A Probability Contrastive Learning Framework for 3D Molecular Representation Learning

Jiayu Qin
University at Buffalo
jiayuqin@buffalo.edu

Jian Chen
University at Buffalo
jchen378@buffalo.edu

Rohan Sharma
University at Buffalo
rohanjag@buffalo.edu

Jingchen Sun
University at Buffalo
jsun39@buffalo.edu

Changyou Chen
University at Buffalo
changyou@buffalo.edu

Abstract

Contrastive Learning (CL) plays a crucial role in molecular representation learning, enabling unsupervised learning from large scale unlabeled molecule datasets. It has inspired various applications in molecular property prediction and drug design. However, existing molecular representation learning methods often introduce potential false positive and false negative pairs through conventional graph augmentations like node masking and subgraph removal. The issue can lead to suboptimal performance when applying standard contrastive learning techniques to molecular datasets. To address the issue of false positive and negative pairs in molecular representation learning, we propose a novel probability-based contrastive learning (CL) framework. Unlike conventional methods, our approach introduces a learnable weight distribution via Bayesian modeling to automatically identify and mitigate false positive and negative pairs. This method is particularly effective because it dynamically adjusts to the data, improving the accuracy of the learned representations. Our model is learned by a stochastic expectation-maximization process, which optimizes the model by iteratively refining the probability estimates of sample weights and updating the model parameters. Experimental results indicate that our method outperforms existing approaches in 13 out of 15 molecular property prediction benchmarks in MoleculeNet dataset and 8 out of 12 benchmarks in the QM9 benchmark, achieving new state-of-the-art results on average.

1 Introduction

We investigate the problem of learning representations from molecules, a field known as molecular representation learning (MRL). MRL has gained significant attention due to its critical role in enabling learning from limited supervised data for applications such as molecular property prediction [1,2,3] and drug design [4,5,6]. Molecular representation learning involves creating models that can derive meaningful and generalizable representations of molecules, which can then be used to enhance various downstream applications. Among the most common methods in MRL is contrastive learning (CL), which leverages large-scale unlabeled molecular datasets to learn robust representations. CL works by contrasting different augmentations of the same molecule to ensure that the model learns to recognize the essential features of the molecule, thereby improving performance on tasks such as molecular property prediction and drug design.

With the success of contrastive learning methods in computer vision and multi-modality pretraining [7,8], various contrastive learning approaches have been proposed for molecular representation learning. MolCLR [9] introduces a contrastive learning framework specifically for molecular represen-

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

tation learning. It employs atom masking and edge removal as data augmentations, which enhances the performance of Graph Neural Network (GNN) models on a variety of downstream molecular property prediction benchmarks. In contrast, GraphMVP [10] incorporates both 2D topology and 3D geometry during pre-training, though its downstream tasks primarily utilize 2D topology. These methods highlight different strategies for applying contrastive learning to molecular data, focusing on unique aspects of molecular structures to improve learning efficacy.

Although existing works have demonstrated the success of contrastive learning in molecular property predictions, they still face a significant drawback: the reliability of "positive" and "negative" labels in augmented molecule pairs. For example, MolCLR [9] uses augmentations like atom masking and edge removal, which can lead to false negative pairs when molecules with similar structures and chemical properties are labeled as negatives. Similarly, GraphMVP [10], which incorporates both 2D topology and 3D geometry, can also mislabel structurally similar augmented molecules as negatives due to its augmentation processes. These augmentations often remove parts of the molecular graph, such as nodes, edges, and subgraphs, resulting in potentially incorrect pairings. This issue is exacerbated by the large volume and extensive augmentations applied to molecular datasets, naturally leading to numerous falsely aligned pairs.

The fundamental problem lies in the random nature of these augmentations. Existing molecular contrastive learning methods assign hard positive and negatives to molecule pairs and do not account for the probabilistic relationships between molecules. Figure 3 provides an example of false positives and negatives resulting from graph augmentations in MolCLR [9] ,where two distinct graph augmentations are applied to enhance two different molecules. The augmented molecule pair originating from the same molecule is categorized as positive, while other molecule pairs within the same batch are considered negative. However, as illustrated in the figure, the correct contrastive learning setup should consider molecules with structural similarities as positive pairs, even when they originates from different molecules. In contrast, the same molecule subjected to different augmentation methods may also be considered negative due to structural dissimilarities. Existing methods like MolCLR [9] fail to maintain this distinction, where augmented pairs from the same molecule are always treated as positive, while pairs from different molecules within the same batch are always treated as negative, regardless of their structural similarity. This mislabeling results in false positives and negatives, undermining the effectiveness of the contrastive learning process.

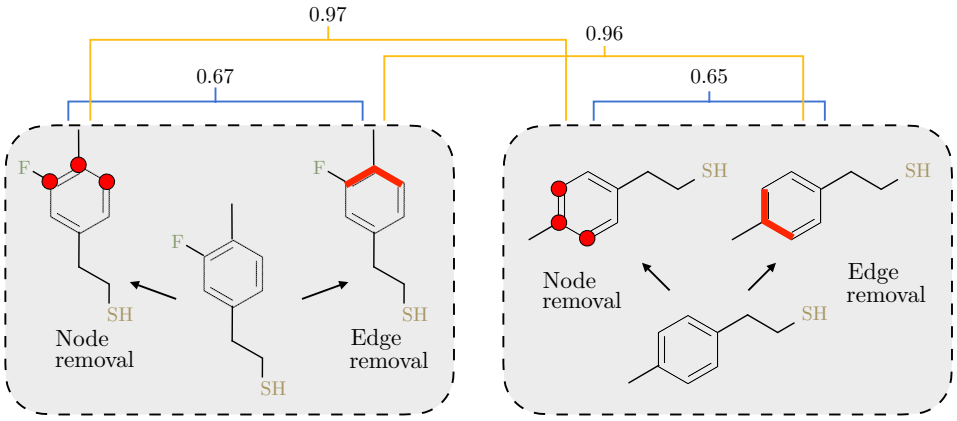

Figure 1: **Existing problem in molecular contrastive learning.** Adopt node removal and edge removal for molecular contrastive learning can lead to false positive and false negative problems. Blue lines indicate positive pairs and yellowing lines indicate negative pairs. The numbers on each line indicate the chemical similarity between the augmented pair of molecules. In this case, positive pairs indeed have lower similarity than negative pairs.

To overcome the aforementioned issue, we introduce a generalization of existing contrastive learning frameworks for molecular representation learning with probabilistic modeling. Our approach introduces data-pair weights as additional random variables, and dynamically infers optimal weights to account for false positive and false negative pairs, which can effectively address the mislabeling problem in previous methods. By incorporating a probability framework, we can effectively manage the uncertainty in data pair assignments. Specifically, we introduce a novel Bayesian inference

methods with Bayesian data augmentation to automatically infer these weights through posterior sampling. This allows us to optimize the model parameters efficiently using stochastic expectation maximization.

