predicting drug-drug interactions (DDI), which aggregates information from all atoms within a pair of molecules, not just within individual molecules. Recently, CGIB [21] and CMRL [22] have introduced a comprehensive framework for MRL tasks, such as predicting solvation-free energy, chromophore-solute interactions, and drug-drug interactions. These models achieve this by identifying core functional groups involved in molecular interactions using information bottleneck and causal theory, respectively. However, prior studies have largely ignored molecules' 3D geometric information despite its well-established importance in comprehending various molecular properties.

**3D Pre-training for Molecular Property Prediction.** Recently, the molecular science community has shown increasing interest in pre-training machine learning models with unlabeled data, primarily due to the scarcity of labeled data for downstream tasks [22, 37, 44]. A promising approach in this area leverages molecules' inherent nature, which can be effectively represented as both 2D topological graphs and 3D geometric graphs. For instance, 3D Infomax [36] aims to enhance mutual information between 2D and 3D molecular representations using contrastive learning. GraphMVP [24] extends this concept by introducing a generative pre-training framework alongside contrastive learning. More recently, Noisy Nodes [46] and MoleculeSDE [25] have introduced methods to learn the 3D geometric distribution of molecules using a denoising framework, thereby uncovering the connection between the score function and the force field of molecules. Although the 3D structure of molecules has been effectively leveraged in pre-training for predicting single molecular properties, it remains surprisingly underexplored in the context of molecular relational learning (MRL). We provide more detailed explanations with the figure in Appendix A.

# 3 Preliminaries

## 3.1 Problem Statement

**Notations.** Given a molecule $g$, we first consider a 2D molecular graph, denoted as $g_{\text{2D}} = (\mathbf{X}, \mathbf{A})$, where $\mathbf{X} \in \mathbb{R}^{N \times F}$ represents the atom attribute matrix, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, with $\mathbf{A}_{ij} = 1$ if a covalent bond exists between atoms $i$ and $j$. Additionally, we define a 3D conformer as $g_{\text{3D}} = (\mathbf{X}, \mathbf{R})$, where $\mathbf{R} \in \mathbb{R}^{N \times 3}$ is the matrix of 3D coordinates, each row representing the spatial position of an individual atom.

**Task Description.** Given a 2D molecular graph pair $(g^1_{\text{2D}}, g^2_{\text{2D}})$ and 3D conformer pair $(g^1_{\text{3D}}, g^2_{\text{3D}})$, our goal is to pre-train the 2D molecular encoders $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ simultaneously with the virtual interaction geometry $g_{\text{vr}}$, derived from the 3D conformer pair. Then, the pre-trained 2D molecular encoders $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ are utilized for various MRL downstream tasks.

## 3.2 2D MRL Model Architecture

In this paper, we mainly focus on 1) the construction of virtual interaction geometry, and 2) pre-training strategies for MRL. Therefore, we employ existing model architectures for 2D MRL, i.e., CIGIN [32], which provides a straightforward yet effective framework for MRL as depicted in Figure 2 (a). Specifically, for each pair of 2D molecular graphs, denoted as $g^1_{\text{2D}}$ and $g^2_{\text{2D}}$, the graph neural networks (GNNs)-based molecular encoders $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ initially produce an atom embedding matrix for each molecule, formulated as:

$$\mathbf{E}^1 = f^1_{\text{2D}}\ (g^1_{\text{2D}}), \quad \mathbf{E}^2 = f^2_{\text{2D}}\ (g^2_{\text{2D}}), \tag{1}$$

where $\mathbf{E}^1 \in \mathbb{R}^{N^1 \times d}$ and $\mathbf{E}^2 \in \mathbb{R}^{N^2 \times d}$ are the atom embedding matrices for $g^1_{\text{2D}}$ and $g^2_{\text{2D}}$, containing $N^1$ and $N^2$ atoms, respectively. Next, we capture the interactions between nodes in $g^1_{\text{2D}}$ and $g^2_{\text{2D}}$ using an interaction matrix $\mathbf{I} \in \mathbb{R}^{N^1 \times N^2}$, defined by $\mathbf{I}_{ij} = \mathsf{sim}(\mathbf{E}^1_i, \mathbf{E}^2_j)$, where $\mathsf{sim}(\cdot, \cdot)$ represents the cosine similarity measure. Subsequently, we derive new embedding matrices $\tilde{\mathbf{E}}^1 \in \mathbb{R}^{N^1 \times d}$ and $\tilde{\mathbf{E}}^2 \in \mathbb{R}^{N^2 \times d}$ for each graph, reflecting their respective interactions. This is computed using $\tilde{\mathbf{E}}^1 = \mathbf{I} \cdot \mathbf{E}^2$ and $\tilde{\mathbf{E}}^2 = \mathbf{I}^\top \cdot \mathbf{E}^1$, where $\cdot$ denotes matrix multiplication. Here, $\tilde{\mathbf{E}}^1$ represents the node embeddings of $g^1_{\text{2D}}$ that incorporates the interaction information with nodes in $g^2_{\text{2D}}$, and similarly for $\tilde{\mathbf{E}}^2$. To obtain the final node embeddings, we concatenate the original and interaction-based embeddings for each graph, resulting in $\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1) \in \mathbb{R}^{N^1 \times 2d}$ and $\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2) \in \mathbb{R}^{N^2 \times 2d}$. Finally, we apply the Set2Set function [40] to compute the graph-level embeddings $\mathbf{z}^1_{\text{2D}}$ and $\mathbf{z}^2_{\text{2D}}$ for graph $g^1_{\text{2D}}$ and $g^2_{\text{2D}}$, respectively.

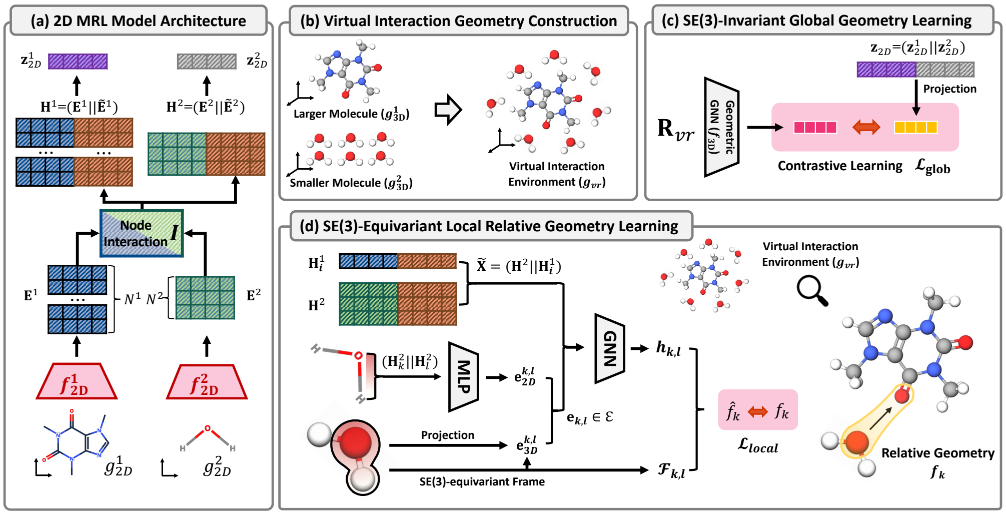

(a) 2D MRL Model Architecture
$z^1_{2D}$
$z^2_{2D}$
$H^1=(E^1||\tilde{E}^1)$
$H^2=(E^2||\tilde{E}^2)$
Node Interaction $I$
$E^1$
$N^1$ $N^2$
$E^2$
$f^1_{2D}$
$f^2_{2D}$
$g^1_{2D}$
$g^2_{2D}$
(b) Virtual Interaction Geometry Construction
Larger Molecule ($g^1_{3D}$)
Smaller Molecule ($g^2_{3D}$)
Virtual Interaction Environment ($g_{vr}$)
(c) SE(3)-Invariant Global Geometry Learning
$z_{2D}=(z^1_{2D}||z^2_{2D})$
Projection
$R_{vr}$
Geometric GNN ($f_{3D}$)
Contrastive Learning $\mathcal{L}_{glob}$
(d) SE(3)-Equivariant Local Relative Geometry Learning
$H^1_i$
$H^2$
$\tilde{X} = (H^2||H^1_i)$
$(H^2_k||H^2_l)$
MLP
$e^{k,l}_{2D}$
Projection
$e^{k,l}_{3D}$
$e_{k,l} \in \mathcal{E}$
SE(3)-equivariant Frame
GNN
$h_{k,l}$
$\mathcal{F}_{k,l}$
$\hat{f}_k$ $f_k$
$\mathcal{L}_{local}$
Virtual Interaction Environment ($g_{vr}$)
Relative Geometry $f_k$

Figure 2: Overall Framework: (a) 2D MRL model architecture (Section 3.2). (b) Virtual interaction geometry construction (Section 4.1). (c) $SE(3)$-Invariant Global Geometry Learning (Section 4.2.1). (d) $SE(3)$-Equivariant Local Relative Geometry Learning (Section 4.2.2).

# 4 Methodology

In this section, we introduce our method, named 3DMRL, a novel pre-training framework for MRL utilizing 3D geometry information. Specifically, in Section 4.1, we introduce how to construct the virtual interaction geometry that can be utilized instead of expensive calculation of real interaction geometry of molecules. Then, in Section 4.2, we present two complementary geometric pre-training strategies for the 2D MRL model to acquire representations aligned with the constructed virtual interaction geometry in both global and local perspectives. The overall framework is depicted in Figure 2, and the pseudocode for the entire framework is provided in Appendix F.

## 4.1 Virtual Interaction Geometry Construction

While the 3D geometry of molecules plays a significant role in predicting molecular properties, acquiring this information involves a trade-off between cost and accuracy. For example, RDKit's ETKDG algorithm [20] is fast but less accurate. In contrast, the widely adopted metadynamics method, CREST [12], achieves a more balanced compromise between speed and accuracy, yet still requires around 6 hours to process a drug-like molecule. This challenge is even more pronounced in molecular interaction systems, which necessitates not just the geometry of individual molecules but also the relative spatial arrangements between multiple molecules [6]. Moreover, an appropriate initial geometry of a molecular interaction system is highly dependent on individual molecules in the system [27]. For this reason, a data-agnostic process for generating the initial molecular geometry is crucial for flexible and robust representation learning on molecular interaction systems.

Drawing inspiration from the explicit solvent models used in traditional molecular dynamics simulations [9], we propose a one-to-many geometric configuration that involves a relatively larger molecule $g^1_{\text{3D}}$, determined based on its radius, surrounded by multiple smaller molecules $g^2_{\text{3D}}$ as shown in Figure 2 (b) [28]. Specifically, for a given conformer pair $(g^1_{\text{3D}} = (\mathbf{X}^1, \mathbf{R}^1), g^2_{\text{3D}} = (\mathbf{X}^2, \mathbf{R}^2))$, we create an environment by arranging the smaller molecules $(g^{2,1}_{\text{3D}}, \ldots, g^{2,i}_{\text{3D}}, \ldots, g^{2,n}_{\text{3D}})$ around a centrally placed larger molecule $g^1_{\text{3D}}$ as follows:

**[Step 1] Select Target Atoms in the Larger Molecule.** We start by randomly selecting $n$ atoms from the larger molecule $g^1_{\text{3D}}$ that are not part of any aromatic ring. This choice is based on the fact that aromatic rings are more stable and less likely to engage in chemical reactions.

