A Group Symmetric Stochastic Differential Equation Model for Molecule Multi-modal Pretraining

Shengchao Liu [*1,2] **Weitao Du** [*3] **Zhiming Ma** [3] **Hongyu Guo** [4] **Jian Tang** [1,5]

Abstract

Molecule pretraining has quickly become the go-to schema to boost the performance of AI-based drug discovery. Naturally, molecules can be represented as 2D topological graphs or 3D geometric point clouds. Although most existing pertaining methods focus on merely the single modality, recent research has shown that maximizing the mutual information (MI) between such two modalities enhances the molecule representation ability. Meanwhile, existing molecule multi-modal pretraining approaches approximate MI based on the representation space encoded from the topology and geometry, thus resulting in the loss of critical structural information of molecules. To address this issue, we propose MoleculeSDE. MoleculeSDE leverages group symmetric (*e.g.*, SE(3)-equivariant and reflection-antisymmetric) stochastic differential equation models to generate the 3D geometries from 2D topologies, and vice versa, *directly* in the input space. It not only obtains tighter MI bound but also enables prosperous downstream tasks than the previous work. By comparing with 17 pretraining baselines, we empirically verify that MoleculeSDE can learn an expressive representation with state-of-the-art performance on 26 out of 32 downstream tasks. The source codes are available in this repository.

1. Introduction

Artificial intelligence (AI) for drug discovery has recently attracted a surge of research interest in both the machine learning and cheminformatics communities, demonstrating encouraging outcomes in many challenging drug discovery tasks (Gilmer et al., 2017; Gómez-Bombarelli et al., 2018; Liu et al., 2019; Corso et al., 2020; Jin et al., 2020; Rampášek et al., 2022; Liu et al., 2022; Jumper et al., 2021; Demirel et al., 2021; Hsu et al., 2022; Liu et al., 2023b; Yao et al., 2023). These successes are primarily attributed to the informative representations of molecules.

Molecules can be naturally represented as topological graphs, where atoms and covalent bonds are the nodes and edges. Additionally, molecular structures (a.k.a. *conformations*) can be treated as 3D geometric graphs, where the atoms are the point clouds in the 3D Euclidean space. Based on such two modalities, tremendous representation methods have been proposed in a supervised setting (Gilmer et al., 2017; Thomas et al., 2018). Further, by leveraging a large number of molecule datasets (Irwin et al., 2012; Hu et al., 2020b; Axelrod & Gomez-Bombarelli, 2022; Xu et al., 2021a), *molecule pretraining strategies* (Hu et al., 2019; Sun et al., 2022; Liu et al., 2022) have proven their effectiveness in learning robust and expressive molecule representations. However, most such pertaining works focus on exploring the 2D topology modality, and typical algorithms include reconstructing the masked substructures (Hu et al., 2019; Liu et al., 2019) and aligning the positive subgraph pairs and contrasting the negative pairs (Sun et al., 2020; You et al., 2020b; Wang et al., 2021) simultaneously. Recently, there have also been successful explorations (Liu et al., 2023a; Jiao et al., 2022) on the 3D geometric pretraining, where the key idea is to reconstruct the masked distances or coordinates through a group symmetric reconstruction operation.

Nevertheless, despite its shown potential for forming high-quality representations, multi-modal pretraining over the molecular 2D topologies and 3D conformations has been under-explored. GraphMVP (Liu et al., 2021a) is the first to build a unified multi-modal self-supervised learning (SSL) paradigm. It introduces contrastive and generative learning to estimate the mutual information (MI) between the two modalities. Contrastive learning treats the topologies and conformations as positive if and only if they are referred to the same molecules. It aims to align the positive pairs while distributing the negative pairs. Such a contrastive idea has also been studied in (Stärk et al., 2022). More encourag-

[*]Equal contribution [1]Mila - Québec AI Institute, Canada [2]Université de Montréal, Canada [3]Chinese Academy of Sciences, China [4]National Research Council Canada, Canada [5]HEC Montréal, Canada. Correspondence to: Shengchao Liu <liusheng@mila.quebec>, Weitao Du <duweitao@amss.ac.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

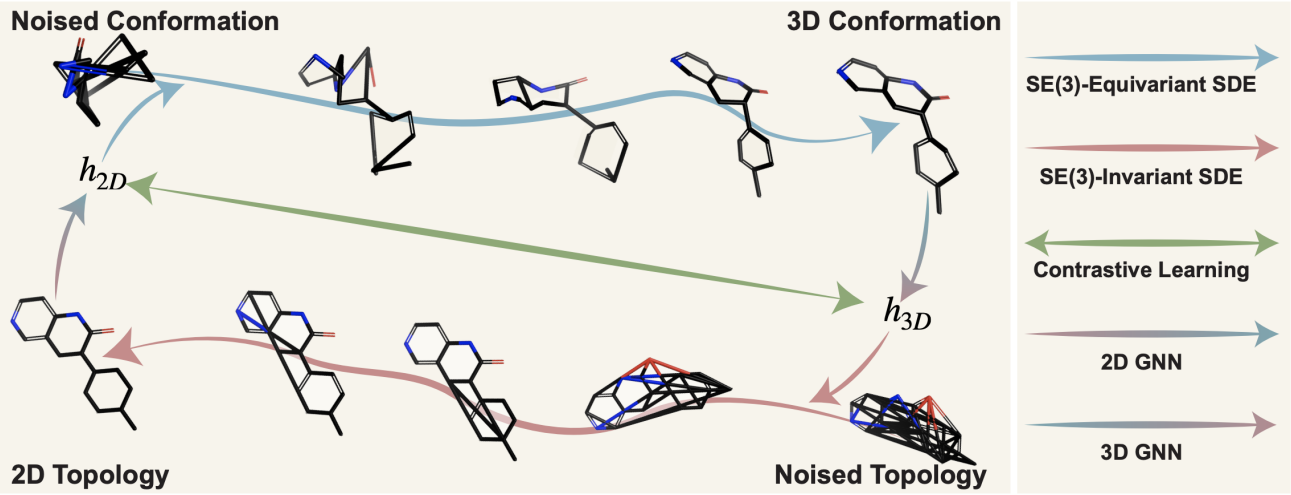


Figure 1. Illustration of the MoleculeSDE pretraining. It is composed of one contrastive learning and two generative learning objectives. Contrastive learning aims to align the 2D topological and 3D conformational representations for the same molecule. The two generative learning objectives are molecule conditional generation from 2D topology to 3D conformation and from 3D conformation to 2D topology, respectively. The generative modeling from topology to conformation is an SE(3)-equivariant diffusion process that satisfies the physical symmetry in molecule geometries. The other direction from conformation to topology is an SE(3)-invariant diffusion process since only the invariant type-0 features (nodes and edges) are considered. We further include a demo, showing the trajectory of this process.

ingly, GraphMVP demonstrates the benefits of generative SSL, which aims at reconstructing the conformations from the topologies (or vice versa) by introducing a proxy to the evidence lower bound of MI estimation, *i.e.*, the *variational representation reconstruction (VRR)*. VRR transforms the reconstruction on the data space (*i.e.*, molecular topologies or conformations) to the representation space that compresses input features. Hence, the use of such a proxy can result in the loss of important structural information of molecules.

Our Approach. To address the aforementioned issue, we propose MoleculeSDE, a multi-modal pretraining method on molecules' topologies and conformations. As illustrated in Figure 1, MoleculeSDE contains both contrastive and generative SSLs. The former adopts the EBM-NCE in (Liu et al., 2021a), and the latter leverages the stochastic differential equation (SDE) framework (Song et al., 2020; Du et al., 2022a). Such design brings in two main benefits. First, for the objective function, the generative loss in GraphMVP is a proxy to the MI, while the diffusion process in MoleculeSDE leads to a more accurate MI estimation with less information loss. We list a brief performance comparison of two methods in Table 1 and will provide theoretical insights that MoleculeSDE can lead to a more accurate MI estimation. Second, the SDE-based generative SSL enables prosperous downstream tasks. For example, MoleculeSDE enables conformation generation (CG) on tasks where only 2D topologies are available (Wu et al., 2018). Based on this, we can apply more advanced geometric modeling methods for prediction. As shown in Section 5, such generated conformations lead to improved predictive performance over existing CG methods (Shi et al., 2021; Du et al., 2022b).

Table 1. Downstream tasks' performance comparison with *merely* generative pretraining. The complete results are in Appendix H.

Model	Tox21 ↑	MUV ↑	Bace ↑	GAP ↓	U0 ↓	Aspirin ↓
VRR (GraphMVP)	73.6	75.5	72.7	44.64	13.96	1.177
SDE (MoleculeSDE)	**75.6**	**80.1**	**79.0**	**42.75**	**11.85**	**1.087**

The core components of the proposed MoleculeSDE are the two SDE generative processes. The first SDE aims to convert from topology to conformation. This conversion needs to satisfy the physical nature of molecules: the molecules' physical and chemical attributes need to be equivariant to the rotations and translations in the 3D Euclidean space, *i.e.*, the **SE(3)-equivariant and reflection-antisymmetric** property (Appendix B), and we use **SE(3)-equivariance** for short. We note that existing topology to conformation deep generative methods are either SE(3)-invariant (Shi et al., 2021) or not SE(3)-equivariant (Zhu et al., 2022). We propose an SE(3)-equivariant diffusion process by building equivariant local frames. The inputs of local frames are SE(3)-equivariant vector features (*e.g.*, the atom coordinates), and the local frames transform them into three SE(3)-invariant features, which will be transformed back to the equivariant data using tensorziation (Du et al., 2022b). The second SDE targets the conformation to topology reconstruction task in the discrete topological space. The main challenge for this task is to adopt the diffusion model for discrete data generation. We follow the recent work on graph diffusion generation (Jo et al., 2022), where the joint generation of atom and bond leads to better estimation.

To the best of our knowledge, we are the first to build an **SE(3)-equivariant and reflection-antisymmetric** SDE for the topology to conformation generation and also the first to

devise an SE(3)-invariant SDE for the conformation to topology generation for representation learning. We also note that our proposed MoleculeSDE is agnostic to the backbone representation methods since the SDE process is disentangled with the representation function, as illustrated in Figure 1.

Our main contributions include: (1) We propose a group symmetric pretraining method, MoleculeSDE, on the 2D and 3D modalities of molecules. (2) We provide theoretical insights on the tighter MI estimation of MoleculeSDE over previous works. (3) We show that MoleculeSDE enables prosperous downstream tasks. (4) We empirically verify that MoleculeSDE retains essential knowledge from both modalities, resulting in state-of-the-art performance on 26 out of 32 downstream tasks compared with 17 competitive baselines.

2. Related Work

Molecule SSL pretraining on a single modality. The pretraining on *2D molecular topology* shares common ideas with the general graph pretraining (Sun et al., 2022; Wang et al., 2022a). One classical approach (Hu et al., 2019; Liu et al., 2019) is to mask certain key substructures of molecular graphs and then perform the reconstruction in an auto-encoding manner. Another prevalent molecule pretraining method is contrastive learning (Oord et al., 2018), where the goal is to align the views from the positive pairs and contrast the views from the negative pairs simultaneously. For example, ContexPred (Hu et al., 2019) constructs views based on different radii of neighborhoods, Deep Graph InfoMax (Veličković et al., 2018) and InfoGraph (Sun et al., 2020) treat the local and global graph representations as the two views, MolCLR (Wang et al., 2022c) and GraphCL (You et al., 2020a) create different views using discrete graph augmentation methods.

Recent studies start to explore the *3D geometric pretraining* on molecules. GeoSSL (Liu et al., 2023a) proposes maximizing the mutual information between noised conformations using an SE(3)-invariant denoising score matching, and a parallel work (Zaidi et al., 2022) is a special case of GeoSSL using only one denoising layer. 3D-EMGP (Jiao et al., 2022) is also a parallel work, but it is E(3)-equivariant, which needlessly satisfies the reflection-equivariant constraint in molecular conformation distribution.

Molecule SSL pretraining on multiple modalities. The GraphMVP (Liu et al., 2021a) proposes one contrastive objective (EBM-NCE) and one generative objective (variational representation reconstruction, VRR) to optimize the mutual information between the topological and conformational modalities. Specifically for VRR, it is a proxy loss to the evidence lower bound (ELBO) by doing the reconstruction in the representation space, which may risk losing critical information. 3D InfoMax (Stärk et al., 2022) is a special case of GraphMVP, where only the contrastive loss is considered. Notice that MoleculeSDE solves the approximation issue in GraphMVP with two SDE processes.

3. Preliminaries

2D topological molecular graph. A topological molecular graph is denoted as $g_{2\mathrm{D}} = (\boldsymbol{X}, \boldsymbol{E})$, where $\boldsymbol{X}$ is the atom attribute matrix and $\boldsymbol{E}$ is the bond attribute matrix. The 2D graph representation with graph neural network (GNN) is:

$$\boldsymbol{H}_{2\mathrm{D}} = \text{GNN-2D}(T_{2\mathrm{D}}(g_{2\mathrm{D}})) = \text{GNN-2D}(T_{2\mathrm{D}}(\boldsymbol{X}, \boldsymbol{E})), \quad (1)$$

where $T_{2\mathrm{D}}$ is the data transformation on the 2D topology, and GNN-2D is the representation function. $\boldsymbol{H}_{2\mathrm{D}} = [h^0_{2\mathrm{D}}, h^1_{2\mathrm{D}}, \ldots]$, where $h^i_{2\mathrm{D}}$ is the i-th node representation.

3D conformational molecular graph. The molecular conformation is denoted as $g_{3\mathrm{D}} = (\boldsymbol{X}, \boldsymbol{R})$, where $\boldsymbol{R} = \{\boldsymbol{r}^1, \boldsymbol{r}^2, \ldots\}$ is the collection of 3D coordinates of atoms. The conformational representation is:

$$\boldsymbol{H}_{3\mathrm{D}} = \text{GNN-3D}(T_{3\mathrm{D}}(g_{3\mathrm{D}})) = \text{GNN-3D}(T_{3\mathrm{D}}(\boldsymbol{X}, \boldsymbol{R})), \quad (2)$$

where $T_{3\mathrm{D}}$ is the data transformation on the 3D geometry, and GNN-3D is the representation function. $\boldsymbol{H}_{3\mathrm{D}} = [h^0_{3\mathrm{D}}, h^1_{3\mathrm{D}}, \ldots]$, where $h^i_{3\mathrm{D}}$ is the i-th node representation. In our approach, we take the masking as the transformation for both 2D and 3D GNN, and the masking ratio is M. In what follows, we use $\boldsymbol{x}$ and $\boldsymbol{y}$ for the 2D and 3D graphs for notation simplicity, *i.e.*, $\boldsymbol{x} \triangleq g_{2\mathrm{D}}$ and $\boldsymbol{y} \triangleq g_{3\mathrm{D}}$.

