A HIERARCHICAL DIFFUSION-BASED MODEL FOR 3D DRUG-LIKE MOLECULE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating desirable molecular structures is a fundamental problem for drug discovery, and recently there is a growing interest in developing deep learning models to accelerate this process. Despite the considerable progress we have achieved, existing methods usually generate atoms auto-regressively and ignore intrinsic local structures such as rings, which is however ubiquitous in drug-like biomolecules, making the generated structures still far away from satisfactory. In this paper, we propose a hierarchical diffusion based generative model for drug-like molecule generation in 3D space, effectively preserving the validity of local segments. Our method first generates coarse-grained geometries via an equivariant diffusion process, where nodes represent possible fragments. Then, the coarse nodes are decoded as fine-grained atoms to assemble atomic molecular structures. However, such a process is non-trivial since generated neighborhood fragments may suffer from atom-bonds conflicts, preventing them to be connected. We view this problem as a constrained generative modeling task and propose a novel Monte Carlo sampling approach which iteratively refines fragments to achieve effective valid molecules sampling. Finally, extensive experiments demonstrate that the proposed method could consistently improve the quality of molecule generation over existing methods, especially for larger drug-like biomolecules.

1 INTRODUCTION

In recent years, we have witnessed huge success in various applications using deep learning for computational biology and chemistry problems (Jumper et al., 2021; Gawehn et al., 2016; Kell et al., 2020). Among them, deep generative models have specifically shown great promise in modeling complex graph-like structures in the field of life science, *e.g.*, biomolecules and proteins. For example, AlphaFold II (Jumper et al., 2021) has demonstrated the effectiveness of data-driven methods for predicting highly complex conformations. Besides, with the rapid growth of graph representation learning (Wu et al., 2020), great progress has also been achieved for molecular modeling, ranging from generating molecular atom-bond graphs (Li et al., 2018b; Liu et al., 2018; Jin et al., 2018a) to generating molecular conformations from graphs (Xu et al., 2022).

Despite the significant progress achieved, a remaining but vital research direction in this track is de novo design of drug molecules in 3D space. The molecular conformations in 3D physical space can directly determine many functional properties of compounds, *e.g.*, energy, pocket binding affinity. Therefore, integrating the 3D conformation into the molecule design process enjoys several advantages over only involving topological information in many important applications, *e.g.*, structure-based drug design (Peng et al., 2022; Luo et al., 2021), molecular dynamic simulation (Hansson et al., 2002), 3D similarity searching (Shin et al., 2015). Thus, a more natural and meaningful scenario is to directly generate molecules in 3D, modeling the joint distribution of molecular category, topological structure, and the conformations through a single principled generative model (Gebauer et al., 2019; Satorras et al., 2022; Hoogeboom et al., 2022).

However, how to generate desirable molecules in 3D space, especially large biomolecules, remains an unsolved problem. Current existing approaches (Gebauer et al., 2019; Satorras et al., 2022; Hoogeboom et al., 2022) typically generate the 3D molecules in the atomic resolution with an auto-regressive approach, which may result in serious problems for generating drug-like molecules. Firstly, the atom-level generation manner, though enjoys higher flexibility to place each atom, lacks

necessary constraints to obtain reliable molecule structures. For example, without imposing euclidean geometric constraints, the generated 3D aromatic rings could seriously violate basic chemical rules. Without strong dependency constraint on the atom generation, the obtained molecule may suffer from compatibility problem between fragments, that is, two fragments is unable to connect to each other. Secondly, the auto-regressive approach suffers from error accumulation problem. Therefore, existing auto-regressive based 3D molecule generation models are limited in generating large molecules, which prevents these models from wide applications in real scenarios.

In this paper, different from previous work that generates molecules at atom-level, we instead propose a novel hierarchical diffusion-based model generating 3D molecules in a dual-phase fashion. Our model first generates molecules by coarsened structures instead of atomic graphs, where each coarse-grained node represents a certain type of fragment. Then in the second phase, the coarsened nodes are further decoded to corresponding atomic structures and all substructures are assembled as realistic molecules that preserve valid local structures. Such a two-stage generation process nicely mimic chemistry expert's drug design process as combining fragments from a pre-defined functional group database. In this way, important inductive biases in this area are encoded in our model, e.g. most important chemical properties are determined by the fragments with rigid local 3D structure. Furthermore, from machine learning's perspective, the fragment based representation of molecules significantly reduce unnecessary degrees of freedom in the atomic based methods, thus will lead to global optimum convergence and better generalization ability. Nevertheless, such a two-stage framework is indeed challenging, because generated fragments are required to be connected to a single reasonable molecular structure. In this work, we treat 3D molecule generation as a constraint generation problem, and provide both principled modeling and sampling to address this problem. Inspired by the concept of pharmacophore in drug design that fragments could be divided into different binding groups by their functions, we propose to generate the *fragment representation* instead of a single *deterministric fragment*. Specifically, we utilize two sets of chemically interpretable features to represent a fragment. Then we propose a geometric diffusion model to generate these fragment representations and their Cartesian coordinates in an efficient non-autoregressive manner. After that, we propose an iterative Monte Carlo sampling algorithm to preserve the chemical validity and drug-likeness of sampled structures. Specifically, we iteratively sample fragments from the fragment representation and continue this refinement procedure until a valid geometry is sampled.

We conducted extensive experiments on several 3D molecular generation benchmarks to compare our proposed model with the competitive baseline models. Specifically, instead of small toy molecules from QM9, we trained our model on $\text{GEOM}_{\text{DRUG}}$ (Axelrod & Gomez-Bombarelli, 2022) dataset which contains more realistic drug-like molecules, and evaluated two sets of well-designed and comprehensive metrics. Quantitative results show that our method can always generate molecules with better drug-like chemical properties and less unstable or unrealistic substructures. Compared to the baseline model, our method can generate conformations that are much closer to those generated with expensive cheminformatics software. Visualized results also demonstrate our model is capable of generating molecules of higher quality both at 2D graph level and 3D conformation level. All experimental results suggest that our model enjoys a much higher capacity to sample 3D molecules from the drug chemical space.

2 Related work

In earlier works, molecule generation tasks are tackled by generating sequential representations of molecules, smiles *e.g.* Kusner et al. (2017); Dai et al. (2018); Segler et al. (2018). Motivated by rapid development in the graph neural network, researchers then use graph generative model to generate novel molecules *e.g.* Jin et al. (2018b;a); Li et al. (2018a) However, both text and graph ignore the 3D position of the atoms which is important for molecule properties. Research has been addressing the task of generating 3D molecules these days. MolVAE (Ragoza et al., 2020) generate ligand-like molecules in a coarse-grained 3D grid using 3D-CNN. These studies did not guarantee rotation + translation equivariant, so they output conformations of low quality and can only sample from the embedding space of a seed molecule. Most works that take equivariance into account deal with small organic molecules from the QM9 dataset. G-schNet (Gebauer et al., 2019) applied SchNet (Schütt et al., 2017) to acquire equivariant latent space and generate molecules by sampling atom types and distances iteratively. With the rapid development of equivariant network

architectures, SphereNet (Liu et al., 2021) and EGNN (Satorras et al., 2021) are also adapted to generative models for molecule generation. G-SphereNet (Luo & Ji, 2022) used the flow-based model to formalize the generative procedure which defines the angle, torsion, and distance for coordinates computing. There are two generative models derived from EGNN. E(n)-flow (Satorras et al., 2022) and EDM (Hoogeboom et al., 2022) are both non-autoregressive models that generate atom types and atom coordinates in one forward pass. The only difference is that E(n)-flow adapted continues normalizing flow as a generative network while EDM applied a diffusion network. EDM was the only work that succeeds to generate drug-like molecules by training on the $\text{GEOM}_{\text{DRUG}}$ dataset (Axelrod & Gomez-Bombarelli, 2022) that usually have larger molecular weights than simple organic molecules from the QM9. However, EDM always generates unrealistic ring systems and broken molecules, which have proven to be unsuitable for generating drug-like molecules. In the literature of hierarchical graph generation, most previous works derive the hierarchical structure in graph based on some intrinsic rules: Some use the different granularity levels, *e.g.* atom-to-motif (Jin et al., 2020; 2019) or edge-to-node (Xianduo et al., 2022), as the modeling hierarchy; (Zhou et al., 2019; Chauhan et al., 2019) used predefined rules to distinguish different nodes into different levels; (Mi et al., 2021) utilized the natural topology in graph to get the hierarchy. (Kuznetsov & Polykovskiy, 2021) get the hierarchy by adding latent variables to different module layers of the model. While in our method, we use the learnable decoding module to approximate a semantic guided hierarchy, *i.e.*, a coarse feature could refer to a cluster of property-level/element-level related fragments. Such hierarchies are inspired by the "pharmacophore" to fragment process in traditional drug design.

3 PRELIMINARIES

3.1 NOTATIONS AND FRAGMENT-BASED 3D GENERATION

We study the problem of generating 3D geometries of molecules with fragments as the smallest unit. Let $\mathcal{G} = \{G_i\}_{i=1}^m$ be the empirical distribution of the 3D graphs. Here, each 3D graph G_i consists of the fragment set $\mathcal{V}_i$ and the edge set $\mathcal{E}_i$. More specifically, every fragment $V_i \in \mathcal{V}_i$ represents a combination of several atoms and bonds, *e.g.*, a benzene ring could be a fragment that includes six carbon atoms and aromatic bonds. And each edge $E_{mn} \in \mathcal{E}_i$ indicates that a bond/atom is shared by two fragments V_m and V_n. And we also use the variable $\mathcal{E}$ to scoop the attachment information, *e.g.* the area of the molecule surface intersects with the neighbor fragments. The fragment-based 3D generation model aims to learn a probabilistic model $P_\theta(\cdot)$ to fit the empirical data set, which could also be easily sampled from, the sampled fragments are further integrated into the 3D molecule.