**[Step 2] Positioning the Smaller Molecules.** Each smaller molecule in $(g^{2,1}_{\text{3D}}, \ldots, g^{2,i}_{\text{3D}}, \ldots, g^{2,n}_{\text{3D}})$ is then placed close to one of the $n$ selected atoms in the larger molecule $g^1_{\text{3D}}$. This positioning is achieved by transiting and rotating the original 3D coordinates $\mathbf{R}^2$ of the smaller molecule $g^2_{\text{3D}}$, following the method widely employed in computational chemistry [19].

- **[Step 2-1] Determine Transition Direction.** For flexible and robust molecular relational learning, we follow a widely used strategy that samples initial geometries from parameterized stochastic processes [42]. Specifically, we generate a normalized random Gaussian noise vector $\varepsilon$ (with a norm of 1), which will be used to set the direction for the transition. We then scale this direction vector $\varepsilon$ by the radius of the smaller molecule, $r^2$, to establish the transition distance.

- **[Step 2-2] Transit and Rotate to the New Position.** The new 3D coordinates for each smaller molecule are determined using the formula $\mathbf{R}^{2,i} = \mathbf{R}^2 + \varepsilon_i * r^2 + \mathbf{R}^1_i$, where $\mathbf{R}^1_i \in \mathbb{R}^3$ represents the 3D position of the $i$-th selected atom in the larger molecule $g^1_{\text{3D}}$. This operation is performed through broadcasting, meaning $\mathbf{R}^1_i$ and $\varepsilon_i$ are added to each row of $\mathbf{R}^2$. Additionally, we apply a random rotation matrix to rotate the small molecule after its transition. This transition and rotation operations ensure that each smaller molecule is positioned close to its corresponding selected atom on the larger molecule, simulating a realistic interaction environment.

**[Step 3] Constructing Virtual Interaction Geometry.** After positioning each smaller molecule $g^{2,i}_{\text{3D}}$ near the $i$-th selected atom in the larger molecule $g^1_{\text{3D}}$, we compile all the 3D coordinates to form a unified virtual environment $g_{\text{vr}}$. This process involves combining the coordinate matrix $\mathbf{R}^1$ of the larger molecule $g^1_{\text{3D}}$, with the transited coordinates $(\mathbf{R}^{2,1}, \ldots, \mathbf{R}^{2,i}, \ldots, \mathbf{R}^{2,n})$ of the smaller molecules $(g^{2,1}_{\text{3D}}, \ldots, g^{2,i}_{\text{3D}}, \ldots, g^{2,n}_{\text{3D}})$, resulting in $\mathbf{R}_{\text{vr}} = (\mathbf{R}^1 \| \mathbf{R}^{2,1} \| \ldots \| \mathbf{R}^{2,i} \| \ldots \| \mathbf{R}^{2,n}) \in \mathbb{R}^{(N^1 + n \cdot N^2) \times 3}$. Additionally, it involves concatenating all the atom attribute matrices to form $\mathbf{X}_{\text{vr}} = (\mathbf{X}^1 \| \mathbf{X}^2 \| \ldots \| \mathbf{X}^2) \in \mathbb{R}^{(N^1 + n \cdot N^2) \times F}$, thereby defining the virtual interaction geometry as $g_{\text{vr}} = (\mathbf{X}_{\text{vr}}, \mathbf{R}_{\text{vr}})$. Note that multiple small molecules share the same attribute matrix $\mathbf{X}^2$, since we use the atom attribute irrelevant to the atomic coordinates.

Note that such randomized configurations of interaction environment is a well-established strategy in molecular simulations. For instance, protein–ligand docking protocols (e.g., Rosetta) often initialize ligands in random orientations relative to the protein before searching for binding modes. Similarly, Monte Carlo insertion methods like Widom's test-particle approach randomly insert solvent molecules to explore solute–solvent configurations without bias. Moreover, while we construct the virtual interaction geometry (**Step 1** to **Step 3**) at each epoch during the pre-training phase, the virtual environment can be generated in real time because transition and rotation are matrix operations. Therefore, we argue that our approach allows efficient sampling over a wide range of distances and orientations while remaining physically sound: in the limit of sufficient sampling, no unphysical configuration is favored, and the process mimics the early stages of solvation when solvent molecules approach from arbitrary directions. In Section 5 and Appendix E.4, we analyze the environment in various aspects, justifying the proposed approach for constructing a virtual interaction environment.

## 4.2 Pre-training Strategies

Once the virtual interaction geometry is established, we pre-train the 2D MRL model using two complementary geometry learning strategies: $SE(3)$-invariant global geometry learning (Section 4.2.1) and $SE(3)$-equivariant local relative geometry learning (Section 4.2.2).

### 4.2.1 $SE(3)$-Invariant Global Geometry Learning

Given a paired 2D molecular graphs $(g^1_{\text{2D}}, g^2_{\text{2D}})$ and its corresponding 3D virtual interaction geometry $g_{\text{vr}}$, we first encode them with a 2D MRL model, and a geometric deep learning model, respectively. For 2D molecular graphs, we compute the molecule-level representations, $\mathbf{z}^1_{\text{2D}}$ and $\mathbf{z}^2_{\text{2D}}$, for each molecule $g^1_{\text{2D}}$ and $g^2_{\text{2D}}$, respectively, as outlined in the Section 3.2. Following this, we derive the 2D interaction representation $\mathbf{z}_{\text{2D}}$, by concatenating these two representations, i.e., $\mathbf{z}_{\text{2D}} = (\mathbf{z}^1_{\text{2D}} || \mathbf{z}^2_{\text{2D}})$. On the other hand, to encode the 3D virtual interaction geometry $g_{\text{vr}} = (\mathbf{X}_{\text{vr}}, \mathbf{R}_{\text{vr}})$, we use geometric GNNs $f_{\text{3D}}$ that output $SE(3)$ invariant [7] representations $\mathbf{z}_{\text{3D}}$ given the coordinates of atoms $\mathbf{R}_{\text{vr}}$ in virtual interaction geometry [34], i.e., $\mathbf{z}_{\text{3D}} = f_{\text{3D}}(\mathbf{R}_{\text{vr}})$. Then, as shown in Figure 2 (c), we align the 2D interaction representation $\mathbf{z}_{\text{2D}}$ and the 3D geometry representation $\mathbf{z}_{\text{3D}}$ via Normalized temperature-scaled cross entropy loss [3] as follows:

$$\mathcal{L}_{\text{glob}} = -\frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \left[ \log \frac{e^{\texttt{sim}(\mathbf{z}_{\text{2D},i}, \mathbf{z}_{\text{3D},i})/\tau}}{\sum_{k=1}^{N_{\text{batch}}} e^{\texttt{sim}(\mathbf{z}_{\text{2D},i}, \mathbf{z}_{\text{3D},k})/\tau}} + \log \frac{e^{\texttt{sim}(\mathbf{z}_{\text{3D},i}, \mathbf{z}_{\text{2D},i})/\tau}}{\sum_{k=1}^{N_{\text{batch}}} e^{\texttt{sim}(\mathbf{z}_{\text{3D},i}, \mathbf{z}_{\text{2D},k})/\tau}} \right].$$

where $\texttt{sim}(\cdot, \cdot)$ represents cosine similarity, $\tau$ denotes the temperature hyperparameter, and $N_{\text{batch}}$ refers to the number of pairs within a batch. By training the 2D MRL model to output interaction representations that align with the 3D interaction geometry, the model effectively learns the overall global geometry of molecular interactions during the pre-training phase.

### 4.2.2 $SE(3)$-Equivariant Local Relative Geometry Learning

Beyond the overall global geometry of interaction, it is essential to learn about the intermolecular local relative geometry between molecules during molecular interactions, as their localized relative geometry governs how molecules interact in various environments. To achieve this, we propose pre-training the 2D MRL model to learn local relative geometry by predicting the 3D geometry of the paired molecule, specifically by training a smaller 2D molecule encoder to predict the geometry of the larger molecule. However, predicting relative geometry from a 2D representation is challenging because the prediction must adhere to the physical properties of the molecule, specifically being equivariant to rotations and transitions in 3D Euclidean space, also known as *SE(3)-equivariance* [7]. To address this, we propose predicting the relative geometry between molecules by utilizing local frame [5], which allows for flexible conversion between invariant and equivariant features.

More specifically, given the position $\mathbf{R}^{2,i}$ of the $i$-th small molecule $g_{\text{3D}}^{2,i}$ in the constructed virtual interaction geometry, we first define an orthogonal local frame $\mathcal{F}_{k,l}$ between atoms $k$ and $l$ within molecule $g_{\text{3D}}^{2,i}$ as follows:

$$\mathcal{F}_{k,l} = \left( \frac{\mathbf{r}_k - \mathbf{r}_l}{||\mathbf{r}_k - \mathbf{r}_l||}, \frac{\mathbf{r}_k \times \mathbf{r}_l}{||\mathbf{r}_k \times \mathbf{r}_l||}, \frac{\mathbf{r}_k - \mathbf{r}_l}{||\mathbf{r}_k - \mathbf{r}_l||} \times \frac{\mathbf{r}_k \times \mathbf{r}_l}{||\mathbf{r}_k \times \mathbf{r}_l||} \right), \tag{2}$$

where $\mathbf{r}_k \in \mathbb{R}^3$ and $\mathbf{r}_l \in \mathbb{R}^3$ indicate the position of atoms $k$ and $l$ in constructed virtual interaction geometry, respectively. For simplicity, please note that we will omit the molecule index $i$ in the notation from here. With the established local frame, we derive the invariant 3D feature for the edge between atoms $k$ and $l$ by projecting their coordinates into the local frame, i.e., $\mathbf{e}_{\text{3D}}^{k,l} = \text{Projection}_{\mathcal{F}_{k,l}}(\mathbf{r}_k, \mathbf{r}_l) \in \mathbb{R}^d$. Additionally, we obtain the 2D invariant edge feature between atoms $k$ and $l$ by concatenating the respective features from the 2D molecular graph, i.e., $\mathbf{e}_{\text{2D}}^{k,l} = \text{MLP}(\mathbf{H}_k^2 || \mathbf{H}_l^2) \in \mathbb{R}^d$. Now that we have both invariant 2D and 3D features, we can derive the final invariant edge feature $\mathbf{e}^{k,l}$ by combining these invariant edge features as follows:

$$\mathbf{e}_{k,l} = \mathbf{e}_{\text{2D}}^{k,l} + \mathbf{e}_{\text{3D}}^{k,l}. \tag{3}$$

We define the edge feature set $\mathcal{E}$, which includes $\mathbf{e}_{k,l}$ for every possible pair of atoms.

