MDM: Molecular Diffusion Model for 3D Molecule Generation

Lei Huang[1, 2*], **Hengtong Zhang**[2 †], **Tingyang Xu**[2], **Ka-Chun Wong**[1 †]

[1] City University of Hong Kong
[2]Tencent AI Lab
lhuang93-c@my.cityu.edu.hk, htzhang.work@gmail.com,
tingyangxu@tencent.com, kc.w@cityu.edu.hk

Abstract

Molecule generation, especially generating 3D molecular geometries from scratch (i.e., 3D *de novo* generation), has become a fundamental task in drug design. Existing diffusion based 3D molecule generation methods could suffer from unsatisfactory performances, especially when generating large molecules. At the same time, the generated molecules lack enough diversity. This paper proposes a novel diffusion model to address those two challenges.

First, interatomic relations are not included in molecules' 3D point cloud representations. Thus, it is difficult for existing generative models to capture the potential interatomic forces and abundant local constraints. To tackle this challenge, we propose to augment the potential interatomic forces and further involve dual equivariant encoders to encode interatomic forces of different strengths. Second, existing diffusion-based models essentially shift elements in geometry along the gradient of data density. Such a process lacks enough exploration in the intermediate steps of the Langevin dynamics. To address this issue, we introduce a distributional controlling variable in each diffusion/reverse step to enforce thorough explorations and further improve generation diversity.

Extensive experiments on multiple benchmarks demonstrate that the proposed model significantly outperforms existing methods for both unconditional and conditional generation tasks. We also conduct case studies to help understand the physicochemical properties of the generated molecules. The codes are available at https://github.com/tencent-ailab/MDM

Introduction

De novo molecule generation, which automatically generates valid chemical structures with desirable properties, has become a crucial task in the domain of drug discovery. However, given the huge diversity of atom types and chemical bonds, the manually daunting task in proposing valid, unique, and property-restricted molecules is extraordinarily costly. To tackle such a challenge, a multitude of generative machine learning models (Zang and Wang 2020; Satorras et al. 2021; Gebauer, Gastegger, and Schütt 2019; Hoogeboom et al. 2022), which automatically generate molecular geometries (i.e., 2D graphs or 3D point clouds) from scratch, has been proposed in the past decade.

Among those studies, 3D molecule generation has become an emerging research topic due to its capability of directly generating 3D coordinates of atoms, which are important in determining molecules' physical-chemical properties. Early studies on 3D molecule generation adopt autoregressive models such as normalized flow (Satorras et al. 2021) to determine the types and 3D positions of atoms one by one. Nevertheless, these models suffer from deviation accumulations, especially when invalid structures are generated in the early steps (i.e. initial condition vulnerability). Such a major drawback leads to unsatisfactory generation results in terms of molecule validity and stability. Later, inspired by the success of diffusion models, (Hoogeboom et al. 2022) proposed a cutting-edge diffusion based 3D generation model that significantly improves the validity of generated molecules. The diffusion based generation model (Hoogeboom et al. 2022) defines a Markov chain of diffusion steps to add random noises to 3D molecule geometries and then learns a reverse process to construct desired 3D geometries step-by-step.

Nonetheless, diffusion-based 3D generation models still suffer from two non-negligible drawbacks: First, unlike 2D generation, in which chemical bonds are represented as graph edges, molecular geometries in 3D generation are represented as point clouds (Satorras et al. 2021; Gebauer, Gastegger, and Schütt 2019; Hoogeboom et al. 2022). Hence, it is difficult for 3D generative models to capture the abundant local constraint relations between adjacent atoms with no explicit indications for chemical bonds. Such a significant drawback leads to unsatisfactory performance on datasets with large molecules, for instance, the GEOM-Drugs dataset (Axelrod and Gomez-Bombarelli 2022) with average 46 atoms per molecule.