It is worth mentioning that while MolCLR [9] authors introduced i-MolCLR [45] to address similar issues by penalizing faulty negatives with a fingerprint-based similarity metric and a motif-level data augmentation called fragment contrast, our method offers distinct advantages. Unlike i-MolCLR which relies on direct fingerprint similarity, our approach introduces a novel probabilistic contrastive learning framework. This framework dynamically infers weight distributions and optimizes through stochastic expectation maximization, eliminating the need for explicit Tanimoto similarity calculations. Our method addresses the issue of false negative pairs more fundamentally and efficiently, providing a more robust solution for molecular contrastive learning.

In addition, our method is flexible and can be applied to different molecular representation learning framework. In this paper, we first integrate our method into MolCLR [9] series model and benchmark the performance on 2D non-charality MoleculeNet [11] dataset. We then integrated our method into Uni-Mol [21] and evaluate its performance on MoleculeNet [11]. We also trained and evaluated our model on the QM9 [44] dataset, following Equiformer [46]. With molecular property prediction tasks, we aim to test our model's ability in extracting useful features from molecular. Extensive experiments show that our method outperforms all other molecular representation learning baselines, including contrastive and non-contrastive methods.

The contributions of this paper can be summarized as follows:

- To tackle the challenges posed by false positive and negative pairs, we introduce a probability method for molecular contrastive learning. By introducing different weights as random variables to various false positive and negative pairs, we effectively mitigate the impact of these erroneous pairs on the learning process.
- To optimize our probabilistic contrastive learning framework, we propose a novel and effective optimization algorithm based on Bayesian data augmentation and stochastic expectation maximization, to simultaneously perform posterior inference and model optimization.
- Through extensive and large-scale experiments, we demonstrate enhanced performance across multiple public benchmarks for molecular representation learning, validating the effectiveness of our proposed method.

2 Methods

2.1 Learning Representations from Molecular Graphs

We begin by elucidating the foundational setup and notation in molecular contrastive learning. Molecules can be represented as 2D or 3D graphs depending on datasets. 2D molecule graphs have atoms as nodes and bond as edges. 3D molecule graphs additionally adds spacial positions of the atoms. For simplicity, we adopt static atom positions in this paper.

In molecular representation learning, as illustrated in Figure 2, we start by randomly sampling a batch of N molecules. Each molecule, represented as $\mathbf{x}_i$, undergoes stochastic augmentation strategies to generate two augmented versions, denoted as $(\mathbf{x}_i, \mathbf{x'}_i)$. These augmentations involve methods such as atom masking, edge perturbation, and subgraph removal, transforming the original molecular structure while preserving its core characteristics. Among the resulting $2N$ augmented molecules, each pair $(\mathbf{x}_i, \mathbf{x'}_i)$ is treated as a positive pair, while the remaining $2(N-1)$ augmented molecules within the same batch are considered negative samples. This setup allows us to utilize contrastive learning effectively by distinguishing between similar and dissimilar molecular structures. A neural network encoder $f(\mathbf{x}; \boldsymbol{\theta})$, parameterized by θ, is employed to extract representation vectors z from the augmented molecular samples. In this paper, we utilize three different types of encoders in various experiments, as depicted in Figure 2 B, C, and D. These encoders include Graph Neural Networks (GNNs) and Transformers, each providing unique advantages for capturing the intricate features of molecular structures.

Let $s_{i^+} \triangleq \mathrm{sim}(\mathbf{z}_i, \mathbf{z'}_i)$ represent the similarity score between the positive pair $(\mathbf{x}_i, \mathbf{x'}_i)$ after the encoder, and $s_{ik^-} \triangleq \mathrm{sim}\left(\mathbf{z}_i, \mathbf{z}_k\right)$ signifies the similarity score between the negative pair $(\mathbf{x}_i, \mathbf{x}_k)$, and $\mathrm{sim}(\cdot, \cdot)$ represents any positive-valued similarity metric. In this paper, we adopt the commonly

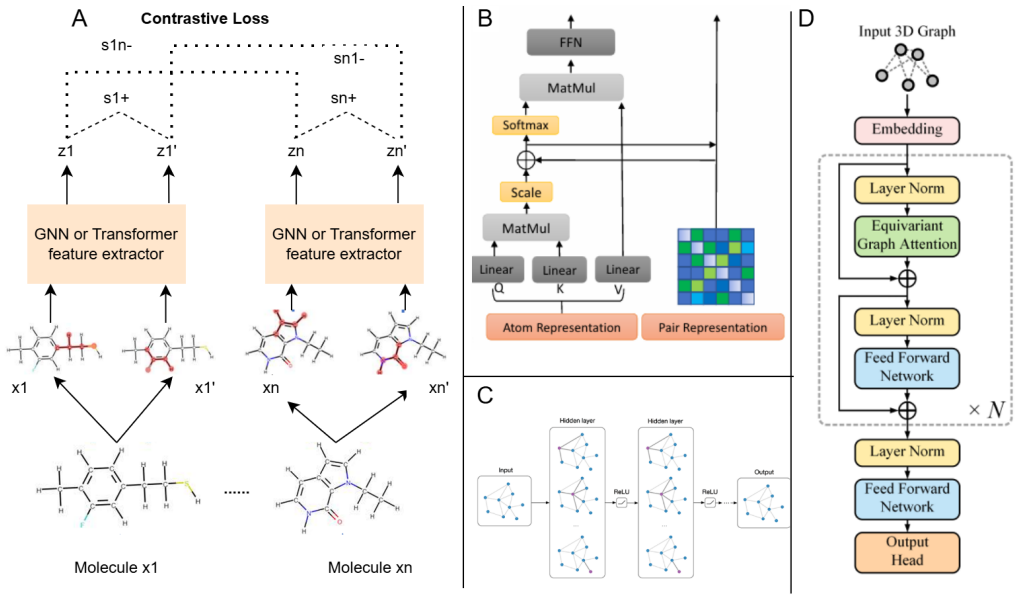

Figure 2: (A) **Molecular contrastive learning** Molecules are represented as 2D or 3D molecule graphs. Two stochastic augmentation strategies are applied to each graph, resulting in two augmentations. A feature extractor is used to extract features and contrastive loss is used to maximize the similarity of positive pairs and minimize the similarity of negative pairs B,C,D: Different architectures used as feature extractors in different experiments. (B) Uni-Mol [21] architecture used in MoleculeNet [11] Dataset experiment. (C) GCN [50] architecture from MolCLR [9] used in Non-Chirality MoleculeNet [11] experiment. (D) Equiformer [46] architecture used in QM9 [44] dataset experiment.

used exponential cosine similarity, defined as $\text{sim}(\mathbf{z}_1, \mathbf{z}_2) \triangleq e^{\mathbf{z}_1^T \mathbf{z}_2 / \|\mathbf{z}_1\| \|\mathbf{z}_2\| \tau}$, where τ denotes a temperature parameter.