With the invariant final edge feature set $\mathcal{E}$, we can further process the small molecule information through GNNs to predict the geometry of the larger molecule. To achieve this, we first obtain the atom features specific to the $i$-th small molecule by concatenating the $i$-th atom representation of the larger molecule (to which the $i$-th small molecule is assigned) with each atom representation of the small molecule, i.e., $\tilde{\mathbf{X}} = (\mathbf{H}^2 || \mathbf{H}_i^1) \in \mathbb{R}^{N^2 \times 4d}$ using broadcasting. This approach allows the model to learn a more precise geometry by incorporating the features of the assigned atom in the larger molecule. Next, with the edge feature set $\mathcal{E}$ and the atom feature $\tilde{\mathbf{X}}$, we derive the final edge representation $\mathbf{h}_{k,l}$ through multiple GNN layers, represented as $\mathbf{h}_{k,l} = \text{GNN}(\tilde{\mathbf{X}}, \mathcal{E})$. Finally, we determine the relative geometry $\hat{f}_k$ between the atom $k$ of the small molecule and the central larger molecule by combining the final invariant edge representation $\mathbf{h}_{k,l}$ with our $SE(3)$-equivariant frame $\mathcal{F}_{k,l}$ as follows:

$$\hat{f}_k = \sum_l \mathbf{h}_{k,l} \odot \mathcal{F}_{k,l}, \tag{4}$$

where $\odot$ indicates element-wise product. This approach guarantees our predicted relative geometry $\hat{f}_k$ to be $SE(3)$-equivariant. Then, we calculate the relative geometry prediction loss as follows:

$$\mathcal{L}_{\text{local}} = \frac{1}{n \cdot N^2} \sum_{i=1}^{n} \sum_{k=1}^{N^2} ||f_k^i - \hat{f}_k^i||_2^2, \tag{5}$$

where $f_k^i$ represents the ground truth relative geometry between the larger molecule and the $k$-th atom of the $i$-th small molecule. We define the relative geometry $f_k^i$ as the direction between the $k$-th atom of the $i$-th small molecule and the $i$-th atom of the larger molecule to which the small molecule is attached, i.e., $f_k^i = (\mathbf{R}_k^{2,i} - \mathbf{R}_i^1)/||\mathbf{R}_k^{2,i} - \mathbf{R}_i^1||_2$. Note that $\mathcal{L}_{\text{local}}$ is calculated for every molecule pair in the batch, although we have omitted this notation for simplicity.

Finally, we pre-train the 2D MRL model by jointly optimizing two proposed losses, i.e., $SE(3)$-invariant global geometry loss and $SE(3)$-equivariant local relative geometry loss, as follows:

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}_{\text{glob}} + \alpha \cdot \mathcal{L}_{\text{local}}, \tag{6}$$

where $\alpha$ is a hyperparameter that determines the trade-off between the global geometry loss and the local geometry loss. After task-agnostic pre-training, the 2D molecular encoders $f_{\text{2D}}^1$ and $f_{\text{2D}}^2$ are fine-tuned for specific downstream tasks where access to 3D geometric information is limited.

### 4.3 Discussion

While we define local relative geometry for learning fine-grained interactions between molecules, we can view local relative geometry as an *interaction force* between molecules. This provides a physically motivated supervision signal rooted in classical intermolecular forces, many of which are central and act along the internuclear axis. For example, van der Waals interactions (described by the Lennard-Jones potential) exhibit repulsive or attractive forces directed along this axis. At short distances, repulsion dominates and aligns directly outward between nuclei.

This supervision scheme serves as a central-force approximation, consistent with classical force fields, and offers a lightweight surrogate for full force labels, which would require costly quantum chemistry or MD simulations. Notably, SchNet[34] demonstrated that even approximate force signals improve learning of molecular interactions. Our direction-based supervision enables the model to learn geometric features like hydrogen bond alignment or steric repulsion trajectories in an SE(3)-equivariant manner.

Since solvent atoms are placed near specific solute atoms, the dominant interaction direction aligns with the interatomic vector, making it a reasonable proxy for the net force axis. Thus, the unit direction vector serves as a pseudo-force label, conveying the primary interaction axis and encouraging the model to encode directionality of intermolecular interactions.

## 5 Experiments

### 5.1 Experimental Setup

**Downstream Tasks.** Following a prior study [21], we employ **ten** datasets to comprehensively evaluate the performance of 3DMRL on two tasks: 1) molecular interaction prediction, and 2) drug-drug interaction (DDI) prediction. For the molecular interaction prediction task, we utilize the **Chromophore** dataset [16], which pertains to three optical properties of chromophores, along with five other datasets related to the solvation free energy of solutes: **MNSol** [26], **FreeSolv** [29], **CompSol** [30], **Abraham** [13], and **CombiSolv** [39]. In the Chromophore dataset, we focus on the maximum absorption wavelength (**Absorption**), maximum emission wavelength (**Emission**), and excited state lifetime (**Lifetime**) properties. For the DDI prediction task, we use two datasets: **ZhangDDI** [48] and **ChChMiner** [49], both of which contain labeled DDI data. We provide further details on pre-training and downstream task datasets in Appendix B.1 and B.2, respectively.

**Baseline methods.** We validate the effectiveness of 3DMRL by using it to enhance various recent state-of-the-art molecular relational learning methods, including **MPNN** [11], **AttentiveFP** [43], **CIGIN** [32], **CGIB** [21], and **CGIB$_{\text{Cont}}$** [21]. Additionally, we compare our proposed pre-training framework, 3DMRL, with recent molecular pre-training approaches that aim to learn 3D structure of individual molecules, such as **3D Infomax** [36], **GraphMVP** [24], and **MoleculeSDE** [25]. It is important to note that these approaches involve pre-training a single encoder for molecular property prediction (**MPP Pre-training** in Table 2), whereas our work is pioneering in training two separate encoders simultaneously during pre-training for molecular relational learning (**MRL Pre-training** in Table 2). For the baseline methods, we use the original authors' code and conduct the experiments in the same environment as 3DMRL to ensure a fair comparison. Moreover, while we choose to mainly

Table 1: Performance improvement in molecular interaction tasks across different models with our proposed pre-training strategy (RMSE) (↓). We conduct 15 independent runs for each model and report their mean along with the standard deviation (in parentheses). Colors indicate the performance improvement compared to the models trained from scratch.

| **Model** | **Chromophore** | | | **MNSol** | **FreeSolv** | **CompSol** | **Abraham** | **CombiSolv** |
|---|---|---|---|---|---|---|---|---|
| | **Absorption** | **Emission** | **Lifetime** | | | | | |
| MPNN | 22.00 (0.30) | 26.34 (0.41) | 0.789 (0.021) | 0.643 (0.005) | 1.127 (0.110) | 0.420 (0.018) | 0.640 (0.008) | 0.614 (0.031) |
| + 3DMRL | 19.96 (0.12) | 25.21 (0.31) | 0.753 (0.018) | 0.609 (0.008) | 1.068 (0.087) | 0.377 (0.020) | 0.550 (0.051) | 0.599 (0.025) |
| Improvement | 9.27% | 4.29% | 4.56% | 5.28% | 5.24% | 10.24% | 14.06% | 2.44% |
| AttentiveFP | 22.86 (0.30) | 28.70 (0.23) | 0.871 (0.010) | 0.570 (0.021) | 1.019 (0.070) | 0.350 (0.008) | 0.426 (0.042) | 0.471 (0.028) |
| + 3DMRL | 22.80 (0.61) | 28.54 (1.97) | 0.784 (0.013) | 0.562 (0.031) | 0.901 (0.059) | 0.271 (0.009) | 0.378 (0.027) | 0.448 (0.011) |
| Improvement | 0.26% | 0.55% | 9.99% | 1.40% | 11.57% | 22.57% | 11.26% | 4.88% |
| CIGIN | 19.66 (0.69) | 25.84 (0.23) | 0.821 (0.017) | 0.582 (0.022) | 0.958 (0.116) | 0.369 (0.018) | 0.421 (0.018) | 0.464 (0.002) |
| + 3DMRL | 18.00 (0.17) | 24.21 (0.09) | 0.729 (0.014) | 0.528 (0.019) | 0.839 (0.105) | 0.277 (0.006) | 0.371 (0.031) | 0.435 (0.006) |
| Improvement | 8.44% | 6.30% | 11.20% | 9.28% | 12.42% | 24.93% | 11.87% | 6.25% |
| CGIB | 18.37 (0.35) | 24.52 (0.25) | 0.808 (0.015) | 0.562 (0.008) | 0.876 (0.037) | 0.321 (0.002) | 0.404 (0.037) | 0.448 (0.008) |
| + 3DMRL | 17.93 (0.35) | 23.92 (0.29) | 0.733 (0.009) | 0.538 (0.020) | 0.842 (0.078) | 0.274 (0.002) | 0.370 (0.027) | 0.442 (0.015) |
| Improvement | 2.40% | 5.90% | 9.28% | 4.27% | 3.88% | 14.64% | 8.42% | 1.33% |
| CGIB$_{\text{Cont}}$ | 18.59 (0.24) | 24.68 (0.49) | 0.803 (0.019) | 0.561 (0.012) | 0.897 (0.098) | 0.333 (0.005) | 0.404 (0.039) | 0.452 (0.015) |
| + 3DMRL | 17.90 (0.17)** | 23.94 (0.24) | 0.720 (0.020) | 0.524 (0.018)* | 0.863 (0.075) | 0.284 (0.007) | 0.372 (0.021) | 0.441 (0.022) |
| Improvement | 3.71% | 3.00% | 10.33% | 6.59% | 3.79% | 14.71% | 7.92% | 2.43% |

compare 2D encoder pre-training approach, we also compare 3D encoder pre-training approaches [15, 31, 8] in Appendix E.5. We provide more details on the compared methods in Appendix C.

**Evaluation protocol.** Following Pathak et al. [32], for the molecular interaction prediction task, we evaluate the models under a 5-fold cross-validation scheme. The dataset is randomly split into 5 subsets and one of the subsets is used as the test set, while the remaining subsets are used to train the model. A subset of the test set is selected as the validation set for hyperparameter selection and early stopping. We repeat 5-fold cross-validation three times (i.e., 15 runs in total) and report the accuracy and standard deviation of the repeats. For the DDI prediction task [21], we conduct experiments on two different *out-of-distribution* scenarios, namely **molecule split** and **scaffold split**. For the **molecule split**, the performance is evaluated when the models are presented with new molecules not included in the training dataset. In the **scaffold split** setting [14], just like in the molecule split, molecules corresponding to scaffolds that were not seen during training will be used for testing. For both splits, we repeat 5 independent experiments with different random seeds on split data, and report the accuracy and the standard deviation of the repeats. In both scenarios, we split the data into training, validation, and test sets with a ratio of 60/20/20%. We provide details on evaluation protocol, model implementation, and model training in Section D.

### 5.2 Experimental Results

We begin by comparing each model architecture trained from scratch with the same architecture pre-trained using our proposed strategy, referred to as +3DMRL in Table 1. We have the following observations: **1)** 3DMRL obtains consistent improvements over the base graph neural networks in all 40 tasks (across various datasets and neural architectures), achieving up to 24.93% relative reduction in RMSE. While the paper is written based on CIGIN for better understanding in Section 3.2, we could observe performance improvements not only in CIGIN but also in various other model architectures, demonstrating the versatility of proposed pre-training strategies. We further demonstrate how our pre-training strategies are adopted to various model architectures in Appendix C.

