Direct Molecular Conformation Generation

Jinhua Zhu 1  Yingce Xia 2  Chang Liu 2  Lijun Wu 2  Shufang Xie 2  Tong Wang 2  Yusong Wang 3  Wengang Zhou 1  Tao Qin 2  Houqiang Li 1  Tie-Yan Liu 2

Abstract

Molecular conformation generation aims to generate three-dimensional coordinates of all the atoms in a molecule and is an important task in bioinformatics and pharmacology. Previous distance-based methods first predict interatomic distances and then generate conformations based on them, which could result in conflicting distances. In this work, we propose a method that directly predicts the coordinates of atoms. We design a dedicated loss function for conformation generation, which is invariant to roto-translation of coordinates of conformations and permutation of symmetric atoms in molecules. We further design a backbone model that stacks multiple blocks, where each block refines the conformation generated by its preceding block. Our method achieves state-of-the-art results on four public benchmarks: on small-scale GEOM-QM9 and GEOM-Drugs which have 200K training data, we can improve the previous best matching score by 3.5% and 28.9%; on large-scale GEOM-QM9 and GEOM-Drugs which have millions of training data, those two improvements are 47.1% and 36.3%. This shows the effectiveness of our method and the great potential of the direct approach. Our code is released at https://github.com/DirectMolecularConfGen/DMCG.

1. Introduction

Molecular conformation generation aims to generate 3D coordinates of all the atoms of a molecule, which can then be used in molecular property prediction (Axelrod & Gomez-Bombarelli, 2021), docking (Roy et al., 2015), structure-based virtual screening (Kontoyianni, 2017), etc. While molecular conformation is physically obtainable via X-ray crystallography, it is prohibitively costly for industry-scale tasks (Mansimov et al., 2019). Ab initio methods, e.g., based on density functional theory (DFT) (Parr, 1980; Baseden & Tye, 2014), can accurately predict molecular shapes, but take up to several hours per small molecule (Hu et al., 2021). To handle large molecules, people turn to leverage classical force fields, like UFF (Rappe et al., 1992) or MMFF (Halgren, 1996), to estimate conformations, which is efficient but unacceptably inaccurate (Kanal et al., 2018).

Recently, machine learning methods have attracted much attention for conformation generation due to their accuracy and efficiency. Two kinds of approaches have been studied: interatomic distance-based approach and direct approach. Given a molecule, the first approach first predicts interatomic distances (Simm & Hernandez-Lobato, 2020; Shi et al., 2020; Xu et al., 2021a;b) or the gradients w.r.t interatomic distances (Shi et al., 2021; Luo et al., 2021b) and then generates atomic coordinates based the distances or the gradients. The advantage of this approach is the invariance under rotation and translation of coordinates. However, the predicted distances might conflict with each other, e.g., violating the triangle inequality for distances among three atoms. Particularly, we found that GraphDG (Simm & Hernandez-Lobato, 2020), a representative distance-based method, produces distance matrices violating the triangle inequality on 8.65% of molecules in the GEOM-Drugs (Axelrod & Gomez-Bombarelli, 2021) test set.

The second approach directly generates coordinates of all atoms in a molecule without using interatomic distances or their gradients (Mansimov et al., 2019). This is a straightforward and natural choice that avoids all kinds of distance violations, and has demonstrated remarkable performance by AlphaFold 2 (Jumper et al., 2021) for protein structure prediction. A challenge of this approach is to maintain roto-translation invariance and permutation invariance. Specifically, (1) rotating and translating the coordinates of all atoms together do not change the conformation of a molecule, which should be taken into consideration for the direct approach; (2) As shown in Figure 1, due to the symmetry of the pyrimidine part along the C-S bond (atom 11 and 12), some other constraints might be violated, such as degree of freedom of the distance matrix and rank of squared distance matrix. See Appendix D for more details.

1University of Science and Technology of China 2Microsoft Research Asia 3Xi’an Jiaotong University.
Main Contact: Jinhua Zhu <teslazhu@mail.ustc.edu.cn>, Yingce Xia <yingce.xia@microsoft.com>, Chang Liu <chang.liu@microsoft.com>.

Copyright 2022 by the author(s).
We conduct experiments on four benchmarks: GEOM-QM9 and GEOM-Drugs with the small-scale setting (Shi et al., 2021) and large-scale setting (Axelrod & Gomez-Bombarelli, 2021). The small-scale GEOM-QM9 and GEOM-Drugs have 200K molecule-conformation pairs for training, and the large-scale GEOM-QM9 and GEOM-Drugs have 1.37M and 2.0M training pairs. Our method achieves state-of-the-art results on all of them, demonstrating the effectiveness of our method. Specifically, on small-scale GEOM-QM9, our method improves the mean coverage score and mean matching score by 5.3% and 3.5%. On small-scale GEOM-Drugs, the improvements are 22.8% and 28.9%. On the large-scale settings, the improvements are more significant: 10.2% and 47.1% for GEOM-QM9, and 35.7% and 36.3% for GEOM-Drugs.

Our main contributions are summarized as follows:

1) We design a dedicated loss function, that can maintains both permutation invariance on symmetric atoms and roto-translation invariance on conformations.

2) We design a new model that iteratively refines the conformation. Our model is inspired by multiple advanced architectures like GATv2 (Brody et al., 2021) and GN block (Battaglia et al., 2018), which can effectively model molecules.

3) Our method outperforms strong baselines and achieves state-of-the-art results on all four benchmarks we tested.

Problem Definition: Let $G = (V, E)$ denote a molecular graph, where $V$ and $E$ are collections of atoms and bonds, respectively. Specifically, $V = \{v_1, v_2, \cdots, v_{|V|}\}$ with the $i$-th atom $v_i$. Let $e_{ij}$ denote the bond between atom $v_i$ and $v_j$. For ease of reference, we simply use $i \in V$ and $(i, j) \in E$ to denote the $i$-th atom in $V$ and the bond $e_{ij}$ in $E$. Let $N(i)$ denote the neighbors of atom $i$, i.e., $N(i) = \{j \mid (i, j) \in E\}$. We use $R$ to represent the conformation of $G$, where $R \in \mathbb{R}^{|V| \times 3}$. The $i$-th row of $R$ (denoted as $R_i$) is the coordinate of atom $v_i$. Given a graph $G = (V, E)$, our task is to learn a mapping, that can output the coordinates $R$ of all atoms in $V$, i.e., $R \in \mathbb{R}^{|V| \times 3}$.

2. Training Objective

Molecular conformations have several well-known properties: (1) They are roto-translation invariant (Mansimov et al., 2019). (2) If a molecule has a symmetric substructure, its conformation should be invariant w.r.t the permutation of the corresponding symmetric atoms (Lowe & Peterson, 2011). (3) A molecule usually has a set of diverse conformations (O’Boyle et al., 2011). Thus, for machine learning based molecular conformation generation, the training objective should be roto-translation and permutation invariant and encourage diversity. In this section, we design a training objective that satisfies the three properties.

2.1. Roto-translation Invariance

Let $\rho(\cdot)$ denote a roto-translation operation, i.e., $\rho(R) = RQ + b$, where the orientation-preserving orthogonal transformation $Q \in SO(3) \subset \mathbb{R}^{3 \times 3}$ (i.e., the 3D rotation group) represents a rotation and the vector $b \in \mathbb{R}^{1 \times 3}$ represents a translation.