Moreover, training diffusion models is essentially equivalent to a denoising score matching process with Langevin dynamics as existing literature (Ho, Jain, and Abbeel 2020; Song et al. 2021) suggests, in which the elements in a geometry (i.e., points in a 3D point cloud) shift along the gradient of data density at each timestep. Thus, the generation dynamics by given fixed initialized noise may concentrate around a common trajectory and lead to similar generation results, even with the standard Gaussian noise compensa-

[*]This work is done when Lei Huang works as an intern in Tencent AI Lab.

[†]Corresponding authors: Hengtong Zhang and Ka-Chun Wong.

tions in the sampling process. Such a phenomenon hurts the diversity of generated molecules in practice.

In this paper, we propose a novel model named MDM (**M**olecular **D**iffusion **M**odel) to tackle these drawbacks. First, we propose to treat atoms pairs with atomic spacing below the specified threshold[1] as covalently bonded since chemical bonds can dominate the interatomic force when two atoms are close enough to each other. We can thus construct augmented bond-like linkages between adjacent atoms. On the other hand, for the atom pairs with atomic spacing above the threshold, the van der Waals forces dominate the interatomic force. Those two types suggest different strengths between atoms and thus should be treated distinctly. Given such intuition, we deploy separated equivariant networks to explicitly model the destination between these two kinds of inter-atom bonds.

Moreover, to enhance the diversity of molecular generation, we introduce latent variables, interpreted as controlling representations in each diffusion/reverse step of a diffusion model. Thus, each diffusion/reverse step is conditioned to a distributional (e.g. Gaussian) representation that can be effectively explored. In the generation[2] phase, by sampling from the underlying distribution of the variable in each step, we can enforce thorough explorations to generate diverse 3D molecule geometries.

Experiments on two molecule datasets (i.e., QM9 (Ramakrishnan et al. 2014) and GEOM-Drugs (Axelrod and Gomez-Bombarelli 2022)) demonstrate that the proposed MDM outperforms the state-of-the-art model EDM (Hoogeboom et al. 2022) by a wide margin, especially on the drug-like GEOM-Drugs dataset that consists of molecules with a large number of atoms (46 atoms on average compared with 18 in QM9). Remarkably, the uniqueness and novelty metric, which characterizes the diversity of generative molecules, is improved by 4.1% to 31.4% compared with EDM. We also present studies on targeted molecular generation tasks to show that the proposed model is capable of generating molecules with chosen properties without scarifying the validity and stability of generated molecules.

Related Work

Deep generative models have recently exhibited their effectiveness in modeling the density of real-world molecule data for molecule design and generation. Various methods first consider molecule generation in a 2D fashion. Some methods (Dai et al. 2018; Gómez-Bombarelli et al. 2018; Grisoni et al. 2020) utilize sequential models such as RNN to generate SMILES (Weininger 1988) strings of molecules while other models focus on molecular graphs whose atoms and chemical bonds are represented by nodes and edges. Generally, these methods incorporate the variational autoencoder (VAE)-based models (Jin, Barzilay, and Jaakkola 2018), generative adversarial network (GAN) (De Cao and Kipf 2018) and normalizing flows (Zang and Wang 2020; Luo, Yan, and Ji 2021) to generate the atom types and the corresponding chemical bonds in one-shot or auto-regressive manners. Although these studies are able to generate valid and novel molecule graphs, they ignore the 3D structure information of molecules which is crucial for determining molecular properties.

Recently, generating molecules in 3D space has gained a lot of attention. For instance, G-Schnet (Gebauer, Gastegger, and Schütt 2019) employs an auto-regressive process equipped with Schnet (Schütt et al. 2017) in which atoms and bonds are sampled iteratively. E-NF (Garcia Satorras et al. 2021) instead utilizes a one-shot framework based on an equivariant normalizing flow to generate atom types and coordinates at one time. Recently, inspired by the success of diffusion models (Sohl-Dickstein et al. 2015) in various tasks (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021; Kong et al. 2021), (Hoogeboom et al. 2022) adopts the diffusion model to generate novel molecules in 3D space. However, it only utilizes the fully connected adjacent matrix thus ignoring the intrinsic topology of the molecular graph.