Additionally, we compare our pre-training strategies with recent molecular pre-training approaches proposed for molecular property prediction (MPP) of a single molecule. Table 2 (a) and (b) show the results for the molecular interaction prediction task, and the drug-drug interaction (DDI) task, respectively. As these approaches are originally designed for single molecules, we first pre-train the GNNs using each strategy, then incorporate the pre-trained GNNs into the CIGIN architecture and fine-tune them for various MRL downstream tasks. We have the following observations: **2)** Although MPP pre-training methods have demonstrated success in molecular property prediction in prior studies, they did not yield satisfactory results in molecular relational learning tasks and, in some cases, even resulted in negative transfer. This highlights the need for creating specialized pre-training strategies tailored to MRL tasks. We further demonstrate the MPP pre-training strategy with a large-scale dataset still performs worse than 3DMRL in Appendix E.1. **3)** On the other hand, pre-training

Table 2: Performance of CIGIN model on (a) molecular interaction tasks using different pre-training strategies (RMSE) (↓) and (b) out-of-distribution DDI tasks using different pre-training strategies (AUROC) (↑). For each dataset, we highlight the best method **in bold**.

| Strategy | (a) Molecular Interaction Tasks (RMSE ↓) | | | | | | | | (b) Drug-Drug Interaction Task (AUROC ↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chromophore | | | MNSol | FreeSolv | CompSol | Abraham | CombiSolv | (c) Molecule Split | | (d) Scaffold Split | |
| | Absorption | Emission | Lifetime | | | | | | ZhangDDI | ChChMiner | ZhangDDI | ChChMiner |
| No Pre-training | 19.66 (0.69) | 25.84 (0.23) | 0.821 (0.017) | 0.567 (0.014) | 0.884 (0.074) | 0.331 (0.029) | 0.412 (0.028) | 0.458 (0.002) | 71.75 (0.76) | 76.21 (1.19) | 70.96 (1.40) | 75.81 (0.79) |
| **MPP (molecular property prediction) Pre-training** | | | | | | | | | | | | |
| 3D Infomax | 18.71 (0.61) | 24.59 (0.22) | 0.790 (0.022) | 0.585 (0.015) | 0.873 (0.103) | 0.321 (0.041) | 0.426 (0.036) | 0.464 (0.004) | 71.01 (2.19) | 76.05 (1.30) | 70.90 (1.63) | 74.87 (1.08) |
| GraphMVP | 18.40 (0.62) | 24.73 (0.14) | 0.797 (0.022) | 0.561 (0.025) | 1.010 (0.115) | 0.301 (0.025) | 0.418 (0.020) | 0.437 (0.015) | 71.82 (1.44) | 76.42 (1.68) | 71.73 (0.95) | 76.13 (1.01) |
| MoleculeSDE | 18.56 (0.24) | 24.91 (0.10) | 0.836 (0.040) | 0.564 (0.018) | 0.971 (0.122) | 0.308 (0.024) | 0.426 (0.028) | 0.454 (0.012) | 70.07 (0.58) | 76.37 (1.14) | 69.46 (1.55) | 76.03 (1.13) |
| **MRL (molecular relational learning) Pre-training** | | | | | | | | | | | | |
| 3DMRL | **18.00** (0.17) | **24.21** (0.09) | **0.729** (0.014) | **0.528** (0.019) | **0.839** (0.105) | **0.277** (0.006) | **0.371** (0.031) | **0.435** (0.006) | **74.00** (0.72) | **78.93** (0.59) | **74.85** (1.58) | **78.56** (1.03) |

with 3DMRL consistently delivers significant performance improvements across downstream tasks. This validates the effectiveness of our approach, as it successfully integrates scientific knowledge into the pre-training strategy, enhancing the model's overall performance. **4)** Additionally, for the DDI task in Table 2 (b), we observed that the performance improvement is more pronounced in challenging scenarios ((d) Scaffold split) compared to less difficult ones ((c) Molecule split). This highlights the enhanced generalization ability of 3DMRL in out-of-distribution scenarios, demonstrating its potential for drug discovery applications where robust generalization across unknown molecules is essential. We explore the *extrapolation* capability of 3DMRL in Appendix E.2.

### 5.3 Model Analysis

**Ablation Studies.** To further understand our model, we conduct an ablation study to investigate the impact of two key components on the final performance. Specifically, as shown in Equation 6, the objective function contains two terms: (i) global geometry loss and (ii) intermolecular local geometry loss; we curate two variants that involve only (i) (denoted **only glob.**) and only (ii) (denoted **only local**) in Figure 3. As shown in Figure 3, learning the global geometry plays a particularly critical role. Removing it from 3DMRL results in a significant performance drop, even falling below MPP pre-training strategies such as 3D Infomax and GraphMVP. This is because the global geometry loss allows the model to capture the overall interaction geometry at the molecular level, while the local geometry loss focuses on learning more fine-grained, atom-level interactions. However, combining both losses, as in 3DMRL, yields the best results, demonstrating the importance of leveraging the strengths of both levels of granularity. We provide further detailed results of ablation studies in Appendix E.3.

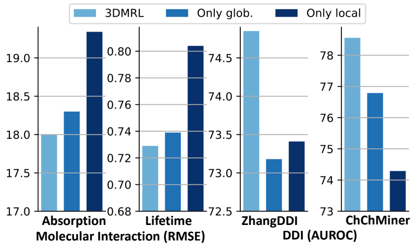


Figure 3: Ablation studies.

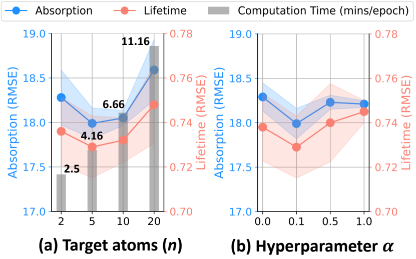
**Molecule Collision Analysis.** While the virtual environment is designed to carefully mimic the nature of molecular interactions, as discussed in Section 4.1, molecule collisions can still occur within the environment. To first examine how molecule collisions affect model performance, we created a "No Radius" model that does not take the radius into account during pre-training. Looking at the atomic overlap in Table 3, we observed that 3DMRL, which utilizes radius information, significantly reduces the overlap ratio between molecules compared to the "No Radius" configuration. Moreover, we found that in the "No Radius" case, where the atomic overlap ratio is very high, the performance was much lower than that of the 3DMRL. To further investigate whether further reducing atomic overlap would be helpful, we experimented with several additional configurations. The "Fixed direction" configuration was designed to prevent overlap caused by random direction placement by positioning the solvent along the direction from the origin to the target atom. "Twice radius" refers to multiplying the radius in the equation in line 170. These methods reduce atomic overlap by decreasing randomness and increasing the distance between molecules, respectively; however, in terms of performance, they were either similar to or worse than 3DMRL.

Table 3: Model performance in various 3D interaction environments with reduced collision.

| | Atomic | Time | Performance | | |
|---|---|---|---|---|---|
| | Overlap | (min/epoch) | Absorption | Emission | Lifetime |
| No Radius | 73.84% | 5.30 | 18.68 | 25.37 | 0.745 |
| Fixed Direction | 19.28 % | 5.30 | 18.26 | 24.24 | 0.734 |
| Twice Radius | 10.28 % | 5.32 | 18.23 | 24.25 | 0.730 |
| Regenerate | 0.0% | 23.01 | 18.20 | **23.86** | **0.727** |
| 3DMRL | 25.12% | 5.32 | **18.00** | 24.21 | 0.729 |

The experimental results showed that both methods were able to reduce atomic overlap, but in terms of performance, they were either similar to or worse than 3DMRL. Lastly, we introduce "Regenerate," which regenerates the 3D virtual environment every time a collision occurs between any molecules in a virtual environment. Although the collision between molecules can certainly be avoided in this case, this approach incurs high computational complexity. In Table 3, we observe that the performance gain of "Regenerate" is minimal, despite its significantly higher computational requirements. Based on these results, we argue that 3DMRL strikes an appropriate balance between computation and performance.

**Sensitivity analysis on $n$.** Moreover, we conduct a sensitivity analysis to explore the empirical effect of the number of target atoms $n$, which determines the number of small molecules in a virtual interaction geometry. In Figure 4 (a), we observe that the model performs the best when using five small molecules to construct the virtual interaction geometry. More specifically, using too few small molecules ($n = 2$) results in poorer performance, as it fails to adequately simulate real-world interaction environments. On the other hand, the model performance also declines as the number of small molecules increases, likely due to the 3D geometry encoder overfitting to the small molecules with an excessive count. Furthermore, we observe that as the number of target atoms increases, more extensive computational resources are required to encode the 3D interaction geometry during pre-training. Hence, selecting an appropriate number of target atoms is crucial for both model performance and computational efficiency. We provide further analyses on virtual interaction geometry and other tasks in Appendix E.4.



(a) Target atoms ($n$)

(b) Hyperparameter $\alpha$

Figure 4: Sensitivity analysis.

**Sensitivity analysis on $\alpha$.** We conduct sensitivity analysis on $\alpha$, which controls the weight of local geometry loss, in Equation 6. In Figure 4 (b), the model's performance declines as $\alpha$ increases from 0.1, primarily because it overly emphasizes atom-level interactions between the molecules instead of considering the overall interaction geometry. Conversely, we also notice a drop in performance when local geometry loss is not utilized ($\alpha = 0.0$), as this causes the model to lose ability in learning fine-grained atom-level interactions. It is important to note that while we set $n = 5$ and $\alpha = 0.1$ during pre-training, models pre-trained with varying $n$ and $\alpha$ consistently outperform those trained from scratch, demonstrating 3DMRL's robustness.

## 6 Conclusion

In this work, we propose 3DMRL, a novel pre-training framework that effectively integrates 3D geometric information into MRL. By constructing a virtual interaction geometry and utilizing local and global geometry prediction, our approach effectively incorporates complex 3D interaction geometry information into 2D MRL models. Experimental results demonstrate that 3DMRL significantly enhances the performance of 2D MRL models across various downstream tasks and neural architectures, validating the importance of incorporating 3D geometric data.

For **future work**, we intend to develop and train a virtual interaction geometry generator capable of mimicking MD trajectories of molecular interactions. We will then substitute this generator for the purely random generation method currently used in Section 4.1, providing a more physically informed signal.. Furthermore, we plan to broaden this research in drug-target binding affinity prediction, a core task in drug discovery that involves complex protein structures as the larger molecule.

## Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02304967, AI Star Fellowship (KAIST)). Additionally, this research received funding from the National Research Foundation of Korea (NRF) through two separate grants: RS-2024-00335098 (funded by the Korea government (MSIT)) and RS-2022-NR068758 (funded by the Ministry of Science and ICT).

## References

[1] Atkins, P. W., De Paula, J., and Keeler, J. *Atkins' physical chemistry*. Oxford university press, 2023.

[2] Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

[3] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

[4] Deac, A., Huang, Y.-H., Veličković, P., Liò, P., and Tang, J. Drug-drug adverse effect prediction with graph co-attention. *arXiv preprint arXiv:1905.00534*, 2019.

[5] Du, W., Zhang, H., Du, Y., Meng, Q., Chen, W., Zheng, N., Shao, B., and Liu, T.-Y. Se (3) equivariant graph neural networks with complete local frames. In *International Conference on Machine Learning*, pp. 5583–5608. PMLR, 2022.

[6] Durrant, J. D. and McCammon, J. A. Molecular dynamics simulations and drug discovery. *BMC biology*, 9:1–9, 2011.

[7] Duval, A., Mathis, S. V., Joshi, C. K., Schmidt, V., Miret, S., Malliaros, F. D., Cohen, T., Lio, P., Bengio, Y., and Bronstein, M. A hitchhiker's guide to geometric gnns for 3d atomic systems. *arXiv preprint arXiv:2312.07511*, 2023.

[8] Feng, S., Ni, Y., Lan, Y., Ma, Z.-M., and Ma, W.-Y. Fractional denoising for 3d molecular pre-training. In *International Conference on Machine Learning*, pp. 9938–9961. PMLR, 2023.

[9] Frenkel, D. and Smit, B. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

[10] Fu, Y., Lu, Y., Wang, Y., Zhang, B., Zhang, Z., Yu, G., Liu, C., Clarke, R., Herrington, D. M., and Wang, Y. Ddn3. 0: Determining significant rewiring of biological network structure with differential dependency networks. *Bioinformatics*, pp. btae376, 2024.

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

[12] Grimme, S. Exploration of chemical compound, conformer, and reaction space with meta-dynamics simulations based on tight-binding quantum chemical calculations. *Journal of chemical theory and computation*, 15(5):2847–2862, 2019.

[13] Grubbs, L. M., Saifullah, M., Nohelli, E., Ye, S., Achi, S. S., Acree Jr, W. E., and Abraham, M. H. Mathematical correlations for describing solute transfer into functionalized alkane solvents containing hydroxyl, ether, ester or ketone solvents. *Fluid phase equilibria*, 298(1): 48–53, 2010.