SE(3)-Equivariance and Reflection-Antisymmetry. Two 3D geometric graphs $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are SE(3)-isometric if there exists an element $g \in \mathrm{SE}(3)$ such that $\boldsymbol{R}_2 = g\boldsymbol{R}_1$, where $g \in \mathrm{SE}(3)$ is a 3D rotation or translation acting on each node (atom) of $\boldsymbol{R}_1$. In this article, we will consider vector-valued functions defined on the 3D molecular graph. Specifically, given a conformation-related function $f(\boldsymbol{r}) : \mathbf{R}^3 \to \mathbf{R}^3$ on the graph $\boldsymbol{R}$, we say it's equivariant if

$$f(g\boldsymbol{r}) = gf(\boldsymbol{r}) \quad (3)$$

for arbitrary $g \in \mathrm{SE}(3)$. Since different chiralities can lead to different chemical properties (Clayden et al., 2012; Cotton, 1991), the function $f(\boldsymbol{r})$ we consider in this article is SE(3)-equivariant and reflection-antisymmetric.

Stochastic Differential Equation (SDE). The score-based generative modeling with stochastic differential equations (SDEs) (Song et al., 2020) provides a novel and expressive tool for distribution estimation. It is also a united framework including denoising score matching (Vincent, 2011; Song & Ermon, 2019) and denoising diffusion (Ho et al., 2020). In general, these methods can be split into two processes: the forward and backward processes. The forward process is a parameter-free deterministic process, and it diffuses a data point $\boldsymbol{x}$ into a random noise by adding noises, as

$$d\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)dt + g(t)dw_t, \quad (4)$$

with $f(\boldsymbol{x}_t, t)$ the vector-value drift coefficient, $g(t)$ the diffusion coefficient, and w_t the Wiener process. Note that Equation (4) induces a family of densities $\boldsymbol{x}_t \sim p_t(\cdot)$. On the other hand, the backward process generates real data from the stationary distribution of Equation (4) by evolving along the following SDE:

$$d\boldsymbol{x}_t = [f(\boldsymbol{x}_t, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)]dt + g(t)dw_t. \quad (5)$$

Thus, the learning objective is to estimate $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$. This derivative term is called *score* in the literature of score matching (Vincent, 2011; Song & Ermon, 2019; Song et al., 2020). We will use this framework in MoleculeSDE to generate 3D conformation and 2D topology.

4. The MoleculeSDE Method

In Section 4.1, we first provide the mutual information (MI) aspect in molecule multi-modal pretraining, and then we present the limitation of VRR. We discuss two SDE models as generative pretraining in Sections 4.2 and 4.3, respectively. The ultimate learning objective and inference process are illustrated in Section 4.4. Additionally in Section 4.5, we provide theoretical insights on how MoleculeSDE obtains a more accurate MI estimation.

4.1. An Overview from Mutual Information Perspective

Mutual information (MI) measures the non-linear dependency between random variables, and it has been widely adopted as the principle for self-supervised pretraining (Oord et al., 2018; Hjelm et al., 2018). The expectation is that, by maximizing the MI between modalities, the learned representation can keep the most shared information. Thus, MI-guided SSL serves as an intuitive and powerful framework for representation pretraining.

GraphMVP (Liu et al., 2021a) transforms the MI maximization into the summation of two conditional log-likelihoods:

$$\mathcal{L}_{\text{MI}} = \frac{1}{2}\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x}|\boldsymbol{y})\right]. \quad (6)$$

GraphMVP (Liu et al., 2021a) further solves Equation (6) by proposing a contrastive loss (EBM-NCE) and a generative loss (variational representation reconstruction, VRR). VRR conducts the reconstruction on the representation space, *i.e.*, from 2D (3D) data to the 3D (2D) representation space (details in Appendix F). The main advantage of VRR is its simple implementation without topology or conformation reconstruction, yet the trade-off is that it can lose critical information since it is only a proxy solution to generative learning.

Thus, to this end, we raise one question: *If there exists a more accurate conditional density estimation method for generative pretraining?* The answer is yes, and we propose MoleculeSDE. MoleculeSDE utilizes two stochastic differential (SDE) models to estimate Equation (6). SDE is a broad generative model class (Karras et al., 2022) where a neural network is used to model the score (Song & Kingma, 2021) of various levels of noise in a diffusion process (Ho et al., 2020). To adapt it in MoleculeSDE, we propose an SDE from 2D topology to 3D conformation (Section 4.2) and an SDE from 3D conformation to 2D topology (Section 4.3). We also want to highlight that such reconstructions are challenging, as both the 2D topologies and 3D conformations are highly structured: the 2D topologies are permutation invariant, and the 3D conformations additionally obey the SE(3)-equivariance.

MoleculeSDE has three main advantages. (1) MoleculeSDE is a powerful generative pretraining method, as the SDE models have shown promising performance in applications including image generation (Ho et al., 2020; Song et al., 2020) and geometric representation (Liu et al., 2023a; Zaidi et al., 2022). (2) MoleculeSDE is a more accurate estimation to Equation (6) than previous methods. Thus, the pretrained representation contains more critical information with a more accurate MI estimation. (3) MoleculeSDE enables prosperous downstream tasks, like topology to conformation generation for property prediction (Section 5.4).

4.2. An SE(3)-Equivariant Conformation Generation

The first objective we consider is the conditional generation from topology to conformation, $p(\boldsymbol{y}|\boldsymbol{x})$. One thing to highlight is that the molecule 3D conformation needs to satisfy the physical property, *i.e.*, it needs to be equivariant to the rotation and transition in the 3D Euclidean space, which is known as the *SE(3)-equivariance and reflection-antisymmetry* property (Appendix B). Notice that for notation simplicity, we may call it *SE(3)-equivariance*, and only expand into details in the "Local frame" paragraph below.

The core module in SDE is the score network, $S_\theta^{\text{2D}\to\text{3D}}$. To satisfy the physical nature of molecule 3D structure, such a score network needs to be SE(3)-equivariant. Specifically, the input includes the 2D graph $\boldsymbol{x}$, the noised 3D information $\boldsymbol{y}_t$ at time t, and the time t. The output is the SE(3)-equivariant 3D scores at time t, accordingly. The goal is to use $S_\theta^{\text{2D}\to\text{3D}}$ to estimate the score $\nabla \log p_t(\boldsymbol{y}_t|\boldsymbol{x})$.

To learn $p(\boldsymbol{y}|\boldsymbol{x})$, we formulate it as solving an SDE problem. Then based on the score network, the training objective is:

$$\mathcal{L}^{\text{2D}\to\text{3D}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\mathbb{E}_t\mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{y}} \left[\left\|\nabla_{\boldsymbol{y}_t}\log p_t(\boldsymbol{y}_t|\boldsymbol{y},\boldsymbol{x}) - S_\theta^{\text{2D}\to\text{3D}}(\boldsymbol{x},\boldsymbol{y}_t,t)\right\|_2^2\right]. \quad (7)$$

Coordinate reconstruction. Notice that both 2D topologies and 3D conformations share the same atom information (atom types), so in this subsection specifically, by reconstructing $\boldsymbol{y}$, we are referring to reconstructing the coordi-

nates $\boldsymbol{R}$. Thus, the objective function becomes:

$$\mathcal{L}_{\text{2D}\to\text{3D}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{R}}\mathbb{E}_t\mathbb{E}_{\boldsymbol{R}_t|\boldsymbol{R}} \left[\left\|\nabla_{\boldsymbol{R}_t}\log p_t(\boldsymbol{R}_t|\boldsymbol{R},\boldsymbol{x}) - S_\theta^{\text{2D}\to\text{3D}}(\boldsymbol{x},\boldsymbol{R}_t,t)\right\|_2^2\right], \tag{8}$$

Local frame. Before going into details of the score network, we want to introduce the SE(3)-equivariant & reflection-antisymmetric local frame (Appendix B). Such equivariant frames are introduced to fill in the gap between invariant 2D features and the output SE(3) vector field. It is equivalent to a 3D coordinate system that transforms equivariantly with the geometric graph. Through equivariant frames, we can project noised 3D coordinates into invariant scalars (isomers are projected differently), such that they are ready to be combined with invariant 2D features. On the other hand, by projecting back the 2D graph's invariant predictions into an equivariant frame, our final output can transform equivariantly with respect to global rotation and translation. We leave the precise formulations of our local frames in Appendix B. Briefly in MoleculeSDE, we focus on the equivariant frame attached on each edge $(\boldsymbol{r}^i, \boldsymbol{r}^j)$:

$$\boldsymbol{t}_{\text{frame}}^{ij} = \text{Local-Frame}(\boldsymbol{r}^i, \boldsymbol{r}^j). \tag{9}$$

SE(3)-equivariant score network. Then we introduce how to build an SE(3)-equivariant score network based on the local frame. We first concat the atom representations h_{2D} into the atom pairwise representations, as $e_{\text{2D}}^{ij} = \text{MLP}(\text{concat}\{h_{\text{2D}}^i \,||\, h_{\text{2D}}^j\})$ for the i-th and j-th atoms. Then the 2D pairwise representations are further added to the 3D pairwise representations $e_{\text{3D}}^{ij} = \text{projection}_{\boldsymbol{t}_{\text{frame}}^{ij}}(\boldsymbol{r}^i, \boldsymbol{r}^j)$, produced by the equivariant frames. Based on such frame feature, the final **invariant** edge feature e^{ij} is defined by:

$$e^{ij} = \text{rbf}(r^{ij}) \odot e_{\text{2D}}^{ij} + e_{\text{3D}}^{ij}, \tag{10}$$

where r^{ij} denotes the relative distance between the i-th and j-th atoms, and we use the radial basis function (RBF) to embed such distance features. Note that the input 3D coordinates and the corresponding distance matrix $\{r^{ij}\}$ are based on the diffused positions at a given diffusion step, rather than the ground truth 3D conformation.

Then we process e^{ij} through multiple graph attention layers (Shi et al., 2020): $h^{ij} = \text{Attention}(e^{ij})$. Finally, by pairing the invariant aggregated edge features h^{ij} with our SE(3)-equivariant frames $\boldsymbol{t}_{\text{frame}}^{ij}$, we get the vector-valued score function: $S(\boldsymbol{r}^i) = \sum_j h^{ij} \odot \boldsymbol{t}_{\text{frame}}^{ij}$. Here, our equivariant construction guarantees that the output vector field is SE(3)-equivariant and reflection-antisymmetric.

Discussions. We want to clarify the following points between molecule geometric modeling (h_{3D}) and score network ($S_\theta^{\text{2D}\to\text{3D}}$). (1) The score network proposed here is SE(3)-equivariant and reflection-antisymmetric. The input to the score network is the topology and diffused conformation, so it cannot be shared with the molecule geometric modeling, where the input is the ground-truth 3D conformation. (2) To the best of our knowledge, we are the first to propose the SE(3)-equivariant and reflection-antisymmetric SDE for the topology to conformation generation task.

4.3. An SE(3)-Invariant Topology Generation

The second objective is to reconstruct the 2D topology from 3D conformation, *i.e.*, $p(\boldsymbol{x}|\boldsymbol{y})$. Note that the 2D topology information (atoms and bonds) belongs to the type-0 feature (Thomas et al., 2018), thus such a generative process should satisfy the SE(3)-invariance property. If we formulate it as an SDE problem, then the training objective is:

$$\mathcal{L}^{\text{3D}\to\text{2D}} = \mathbb{E}_{\boldsymbol{y},\boldsymbol{x}}\mathbb{E}_t\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}} \left[\left\|\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{x},\boldsymbol{y}) - S_\theta^{\text{3D}\to\text{2D}}(\boldsymbol{y},\boldsymbol{x}_t,t)\right\|_2^2\right], \tag{11}$$

where $S_\theta^{\text{3D}\to\text{2D}}$ is the score network.

SE(3)-invariant score network. For modeling $S_\theta^{\text{3D}\to\text{2D}}$, it needs to satisfy the SE(3)-invariance symmetry property. The inputs are 3D conformational representation $\boldsymbol{y}$, the noised 2D information $\boldsymbol{x}_t$ at time t, and time t. The output of $S_\theta^{\text{3D}\to\text{2D}}$ is the invariant 2D score function at time t, as $\nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$. As introduced in Section 3, the diffused 2D information contains two parts: $\boldsymbol{x}_t = (\boldsymbol{X}_t, \boldsymbol{E}_t)$, so the corresponding forward SDE is a joint variant of Equation (4):

$$\begin{cases} \mathrm{d}\boldsymbol{X}_t = f_{1,t}(\boldsymbol{X}_t, \boldsymbol{E}_t)dt + g_1(t)\mathrm{d}w_t^1, \\ \mathrm{d}\boldsymbol{E}_t = f_{2,t}(\boldsymbol{X}_t, \boldsymbol{E}_t)dt + g_2(t)\mathrm{d}w_t^2, \end{cases} \tag{12}$$

where w_t^1 and w_t^2 are two independent Brownian motion. Then the score network $S_\theta^{\text{3D}\to\text{2D}}$ is also decomposed into two parts for the atoms and bonds: $S_\theta^{\boldsymbol{X}_t}(\boldsymbol{x}_t)$ and $S_\theta^{\boldsymbol{E}_t}(\boldsymbol{x}_t)$.

Similar to the topology to conformation generation procedure, we first merge the 3D representation $\boldsymbol{z}_y$ with the diffused atom feature $\boldsymbol{X}_t$ as $H_0 = \text{MLP}(\boldsymbol{X}_t) + \boldsymbol{H}_{\boldsymbol{y}}$. Then we apply a GCN as the score network to estimate the node-level score, as $S_\theta^{\boldsymbol{X}_t}(\boldsymbol{x}_t) = \text{MLP}(\text{concat}\{H_0||\cdots||H_L\})$, where $H_{i+1} = \text{GCN}(H_i, \boldsymbol{E}_t)$ and L is the number of GCN layer. On the other hand, the edge-level score is modeled by an unnormalized dot product attention $S_\theta^{\boldsymbol{E}_t}(\boldsymbol{x}_t) = \text{MLP}(\{\text{Attention}(H_i)\}_{0\le i\le L})$.