2.2 Probability Weighted Contrastive Learning

We describe the proposed probability framework for molecular contrastive learning. In standard contrastive learning, one tries to encode data samples to a latent space such that positive pairs stay close to each other while negative pairs are pushed away. The contrastive loss function is:

$$\mathcal{L} = \frac{1}{N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)], \text{ with } \ell(i,j) = -\log \frac{s_{i+}}{s_{i+} + \sum_{k=1}^{2N} \mathbb{I}_{[k \neq i,j]} s_{i,k-}}$$

As mentioned, one issue of directly applying the contrastive learning into molecular representation learning is the potential false positive positive and negative molecular pairs, as discussed in the introduction. This could confuse the learning, ending up with sub-optimal representations. Is there a way to automatically identify and differentiate these pair data? In the following, we propose a Bayesian approach to address this issue that allows the algorithm for automatic inference of the degree of positiveness and negativeness of data pairs, involving enhancing the standard contrastive loss by incorporating learnable stochastic weights for all data pairs. To be more specific, we introduce local learnable weights, denoted as w_i^+ for each positive pair and w_{ik}^- for each negative pair. We then define a weighted contrastive loss based on these introduced weights. This modification aims to mitigate the issues by automatically assigning relatively lower weights (or no weights) to false positive and false negative pairs;

$$\mathcal{L}_w = \frac{1}{N} \sum_{k=1}^{N} [\bar{\ell}(2k-1, 2k) + \bar{\ell}(2k, 2k-1)], \quad \bar{\ell}(i,j) = -\log \frac{w_i^+ s_{i+}}{w_i^+ s_{i+} + \sum_{k=1}^{2N} \mathbb{I}_{[k \neq i,j]} w_{ik}^- s_{ik-}} \quad (1)$$

One problem with this formulation, however, is that it is not realistic to compute and store all the weights in the learning process. This precaution arises from the quadratic growth in the number of

weights to be calculated as the training data size increases. Furthermore, the random nature of our augmentation method further adds complexity to the pre-calculation and storage of these weights.

A straightforward baseline for calculating these weights can be envisioned as follows: we can consider these weights in a binary fashion, with all weights initialized to one. In the learning process, if for some positive pairs the similarity score falls below a specified threshold, we set the corresponding weights to zero, marking these positive pairs as false positives. Conversely, if for some negative pairs the similarity score exceeds a threshold, we set the associated weights to zero, indicating false negatives. A challenge associated with this baseline method, however, lies in the establishment of a rigid similarity threshold to create a binary division of weights between zero and one. This approach proves less suitable for our molecular contrastive task as these heuristically chosen thresholds might not be optimal.

To address this challenge, we propose a principled Bayesian approach that allows adaptively inferring the optimal weights by Bayesian inference. Specifically, we treat the weights to be random variables and assign appropriate priors to them. We consider two types of priors: a Bernoulli prior to model weights as binary random variables and a Gamma prior to represent them as positive values. For simplicity, we model positive weights using the Gamma distribution and negative weights using either the Gamma distribution or the Bernoulli distribution, as expressed by the following formulas:

Option 1 - Gamma priors for continuous weighting:

$$w_i^+ \sim \text{Gamma}(a_+, b_+), w_{ik}^- \sim \text{Gamma}(a_-, b_-).$$

Option 2 - Bernoulli priors for selective weighting:

$$w_i^+ \sim \text{Gamma}(a_+, b_+), \quad w_{ik}^- \sim \text{Bernoulli}(\bar{a_-}).$$

here, a_+, b_+, a_- and $b_-$are shape and rate parameters for Gamma distribution and $\bar{a_-}$ is the probability parameter for Bernoulli distribution.

With our reformulation, we can define a joint distribution over the global model parameter and local random weight variables $w_i^+$and w_{ik}^-, as:

$$p\left(\left\{w_i^+\right\}, \left\{w_{ik}^-\right\}, \boldsymbol{\theta}; \mathcal{D}\right) \propto \prod_{\mathbf{x}_i \in \mathcal{D}} \frac{w_i^+ s_{i^+}}{w_i^+ s_{ij^+} + \sum_{k=1}^K w_{ik}^- s_{ik^-}} p(\{w_i^+\}) p(\{w_{ik}^-\}) p(\boldsymbol{\theta}). \tag{2}$$

One problem with the above formulation, however, is that posterior inference of the weights is challenging, due to the lack of convenience posterior distributions.

Fortunately, inspired by [27], we can introduce an augmented random variable u_i that is associated to data point $\mathbf{x}_i$. Consequently, we can define an augmented joint posterior distribution of the random variables $\boldsymbol{\theta}, \mathbf{u}, \mathbf{w}$, denoted as $p\left(\left\{w_i^+\right\}, \left\{w_{ik}^-\right\}, \boldsymbol{\theta} \mid \mathcal{D}\right)$ [1], to be

$$p(\boldsymbol{\theta}, \mathbf{u}, \mathbf{w} \mid \mathcal{D}) \propto \prod_{i:\mathbf{x}_i \in \mathcal{D}} w_i^+ s_i + e^{-\mathbf{u}_i w_i^+ s_i +} \prod_k e^{-u_i w_{ik}^- s_{ik}-} p\left(\left\{w_i^+\right\}\right) p\left(\left\{w_{ik}^-\right\}\right) p(\boldsymbol{\theta}), \tag{3}$$

where $\mathbf{u} \triangleq \left\{u_1, u_2, \cdots, u_{|\mathcal{D}|}\right\}$ and $\mathbf{w} \triangleq \left\{w_i^+\right\} \cup \left\{w_{ik}^-\right\}$. It is worth noting that this joint distribution is equivalent to the original distribution (2), because (2) is recovered if one marginalize out the auxiliary random variables $\mathbf{u}$ in (3). In other words, optimization thought (3) is equivalent to optimization over (2). Consequently, we can perform learning and inference based on the augmented posterior of $p(\boldsymbol{\theta}, \mathbf{u}, \mathbf{w} \mid \mathcal{D})$, which preserves a much convenient form for posterior inference. In the following, we propose an efficient algorithm based on stochastic expectation maximization (stochastic EM) to alternatively infer the local random variables $\mathbf{w}$ and optimize the global model parameter $\boldsymbol{\theta}$.

2.3 Efficient Inference and Learning with Stocastic Expectation Maximization

We propose a stochastic EM algorithm for efficient inference and learning of our model. Stochastic EM [31] is a stochastic variant of the EM algorithm, which is an iterative method for finding the

[1] In the sense that marginalizing over the augmented random variables $\left\{w_i^+\right\}$and $\left\{w_{ik}^-\right\}$in $p\left(\theta, \mathbf{U}, \left\{w_i^+\right\}, \left\{w_{ik}^-\right\} \mid \mathcal{D}\right)$ gives back to the original $p\left(\left\{w_i^+\right\}, \left\{w_{ik}^-\right\}, \boldsymbol{\theta}; \mathcal{D}\right)$. Thus, learning and inferences on the two forms are equivalent.

maximum likelihood of model parameters in statistical models when data is only partially, or when model depends on unobserved latent variables [35].