3.2 EQUIVARIANCE AND SE(3)-INVARIANT DENSITY ESTIMATION

Equivariance widely exists in the physical world, especially in the atomic systems. For example, the vector fields of atomic forces should rotate or translate correspondingly with the 3D positions of the molecule. Thus integrating such inductive bias into the function modeling has appealing properties and has been widely explored(Wu et al., 2018; Schütt et al., 2017; Satorras et al., 2021). More specifically, given two transformation T_g and S_g acting on the space $\mathcal{X}$ and $\mathcal{Y}$, a function f is considered as equivariant with respect to the group G if the following is satisfied:

$$f \circ T_g(x) = S_g \circ f(x) \tag{1}$$

In this task, we mainly focus on the SE(3) group, *i.e.*, the group of rotation and translation in the 3D space. For generative modeling of 3D molecule graph, the density function of the model distribution $P_\theta(.)$ should be SE(3)-invariant, *i.e.*, $P_\theta(x) = P_\theta(T_g(x))$. This is, the likelihood of a specific conformation of some determined molecule graph should not be influenced by the transformations such as the rotation and translation. To this end, previous methods either directly model the invariant components, *e.g.*, bond length, or use some invariant base distribution and model the transformation by the equivariant neural network (Hoogeboom et al., 2022; Satorras et al., 2022).

3.3 DENOISING DIFFUSION PROBABILISTIC MODEL

Denoising diffusion probabilistic model (DDPM) Yang et al. (2022b); Sohl-Dickstein et al. (2015)provides a powerful generative modeling tool by reversing a diffusion process. More specifically, the *diffusion* process project the noise into the ground truth data and the *generative* process

learn to reverse such process. The two processes imply a latent variable model, where $\mathbf{x}_1, \cdots, \mathbf{x}_{t-1}$ are the latent variables. The *forward* process could be seen as a fixed approximate posterior distribution:

$$q_(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \tag{2}$$

Here $\beta_1, \cdots, \beta_T$ corresponds to a fixed variance schedule. For simplicity, we let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, the forward pass for arbitrary time step has an analytic form, *i.e.*, $q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right)$. The *generative* process parameterized the transition kernel $P_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ of the Markov chains, the corresponding likelihood function could be derived as:

$$P_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_t^2 \mathbf{I}\right) \quad P_\theta(\mathbf{x}_0) = \int p\left(\mathbf{x}_T\right) P_\theta\left(\mathbf{x}_{0:T-1} \mid \mathbf{x}_T\right) \mathrm{d}\mathbf{x}_{1:T} \tag{3}$$

Here the μ_θ refers to parameterized means function and the σ_t^2 is the predefined variance. For the initial distribution $P_\theta(\mathbf{x}_T)$, one natural selection could be the standard Gaussian distribution.

4 METHODS

In this section, we will first emphasize several challenges of building fragment based molecule generative model and introduce the overall probabilistic framework. Then we discuss our generative model for the high-level features, *i.e.*, the Coarse Set diffusion model, in Sec. 4.2. After that, we will show how to decode the specific fragment type and ensemble them together to get both the molecule graphs and the 3D conformations. The whole framework could be found in the Fig. 1.

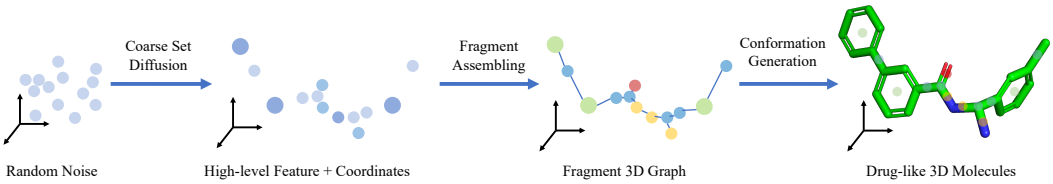


Figure 1: An overview of the hierarchical diffusion based generation model.

4.1 TOWARDS THE CHALLENGES OF FRAGMENT BASED GENERATION

There are generally two generative modeling fashions for molecule generation, *i.e.*, autoregressive and non-autoregressive. Modeling in the autoregressive fashion could in some sense reduce the conflict among the fragments during sampling. However, in important application scenarios, *e.g.*, pocket-guided generation, auto-regressive models will focus more on the local information due to the intrinsic property and hence lack the ability to be consistent with the full context space. Non-autoregressive generation is promising for its natural advantages of globally modeling (De Cao & Kipf, 2018; Kwon et al., 2019; Satorras et al., 2022; Hoogeboom et al., 2022) and which has also been demonstrated in the atom-level generation with superior performance. Though there are several appealing properties, non-autoregressive model at the fragment-level indeed implies the following structure generation procedure under hard constraint.

$$(V, E) \sim P_\theta(\cdot), s.t. \ \forall_{0 \le i,j < n, i \ne j, \mathbb{I}(E_{ij}=1)} \ (V_i, E_{ij}, V_j) \in \mathcal{W} \tag{4}$$

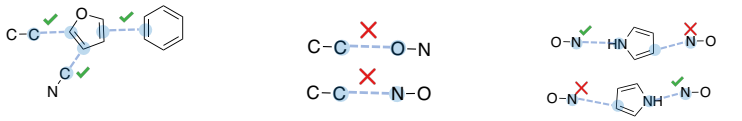
Here $\mathcal{W}$ stands for the valid bigrams of the consistent neighborhoods. For example, furan can assemble with benzene by merging a double carbon-carbon bond to form a valid double-ring system. However, the pyrrole cannot be assembled with two neighbors that need to be attached to a single nitrogen atom within the fragment, as illustrated in Figure 2. Note that the problem of avoiding fragment conflict has high complexity and brings the so-called "combinatorial exploding" issues. For non-autoregressive modeling fashion, the complexity increases exponentially with the structure size.

One direct solution to get valid molecules from non-autoregressive generative model is to conduct rejection sampling, *i.e.*, only accept the connectable molecules. Nevertheless, rejection sampling is not applicable in practice due to the extremely low acceptance rate. It is possible to relax such hard constraint through a learnable module, *e.g.*, $P_\phi(\cdot|V_i, V_j, E_{ij})$, which could result in another generative as $P_{\theta,\phi}(V, E) = P_\theta(V, E) \prod_{0 \le i,j < n, i \ne j} P_\phi(1|V_i, V_j, E_{ij})$. Unfortunately, the Gibbs

sampling procedure for such model still suffers from efficiency issues. To preserve the efficiency, we consider modeling the constraint in a reverse direction, *e.g.*, $P_\phi(V|\cdot)$. To integrate the domain prior and reduce the model complexity, we design the variable coarse set (H) as the latent variable and which could be expressed $P_{\theta,\phi}(V,E) = P_\theta(H)P_\phi(V,E|H)$. And such model is trained through maximum likelihood with the designing rule as a fixed approximate posterior ($Q(H|V,E)$), and we leave its detailed implementation in the next section:

$$\mathbb{E}_{(V,E)\sim P_{\text{data}}} \log \sum_{H\in\mathbb{H}} P_\theta(H)P_\phi(V,E|H) \geq \tag{5}$$
$$\mathbb{E}_{(V,E)\sim P_{\text{data}}} \mathbb{E}_{H\sim Q(H|V,E)}[\underbrace{\log P_\theta(H)}_{\text{Coarse Set Diffusion}} + \underbrace{\log P_\phi(V,E|H)}_{\text{Fragment Assembling}} - \underbrace{\log Q(H|V,E)}_{\text{Constant Term}}]$$

$\mathbb{H}$ stands for the possible support of H and above inequality holds due to the concavity of logrithm. The Coarse Set is inspired by the important concept of "pharmacophore" in drug design, which represents the category of fragments' functionality. And the whole process is like first determining the position and category of each component, then finding the connectable fragments from small subsets, and assembling them. Such framework could maintain the global modeling property of non-autoregressive methods and also significantly reduce the complexity of finding the connectable fragments.



(a) Conflict Free (b) One-hop Conflict (c) Two-hop Conflict

Figure 2: An illustration of fragment conflicts. One-hop conflict means the two linked fragments do not share any elements to form a valid edge. Two-hop conflict represents that though linked fragments can form edges by sharing the same atom/bond, conflicts occur when the valence is violated

4.2 Coarse Set Diffusion Model

Motivated by the pharmacophore concept, we introduce the "Coarse Set" vector as a numerical representation by quantizing some intrinsic properties. On one hand, We should carefully design the Coarse Set feature to be discriminative enough for fragments and molecules with different chemical and geometrical properties. On the other hand, this also allows us to easily integrate our domain knowledge as inductive bias into the model. We show two different feature designs:

Property-based Feature: We summarize the important properties which are the most widely used in drug discovery into an 8-dimension vector. Such property includes the number of hydrogen bonds and rings, and the area of different surfaces etc.

Element-based Feature: We also propose a simplified version by directly using the histogram of element frequency, *i.e.*, a 3-dimension vector, as the Coarse Set feature. This is inspired by the fact that elements with the same number of valence electrons share the same properties.