4.4. Learning and Inference of MoleculeSDE

Learning. In addition to the two generative objectives, we also consider a contrastive loss, EBM-NCE (Liu et al., 2021a). EBM-NCE can be viewed as another way to approximate the mutual information $I(X;Y)$, and it is expected to be complementary to the generative SSL. Therefore, our final objective is

$$\mathcal{L}_{\text{MoleculeSDE}} = \alpha_1\mathcal{L}_{\text{Contrastive}} + \alpha_2\mathcal{L}^{\text{2D}\to\text{3D}} + \alpha_3\mathcal{L}^{\text{3D}\to\text{2D}}, \tag{13}$$

where $\alpha_1, \alpha_2, \alpha_3$ are three coefficient hyperparameters.

Inference. After we train the SDE model from 2D topologies to 3D conformations, we can generate 3D conformations from a 2D topology by 'reversing' the forward SDE. More precisely, we take the Predictor-Corrector sampling method (Song et al., 2020) as tailored to our continuous framework. Further, generating 3D conformations from 2D topologies enable us to conduct prosperous downstream tasks such as property prediction with 2D and 3D data.

4.5. Theoretical Insights of MoleculeSDE

Since the two terms in Equation (6) are in the mirroring direction, here we take $\boldsymbol{x}|\boldsymbol{y}$ for theoretical illustrations. The other direction can be obtained similarly. We adapt the continuous diffusion framework proposed in (Song et al., 2020), in which the Markov chain denoising generative model (DDPM) is included as a discretization of the continuous Ornstein–Uhlenbeck process. The diffusion framework originates from the noised score-matching scheme (Vincent, 2011) of training the energy-based model (EBM), in which the authors introduced a noised version of $p(\boldsymbol{y})$ by adding noise to each data point $\tilde{\boldsymbol{y}} = \boldsymbol{y} + \epsilon$, where ϵ is sampled from a scaled normal distribution. Then,

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{p(\tilde{\boldsymbol{y}})} \left\| \nabla_{\boldsymbol{y}} \log p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) - \nabla_{\boldsymbol{y}} \log p_{\theta}(\tilde{\boldsymbol{y}}|\boldsymbol{x}) \right\|_2^2 \qquad (14) \\ =&\min_{\theta} \mathbb{E}_{p(\tilde{\boldsymbol{y}},\boldsymbol{y})} \left\| \nabla_{\boldsymbol{y}} \log p(\tilde{\boldsymbol{y}}|\boldsymbol{y},\boldsymbol{x}) - \nabla_{\boldsymbol{y}} \log p_{\theta}(\tilde{\boldsymbol{y}}|\boldsymbol{x}) \right\|_2^2 + C, \end{aligned}$$

where the conditional score function $\nabla_{\boldsymbol{y}} \log p(\tilde{\boldsymbol{y}}|\boldsymbol{y},\boldsymbol{x})$ is analytically tractable. The diffusion generative model further pushes the one-step (14) to a continuous noising process from raw data $\boldsymbol{y}$ to $\boldsymbol{y}_t$ for $0 \leq t \leq T$. We call $\boldsymbol{y}_t$ the nosing (forward) diffusion process starting at y, which is usually formulated as the solution of a stochastic differential equation. Then, the corresponding (continuous) score matching loss is:

$$\min_{\theta} \mathbb{E}_t \mathbb{E}_{\boldsymbol{y}} \mathbb{E}_{\boldsymbol{y}_t|\boldsymbol{y}} \left\| \nabla_{\boldsymbol{y}_t} \log p_t(\boldsymbol{y}_t|\boldsymbol{y},\boldsymbol{x}) - \nabla_{\boldsymbol{y}_t} \log p_{\theta,t}(\boldsymbol{y}_t|\boldsymbol{x}) \right\|_2^2. \qquad (15)$$

It's worth mentioning that the weighted continuous score matching is equivalent to learning the infinitesimal reverse of the noising process from t to $t+\Delta t$ for each time t, which greatly reduces the difficulty of recovering p_{data} from the white noise in one shot (Ho et al., 2020).

To make a connection between Equation (15) and Equation (6), it's crucial to relate score matching with **maximal log-likelihood** method. To solve this problem, (Huang et al., 2021) defined a key quantity (ELBO) $\mathcal{E}_{\theta}^{\infty}(\boldsymbol{y}|\boldsymbol{x})$ as a functional on the infinite-dimensional path space (consists of all stochastic paths starting at $\boldsymbol{y}$). Then, the authors show that

$$\mathbb{E}_{p(\boldsymbol{y})} \log p_{\theta,T}(\boldsymbol{y}|\boldsymbol{x}) \geq \mathbb{E}_{p(\boldsymbol{y})} \mathcal{E}_{\theta}^{\infty}(\boldsymbol{y}|\boldsymbol{x}), \qquad (16)$$

where the probability $p_{\theta,T}(\boldsymbol{y}|\boldsymbol{x})$ corresponds to the marginal distribution of a parameterized (denoted by θ) SDE at time T. Moreover, the ELBO $\mathbb{E}_{p(\boldsymbol{y})} \mathcal{E}_{\theta}^{\infty}(\boldsymbol{y}|\boldsymbol{x})$ is equivalent to the score matching loss. Therefore, training the diffusion model is equivalent to maximizing a lower bound of the likelihood defined by SDEs. Since the variational capacity of the infinite-dimensional SDE space is larger than previous models, we expect to find a better estimation of Equation (6).

5. Experiments

MoleculeSDE enables both a pretrained 2D and 3D representation and can be further fine-tuned toward the downstream tasks. Meanwhile, another main advantage of MoleculeSDE is that it also learns an SDE model from topology to conformation. Such a design enables us to adopt more versatile downstream tasks. For instance, there is a wide range of molecular property prediction tasks (Wu et al., 2018) considering only the 2D topology, yet the 3D conformation has proven to be beneficial towards such property prediction tasks (Liu et al., 2021a). Thus, with the pretrained generative model $p(\boldsymbol{y}|\boldsymbol{x})$, we can generate the corresponding 3D structure for each molecule topology and apply the pretrained 3D encoders, which is expected to improve the performance further. A visual illustration of such three categories of downstream tasks is in Figure 2.

5.1. Pretraining and Baselines

Dataset. For pretraining, we use PCQM4Mv2 (Hu et al., 2020a). It's a sub-dataset of PubChemQC (Nakata & Shimazaki, 2017) with 3.4 million molecules with both the topological graph and geometric conformations. We are aware of the Molecule3D (Xu et al., 2021b) dataset, which is also extracted from PubChemQC (Nakata & Shimazaki, 2017). Yet, after confirming with the authors, certain mismatches exist between the 2D topologies and 3D conformations. Thus, in this work, we use PCQM4Mv2 for pretraining.

Baselines for 2D topology pretraining. Enormous 2D topological pretraining methods have been proposed (Liu et al., 2021c; Xie et al., 2021; Wu et al., 2021; Liu et al., 2021b). Recent works (Sun et al., 2022) re-explore the effects of these pretraining methods, and we pick up the most promising ones as follows. AttrMask (Hu et al., 2019; Liu et al., 2019), ContexPred (Hu et al., 2019), Deep Graph Infomax (Veličković et al., 2018) and InfoGraph (Sun et al., 2020), MolCLR (Wang et al., 2022c) and GraphCL (You et al., 2020a). The detailed explanations are in Section 2.

Baselines for 3D conformation pretraining. The 3D conformation SSL pretraining has been less explored. We adopt the comprehensive baselines from (Liu et al., 2023a). The type prediction, distance prediction, and angle prediction predict the masked atom type, pairwise distance, and triplet angle, respectively. The 3D InfoGraph predicts whether the node- and graph-level 3D representation are for the same molecule. RR, InfoNCE, and EBM-NCE are to max-

Table 2. Results for molecular property prediction tasks (with 2D topology only). For each downstream task, we report the mean (and standard deviation) ROC-AUC of 3 seeds with scaffold splitting. The best and second best results are marked **bold** and **bold**, respectively.

Pre-training	BBBP ↑	Tox21 ↑	ToxCast ↑	Sider ↑	ClinTox ↑	MUV ↑	HIV ↑	Bace ↑	Avg ↑
– (random init)	68.1±0.59	75.3±0.22	62.1±0.19	57.0±1.33	83.7±2.93	74.6±2.35	75.2±0.70	76.7±2.51	71.60
AttrMask	65.0±2.36	74.8±0.25	62.9±0.11	**61.2±0.12**	**87.7±1.19**	73.4±2.02	76.8±0.53	79.7±0.33	72.68
ContextPred	65.7±0.62	74.2±0.06	62.5±0.31	**62.2±0.59**	77.2±0.88	75.3±1.57	77.1±0.86	76.0±2.08	71.28
InfoGraph	67.5±0.11	73.2±0.43	63.7±0.50	59.9±0.30	76.5±1.07	74.1±0.74	75.1±0.99	77.8±0.88	70.96
MolCLR	66.6±1.89	73.0±0.16	62.9±0.38	57.5±1.77	86.1±0.95	72.5±2.38	76.2±1.51	71.5±3.17	70.79
3D InfoMax	68.3±1.12	76.1±0.18	64.8±0.25	60.6±0.78	79.9±3.49	74.4±2.45	75.9±0.59	79.7±1.54	72.47
GraphMVP	69.4±0.21	76.2±0.38	64.5±0.20	60.5±0.25	86.5±1.70	76.2±2.28	76.2±0.81	**79.8±0.74**	73.66
MoleculeSDE (VE)	**73.2±0.48**	**76.5±0.33**	**65.2±0.31**	59.6±0.82	86.6±3.73	**79.9±0.19**	**78.5±0.28**	**80.4±0.92**	**74.98**
MoleculeSDE (VP)	**71.8±0.76**	**76.8±0.34**	**65.0±0.26**	60.8±0.39	**87.0±0.53**	**80.9±0.37**	**78.8±0.92**	79.5±2.17	**75.07**

Table 3. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error, and the best and the second best results are marked in **bold** and **bold**, respectively.

Pretraining	$\alpha \downarrow$	$\nabla\mathcal{E} \downarrow$	$\mathcal{E}_{\text{HOMO}} \downarrow$	$\mathcal{E}_{\text{LUMO}} \downarrow$	$\mu \downarrow$	$C_v \downarrow$	$G \downarrow$	$H \downarrow$	$R^2 \downarrow$	$U \downarrow$	$U_0 \downarrow$	ZPVE $\downarrow$
– (random init)	0.060	44.13	27.64	22.55	0.028	0.031	14.19	14.05	0.133	13.93	13.27	1.749
Type Prediction	0.073	45.38	28.76	24.83	0.036	0.032	16.66	16.28	0.275	15.56	14.66	2.094
Distance Prediction	0.065	45.87	27.61	23.34	0.031	0.033	14.83	15.81	0.248	15.07	15.01	1.837
Angle Prediction	0.066	48.45	29.02	24.40	0.034	0.031	14.13	13.77	0.214	13.50	13.47	1.861
3D InfoGraph	0.062	45.96	29.29	24.60	0.028	0.030	13.93	13.97	0.133	13.55	13.47	1.644
RR	0.060	43.71	27.71	22.84	0.028	0.031	14.54	13.70	**0.122**	13.81	13.75	1.694
InfoNCE	0.061	44.38	27.67	22.85	**0.027**	0.030	13.38	13.36	**0.116**	13.05	13.00	1.643
EBM-NCE	0.057	43.75	27.05	22.75	0.028	0.030	12.87	12.65	0.123	13.44	12.64	1.652
3D InfoMax	0.057	42.09	25.90	21.60	0.028	0.030	13.73	13.62	0.141	13.81	13.30	1.670
GraphMVP	0.056	41.99	25.75	**21.58**	**0.027**	**0.029**	13.43	13.31	0.136	13.03	13.07	1.609
GeoSSL-1L	0.058	42.64	26.32	21.87	0.028	0.030	12.61	12.81	0.173	12.45	12.12	1.696
GeoSSL	0.056	42.29	**25.61**	21.88	**0.027**	**0.029**	**11.54**	**11.14**	0.168	**11.06**	**10.96**	1.660
MoleculeSDE (VE)	**0.056**	**41.84**	25.79	21.63	**0.027**	**0.029**	**11.47**	**10.71**	0.233	**11.04**	**10.95**	**1.474**
MoleculeSDE (VP)	**0.054**	**41.77**	**25.74**	**21.41**	**0.026**	**0.028**	13.07	12.05	0.151	12.54	12.04	**1.587**

imize the MI between the conformation and augmented conformation using different objective functions, respectively. GeoSSL optimizes the same objective function using denoising score matching. Another work (Zaidi et al., 2022) is a special case of GeoSSL with one layer of denoising, and we name it GeoSSL-1L.

Baselines for 2D-3D multi-modality pretraining. There are two baselines on the 2D-3D multi-modal pretraining: vanilla GraphMVP (Liu et al., 2021a) utilizes both the contrastive and generative SSL, and 3D InfoMax (Stärk et al., 2022) only uses the contrastive learning part in GraphMVP.

Backbone models and MoleculeSDE. For all the baselines and MoleculeSDE, we use the same backbone models to better verify the effectiveness of the pretraining algorithms. We take the GIN model (Xu et al., 2018) and SchNet model (Schütt et al., 2018) for modeling 2D topology and 3D conformation, respectively. For MoleculeSDE training, we consider both the Variance Exploding (VE) and Variance Preserving (VP) (details in Appendix E).

5.2. Downstream with 2D Topology

We consider eight binary classification tasks from MoleculeNet (Wu et al., 2018). The results are in Table 2. We can observe that MoleculeSDE works best on 6 out of 8 tasks, and both the VE and VP version of MoleculeSDE pretraining can reach the best average performance.

5.3. Downstream with 3D Conformation

We consider 12 tasks from QM9 (Ramakrishnan et al., 2014) and 8 tasks from MD17 (Chmiela et al., 2017). QM9 is a dataset of 134K molecules consisting of 9 heavy atoms, and the 12 tasks are related to the quantum properties, such as energies at various settings. MD17 is a dataset on molecular dynamics simulation, and the 8 tasks correspond to 8 organic molecules. The goal is to predict the forces at different 3D positions. The results are in Tables 3 and 4, and MoleculeSDE can reach the best performance on 9 tasks in QM9 and 7 tasks in MD17.

5.4. Downstream with Topology to Conformation

We note that the pretraining in MoleculeSDE does two things: representation learning and topology/conformation generation. Such a conformation generation pretraining enables more prosperous downstream tasks. Here we consider 4 MoleculeNet tasks where only the 2D molecular graphs are available. Then we apply the SE(3)-equivariant conformation generation in MoleculeSDE, after which a 3D GNN will be trained for property prediction (Figure 2).