In our setting, the objective of stocastic EM is to maximize the posterior in equation 4. The basic idea is to alternatively 1) optimizing model parameter $\boldsymbol{\theta}$ with fixed $(\mathbf{u}, \mathbf{w})$ and 2) sampling $(\mathbf{u}, \mathbf{w})$ with fixed $\boldsymbol{\theta}$. To this end, we follow standard procedures in stochastic EM to divide the learning into three steps: Simulation, Stochastic Expectation, and Maximization. Specifically, simulation corresponds to sampling local random variables $\mathbf{u}$ and w for a batch of data; stochastic expectation then uses the sampled auxiliary random variables to update the model parameter $\boldsymbol{\theta}$ by maximizing a stochastic objective $Q(\boldsymbol{\theta})$, defined as: $Q_{t+1}(\boldsymbol{\theta}) = Q_t(\boldsymbol{\theta}) + \lambda_t \left(\log p(\boldsymbol{\theta}, \mathbf{u}, \mathbf{w} \mid \mathcal{D}) - Q_t(\boldsymbol{\theta})\right)$ at iteration $t+1$, where $\{\lambda_t\}$ is a sequence of decreasing weights. And maximization corresponds to maximizing the stochastic objective constructed in the previous step. In the following, we detail the three steps.

Simulation Given the joint posterior distribution in equation 3 and the current batch of data, the posterior distributions of the local random variables $\mathbf{u}$ and $\mathbf{w}$ can be directly read out, which simply follow Gamma or Bornoulli distributions of the following forms:

$$
\begin{aligned}
&u_i \mid \left\{w_i^+, w_{ik}^-, \boldsymbol{\theta}\right\} \sim \text{Gamma}\left(a_u, b_u + w_i^+ s_{i^+} + \sum w_{ik}^- s_{ik^-}\right), \forall i, \text{ and} \\
&w_i^+ \mid \{\mathbf{u}, \boldsymbol{\theta}\} \sim \text{Gamma}\left(1 + a_+, u_i s_{i+} + b_+\right), \text{and} \\
&\text{Option 1: } w_{ik}^- \mid \{\mathbf{u}, \boldsymbol{\theta}\} \sim \text{Gamma}\left(a_-, u_i s_{ik^-} + b_-\right), \forall i, k \\
&\text{Option 2: } w_{ik}^- \mid \{\mathbf{u}, \boldsymbol{\theta}\} \sim \text{Bernoulli}\left(\frac{a_- e^{-u_i s_{ik-}}}{1 - a_- + a_- e^{-u_i s_{ik-}}}\right)
\end{aligned}
$$

Stochastic Expectation We then proceed to calculate the stochastic expectation based on the simulated local random variables above. For notation simplicity, we define $Q_0(\boldsymbol{\theta}) = 0$. Then we can reformulate $Q_{t+1}(\boldsymbol{\theta})$ by decomposing the recursion, resulting in

$$
Q_{t+1}(\boldsymbol{\theta}) = \sum_{\tau=0}^{t} \tilde{\lambda}_\tau \log p\left(\boldsymbol{\theta}, \mathbf{u}_\tau, \mathbf{w}_\tau \mid \mathcal{D}_\tau\right), \text{ where } \tilde{\lambda}_\tau \triangleq \lambda_\tau \prod_{t'=\tau+1}^{t} \left(1 - \lambda_{t'}\right), \tag{4}
$$

where τ indexes the minibatch and the corresponding local random variables at the current time τ.

Maximization The stochastic expectation objective (4) provides a convenient form for stochastic optimization over time, similar to online optimization (Bent & Van Hentenryck, 2005). Specifically, at each time t, we can initialize the parameter θ from the last step, and update it by stochastic gradient ascent on the log-likelihood, $\log p\left(\boldsymbol{\theta}, \mathbf{u}_\tau, \mathbf{w}_\tau \mid \mathcal{D}_\tau\right)$ calculated from the current batch of data. To reduce variance, we propose to optimize a marginal version by integrating out $\mathbf{u}_\tau$ from $p\left(\boldsymbol{\theta}, \mathbf{u}_\tau, \mathbf{w}_\tau \mid \mathcal{D}_\tau\right)$, which essentially reduces to our original weighted contrastive loss in equation (1). With the above steps, it is ready to optimize the model by stochastic EM. The detailed steps are described in the Algorithm 1.

Algorithm 1 Contrastive Learning with Stochastic EM

```
Initialize θ; set t = 1
for a batch of molecules in loader do
    Augment each molecule x_i into a pair (x_i, x'_i)
    Calculate positive/negative similarity scores s^+ and s^- for all the molecule pairs
    Initialize all the weights w^+ and w^- to be one
    for k = 1 to iter [4 in practice] do
        Sample u and w according to distributions
    end for
    Calculate the weighted contrastive loss in equation 2 with the sampled w on the current batch of data
    Update the model parameter by stochastic gradient descent with the calculated weighted contrastive loss
    t = t + 1
end for
```

3 Related works

Contrastive Learning As a popular self-supervised learning paradigm, contrastive learning focuses on learning semantically informative representations for downstream tasks [13-16]. The most widely used loss function is InfoNCE [17] which pulls in the representations between positive sample pairs while pushing away that between negative sample pairs.

Molecular Representation Learning Representation learning on large-scale unlabeled molecules attracts much attention recently. SMILES-BERT [18] is pretrained on SMILES strings of molecules using BERT. Subsequent works are mostly pretraining on 2D molecular topological graphs [19,20]. MolCLR [9] applies data augmentation to molecular graphs at both node and graph levels, using a self-supervised contrastive learning strategy to learn molecular representations. Further, several recent works try to leverage the 3D spatial information of molecules, and focus on contrastive or transfer learning between 2D topology and 3D geometry of molecules. For example, GraphMVP [10] proposes a contrastive learning GNN-based framework between 2D topology and 3D geometry. GEM [22] uses bond angles and bond length as additional edge attributes to enhance 3D information. Uni-Mol [21] is a universal 3D molecular pretraining framework that significantly enlarges the representation ability and application scope in drug design.

Noisy Pairs in Contrastive Learning Noisy data pair problem have been found and studied in visual contrastive learning community. NLIP [28] enforces the pairs with larger noise probability to have fewer similarities in embedding space to improve the model training. [29] apply noise estimation component to adjust the consistency between different modalities for the action recognition task. RINCE [30] uses a ranked ordering of positive samples to improve InfoNCE loss.

Stochastic Expectation Maximization Stochastic EM [31] stands as a pivotal algorithm in machine learning and probabilistic modeling for large-scale Bayesian inference. Building upon the foundations of the classical Expectation-Maximization (EM) algorithm [32], Stochastic EM offers an efficient solution for parameter estimation in situations involving vast datasets or latent variables, e.g., to maximize the $\log$-likelihood of $p(\mathbf{z}, \mathcal{D} \mid \boldsymbol{\theta})$, where $\mathcal{D}$ is the dataset, $\mathbf{z}$ is the local random variable and $\boldsymbol{\theta}$ is the global model parameter. By leveraging the power of mini-batch sampling, Stochastic EM strikes a balance between computational scalability and estimation accuracy. It has found widespread utility in various domains, including clustering [33], topic modeling [34], and latent variable modeling [35], making it an indispensable tool to cope with complex probabilistic models and extensive data and a natural fit to our problem.