Apart from molecule generation, the task discussed in this paper is also related to conformation prediction (Mansimov et al. 2019; Köhler, Klein, and Noé 2020; Xu et al. 2021; Guan et al. 2022). Although both molecule generation and conformation prediction output 3D molecule geometries, the settings of these two tasks are different. The former can directly generate a complete molecule while the latter additionally requires molecular graphs as inputs and only outputs atom coordinates.

Preliminaries

Notation. Let $\mathcal{G} = (A, R)$ denote the 3D molecular geometry as where $A \in \{0,1\}^{n\times f}$ denotes the atom features, including atom types and atom charges. $R \in \mathbb{R}^{n\times 3}$ denotes the atom coordinates.

Diffusion Model

The diffusion model is formulated as two Markov chains: *diffusion process* and *reverse process* (a.k.a denoising process). In the upcoming paragraphs, we will elaborate on these two processes.

Diffusion Process. Given the real molecular geometry $\mathcal{G}_0$, the forward diffusion process gradually diffuses the data into a predefined noise distribution with the time setting $1 \dots T$, like the physical phenomenon. The diffusion model is formulated as a fixed Markov chain which gradually adds Gaussian noise to the data with a variance schedule $\beta_1 \dots \beta_T (\beta_t \in (0,1))$:

$$\begin{aligned} q\left(\mathcal{G}_{1:T} \mid \mathcal{G}_0\right) &= \prod_{t=1}^{T} q\left(\mathcal{G}_t \mid \mathcal{G}_{t-1}\right), \\ q\left(\mathcal{G}_t \mid \mathcal{G}_{t-1}\right) &= \mathcal{N}\left(\mathcal{G}_t; \sqrt{1-\beta_t}\mathcal{G}_{t-1}, \beta_t I\right), \end{aligned} \tag{1}$$

where $\mathcal{G}_{t-1}$ is mixed with the Gaussian noise to obtain $\mathcal{G}_t$ and β_t controls the extent of the mixture. By setting $\bar{\alpha}_t = \prod_{s=1}^{t} 1 - \beta_s$, a delightful property of the diffusion process is achieved that any arbitrary time step, t, sampling of the

[1] The distance value threshold varies for different types of bonds.

[2] Or sampling phase in the context of the diffusion model.