Let $R \in \mathbb{R}^{|V| \times 3}$ and $\hat{R} \in \mathbb{R}^{|V| \times 3}$ denote the groundtruth conformation and the generated conformation. The roto-translation invariant loss is defined as follows:

$$\ell_{RT}(R, \hat{R}) = \min_{\rho} \|\rho(\hat{R}) - R\|_F^2,$$

(1)

where $\| \cdot \|_F$ denotes the Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i,j} |A_{ij}|^2$. $\ell_{RT}$ is invariant to the rotation and translation of either of the two input conformations.

Karney (2007) propose to use quaternions (Hamilton, 1840)
to solve Eqn.(1). A quaternion \( q \) is an extension of complex numbers, \( q = q_0 \mathbf{u} + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} \), where \( q_0, q_1, q_2, q_3 \) are real scalars and \( \mathbf{u}, \mathbf{i}, \mathbf{j}, \mathbf{k} \) are orientation vectors. With quaternions, any rotation operation is specified by a \( 3 \times 3 \) matrix, where each element in the matrix is the summation/multiplication of \( q_0 \) to \( q_3 \). The solution to Eqn.(1) is the minimal eigenvalue of a \( 4 \times 4 \) matrix obtained by algebraic operations on \( R \) and \( \hat{R} \). To stabilize training, we stop gradient back-propagation through \( \rho \).

2.2. Permutation Invariance

As introduced in Section 1, the symmetry of molecule structure should be considered, and we propose a permutation invariant loss function in this subsection. Given a molecule graph \( G \) and a numbering of its atoms (as the example shown in Figure 1), a permutation \( \sigma \) is a bijection from the set \( \{1, 2, \cdots, |V|\} \) to itself. However, not all permutations yield the same conformation, and permutations only on symmetric fragments in the graph call for our attention. Let \( S \) denote the set of such permutations, which are formally characterized by: (1) \( \forall i \in V \), atom \( i \) and atom \( \sigma(i) \) have the same label; (2) There exists a bond between atoms \( i \) and \( j \) iff there exists a bond between atoms \( \sigma(i) \) and \( \sigma(j) \) in the same molecular graph. We note that molecular graphs are different from common graphs as the bond defines different edge types. Therefore, we assign the label to each atom as the union of the atom type itself and also the types of all the bonds connected to it. For example, in Figure 1, the label of atom 11 is “S-2Single”, and the label of atom 17 is “N-2Aromatic”. For the molecule in Figure 1, \( S \) contains two elements: the identity permutation, and a permutation that exchanges 13 and 14 to 16. The permutation invariant loss is then defined as:

\[
\ell_P(R, \hat{R}) = \min_{\sigma \in S} \| \sigma(\hat{R}) - R \|_F^2. \tag{2}
\]

Inspired by Meli & Biggin (2020), to solve Eqn.(2), we first use the graph_tool toolkit\(^2\) to find all permutations in \( S \), and then find the minimum for a given \( \hat{R} \) and \( R \) by enumerating all \( \sigma \in S \).

By combining Eqn.(1) and Eqn.(2), we obtain a roto-translation invariant and permutation invariant loss:

\[
\ell_{\text{RTP}}(R, \hat{R}) = \min_{\rho; \sigma \in S} \| R - \rho(\sigma(\hat{R})) \|_F^2. \tag{3}
\]

2.3. Encouraging Diversity

While the loss in Eqn.(3) ensures proper invariance, it tends to generate deterministic conformations without diversity. To encourage diversity, we introduce probabilistic sampling into conformation generation and refine the above loss in a probabilistic manner. Specifically, we introduce a latent random variable \( z \) and sample conformations from a conditional distribution

\[
p(R|z, G) \propto \exp \left\{ -\frac{1}{\gamma} \ell_{\text{RTP}}(R, \hat{R}(z, G)) \right\}, \tag{4}
\]

where \( \gamma > 0 \) is a hyperparameter (see Appendix B.3 for explanation), and \( \hat{R}(z, G) \) is our main model that outputs conformation coordinates from \( z \) and \( G \), whose implementation is detailed in Section 2.5 below.

Remark: Note that the usage of \( \ell_{\text{RTP}} \) guarantees that the probabilistic model assigns the same probability to all roto-translation and permutation equivalent coordinates, which describe the same conformation. Generation diversity is achieved by the diversity of random samples of \( z \) from a prior distribution, e.g., \( p(z) = N(0, I) \).

2.4. Overall Objective

As a probabilistic model, the goal now is to match the model-defined conformation distribution of a given molecule graph \( p(R|G) = \int p(z) p(R|z, G) dz \) to data. This can be done by maximum likelihood estimate, but evaluating this integral is intractable. This can be managed using the techniques in variational auto-encoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014; Sohn et al., 2015), where a conditional inference model \( q(z|R, G) \) is introduced, and it relates to the log-likelihood via the following identity:

\[
\log p(R|G) = \mathbb{E}_{q(z|R, G)}[\log p(R|z, G)]
- \mathbb{D}_{\text{KL}}(q(z|R, G)\|p(z|R, G))
+ \mathbb{D}_{\text{KL}}(q(z|R, G)\|p(z))
\]

where \( \mathbb{D}_{\text{KL}} \) denotes the Kullback-Leibler (KL) divergence, and \( \mathbb{D}_{\text{KL}}(q(z|R, G)\|p(z)) \) is the true posterior. Due to the nonnegativity of the third term, the first two terms form a lower bound of \( \mathbb{D}_{\text{KL}}(q(z|R, G)\|p(z)) \). Moreover, as the left-hand-side is independent of \( q(z|R, G) \), tightening the bound (i.e., minimizing the third term) can be achieved by maximizing the bound w.r.t \( q(z|R, G) \). Therefore, we have the following loss to minimize (negative sign added):

\[
\mathbb{E}_{q(z|R, G)}[-\log p(R|z, G)] + \mathbb{D}_{\text{KL}}(q(z|R, G)\|p(z)).
\]

When \( q(z|R, G) \) is properly chosen, the loss is tractable to optimize. We specify \( q(z|R, G) = N(z|\mu_{R,G}, \Sigma_{R,G}), \) where the conditional mean and variance are outputs from an encoder (i.e. \( \varphi_{2D} \) in Section 2.5). It enables tractable loss optimization via reparameterization (Kingma & Welling, 2014): \( z \sim q(z|R, G) \) is equivalent to \( z = \mu_{R,G} + \Sigma_{R,G}\epsilon \) where \( \epsilon \sim N(0, I) \). With the above specification, we get the following overall objective to minimize in training:

\[
\min_{\mu_{R,G}, \Sigma_{R,G}} \mathbb{E}_{\epsilon \sim N(0,I)}[\ell_{\text{RTP}}(R, \hat{R}(\mu_{R,G}, \Sigma_{R,G}, G))]
+ \beta \mathbb{D}_{\text{KL}}(N(\mu_{R,G}, \Sigma_{R,G})\|N(0, I)), \tag{5}
\]

where the minimization is applied over the \( \hat{R}(z, G) \) model and the \( (\mu_{R,G}, \Sigma_{R,G}) \) model. Note we have introduced an

\(^2\)https://graph-tool.skewed.de
According to the above specification, \( \psi \) minimizes Eqn. (5) for training. Recall that \( \psi \) is related to \( \varphi_{2D}, \varphi_{3D}, \varphi_{dec}, \) and \( \mu_{R,G}, \Sigma_{R,G} \) are related to \( \varphi_{3D} \).