4 Experiments

We evaluate our method on molecular property prediction tasks. Our approach is designed to be a versatile component that can be seamlessly integrated with various molecular property prediction datasets and models. In this study, we integrate our model into three different existing models: Uni-Mol [21], I-MolCLR [45], Equiformer [46] and assess its performance on three distinct datasets: MoleculeNet [11], MoleculeNet without chirality, and the QM9 [44] dataset. For all experiments, we provide detailed experiment settings in Appendix C.

4.1 The MoleculeNet Dataset

MoleculeNet [11] is a popular benchmark for molecular property prediction, including datasets focusing on different molecular properties, from quantum mechanics and physical chemistry to biophysics and physiology. For a fair comparison, we integrated our method into Uni-Mol [21] framework. We applied both the *Gamma* and *Bernoulli* versions of our method, as shown in Table 1. In our contrastive learning framework, we used the representation of the [CLS] token as the final encoded representation, representing the entire molecule. Additionally, we incorporated the original three-dimensional recovery loss as an extra loss function. The model was trained on the same large-scale dataset, including 19 million molecules and 209 million conformations, as in the original paper. We used the same evaluation metrics: ROC_AUC for classification tasks and RMSE and MAE for regression tasks.

As shown in Table 1 and 2, our method outperforms Uni-Mol [21] and GEM [22], the current state-of-the-art methods, with an average gain of 1.3 percent in classification tasks and 7.6 percent in regression tasks. This substantiates that our approach facilitates more flexible training with a higher tolerance for false positive and false negative data pairs, thereby enhancing the model's performance in molecular representation learning.

4.2 Non-Chirality version MoleculeNet

In order to make a fair comparison with I-MolCLR [45], we also integrated our method into MolCLR [9] framework. MolCLR and I-MolCLR are 2D based methods, their experiments are conducted on different version of MoleculeNet dataset that does not consider chirality. We adopted the same dataset, augmentation, GNN-based encoder and other settings. As shown in Table 3, our method

Table 1: ROC_AUC on molecular property prediction classification tasks (Higher is better)

Datasets	BBBP	BACE	ClinTox	Tox21	ToxCast	SIDER	HIV	PCBA	MUV
# Molecules	2039	1513	1478	7831	8575	1427	41127	437929	93078
# Tasks	1	1	2	12	617	27	1	128	17
D-MPNN [52]	71.0	80.9	90.6	75.9	65.5	57.0	77.1	86.2	78.6
Attentive FP [53]	64.3	78.4	84.7	76.1	63.7	60.6	75.7	80.1	76.6
N-Gram$_{RF}$[54]	69.7	77.9	77.5	74.3	–	66.8	77.2	–	76.9
N-Gram$_{XGB}$[54]	69.1	79.1	87.5	75.8	–	65.5	78.7	–	74.8
PretrainGNN [55]	68.7	84.5	72.6	78.1	65.7	62.7	79.9	86.0	81.3
GraphMVP [10]	72.4	81.2	79.1	75.9	63.1	63.9	77.0	–	77.7
GEM [22]	72.4	85.6	90.1	78.1	69.2	**67.2**	80.6	86.6	81.7
MolCLR [9]	72.2	82.4	91.2	75.0	–	58.9	78.1	–	79.6
Uni-Mol [21]	72.9	85.7	**91.9**	79.6	69.6	65.9	80.8	88.5	82.1
Ours (Gamma)	**76.7**	**88.2**	89.4	**80.1**	**69.9**	63.6	**83.0**	**89.6**	79.0
Ours (Bernoulli)	73.7	84.3	85.3	79.8	68.8	64.9	80.8	89.3	**82.9**

Table 2: Performance on molecular property prediction regression tasks (Lower is better)

Datasets	ESOL	FreeSolv	Lipo	QM7	QM8	QM9	MEAN (RMSE)	MEAN (MAE)
# Molecules	1128	642	4200	6830	21786	133885		
# Metric	RMSE↓			MAE↓				
D-MPNN [52]	1.050	2.082	0.683	103.5	0.0190	0.00814	1.272	34.509
GROVERlarge [56]	0.895	2.272	0.823	92.0	0.0224	0.00986	1.33	30.67
MolCLR [9]	1.271	2.594	0.691	66.8	0.0178	-	1.519	-
GraphMVP [10]	1.029	-	0.681	-	-	-	-	-
GEM [22]	0.798	1.877	0.660	58.9	0.0171	0.00746	1.112	19.642
Uni-Mol [21]	0.788	1.480	0.603	41.8	0.0156	0.00467	0.957	13.940
Ours (Gamma)	0.775	1.420	**0.590**	**38.5**	**0.0142**	**0.00395**	0.928	**12.839**
Ours (Bernoulli)	**0.664**	**1.358**	0.626	55.6	0.0154	0.0056	**0.883**	18.541

outperforms I-MolCLR on 7 out of 9 downstream tasks and got an average of 2 points increase on non-chirality MoleculeNet classification datasets.

Table 3: Comparison against i-MolCLR on non-chirality MoleculeNet dataset

Without Chirality	BBBP	BACE	ClinTox	Tox21	SIDER	HIV	MUV	MEAN
I-MOLCLR [45]	76.4	88.5	95.4	79.9	69.9	80.8	90.8	83.1
Our Method	**78.3**	**94.8**	91.4	**84.9**	**72.7**	**85.5**	88.0	**85.1**

4.3 QM9 Dataset

The QM9 dataset [44] is another popular dataset in molecular property prediction, it consists of 134k small molecules, and the goal is to predict their quantum properties. For this dataset, we choose equiformer [46] as a baseline method. The data partition we use has 110k,10k,and 11k molecules in training, validation and testing sets. We use both our contrastive loss function and original minimize mean absolute error(MAE) as training objectives.

As shown in 4, we get state of the art result in 8 out of 12 baselines. The increase is relatively subtle compared with other dataset, we argue that this is due to the fact that QM9 is relatively small regarding number of molecules in training set, and also the saturation on performance achieved by different methods.

4.4 Ablation Study

Distribution of similarity scores Our method is largely motivated by the observation that previous MCL approaches neglect potential semantic dissimilarity between positive samples and that accounting for this phenomenon can improve learned molecule representations. In Figure A(See Appendix A), we plot the distribution of similarity scores for both positive and negative samples. Figure A left reveals that our method yields larger similarity scores with lower variance for positive pairs compared to MolCLR [9] baseline which uses standard contrastive learning method. Figure A right reveals that our method also mitigates the false negative problem in standard CL. It also shows that our method sometimes assigns lower similarity scores to positive pairs. While it may seem counter intuitive to assign lower similarity scores to positive samples, we argue that doing so is the very reason our method captures dissimilarity between positive pairs. By allowing some degree of alignment between

Table 4: Experiment results on QM9 dataset

Methods	α	ΔE	E_homo	E_lumo	μ	Cv	G	H	R^2	μ	μ0	ZPVE
GraphCL [47]	0.066	45.5	26.8	22.9	0.027	0.028	10.2	9.6	0.095	9.7	9.6	1.42
JOAOv2 [48]	0.066	45.0	27.8	22.2	0.027	0.028	9.9	9.2	0.087	9.8	9.5	1.43
3D-MGP [49]	0.057	37.1	21.3	18.2	**0.020**	0.026	9.3	**8.7**	0.092	8.6	**8.6**	1.38
Transformer-M [50]	0.041	27.4	17.5	16.2	0.037	0.022	9.63	9.39	**0.075**	9.41	9.37	1.18
Equiformer [46]	0.046	30	**15**	14	0.011	0.023	7.63	6.63	**0.251**	6.74	6.59	1.26
Ours	**0.037**	**24.2**	21.1	**13.7**	0.022	**0.022**	**6.2**	**6.31**	0.082	**7.22**	9.40	**1.09**

the right set of negative examples, our method is able to minimize the inconsistencies between shared context of related positives and negatives. This in turn allows us to learn an overall more coherent representation space, resulting in increased robustness and downstream performance.