Table 4. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is mean absolute error, and the best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
– (random init)	1.203	0.380	0.386	0.794	0.587	0.826	0.568	0.773
Type Prediction	1.383	0.402	0.450	0.879	0.622	1.028	0.662	0.840
Distance Prediction	1.427	0.396	0.434	0.818	0.793	0.952	0.509	1.567
Angle Prediction	1.542	0.447	0.669	1.022	0.680	1.032	0.623	0.768
3D InfoGraph	1.610	0.415	0.560	0.900	0.788	1.278	0.768	1.110
RR	1.215	0.393	0.514	1.092	0.596	0.847	0.570	0.711
InfoNCE	1.132	0.395	0.466	0.888	0.542	0.831	0.554	0.664
EBM-NCE	1.251	0.373	0.457	0.829	0.512	0.990	0.560	0.742
3D InfoMax	1.142	0.388	0.469	0.731	0.785	0.798	0.516	0.640
GraphMVP	1.126	0.377	0.430	0.726	0.498	0.740	0.508	0.620
GeoSSL-1L	1.364	0.391	0.432	0.830	0.599	0.817	0.628	0.607
GeoSSL	**1.107**	0.360	0.357	0.737	0.568	0.902	**0.484**	0.502
MoleculeSDE (VE)	**1.112**	**0.304**	**0.282**	**0.520**	**0.455**	**0.725**	0.515	**0.447**
MoleculeSDE (VP)	1.244	**0.315**	**0.338**	**0.488**	**0.432**	**0.712**	**0.478**	**0.468**

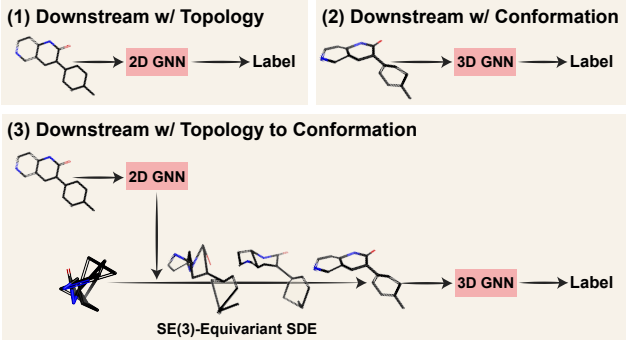


Figure 2. Illustration on three downstream tasks. The first two cover single-modal information only, and we fine-tune the pretrained 2D and 3D GNN from MoleculeSDE, respectively. The last downstream tasks contain topology only, then we use the pretrained 2D GNN and SE(3)-equivariant SDE model to generate conformations, followed by a 3D GNN for property prediction.

Table 5. Results for molecular property prediction with **SchNet** as backbone. The geometric structures (conformers) are generated using either MMFF or MoleculeSDE. We report the mean (and standard deviation) ROC-AUC of 3 seeds with scaffold splitting for each downstream task. CG denotes Conformation Generation.

Model	CG Method	BBBP ↑	Sider ↑	ClinTox ↑	Bace ↑
GIN	–	64.1±1.79	58.4±0.50	63.1±7.21	76.5±2.96
SchNet	MMFF	61.4±0.29	59.4±0.27	64.6±0.50	74.3±0.66
SchNet	ConfGF	62.7±1.97	60.1±0.87	64.1±2.83	73.2±3.53
SchNet	ClofNet	61.7±1.19	56.0±0.10	58.2±0.44	62.5±0.17
SchNet	MoleculeSDE	**65.2±0.43**	**60.5±0.39**	**72.9±1.02**	**78.6±0.40**

We consider three classic and state-of-the-art topology-to-conformation generation baselines. Merck molecular force field (MMFF) (Halgren, 1996) is a heuristic method using physical simulation. ConfGF (Shi et al., 2021) is an SE(3)-invariant conformation generation method using score matching, and ClofNet (Du et al., 2022b) is the state-of-the-art SE(3)-equivariant conformation generation method using GNN. With the generated conformations, we apply a SchNet model (without pretraining) for property prediction. We also add a 2D GNN baseline with the atom and bond type information. As shown in Table 5, we can observe that the CG baselines act worse than the 2D GNN for property prediction. Yet, MoleculeSDE can beat both the 2D and CG baselines. More discussions are in Appendix H.

5.5. Discussion on MoleculeSDE

For pretraining, data reconstruction is stronger than latent representation reconstruction. Starting from BOYL (Grill et al., 2020) and SimSiam (Chen & He, 2021), the non-contrastive SSL methods have been widely explored. GraphMVP (Liu et al., 2021a) summarizes that these methods essentially reconstruct the latent representation space. Our proposed MoleculeSDE further proves that directly applying the data reconstruction is superior on the graph data. This observation also aligns well in the vision domain (He et al., 2022). Complete results can be found in Appendix H.

6. Conclusion and Outlook

We proposed MoleculeSDE, a group symmetric pretraining method on the 2D topology and 3D geometry modalities of molecules. MoleculeSDE introduces the first SE(3)-equivariant and reflection-antisymmetric SDE for the topology to conformation generation and also the first SE(3)-invariant SDE for the conformation to topology generation for molecule representation learning. We provide theoretical insights that MoleculeSDE obtains tighter MI estimation over previous works. We also empirically verified that MoleculeSDE retains essential knowledge from both modalities, resulting in state-of-the-art performance on 26 out of 32 tasks compared to 17 competitive baselines.

We note that multi-modal pretraining has been widely explored in drug discovery, not only between the topologies and conformations (*i.e.*, the chemical structures), but also between the natural language and chemical structures. This research track is not exclusive to our work, and we believe that this can be a promising direction in the future exploration of the foundation model for molecule discovery.

Acknowledgement

This project is supported by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, the Canada CIFAR AI Chair Program, collaboration grants between Microsoft Research and Mila, Samsung Electronics Co., Ltd., Amazon Faculty Research Award, Tencent AI Lab Rhino-Bird Gift Fund and two NRC Collaborative R&D Projects (AI4D-CORE-06, AI4D-CORE-08). This project was also partially funded by IVADO Fundamental Research Project grant PRF-2019-3583139727.

References

Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):1–14, 2022.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., and Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

Clayden, J., Greeves, N., and Warren, S. *Organic chemistry*. Oxford university press, 2012.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33: 13260–13271, 2020.

Cotton, F. A. *Chemical applications of group theory*. John Wiley & Sons, 1991.

Demirel, M. F., Liu, S., Garg, S., Shi, Z., and Liang, Y. Attentive walk-aggregating graph neural networks. *Transactions on Machine Learning Research*, 2021.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Du, W., Yang, T., Zhang, H., and Du, Y. A flexible diffusion model. *arXiv preprint arXiv:2206.10365*, 2022a.

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, 2022b.

Du, W., Du, Y., Wang, L., Feng, D., Wang, G., Ji, S., Gomes, C., and Ma, Z.-M. A new perspective on building efficient and expressive 3d equivariant graph neural networks. *arXiv preprint arXiv:2304.04757*, 2023.

Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wang, F., Wu, H., and Wang, H. Chemrl-gem: Geometry enhanced molecular representation learning for property prediction. *arXiv preprint arXiv:2106.06130*, 2021.

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.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Halgren, T. A. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022.

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

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020a.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020b.

Huang, C.-W., Lim, J. H., and Courville, A. C. A variational perspective on diffusion-based generative models and score matching. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22863–22876. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/c11abfd29e4d9b4d4b566b01114d8486-Paper.pdf.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

Jiao, R., Han, J., Huang, W., Rong, Y., and Liu, Y. 3d equivariant molecular graph pretraining. *arXiv preprint arXiv:2207.08824*, 2022.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pp. 4839–4848. PMLR, 2020.

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514*, 2022.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

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

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*, 2021a.

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

Liu, S., Guo, H., and Tang, J. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=CjTHVo1dvR.

Liu, S., Zhu, Y., Lu, J., Xu, Z., Nie, W., Gitter, A., Xiao, C., Tang, J., Guo, H., and Anandkumar, A. A text-guided protein design framework. *arXiv preprint arXiv:2302.04611*, 2023b.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., and Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021b.

Liu, Y., Pan, S., Jin, M., Zhou, C., Xia, F., and Yu, P. S. Graph self-supervised learning: A survey. *arXiv preprint arXiv:2103.00111*, 2021c.

Nakata, M. and Shimazaki, T. Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*, 2022.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pp. 9558–9568. PMLR, 2021.

Shi, Y., Huang, Z., Wang, W., Zhong, H., Feng, S., and Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y. and Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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.

Sun, F.-Y., Hoffmann, J., Verma, V., and Tang, J. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations, ICLR*, 2020.

Sun, R., Dai, H., and Yu, A. W. Rethinking of graph pretraining on molecular representation. 2022.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, H., Kaddour, J., Liu, S., Tang, J., Kusner, M., Lasenby, J., and Liu, Q. Evaluating self-supervised learning for molecular graph embeddings. *arXiv preprint arXiv:2206.08005*, 2022a.

Wang, L., Zhou, Y., Wang, Y., Zheng, X., Huang, X., and Zhou, H. Regularized molecular conformation fields. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=7XCFxnG8nGS.

Wang, Y., Wang, J., Cao, Z., and Farimani, A. B. Molclr: Molecular contrastive learning of representations via graph neural networks. *arXiv preprint arXiv:2102.10056*, 2021.

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

Wu, L., Lin, H., Gao, Z., Tan, C., Li, S., et al. Self-supervised on graphs: Contrastive, generative, or predictive. *arXiv preprint arXiv:2105.07342*, 2021.

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

Xie, Y., Xu, Z., Zhang, J., Wang, Z., and Ji, S. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021.

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

Xu, Z., Luo, Y., Zhang, X., Xu, X., Xie, Y., Liu, M., Dickerson, K., Deng, C., Nakata, M., and Ji, S. Molecule3d: A benchmark for predicting 3d geometries from molecular graphs. *arXiv preprint arXiv:2110.01717*, 2021a.

Xu, Z., Luo, Y., Zhang, X., Xu, X., Xie, Y., Liu, M., Dickerson, K. A., Deng, C., Nakata, M., and Ji, S. Molecule3d: A benchmark for predicting 3d geometries from molecular graphs, 2021b. URL https://openreview.net/forum?id=m5rEiGxOGiL.

Yao, H., Yang, X., Pan, X., Liu, S., Koh, P. W., and Finn, C. Leveraging domain relations for domain generalization. *arXiv preprint arXiv:2302.02609*, 2023.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020a.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823, 2020b.

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.

Zhu, J., Xia, Y., Liu, C., Wu, L., Xie, S., Wang, T., Wang, Y., Zhou, W., Qin, T., Li, H., et al. Direct molecular conformation generation. *arXiv preprint arXiv:2202.01356*, 2022.

A. Comparison to Related Works

In Table 6, we provide a comprehensive overview of existing works on single-modal and multi-modal pretraining methods. We categorize the pretraining methods into generative and contrastive learning methods

Table 6. Comparison between MoleculeSDE and existing graph SSL methods.

Pre-training	2D Topology		3D Conformation		2D Topology and 3D Conformation	
	Generative	Contrastive	Generative	Contrastive	Generative	Contrastive
AttrMask (Hu et al., 2019; Liu et al., 2019)	✓	-	-	-	-	-
InfoGraph (Veličković et al., 2018; Sun et al., 2020)	-	✓	-	-	-	-
ContexPred (Hu et al., 2019)	-	✓	-	-	-	-
GraphCL (You et al., 2020a)	-	✓	-	-	-	-
Atom Type Prediction (Liu et al., 2023a)	-	-	✓	-	-	-
Distance Prediction (Fang et al., 2021; Liu et al., 2023a)	-	-	✓	-	-	-
Angle Prediction (Fang et al., 2021; Liu et al., 2023a)	-	-	✓	-	-	-
3D Infograph (Liu et al., 2023a)	-	-	-	✓	-	-
MI-RR Prediction (Liu et al., 2023a)	-	-	✓	-	-	-
MI-InfoNCE Prediction (Liu et al., 2023a)	-	-	-	✓	-	-
MI-EBM-NCE Prediction (Liu et al., 2023a)	-	-	-	✓	-	-
GeoSSL-1L (Zaidi et al., 2022)	-	-	✓	-	-	-
GeoSSL (Liu et al., 2023a)	-	-	✓	-	-	-
3D InfoMax (Stärk et al., 2022)	-	-	-	-	-	✓
GraphMVP (Liu et al., 2021a)	-	-	-	-	✓	✓
GraphMVP-C (Liu et al., 2021a)	-	✓	-	-	✓	✓
GraphMVP-G (Liu et al., 2021a)	✓	-	-	-	✓	✓
MoleculeSDE (ours)	-	-	-	-	✓	✓

B. Group Symmetry and Local Frame

B.1. SE(3)/E(3) Group action and representations

In this article, a 3D molecular graph is represented by a 3D point cloud. The corresponding symmetry group is $SE(3)$, which consists of translations and rotations. Recall that we define the notion of equivariance functions in $\mathbf{R}^3$ in the main text through group actions. Formally, the group SE(3) is said to act on $\mathbf{R}^3$ if there is a mapping $\phi : SE(3) \times \mathbf{R}^3 \to \mathbf{R}^3$ satisfying the following two conditions:

1. if $e \in SE(3)$ is the identity element, then
$$\phi(e, \boldsymbol{r}) = \boldsymbol{r} \quad \text{for } \forall \boldsymbol{r} \in \mathbf{R}^3.$$
2. if $g_1, g_2 \in SE(3)$, then
$$\phi(g_1, \phi(g_2, \boldsymbol{r})) = \phi(g_1 g_2, \boldsymbol{r}) \quad \text{for } \forall \boldsymbol{r} \in \mathbf{R}^3.$$

Then, there is a natural $SE(3)$ action on vectors $\boldsymbol{r}$ in $\mathbf{R}^3$ by translating $\boldsymbol{r}$ and rotating $\boldsymbol{r}$ for multiple times. For $g \in SE(3)$ and $\boldsymbol{r} \in \mathbf{R}^3$, we denote this action by $g\boldsymbol{r}$. Once the notion of group action is defined, we say a function $f : \mathbf{R}^3 \to \mathbf{R}^3$ that transforms $\boldsymbol{r} \in \mathbf{R}^3$ is equivariant if:
$$f(g\boldsymbol{r}) = gf(\boldsymbol{r}), \quad \text{for } \forall \ \boldsymbol{r} \in \mathbf{R}^3.$$
On the other hand, $f : \mathbf{R}^3 \to \mathbf{R}^1$ is invariant, if f is independent of the group actions:
$$f(g\boldsymbol{r}) = f(\boldsymbol{r}), \quad \text{for } \forall \ \boldsymbol{r} \in \mathbf{R}^3.$$
For some scenarios, our problem is chiral sensitive. That is, after mirror reflecting a 3D molecule, the properties of the molecule may change dramatically. In these cases, it's crucial to include reflection transformations into consideration. More precisely, we say an $SE(3)$-equivariant function f is **reflection-antisymmetric**, if:
$$f(\rho\boldsymbol{r}) \neq f(\boldsymbol{r}), \tag{17}$$
for some reflections $\rho \in E(3)$.