The inference workflow is shown in Figure 2(b), where

Additional hyperparameter \( \beta \) in the spirit of \( \beta \)-VAE (Higgins et al., 2016) is leveraged to handle the strength of the prior regularization. The second term in Eqn. (5) has a closed-form expression for optimization.

### 2.5. Training and Inference Flow

Now we show the training and inference workflow. The training process involves three modules, \( \varphi_{2D}, \varphi_{3D} \) and \( \varphi_{dec} \). The workflow is illustrated in Figure 2(a). Specifically,

1. The encoder \( \varphi_{2D} \) takes the molecular graph \( G \) as its input, and outputs several representations: \( H^{(0)}_{V} \in \mathbb{R}^{|V| \times d} \) for all atoms, \( H^{(0)}_{E} \in \mathbb{R}^{|E| \times d} \) for all bonds, a global graph feature \( U^{(0)} \in \mathbb{R}^d \), and initial conformation \( \hat{R}^{(0)} \in \mathbb{R}^{l \times 3} \). Note \( d \) is the dimension of the representations. Formally, \( (H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)}, \hat{R}^{(0)}) = \varphi_{2D}(G) \).

2. The encoder \( \varphi_{3D} \) extracts features of the conformation \( R \) for constructing the conditional inference module \( \psi(z|R,G) \). According to the above specification, \( \varphi_{3D} \) only needs to output the mean and variance of the Gaussian, or formally, \( (\mu_{R,G}, \Sigma_{R,G}) = \varphi_{3D}(R,G) \).

3. We randomly sample a variable \( z \) from the Gaussian distribution \( N(\mu_{R,G}, \Sigma_{R,G}) \), and then feed \( H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)}, \hat{R}^{(0)}, z \) into the decoder \( \varphi_{dec} \) to obtain the conformation \( \hat{R}(z,G) \). That is, \( \hat{R}(z,G) = \varphi_{dec}(\varphi_{2d}(G), z) = \varphi_{dec}(H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)}, \hat{R}^{(0)}, z) \). Note that sampling \( z \sim N(\mu_{R,G}, \Sigma_{R,G}) \) is equivalent to sampling \( \epsilon \sim N(0, I) \) and then setting \( z = \mu_{R,G} + \Sigma_{R,G} \epsilon \).

4. After obtaining \( \hat{R}(z,G) \) and \( N(\mu_{R,G}, \Sigma_{R,G}) \), we optimize Eqn. (5) for training. Recall that \( \hat{R} \) is related to \( \varphi_{2D}, \varphi_{3D}, \varphi_{dec}, \) and \( \mu_{R,G}, \Sigma_{R,G} \) are related to \( \varphi_{3D} \).

The well-trained \( \varphi_{2D} \) and \( \varphi_{dec} \) are leveraged: (1) Given a molecular graph \( G \), we use \( \varphi_{2D} \) to encode \( G \) and obtain \( \hat{R}^{(0)}, H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)} \); (2) We sample a random variable \( z \) from Gaussian \( N(0, I) \); (3) We feed \( \hat{R}^{(0)}, H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)}, z \) into \( \varphi_{dec} \) and obtain the eventual conformation \( \hat{R}(z,G) \). Note that \( \varphi_{3D} \) is not used in inference phase.

### 3. Model Architecture

The encoders \( \varphi_{2D}, \varphi_{3D} \) and the decoder \( \varphi_{dec} \) share the same architecture. They all stack \( L \) identical blocks. Due to space limitations, here we take the decoder \( \varphi_{dec} \) as an example to introduce its \( l \)-th block, and leave the details of \( \varphi_{2D} \) and \( \varphi_{3D} \) to Appendix A.

Figure 3 shows the architecture of the \( l \)-th block of \( \varphi_{dec} \). Roughly speaking, this block takes the outputs from its preceding block (including the conformation \( \hat{R}^{(l-1)} \), atom representations \( H^{(l-1)}_{V} \), edge representations \( H^{(l-1)}_{E} \) and the global representation \( U^{(l-1)} \) of the whole molecule) and outputs refined conformation and representations of atoms, bonds, the whole graph. The process is repeated until the eventual output \( \hat{R}^{(L)} \) is obtained. For the input of the first block (i.e., \( l = 1 \)), the \( H^{(0)}_{V}, H^{(0)}_{E}, U^{(0)} \) and \( \hat{R}^{(0)} \) are the outputs of \( \varphi_{2D} \).

We use a variant of the GN block (Battaglia et al., 2018) as the backbone of our model due to its superior performance in molecular modeling. In each block, we first update bond representations, then atom representations, and finally the global molecule representation and the conformation. For ease of reference, let \( h_{i}^{(l)} \) denote the representation of atom \( i \) output by the \( l \)-th block, and \( h_{ij}^{(l)} \) the representation of the bond between atom \( i \) and \( j \).

Mathematically, the \( l \)-th block takes following operations:

1. **Update bond representations**: We first incorporate the
coordinate information into the representations, i.e.,
\[ h^{(l)}_i = h^{(l-1)}_i + MLP(R_i^{(l-1)}) + z, \forall i \in V, \]
\[ h^{(l)}_{ij} = h^{(l-1)}_{ij} + MLP(||R_i^{(l-1)} - R_j^{(l-1)}||), \forall (i, j) \in E, \]
where \( z \sim \mathcal{N}(\mu_{R,G}, \Sigma_{R,G}) \). After that, the bond representations are updated as follows: \( \forall (i, j) \in E, \)
\[ h^{(l)}_{ij} = h^{(l-1)}_{ij} + MLP(h^{(l-1)}_i, R_{ij}, h^{(l-1)}_j, U^{(l-1)}). \]

(2) **Update atom representations:** \( \forall i \in V, \)
\[ \hat{h}^{(l)}_i = \sum_{j \in \mathcal{N}(i)} \alpha_j W_{\text{concat}}(\hat{h}^{(l)}_{ij}, \hat{h}^{(l-1)}_j), \]
where \( \alpha_j \propto \exp(a^T \zeta(W_q h^{(l-1)}_j + W_k \text{concat}(\hat{h}^{(l-1)}_j, \hat{h}^{(l)}_i))); \)
\[ h^{(l)}_i = \hat{h}^{(l)}_i + MLP(\hat{h}^{(l-1)}_i, \hat{h}^{(l)}_i, U^{(l-1)}). \]

In Eqn.(6), \( a, W_q, W_v \) and \( W_k \) are the parameters to be learned, \( \text{concat}(\cdot, \cdot) \) is the concatenation of two vectors and \( \zeta \) is the leaky ReLU activation. For atom \( v_i \), we first use GATv2 (Brody et al., 2021) to aggregate the representations from its connected bonds to obtain \( \hat{h}^{(l)}_i \), and then update \( v_i \) based on \( \hat{h}^{(l)}_i, \hat{h}^{(l-1)}_i \) and \( U^{(l-1)} \).

(4) **Update global molecule representation:** \( \forall i \in V, \)
\[ U^{(l)} = U^{(l-1)} + MLP\left(\frac{1}{|V|} \sum_{i=1}^{|V|} h^{(l)}_i, \frac{1}{|E|} \sum_{i,j} h^{(l)}_{ij}, U^{(l-1)}\right). \]

(5) **Update the conformation:** \( \forall i \in V, \)
\[ R_i^{(l)} = \text{MLP}(h^{(l)}_i), \quad m^{(l)} = \frac{1}{|V|} \sum_{j=1}^{|V|} R_j^{(l)}, \]
\[ \tilde{R}_i^{(l)} = \tilde{R}_i^{(l)} - m^{(l)} + R_i^{(l-1)}. \]

An important step in Eqn.(7) is that, after making initial prediction \( \tilde{R}_i^{(l)} \), we calculate its center and normalize their coordinates by moving the center to the origin. This normalization ensures that the coordinates generated by each block are in reasonable numeric ranges.