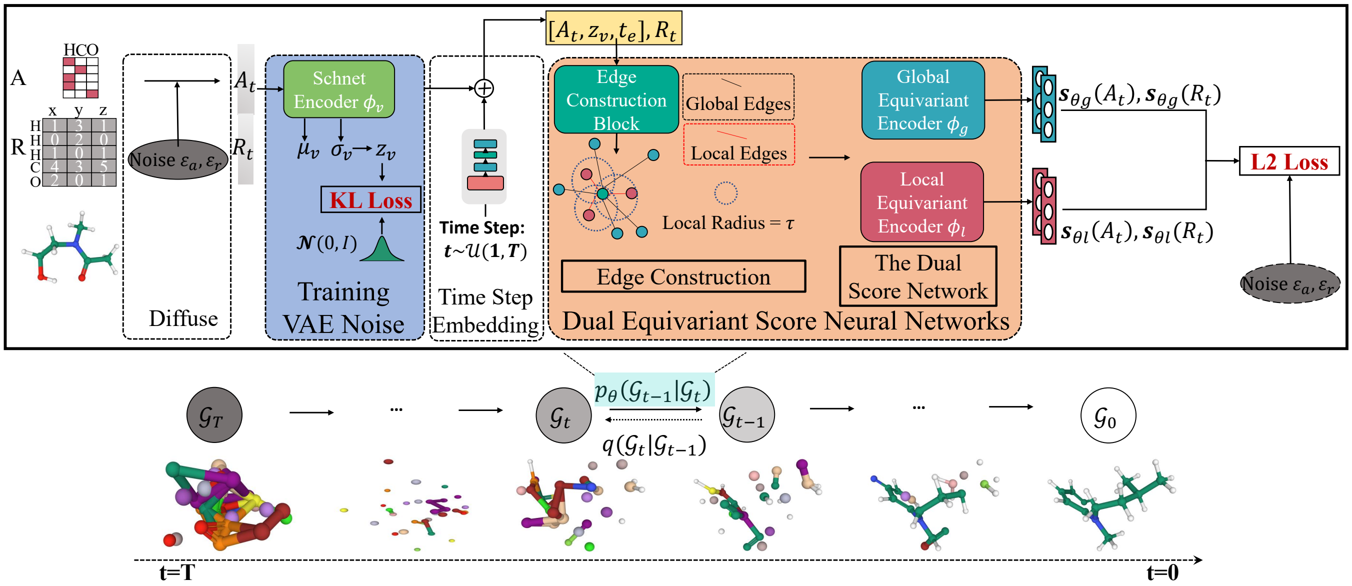
HCO
A
x y z
R
H
H
H
C
O
Noise $\varepsilon_a, \varepsilon_r$
Diffuse
A_t
R_t
Schnet Encoder ϕ_v
μ_v $\sigma_v \rightarrow z_v$
KL Loss
$\mathcal{N}(0, I)$
Training VAE Noise
Time Step: $t \sim \mathcal{U}(1, T)$
Time Step Embedding
$[A_t, z_v, t_e], R_t$
Edge Construction Block
Global Edges
Local Edges
Local Radius = τ
Edge Construction
Global Equivariant Encoder ϕ_g
Local Equivariant Encoder ϕ_l
The Dual Score Network
Dual Equivariant Score Neural Networks
$s_{\theta g}(A_t), s_{\theta g}(R_t)$
$s_{\theta l}(A_t), s_{\theta l}(R_t)$
L2 Loss
Noise $\varepsilon_a, \varepsilon_r$
$\mathcal{G}_T$
$\mathcal{G}_t$
$p_\theta(\mathcal{G}_{t-1}|\mathcal{G}_t)$
$q(\mathcal{G}_t|\mathcal{G}_{t-1})$
$\mathcal{G}_{t-1}$
$\mathcal{G}_0$
t=T
t=0

Figure 1. Overview of the training process of the proposed model MDM. The model would train each molecule which includes the atom features and atom coordinates with a stochastic time step. For the reverse process, the final molecule is generated by denoising the initial state $\mathcal{G}_T \sim \mathcal{N}(0, I)$ gradually with the Markov kernels $p_\theta(\mathcal{G}_{t-1} \mid \mathcal{G}_t)$. Symmetrically, the diffusion process is achieved by adding the noise with the posterior distribution $q(\mathcal{G}_t \mid \mathcal{G}_{t-1})$ until the molecule is degenerated into the white noise when the time step is large enough. It is also should be noted that the global and local equivariant encoders have the same structure.

data has a closed-form formulation via a reparameterization trick as:

$$q\left(\mathcal{G}_t \mid \mathcal{G}_0\right) = \mathcal{N}\left(\mathcal{G}_t; \sqrt{\bar{\alpha}_t}\mathcal{G}_0, \left(1 - \bar{\alpha}_t\right) I\right). \tag{2}$$

With step t gradually rising, the final distribution will be closer to the standard Gaussian distribution because $\sqrt{\bar{\alpha}_t} \rightarrow 0$ and $(1 - \bar{\alpha}_t) \rightarrow 1$ if $t \rightarrow \infty$.