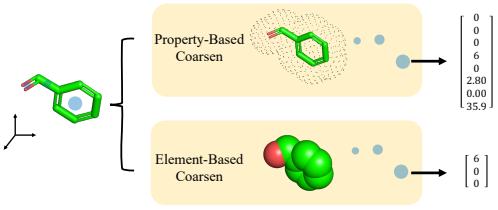


Figure 3: Illustration of converting a 3D Benzaldehyde into the 3D coarse set using two different sets of features

The explanation and detailed implementations is provide in Fig. 3, Table 9 and Table 10. The Coarse Set Diffusion (CS-Diff) model aims to model both the feature and position distribution based on the diffusion probabilistic framework as introduced in Sec. 3.3. Formally, we denote the whole variable as $H = [H_f, H_p]$, H_f stands for the variable of designed feature and H_p stands for the position related variable. There are several possible ways to represent the 3D conformation systems in fragment level, *e.g.*, the dihedral angle between neighbor fragments, and distance matrix. Surprisingly, we find that using the center position, *i.e.*, the average of all the atoms in a fragment, could be sufficient to determine conformations

as shown in Sec. 5.2. Note the center position of fragments could be seen as the center of the conformations sphere which includes all possible conformations generated from the degree of freedom on rotation. The connected fragments correspond to the tangent condition which actually eliminates a such degree of freedom. Formally, for each fragment V, $Q(H|V,E)$ in Eq. 5 is implemented by property-based/element-based feature extraction to get H_f and average all the atom coordinates to get H_p. Specifically, the property-based H_f could not only be determined by fragment type V also depends on the attachment E, *e.g.* the connection to neighbor fragments.

The modeling process of H_f could mostly follow the diffusion model with Gaussian noise for step $t > 0$. While we find that the 0-th term for continuous feature H_f^{int} and H_f^{cont}, *i.e.* $\mathcal{L}_0$, should be coped with carefully as following to get better empirical performance:

$$\mathcal{L}_0(H_f^{\text{int}}, H_f^{\text{cont}}) = -\log[\int_{H_f^{\text{int}}-\frac{1}{2}}^{H_f^{\text{int}}+\frac{1}{2}} \mathcal{N}\left(\boldsymbol{u} \mid \boldsymbol{x}_0^{(H_f^{\text{int}})}, \sigma_0\right) \mathrm{d}\boldsymbol{u}] - \log \mathcal{N}\left(H_f^{\text{cont}} \mid \frac{\boldsymbol{x}_0^{(H_f^{\text{cont}})}}{\alpha_0} - \frac{\sigma_0}{\alpha_0}\hat{\boldsymbol{\epsilon}}_0, \frac{\sigma_0^2}{\alpha_0^2}\mathbf{I}\right) \tag{6}$$

Next, we describe the generation for H_p. To make the likelihood function in Eq. 3 to be SE(3)-invariant, we set the initial distribution under zero center of mass (CoM) systems (Köhler et al., 2020), *i.e.*, applying a CoM-free Gaussian:

$$\mathcal{N}\left(H_p \mid \mathbf{0}, \sigma^2\mathbf{I}\right) = (\sqrt{2\pi}\sigma)^{-(M-1)\cdot n} \exp\left(-\frac{1}{2\sigma^2}\|H_p\|^2\right) \tag{7}$$

Here $H_p \in \mathbb{R}^{M\times n}$, where M is the number of fragment nodes and n equals the coordinate dimension. Besides, an equivariant Markov transition kernel is constructed under the widely applied noise parameterization (Ho et al., 2020):

$$\mu_\theta\left(H_p^t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(H_p^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(H_p^t, t\right)\right) \tag{8}$$

If could be demonstrated that if the ϵ_θ is parameterized by SE(3)-equivariant networks, then the transitional kernel $P_\theta(H_p^{t-1}|H_p^{t-1})$ is also SE(3)-equivariant, *i.e.*, $P_\theta\left(H_p^{t-1} \mid H_p^t\right) = P_\theta\left(T_g\left(H_p^{t-1}\right) \mid T_g\left(H_p^t\right)\right)$ (Xu et al., 2022). We leave the detailed discussion in Appendix A.2.

4.3 Fragment Assembling

In this section, we introduce the detailed process of generating fine-grained fragment types and determine the connection scheme to assemble them into valid drug-like 3D molecules. The process corresponding to the term $P_\phi(V,E|H)$ in Eq. 5. We briefly introduce the decoding logic here and leave the details in the following paragraphs. During decoding, the fragment node could be categorized into fine-grained or coarse-grained. Note that in the beginning, all the nodes are coarse-grained. For each decoding step, we first predict a focal node from the fine-grained nodes with a parameterized neural network module, ϕ_{focal}. And then we utilize a link prediction network, *i.e.* ϕ_{edge}, to identify a new linked node of the focal node among all the coarse-grained nodes. At last, we obtain the fine-grained fragment type of the above new linked coarse-grained node with the help of the other network, ϕ_{node}. The above procedure is illustrated in Figure 4. For all three steps, the prediction modules are conditioned on all coarse-grained nodes and fine-grained nodes and implemented with equivariant message passing. After decoding of a fine-grained node, an iterative refinement procedure is conducted to correct the bias with the help of a mask predict module ϕ_{refine}. Given all fine-grained nodes decoded, we obtain the attachment between linked fragments using local structures as the scoring function. The formal sampling algorithm is left in the Appendix A.7. And we emphasize several key elements of our assembling module in the following:

Message Passing for Decoding To avoid conflict as shown in Figure 2, we applied vanilla EGNN for ϕ_{focal} and three-step Tree EGNN for ϕ_{edge}, ϕ_{node}. ϕ_{edge} firstly aggregate information of all fine-grained nodes to the focal node by tree bottom-up pattern, in which the focal node is the root of the tree structure. After the new edge is predicted, the network broadcast the addition of new edge to all fine-grained nodes in a tree top-down pattern. Finally, ϕ_{node} aggregates the information from all fine-grained nodes in the bottom-up pattern to the new node for decoding the fine-grained fragment type. The message passing pattern could also be found in Figure 4 and a more detailed description is in Appendix A.3.

Iterative Refinement The iterative refinement process aims to correct the bias that lies in the decoded fine-grained nodes. To this end, we involve a mask prediction model ϕ_{refine} to approximate

the probability of each decoded fine-grained fragment conditioning on all coarse-grained nodes and the other fine-grained nodes. With such a mask predict module, we could adopt Monte Carlo sampling with node replacement as the transition to correct the error before the current state. The target distribution is defined as: $P = \prod_{n \in T} P_{\phi_{\text{refine}}}(\text{FRAG}(n) \mid G, T \setminus n)$. We denote T as a fine-grained subgraph, G as the coarse-grained full-size graph and FRAG() as a function that returns the fragment type for a specific node.

Training and Objective During training, we start by firstly randomly sampling a connected subgraph. Then a random leaf node is picked, and we simulate the generation of this node: We keep all the fine-grained fragment types and edges of the subgraph except the selected node. And for all nodes not in the subgraph, we only maintain the coarse features. ϕ_{focal} is trained based on the above feature to predict the parent of the selected node among the nodes in the subgraph; ϕ_{edge} are trained to predict the edge link between the parent node and the selected node among all other coarse-grained nodes. Note here we use the coarse feature of the selected node; And ϕ_{node} learns the fine-grained fragment type of the selected node. For iterative refinement part, we just randomly mask a node's fine-grained feature on random sampled subgraphs, and ϕ_{refine} is trained to reconstruct its masked fragment type. Detailed implementation and objective can be found in the Appendix A.7.

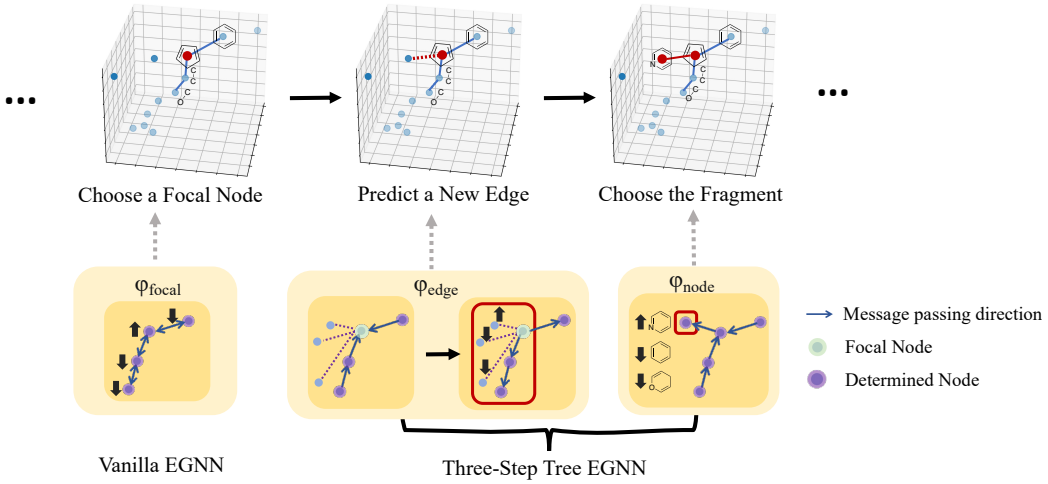


Figure 4: At each time step t, our model (1) chooses a focal node among coarse-grained nodes for edge formation (2) links the edge from all candidate new edges marked with dashed lines (3) generate the exact fragment type

Attachment Determination: Given all fine-grained features and linked relations determined, we need to decide which atoms within two linked fragments can be merged as one for attachment. To this end, we enumerate all possible attachments for linking two fragments and select the one that has the closest fragment center geometric to our generated fragment coordinate. We use RDkit to generate local conformation following Jing et al. (2022). And Root-mean-square deviation (RMSD) is applied to measure the difference between fragment centers. After this step, we obtain the complete molecule graph.

4.4 Atom Conformation Generation

Given the fine-grained fragment and the corresponding center positions, we discuss how we generate coordinates of atoms. Due to the fact that the set of stable local structures is quite limited and could be accurately generated by cheminformatics software, we follow previous works (Jing et al., 2022) to use RDKit ETKDG (Riniker & Landrum, 2015) to predict local conformation. And to plug the local conformations into each molecule coordinate system, the rotation matrix and translation vector are acquired. Here we use Kabsch Algorithm (Kabsch, 1976) to compute the rotation and translation factors based the software predicted fragment center and generated fragment center by previous assembling step. The detailed algorithm is introduced in Appendix A.4.