We use \( \tilde{R}_i^{(L)} \) output by the last block in \( \varphi_{\text{dec}} \) as the final prediction of the conformation.

### 4. Discussions

CVGAE (Mansimov et al., 2019) is an early attempt to directly generating conformation. Unfortunately, its performance is not as good as distance-based methods developed afterwards (Shi et al., 2020; Simm & Hernández-Lobato, 2020). Our method, pursuing the same spirit, makes several finer designs: (1) We design a dedicated training objective that takes the invariance of both roto-translation and permutation on symmetric atoms into consideration. (2) We iteratively refine the output of each block, which is effective for conformation generation (see Figure 5 for ablation study).

In comparison, CVGAE only outputs the conformation in the last layer. (3) Our model integrates several advanced and more effective modules, including GATv2 (Brody et al., 2021) and GN block (Battaglia et al., 2018), while CVGAE mainly leverages GRU (Bahdanau et al., 2015) and its variants on graphs, which are outperformed by the modules used in our model.

ConfGF (Shi et al., 2021) and DGSM (Luo et al., 2021b) are two recent works that can also directly output the coordinates. They both model the gradient of log-density w.r.t interatomic distances, and then generate coordinates by running Langevin dynamics using the gradients. The gradient model is learned via score-matching. ConfGF considers the distances of 1-hop, 2-hop and 3-hop neighbors, and DGSM also considers distances of two randomly sampled nodes to model non-bonded distances. In comparison, we completely get rid of modeling distances. More importantly, the permutation invariance of symmetric atoms are not considered in those works. Ganea et al. (2021) propose another method for conformation generation: they first build the local structure (LS) by predicting the coordinates of non-terminal atoms, and then refine the LS by the predicted distances and dihedral angles. In comparison, our method does not require refinement based on the predicted distances and angles. Furthermore, although Ganea et al. (2021) use a permutation invariant loss, they only consider the terminal atoms, which is quite limited. We consider all symmetric atoms.

There are some other works on conformation generation, but they target at different problems. G-SchNet (Gebauer et al., 2019) takes some properties as input (not 2D graph) and output a conformation with desired properties. Luo et al. (2021a) focus on generating a conformation that can bind with specific binding pocket. We can combine our method with them in the future.

### 5. Experiments

#### 5.1. Settings

**Datasets:** Following prior works (Xu et al., 2021a; Shi et al., 2021), we use the GEOM-QM9 and GEOM-Drugs datasets (Axelrod & Gomez-Bombarelli, 2021) for conformation generation. We verify our method on both small-scale setting and large-scale setting. For the small-scale setting, we use the same datasets provided by Shi et al. (2021) for fair comparison with prior works. The training, validation and test sets of the two datasets consist of 200K, 2.5K and 22408 (for GEOM-QM9) / 14324 (for GEOM-Drugs) molecule-conformation pairs respectively. After that, we work on the large-scale setting by sampling larger datasets from the original GEOM to validate the scalability of our method. We use all data in GEOM-QM9 and
Model configuration: All of $\varphi_{2D}$, $\varphi_{3D}$ and $\varphi_{dec}$ have 3/6 blocks for the small-/large-scale settings. The dimension $d$ of the features is 256. Inspired by the feed-forward layer in Transformer (Vaswani et al., 2017), MLP also consists of two sub-layers, where the first one maps the input features from dimension 256 to hidden states, followed by Batch Normalization and ReLU activation. Then the hidden states is mapped to 256 again using linear mapping. More details are summarized in Appendix B.1.

Evaluation: Assuming in the test set, the molecule $x$ has $N_x$ conformations. Following Shi et al. (2020; 2021), for each molecule $x$ in the test set, we generate $2N_x$ conformations. Let $S_g$ and $S_r$ denote all generated and ground truth conformations respectively. We use coverage score (COV) and matching score (MAT) to evaluate the generation quality. To measure the difference between $R$ and $\hat{R}$, we use the GetBestRMS in the RDKit package and denote the root-mean-square deviation as $\text{RMSD}(R, \hat{R})$. We have
\[
\text{COV}(S_g, S_r) = \frac{1}{|S_g|} \left| \left\{ R \in S_r : \text{RMSD}(R, \hat{R}) < \delta, \exists \hat{R} \in S_g \right\} \right|;
\]
\[
\text{MAT}(S_g, S_r) = \frac{1}{|S_g|} \sum_{R \in S_r} \min_{\hat{R} \in S_g} \text{RMSD}(R, \hat{R}).
\]
A good method should have a high COV score and a low MAT score. Following Shi et al. (2021), the $\delta$’s are set as 0.5 and 1.25 for QM9 and Drugs, respectively.

Baselines: (1) RDKit, which is a widely used toolkit and generates the conformation based on the force fields; (2) CVGAE (Mansimov et al., 2019), which is an early attempt to generate raw coordinates; (3) GraphDG (Simm & Hernández-Lobato, 2020), a representative distance-based method with VAE; (4) CGCF (Xu et al., 2021a), which is another distance-based method leveraging continuous normalizing flow; (5) ConfVAE (Xu et al., 2021b), an end-to-end framework for molecular conformation generation, which still uses the pairwise distances among atoms as intermediate variables; (6) ConfGF (Shi et al., 2021) and DGSM (Luo et al., 2021b), which uses score matching to generate the gradients w.r.t distances and then recover the conformation; (7) GeoMol (Ganea et al., 2021), which predicts local atomic 3D structures and torsion angles. Considering Ganea et al. (2021) use a different data split from previous work, we reproduce their method following the more commonly used data split (Xu et al., 2021a; Shi et al., 2021).

5.2. Results

The results on the small-scale and large-scale datasets are in Table 1 and Table 2 respectively.

We have the following observations:
(1) On the four settings in Table 1 and Table 2, our method achieves state-of-the-art results on all of them. The median COV(%) being 100% means that for more than half of the reference molecules, these exist generated molecules that are close to them within a predefined threshold. These results show the effectiveness and scalability of our method.

(2) Our method achieves more improvement on molecules with more heavy atoms. Take the results in Table 1 as an example. On average, GEOM-QM9 and GEOM-Drugs have 8.8 and 24.9 heavy atoms respectively. In terms of MAT mean values, on GEOM-QM9, our method improves ConfGF and DGSM by 22.7% and 3.5%, while on GEOM-Drugs, the improvements are 61.0% and 40.5%. The results demonstrate the effectiveness of our method on large molecules. More results and analysis are in Appendix C.5.

(3) Our method is much more sample-efficient than methods based on Langevin dynamics like ConfGF, since we can generate IID samples free of the auto-correlation in a Markov chain. ConfGF requires 5000 sequential forward steps, while we only need to sample once from $\mathcal{N}(0, I)$ and forward through the model. For a fair comparison, following the official implementation of ConfGF, we split the test sets of small-scale GEOM-QM9 and GEOM-Drugs into 200 batches. ConfGF requires 8511.60 and 11830.42 seconds to decode QM9 and Drugs test sets, while our method only requires 32.68 and 54.89 seconds respectively. Our method speeds up the decoding more than 200 times.