B.2. Equivariant Frames

Frame is a popular terminology in science areas. In physics, frame is equivalent to a coordinate system. For example, we may assign a frame to all observers, although different observers may collect different data under different frames, the underlying physics law should be the same. In other words, denote the physics law by f, then f should be an equivariant function.

Since there are three orthogonal directions in $\mathbf{R}^3$, a frame in $\mathbf{R}^3$ consists of three orthogonal vectors:

$$F = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3).$$

Once equipped with a frame (coordinate system), we can project all geometric quantities to this frame. For example, an abstract vector $\boldsymbol{r} \in \mathbf{R}^3$ can be written as $\boldsymbol{r} = (r_1, r_2, r_3)$ under frame F, if: $\boldsymbol{r} = r_1\boldsymbol{e}_1 + r_2\boldsymbol{e}_2 + r_3\boldsymbol{e}_3$. An equivariant frame further requires the three orthonormal vectors in $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ to be equivariant. Intuitively, an equivariant frame will transform according to the global rotation or translation of the whole system. Once equipped with an equivariant frame, we can project equivariant vectors into this frame:

$$\boldsymbol{r} = \tilde{r}_1\boldsymbol{e}_1 + \tilde{r}_2\boldsymbol{e}_2 + \tilde{r}_3\boldsymbol{e}_3. \tag{18}$$

We call the process of $\boldsymbol{r} \to \tilde{r} := (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3)$ the **projection** operation. Since $\tilde{r}_i = \boldsymbol{e}_i \cdot \boldsymbol{r}_i$ is expressed as an inner product between equivariant vectors, we know that $\tilde{r}$ consists of scalars.

In this article, we assign an equivariant frame to each edge. Therefore, we call them the local frames that reflect the local geometry around each chemical bond, following (Du et al., 2022b; 2023). Consider node i and one of its neighbors j with positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. The orthonormal equivariant frame $\mathcal{F}_{ij} := (\boldsymbol{e}_1^{ij}, \boldsymbol{e}_2^{ij}, \boldsymbol{e}_3^{ij})$ is defined with respect to $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as follows:

$$(\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \frac{\boldsymbol{x}_i \times \boldsymbol{x}_j}{\|\boldsymbol{x}_i \times \boldsymbol{x}_j\|}, \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} \times \frac{\boldsymbol{x}_i \times \boldsymbol{x}_j}{\|\boldsymbol{x}_i \times \boldsymbol{x}_j\|}). \tag{19}$$

Note that this frame is translation invariant if the system's mass is zero.

Reflection-AntiSymmetric Since we implement the cross product $\times$ for building the local frames, the third vector in the frame is a pseudo-vector. Then, the **projection** operation is not invariant under reflections (the inner product between a vector and a pseudo-vector change signs under reflection). Therefore, our model is able to discriminate two 3D geometries with different chirality.

Our local frames also enable us to output equivariant vectors by multiplying scalars (v_1, v_2, v_3) with the frame: $\boldsymbol{v} = v_1 \cdot \boldsymbol{e}_1 + v_2 \cdot \boldsymbol{e}_2 + v_3 \cdot \boldsymbol{e}_3$. It's easy to check that $\boldsymbol{v}$ is a $SE(3)$ equivariant (reflection-antisymmetric) vector.

C. Denoising Score Matching

C.1. Energy-Based Model (EBM)

Energy-based model (EBM) is a powerful tool for modeling the data distribution. The formulation is:

$$p_\theta(\boldsymbol{x}) = \frac{\exp(-E(\boldsymbol{x}))}{A} = \frac{\exp(f(\boldsymbol{x}))}{A}, \tag{20}$$

where the bottleneck is the intractable partition function $A = \int_{\boldsymbol{x}} \exp(-E(\boldsymbol{x}))d\boldsymbol{x}$. Recently, there have been big progress (Song & Kingma, 2021) in solving such an intractable function, including contrastive divergence (Hinton, 2002), score matching (Hyvärinen & Dayan, 2005; Song et al., 2020), and noise contrastive estimation (Gutmann & Hyvärinen, 2010).

C.2. Score Matching

There exists a family of solutions called **score matching** (SM) to solve Equation (20). The core idea of SM is that for the generative task, we do not need to directly estimate the density, but we just need to know the score or gradient of the data distribution, *i.e.*, $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$. Then a Markov chain Monte Carlo (MCMC) strategy can be adopted for data generation.

Score The score is defined as the gradient of log-likelihood w.r.t. the data $\boldsymbol{x}$:

$$s_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} E(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log A = -\nabla_{\boldsymbol{x}} E(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}). \tag{21}$$

Thus, by taking the score, *i.e.*, the gradient w.r.t. the data, the partition function term will disappear since it is a constant. The SM transforms the density estimation problem into a score (gradient) matching problem: if the first-order gradient of function ($s_\theta(x)$) can match, then the learned model distribution with EBM is able to capture the data distribution precisely.

Explicit Score Matching (ESM) For training the model, originally, SM (Hyvärinen & Dayan, 2005) applies the Fisher divergence to measure the discrepancy between data distribution and model distribution, terms explicit score matching (ESM):

$$\begin{aligned} D_F(p_{\text{data}}(\boldsymbol{x})||p_\theta(\boldsymbol{x})) &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}}\|\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) - \nabla_x \log p_{\text{data}}(\boldsymbol{x})\|_2^2 \\ &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}}\|s_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|_2^2. \end{aligned} \tag{22}$$

The expectation w.r.t. $p_{\text{data}}(\boldsymbol{x})$ can be approximated using Monte Carlo sampling, yet the second term of Equation (22) is intractable to compute since it needs to know $\nabla_x \log p_{\text{data}}(\boldsymbol{x})$. There are multiple solutions to this, including Implicit Score matching (ISM) (Hyvärinen & Dayan, 2005) which rewrites Equation (22) using integration by parts; the other appealing solution is the denoising score matching (DSM). Both are introduced below.

Implicit Score Matching (ISM) It is still impractical to calculate Equation (22) due to the $\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})$ term. Under certain conditions (Hyvärinen & Dayan, 2005), the Fisher divergence can be rewritten using integration by parts, and we can turn it to the implicit score matching (ISM), *i.e.*, ESM to ISM:

$$\begin{aligned} \mathbb{E}_{p_{\text{data}}}[\|\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x}) - s_\theta(\boldsymbol{x})\|^2] &= \mathbb{E}_{p_{\text{data}}}\left[\frac{1}{2}(\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}))^2 + \text{tr}(\nabla_{\boldsymbol{x}}^2 E_\theta(\boldsymbol{x}))\right] + C \\ &= \mathbb{E}_{p_{\text{data}}}\left[\frac{1}{2}(\|s_\theta(\boldsymbol{x})\|^2 + \text{tr}(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}))\right] + C, \end{aligned} \tag{23}$$

where C is the constant. The drawback of Equation (23) is that it requires computing the Trace of Hessian. It is computationally expensive as the computation of the full second derivatives is quadratic in the dimensionality of data.

Denoising Score Matching (DSM) Along this line, denoising score matching (DSM) (Vincent, 2011) proposes an elegant solution by connecting the SM with denoising autoencoder. It first perturbs the data with a noise distribution, *i.e.*, $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})$, and the goal is to use SM to approximate the $q_\sigma(\tilde{\boldsymbol{x}}) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})]$ with a model distribution $p_\theta(\tilde{\boldsymbol{x}})$. DSM (Vincent, 2011) then calculates the Fisher divergence between the perturbed data distribution and perturbed model distribution, which leads to the following equation:

$$\begin{aligned} D_F(q_\sigma(\tilde{\boldsymbol{x}})||p_\theta(\tilde{\boldsymbol{x}})) &= \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\boldsymbol{x}})}\left[\|\nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}) - \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{q_\sigma(\boldsymbol{x},\tilde{\boldsymbol{x}})}\left[\|\nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) - s_\theta(\tilde{\boldsymbol{x}})\|^2\right] + C. \end{aligned} \tag{24}$$

The detailed derivation of Equation (24) can be found in Appendix C.3. This is an elegant solution because under the Gaussian kernel, *i.e.*, $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 I)$, we can have an analytical solution to $\nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \frac{1}{\sigma^2}(\boldsymbol{x} - \tilde{\boldsymbol{x}})$. This is essentially a direction moving from $\tilde{\boldsymbol{x}}$ back to $\boldsymbol{x}$, and DSM makes the score to match it. Finally, the objective becomes:

$$\begin{aligned} \mathcal{L}_{\text{DSM}} &\approx \frac{1}{2N}\sum_{i=1}^{N}\left[\|\frac{\tilde{\boldsymbol{x}}_i - x_i}{\sigma^2} + s_\theta(\tilde{\boldsymbol{x}}_i)\|^2\right], \\ &= \frac{1}{2}\mathbb{E}_{q_\sigma(\boldsymbol{x},\tilde{\boldsymbol{x}})}\left[\|\frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{\sigma^2} - s_\theta(\tilde{\boldsymbol{x}})\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\left[\|\frac{\tilde{\boldsymbol{x}} - \boldsymbol{x}}{\sigma^2} + s_\theta(\tilde{\boldsymbol{x}})\|^2\right]. \end{aligned} \tag{25}$$

Additionally, (Vincent, 2011) also proves that DSM is equivalent to ESM. Though there exists certain drawbacks (Song & Kingma, 2021), DSM serves as a promising tool to enable the SM family as a more applicable solution to EBM.

Noise Conditional Score Network (NCSN) Recently, (Song & Ermon, 2019) finds that perturbing data with random Gaussian noise makes the data distribution more powerful than SM model. Thus it proposes Noise Conditional Score Network (NCSN) that can perturb the data using various levels of noise and estimates scores at all levels simultaneously. More concretely, NCSN chooses the Gaussian kernel as noise distribution, *i.e.*, $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 I)$. With L levels of noises, it extends Equation (25) as the following new objective:

$$\begin{aligned} \ell(\theta;\sigma_i) &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\mathbb{E}_{q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\Big[\|\frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma_i^2} + s_\theta(\tilde{\boldsymbol{x}})\|^2\Big] \\ \mathcal{L}_{\text{NCSN}} &= \frac{1}{L}\sum_{l=1}^{L}\lambda(\sigma_i)\ell(\theta;\sigma_i), \end{aligned} \tag{26}$$

where $\lambda(\sigma_i) > 0$ is a coefficient function on σ_i.

Sampling for SM For the SM family (including ESM, ISM, DSM and NCSN), once we have the score, we can sample the data by a MCMC sampling method called Langevin dynamics:

$$\tilde{\boldsymbol{x}}_{t+1} = \tilde{\boldsymbol{x}}_t + \frac{\epsilon^2}{2}\nabla_x \log p_\theta(\boldsymbol{x}_T) + \epsilon z_t = \tilde{\boldsymbol{x}}_t + \frac{\epsilon^2}{2}s(\boldsymbol{x}_t) + \epsilon z_t. \tag{27}$$

Discussion Till now, the SM family provides a unique solution for the generative task. It may seem likelihood-free, but recently, another track on diffusion model found that indeed these two research lines can contribute to the same formulation (Song et al., 2020). The only difference is that the diffusion model starts with a variational approximation perspective.

C.3. Proof of DSM

Proof of Equation (24).
First to put this into the ESM, we can have:

$$\begin{aligned} D_F(q(\tilde{\boldsymbol{x}})||p_\theta(\tilde{\boldsymbol{x}})) &= \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\|\nabla_{\tilde{\boldsymbol{x}}}\log p_\theta(\tilde{\boldsymbol{x}}) - \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\|_2^2 \\ &= \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\Big[\|\nabla_{\tilde{\boldsymbol{x}}}\log p_\theta(\tilde{\boldsymbol{x}})\|^2 + 2\cdot\langle\nabla_{\tilde{\boldsymbol{x}}}\log p_\theta(\tilde{\boldsymbol{x}}), \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\rangle + \|\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\|^2\Big] \\ &= \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\Big[\|s_\theta(\tilde{\boldsymbol{x}})\|^2 + 2\cdot\langle s_\theta(\tilde{\boldsymbol{x}}), \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\rangle\Big] + C_1, \end{aligned} \tag{28}$$

where $C_1 = \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\|\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\|^2$ is a constant and does not depend on the model parameter θ.

Then let's take out the second term, and we can have following:

$$\begin{aligned} &\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\Big[\langle\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}), s_\theta(\tilde{\boldsymbol{x}})\rangle\Big] \\ =&\int_{\tilde{\boldsymbol{x}}} q(\tilde{\boldsymbol{x}})\langle\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}), s_\theta(\tilde{\boldsymbol{x}})\rangle d\tilde{\boldsymbol{x}} \\ =&\int_{\tilde{\boldsymbol{x}}}\langle\nabla_{\tilde{\boldsymbol{x}}} q(\tilde{\boldsymbol{x}}), s_\theta(\tilde{\boldsymbol{x}})\rangle d\tilde{\boldsymbol{x}} \\ =&\int_{\tilde{\boldsymbol{x}}}\langle\nabla_{\tilde{\boldsymbol{x}}}\int_x q(\boldsymbol{x})\cdot q(\tilde{\boldsymbol{x}}|x)dx, s_\theta(\tilde{\boldsymbol{x}})\rangle d\tilde{\boldsymbol{x}} \\ =&\int_{\tilde{\boldsymbol{x}}}\int_x q(\boldsymbol{x})\cdot\langle\nabla_{\tilde{\boldsymbol{x}}} q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle dx d\tilde{\boldsymbol{x}} \\ =&\int_{\tilde{\boldsymbol{x}}}\int_x q(\boldsymbol{x})\cdot\langle q(\tilde{\boldsymbol{x}}|x)\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle dx d\tilde{\boldsymbol{x}} \\ =&\int_{\tilde{\boldsymbol{x}}}\int_x q(\boldsymbol{x})q(\tilde{\boldsymbol{x}}|x)\cdot\langle\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle dx d\tilde{\boldsymbol{x}} \\ =&\mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\Big[\langle\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle\Big] \end{aligned} \tag{29}$$

So we can put this back to Equation (28) and let ESM be as:

$$\begin{aligned} D_F(q(\tilde{\boldsymbol{x}})||p_\theta(\tilde{\boldsymbol{x}})) &= \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\left[\|s_\theta(\tilde{\boldsymbol{x}})\|^2 + 2\cdot\langle s_\theta(\tilde{\boldsymbol{x}}), \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}})\rangle\right] + C_1 \\ &= \frac{1}{2}\mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\left[\|s_\theta(\tilde{\boldsymbol{x}})\|^2\right] + \mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\left[\langle \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle\right] + C_1. \end{aligned} \tag{30}$$

And then we can get the following equivalent objective by some special reconstruction:

$$\begin{aligned} D_F(q(\tilde{\boldsymbol{x}})||p_\theta(\tilde{\boldsymbol{x}})) &= \frac{1}{2}\mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\left[\|s_\theta(\tilde{\boldsymbol{x}})\|^2\right] + \mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\left[\langle \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x), s_\theta(\tilde{\boldsymbol{x}})\rangle\right] + C_2 + \Delta \\ &= \frac{1}{2}\mathbb{E}_{q(x,\tilde{\boldsymbol{x}})}\left[\|s_\theta(\tilde{\boldsymbol{x}}) - \nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x)\|^2\right] + \Delta. \end{aligned} \tag{31}$$

where $C_2 = \frac{1}{2}\mathbb{E}_{q(\tilde{\boldsymbol{x}})}\|\nabla_{\tilde{\boldsymbol{x}}}\log q(\tilde{\boldsymbol{x}}|x)\|^2$ is a re-constructured constant.
End of proof.