Comparisons with the Standard Contrastive Learning We conducted an ablation study to showcase that our method of probablistic framework of contrastive learning has already achieved strong emperical results and demonstrate the improvement brought by adding the 3D-aware loss functions on MoleculeNet [11] classification dataset. We first examined the effect of adding the probabilistic framework to the standard contrastive loss, and the 3D-aware loss functions as implemented in Uni-Mol [21].

Table 5: Ablation Study on MoleculeNet Classification Datasets

	BBBP	BACE	ClinTox	Tox21	ToxCast	SIDER	HIV	PCBA	MUV	MEAN
Standard CL	69.3	81.5	84.1	75.5	63.4	58.9	78.3	84.1	72.5	75.2
CL + 3D Loss	75.1	86.8	87.9	78.9	68.5	62.8	81.8	88.0	77.1	78.1
CL + Probabilistic Framework	74.1	86.3	88.2	79.5	68.2	63.1	82.5	88.4	77.1	78.6
CL + Both	**76.7**	**88.2**	**89.4**	**80.1**	**69.9**	**63.6**	**83.0**	**89.6**	**79.0**	**80.1**

Table 6 presents the results of our ablation study. Incorporating the probabilistic framework resulted in a great improvement of 3.4-point increase in ROC-AUC, significantly enhances the model's performance. On the other hand, introducing the additional loss component led to an increase in ROC-AUC by 2.9 points, demonstrating its secondary role in enhancing the model's performance. When we adopt both of them, we can get the final ROC-AUC of 80.1 average on MoleculeNet classification datasets.

Hyperparameters We also conducted an ablation study to determine the optimal hyperparameters (e.g., a_+, a_-) on MoleculeNet classification datasets. We selected a_+, a_-, b_+, and b_- from the range $[1, 5, 10]$. Table 6 indicates that our method achieves the best performance with $a_+ = 5$ and $a_- = b_+ = b_- = 1$. Tuning different hyperparameters affects performance, with an increase in a_+ from 1 to 5 leading to a 1.6 percent performance gain.

Table 6: Abalation studies on hyperparameters for MoleculeNet classification tasks

a_+	1	5	10	5	5	5	5
a_-	1	1	1	1	1	5	10
b+	1	1	1	5	10	5	5
b_-	1	1	1	1	1	5	10
Avg. ROC-AUC (%)	78.8	**80.4**	79.6	79.3	80.0	79.4	79.3

5 Conclusion

In this paper, we investigate an important yet unnoticeable limitation of molecular contrastive learning, where augmented graph data come with false positive and false negative data pairs. As a remedy, we propose a principled solution to molecular contrastive learning by reformulating it into a probability framework and introducing random weights for data pairs. With a Bayesian data augmentation technique, the random weights can be efficiently inferred via sampling, and the model parameter can be effectively optimized via stochastic expectation maximization.

The effectiveness of our innovative approach has been proven through rigorous evaluations on multiple molecular property prediction and protein-ligand binding pose benchmarks. The results also showcase the wide-ranging applicability and improved robustness of our proposed method over both standard contrastive learning method and non-contrastive learning method for learning molecular representations.

We believe our method is a valuable addition to the literature on molecular contrastive representation learning, which can further boost the performance of state-of-the-art molecular representation learning models for drug design.

References

[1] Rong Y, Bian Y, Xu T, et al. Self-supervised graph transformer on large-scale molecular data[J]. Advances in neural information processing systems, 2020, 33: 12559-12571.

[2] Wang Y, Wang J, Cao Z, et al. Molecular contrastive learning of representations via graph neural networks[J]. Nature Machine Intelligence, 2022, 4(3): 279-287.

[3] Fang X, Liu L, Lei J, et al. Geometry-enhanced molecular representation learning for property prediction[J]. Nature Machine Intelligence, 2022, 4(2): 127-134.

[4] Koukos P I, Xue L C, Bonvin A M J J. Protein–ligand pose and affinity prediction: Lessons from D3R Grand Challenge 3[J]. Journal of computer-aided molecular design, 2019, 33: 83-91.

[5] Liu S, Wang H, Liu W, et al. Pre-training molecular graph representation with 3d geometry[J]. arXiv preprint arXiv:2110.07728, 2021.

[6] Méndez-Lucio O, Ahmad M, del Rio-Chanona E A, et al. A geometric deep learning approach to predict binding conformations of bioactive molecules[J]. Nature Machine Intelligence, 2021, 3(12): 1033-1039.

[7] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.

[8] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PMLR, 2021: 8748-8763.

[9] Wang Y, Wang J, Cao Z, et al. Molecular contrastive learning of representations via graph neural networks[J]. Nature Machine Intelligence, 2022, 4(3): 279-287.

[10] Liu S, Wang H, Liu W, et al. Pre-training molecular graph representation with 3d geometry[J]. arXiv preprint arXiv:2110.07728, 2021.

[11] Wu Z, Ramsundar B, Feinberg E N, et al. MoleculeNet: a benchmark for molecular machine learning[J]. Chemical science, 2018, 9(2): 513-530.

[12] Koukos P I, Xue L C, Bonvin A M J J. Protein–ligand pose and affinity prediction: Lessons from D3R Grand Challenge 3[J]. Journal of computer-aided molecular design, 2019, 33: 83-91.

[13] Li Y, Yang M, Peng D, et al. Twin contrastive learning for online clustering[J]. International Journal of Computer Vision, 2022, 130(9): 2205-2221.

[14] Chuang C Y, Robinson J, Lin Y C, et al. Debiased contrastive learning[J]. Advances in neural information processing systems, 2020, 33: 8765-8775.

[15] You Y, Chen T, Sui Y, et al. Graph contrastive learning with augmentations[J]. Advances in neural information processing systems, 2020, 33: 5812-5823.

[16] Hu P, Zhu H, Lin J, et al. Unsupervised contrastive cross-modal hashing[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(3): 3877-3889.

[17] Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding[J]. arXiv preprint arXiv:1807.03748, 2018.

[18] Wang S, Guo Y, Wang Y, et al. Smiles-bert: large scale unsupervised pre-training for molecular property prediction[C]//Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics. 2019: 429-436.

[19] Li P, Wang J, Qiao Y, et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery[J]. Briefings in Bioinformatics, 2021, 22(6): bbab109.

[20] Rong Y, Bian Y, Xu T, et al. Self-supervised graph transformer on large-scale molecular data[J]. Advances in neural information processing systems, 2020, 33: 12559-12571.