5.3. Molecular Docking

Molecular docking (Roy et al., 2015) is a widely used technique in drug discovery, which aims to find the optimal binding conformation of a drug (i.e., the small molecule) in the pocket of a given target protein and the corresponding binding affinity. In most cases, the molecular docking algorithms treat proteins as rigid bodies and take one conformation of the small molecules as the initial structure inputs. The algorithms then search for the optimal conformation in the conformation space of the small molecules guided by the scoring function. However, due to the complexity of the conformation space, it is difficult for the algorithm to converge to a global minimum. Therefore, the choice of the initial structure often leads to different binding conformations and needs to be taken seriously.

Previously, RDKit was often used to generate initial conformations of small molecules, which usually got reasonable but not optimal results after docking. To verify the effectiveness of our method, we compares the docked poses which take initial conformations generated by our method, ConfGF and RDKit as the initial conformations for docking respectively.
Table 1: Experimental results on small-scale datasets. Bond fonts indicate the best results.

| Dataset | COV(%)↑ | MAT (Å)↓ | COV(%)↑ | MAT (Å)↓ |
|---------|---------|----------|---------|----------|
| QM9     | Mean    | Median   | Mean    | Median   |
| RDKit   | 83.26   | 90.78    | 0.3447  | 0.2935   |
| CVGAE   | 0.09    | 0.00     | 1.6713  | 1.6088   |
| GraphDG | 73.33   | 84.21    | 0.4245  | 0.3973   |
| CGCF    | 78.05   | 82.48    | 0.4219  | 0.3900   |
| ConfVAE | 80.42   | 85.31    | 0.4066  | 0.3891   |
| GeoMol  | 71.26   | 72.00    | 0.3731  | 0.3731   |
| ConfGF  | 88.49   | 94.13    | 0.2673  | 0.2685   |
| Ours    | 96.34   | 99.53    | 0.2065  | 0.2003   |

Table 2: Experimental results on large-scale datasets. Bond fonts indicate the best results.

| Dataset | COV(%)↑ | MAT (Å)↓ | COV(%)↑ | MAT (Å)↓ |
|---------|---------|----------|---------|----------|
| QM9     | Mean    | Median   | Mean    | Median   |
| RDKit   | 81.61   | 85.71    | 0.2643  | 0.2472   |
| CVGAE   | 0.00    | 0.00     | 1.4687  | 1.3758   |
| GraphDG | 13.48   | 5.71     | 0.9511  | 0.9180   |
| CGCF    | 81.48   | 86.95    | 0.3598  | 0.3684   |
| ConfVAE | 80.18   | 85.87    | 0.3684  | 0.3776   |
| ConfGF  | 89.21   | 95.12    | 0.2809  | 0.2837   |
| Ours    | 98.34   | 100.00   | 0.1486  | 0.1340   |

We use Smina (Koes et al., 2013) for molecular docking and make evaluation on PDBbind refined set (Liu et al., 2017) which is a comprehensive collection of experimentally measured binding affinity for all biomolecular complexes deposited in the Protein Data Bank (wwp, 2019). We randomly select 100 protein-ligand pairs for evaluation. Appendix B.2 shows detailed optimization hyper-parameters.

Two metrics were used to evaluate the results of docking. One is the binding affinity, which measures how well a molecule fits the binding site. A smaller value indicates better binding affinity. The other is the root-mean-square deviation (RMSD, the smaller, the better) compared to the crystal complex structure. As shown in Figure 4(a), the distributions for the three methods have the similar shape but our method is much more left-shifted than the others. This shows that for the same small molecule, our method tends to help docking to find conformations with higher binding affinities. Furthermore, docking tends to find lower RMSD binding conformations using the conformation generated by our method as the initial conformation, suggesting that our method can help docking to find binding conformations that are closer to the native crystal structures (Figure 4(b)). We also summarize the mean values of the binding affinity and RMSD of different algorithms in Table 3. All these results show that our method provides more proper initial conformations for molecular docking and thus facilitates the real application in computer-aided drug discovery.

Table 3: Mean values of the binding affinity scores and RMSD scores of different methods.

| Method | -Affinity (kcal/mol) | RMSD (Å) |
|--------|----------------------|----------|
| RDKit  | -8.85                | 4.24     |
| ConfGF | -9.87                | 3.91     |
| Ours   | -10.78               | 3.73     |

5.4. Property Prediction

In addition to conformation generation task, we also conduct experiments on property prediction task, which is to predict molecular property based on an ensemble of generated
conformation (Axelrod & Gomez-Bombarelli, 2021). We first randomly choose 30 molecules from GEOM-QM9 test sets, and then sample 50 conformations for each molecule using RDKit, ConfGF and our method. We use the quantum chemical calculation package Psi4 (Smith et al., 2020) to calculate the energy, HOMO and LUMO for each generated conformation and groundtruth conformation. Next, we calculate the ensemble properties of average energy $E$, lowest energy $E_{min}$, average HOMO-LUMO gap $\Delta \epsilon$, minimum gap $\Delta \epsilon_{min}$ and maximum gap $\Delta \epsilon_{max}$ based on the conformational properties of each molecule. We use mean absolute error to measure the property differences between the generated conformations and groundtruth conformations.

Table 4: Mean absolute error of predicted ensemble properties. (Unit: eV).

| Methods | $E$   | $E_{min}$ | $\Delta \epsilon$ | $\Delta \epsilon_{min}$ | $\Delta \epsilon_{max}$ |
|---------|-------|-----------|---------------------|--------------------------|-------------------------|
| RDKit   | 0.8875| 0.6530    | 0.3484              | 0.5570                   | 0.2399                  |
| GraphDG | 45.1088| 9.2868    | 3.8970              | 6.6997                   | 1.7724                  |
| ConfGF  | 2.8349| 0.2012    | 0.6903              | 4.9221                   | 0.1820                  |
| Ours    | 0.4324| 0.1364    | 0.2057              | 1.3229                   | 0.1509                  |

The results are shown in Table 4. Our method significantly outperforms GraphDG and ConfGF, which shows the effectiveness of our method. We can observe that RDKit achieves the best results on $\Delta \epsilon_{min}$, and we will combine our method with RDKit in the future. The evaluation results using median absolute error are reported in Appendix C.6.

5.5. Ablation Study

We conduct ablation study on the small-scale GEOM-Drugs dataset. The results are shown in Table 5.

1. We remove the permutation invariant loss and use the roto-translation invariant loss only, i.e., the $\ell_{RTP}$ in Eqn.(5) is replaced with $\ell_{RT}$ defined in Eqn.(2). The results are denoted as “No $\ell_T$” in Table 5.

| Methods | COV(%)† | MAT (Å)↓ |
|---------|---------|----------|
| Ours    | Mean    | Median   | Mean    | Median   |
| No $\ell_T$ | 77.78   | 86.00    | 1.0657  | 1.0563  |
| No attention | 94.99   | 100.00   | 0.7611  | 0.7581  |
| No normalization | 92.77  | 98.68    | 0.8002  | 0.7977  |

18.91 while MAT increases 0.3434. We also visualize several cases in Appendix C.4 to compare the results with or without $\ell_P$. (2) Without attentively aggregating the atom features, the mean COV drops 1.70 points and MAT score increases 0.0345 points. (3) Without the conformation normalization, the performance is also hurt. These results demonstrate the importance of the components in our method.

Finally, we compute the COV and MAT scores of $\hat{R}^{(l)}$ against the groundtruth, which is the output conformation of the $l$-th block in the decoder. $\hat{R}^{(l)}$ is the output of $\varphi_{2D}$. The results are shown in Figure 5. We can see that iteratively refining the conformations can improve the performances, which shows the effectiveness of our design. This phenomenon is consistency with the discovery in machine translation (Xia et al., 2017), image synthesis (Chen & Koltun, 2017) and protein structure prediction (Jumper et al., 2021).