[14] Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.

[15] Jiao, R., Han, J., Huang, W., Rong, Y., and Liu, Y. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8096–8104, 2023.

[16] Joung, J. F., Han, M., Jeong, M., and Park, S. Experimental database of optical properties of organic compounds. *Scientific data*, 7(1):1–6, 2020.

[17] Joung, J. F., Han, M., Hwang, J., Jeong, M., Choi, D. H., and Park, S. Deep learning optical spectroscopy based on experimental database: potential applications to molecular design. *JACS Au*, 1(4):427–438, 2021.

[18] Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Kuroshima, D., Kilgour, M., Tuckerman, M. E., and Rogal, J. Machine learning classification of local environments in molecular crystals. *Journal of chemical theory and computation*, 20 (14):6197–6206, 2024.

[20] Landrum, G. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[21] Lee, N., Hyun, D., Na, G. S., Kim, S., Lee, J., and Park, C. Conditional graph information bottleneck for molecular relational learning. In *International Conference on Machine Learning*, pp. 18852–18871. PMLR, 2023.

[22] Lee, N., Yoon, K., Na, G. S., Kim, S., and Park, C. Shift-robust molecular relational learning with causal substructure. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1200–1212, 2023.

[23] Lim, H. and Jung, Y. Delfos: deep learning model for prediction of solvation free energies in generic organic solvents. *Chemical science*, 10(36):8306–8315, 2019.

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

[25] Liu, S., Du, W., Ma, Z.-M., Guo, H., and Tang, J. A group symmetric stochastic differential equation model for molecule multi-modal pretraining. In *International Conference on Machine Learning*, pp. 21497–21526. PMLR, 2023.

[26] Marenich, A. V., Kelly, C. P., Thompson, J. D., Hawkins, G. D., Chambers, C. C., Giesen, D. J., Winget, P., Cramer, C. J., and Truhlar, D. G. Minnesota solvation database (mnsol) version 2012. 2020.

[27] Martínez, L., Andrade, R., Birgin, E. G., and Martínez, J. M. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of computational chemistry*, 30(13):2157–2164, 2009.

[28] Megyes, T., Bálint, S., Peter, E., Grósz, T., Bakó, I., Krienke, H., and Bellissent-Funel, M.-C. Solution structure of nano3 in water: Diffraction and molecular dynamics simulation study. *The Journal of Physical Chemistry B*, 113(13):4054–4064, 2009.

[29] Mobley, D. L. and Guthrie, J. P. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28(7):711–720, 2014.

[30] Moine, E., Privat, R., Sirjean, B., and Jaubert, J.-N. Estimation of solvation quantities from experimental thermodynamic data: Development of the comprehensive compsol databank for pure and mixed solutes. *Journal of Physical and Chemical Reference Data*, 46(3):033102, 2017.

[31] Ni, Y., Feng, S., Ma, W.-Y., Ma, Z.-M., and Lan, Y. Sliced denoising: A physics-informed molecular pre-training method. *arXiv preprint arXiv:2311.02124*, 2023.

[32] Pathak, Y., Laghuvarapu, S., Mehta, S., and Priyakumar, U. D. Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 873–880, 2020.

[33] Ryu, J. Y., Kim, H. U., and Lee, S. Y. Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the national academy of sciences*, 115(18):E4304–E4311, 2018.

[34] Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[35] Silverman, R. B. and Holladay, M. W. *The organic chemistry of drug design and drug action*. Academic press, 2014.

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

[37] Velez-Arce, A., Huang, K., Li, M. M., Lin, X., Gao, W., Fu, T., Kellis, M., Pentelute, B. L., and Zitnik, M. Tdc-2: Multimodal foundation for therapeutic science. *bioRxiv*, pp. 2024–06, 2024.

[38] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[39] Vermeire, F. H. and Green, W. H. Transfer learning for solvation free energies: From quantum chemistry to experiments. *Chemical Engineering Journal*, 418:129307, 2021.

[40] Vinyals, O., Bengio, S., and Kudlur, M. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[41] Wang, Y., Min, Y., Chen, X., and Wu, J. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In *Proceedings of the Web Conference 2021*, pp. 2921–2933, 2021.

[42] Wu, F. and Li, S. Z. Diffmd: a geometric diffusion model for molecular dynamics simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5321–5329, 2023.

[43] Xiong, Z., Wang, D., Liu, X., Zhong, F., Wan, X., Li, X., Li, Z., Luo, X., Chen, K., Jiang, H., et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

[44] Xu, B., Lu, Y., Li, C., Yue, L., Wang, X., Hao, N., Fu, T., and Chen, J. Smiles-mamba: Chemical mamba foundation models for drug admet prediction. *arXiv preprint arXiv:2408.05696*, 2024.

[45] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

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

[47] Zhang, J., Zhang, H., Wu, T., Wang, Q., and Van Der Spoel, D. Comparison of implicit and explicit solvent models for the calculation of solvation free energy in organic solvents. *Journal of chemical theory and computation*, 13(3):1034–1043, 2017.

[48] Zhang, W., Chen, Y., Liu, F., Luo, F., Tian, G., and Li, X. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *BMC bioinformatics*, 18(1):1–12, 2017.

[49] Zitnik, M., Sosic, R., and Leskovec, J. Biosnap datasets: Stanford biomedical network dataset collection. *Note: http://snap. stanford. edu/biodata Cited by*, 5(1), 2018.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, abstract and introduction accurately reflect the paper's contributions and scope.

   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: In Section 5.3, we have discussed that our approach generates the configuration with collapsed molecules.

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: The paper 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 provide the source code in the external URL along with the implementation details in the paper.

   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 provide access to the code.

   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 have provided a detailed evaluation protocol and hyperparameters for model training in Appendix D.

   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 provide statistical error for each table.

   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 have provided computing resources 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: We follow the NeurIPS 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: [NA]

    Justification: Since our model is about science application, we believe there are no potential societal impacts of the work.

    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: This paper poses no such risks.

    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 have adequately cited the relevant works and data sources.

    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: Our source code is well documented in an external URL, and also describes the implementation details in the Appendix.

    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 does not involve crowd sourcing 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 does not involve crowd sourcing 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We only use LLMs for editing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

# Supplementary Material for 3D Interaction Geometric Pre-training for Molecular Relational Learning

# A Molecular Relational Learning

In this section, we provide further clarification on molecular relational learning by contrasting it with conventional molecular property prediction tasks. As illustrated in Figure 5 (a), conventional molecular property prediction focuses on learning the properties of a single molecule. Models like GraphMVP, 3D Infomax, and MoleculeSDE utilize the 3D information of individual molecules during pre-training to improve performance in downstream tasks aimed at predicting single molecular properties.

In contrast, as shown in Figure 5 (b), molecular relational learning focuses on learning the properties of molecules after their interactions. Our pre-training approach trains both encoders simultaneously to learn 3D information from the virtual environment $g_{vr}$. What sets our approach apart from traditional molecular pretraining is its specific tailoring to our Molecular Relational Learning strategy. This allows the two encoders to learn how paired molecules interact in 3D space, which is essential for various downstream tasks in Molecular Relational Learning.

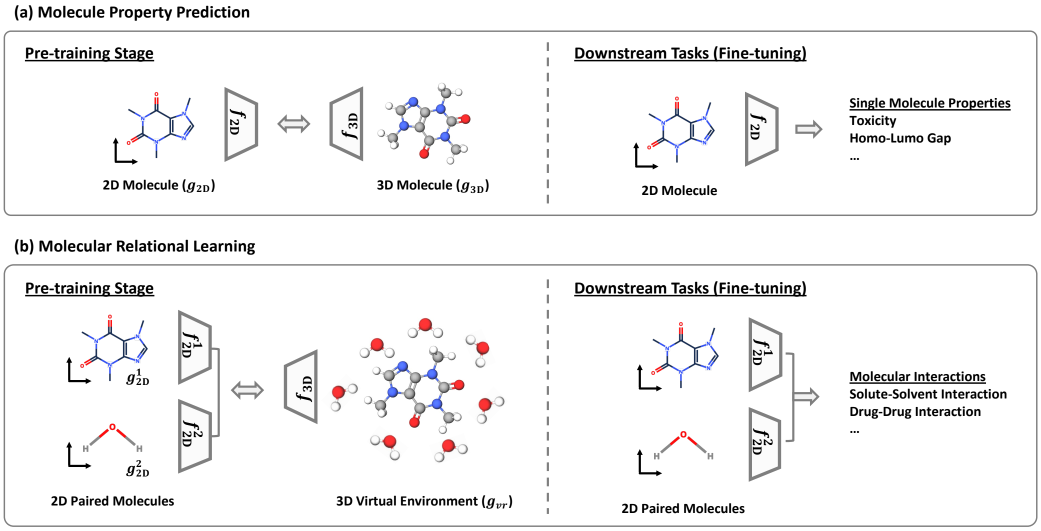


Figure 5: Difference between the conventional pre-training strategy for (a) molecular property prediction and our (b) molecular relational learning.

# B Datasets

## B.1 Pre-Training Datasets

We utilize three distinct datasets, i.e., **Chromophore**, **CombiSolv**, and **DDI**, to pre-train 3DMRL for each downstream task as described in Section 5. Specifically, we use the **Chromophore** dataset for downstream tasks involving the optical properties of chromophores, the **CombiSolv** dataset for tasks related to the solvation free energy of solutes, and the **DDI** dataset, which we created for the drug-drug interaction task.

- The **Chromophore** dataset [16] consists of 20,236 combinations derived from 6,815 chromophores and 1,336 solvents, provided in SMILES string format. For pre-training, we initially convert chromophores and solvents into their respective 3D structures via rdkit, resulting in 6,524 3D structures for chromophores and 1,255 for solvents. These 6,524 unique chromophores are then randomly paired with the 1,255 solvents to generate a sufficient number of pairs. Out of the possible 8,187,620 chromophore-solvent combinations, we randomly sample 1%, which corresponds to 81,876 pairs, for pre-training.
- The **CombiSolv** dataset [39] contains 10,145 combinations derived from 1,368 solutes and 291 solvents, provided in SMILES string format. Similar to our approach with the Chromophore dataset, we first convert solutes and solvents into their corresponding 3D structures, yielding

1,368 3D structures for solutes and 290 for solvents. From the potential random combinations, we select 79,344 solute-solvent pairs, representing 20% of all possible pairs.

- For the **DDI** dataset, we compile drug-drug pairs from the ZhangDDI [48], ChChMiner [49], and DeepDDI [33] datasets. From a total of 235,547 positive pairs, we randomly sample 40% (i.e., 94,218 pairs) for use as the pre-training dataset. While chromophores and solutes act as the larger molecule $g^1$ in molecular interaction tasks, in the DDI dataset, we designate the drug with the larger radius as the larger molecule.

### B.2 Downstream Task Datasets

**Molecular Interaction Prediction.** For the molecular interaction prediction task, we transform the SMILES strings into graph structures using the CIGIN implementation available on GitHub [2][32]. Regarding the datasets related to solvation free energies, such as MNSol, FreeSolv, CompSol, Abraham, and CombiSolv, we utilize SMILES-based datasets from previous studies [39]. Following previous work [21], we specifically filter the data to include only solvation free energies measured at temperatures of 298 K ($\pm$ 2) and exclude any data involving ionic liquids and ionic solutes [39].