[21] Zhou G, Gao Z, Ding Q, et al. Uni-mol: A universal 3d molecular representation learning framework[J]. 2023.

[22] Fang X, Liu L, Lei J, et al. Geometry-enhanced molecular representation learning for property prediction[J]. Nature Machine Intelligence, 2022, 4(2): 127-134.

[23] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[24] Hu W, Liu B, Gomes J, et al. Strategies for pre-training graph neural networks[J]. arXiv preprint arXiv:1905.12265, 2019.

[25] Li P, Wang J, Qiao Y, et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery[J]. Briefings in Bioinformatics, 2021, 22(6): bbab109.

[26] Ying C, Cai T, Luo S, et al. Do transformers really perform badly for graph representation?[J]. Advances in neural information processing systems, 2021, 34: 28877-28888.

[27] Chen C, Zhang J, Xu Y, et al. Why do we need large batchsizes in contrastive learning? a gradient-bias perspective[J]. Advances in Neural Information Processing Systems, 2022, 35: 33860-33875.

[28] Huang R, Long Y, Han J, et al. Nlip: Noise-robust language-image pre-training[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(1): 926-934.

[29] Han H, Zheng Q, Luo M, et al. Noise-tolerant learning for audio-visual action recognition[J]. IEEE Transactions on Multimedia, 2024.

[30] Hoffmann D T, Behrmann N, Gall J, et al. Ranking info noise contrastive estimation: Boosting contrastive learning via ranked positives[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(1): 897-905.

[31] Nielsen S F. The stochastic EM algorithm: estimation and asymptotic results[J]. Bernoulli, 2000: 457-489.

[32] Lin, D. (2011). An Introduction to Expectation-Maximization.

[33] Allassonnière S, Chevallier J. A new class of stochastic EM algorithms. Escaping local maxima and handling intractable sampling[J]. Computational statistics and data analysis, 2021, 159: 107159.

[34] Zaheer M, Wick M, Tristan J B, et al. Exponential stochastic cellular automata for massively parallel inference[C]//Artificial Intelligence and Statistics. PMLR, 2016: 966-975.

[35] Zhang S, Chen Y. Computation for latent variable model estimation: A unified stochastic proximal framework[J]. psychometrika, 2022, 87(4): 1473-1502.

[36] Berman H M, Westbrook J, Feng Z, et al. The protein data bank[J]. Nucleic acids research, 2000, 28(1): 235-242.

[37] Le Guilloux V, Schmidtke P, Tuffery P. Fpocket: an open source platform for ligand pocket detection[J]. BMC bioinformatics, 2009, 10: 1-11.

[38] Liu Z, Li Y, Han L, et al. PDB-wide collection of binding data: current status of the PDBbind database[J]. Bioinformatics, 2015, 31(3): 405-412.

[39] Trott O, Olson A J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading[J]. Journal of computational chemistry, 2010, 31(2): 455-461.

[40] Eberhardt J, Santos-Martins D, Tillack A F, et al. AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings[J]. Journal of chemical information and modeling, 2021, 61(8): 3891-3898.

[41] Quiroga R, Villarreal M A. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening[J]. PloS one, 2016, 11(5): e0155183.

[42] Koes D R, Baumgartner M P, Camacho C J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise[J]. Journal of chemical information and modeling, 2013, 53(8): 1893-1904.

[43] Morris G M, Huey R, Lindstrom W, et al. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility[J]. Journal of computational chemistry, 2009, 30(16): 2785-2791.

[44] Ramakrishnan R, Dral P O, Rupp M, et al. Quantum chemistry structures and properties of 134 kilo molecules[J]. Scientific data, 2014, 1(1): 1-7.

[45] Wang Y, Magar R, Liang C, et al. Improving molecular contrastive learning via faulty negative mitigation and decomposed fragment contrast[J]. Journal of Chemical Information and Modeling, 2022, 62(11): 2713-2725.

[46] Liao Y L, Smidt T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs[J]. arXiv preprint arXiv:2206.11990, 2022.

[47] You Y, Chen T, Sui Y, et al. Graph contrastive learning with augmentations[J]. Advances in neural information processing systems, 2020, 33: 5812-5823.

[48] You Y, Chen T, Shen Y, et al. Graph contrastive learning automated[C]//International Conference on Machine Learning. PMLR, 2021: 12121-12132.

[49] Jiao R, Han J, Huang W, et al. Energy-motivated equivariant pretraining for 3d molecular graphs[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(7): 8096-8104.

[50] Luo S, Chen T, Xu Y, et al. One transformer can understand both 2d 3d molecular data[C]//The Eleventh International Conference on Learning Representations. 2022.

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

[52] Yang K, Swanson K, Jin W, et al. Analyzing learned molecular representations for property prediction[J]. Journal of chemical information and modeling, 2019, 59(8): 3370-3388.

[53] Xiong Z, Wang D, Liu X, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism[J]. Journal of medicinal chemistry, 2019, 63(16): 8749-8760.

[54] Liu S, Demirel M F, Liang Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules[J]. Advances in neural information processing systems, 2019, 32.

[55] Hu W, Liu B, Gomes J, et al. Strategies for pre-training graph neural networks[J]. arXiv preprint arXiv:1905.12265, 2019.

[56] Rong Y, Bian Y, Xu T, et al. Self-supervised graph transformer on large-scale molecular data[J]. Advances in neural information processing systems, 2020, 33: 12559-12571.

A Similarity Score Distribution

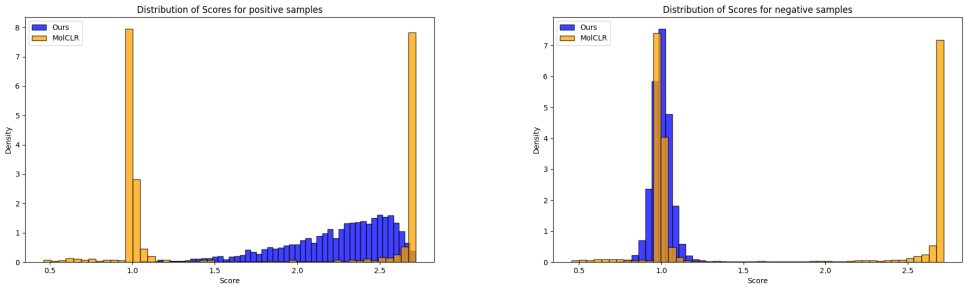

Figure 3: **Similarity Scores** – Similarity scores distribution for negative pairs in joint space after pre-training with original MolCLR loss and our proposed loss is provided. Compared to Using pretrained MolCLR model, our method yields similarity scores with lower mean and lower variance for negative pairs. While MolCLR have two peaks of negatives similarity scores around 1 and 2.7, our method concentrates them at only one peak of 1.Our method yields similarity scores with higher mean and lower variance for positive pairs. Our method concentrates at higher levels as it allows for some degree of semantic dissimilar between positives. The similarity scores are dot similarity, they are not normalized to enhance the difference for visual purposes.

B Limitations

In this section, we discuss the limitations of our proposed EM-based algorithm for molecular contrastive learning.