Reverse Process. The reverse process aims to learn a process to reverse the diffusion process back to the distribution of the real data. Assume that there exists a reverse process, $q\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right)$. Then such a process could generate valid molecules from a standard Gaussian noise following a Markov chain from T back to 0 as shown in Figure 1. However, it is hard to estimate such distribution, $q\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right)$. Hence, a learned Gaussian transitions $p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right)$ is devised to approximate $q\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right)$ at every time step:

$$\begin{aligned} p_\theta\left(\mathcal{G}_{0:T-1} \mid \mathcal{G}_T\right) &= \prod_{t-1}^{T} p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right), \\ p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t\right) &= \mathcal{N}\left(\mathcal{G}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathcal{G}_t, t\right), \sigma_t^2 I\right), \end{aligned} \tag{3}$$

where $\boldsymbol{\mu}_\theta$ denotes the parameterized neural networks to approximate the mean, and σ_t^2 denotes user defined variance.

To learn the $\boldsymbol{\mu}_\theta\left(\mathcal{G}_t, t\right)$, we adopt the following parameterization of $\boldsymbol{\mu}_\theta$ following Ho, Jain, and Abbeel (2020):

$$\boldsymbol{\mu}_\theta\left(\mathcal{G}_t, t\right) = \frac{1}{\sqrt{1 - \beta_t}}\left(\mathcal{G}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathcal{G}_t, t\right)\right), \tag{4}$$

where $\boldsymbol{\epsilon}_\theta$ is a neural network w.r.t trainable parameters θ.

From another perspective, the reverse process, which eliminates the noise part of the data added in the diffusion process at each time step, is equivalent to a moving process on the data distribution that initially starts from a low-density region to the high-density region of the distribution led by the logarithmic gradient. Therefore, the negative eliminated noise part $-\boldsymbol{\epsilon}_\theta \cdot \sigma$ is also regarded as the *(stein) score* (Liu, Lee, and Jordan 2016), the logarithmic density of the data point at every time step. This equivalence is also reflected in the previous work (Song et al. 2021). For simplicity, we utilize $\boldsymbol{s}_\theta$ for all the related formulas in the following sections.

Now we can parameterize $\boldsymbol{\mu}_\theta\left(\mathcal{G}_t, t\right)$ as:

$$\boldsymbol{\mu}_\theta\left(\mathcal{G}_t, t\right) = \frac{1}{\sqrt{1 - \beta_t}}\left(\mathcal{G}_t + \beta_t \cdot \boldsymbol{s}_\theta\left(\mathcal{G}_t, t\right)\right). \tag{5}$$

The complete sampling process resembles Langevin dynamics with $\boldsymbol{s}_\theta$ as a learned gradient of the data density.

MDM: Molecular Diffusion Model

In this section, we present our proposed model MDM, a molecular diffusion model. As shown in Figure 1, we design *dual equivariant score neural networks* to handle two levels of edges: local edges within a predefined radius to model the intramolecular force such as covalent bonds and global edges to capture van der Waals forces. Furthermore, We introduce a *VAE module* inside the diffusion model to produce conditional noise which will avoid determining the output of the whole model and improve the generation diversity. Then, we describe how the *training phase* and *sampling phase* of MDM work.

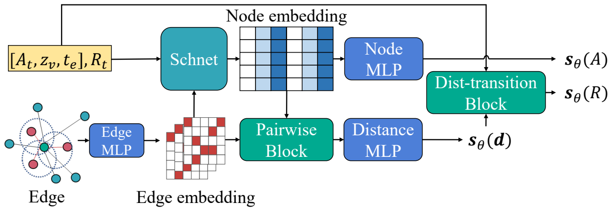

Figure 2. The illustration of the equivariant encoder where Schnet follows the implementation of Schütt et al. (2017).