5 EXPERIMENTS

In this work, we mainly focus on the $\text{GEOM}_{\text{DRUG}}$ dataset (Axelrod & Gomez-Bombarelli, 2022), which includes 304k drug-like molecules with an average molecule weight of 360. We also tested our model on CrossDocked2020 (Francoeur et al., 2020), which 100k 3D ligand structures are extracted from complexs. We proposed two types of high-level feature and implement two types of CS-Diff model accordingly, $\text{CS-Diff}_{\text{E}}$ denotes the implementation that takes element-level feature as high-level feature input, while $\text{CS-Diff}_{\text{P}}$ is the implementation that takes property-based high-level feature input. We mainly compare our method with EDM (Hoogeboom et al., 2022) and G-SpherNet (Luo & Ji, 2022) on these 2 drug-like datasets. Though our model is not designed for generating smaller molecules, results of training our model on QM9 (Axelrod & Gomez-Bombarelli, 2022) are provided for sanity check in Table 4. We also provide the anonymous code link in Appendix. A.8.

5.1 DRUG-LIKENESS EVALUATION

Table 1: Properties of the generated molecules. Δ indicates that the evaluated metrics are computed as the difference between sampled molecules and the ground truth and the absolute values are listed in the (). '-r' is the notation that this model sample molecule without any iterative refinement.

	QED↑	RA↑	MCF↑	SAS↓	$\Delta_{\text{LogP}}\downarrow$ (logP)	$\Delta_{\text{MW}}\downarrow$ (MW)
G-SphereNet	0.382	–	0.489	–	2.306 (0.623)	170.7 (89.37)
EDM	0.608	0.441	0.621	4.054	0.566 (2.363)	23.71 (336.4)
$\text{CS-Diff}_{\text{E}}$ (-r)	0.628	0.626	0.681	3.669	0.638 (2.291)	27.33 (332.8)
$\text{CS-Diff}_{\text{P}}$ (-r)	0.635	0.638	0.656	**3.512**	0.185 (2.744)	**10.33 (349.8)**
$\text{CS-Diff}_{\text{E}}$	0.632	0.548	**0.727**	3.859	0.653 (2.276)	30.33 (329.8)
$\text{CS-Diff}_{\text{P}}$	**0.639**	**0.643**	0.659	3.547	**0.128 (2.801)**	13.33 (346.8)
$\text{GEOM}_{\text{DRUG}}$	0.658	0.915	0.774	4.018	0.000 (2.929)	0.000 (360.1)

The purpose of our proposed generation method is to fabricate molecules that are similar to authentic drug molecules from scratch, thus it is important to measure how drug-likely are those fabricated molecules to true drug molecules. In this section, we mainly measure the drug-likeliness of a molecule in 6 aspects. **Quantitative estimate of drug-likeness (QED)** was built on a series of carefully selected molecular properties to evaluate drug-likeness and is one of the most widely used metrics for virtual screening. **Retrosynthetic accessibility (RA)** is a machine learning-based scoring function that also evaluates synthetical accessibilities. It is more sensitive to unsynthesizable structures than SAS. **Medicinal chemistry filter (MCF)** is the rate of sampled molecules that do not contain any undruggable substructures (Brown et al., 2019). **Synthetic accessibility score (SAS)** is a ruled-based scoring function that evaluates the complexity of synthesizing a structure by organic reactions. **LogP** is the octanol-water partition coefficient which is the main factor that determines the distribution of the drug molecules. **Molecular weight (MW)**, the average molecular weight of generated molecules should be similar to ground truth MW statistics.

5.1.1 RESULTS AND DISCUSSION

Table 1 have shown that our proposed method performs significantly better than EDM (Hoogeboom et al., 2022) in every aspect of property measure when testing on $\text{GEOM}_{\text{DRUG}}$ dataset (Axelrod & Gomez-Bombarelli, 2022). The RA measure indicates that with all the sub-graphs derived from a predefined vocabulary, our model generates molecules that are easier to synthesize in wet labs, it is worth noting that dangerous substructures are also avoided. Δ_{MW} have shown how close the generated molecules are to ground truth MW, G-SphereNet is unable to generate large molecules that could exist in ground truth drug distribution. Especially, G-SphereNet occurs the early-stopping problem due to the error accumulation introduced by auto-regressive sampling, which is proven by our ablation study provided in the Table 7 from Appendix. Therefore, these baselines are incapable of learning the drug molecule distribution, where our results show that we are much closer to ground truth statistics of MW. Because of the extremely small molecule size and poor drug-likeness performance of the G-SphereNet model, we exclude G-SphereNet in followed conformation experiments. Similar results are found when we experiment on CrossDocked2020 (Francoeur et al., 2020), which is shown in Table 3.

Table 2: RMSD of the sampled conformations. 'atom' and 'frag' indicates the granularity (atom, fragment) of the coordinates we used for comparison. COV is the abbreviation for Coverage. MAT is the abbreviation for Matching.

	COV-atom↑	MAT-atom↓	COV-frag↑	MAT-frag↓
EDM	0.489	1.349	0.097	3.234
CS-Diff$_E$	**0.546**	**1.121**	0.153	2.583
CS-Diff$_P$	0.490	1.166	**0.202**	**2.431**
GEOM$_{\text{DRUG}}$	0.589	0.494	0.435	1.494

We also carried out an ablation study in which we removed the iterative refining step in the sampling process. According to the inadequate scores of QED and MCF, the model samples much simpler molecules when without iterative refining, the reason for this change is that without refining, our denoising model tends to choose fragments that are easier to assemble while the safety and drug-likeness are sacrificed.

5.2 CONFORMATION QUALITY EVALUATION

In addition to drug-like property evaluation, we also evaluate the conformation of our generated 3D molecules. RMSD is defined as the normalized Frobenius norm of the two conformations coordinates. However, all molecules we sampled are novel so there isn't any existing ground truth for us to evaluate. To obtain ground truth conformations, we applied the same experimental procedure as Axelrod & Gomez-Bombarelli (2022). This standard method has proven to be accurate enough for 3D conformation generation and utilized as ground truth in Xu et al. (2022); Jing et al. (2022). A computational costly molecular dynamic simulation is carried out for all molecular graphs. The detailed experimental procedure is described in Appendix A.5. Let $\mathcal{C}$ denote the set of MD simulated conformations and C denote the generated conformation. We define **Cov**erage metric and **Mat**ching metric as follows:

$$\text{COV}\left(C, \mathcal{C}^*\right) = \frac{1}{|\mathcal{C}^*|}\left\{\sum_{C^* \in \mathcal{C}^*} \mathbb{1}(\text{RMSD}(C, C^*) \leq \delta)\right\}, \tag{9}$$

$$\text{MAT}\left(C, \mathcal{C}^*\right) = \min_{C^* \in \mathcal{C}^*} \text{RMSD}(C, C^*), \tag{10}$$

where C denotes the conformation generated by our model, $\mathcal{C}^*$ represents the ground truth set of conformations sampled with MD simulation, C^* denotes a ground truth instance in $\mathcal{C}^*$, $\mathbb{1}(\cdot)$ is the indicator function which evaluates to 1 when the input is true otherwise 0, δ is the similarity threshold set to 2 Å in practice. The coverage metric describes the rate of ground truth conformations that are similar to the generated conformation, and in turn, indicates how likely the generated molecule is in a low energy state. The matching value is the least RMSD value possible between the generated molecule and the conformation in the ground truth set. To measure conformations quality at different levels, we ran experiments both on atom coordinates and fragment coordinates.

Result and Discussion We can see from Table 2 that our model outperforms EDM on all the metrics we tested. Though our model only generates center coordinates for the fragments and the atom coordinates depend highly on predefined rules, our model is able to achieve impressive results both on the atom level and the fragment level. According to the results we visualize in Figure 5 and Figure 9, the structures generated from EDM are more chaotic, on the contrary, our model generates much more stable molecular scaffolds.

6 CONCLUSION

This paper is concerned with 3D molecule generation. To address the irrational molecule structure and biased molecule size problems caused by existing atomic auto-regressive models, a hierarchical diffusion probabilistic model is proposed. We also carefully design our method so that it can solve the combinatorially constrained structure generation problem introduced by non-autoregressive fragment generation modeling in an ordering agnostic way. Our model generates better drug-likeliness molecules, in terms of several widely used evaluation metrics. We believe that the proposed framework could inspire general solutions for other constrained structure generation tasks, such as Dispatching Route Generation (Ding et al., 2021), Optimal Experiment Design (Le Bras et al., 2012), and Protein Alignment Generation (Xu et al., 2015).

REFERENCES

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

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing, 2021. URL https://arxiv.org/abs/2110.02905.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. *arXiv preprint arXiv:1811.02549*, 2018.

Varsha Chauhan, Alexander Gutfraind, and Ilya Safro. Multiscale planar graph generation. *Applied Network Science*, 4(1):1–28, 2019.

Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*, 2018.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Fan Ding, Jianzhu Ma, Jinbo Xu, and Yexiang Xue. Xor-cd: Linearly convergent constrained structure generation. In *International Conference on Machine Learning*, pp. 2728–2738. PMLR, 2021.

Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020.

Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M Glass, and Jimeng Sun. Mimosa: Multi-constraint molecule sampling for molecule optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 125–133, 2021.

Erik Gawehn, Jan A Hiss, and Gisbert Schneider. Deep learning in drug discovery. *Molecular informatics*, 35(1):3–14, 2016.

Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a4d8e2a7e0d0c102339f97716d2fdfb6-Paper.pdf.

Tomas Hansson, Chris Oostenbrink, and WilfredF van Gunsteren. Molecular dynamics simulations. *Current Opinion in Structural Biology*, 12(2):190–196, 2002. ISSN 0959-440X. doi: https://doi.org/10.1016/S0959-440X(02)00308-1. URL https://www.sciencedirect.com/science/article/pii/S0959440X02003081.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? *arXiv preprint arXiv:1905.10617*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021. URL https://arxiv.org/abs/2102.05379.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018a.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *arXiv preprint arXiv:1812.01070*, 2018b.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical graph-to-graph translation for molecules. *arXiv preprint arXiv:1907.11223*, 2019.

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

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation, 2022. URL https://arxiv.org/abs/2206.01729.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32 (5):922–923, 1976.

Douglas B Kell, Soumitra Samanta, and Neil Swainston. Deep learning and generative methods in cheminformatics and chemical biology: navigating small molecule space intelligently. *Biochemical Journal*, 477(23):4559–4580, 2020.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International conference on machine learning*, pp. 1945–1954. PMLR, 2017.

Maksim Kuznetsov and Daniil Polykovskiy. Molgrow: A graph normalizing flow for hierarchical molecular generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8226–8234, 2021.

Youngchun Kwon, Jiho Yoo, Youn-Suk Choi, Won-Joon Son, Dongseon Lee, and Seokho Kang. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics*, 11(1):1–10, 2019.

Ronan Le Bras, Carla Gomes, and Bart Selman. From streamlined combinatorial search to efficient constructive procedures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pp. 499–506, 2012.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):1–24, 2018a.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018b.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks, 2021. URL https://arxiv.org/abs/2102.05013.

Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

Youzhi Luo and Shuiwang Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=C03Ajc-NS5W.

Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4227–4236, 2021.

Joshua Mitton, Hans M Senn, Klaas Wynne, and Roderick Murray-Smith. A graph vae and graph transformer approach to generating molecular graphs. *arXiv preprint arXiv:2104.04345*, 2021.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. *arXiv preprint arXiv:2205.07249*, 2022.

Alexander Powers, Helen Yu, Patricia Suriana, and Ron Dror. Fragment-based ligand generation guided by geometric deep learning on protein-ligand structure. *bioRxiv*, 2022.

Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Learning a continuous representation of 3d molecular structures with deep generative models. *arXiv preprint arXiv:2010.08687*, 2020.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574, 2015.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks, 2021. URL https://arxiv.org/abs/2102.09844.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows, 2022.

Florian Schmidt. Generalization in generation: A closer look at exposure bias. *arXiv preprint arXiv:1910.00292*, 2019.

Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. 2017. doi: 10.48550/ARXIV.1706.08566. URL https://arxiv.org/abs/1706.08566.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131, 2018.

Woong-Hee Shin, Xiaolei Zhu, Mark Gregory Bures, and Daisuke Kihara. Three-dimensional compound comparison methods and their application in drug discovery. *Molecules*, 20(7):12841–12862, 2015.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pp. 412–422. Springer, 2018.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Nathaniel Thomas, Tess E. Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://arxiv.org/abs/1802.08219.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Clement Vignac and Pascal Frossard. Top-n: Equivariant set and graph generation without exchangeability. *arXiv preprint arXiv:2110.02096*, 2021.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds, 2018. URL https://arxiv.org/abs/1811.07246.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Song Xianduo, Wang Xin, Song Yuyuan, Zuo Xianglin, and Wang Ying. Hierarchical recurrent neural networks for graph generation. *Information Sciences*, 589:250–264, 2022.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery. *arXiv preprint arXiv:2103.10432*, 2021.

Jinbo Xu, Sheng Wang, and Jianzhu Ma. *Protein homology detection through alignment of markov random fields: using MRFalign*. Springer, 2015.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Nianzu Yang, Huaijin Wu, Junchi Yan, Xiaoyong Pan, Ye Yuan, and Le Song. Molecule generation for drug design: a graph learning perspective. *arXiv preprint arXiv:2202.09212*, 2022a.

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation, 2022b. URL https://arxiv.org/abs/2203.09481.

Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 7924–7936. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/41da609c519d77b29be442f8c1105647-Paper.pdf.

Dawei Zhou, Lecheng Zheng, Jiejun Xu, and Jingrui He. Misc-gan: A multi-scale generative model for graphs. *Frontiers in big Data*, 2:3, 2019.

A Appendix

A.1 Discussion and Implementation for Fragmentizing the Molecule

To decrease the freedom in modeling large-size molecules, many models adopt fragment-based generation instead of building a model directly on atoms (Yang et al., 2022a). A number of methods are developed to break a molecule into a set of fragments. A good decomposing algorithm should satisfy that the derived fragment vocabulary needs to cover most of the molecular structures and also maintains a reasonable vocabulary size. JT-VAE (Jin et al., 2018a) is the first deep-learning method that generates molecule graphs at the fragment level. It derived fragments by applying the minimum spanning tree algorithm to keep all chemical bond information while avoiding cycles. JT-VAE (Jin et al., 2018a) succeed to cover all buyable structures with a vocabulary size of less than 800. Recent works like MARS (Xie et al., 2021), FREED (Yang et al., 2021), MIMOSA (Fu et al., 2021), FragSBDD (Powers et al., 2022) though applied different criterion on breaking bonds to generate fragment vocabulary, fragments of low frequency need to be removed from the vocabulary to keep the vocabulary size reasonable.

The chemical space of drug-like molecules is enormous. Leaving out fragments of low frequency is undesirable. Therefore, we adopt the tree decomposition algorithm from Jin et al. (2018a) in a 3D space. The procedure of processing the molecules into fragment graphs is a four-step process. **Extract components** We extract the set of chemical bonds which do not belongs to any rings and

the set of simple rings which only represent a single topological cycle from the molecules. **Merging** The Bridged ring is a cluster of important chemical structures. They possess uncommon 3D conformation. Therefore, all pairs of rings are merged if the ring pair has more than two overlapping atoms. **Edge linking** Cycles in the fragment graph will cause problematic modeling since the decomposition for a molecule is not unique. To avoid cycles, the intersecting atom which connects more than 3 bonds is added to the graph as a fragment. Edges are linked between all fragment pairs that have overlapping atoms. The minimum spanning tree algorithm is run on this graph to remove overlapping edges. **3D coarse set** At last, we assign 3D geometric information using the center of mass of the atoms within the fragment and coarse features for each fragment.

A.2 Coarse Set diffusion model

In this section, we describe the non-autoregressive high-level feature generative model and its likelihood computation. Though diffusion models have been receiving outstanding results in computer vision (Yang et al., 2022b; Ho et al., 2020; Vahdat et al., 2021), it was nontrivial to apply directly on molecule fragment graphs. The graph features include integer features, continuous features, and continuous coordinates. These different vectors require different likelihood computationsXu et al. (2022); Hoogeboom et al. (2022).

The diffusion model adds noise sequentially to the feature and coordinates like Eq. 2. At the time t, the data distribution of invariant features is expected to approximate the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. While in order to guarantee equivariance, the prior distribution for coordinates needs more constraint, it has been proven that moving the normal distribution into a linear subspace where $\sum_i^{i=3} H_{pi} = \mathbf{0}$ (Xu et al., 2022).

The model minimizes the lower bound of the log-likelihood:

$$\log P(H) \geq \mathcal{L}_0 + \mathcal{L}_{\text{base}} + \sum_{t=1}^{T} \mathcal{L}_t \tag{11}$$

where:

$$\mathcal{L}_0 = \log P\left(H \mid \boldsymbol{x}_0\right) \tag{12}$$

$$\mathcal{L}_{\text{base}} = -\mathrm{KL}\left(q\left(\boldsymbol{x}_T \mid H\right) \mid P\left(\boldsymbol{x}_T\right)\right) \tag{13}$$

$$\mathcal{L}_t = -\mathrm{KL}\left(q\left(\boldsymbol{x}_s \mid H, \boldsymbol{x}_t\right) \mid P\left(\boldsymbol{x}_s \mid \boldsymbol{x}_t\right)\right) \tag{14}$$

$\mathcal{L}_t$ and $\mathcal{L}_{base}$ can be computed easily by estimating the KL divergence between the estimated distribution and the target Gaussian distribution. However, $\mathcal{L}_0$ needs special treatment. Following the previous works (Hoogeboom et al., 2022; 2021), we define the $\mathcal{L}_0$ as follows:

$$P\left(H_f^{int} \mid \boldsymbol{x}_0^{(H)}\right) = \int_{H_f^{int}-\frac{1}{2}}^{H_f^{int}+\frac{1}{2}} \mathcal{N}\left(\boldsymbol{u} \mid \boldsymbol{x}_0^{(H_f^{int})} \sigma_0\right) \mathrm{d}\boldsymbol{u} \tag{15}$$

$$P\left(H_f^{cont} \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(H_f^{cont} \mid \boldsymbol{x}_0^{(H_f^{cont})}/\alpha_0 - \sigma_0/\alpha_0 \hat{\boldsymbol{\epsilon}}_0, \sigma_0^2/\alpha_0^2 \mathbf{I}\right) \tag{16}$$

$$P\left(H_p \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(H_f^{cont} \mid \boldsymbol{x}_0^{(H_p)}/\alpha_0 - \sigma_0/\alpha_0 \hat{\boldsymbol{\epsilon}}_0, \sigma_0^2/\alpha_0^2 \mathbf{I}\right) \tag{17}$$

For integer features, we centered the distribution to h_{int} and integrate from $-1/2$ to $1/2$. While for continuous features and coordinates, the variance of the distribution is still approximated by the network. During sampling, our model used a regular reverse diffusion to generate features and coordinates. The only difference is that the integer feature dimensions are normalized using the round function.