D. Diffusion Model

Another generative modeling track is the denoising diffusion probabilistic model (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020). The diffusion model is composed of two processes: a forward process that adds noise to the data and a backward process that does denoising to generate the true data. Below we give a brief summary of the Gaussian diffusion model introduced in (Ho et al., 2020).

D.1. Pipeline of Denoising Diffusion Probabilistic Model

Forward process Give a data point from the real distribution $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$, the forward diffusion process is that we add small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. The step sizes are controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^T$:

$$q(\boldsymbol{x}_T|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_T; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t I), \tag{32}$$

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_T|\boldsymbol{x}_{t-1}). \tag{33}$$

A nice property of the forward process is that we can sample $\boldsymbol{x}_t$ at any arbitrary timestep t in a closed form using the reparameterization trick. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, we have:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I). \tag{34}$$

Then using the Bayes theorem, $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T, \boldsymbol{x}_0)$ can be written as a Guassian:

$$\begin{aligned} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T, \boldsymbol{x}_0) &= \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}(\boldsymbol{x}_T, \boldsymbol{x}_0), \tilde{\beta}_t I) \\ &= \mathcal{N}(\boldsymbol{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\big(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_t\big), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t) \end{aligned} \tag{35}$$

Reverse process Under a reasonable setting for β_t and T (Dhariwal & Nichol, 2021), the distribution $q(\boldsymbol{x}_T)$ is nearly an isotropic Gaussian, and sampling $\boldsymbol{x}_T$ is trivial. Then for the reverse process, we need $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)$. (Sohl-Dickstein et al., 2015) claims that as $T \to \infty$ and $\beta_t \to 0$, $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ approaches a diagonal Gaussian distribution. To this end, it is sufficient to train a neural network to predict a mean μ_θ and a diagonal covariance matrix Σ_θ:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_T, t), \Sigma_\theta(\boldsymbol{x}_T, t)). \tag{36}$$

Parameterization and variational lower bound The hidden variables are $x_{1:T}$, and inference is to infer the latent variables, *i.e.*, $p(x_{1:T}|\boldsymbol{x}_0)$. Variational inference is to use $p(x_{1:T}|\boldsymbol{x}_0)$ to estimate the true posterior $q(x_{1:T}|\boldsymbol{x}_0)$. If we use

$x_{1:T}$ as z, and $\boldsymbol{x}_0$ as x, then it is resemble to the VAE. Recall that

$$\begin{aligned} KL(q(z|x)||p(z|x)) &= \mathbb{E}_{q(z|x)}\big[\log \frac{q(z|x)}{p(z|x)}\big] \\ &= \mathbb{E}_{q(z|x)}\big[\log \frac{q(z|x)p(x)}{p(x,z)}\big] \\ &= \log p(x) + \mathbb{E}_{q(z|x)}\big[\log \frac{q(z|x)}{p(x,z)}\big]. \end{aligned} \tag{37}$$

To adapt this to the diffusion model setting, we can have:

$$KL(q(x_{1:T}|\boldsymbol{x}_0)||p(x_{1:T}|\boldsymbol{x}_0)) = \log p(\boldsymbol{x}_0) + \mathbb{E}_{q(x_{1:T}|\boldsymbol{x}_0)}\big[\log \frac{q(x_{1:T}|\boldsymbol{x}_0)}{p(x_{0:T})}\big], \tag{38}$$

and our goal becomes to maximize the variational lower bound (VLB):

$$\begin{aligned} \mathcal{L}_{VLB} &= \mathbb{E}_q\Big[\log \frac{q(x_{1:T}|\boldsymbol{x}_0)}{p_\theta(x_{0:T})}\Big] \\ &= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\boldsymbol{x}_T|\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_T)\prod_{t=1}^T p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)}\Big] \\ &= \mathbb{E}_q\Big[-\log p_\theta(\boldsymbol{x}_T) + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_T|\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)} + \log \frac{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}\Big] \\ &= \mathbb{E}_q\Big[-\log p_\theta(\boldsymbol{x}_T) + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_T|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)} + \log \frac{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}\Big] \\ &= \mathbb{E}_q\Big[-\log p_\theta(\boldsymbol{x}_T) + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)\cdot q(\boldsymbol{x}_T|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)\cdot q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} + \log \frac{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}\Big] \quad \text{Baye's rule} \\ &= \mathbb{E}_q\Big[-\log p_\theta(\boldsymbol{x}_T) + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)} + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} + \log \frac{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}\Big] \\ &= \mathbb{E}_q\Big[-\log p_\theta(\boldsymbol{x}_T) + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} + \log \frac{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}\Big] \\ &= \mathbb{E}_q\Big[\log \frac{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_T)} + \sum_{t=2}^T \log \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} - \log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\Big] \\ &= KL[q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_T)] + \sum_{t=2}^T KL[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)] - \mathbb{E}_q[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] \\ &= \underbrace{KL[q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_T)]}_{\mathcal{L}_T} + \sum_{t=2}^T \underbrace{KL[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)]}_{\mathcal{L}_{t-1}} - \underbrace{\mathbb{E}_q[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\mathcal{L}_0} \end{aligned} \tag{39}$$

Thus, we want to model $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)$ with parameterization $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)$. According to Equation (35), we can have:

$$\begin{aligned} \mathcal{L}_t &= \mathbb{E}_q\Big[\log \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T,\boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)}\Big] \\ &= \mathbb{E}_{\boldsymbol{x}_0,z}\Big[-\frac{1}{2\|\Sigma_\theta\|^2}\Big\|\tilde{\mu}_t(\boldsymbol{x}_T,\boldsymbol{x}_0) - \mu_\theta(\boldsymbol{x}_T,t)\Big\|^2\Big] \\ &= \mathbb{E}_{\boldsymbol{x}_0,z}\Big[-\frac{1}{2\|\Sigma_\theta\|^2}\cdot\frac{1}{\alpha_t}\cdot\Big\|\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_t - \boldsymbol{x}_T + \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_\theta(\boldsymbol{x}_T,t)\Big\|^2\Big] \\ &= \mathbb{E}_{\boldsymbol{x}_0,z}\Big[-\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|^2}\cdot\|z_t - z_\theta(\boldsymbol{x}_T,t)\|^2\Big] \\ &= \mathbb{E}_{\boldsymbol{x}_0,z}\Big[-\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|^2}\cdot\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}z_t,t)\|^2\Big], \end{aligned} \tag{40}$$

Simplification There is a nice strategy proposed in (Ho et al., 2020): the objective function in Equation (40) can be simplified by ignoring the weighting term:

$$\mathcal{L}_t^{simple} = \mathbb{E}_{\boldsymbol{x}_0, z}\left[\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2\right] \tag{41}$$

D.2. Important Tricks

The DDPM (Ho et al., 2020) also adopts the following tricks in the training and inference.

- $\Sigma_\theta(\boldsymbol{x}_T, t) = \sigma_t^2 I$. Then DDPM empirically tests $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\hat{\alpha}_{t-1}}{1-\hat{\alpha}_t}\beta_t$. Both have similar results. And these two are the two extreme choices corresponding to the upper and lower bounds on reverse process entropy with coordinatewise unit variance (Sohl-Dickstein et al., 2015).
- The second trick is that we model $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_T, t); \sigma_t^2 I)$. Then the loss term becomes:

$$L_{t-1} = \mathbb{E}_q[\frac{1}{2\sigma_t^2}\|\tilde{\mu}(\boldsymbol{x}_T, \boldsymbol{x}_0) - \mu_\theta(\boldsymbol{x}_T, t)\|^2] \tag{42}$$

  So one straightforward way is to directly model the mean, *i.e.*, to match μ_θ and $\tilde{\mu}$.
- Meanwhile, during the diffusion process, we can have $\boldsymbol{x}_0$ and $\boldsymbol{x}_T$. Thus, we can have $\tilde{\mu}(\boldsymbol{x}_T, \boldsymbol{x}_0)$ as a function of of $(\boldsymbol{x}_T, \epsilon)$ or $(\boldsymbol{x}_0, \epsilon)$. In specific, we can write it as:

$$L_{t-1} = \mathbb{E}_q[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\alpha_t}}\big(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\big) - \mu_\theta(\boldsymbol{x}_T, t)\|^2] \tag{43}$$

  Since $\boldsymbol{x}_T$ can be obtained by $\boldsymbol{x}_0$, then we may as well model $\mu_\theta(\boldsymbol{x}_T, t) = \tilde{\mu}_t(\boldsymbol{x}_T, \boldsymbol{x}_0(\boldsymbol{x}_T, \epsilon_t)) = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_T, t))$, and the objective function becomes:

$$\begin{aligned} L_{t-1} &= \mathbb{E}_q[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\alpha_t}}\big(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\big) - \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_T, t))\|^2] \\ &= \mathbb{E}_q[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(\boldsymbol{x}_T, t)\|^2] \end{aligned} \tag{44}$$

- Thus, during sampling, the mean is

$$\mu_\theta(\boldsymbol{x}_T, t) = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_T, t)), \tag{45}$$

  thus the sampling is obtained by:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_T, t)) + \sqrt{\beta_t}\epsilon \qquad \text{//DDPM's paper} \tag{46}$$

- Further, if we want to model the score, *i.e.*, the $\log_{\tilde{x}} \log q(\tilde{x}|x)$, and then the score network defined here needs to have a shift:

$$\begin{aligned} \log_{\tilde{x}} q(\tilde{x}|x) &= \log_{\boldsymbol{x}_T} q(\boldsymbol{x}_T|\boldsymbol{x}_0) \\ &= -\log_{\boldsymbol{x}_T} \mathcal{N}(\boldsymbol{x}_T; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I) \\ &= -\frac{\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} \\ &= \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T + \beta_t\epsilon_\theta(\boldsymbol{x}_T, t)) + \sqrt{\beta_t}\epsilon. \end{aligned} \tag{47}$$

E. Stochastic Differential Equation

A more recent work (Song et al., 2020) unifies the score matching and DDPM into a unified framework, the stochastic differential equation (SDE). First let's do a quick recap on the NCSN and DDPM.

E.1. Review of NCSN and DDPM

NCSN The objective function is:

$$\mathcal{L} = \sum_{t=1}^{T} \sigma_t^2 \cdot \mathbb{E}_{p_{data}(x)} \mathbb{E}_{q_{\sigma_t}(\boldsymbol{x}_T|x)} \Big[\Big\| s_\theta(\boldsymbol{x}_T, \sigma_t) - \nabla_{\boldsymbol{x}_T} \log q_{\sigma_t}(\tilde{x}|x) \Big\|^2 \Big]. \tag{48}$$

There is no notion of forward and backward, and the **sampling** is achieved by the Langevine dynamics:

$$\boldsymbol{x}_T^m = \boldsymbol{x}_T^{m-1} + \delta s_\theta(\boldsymbol{x}_T^{m-1}, \sigma_t) + \sqrt{2\delta_i}\epsilon, \qquad m = 1, \ldots, M, \tag{49}$$

where δ is the step size and $\epsilon \sim \mathcal{N}(0, I)$. The above is repeated with:

- From $t = T$ to $t = 1$.
- $\boldsymbol{x}_T^0 \sim \mathcal{N}(0, \sigma_T^2 I)$ for $t = T$.
- $\boldsymbol{x}_T^0 = x_{t+1}^M$ when $t < T$.

DDPM The **forward** (non-modeling) is:

$$p(\boldsymbol{x}_T|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_T; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)I). \tag{50}$$

The **backward** (modeling part) is:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T + \beta_t s_\theta(\boldsymbol{x}_T, t)), \beta_t I\Big) \tag{51}$$

The **objective** function is:

$$\mathcal{L} = \sum_{t=1}^{T} \mathbb{E}_{p_{data}(\boldsymbol{x}_0)} \mathbb{E}_{p_{\alpha_t}(\boldsymbol{x}_T|\boldsymbol{x}_0)} \Big[\Big\| s_\theta(\boldsymbol{x}_T, t) - \nabla_{\boldsymbol{x}_T} \log q(\boldsymbol{x}_T|\boldsymbol{x}_0) \Big\|^2 \Big] \tag{52}$$

The **sampling** is as follows:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_T + \beta_t s_\theta(\boldsymbol{x}_T, t)) + \sqrt{\beta_t}\epsilon_t, \qquad t = T, \ldots, 1, \tag{53}$$

where $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$ and $\epsilon_t \sim \mathcal{N}(0, I)$. This is called the ancestral sampling since it amounts to performing ancestral sampling from the graphical mode (Song et al., 2020).

Comparison of NCSN and DDPM Note that in DDPM, it is modeling $KL(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T))$, while NCSN is modeling $s_\theta(\boldsymbol{x}_T, t) - \nabla_{\boldsymbol{x}_T} \log p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_T)$ directly. Essentially, these two are equivalent, because:

$$-\sqrt{1-\bar{\alpha}_t} s_\theta(\boldsymbol{x}_T, t) = \epsilon_\theta(\boldsymbol{x}_T, t). \tag{54}$$

E.2. Stochastic Differential Equation

Then we introduce the how NCSN and DDPM are solutions to Stochastic Differential Equation (SDE). The SDE is also formulated with the forward and backward processes.

Forward process is

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)dw, \tag{55}$$

where $f(\boldsymbol{x}, t)$ is the vector-value drift coefficient, $g(t)$ is the diffusion coefficient, and dw is the Wiener process.