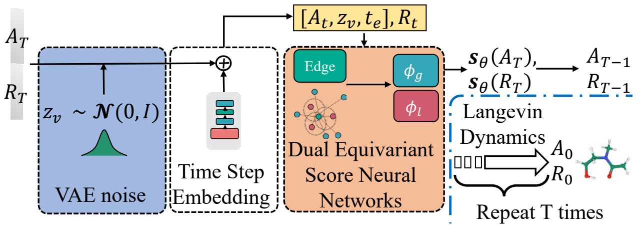

Figure 3. Overview of the sampling process of the proposed model MDM. A_T and R_T are sampled from $\mathcal{N}(0, I)$.

Dual Equivariant Score Neural Networks

As molecular geometries are roto-translation *invariant*, we should take this property into account when devising the Markov kernels. In essence, Köhler, Klein, and Noé (2020) proposed an equivariant invertible function to transform an *invariant* distribution into another *invariant* distribution. This theorem is also applied to the diffusion model (Xu et al. 2022). If $p(\mathcal{G}_T)$ is *invariant* and the neural network q_θ which learns to parameterize $p(\mathcal{G}_{t-1} \mid \mathcal{G}_t)$ is *equivariant*, then the distribution $p(\mathcal{G}_0)$ is also *invariant*. Therefore, we utilize an *equivariant* Markov kernel to achieve this desired property.

Edge Construction. Recalls that previous works (Köhler, Klein, and Noé 2020; Hoogeboom et al. 2022) consider the fully connected edges to feed into the equivariant graph neural network. However, the fully connected edges connect all the atoms and treat the interatomic effects equally but regret the effects of covalent bonds. Therefore, we further define the edges within the radius τ as local edges to simulate the covalent bonds and the rest of the edges in the fully connected edges as global edges to capture the long distance information such as van der Waals force.

Practically, we set the local radius τ as 2Å because almost all of the chemical bonds are no longer than 2Å. The atom features and coordinates with the local edges and global edges are fed into the dual equivariant encoder, respectively. Specifically, the local equivariant encoder models the intramolecular force such as the real chemical bonds via local edges while the global equivariant encoder captures the interactive information among distant atoms such as van der Waals force via global edges.

The Equivariant Markov Kernels. Both local and global equivariant encoders share the same architecture as Equivariant Markov Kernels. Intuitively, atom features, A, are invariant to any rigour transformations on atom coordinates R, while R should be equivariant to such transformations. Therefore, we design the equivariant Markov Kernels as Figure 2 to tackle invariance and equivariance of A and R, respectively.

First, we consider the invariance of the model for atomic features, A. Firstly, we utilize an **Edge MLP** to obtain edge embeddings as

$$\mathbf{h}_{e_{ij}} = \mathrm{MLP}(\mathbf{d}_{ij}, e_{ij}), \tag{6}$$

where $\mathbf{d}_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|_2$ denotes the Euclidean distance between the positions of atom i and atom j, and e_{ij} denotes the edge features between i^{th} atom and j^{th} atom. Then we adopt **Schnet** with L layers to achieve invariance:

$$\begin{aligned}\mathbf{h}_{iA}^0 =& \mathrm{MLP}(A_i), \mathbf{h}_{iR}^0 = \mathrm{MLP}(R_i), \mathbf{h}_i^0 = [\mathbf{h}_{iA}^0, \mathbf{h}_{iR}^0], \\ \mathbf{h}_i^{(l+1)} =& \sigma\left(\boldsymbol{W}_0^l \mathbf{h}_i^{(l)} + \sum_{j \in N(i)} \boldsymbol{W}_1^l \phi_w\left(\mathbf{d}_{ij}\right) \odot \boldsymbol{W}_2^l \mathbf{h}_j^{(l)}\right),\end{aligned} \tag{7}$$

where $l \in (0, 1, \ldots, L)$ indicates the l^{th} layer of Schnet, $\boldsymbol{W}^l$s represent the learning weights. Then, we denote the final outputs of **Schnet**, $\mathbf{h}_i$, as node embeddings, $\sigma(\cdot)$ denotes non-linear activation such as ReLU, and $\phi_w(\cdot)$ denotes a weight network. Then, the final outputs of **Schnet** $\mathbf{h}_i = \mathbf{h}_i^{(L)}$ are denoted as node embeddings.