- The **Chromophore** dataset [16] consists of 20,236 combinations derived from 6,815 chromophores and 1,336 solvents, provided in SMILES string format. This dataset includes optical properties sourced from scientific publications, with unreliable experimental results being excluded after thorough examination of absorption and emission spectra. In our work, we assess model performance by predicting three key properties: **maximum absorption wavelength (Absorption)**, **maximum emission wavelength (Emission)**, and **excited state lifetime (Lifetime)**, which are crucial for designing chromophores for specific applications. To ensure the integrity of each dataset, we remove any NaN values that were not reported in the original publications. Additionally, following previous work [21], for the Lifetime data, we apply log normalization to the target values to mitigate skewness in the dataset, thereby enhancing training stability.
- The **MNSol** dataset [26] features 3,037 experimentally measured free energies of solvation or transfer for 790 distinct solutes and 92 solvents. For our study, we focus on 2,275 pairs comprising 372 unique solutes and 86 solvents, in alignment with prior research [39].
- The **FreeSolv** dataset [29] offers 643 hydration free energy values, both experimental and calculated, for small molecules in water. In our research, we utilize 560 experimental measurements, consistent with the dataset selection criteria from previous studies [39].
- The **CompSol** dataset [30] has been designed to illustrate the impact of hydrogen-bonding association effects on solvation energies. For our study, we analyze 3,548 solute-solvent pairs, encompassing 442 distinct solutes and 259 solvents, in accordance with prior research parameters [39].
- The **Abraham** dataset [13], curated by the Abraham research group at University College London, provides extensive data on solvation. For this study, we focus on 6,091 solute-solvent combinations, comprising 1,038 distinct solutes and 122 solvents, as outlined in previous research [39].
- The **CombiSolv** dataset [39] integrates the data from MNSol, FreeSolv, CompSol, and Abraham, encompassing a total of 10,145 solute-solvent combinations. This dataset features 1,368 unique solutes and 291 distinct solvents.

**Drug-Drug Interaction (DDI) Prediction.** In the drug-drug interaction prediction task, we utilize the positive drug pairs provided in the MIRACLE GitHub repository[3], which excludes data instances that cannot be represented as graphs from SMILES strings. To create negative samples, we generate a corresponding set by sampling from the complement of the positive drug pairs. This approach is applied to both datasets. Additionally, for the classification task, we adhere to the graph conversion process outlined by MIRACLE [41].

- The **ZhangDDI** dataset [48] includes data on 548 drugs and 48,548 pairwise interactions, along with various types of similarity information pertaining to these drug pairs.

[2] `https://github.com/devalab/CIGIN`

[3] `https://github.com/isjakewong/MIRACLE/tree/main/MIRACLE/datachem`

Table 4: Statistics of datasets. $\mathcal{G}^1$ and $\mathcal{G}^2$ are defined in Section 5.1.

| Task | Dataset | | $\mathcal{G}^1$ | $\mathcal{G}^2$ | # $\mathcal{G}^1$ | # $\mathcal{G}^2$ | # Pairs |
|---|---|---|---|---|---|---|---|
| Molecular Interaction | Chromophore [4] | Absorption | Chromophore | Solvent | 6,416 | 725 | 17,276 |
| | | Emission | Chromophore | Solvent | 6,412 | 1,021 | 18,141 |
| | | Lifetime | Chromophore | Solvent | 2,755 | 247 | 6,960 |
| | MNSol [5] | | Solute | Solvent | 372 | 86 | 2,275 |
| | FreeSolv [6] | | Solute | Solvent | 560 | 1 | 560 |
| | CompSol [7] | | Solute | Solvent | 442 | 259 | 3,548 |
| | Abraham [8] | | Solute | Solvent | 1,038 | 122 | 6,091 |
| | CombiSolv [9] | | Solute | Solvent | 1,495 | 326 | 10,145 |
| Drug-Drug Interaction | ZhangDDI [10] | | Small-molecule Drug | Small-molecule Drug | 544 | 544 | 40,255 |
| | ChChMiner [11] | | Small-molecule Drug | Small-molecule Drug | 949 | 949 | 21,082 |

- The **ChChMiner** dataset [49] comprises 1,322 drugs and 48,514 annotated DDIs, sourced from drug labels and scientific literature.

Despite the **ChChMiner** dataset containing a significantly higher number of drug instances compared to the **ZhangDDI** dataset, the number of labeled DDIs is nearly equivalent. This suggests that the **ChChMiner** dataset exhibits a much sparser network of relationships between drugs.

# C Baselines Setup

To validate the effectiveness of 3DMRL, we primarily evaluate molecular relational learning model architectures trained from scratch for downstream tasks, as well as the same models that are first pre-trained with 3DMRL and then fine-tuned for various downstream tasks. We include the following molecular relational learning model architectures:

- **MPNN** (Message Passing Neural Networks) [11] was originally proposed to predict the various chemical properties of a single molecule. For molecular relational learning tasks, we independently encode each molecule in a pair using MPNN and then concatenate their representations.

  To apply 3DMRL for **MPNN**, we first obtain the atom representation matrices $\mathbf{E}^1$ and $\mathbf{E}^2$ using $f^1_{\text{2D}}$ and $f^1_{\text{2D}}$, which are MPNNs. Then, we directly use $\mathbf{E}^1$ and $\mathbf{E}^2$ instead of the $\mathbf{H}^1$ and $\mathbf{H}^2$, which considers the interaction between two molecules in Section 3.2. That is, we obtain graph-level embeddings $\mathbf{z}^1_{\text{2D}}$ and $\mathbf{z}^2_{\text{2D}}$ via $\mathbf{E}^1$ and $\mathbf{E}^2$ with Set2set readout function. Following contrastive learning is done with $\mathbf{z}^1_{\text{2D}}$ and $\mathbf{z}^2_{\text{2D}}$, and the edge representations $\mathbf{e}^{k,l}_{\text{2D}}$ and and initial atom representations for relative geometry $\hat{\mathbf{X}}$ is obtained through $\mathbf{E}^1$ and $\mathbf{E}^2$. One can simply alternate $\mathbf{H}^1$ and $\mathbf{H}^2$ in Section 4 to $\mathbf{E}^1$ and $\mathbf{E}^2$.
- **AttentiveFP** [43] was also initially proposed to predict various chemical properties of individual molecules by employing a graph attention mechanism to gather more information from relevant molecular datasets. For molecular relational learning tasks, we independently encode each molecule in a pair using MPNN and then concatenate their representations.

  More specifically, **AttentiveFP** first obtain atom representation matrices $\mathbf{H}^1$ and $\mathbf{H}^2$ using $f^1_{\text{2D}}$ and $f^1_{\text{2D}}$, which consist of GAT and GRU layers. Then, the model obtain initial molecule representation $\tilde{\mathbf{z}}^1_{\text{2D}}$ and $\tilde{\mathbf{z}}^2_{\text{2D}}$ which are further enhanced by considering other molecules in a batch through GAT layers. After passing multiple GAT layers, the model obtain final molecule representations $\tilde{\mathbf{z}}^1_{\text{2D}}$ and $\tilde{\mathbf{z}}^2_{\text{2D}}$. In our framework, contrastive learning is done with $\mathbf{z}^1_{\text{2D}}$ and $\mathbf{z}^2_{\text{2D}}$,

[4] https://figshare.com/articles/dataset/DB_for_chromophore/12045567/2

[5] https://conservancy.umn.edu/bitstream/handle/11299/213300/MNSolDatabase_v2012.zip?sequence=12&isAllowed=y

[6] https://escholarship.org/uc/item/6sd403pz

[7] https://aip.scitation.org/doi/suppl/10.1063/1.5000910

[8] https://www.sciencedirect.com/science/article/pii/S0378381210003675

[9] https://ars.els-cdn.com/content/image/1-s2.0-S1385894721008925-mmc2.xlsx

[10] https://github.com/zw9977129/drug-drug-interaction/tree/master/dataset

[11] http://snap.stanford.edu/biodata/datasets/10001/10001-ChCh-Miner.html

and the edge representations $\mathbf{e}_{\text{2D}}^{k,l}$ and and initial atom representations for relative geometry $\hat{\mathbf{X}}$ is obtained through $\mathbf{H}^1$ and $\mathbf{H}^2$.

- **CIGIN** (Chemically Interpretable Graph Interaction Network) [32] proposes to model the interaction between the molecules through a dot product between atoms in paired molecules. By doing so, they successfully predict the solubility of drug molecules. We provide detailed descriptions on how to apply 3DMRL for CIGIN in Section 4.
- **CGIB** (Conditional Graph Information Bottleneck) and **CGIB**$_{\text{cont}}$ (Conditional Graph Information Bottleneck with Contrastive Learning)[21] aim to enhance generalization in molecular relational learning by identifying the core substructure of molecules during chemical reactions, based on the information bottleneck theory. While CIGIN is limited to predicting drug solubility, **CGIB** and **CGIB**$_{\text{cont}}$ extend molecular relational learning to predict the optical properties of chromophores in various solvents, molecule solubility in various solvents, and drug-drug interactions.

  **CGIB** and **CGIB**$_{\text{cont}}$ model architectures are highly similar to CIGIN, but they have another branch named *compress module*, which aims to inject noise to the atoms that are not important during the model. Specifically, they obtain $\mathbf{T}^1$ that is node representation matrix with noise, and obtain $\mathbf{z}_{\mathcal{G}^1_{\text{CIB}}}$ from the noise injected matrix along with $\mathbf{z}_{\mathcal{G}^1}$ and $\mathbf{z}_{\mathcal{G}^2}$ which are obtained from $\mathbf{H}^1$ and $\mathbf{H}^2$, respectively. To apply 3DMRL for **CGIB**, we pre-train the model without noise injection module, thereby using $\mathbf{H}^1$, $\mathbf{H}^2$, $\mathbf{z}_{\mathcal{G}^1}$, and $\mathbf{z}_{\mathcal{G}^2}$ in **CGIB** as $\mathbf{H}^1$, $\mathbf{H}^2$, $\mathbf{z}^1_{\text{2D}}$, and $\mathbf{z}^2_{\text{2D}}$ in Section 4. After pre-training staget, all the modules including noise injection module is trained for the downstream tasks.

In addition to the model architectures, we also compare the recent state-of-the-art molecular pre-training methods based on CIGIN architecture. Since molecular pre-training methods are specifically designed for a single molecule, we pre-train each molecule encoder in CIGIN architecture and adopted the pre-trained weights for molecular relational learning downstream tasks. In Section 5, we include following molecular pre-training approaches:

- **No pre-training** does not involve pertaining process and fine-tune the model using labeled data
- **3D Infomax** [36] increase the mutual information between 2D and 3D molecular representations using contrastive learning
- **GraphMVP** [24] incorporates a generative pre-training framework in addition to contrastive learning
- **MoleculeSDE** [25] designs a denoising framework to capture the 3D geometric distribution of molecules, thereby revealing the relationship between the score function and the molecular force field.

To apply these approaches for MRL, we first pre-train the each encoder $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ in Section 3.2 with the above approaches. Then, the pre-trained encoders $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ are utilized to output the representations $\mathbf{E}^1$ and $\mathbf{E}^2$, following the remaining pipeline of the model outlined in Section 3.2. That is, each molecule encoder $f^1_{\text{2D}}$ and $f^2_{\text{2D}}$ implicitly possesses knowledge about the 3D structure of individual molecules, but not the complex interaction geometry between multiple molecules.