Figure 5: The MAT and COV scores of $\hat{R}^{(l)}$ output by different blocks.

Due to space limitations, we leave the discussion about additional constraints on loss functions, the comparison of model sizes and visualization in Appendix C.

6. Conclusions and Future Work

In this work, we propose a new method, that directly generates the coordinates of conformations. For this purpose, we design a dedicated loss function, which is invariant to roto-translation and permutation on symmetric atoms. We also design a new model with many advanced modules (i.e., GATv2, GN block) that can iteratively refine the conformations. Experimental results on both small-scale and large-scale GEOM-QM9 and GEOM-Drugs demonstrate the effectiveness of our method.
For future work, first, we will incorporate chemical rules into deep learning models to improve generation quality. Second, current methods are mainly non-autoregressive, where all coordinates are generated simultaneously. We will study the autoregressive setting so as to further improve the accuracy. Third, we will deeply collaborate with chemists and biologists on more case studies.

References

Protein data bank: the single global archive for 3d macromolecular structure data. *Nucleic Acids Research*, 47(D1): D520–D528, 2019.

Axelrod, S. and Gomez-Bombarelli, R. Geom: Energy-annotated molecular conformations for property prediction and molecular generation, 2021.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Baseden, K. A. and Tye, J. W. Introduction to density functional theory: Calculations by hand on the helium atom. *Journal of Chemical Education*, 91(12):2116–2123, 2014. doi: 10.1021/ed5004788.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Chen, Q. and Koltun, V. Photographic image synthesis with cascaded refinement networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1529, 2017.

Dai, B. and Wipf, D. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019.

Dokmanic, I., Parhizkar, R., Ranieri, J., and Vetterli, M. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

Ganea, O.-E., Pattanaik, L., Coley, C. W., Barzilay, R., Jensen, K., Green, W., and Jaakkola, T. S. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=af_hng9tuNj.

Gebauer, N., Gasteiger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a4d8e2a7e0d0c102339f97716d2f6b6-Paper.pdf.

Gebauer, N., Gasteiger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a4d8e2a7e0d0c102339f97716d2f6b6-Paper.pdf.

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.

Hoffmann, M. and Noé, F. Generating valid euclidean distance matrices. *arXiv preprint arXiv:1910.03131*, 2019.

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Kanal, I. Y., Keith, J. A., and Hutchison, G. R. A sobering assessment of small-molecule force field methods for low energy conformer predictions. *International Journal of Quantum Chemistry*, 118(8): e25512, 2018. doi: https://doi.org/10.1002/qua.25512. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.25512.
Karney, C. F. Quaternions in molecular modeling. *Journal of Molecular Graphics and Modelling*, 25(5):595–604, 2007.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR 2014)*, Banff, Canada, 2014. ICLR Committee.

Koes, D. R., Baumgartner, M. P., and Camacho, C. J. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *J Chem Inf Model*, 53(8):1893–904, 2013. ISSN 1549-960X (Electronic) 1549-9596 (Linking). doi: 10.1021/ci300604z.

Kontoyianni, M. *Docking and Virtual Screening in Drug Discovery*, pp. 255–266. Springer New York, New York, NY, 2017. ISBN 978-1-4939-7201-2. doi: 10.1007/978-1-4939-7201-2_18. URL https://doi.org/10.1007/978-1-4939-7201-2_18.

Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50(2):302–309, 2017.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL http://arxiv.org/abs/1608.03983.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lowe, J. P. and Peterson, K. *Quantum chemistry*. Elsevier, 2011.

Luo, S., Guan, J., Ma, J., and Peng, J. A 3d generative model for structure-based drug design. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021a.

Luo, S., Shi, C., Xu, M., and Tang, J. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 34, 2021b.

Mansimov, E., Mahmood, O., Kang, S., and Cho, K. Molecular geometry prediction using a deep generative graph neural network. *Scientific Reports*, 9(1): 20381, Dec 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-56773-5. URL https://doi.org/10.1038/s41598-019-56773-5.

Meli, R. and Biggin, P. C. spyrmsd: symmetry-corrected rmsd calculations in python. *Journal of Cheminformatics*, 12(1):49, Aug 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-00455-2. URL https://doi.org/10.1186/s13321-020-00455-2.

O’Boyle, N. M., Vandermeersch, T., Flynn, C. J., Maguire, A. R., and Hutchison, G. R. Confab - systematic generation of diverse low-energy conformers. *Journal of Cheminformatics*, 3(1):8, Mar 2011. ISSN 1758-2946. doi: 10.1186/1758-2946-3-8. URL https://doi.org/10.1186/1758-2946-3-8.

Parr, R. G. Density functional theory of atoms and molecules. In Fukui, K. and Pullman, B. (eds.), *Horizons of Quantum Chemistry*, pp. 5–15, Dordrecht, 1980. Springer Netherlands. ISBN 978-94-009-9027-2.

Rapport, A. K., Casewit, C. J., Colwell, K. S., Goddard, W. A., and Skiff, W. M. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society*, 114(25):10024–10035, 1992. doi: 10.1021/ja00051a040.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Roy, K., Kar, S., and Das, R. N. Chapter 10 - other related techniques. In Roy, K., Kar, S., and Das, R. N. (eds.), *Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment*, pp. 357–425. Academic Press, Boston, 2015. ISBN 978-0-12-801505-6. doi: https://doi.org/10.1016/B978-0-12-801505-6.00010-7. URL https://www.sciencedirect.com/science/article/pii/B9780128015056000107.

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1esMkHYPr.

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

Simm, G. N. C. and Hernández-Lobato, J. M. A generative model for molecular distance geometry. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8949–8958. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/simm20a.html.

Smith, D. G., Burns, L. A., Simmonnett, A. C., Parrish, R. M., Schieber, M. C., Galvelis, R., Krause, P., Kruse,
H., Di Remigio, R., Alenaizan, A., et al. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *The Journal of chemical physics*, 152(18):184108, 2020.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, volume 28, pp. 3483–3491, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Winter, R., Noé, F., and Clevert, D.-A. Auto-encoding molecular conformations. *arXiv preprint arXiv:2101.01618*, 2021.

Xia, Y., Tian, F., Wu, L., Lin, J., Qin, T., Yu, N., and Liu, T.-Y. Deliberation networks: Sequence generation beyond one-pass decoding. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/c6036a69be21cb660499b75718a3ef24-Paper.pdf.

Xu, M., Luo, S., Bengio, Y., Peng, J., and Tang, J. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=pAbmlqfheGk.

Xu, M., Wang, W., Luo, S., Shi, C., Bengio, Y., Gomez-Bombarelli, R., and Tang, J. An end-to-end framework for molecular conformation generation via bilevel programming. *arXiv preprint arXiv:2105.07246*, 2021b.
A. Details of $\varphi_{2D}$ and $\varphi_{3D}$

The model architectures of $\varphi_{2D}$ and $\varphi_{3D}$ are similar to $\varphi_{dec}$, with the following differences.