A.3 Equivariant neural network implementation

Improved EGNN In the node/edge sampling process, our nodes are endowed with a set of invariant features and equivariant coordinates. Inspired by the recent equivariant neural networks (EGNN) (Satorras et al., 2021; Thomas et al., 2018; Brandstetter et al., 2021), we propose an

improved version of the equivariant graph neural network. Each layer is formulated as:

$$m_{uv} = \phi_m\left(n_u^l, n_v^l, \left\|x_u^l - x_v^l\right\|^2, \mathbf{e_{uv}^l}\right) \quad x_u^{l+1} = x_u^l + c\tanh\left(\sum_{v\in\mathcal{N}(u)}\left(x_u^l - x_v^l\right)\phi_x\left(m_{uv}\right)\right),$$

$$n_u^{l+1} = \phi_n\left(n_u^l, \sum_{v\in\mathcal{N}(u)}\left(m_{uv}\right)\right) \qquad \mathbf{e_{uv}^{l+1}} = \phi_e\left(\mathbf{e_{uv}^l}, m_{uv}, \left\|x_u^l - x_v^l\right\|^2\right)$$

n and e stands for the node/ edge embedding, while c is a distance constant. x stands for node coordination. All ϕ are normal trainable MLPs. Previous works explore various kinds of techniques to maintain the equivariance of node features, however, the invariant edge features are always ignored to encode into the latent variables. Edge latent variables are needed for edge prediction in our methods. As a result, instead of carrying out the message passing on fully connected graphs with unified edges, we assigned edge features for sampling tasks. $\phi_{focal}, \phi_{edge}, \phi_{node}$ uses this improved network for message passing.

A.4 Algorithm for Fragments to Atoms in 3D

Algorithm 1 Algorithm for Conformation Alignment

Input: Fragment center coordinate: F_{out}, Molecule fragment graph: G
Output: C_{out}

function KABSCH($X \in \mathbb{R}_3, \hat{X} \in \mathbb{R}_3$)
  $X_c = \sum_{i=1}^{n} X_i, \hat{X}_c = \sum_{i=1}^{n} \hat{X}_i$
  $X = X - X_c, \hat{X} = \hat{X} - \hat{X}_c$
  $H = \sum_{i=1}^{n} X\hat{X}^T$
  $H = U\Lambda V^T$
  $R = \left(UV^T\right)^T$
  $t = \hat{X}_c - RX_c$
  return R, t
end function
$C_{\text{in}}, F_{\text{in}} \leftarrow$ RDkit random conformation and fragment positions
$C_{\text{out}} \leftarrow C_{\text{in}}$
for $n \in BFS(G)$ **do**
  $n_{\text{frag}}, n_{\text{atom}} \leftarrow$ fragment index , atom index of n
  $n_{\text{frag}}^{\text{nei}}, n_{\text{atom}}^{\text{nei}} \leftarrow$ fragment index , atom index of n.neighbors
  if n is not root **then**
    $n_{\text{frag}}^{\text{par}}, n_{\text{atom}}^{\text{par}} \leftarrow$ fragment index , atom index of n.parent
    $n_{\text{attach}} \leftarrow n_{\text{atom}}^{\text{par}} \cap n_{\text{atom}}$
    ref $= \{F_{\text{in}}[\, n_{\text{frag}}, n_{\text{frag}}^{\text{nei}}\,], C_{\text{in}}[\, n_{\text{attach}}\,]\}$
    out $= \{F_{\text{out}}[\, n_{\text{frag}}, n_{\text{frag}}^{\text{nei}}\,], C_{\text{out}}[\, n_{\text{attach}}\,]\}$
  else
    ref $= \{F_{\text{in}}[\, n_{\text{frag}}, n_{\text{frag}}^{\text{nei}}\,]\}$
    out $= \{F_{\text{out}}[\, n_{\text{frag}}, n_{\text{frag}}^{\text{nei}}\,]\}$
  end if
  $R, t =$ KABSCH(ref, out)
  $C_{\text{out}}\left[n_{\text{atom}}\right] = RC_{\text{out}}\left[n_{\text{atom}}\right] + t$
end for

A.5 Experimental configuration

Both our model and the baseline model are trained on the $\text{GEOM}_{\text{DRUG}}$ (Axelrod & Gomez-Bombarelli, 2022), CrossDocked2020 (Francoeur et al., 2020) and QM9 (Axelrod & Gomez-Bombarelli, 2022). In $\text{GEOM}_{\text{DRUG}}$ experiments, we randomly selected 4 conformations of each molecule to train our model. To test EDM (Hoogeboom et al., 2022), we removed hydrogen atoms

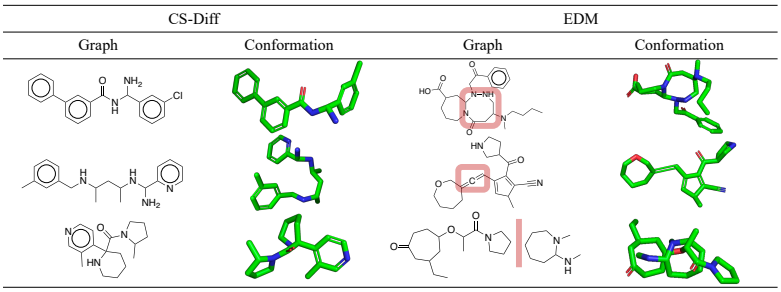


Figure 5: Visualization of the generated molecules, unstable or broken sub-structures from the EDM sampled molecules are highlighted

from the conformations and retrained the EDM model. The implicit hydrogen atoms are reconstructed using RDkit after all other heavy atoms are generated. Because EDM only generates atom types and coordinates, a proportion of sampled molecules are not fully connected. The broken fragments were removed for numerical evaluation. In all non-autoregressive methods, the number of nodes used for sampling is drawn from the size distribution histogram calculated on the training set.

Conformation Generation In this paragraph, we introduce the way to generate ground truth conformation using MD simulation. Firstly, 50 initial conformations are generated for each molecule graph using RDkit and optimized by MMF field. Then, these conformations are further optimized by MD software XTB, while the energy terms are computed for each conformation. At last, we choose the conformation with the minimum energy to sample the ground truth conformations using MD software CREST. To balance between efficiency and accuracy, we set the level of optimization to 'normal' in the software for both energy computing and conformation sampling. It took approximately 16 days to generate conformations for 400 different molecules on a 128-core server.

Table 3: Properties of the generated molecules from CrossDocked2020 (Francoeur et al., 2020). Δ indicates that the evaluated metrics are computed as the difference between sampled molecules and the ground truth and the absolute values are listed in the ().

	QED↑	RA↑	MCF↑	SAS↓	$\Delta_{\text{LogP}}\downarrow$ (logP)	$\Delta_{\text{MW}}\downarrow$ (MW)
G-SphereNet	0.442	–	0.449	–	3.359 (0.351)	200.95 (136.1)
EDM	0.499	0.332	0.613	7.056	2.840 (0.870)	28.40 (308.6)
CS-Diff$_{\text{E}}$	**0.614**	**0.574**	**0.759**	**4.051**	1.872 (1.838)	44.56 (292.4)
CS-Diff$_{\text{P}}$	0.585	0.262	0.687	5.397	**1.176 (2.535)**	**1.15 (338.2)**
CrossDocked	0.619	0.912	0.746	2.564	0.000 (3.710)	0.000 (337.0)

A.6 Additional Experiments

A.6.1 Evaluation on QM9

Metrics Although our model is designed to generate drug-like molecules with relatively large molecule sizes, it can be applied for smaller organic molecule (QM9) generation tasks without effort. We measure the validity and uniqueness metric on 10000 generated small organic molecules and compared them with various baselines by using RDkit. **Baselines** Our method is compared with previous methods. Both graph-based and coordinate-based models are included here. Graph-based methods like Graph VAE (Simonovsky & Komodakis, 2018), GTVAE (Mitton et al., 2021), and Set2GraphVAE (Vignac & Frossard, 2021), do not explicitly define the coordinates, so they need cheminformatic software to generate conformers. On the other hand, 3D coordinate-based models like E-NF (Satorras et al., 2022), G-Schnet (Gebauer et al., 2019), and EDM (Hoogeboom et al., 2022), need cheminformatic software to derive chemical bonds.

Results As shown in Table 4, our method performs comparable results in both validity and uniqueness. Though EDM (Hoogeboom et al., 2022) achieved better performance, our method still outperforms all other models. The slight performance drop compared to EDM could be due to the

information loss ratio during fragmentization on the tiny graphs. Besides, our model is the only 3D method that does not depend on any chemical bond linking software, like Openbabel. Hydrogen atoms can be added by counting the valency for each atom in our method.

Table 4: Validity and Uniqueness over 10000 molecules generated by different models. Geometry: model generation in 3D space. H: model Hydrogen atoms. Bond: model chemical bonds.