# D Implementation Details

## D.1 Evaluation Protocol

Following Pathak et al. [32], for the molecular interaction prediction task, we evaluate the models under a 5-fold cross-validation scheme. The dataset is randomly split into 5 subsets and one of the subsets is used as the test set, while the remaining subsets are used to train the model. A subset of the test set is selected as the validation set for hyperparameter selection and early stopping. We repeat 5-fold cross-validation three times (i.e., 15 runs in total) and report the accuracy and standard deviation of the repeats. For the DDI prediction task [21], we conduct experiments on two different *out-of-distribution* scenarios, namely **molecule split** and **scaffold split**. For the **molecule split**, the performance is evaluated when the models are presented with new molecules not included in the training dataset. Specifically, let $\mathbb{G}$ denote the total set of molecules in the dataset. Given $\mathbb{G}$, we split $\mathbb{G}$ into $\mathbb{G}_{\text{old}}$ and $\mathbb{G}_{\text{new}}$, so that $\mathbb{G}_{\text{old}}$ contains the set of molecules that have been seen in the training phase, and $\mathbb{G}_{\text{new}}$ contains the set of molecules that have not been seen in the training phase.

Then, the new split of dataset consists of $\mathcal{D}_{\text{train}} = \{(\mathcal{G}^1, \mathcal{G}^2) \in \mathcal{D} | \mathcal{G}^1 \in \mathbb{G}_{\text{old}} \wedge \mathcal{G}^2 \in \mathbb{G}_{\text{old}}\}$ and $\mathcal{D}_{\text{test}} = \{(\mathcal{G}^1, \mathcal{G}^2) \in \mathcal{D} | (\mathcal{G}^1 \in \mathbb{G}_{\text{new}} \wedge \mathcal{G}^2 \in \mathbb{G}_{\text{new}}) \vee (\mathcal{G}^1 \in \mathbb{G}_{\text{new}} \wedge \mathcal{G}^2 \in \mathbb{G}_{\text{old}}) \vee (\mathcal{G}^1 \in \mathbb{G}_{\text{old}} \wedge \mathcal{G}^2 \in \mathbb{G}_{\text{new}})\}$. We use a subset of $\mathcal{D}_{\text{test}}$ as the validation set in inductive setting. In the **scaffold split** setting [14], just like in the molecule split, molecules corresponding to scaffolds that were not seen during training will be used for testing. For both splits, we repeat 5 independent experiments with different random seeds on split data, and report the accuracy and the standard deviation of the repeats. In both scenarios, we split the data into training, validation, and test sets with a ratio of 60/20/20%.

### D.2 Model architecture

For the 2D MRL model, following a previous work [32], we use 3-layer MPNNs [11] as our backbone molecule encoder to learn the representation of solute and solvent for the molecular interaction prediction, while we use a GIN [45] to encode both drugs for the drug-drug interaction prediction task [21]. We utilize a hidden dimension of 56 for molecular interaction tasks and 300 for drug-drug interaction tasks, employing the ReLU activation function for both. For the 3D virtual environment encoder $f_{3\text{D}}$, we utilize SchNet [34], which guarantees an *SE(3)-invariant* representation of the environment. For both molecular interaction and drug-drug interaction tasks, we configure SchNet with 128 hidden channels, 128 filters, 6 interaction layers, and a cutoff distance of 5.0.

### D.3 Model training

For model optimization during **Pre-training** stage, we employ the Adam optimizer with an initial learning rate of 0.0005 for the chromophore task, 0.0001 for the solvation free energy task, and 0.0005 for the DDI tasks. The model is optimized over 100 epochs during pre-training.

In the **downstream tasks**, the learning rate was reduced by a factor of $10^{-1}$ after 20 epochs of no improvement in model performance in validation set, following the approach in a previous work [32], with the initial learning rate of 0.005 for the chromophore task, 0.001 for the solvation free energy task, and 0.0005 for the DDI tasks.

**Computational resources.** We perform all pre-training on a 40GB NVIDIA A6000 GPU, whereas all downstream tasks are executed on a 24GB NVIDIA GeForce RTX 3090 GPU.

**Software configuration.** Our model is implemented using Python 3.7, PyTorch 1.9.1, RDKit 2020.09.1, and Pytorch-geometric 2.0.3.

## E Additional Experimental Results

### E.1 Molecular Property Prediction Pre-training with Large-Scale Datasets

Although MPP pre-training approaches demonstrate unsatisfactory performance in Section 5, a positive aspect is their ability to leverage large-scale datasets containing both 2D and 3D molecular information. Consequently, we further explore whether utilizing a large-scale pre-training dataset can enhance MPP pre-training strategies in MRL tasks. To do so, we pre-train the encoders with each strategy with randomly sampled 50K molecules in GEOM dataset [2], which consists of 2D topological information and 3D geometric information, following the previous work [24]. In Table 5, we observe that a large-scale pre-training dataset does not consistently result in performance improvements for MRL downstream tasks and can still cause negative transfer in various tasks. On the other hand, we note that MoleculeSTM benefits the most from the large-scale dataset among the strategies, likely due to the complexity of its denoising framework, which necessitates a large-scale dataset to learn the data distribution effectively. Nevertheless, it still exhibits negative transfer in the FreeSolve dataset and performs worse than 3DMRL, highlighting the need for a pre-training strategy specifically tailored to molecular relational learning.

### E.2 Extrapolation in Molecular Interaction Task

The model's generalization ability in out-of-distribution (OOD) datasets is crucial for its application in real-world scientific discovery processes. To this end, we further conduct experiments on molecular interaction tasks by assuming out-of-distribution scenarios, as shown in Table 6. Specifically, we split the dataset based on molecular structure, i.e., molecule split and scaffold split, similar to the approach

Table 5: Performance comparison of CIGIN model on molecular interaction tasks using different pre-training strategies and pre-training dataset (RMSE) (↓). The blue color signifies a positive transfer between the pre-training task and the downstream task, whereas the orange color denotes a negative transfer between the pre-training task and the downstream task. **Pre-training Dataset** indicates the pre-training datasets used during pre-training.

| **Strategy** | **Pre-training Dataset** | **Chromophore** | | | **MNSol** | **FreeSolv** | **CompSol** | **Abraham** | **CombiSolv** |
|---|---|---|---|---|---|---|---|---|---|
| | | **Absorption** | **Emission** | **Lifetime** | | | | | |
| No Pre-training | - | 19.66 (0.69) | 25.84 (0.23) | 0.821 (0.017) | 0.567 (0.014) | 0.884 (0.074) | 0.331 (0.029) | 0.412 (0.028) | 0.458 (0.002) |
| **MPP (molecular property prediction) Pre-training** | | | | | | | | | |
| 3D Infomax | MRL | 18.71 (0.61) | 24.59 (0.22) | 0.790 (0.022) | 0.585 (0.015) | 0.873 (0.103) | 0.321 (0.041) | 0.426 (0.036) | 0.464 (0.004) |
| | GEOM | 18.82 (0.24) | 25.14 (0.18) | 0.795 (0.021) | 0.589 (0.027) | 0.899 (0.080) | 0.319 (0.019) | 0.418 (0.023) | 0.466 (0.017) |
| GraphMVP | MRL | 18.40 (0.62) | 24.73 (0.14) | 0.797 (0.022) | 0.561 (0.025) | 1.010 (0.115) | 0.301 (0.025) | 0.418 (0.020) | 0.437 (0.015) |
| | GEOM | 18.85 (0.74) | 24.87 (0.54) | 0.784 (0.014) | 0.551 (0.013) | 0.900 (0.059) | 0.325 (0.007) | 0.410 (0.036) | 0.437 (0.007) |
| MoleculeSDE | MRL | 18.56 (0.24) | 24.91 (0.10) | 0.836 (0.040) | 0.564 (0.018) | 0.971 (0.122) | 0.308 (0.024) | 0.426 (0.028) | 0.454 (0.012) |
| | GEOM | 18.72 (0.16) | 24.77 (0.48) | 0.773 (0.023) | 0.560 (0.086) | 0.909 (0.142) | 0.290 (0.008) | 0.399 (0.034) | 0.449 (0.007) |
| **MRL (molecular relational learning) Pre-training** | | | | | | | | | |
| 3DMRL | MRL | **18.00** (0.17) | **24.21** (0.09) | **0.729** (0.014) | **0.528** (0.019) | **0.839** (0.105) | **0.277** (0.006) | **0.371** (0.031) | **0.435** (0.006) |

used in the DDI task in Section 5. It is important to note that this scenario is significantly more challenging than the out-of-distribution DDI task in Section 5 because it involves a regression task, which can also be viewed as an **extrapolation** task. As shown in Table 6, we observe that pre-training approaches generally benefit model performance in extrapolation tasks, with the exception of one case, namely 3D Infomax for the Lifetime dataset. Among the pre-training approaches, 3DMRL performs the best, underscoring the extrapolation capability of 3DMRL.

Table 6: Performance comparison of the CIGIN model on extrapolation in molecular interaction tasks using different pre-training strategies (RMSE) (↓).

| **Strategy** | **Molecule Split** | | | **Scaffold Split** | | |
|---|---|---|---|---|---|---|
| | **Absorption** | **Emission** | **Lifetime** | **Absorption** | **Emission** | **Lifetime** |
| No Pre-training | 27.51 (0.74) | 37.04 (1.07) | 1.205 (0.033) | 59.55 (1.35) | 60.11 (1.98) | 1.221 (0.033) |
| **MPP (molecular property prediction) Pre-training** | | | | | | |
| 3D Infomax | 27.38 (1.19) | 36.98 (1.24) | 1.257 (0.050) | 58.34 (1.89) | 58.67 (1.00) | 1.207 (0.041) |
| GraphMVP | 26.93 (1.89) | 36.51 (0.92) | 1.201 (0.034) | 59.27 (1.57) | 57.67 (1.14) | 1.199 (0.024) |
| MoleculeSDE | 27.26 (1.19) | 36.48 (1.12) | 1.135 (0.077) | 57.75 (0.74) | 58.74 (1.02) | 1.214 (0.010) |
| **MRL (molecular relational learning) Pre-training** | | | | | | |
| 3DMRL | **25.01** (1.51) | **34.66** (0.89) | **1.033** (0.027) | **57.58** (1.62) | **57.53** (1.13) | **1.178** (0.010) |

## E.3 Ablation Studies

We provide further ablation studies on molecular interaction task and drug-drug interaction task in Table 7 and 8, respectively.

Table 7: Further results from ablation studies on molecular interaction tasks.

| Strategy | Chromophore | | | MNSol | FreeSolv | CompSol | Abraham | CombiSolv |
|---|---|---|---|---|---|---|---|---|
| | Absorption | Emission | Lifetime | | | | | |
| Only Glob. | 18.30 (0.16) | 24.70 (0.16) | 0.739 (0.015) | 0.531 (0.022) | 0.874 (0.060) | 0.301 (0.018) | 0.376 (0.029) | 0.458 (0.014) |
| Only Local | 19.34 (0.50) | 24.80 (0.05) | 0.804 (0.011) | 0.587 (0.019) | 1.184 (0.173) | 0.330 (0.028) | 0.391 (0.020) | 0.466 (0.021) |
| 3DMRL | **18.00** (0.17) | **24.21** (0.09) | **0.729** (0.014) | **0.528** (0.019) | **0.839** (0.105) | **0.277** (0.006) | **0.371** (0.031) | **0.435** (0.006) |

Table 8: Further results from ablation studies on drug-drug interaction tasks.

| Strategy | (a) Molecule Split | | (b) Scaffold Split | |
|---|---|---|---|---|
| | ZhangDDI | ChChMiner | ZhangDDI | ChChMiner |
| Only Glob. | 73.09 (0.83) | 77.68 (0.55) | 73.18 (0.59) | 76.79 (1.13) |
| Only Local | 73.45 (1.29) | 75.93 (1.14) | 73.41 (2.28) | 74.29 (1.79) |
| 3DMRL | **74.00** (0.72) | **78.93** (0.59) | **74.85** (1.58) | **78.56** (1.03) |

## E.4 Further Virtual Interaction Environment Analysis

**Sensitivity Analysis on DDI Datasets.** In Table 9, we provide sensitivity analysis results in drug-drug interaction tasks.