Comparing $\varphi_{2D}$ with $\varphi_{dec}$, the differences are the initial conformation $\hat{R}(0)$ and initial features (i.e., the $H_{V}(0)$, $H_{E}(0)$ and $U(0)$). $\varphi_{2D}$ takes a random conformation sampled from uniform distribution in $[-1, 1]$ as input. The initial atom and edge features are the embeddings of the atoms and edges respectively. $\varphi_{2D}$ will also output a prediction of the conformation. Note that the random variable $z$ sampled from Gaussian $\mathcal{N}(\mu_{R,G}, \Sigma_{R,G})$ is not used in $\varphi_{2D}$.

Comparing $\varphi_{3D}$ with $\varphi_{dec}$, the differences are the initial conformation $\hat{R}(0)$, initial features (i.e., the $H_{V}(0)$, $H_{E}(0)$ and $U(0)$) too. $\varphi_{3D}$ takes the groundtruth conformation as input. The initial atom and edge features are the embeddings of the atoms and edges respectively. Another difference is that the fourth step of $\varphi_{dec}$, i.e., Eqn.(7), is not used.

B. More Experiment Details

B.1. More Details About Training

We use AdamW optimizer (Loshchilov & Hutter, 2019) with initial learning rate $\eta_0 = 2 \times 10^{-4}$ and weight decay 0.01. In the first 4000 iterations, the learning rate is linearly increased from $10^{-6}$ to $2 \times 10^{-4}$. After that, we use cosine learning rate scheduler (Loshchilov & Hutter, 2016), where the learning rate at the $t$-th iteration is $\eta_0 (1 + \cos(\pi t / T)) / 2$, where $T$ is the half of the period (i.e., the iteration numbers of 10 epochs in our setting). Similarly, we also use the cosine scheduler to dynamically set the $\beta$ at range $[0.0001, 0.008]$. The batch size is fixed as 128. All models are trained for 100 epochs. For the two small-scale settings, the experiments are conducted on a single V100 GPU. For the two large-scale settings, we use two V100 GPUs for experiments. The detailed hyper-parameters are described in Table 6.

|                      | Small-Scale | Large-Scale |
|----------------------|-------------|-------------|
| Layer number         | 3           | 6           |
| Dropout              | 0.1         | 0.1         |
| Learning rate        | 2e-4        | 2e-4        |
| Batch size           | 128         | 128         |
| Epoch                | 100         | 100         |
| Vae $\beta$ Min     | 0.0001      | 0.001       |
| Vae $\beta$ Max     | {0.001, 0.002, 0.004, 0.008, 0.01} | {0.005, 0.01, 0.02, 0.04, 0.05} |
| Latent size          | 256         | 256         |
| Hidden dimension     | 512         | 1024        |
| GPU number           | 1 × NVIDIA V100 | 2 × NVIDIA V100 |

B.2. More Details About Molecular Docking

For RDKit, we generated one initial conformation as input and set $num\_modes$ to 50 when performing docking. For our method and ConfGF, since the generated conformations are independent and diverse, we randomly selected five of them, performed five independent molecular docking calculations and set $num\_modes$ to 10 to ensure all three methods generate equal number of conformations. Eventually, each method got about 50 binding conformations. The conformation corresponding to the lowest binding affinity was selected as the final docked pose.

B.3. Why We Set $\gamma = 1$ in Eqn.(4)

Under the VAE formulation, Mansimov et al. (2019) also leaned the variance parameter $\gamma$ of the likelihood model. i.e., the $\gamma$ in Eqn.(4). However, a decent VAE analysis (Dai & Wipf, 2019, Thm. 3) revealed that the VAE objective prefers a vanishing $\gamma$. This is desired in the ideal case when the encoder and decoder are optimized perfectly (Dai & Wipf, 2019, Thm. 4), but in practice it would distract the optimizer towards a “lazy way” to focus on $\gamma$ and stagnate the optimization of the encoder and

---

4When using different random seeds, the conformations output by RDKit is not diverse enough. Therefore, we only choose one here.
We verify the following three loss functions:

1. Comparing \( L_1 \) with our method, we can see that when we do not use the permutation invariant loss on symmetric atoms, we can see that using permutation invariant loss on symmetric atoms are important, without which the results significantly drop. (2) Comparing \( L_2 \) with our method, we can see that when we do not use the permutation invariant loss, using more constraints on bond lengths and bond angles can help improve the performances. (3) When using both permutation invariant loss and roto-translation invariant loss, using \( \ell_{bond} \) and \( \ell_{angle} \) will not bring more significant improvement. These results demonstrate that for molecular conformation generation, it is important to consider the permutation of symmetric atoms.

C. More Experimental Results

C.1. Combination with Distance-based and Angle-based Loss Functions

In addition to the matching loss defined in Eqn.(1) which is related to coordinates only, one may be curious about whether using distance-based loss and angle-based can further improve the performance, since the latter two are equivariant to the transformation of coordinates. For ease of reference, let \( R_i \) denote the groundtruth coordinate of atom \( v_i \) and \( \hat{R}_i \) denote the predicted coordinate of atom \( v_i \). Recall in Section 1, we use \( E \) to denote the collection of all bonds. We define \( E_2 \) as \( \{ (i, j, k) | (i, j) \in E, (i, k) \in E, k \neq j \} \).

Inspired by (Winter et al., 2021) and (Ganea et al., 2021), we use the following two functions:

\[
\ell_{angle} = \frac{1}{|E_2|} \sum_{(i, j, k) \in E_2} \| \cosine(\hat{R}_j - \hat{R}_i, R_k - R_i) - \cosine(\hat{R}_j - \hat{R}_i, \hat{R}_k - \hat{R}_i) \|_F, \tag{8}
\]

\[
\ell_{bond} = \frac{1}{|E|} \sum_{(i, j) \in E} \left( \text{distance}(R_j, R_i) - \text{distance}(\hat{R}_j, \hat{R}_i) \right)^2, \tag{9}
\]

where \( \cosine(a, b) = \frac{a^\top b}{\|a\|\|b\|} \) and \( \text{distance}(a, b) = \|a - b\| \), \( a \) and \( b \) are two vectors. That is, we apply additional constraints to bond length and bond angles. Please note that with the above two auxiliary loss functions, our method still generates coordinates directly and does not need to generate intermediate distances and angles.

We verify the following three loss functions:

\[
L_1 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \ell_{RT}(R, \hat{R}(\mu_{R,G} + \Sigma_{R,G} \epsilon, G)) + \beta D_{KL}(\mathcal{N}(\mu_{R,G}, \Sigma_{R,G})|\mathcal{N}(0, I)), \tag{10}
\]

\[
L_2 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \ell_{RT}(R, \hat{R}(\mu_{R,G} + \Sigma_{R,G} \epsilon, G)) + \beta D_{KL}(\mathcal{N}(\mu_{R,G}, \Sigma_{R,G})|\mathcal{N}(0, I)) + \lambda (\ell_{angle} + \ell_{bond}), \tag{11}
\]

\[
L_3 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \ell_{RTP}(R, \hat{R}(\mu_{R,G} + \Sigma_{R,G} \epsilon, G)) + \beta D_{KL}(\mathcal{N}(\mu_{R,G}, \Sigma_{R,G})|\mathcal{N}(0, I)) + \lambda (\ell_{angle} + \ell_{bond}), \tag{12}
\]

where \( \lambda = 0.1 \). Note in Eqn.(10) and Eqn.(11), we use the roto-translation loss only without considering permutation invariant loss on symmetric atoms. We conduct experiments on GEOM-Drugs (small-scale setting). The results are reported in Table 7.