Method	Geometry	H	Bond	Valid (%)	Unique (%)	Valid and unique (%)
Graph VAE			✓	55.7	75.9	42.3
GTVAE			✓	74.6	22.5	16.8
Set2GraphVAE			✓	59.9	93.8	56.2
E-NF	✓	✓		40.2	98.0	39.4
G-Schnet	✓	✓		85.5	93.9	80.3
EDM	✓	✓		**91.9**	**98.7**	**90.7**
CS-Diff$_{\text{E}}$ (ours)	✓		✓	87.8	97.9	86.0
CS-Diff$_{\text{P}}$ (ours)	✓		✓	83.6	98.5	82.3

A.6.2 Evaluation of drug-like properties for fragments

Ring Size Ring Systems with the size of 5-6 are stable chemical groups in organic chemical theories. **HeteroAtom** The number of heteroatoms represents area of the polar surface in the organic molecules, which highly determines the distribution of the drug molecules in the human body, *e.g.*, the drug molecules that can cross the blood-brain barrier always has fewer heteroatoms. **AromaticRing** The Number of aromatic rings in the molecules indicates the ability to form $\pi - \pi$ interaction with proteins or other biomolecules. Aromatic rings also stabilize the molecule into lower energy conformations. **AliphaticRing** The Number of aliphatic rings in the molecules indicates the rigidity of the molecules. Instead of lying in a plane as aromatic rings, aliphatic rings constrained the conformation by contributing a specific torsion angle to the molecule conformation. **Radius** The mean radius of the fragment. A higher radius than the GEOM$_{\text{DRUG}}$ indicates too many rings are generated by the model. A smaller radius indicates the model is not able to construct valid ring systems.

Result and Discussions In addition to the evaluation of properties on the molecule level, we also break all the sampled molecules into fragments and test their performance on additional properties As expected, our method chooses fragments that are similar to that of ground truth statistics. We plotted the distribution of ring size in Fig. 6, the number of our method conforms best with ground truth. It is obvious that ring sizes 5 and 6 are most commonly seen in drug datasets and our sampled results, which are stable. However, on the contrary, the atom-based method such as EDM (Hoogeboom et al., 2022) has failed to capture this basic chemical rule. Refer to Table 5 for additional property evaluation.

Table 5: Properties of the fragments. All molecules are decomposed into fragments for statistical analysis. The model performs better if the generated fragments have more similar properties to GEOM$_{\text{DRUG}}$.

	Ring Size	HeteroAtom	AromaticRing	AliphaticRing	Radius
EDM	6.038	0.605	0.065	0.101	1.265
CS-Diff$_{\text{E}}$	**5.749**	0.630	0.104	**0.083**	1.295
CS-Diff$_{\text{P}}$	5.714	**0.660**	**0.132**	0.082	**1.360**
GEOM$_{\text{DRUG}}$	5.747	0.677	0.134	0.066	1.351

A.6.3 Evalutation of Uniqueness and Diversity on GEOM$_{\text{DRUG}}$

To prove that our method does not occur the issue of mode collapse, we tested the uniqueness of generated molecules and evaluate the similarity of generated molecules with the GEOM$_{\text{DRUG}}$ test set. **Similarity** which measures the average similarity between generated molecules with the most

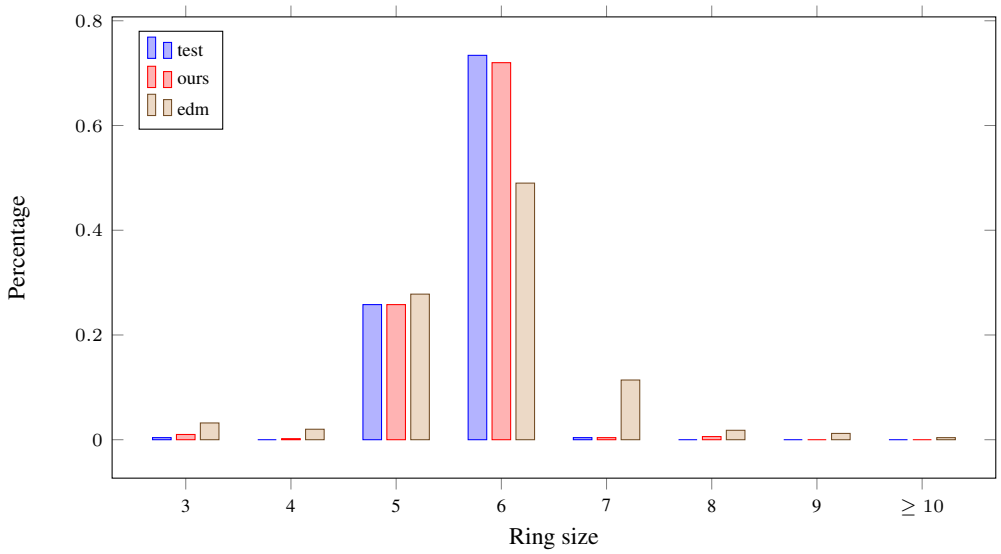


Figure 6: Histogram of the sampled molecules' ring size frequency of our model, EDM, and $\text{GEOM}_{\text{DRUG}}$

similar molecule in the test set. We use the Tanimoto score between ECFP4 fingerprints to measure the similarity between two molecules. High similarity indicates that the method lack generalization. **Unique** is the proportion of unrepeated structures in generated molecules. Both metrics are tested on molecule level and Murko Scaffold level. Numeric results are listed in Table 6 It is quite clear that our model generates more diverse molecules.

Result Our method outperforms EDM (Hoogeboom et al., 2022) on this uniqueness and diversity test. It should be noted that both our method and EDM generate mostly unique molecules when training on $\text{GEOM}_{\text{DRUG}}$. These methods succeed to generate diverse 3D molecules. Combining to results from Table 1, our method is able to generate 3D molecules that are drug-like and diverse.

Table 6: Diversity metrics computed on 1000 drug-like molecules generated by our method with two types of coarse feature and EDM (Hoogeboom et al., 2022).

	Similarity-atom ↓	Unique-atom ↑	Similarity-scaffold ↓	Unique-scaffold ↑
EDM	0.176	1.000	0.189	0.930
$\text{CS-Diff}_{\text{E}}$	**0.164**	1.000	0.171	**0.957**
$\text{CS-Diff}_{\text{P}}$	0.168	1.000	**0.169**	0.946

A.6.4 Experimental proof of error accumulation

One of our motivations for developing a hierarchical method for molecule generation is that we discovered the error accumulation in molecule generative models. This means that when the molecule size increase, the error from the previous steps of generation influences later steps which leads to unrealistic results. This issue has been discussed in the field of natural language modeling (Schmidt, 2019; He et al., 2019; Caccia et al., 2018). To prove this issue exists, we trained our method which represents the non-autoregressive method, and G-Spherenet which represents the autoregressive method on QM9. Both methods are set to generate molecules with the given molecule size. We test the validity of the generated molecules. We also do the same test on $\text{GEOM}_{\text{DRUG}}$, however, the validity of the autoregressive model drops so fast that it cannot generate molecules with more than 20 heavy atoms. Numeric results on QM9 are listed in Table 7. Visualized results of $\text{GEOM}_{\text{DRUG}}$ can be found in Figure 10.

Table 7: Validity of sampled molecules with different sizes trained on QM9 (Axelrod & Gomez-Bombarelli, 2022). All molecules that are broken or marked as invalid in RDkit package are regarded as invalid samples. AR stands for the autoregressive model in G-SphereNet (Luo & Ji, 2022). non-AR stands for our method.

Size	5	6	7	8
Valid (AR)	0.710	0.692	0.690	0.588
Valid (non-AR)	1.000	0.950	0.953	0.991

Table 8: Pearson correlation between the generated condition and true properties. When applying the molecule generation model for conditional generation task, we expect the model to output molecules with properties as the context we input.

	P-SAS↑	P-QED↑
EDM	-0.246	0.048
CS-Diff$_{\text{E}}$	**0.597**	**0.401**

A.6.5 Conditional Generation Experiments

Evaluation metrics In order to demonstrate our model's capacity to capture abstract chemical information without relying on human prior, we conduct a conditional generation experiment using the element coarse feature model. We chose two properties, SAS and QED which stand for the synthesis complexity of the molecules and the drug-likeness. Both of the properties represent high-level information that cannot be tricked by the generative model by simply outputting more specific types of elements. We use the same range of properties for conditional generation and we compute the Pearson correlation between the real properties of the sampled molecules with the input condition.

Result and Discussion It's pretty obvious in Table 8 that our model is able to capture abstract information for the conditional generation task. However, EDM tends to output random guesses for high-level context. We outperform the baseline model by a large margin which indicates our model can be used for generating molecules with desirable properties in the 3D space.

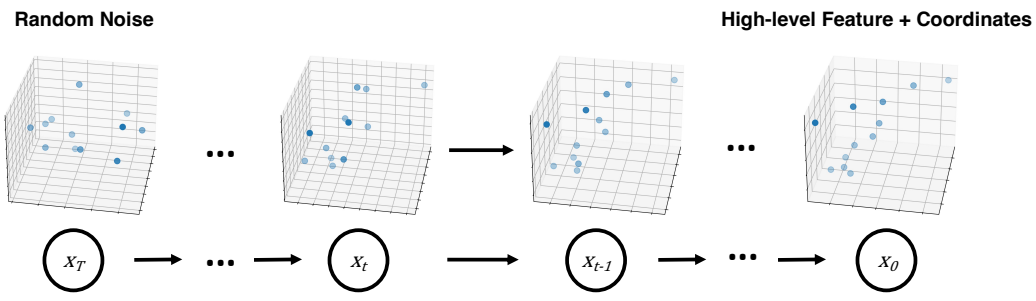


Figure 7: Illustration of the process of sampling high-level features and coordinates. The Whole chain start with the Gaussian noise and the 3D point set is refined by the reverse process of the diffusion model.