Backward process is:

$$d\boldsymbol{x} = [f(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}_T)]dt + g(t)dw \tag{56}$$

Then the questions is how to estimate the score $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$.

E.3. Stochastic Differential Equation and Score Matching

According to (Song et al., 2020), the objective of solutions to SDE can be written in the form of score matching:

$$\mathcal{L} = \mathbb{E}_t \mathbb{E}_{\boldsymbol{x}_0} \mathbb{E}_{\boldsymbol{x}_T|\boldsymbol{x}_0} \Big[\lambda(t) \Big\| s_\theta(\boldsymbol{x}_T, t) - \nabla_{\boldsymbol{x}_T} \log p(\boldsymbol{x}_T|\boldsymbol{x}_0) \Big\|^2 \Big], \tag{57}$$

where $\lambda(t)$ is a weighting function. With sufficient data and model capacity, the optimal $s_\theta(\boldsymbol{x}_T, t)$ equals to $\nabla_x \log p(\boldsymbol{x}_T)$ for almost all $\boldsymbol{x}_T$ and t. Then we will review how to match the NCSN and DDPM into this framework.

NCSN and VE SDE The discretization of VE SDE yields NCSN. The **forward** process is:

$$\begin{aligned} \boldsymbol{x}_T &= \boldsymbol{x}_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\epsilon_{t-1}, \qquad \text{// discrete Markov chain} \\ dx &= \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw, \qquad \text{// continuous SDE} \end{aligned} \tag{58}$$

Assumption: the $\{\sigma_t\}, t = 1, 2, \ldots, T$ is a geometric sequence. Then the transition kernel becomes:

$$dx = \sigma_{min} \Big(\frac{\sigma_{max}}{\sigma_{min}} \Big)^2 \sqrt{2 \log \frac{\sigma_{max}}{\sigma_{min}}} dw \tag{59}$$

and the perturbation kernel can be obtained by

$$p(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_T; \boldsymbol{x}_0, \sigma_{min}^2 \Big(\frac{\sigma_{max}}{\sigma_{min}} \Big)^{2t} I \Big) \tag{60}$$

This always give a process with exploding variance, so this is called VE SDE.

DDPM and VP SDE The forward process is:

$$\begin{aligned} \boldsymbol{x}_T &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}, \qquad \text{// discrete Markov chain} \\ dx &= -\frac{1}{2}(1-\alpha_t)x dt + \sqrt{\beta(t)} dw, \qquad \text{// continuous SDE} \end{aligned} \tag{61}$$

If we use the arithmetic sequence for $\{\beta\}_{t=1}^T$, the transition kernel for VP SDE is:

$$d\boldsymbol{x} = -\frac{1}{2}(\beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})\boldsymbol{x} dt + \sqrt{\beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})} dw \tag{62}$$

and the perturbation kernel is:

$$p(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_T; e^{-\frac{1}{4}t^2(\beta_{\text{max}} - \beta_{\text{min}} - \frac{1}{2}t\beta_{\text{min}}} \boldsymbol{x}_0, I - Ie^{-\frac{1}{2}t^2(\beta_{\text{max}} - \beta_{\text{min}} - t\beta_{\text{min}}} \Big) \tag{63}$$

The drift coefficient is $-\frac{1}{2}(\beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})x$, and the diffusion coefficient is $\beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})$.

F. Mutual Information and Equivalent Conditional Likelihoods

To maximize the mutual information between variable X, Y is equivalent to optimize the following equation:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})} \big[\log p(\boldsymbol{y}|\boldsymbol{x}) + \log p(\boldsymbol{x}|\boldsymbol{y}) \big]. \tag{64}$$

Proof. First we can get a lower bound of MI. Assuming that there exist (possibly negative) constants a and b such that $a \leq H(X)$ and $b \leq H(Y)$, *i.e.*, the lower bounds to the (differential) entropies, then we have:

$$\begin{aligned} I(X;Y) &= \frac{1}{2}\big(H(X) + H(Y) - H(Y|X) - H(X|Y)\big) \\ &\geq \frac{1}{2}\big(a + b - H(Y|X) - H(X|Y)\big) \\ &\geq \frac{1}{2}\big(a + b\big) + \mathcal{L}, \end{aligned} \tag{65}$$

where the loss $\mathcal{L}$ is defined as:

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}\big(-H(Y|X) - H(X|Y)\big) \\ &= \frac{1}{2}\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\Big[\log p(\boldsymbol{x}|\boldsymbol{y})\Big] + \frac{1}{2}\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\Big[\log p(\boldsymbol{y}|\boldsymbol{x})\Big].\end{aligned} \tag{66}$$

End of proof.

Empirically, we use energy-based models to model the distributions. The condition on the existence of a and b can be understood as the requirements that the two distributions $(p_{\boldsymbol{x}}, p_{\boldsymbol{y}})$ are not collapsed.

F.1. Variational Representation Reconstruction

The variational representation reconstruction (VRR) was first introduced in GraphMVP (Liu et al., 2021a). There are two mirroring terms in Equation (64), and here we take one term for illustration. The goal of VRR is to take the variational lower bound to maximize:

$$\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\big[\log p(\boldsymbol{y}|\boldsymbol{x})\big]. \tag{67}$$

The objective function to Equation (67) has a variational lower bound as:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{z}_{\boldsymbol{x}}|\boldsymbol{x})}\big[\log p(\boldsymbol{y}|\boldsymbol{z}_{\boldsymbol{x}})\big] - KL(q(\boldsymbol{z}_{\boldsymbol{x}}|\boldsymbol{x})||p(\boldsymbol{z}_{\boldsymbol{x}})). \tag{68}$$

And VRR proposes a proxy solution b doing the reconstruction on the representation space instead of the data space:

$$\mathcal{L}_{\text{G}} = \mathcal{L}_{\text{VRR}} = \mathbb{E}_{q(\boldsymbol{z}_{\boldsymbol{x}}|\boldsymbol{x})}\big[\|q_{\boldsymbol{x}}(\boldsymbol{z}_{\boldsymbol{x}}) - \text{SG}(h_{\boldsymbol{y}})\|^2\big] + \beta KL(q(\boldsymbol{z}_{\boldsymbol{x}}|\boldsymbol{x})||p(\boldsymbol{z}_{\boldsymbol{x}})). \tag{69}$$

G. Implementation Details of MoleculeSDE

In this section, we illustrate the details of our proposed MoleculeSDE, including the featurization, backbone models, hyperparameters, architectures for the score networks, etc. The solution in Appendix F.1 to Equation (64) is indeed a conditional generative method to solving the self-supervised learning (SSL). Meanwhile, it is only a proxy solution by conducting the reconstruction on the representation space. Thus, we want to explore a more accurate and explicit estimation to the generative reconstruction on the data space (*i.e.*, the 2D topology and 3D geometry of molecules).

G.1. Backbone Models

For the backbone models, we stick with the existing pretraining works (Hu et al., 2019; Wang et al., 2022c; Liu et al., 2021a; 2023a), which can better illustrate the effectiveness of our proposed methods. We use Graph Isomorphism Network (GIN) (Xu et al., 2018) for modeling the 2D topology and SchNet (Schütt et al., 2018) for 3D conformation, respectively.

G.2. Molecule Featurization

The molecule featurization is an essential factor that should be taken into consideration. A recent work (Sun et al., 2022) has empirically verified that utilizing the rich atom feature. We follow this strategy and employ the featurization from MoleculeNet (Wu et al., 2018) and OGB (Hu et al., 2020a). In specific, we have the atom and bond featurization in Table 7.

Table 7. Featurization for atoms and bonds.

	Hyperparameter	Value
Atom Featurization	Atom Type	[0, 118]
	Atom Chirality	{unspecified, unrecognized type, tetrahedral with clockwise rotation, tetrahedral: counter-clockwise rotation}
	Atom Degree	[0, 10]
	Formal Charge	[-5, 5]
	Number of Hydrogen	[0, 8]
	Number of Unpaired Electrons	[0, 4]
	Hybridization	{SP, SP2, SP3, SP3D, SP3D2}
	Is Aromatic	{False, True}
	Is In Ring	{False, True}
Bond Featurization	Bond Type	{single, double, triple, aromatic}
	Bond Stereotype	{none, Z variant, E variant, Cis, Trans, any}
	Is conjugated	{False, True}

We also want to highlight that such an atom featurization is only available for the topological graph; while for 3D conformation, only the atom type information is available. The other atom information requires either the topology information (*e.g.*, degree, number of Hydrogen) or chemical rules (*e.g.*, chirality) to obtain, and they have not been utilized for the molecule geometric modeling (Schütt et al., 2018).

G.3. Pretraining Hyperparameters

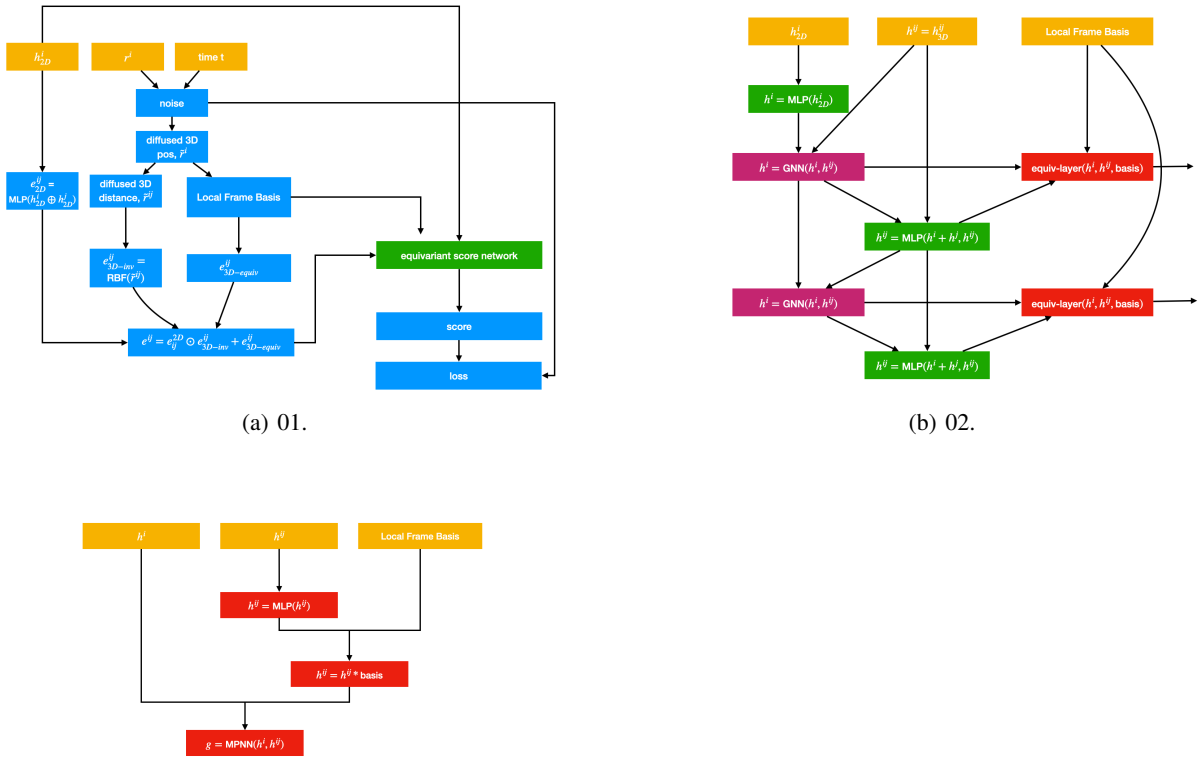
The pretraining pipeline is shown in Figure 1 and the objective function is Section 4.4. Below in Table 8, we illustrate the key hyperparameters used in MoleculeSDE.

Table 8. Hyperparameter specifications for MoleculeSDE.

Hyperparameter	Value
epochs	{50, 100}
learning rate 2D GNN	{1e-5, 1e-6}
learning rate 3D GNN	{1e-5, 1e-6}
SDE option	{VE, VP}
masking ratio M	{0, 0.3}
β	{[0.1, 10]}
number of steps	{1000}
α_1	{0, 1}
α_2	{0}
α_3	{0}

G.4. SE(3)-Equivariant SDE Model: From Topology to Conformation

We list the detailed structure of the SE(3)-equivariant SDE model in Figure 3.



(a) 01.

(b) 02.

(c) 03.

Figure 3. Pipeline for the SE(3)-equivariant SDE model from topology to conformation.

G.5. SE(3)-Invariant SDE Model: From Conformation to Topology

We list the detailed structure of the SE(3)-invariant SDE model in Figure 4.

Similarly with the $2D \to 3D$ procedure, we first merge the 3D representation $\boldsymbol{y}$ with the diffused atom feature $\boldsymbol{X}_t$:

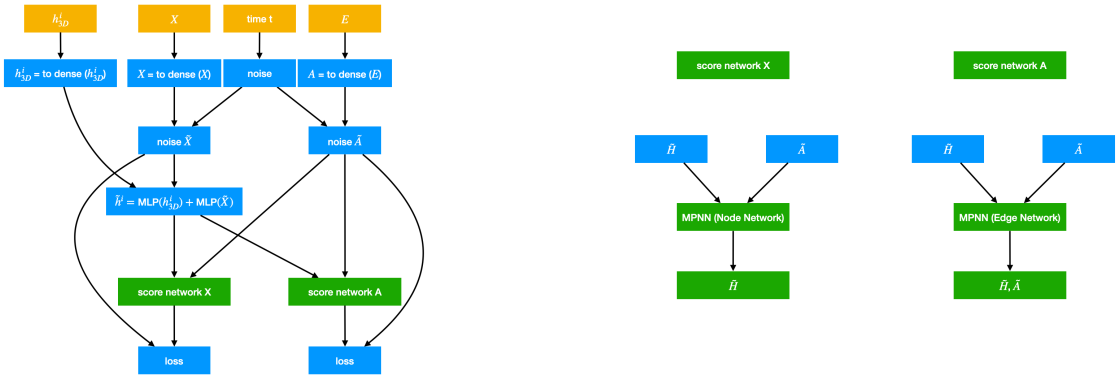
$$H_0 = \mathbf{MLP}(\boldsymbol{X}_t) + \boldsymbol{y}.$$

Since the noised $\boldsymbol{E}_t$ becomes a dense adjacency matrix, we will implement a densed GCN to model $\nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$:

$$S_\theta^{\boldsymbol{X}_t}(\boldsymbol{x}_t) = \mathbf{MLP}(\text{Concate}\{H_0||\cdots||H_L\}),$$

where $H_{i+1} = \mathbf{GCN}(H_i, \boldsymbol{E}_t)$. On the other hand, $\nabla_{\boldsymbol{E}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) \in \mathbf{R}^{n\times n}$ is modeled by an unnormalized dot product attention (without softmax):

$$S_\theta^{\boldsymbol{E}_t}(\boldsymbol{x}_t) = \mathbf{MLP}(\{\text{Attention}(H_i)\}_{0\le i\le L}).$$



(a) 01. (b) 02.