B.1 Assumptions and Robustness

Our approach relies on several strong assumptions, such as the independence of molecular features and the noisiness nature of the input data. In practice, these assumptions may be violated, potentially affecting the performance and robustness of the model. For instance, correlated features could lead to biased estimates of weights, while unnoisy data might degrade the necessity to apply our method in learning representations. Future work could explore methods to relax these assumptions and enhance the model's robustness to such violations.

B.2 Scope of Claims

The empirical results presented in this paper are based on experiments conducted on a specific set of datasets: MoleculeNet and QM9. While these datasets are commonly used in molecular machine learning research, they may not fully represent all possible application domains. Consequently, the generalizability of our findings to other datasets or real-world scenarios might be limited. Further validation on a broader range of datasets is necessary to confirm the wide applicability of our approach.

Also, one limitation of our method is that the performance gains brought by the proposed architectural improvements can depend on datasets and tasks. For small datasets like QM9, the performance gain is not significant.

B.3 Privacy and Fairness

While our work does not specifically address issues of privacy and fairness, these are important considerations for any machine learning model, especially those used in sensitive domains such as healthcare. The potential for bias in molecular datasets, as well as privacy concerns related to molecular data, are areas that require further exploration. Ensuring that our model adheres to ethical standards and mitigates bias is an avenue for future work.

By acknowledging these limitations, we aim to provide a transparent account of our research and encourage future studies to build upon and address these challenges.

Table 7: hyperparameter search space for MoleculeNet dataset

Hyperparameter	Small	Large	HIV
Learning rate	$[5e-5, 8e-5, 1e-4, 4e-4, 5e-4]$	$[2e-5, 1e-4]$	$[2e-5, 5e-5]$
Batch size	$[32, 64, 128, 256]$	$[128, 256]$	$[128, 256]$
Epochs	$[40, 60, 80, 100]$	$[20, 40]$	$[2, 5, 10]$
Pooler dropout	$[0.0, 0.1, 0.2, 0.5]$	$[0.0, 0.1]$	$[0.0, 0.2]$
Warmup ratio	$[0.0, 0.06, 0.1]$	$[0.0, 0.06]$	$[0.0, 0.1]$

C Training details for experiments

C.1 MoleculeNet dataset

We report the detailed hyperparameters setup of during pretraining in 7. Molecular pretraining runs on 4 A6000 GPUs, and the training time is about 48 hours.

C.2 MoleculeNet non-charality

In pre-training, the GNN encoder embeds each molecule graph into a 512-dimension representation $\boldsymbol{h}$. The projection head is modeled by an MLP with one hidden layer maps $\boldsymbol{h}$ into 256-dimensional latent vector $\boldsymbol{z}$. ReLU is implemented as the non-linear activation function. The whole model is pre-trained for 50 epochs with batch size 512 . We use Adam optimizer with an initial learning rate 5×10^{-4} and the weight decay 1×10^{-5}. Additionally. cosine learning rate decay is performed during pre-training.

During fine-tuning, we replace the projection head with a randomly initialized MLP which maps the representation $\boldsymbol{h}$ into the desired property prediction while keeping the pre-trained GNN encoder. The pre-trained model is trained individually for 100 epochs on each task from the benchmarks. We perform a random search of hyperparameters on validation sets and report the results on test sets. For each benchmark, we run three individual runs and report the average. The whole model is implemented on PyTorch Geometric.

C.3 QM9 dataset

We follow the data partitioning scheme used by Equiformer. For the tasks involving $\mu, \alpha, \varepsilon_{\text{HOMO}}, \varepsilon_{\text{LUMO}}, \Delta\varepsilon$, and C_ν, our experimental setup includes a batch size of 64, training for 300 epochs, a learning rate of 5×10^{-4}, and Gaussian radial basis functions with 128 bases. The architecture comprises 6 Transformer blocks, a weight decay of 5×10^{-3}, and a dropout rate of 0.2. Mixed precision training is employed for these tasks.

For the R^2 task, we use a batch size of 48, 300 epochs, a learning rate of 1.5×10^{-4}, Gaussian radial basis functions with 128 bases, 5 Transformer blocks, a weight decay of 5×10^{-3}, and a dropout rate of 0.1, training in single precision.

The ZPVE task also uses a batch size of 48, 300 epochs, a learning rate of 1.5×10^{-4}, Gaussian radial basis functions with 128 bases, 5 Transformer blocks, a weight decay of 5×10^{-3}, and a dropout rate of 0.2, with single precision training.

For the tasks of G, H, U, and U_0, the setup includes a batch size of 48, 300 epochs, a learning rate of 1.5×10^{-4}, Gaussian radial basis functions with 128 bases, 5 Transformer blocks, no weight decay, and no dropout, with single precision training.

We used a single A6000 GPU for training, with the mixed precision tasks taking 81 GPU-hours and single precision tasks taking 151 GPU-hours. The model contains 11.20 million parameters for 6-block configurations and 9.35 million parameters for 5-block configurations.

D Protein-ligand binding task

We also conducted the protein-ligand binding pose prediction task. This is one of the most important tasks in structure based drug design. The task is to predict the complex structure of a protein binding site and a molecular ligand. We need to consider how ligand lays in the pocket, that is, the 6 degrees (3 rotations and 3 translations) of freedom of a rigid movement.

Following Uni-Mol [21], the molecular representation and pocket representation are firstly obtained from their own pretraining models by their own conformations; then, their representations are concatenated as the input of an additional 4-layer Transformer decoder, which is finetuned to learn the pair distances of all heavy atoms in molecule and pocket. Then, with the predicted pair-distance matrix as a scoring function, we first randomly

place the ligand and then optimize the coordinates of its atoms by directly back-propagation the loss between current pair-distance and predicted pair-distance.

For the training data used in finetuning, we use PDBbind General set v.2020[38] (19,443 complexes).

We evaluate our method using the metric binding pose accuracy. Specifically, we keep the pocket conformation fixed, while the ligand conformation is fully flexible. We evaluate the RMSD(root mean squared distance) between the prediction and the ground truth. Following previous works, we use the percentage of results below predefined RMSD thresholds as metrics.

Table 8: Performance on binding pose prediction.

Methods	1.0 Å	1.5 Å	2.0 Å	3.0 Å	5.0 Å
Autodock Vina	44.21	57.54	64.56	73.68	84.56
Vinardo	41.75	57.54	62.81	69.82	76.84
Smina	47.37	59.65	65.26	74.39	82.11
Autodock4	21.75	31.58	35.44	47.02	64.56
Uni-Mol [21]	43.16	68.42	80.35	87.02	94.04
Ours (Bernoulli)	**48.77**	**70.18**	78.95	85.26	94.04
Ours (Gamma)	45.61	69.47	**80.70**	**88.42**	**96.84**

We compare our method with current state-of-the-art baselines, including Autodock Vina[39,40], Vinardo[41], Smina[42], Autodock4[43] and Uni-Mol[21].

The binding pose accuracy results are shown in Table 3. Not surprisingly, our model again outperforms all the baseline methods, achieving state-of-the-art results with our Gamma-prior version model.