Table 9: Property-Based high-level feature

Property	Description	Type
HBA	Numbers of hydrogen bond acceptor	integer
HBD	Numbers of hydrogen bond donor	integer
Charge	Numbers of explicit electric charge	integer
Aromaticity	The size of aromatic ring	integer
Alicyclicity	The size of alicyclic ring	integer
Radius	The radius of the force filed optimized conformation	continuous
PSA	Polar surface area contribution to the conformation	continuous
ASA	Accessible surface area contribution to the conformation	continuous

Table 10: Element-Based high-level feature

Property	Description	Type
Hydrophobicity	Numbers of C element	integer
Hydrogen Bond Center	Numbers of O, N, S, P element	integer
Negative Charge Center	Numbers of F, Cl, Br, I element	integer

A.7 Training and Sampling Algorithm for decoding

Algorithm 2 Training Algorithm for Node/edge decoding

Input: 3D molecules set: $\{G\}$, EGNN networks: $\phi_{\text{focal}}, \phi_{\text{edge}}, \phi_{\text{node}}, \phi_{\text{refine}}$
Output: EGNN networks: $\phi_{\text{focal}}, \phi_{\text{edge}}, \phi_{\text{node}}, \phi_{\text{refine}}$

function C(S, E: subgraph)
    feat $\leftarrow$ Coarse-grained feature of $n, n \in S$
    coord $\leftarrow$ Position of $n, n \in S$
    return feat, coord
end function
function F(S, E: subgraph)
    feat $\leftarrow$ Fine-grained feature of $n, n \in S$
    coord $\leftarrow$ Position of $n, n \in S$
    edge $\leftarrow \{i, j\}, \{i, j\} \in E$
    return feat, coord, edge
end function
function FRAG(n: node)
    feat $\leftarrow$ Fine-grained feature of n
    return feat
end function
for G in $\{\text{G}\}$ **do**
    $T \sim T \in G$, s.t. T is connected subgraph
    $n \sim n \in T$, s.t. n is leaf node
    $m \sim m \in T$, s.t. m is single node
    $\tilde{T} = T \setminus n, V = G \setminus T$
    $\hat{T} = T \setminus m$
    context $= (\text{F}(\tilde{T}), \text{C}(V \cup n))$
    $\mathcal{L}_{\text{sample}} = -\log P_{\phi_{\text{focal}}}(n.\text{parent} \mid \text{context})$
        $-\log P_{\phi_{\text{edge}}}(\{n, n.\text{parent}\} \mid \text{context})$
        $-\log P_{\phi_{\text{node}}}(\text{FRAG}(n) \mid \text{context}, \{n, n.\text{parent}\})$
    Update $\phi_{\text{focal}}, \phi_{\text{edge}}, \phi_{\text{node}} \leftarrow \text{Optimize}(\mathcal{L}_{\text{sample}})$
    $\mathcal{L}_{\text{refine}} = -\log P_{\phi_{\text{refine}}}(\text{FRAG}(m) \mid [(\text{F}(\hat{T}), \text{C}(m)])$
    Update $\phi_{\text{refine}} \leftarrow \text{Optimize}(\mathcal{L}_{\text{refine}})$
end for

Algorithm 3 Sampling Algorithm for Node/edge decoding

Input: Nodes with coarse-grained feature and positions: N
Refine step limit: max steps,
EGNN networks: $\phi_{\text{focal}}, \phi_{\text{edge}}, \phi_{\text{node}}, \phi_{\text{refine}}$
Output: Fine-grained Graph T

function GENERATE STEP(T)
  $n_{\text{focal}} \sim P_{\phi_{\text{focal}}}(n_{\text{focal}} \mid T)$
  $\{n_{\text{focal}}, n_{\text{new}}\} \sim P_{\phi_{\text{edge}}}(\{n_{\text{focal}}, n_{\text{new}}\} \mid T)$
  $T \leftarrow T + \{n_{\text{focal}}, n_{\text{new}}\}$
  fragment $\sim P_{\phi_{\text{node}}}(fragment \mid T)$
  FRAG$(n_{\text{new}}) \leftarrow$ fragment
end function
function REFINE STEP(T)
  $\text{T}_{\text{coarse}} \leftarrow$ coarse-grained T
  $\text{T}_{\text{fine}} \leftarrow$ fine-grained T
  $n_{\text{refine}} = \arg\min_n(P_{\phi_{\text{refine}}}(\text{FRAG}(n) \mid \text{T}_{\text{fine}} \setminus n, \text{T}_{\text{coarse}})), n \in T$
  fragment $\sim P_{\phi_{\text{refine}}}(\text{fragment} \mid \text{T}_{\text{fine}} \setminus n, \text{T}_{\text{coarse}})$
  FRAG$(n_{\text{refine}}) \leftarrow$ fragment
end function
function PROB(T)
  return $\sum_{n \in T}(\log P_{\phi_{\text{refine}}}(\text{FRAG}(n) \mid T \setminus n))$
end function
$T \leftarrow N$
repeat
  $T \leftarrow$ GENERATE STEP(T)
  for i in max steps **do**
    $\hat{T} \leftarrow$ REFINE STEP(T)
    if PROB$(\hat{T})$ > PROB(T) **then**
      Accept: $T \leftarrow \hat{T}$
    else
      Break
    end if
  end for
until $\forall_{n \in T}, n$ is fine-grained node

A.8 REPRODUCIBILITY

We provide anonymous code link[1] for reproducing the results in the paper.

[1] https://drive.google.com/drive/folders/1qoWnZccz9tph8pYJtVXWd8TmuK55GlXJ?usp=sharing

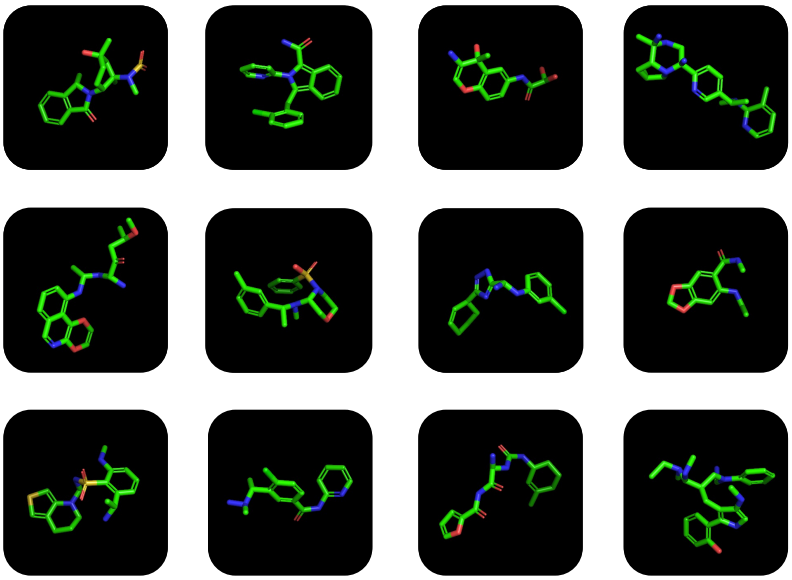


Figure 8: Visualized 3D conformations generated by our model

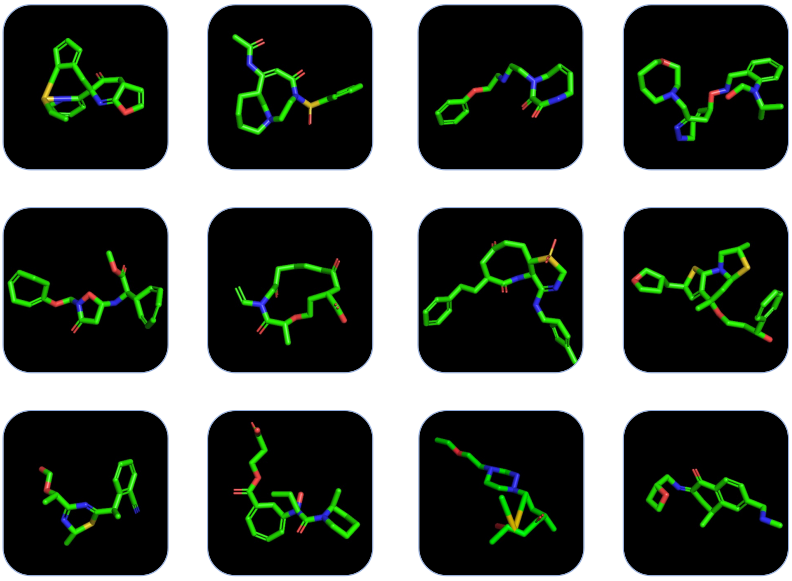


Figure 9: Visualized 3D conformations generated by the baseline model EDM

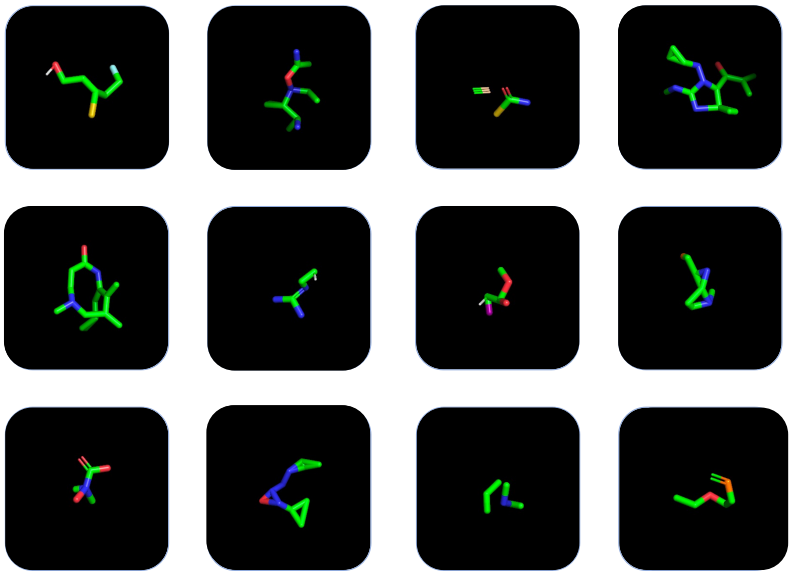


Figure 10: Visualized 3D conformations generated by G-SphereNet