| Methods | COV(%)↑ | MAT (Å)↓ |
|---------|---------|----------|
|         | Mean    | Median   | Mean    | Median   |
| Ours    | 96.69   | 100.00   | 0.7223  | 0.7236   |
| \( L_1 \) | 77.78   | 86.09    | 1.0657  | 1.0563   |
| \( L_2 \) | 92.45   | 98.70    | 0.8983  | 0.9016   |
| \( L_3 \) | 96.01   | 100.00   | 0.7235  | 0.7199   |

We have the following observations:

1. When using both permutation invariant loss and roto-translation invariant loss, using \( \ell_{bond} \) and \( \ell_{angle} \) will not bring more significant improvement. These results demonstrate that for molecular conformation generation, it is important to consider the permutation of symmetric atoms.
C.2. Study of Model Parameters

In this section, we compare the performances of our method and ConfGF. By default, our model has 13.29M parameters, and the ConfGF model has 0.81M parameters. We increase the model size of ConfGF to 12.28M by increasing hidden dimension sizes, and re-run the experiments. The results are shown in Table 8.

| Dataset      | QM9         | Drugs       |
|--------------|-------------|-------------|
| ConfGF (0.81M) | 88.49 | 49.13 | 0.2673 | 0.2685 | 62.15 | 70.93 | 1.1629 | 1.1596 |
| ConfGF (12.28M) | 86.86 | 93.49 | 0.3377 | 0.3450 | 55.36 | 58.20 | 1.2186 | 1.2134 |
| Ours (13.29M) | 96.34 | 99.53 | 0.2065 | 0.2003 | 96.69 | 100.00 | 0.7223 | 0.7236 |

We can see that when we increase the model size of ConfGF, the performance becomes worse, which shows that ConfGF cannot benefit from more parameters. We observe that a larger ConfGF (12.28M) suffers from larger training loss than a smaller ConfGF (0.98M), which shows its limitation.

C.3. Visualization of Molecules

In Figure 6, we visualize the conformation of different methods. We randomly select three molecules from the small-scale GEOM-drug dataset, generate several conformations, and visualize the best-aligned ones with the groundtruth. We can see that our method can generate high-quality conformations than previous methods, which are the most similar to the groundtruth.
C.4. The Impact of the Permutation Invariant Loss

To illustrate the impact of the permutation invariant loss, we show two examples in Figure 7. For these two examples, there exists a rotatable ring at the end of a molecule, where the ring is symmetric to the bond connecting itself to the rest of the molecule. Without the permutation invariant loss (see the row No $\ell_P$), our method fails to generate the coordinates of such rings, but simply puts them in a line. This is because the model is trapped into local optimal. By using the permutation invariant loss, we can successfully recover the conformations of those rings (see the row “Ours”). This shows the importance of using the permutation invariant loss $\ell_P$ as we proposed.

| Molecular Graph | Reference | No $\ell_P$ | Ours |
|-----------------|-----------|-------------|------|
| ![Molecular Graph](image1) | ![Reference](image2) | ![No $\ell_P$](image3) | ![Ours](image4) |

Figure 7: The illustration of the impact of the permutation invariant loss. “No $\ell_P$” means without the permutation invariant loss.

C.5. More Discussions on the Conformation with More Heavy Atoms

In Table 1 and Table 2, we observe that our method works better than distance-based methods (include modeling the distances directly, or the gradients of distances) on molecules with more heavy atoms. Our conjecture is that for these distance-based works, they usually extend the molecular graph with 1,2,3-order neighbors, which is sufficient to determine the 3D structure in principle. For GEOM-QM9 dataset, considering the number of atoms is less than 10, this extended graph is nearly a complete graph and can provide enough signals to reconstruct the 3D structure. Therefore, these distance-based performances are good on GEOM-QM9 dataset. For GEOM-Drugs dataset, the numbers of atoms are much more than those in GEOM-QM9. Although in theory, the distances in a third-order extended graph can reconstruct the 3D structure, practically the signals are still not enough. Our method does not rely on the interatomic distances, and can achieve good results on large molecules.

To verify our conjecture, on GEOM-Drugs, we categorize the molecules based on their numbers of heavy atoms. The number of heavy atoms in the $i$-th group lie in $[10i + 1, 10(i + 1)]$. We compare our method against ConfGF (the code of DGSM is not available) and GraphDG. The results are in Table 9. We have similar observation, that our method brings more improvements than previous method on larger molecules.

C.6. More Results about Property Prediction

The median absolute error of the property prediction is shown in Table 10. We can see that our method still outperforms all deep learning based methods, which demonstrate the effectiveness of our method.

D. More Discussion on Distance-based Methods

Let $G$ be a molecular graph with $N$ atoms ($N \geq 3$). Let $d_{ij}$ denote the distance between atom $i$ and atom $j$. Define $D$ as the distance matrix, which is an $N \times N$ matrix, and $d_{ij}$ locates in the $i$-th row and $j$-th column of $D$. 
Direct Molecular Conformation Generation

| Metric  | COV(%)† | i = 1 | i = 2 | i = 3 | average |
|---------|---------|-------|-------|-------|---------|
| ConfGF  | 99.95   | 66.28 | 15.34 | 62.54 |
| GraphDG| 15.11   | 1.78  | 0.0   | 3.12  |
| Ours    | 100.00  | 97.62 | 90.04 | 96.69 |

| Metric  | MAT(Å)↓ | i = 1 | i = 2 | i = 3 | average |
|---------|---------|-------|-------|-------|---------|
| ConfGF  | 0.7764  | 1.1510| 1.5345| 1.1637|
| GraphDG| 2.0578  | 2.5863| 2.9849| 2.5847|
| Ours    | 0.5305  | 0.7190| 0.8794| 0.7223|

Table 9: COV and MAT mean scores w.r.t numbers of heavy atoms on small-scale GEOM-Drugs. The i indicates that the heavy atom number lies in range \([10i + 1, 10(i + 1)], i \in \{1, 2, 3\}\).

Table 10: Median absolute error of predicted ensemble properties. (Unit: eV).

| Methods | \(E\) | \(E_{\text{min}}\) | \(\Delta \epsilon\) | \(\Delta \epsilon_{\text{min}}\) | \(\Delta \epsilon_{\text{max}}\) |
|---------|-------|-----------------|-----------------|-----------------|-----------------|
| RDKit   | 0.8721| 0.6119          | 0.3057          | 0.4414          | 0.1830          |
| GraphDG| 13.1707| 1.9221         | 3.4136          | 7.6845          | 1.1663          |
| ConfGF  | 1.5167| 0.1972          | 0.6588          | 4.8920          | 0.1686          |
| Ours    | 0.4132| 0.1100          | 0.1276          | 0.8486          | 0.1288          |

A valid distance matrix \(D\) is induced from the \(3N - 6\) degree-of-freedom (DOF) of \(N\) 3D-coordinates excluding global translation and rotation, while the popular practice of independently generating distances to 2- or 3-hop neighbors (Xu et al., 2021a) often introduces more DOF.

Moreover, a distance matrix should have a rank at most 5 after element-wise squared (Dokmanic et al., 2015). In other words, the rank of matrix \(\tilde{D} = \{d_{ij}^2\}_{i,j}\) is at most 5. Such a constraint is hard to guarantee even if the DOF is matched (Simm & Hernández-Lobato, 2020) (e.g., \(\lambda I\) has one DOF but is almost surely full-rank). It also makes gradients ill-defined (Shi et al., 2021) (other distances cannot all be held constant while taking an infinitesimal change to \(d_{ij}\)). Careful treatments (Hoffmann & Noé, 2019) often increase the order of computation complexity.