Table 9: Sensitivity analysis on $n$ in drug-drug interaction tasks.

| | Molecule Split | | Scaffold Split | |
|---|---|---|---|---|
| | ZhangDDI | ChChMiner | ZhangDDI | ChChMiner |
| n = 2 | 73.77 | 77.15 | 74.76 | 77.01 |
| n = 5 | **74.00** | 78.93 | **74.85** | **78.56** |
| n = 10 | 73.96 | **79.12** | 74.36 | 77.76 |
| n = 20 | 73.94 | 78.75 | 74.03 | 77.64 |

**Further Environment Analysis.** While we propose assigning a single small molecule to each target atom in Section 4.1, we also investigate the impact of varying the number of assigned small molecules per atom in the larger molecule. As illustrated in Figure 6, we observe a decline in model performance as the number of small molecules per atom increases, given a fixed number of target atoms $n$. This suggests that modeling interactions between multiple small molecules and a single atom in a larger molecule can degrade model performance. This is consistent with the scientific understanding that, although hydrogen bonding can occasionally allow multiple molecules to interact with a single atom simultaneously, steric and electronic hindrances frequently impede such interactions. Thus, we contend that our proposed virtual interaction geometry appropriately reflects the real-world physics in molecular interactions.

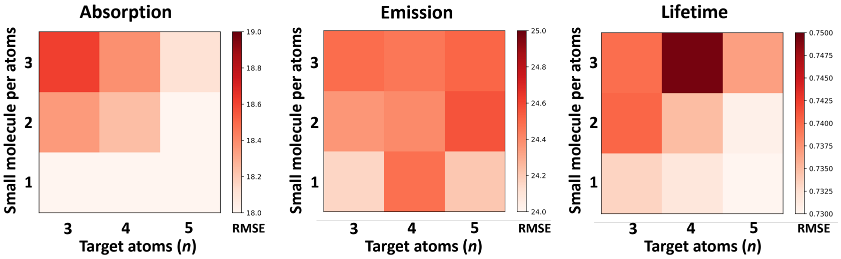


Figure 6: Further environment analysis results.

**Number of Larger Molecules.** In Section 4.1, we initially constructed the virtual geometry in a one-to-many manner (one larger molecule and many smaller molecules) to effectively mimic the explicit solvent model in traditional MD simulations. However, in this section, we explore the many-to-many configurations between larger molecules and smaller molecules. In Table 10, we observe that the best performance was achieved when there was only one larger molecule. However, since the performance differences were not significant, we can conclude that our model is robust across various configurations.

Table 10: Model performance on different number of larger molecules.

| # Larger molecules | **Absorption** | **Emission** | **Lifetime** |
|---|---|---|---|
| 1 (Ours) | **18.00** | **24.21** | **0.729** |
| 2 | 18.28 | 24.43 | 0.738 |
| 3 | 18.37 | 24.35 | 0.749 |

### E.5 3D Encoder Pre-training Approaches

Since the core concept of our paper is to inject 3D information into a 2D encoder, we choose baseline approaches that pre-train 2D molecular encoder with 3D information. In this section, we compare the approaches that pre-train 3D molecular encoder with 3D information. To do so, since the elaborately calculated 3D structure of the molecules is not available for our datasets, we first calculate the 3D structure of the molecules in the dataset using RDKit ETKDG algorithm. However, some of the molecules in the dataset were not able to obtain 3D structures through RDKit ETKDG algorithm. We excluded these molecules from the experiment, and the results are shown below.

Before Conversion: Absorption – 17,276 pairs, Emission – 18,141 pairs, and Lifetime – 6,960 pairs.

After Conversion: Absorption – 16,756 pairs, Emission – 17,525 pairs, and Lifetime – 6,740 pairs.

Table 11: Performance of various 3D encoder pre-training strategies in RMSE (↓). *Note that these results are not directly comparable since some of the molecules in the dataset were not able to obtain 3D structures through RDKit ETKDG algorithm.*

| | **Absorption** | **Emission** | **Lifetime** |
|---|---|---|---|
| 3D-EMGP | 18.62 | **24.06** | 0.753 |
| SliDe | 21.96 | 28.87 | 0.859 |
| Frad | 19.58 | 28.43 | 0.781 |
| 3DMRL | **18.00** | 24.21 | **0.729** |

# F Pseudocode

In this section, we provide pseudocode of 3DMRL in Algorithm 1.

**Algorithm 1** Overall framework of 3DMRL.

```
Input:
        • 2D molecular topology graphs $g^1_{2D}, g^2_{2D}$
        • 3D molecular geometric graphs $g^1_{3D}, g^2_{3D}$
        • 2D graph encoders $f^1_{2D}, f^2_{2D}$
        • 3D Virtual Interaction Geometry Encoder $f_{3D}$
Pre-Training Stage:
For epoch in epochs:
  $\mathbf{z}^1_{2D}, \mathbf{z}^2_{2D}, \mathbf{H}^1, \mathbf{H}^2 =$ 2D MRL ENCODER $(g^1_{2D}, g^2_{2D})$
  $\mathbf{z}_{2D} = (\mathbf{z}^1_{2D} || \mathbf{z}^2_{2D})$
  $g_{vr} =$ VIRTUAL INTERACTION GEOMETRY CONSTRUCTION $(g^1_{3D}, g^2_{3D})$
  $\mathbf{z}_{3D} = f_{3D}(g_{vr})$                    /* Virtual Geometry Encoding via SchNet */
  $\mathcal{L}_{glob.} =$ SE(3) INVARIANT GLOBAL GEOMETRY LEARNING $(\mathbf{z}_{2D}, \mathbf{z}_{3D})$
  $\mathcal{L}_{local} = \frac{1}{n} \sum_{i=1}^{n}$ SE(3) EQUIVARIANT LOCAL RELATIVE GEOMETRY LEARNING $(g^{2,i}_{3D}, \mathbf{H}^1, \mathbf{H}^2)$
  $\mathcal{L}_{pre\text{-}train} = \mathcal{L}_{glob.} + \alpha \cdot \mathcal{L}_{local}$
  Update $f^1_{2D}, f^2_{2D}$, and $f_{3D}$
Function 2D MRL ENCODER $(g^1_{2D}, g^2_{2D})$
    $\mathbf{E}^1 = f^1_{2D}\ (g^1_{2D}), \quad \mathbf{E}^1 = f^2_{2D}\ (g^2_{2D})$
    $\mathbf{I}_{ij} = \mathsf{sim}(\mathbf{E}^1_i, \mathbf{E}^2_j)$
    where $\mathsf{sim}(\cdot, \cdot)$ is cosine similarity
    $\tilde{\mathbf{E}}^1 = \mathbf{I} \cdot \mathbf{E}^2, \quad \tilde{\mathbf{E}}^2 = \mathbf{I}^\top \cdot \mathbf{E}^1$
    $\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1), \quad \mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
    $\mathbf{z}^1_{2D} = \text{Set2set}(\mathbf{H}^1), \quad \mathbf{z}^2_{2D} = \text{Set2set}(\mathbf{H}^2)$
    return $\mathbf{z}^1_{2D}, \mathbf{z}^2_{2D}, \mathbf{H}^1, \mathbf{H}^2$
Function VIRTUAL INTERACTION GEOMETRY CONSTRUCTION $(g^1_{3D}, g^2_{3D})$
    Randomly select $n$ atoms in larger molecule $g^1_{3D}$
    Copy small molecule $g^2_{3D}$ to $n$ small molecules $g^{2,1}_{3D}, \ldots, g^{2,i}_{3D}, \ldots, g^{2,n}_{3D}$
    Generate a normalized random Gaussian noise vector $\varepsilon$
    Create new 3D coordinates for each smaller molecule $g^{2,i}_{3D}$
      $\mathbf{R}^{2,i} = \mathbf{R}^2 + \varepsilon_i * r^2 + \mathbf{R}^1_i$              /* Broadcasting operation */
    Create virtual interaction geometry $g_{vr}$
      $\mathbf{R}_{vr} = (\mathbf{R}^1 \| \mathbf{R}^{2,1} \| \ldots \| \mathbf{R}^{2,i} \| \ldots \| \mathbf{R}^{2,n})$
      $\mathbf{X}_{vr} = (\mathbf{X}^1 \| \mathbf{X}^2 \| \ldots \| \mathbf{X}^2)$
      $g_{vr} = (\mathbf{X}_{vr}, \mathbf{R}_{vr})$
    return $g_{vr}$
Function SE(3) INVARIANT GLOBAL GEOMETRY LEARNING $(\mathbf{z}_{2D}, \mathbf{z}_{3D})$
    return $\mathcal{L}_{glob} = -\frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \left[ \log \frac{e^{\mathtt{sim}(\mathbf{z}_{2D,i}, \mathbf{z}_{3D,i})/\tau}}{\sum_{k=1}^{N_{batch}} e^{\mathtt{sim}(\mathbf{z}_{2D,i}, \mathbf{z}_{3D,k})/\tau}} + \log \frac{e^{\mathtt{sim}(\mathbf{z}_{3D,i}, \mathbf{z}_{2D,i})/\tau}}{\sum_{k=1}^{N_{batch}} e^{\mathtt{sim}(\mathbf{z}_{3D,i}, \mathbf{z}_{2D,k})/\tau}} \right]$
Function SE(3) EQUIVARIANT LOCAL RELATIVE GEOMETRY LEARNING $(g^{2,i}_{3D}, \mathbf{H}^1, \mathbf{H}^2)$
    For all edges $(k, l)$ in $g^{2,i}_{3D}$:
      $\mathcal{F}_{k,l} = \left( \frac{\mathbf{r}_k - \mathbf{r}_l}{||\mathbf{r}_k - \mathbf{r}_l||}, \frac{\mathbf{r}_k \times \mathbf{r}_l}{||\mathbf{r}_k \times \mathbf{r}_l||}, \frac{\mathbf{r}_k - \mathbf{r}_l}{||\mathbf{r}_k - \mathbf{r}_l||} \times \frac{\mathbf{r}_k \times \mathbf{r}_l}{||\mathbf{r}_k \times \mathbf{r}_l||} \right)$,       /* Construct Orthogonal Frame */
      where $\mathbf{r}_k \in \mathbb{R}^3$ indicates the position of atoms $k$.
      $\mathbf{e}^{k,l}_{3D} = \text{Projection}_{\mathcal{F}_{k,l}}(\mathbf{r}_k, \mathbf{r}_l)$          /* Convert to $SE(3)$-Invariant Feature */
      $\mathbf{e}^{k,l}_{2D} = \text{MLP}(\mathbf{H}^2_k || \mathbf{H}^2_l)$
      $\mathbf{e}_{k,l} = \mathbf{e}^{k,l}_{2D} + \mathbf{e}^{k,l}_{3D}$.
    $\tilde{\mathbf{X}} = (\mathbf{H}^2 || \mathbf{H}^1_i)$                               /* Broadcasting operation */
    $\mathbf{h}_{k,l} = \text{GNN}(\tilde{\mathbf{X}}, \mathcal{E})$, where $\mathcal{E}$ indicates all edges in $g^{2,i}_{3D}$      /* Obtain Edge Features */
    $\hat{f}_k = \sum_l \mathbf{h}_{k,l} \odot \mathcal{F}_{k,l}$                    /* Convert to $SE(3)$-equivariant Feature */
    return $\mathcal{L}_{local} = \frac{1}{N^2} \sum_{k=1}^{N^2} ||f^i_k - \hat{f}^i_k||^2_2$
```