Figure 4. Pipeline for the SE(3)-invariant SDE model from conformation to topology.

H. Ablation Studies

This section provides more ablation studies to verify key concepts in molecule pretraining.

H.1. Ablation Study on Generative SSL Pretraining

We first provide a comprehensive comparison of the effect of the generative SSL part.

Pretraining. In GraphMVP (Liu et al., 2021a), the generative SSL is variational representation reconstruction (VRR). In MoleculeSDE, the generative SSL is composed of two SDE models.

Downstream. We consider both the 2D and 3D downstream tasks. We can tell that the generative SSL (SDE)) in MoleculeSDE is better than the generative SSL (VRR) in GraphMVP by a large margin on 27 out of 28 tasks.

Table 9. Ablation studies on generative SSL comparison. Results for molecular property prediction tasks (with 2D topology only). The best results are marked in **bold** and **bold**, respectively.

Pre-training	BBBP ↑	Tox21 ↑	ToxCast ↑	Sider ↑	ClinTox ↑	MUV ↑	HIV ↑	Bace ↑	Avg ↑
VRR (GraphMVP)	62.4±1.71	73.6±1.09	61.4±0.56	57.2±1.11	**86.5±3.02**	75.5±1.58	75.4±0.96	72.7±2.16	70.61
SDE-VE (MoleculeSDE)	**68.8±3.53**	**76.5±0.28**	**64.9±0.14**	**59.2±0.44**	**86.1±2.15**	**77.7±2.15**	**77.0±0.66**	**79.6±0.66**	**73.73**
SDE-VP (MoleculeSDE)	**65.5±3.25**	**75.6±0.36**	**63.4±0.22**	**59.8±0.23**	81.1±1.83	**80.1±1.10**	**78.6±0.31**	**79.0±0.79**	**72.89**

Table 10. Ablation studies on generative SSL comparison. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error. The best results are marked in **bold** and **bold**, respectively.

Pretraining	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
VRR (GraphMVP)	0.058	44.64	27.32	22.50	0.030	0.030	14.96	14.69	0.127	14.35	13.96	1.680
SDE-VE (MoleculeSDE)	**0.056**	**41.84**	**25.79**	**21.63**	**0.027**	**0.029**	**11.47**	**10.71**	**0.233**	**11.04**	**10.95**	**1.474**
SDE-VP (MoleculeSDE)	**0.056**	**42.75**	**25.84**	**21.52**	**0.027**	**0.029**	**11.90**	**11.85**	**0.200**	**12.03**	**11.69**	**1.453**

Table 11. Ablation studies on generative SSL comparison. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is the mean absolute error. The best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
VRR (GraphMVP)	**1.177**	0.389	0.533	**0.828**	0.562	0.806	**0.528**	0.717
SDE-VE (MoleculeSDE)	1.247	**0.364**	**0.448**	**0.735**	**0.483**	**0.785**	**0.480**	**0.575**
SDE-VP (MoleculeSDE)	**1.087**	**0.358**	**0.300**	0.880	**0.517**	**0.788**	0.540	**0.675**

H.2. Ablation Study on Atom Features and Comparison with Conformation Generation Methods

As recently discussed in (Sun et al., 2022), the atom feature plays an important role in molecule modeling, especially for 2D topological modeling. We carefully consider this in our work.

Note that for GIN in Table 2, we are using comprehensive atom and bond features, as shown in Appendix G.2. For the ablation study in Table 5, to make it a fair comparison between GIN and SchNet, we further employed merely the atom type (the same as 3D conformation modeling) and the bond type for 2D topology modeling. We name these two as "GIN with rich features" and the GIN in Table 2 as "GIN with simple features", respectively. The results and comparison with conformation generation methods are shown in Table 12.

Table 12. Ablation study on the effect of rich features for GIN and comparison with SchNet on conformation generation (CG) methods.

Model	CG Method	BBBP	Sider	ClinTox	Bace
GIN with rich features	–	68.1±0.59	57.0±1.33	83.7 ±2.93	76.7±2.51
GIN with simple features	–	64.1±1.79	58.4±0.50	63.1±7.21	76.5±2.96
SchNet	MMFF	61.4±0.29	59.4±0.27	64.6±0.50	74.3±0.66
SchNet	ConfGF	62.7±1.97	60.1±0.87	64.1±2.83	73.2±3.53
SchNet	ClofNet	61.7±1.19	56.0±0.10	58.2±0.44	62.5±0.17
SchNet	MoleculeSDE	65.2±0.43	60.5±0.39	72.9±1.02	78.6±0.40

Observation 1. We can tell that using rich or simple features plays an important role in GIN model. This can be observed when comparing the GIN in Table 2 and SchNet in Table 5, and we summarize them in the first two rows in Table 12.

Observation 2. Additionally, we can tell that SchNet on MoleculeSDE can outperform GIN with simple features, showing that in terms of the MoleculeSDE can extract more useful geometric information. Meanwhile, GIN with rich features performs better on two tasks, especially a large margin in ClinTox. This reveals that the heuristic 2D topological information can also convey some information that is missing in MoleculeSDE.

Thus, the main message we want to deliver to the audience is that MoleculeSDE is better in terms of the conformation generation, and can be combined with the 2D topology modeling for future works.

H.3. Ablation Study on The Effect of Contrastive Learning in MoleculeSDE

Table 13. Ablation studies on α_1 in MoleculeSDE. Results for molecular property prediction tasks (with 2D topology only). The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	BBBP ↑	Tox21 ↑	ToxCast ↑	Sider ↑	ClinTox ↑	MUV ↑	HIV ↑	Bace ↑	Avg ↑
VE	0	68.8±3.53	**76.5±0.28**	64.9±0.14	59.2±0.44	86.1±2.15	77.7±2.15	77.0±0.66	79.6±0.66	73.73
	1	**73.2±0.48**	**76.5±0.33**	**65.2±0.31**	**59.6±0.82**	**86.6±3.73**	**79.9±0.19**	**78.5±0.28**	**80.4±0.92**	**74.98**
VP	0	65.5±3.25	75.6±0.36	63.4±0.22	59.8±0.23	81.1±1.83	80.1±1.10	78.6±0.31	79.0±0.79	72.89
	1	**71.8±0.76**	**76.8±0.34**	**65.0±0.26**	**60.8±0.39**	**87.0±0.53**	**80.9±0.37**	**78.8±0.92**	**79.5±2.17**	**75.07**

Table 14. Ablation studies on α_1 in MoleculeSDE. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is the mean absolute error. The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
VE	0	0.056	**41.84**	25.79	21.63	0.027	**0.029**	**11.47**	**10.71**	0.233	**11.04**	**10.95**	**1.474**
	1	**0.055**	41.88	**25.62**	**21.51**	**0.026**	**0.029**	12.91	12.37	**0.142**	12.68	12.56	1.608
VP	0	0.056	42.75	25.84	21.52	0.027	0.029	**11.90**	**11.85**	0.200	**12.03**	**11.69**	**1.453**
	1	**0.054**	**41.77**	**25.74**	**21.41**	**0.026**	**0.028**	13.07	12.05	**0.151**	12.54	12.04	1.587

Table 15. Ablation studies on α_1 in MoleculeSDE. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is the mean absolute error. The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
VE	0	1.247	0.364	0.448	0.735	0.483	0.785	**0.480**	0.575
	1	**1.112**	**0.304**	**0.282**	**0.520**	**0.455**	**0.725**	0.515	**0.447**
VP	0	**1.087**	0.358	**0.300**	0.880	0.517	0.788	0.540	0.675
	1	1.244	**0.315**	0.338	**0.488**	**0.432**	**0.712**	**0.478**	**0.468**

H.4. PaiNN as Backbone

Table 16. Results on 12 quantum mechanics prediction tasks from QM9, and the backbone model is PaiNN. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error, and the best and the second best results are marked in **bold** and **bold**, respectively.

Pretraining	$\alpha \downarrow$	$\nabla\mathcal{E} \downarrow$	$\mathcal{E}_{\text{HOMO}} \downarrow$	$\mathcal{E}_{\text{LUMO}} \downarrow$	$\mu \downarrow$	$C_v \downarrow$	$G \downarrow$	$H \downarrow$	$R^2 \downarrow$	$U \downarrow$	$U_0 \downarrow$	ZPVE $\downarrow$
–	0.049	42.73	24.46	20.16	0.016	0.025	8.43	7.88	0.169	8.18	7.63	1.419
Distance Prediction	0.049	37.23	22.75	18.26	0.014	0.030	9.31	9.35	0.143	9.85	9.07	1.566
3D InfoGraph	0.047	44.25	24.06	18.54	0.015	0.052	8.81	7.97	0.143	8.68	8.08	1.416
GeoSSL-RR	0.046	41.20	23.93	19.36	0.016	0.025	8.32	8.17	0.174	7.99	8.20	1.438
GeoSSL-InfoNCE	0.045	39.29	23.23	18.40	0.015	0.024	8.34	8.37	**0.127**	7.45	8.34	1.356
GeoSSL-EBM-NCE	0.045	38.87	22.71	17.89	0.014	0.082	8.28	7.35	0.130	7.85	7.68	1.338
3D InfoMax	0.046	36.97	21.31	17.69	0.014	0.024	8.38	7.36	0.135	8.60	7.99	1.453
GraphMVP	0.044	36.03	20.71	17.02	0.014	0.024	8.31	7.36	0.132	7.57	7.34	1.337
GeoSSL-DDM-1L	0.045	36.13	20.59	17.26	0.014	0.024	9.45	8.43	0.128	8.88	8.16	1.380
GeoSSL-DDM	**0.043**	35.55	20.57	**16.95**	0.014	0.024	8.25	7.42	**0.127**	7.36	7.34	1.334
Uni-Mol	0.277	40.56	21.25	23.99	0.014	0.039	9.16	9.14	0.340	9.31	8.59	1.433
MoleculeSDE (VE)	0.044	**34.67**	**20.14**	17.05	**0.013**	**0.023**	**7.64**	**7.05**	0.139	**6.88**	**6.79**	**1.273**
MoleculeSDE (VP)	**0.042**	**35.09**	**20.14**	**16.78**	**0.013**	**0.023**	**8.17**	**7.01**	0.133	**7.30**	**7.05**	**1.315**

Table 17. Results on eight force prediction tasks from MD17, and the backbone model is PaiNN. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is mean absolute error, and the best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin $\downarrow$	Benzene $\downarrow$	Ethanol $\downarrow$	Malonaldehyde $\downarrow$	Naphthalene $\downarrow$	Salicylic $\downarrow$	Toluene $\downarrow$	Uracil $\downarrow$
–	0.572	0.053	0.230	0.338	0.132	0.288	0.141	0.201
Distance Prediction	0.480	0.053	0.200	0.296	0.131	0.265	0.171	0.168
3D InfoGraph	0.554	0.067	0.249	0.353	0.177	0.331	0.179	0.213
GeoSSL-RR	0.559	0.051	0.262	0.368	0.146	0.303	0.154	0.202
GeoSSL-InfoNCE	0.428	0.051	0.197	0.337	0.127	0.247	0.136	0.169
GeoSSL-EBM-NCE	0.435	0.048	0.198	**0.295**	0.143	0.245	0.132	0.172
3D InfoMax	0.479	0.052	0.220	0.344	0.138	0.267	0.155	0.174
GraphMVP	0.465	0.050	0.205	0.316	**0.119**	0.242	0.136	0.168
GeoSSL-DDM-1L	0.436	0.048	0.209	0.320	**0.119**	0.249	0.132	0.177
GeoSSL-DDM	**0.427**	0.047	**0.188**	0.313	0.120	**0.240**	**0.129**	0.167
Uni-Mol	0.487	0.048	0.217	0.329	0.151	0.299	0.141	0.182
MoleculeSDE (VE)	**0.421**	**0.043**	0.195	**0.284**	**0.105**	**0.236**	**0.123**	**0.158**
MoleculeSDE (VP)	0.443	**0.045**	**0.191**	0.301	0.131	0.261	0.140	**0.159**

H.5. Experiments on Conformation Generation

Notice that MoleculeSDE also supports the 2D to 3D conformation generation tasks, and we compare the following state-of-the-art models in Table 18.

Table 18. Results on conformation generation without FF optimization. The datasets are GEOM QM9 and GEOM Drugs.

Methods	GEOM QM9				GEOM Drugs			
	COV (%) $\uparrow$		MAT (Å) $\downarrow$		COV (%) $\uparrow$		MAT (Å) $\downarrow$	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
RDKit	83.26	90.78	0.3447	0.2935	60.91	65.70	1.2026	1.1252
ConfGF (Shi et al., 2021)	88.49	94.13	0.2673	0.2685	62.15	70.93	1.1629	1.1596
DMGC (Zhu et al., 2022)	96.23	99.26	0.2083	0.2014	96.52	100.00	0.7220	0.7161
RMCF-R (Wang et al., 2022b)	–	–	–	–	82.25	90.77	0.839	0.789
RMCF-C (Wang et al., 2022b)	–	–	–	–	87.12	96.26	0.749	0.709
MoleculeSDE (ours)	92.37	97.21	0.2423	0.2356	85.42	99.49	0.9485	0.9041

I. Computational Cost on Pretraining

Table 19. Computational time on 17 pretraining algorithms. All the jobs are running on one single V100 GPU card. The four models in the first block are 2D SSL, the nine models in the second block are 3D SSL, and the four models in the last block are 2D-3D SSL.

Pretraining Algorithm	min / epoch
AttrMask	5.5 min/epoch
ContextPred	14 min/epoch
InfoGraph	6 min/epoch
MolCLR	10 min/epoch
Type Prediction	7.75 min/epoch
Distance Prediction	6.7 min/epoch
Angle Prediction	8 min/epoch
3D InfoGraph	7.5 min/epoch
RR	9.7 min/epoch
InfoNCE	10 min/epoch
EBM-NCE	10.8 min/epoch
GeoSSL-1L	11.2 min/epoch
GeoSSL	18 min/epoch
3D InfoMax	8.6 min/epoch
GraphMVP	11 min/epoch
MoleculeSDE (VE)	30 min/epoch
MoleculeSDE (VP)	30 min/epoch