In order to estimate the gradient of log density of atom features, we utilize one-layer **Node MLP** to map the latent hidden vectors outputted by **Schnet** to score vectors.

$$\boldsymbol{s}_\theta(A_i) = \mathrm{MLP}(\mathbf{h}_i). \tag{8}$$

On the other hand, to achieve equivariance for atomic coordinates, R, in 3D space, we attempt to decompose them to pairwise distances. Therefore, we concatenate the learned edge information and the product of the end nodes vectors of the same edges as **Pairwise Block** followed by a **Distance MLP** to get the gradient of the pairwise distances.

$$\boldsymbol{s}_\theta(\mathbf{d}_{ij}) = \mathrm{MLP}(\left[\mathbf{h}_i \cdot \mathbf{h}_j, \mathbf{h}_{e_{ij}}\right]). \tag{9}$$

Here, we omit t on $\boldsymbol{s}_\theta$ as we only discuss the coordinate part of this function in one time step for simplicity.

Then, MDM operates a transition, called **Dist-transition Block**, to integrate the information of score vectors from a pairwise distance and atomic coordinates R as follows

$$\mathbf{s}_\theta(R_i) = \sum_{j \in N(i)} \frac{1}{\mathbf{d}_{ij}} \cdot \mathbf{s}_\theta(\mathbf{d}_{ij}) \cdot (\mathbf{r}_i - \mathbf{r}_j), \tag{10}$$

where $\boldsymbol{s}_\theta(d)$ is invariant to translation since it only depends on the symmetry-invariant element d and $\mathbf{r}_i - \mathbf{r}_j$ is roto-translation equivariance. Thus $\boldsymbol{s}_\theta(R)$ shares the equivariant property.

Enhanced Diversity via Variational Noising

The diffusion model can be extended to a conditional generation by enforcing the generated samples with the additional given information. Therefore, we employ variational noising to import an additional noise z_v for conditional generation $p_\theta\left(\mathcal{G}_{0:T-1} \mid \mathcal{G}_T, z_v\right)$ and improve the diversity. Specifically, we adopt **Schnet** as the encoder and the subsequent equivariant modules as the decoder. The encoder outputs the mean μ_v and the standard deviation σ_v from which we can obtain the additional noise z_v by the reparameterization trick, $z_v = \mu_v + \sigma_v^2 z, z \sim \mathcal{N}(0, I)$. Hence, Eq. (3) in reverse process of diffusion model becomes

$$\begin{aligned} p_\theta\left(\mathcal{G}_{0:T-1} \mid \mathcal{G}_T, z_v\right) &= \prod_{t-1}^{T} p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t, z_v\right), \\ p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t, z_v\right) &= \mathcal{N}\left(\mathcal{G}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathcal{G}_t, z_v, t\right), \sigma_t^2 I\right). \end{aligned} \tag{11}$$

When forwarding the diffusion process, we sample the variational noise z_v from $\mathcal{N}(0, I)$. We also surprisingly observe that the performance is improved if we apply the polarized sampling strategy. Empirically, the performance of MDM improves significantly when we sample z_v from a uniform distribution $\mathcal{U}(-1, +1)$.

Training

损失函数

Having formulated the diffusion and reverse process, the training of the reverse process is performed by optimizing the usual variational lower bound (ELBO) on negative log-likelihood since the exact likelihood is intractable to calculate:

$$\begin{aligned} &\mathbb{E}\left[-\log p_\theta\left(\mathcal{G}\right)\right] \leq \mathbb{E}_{q(\mathcal{G}_0)}\left[-\log\left(\frac{p_\theta\left(\mathcal{G}_{0:T}\right)}{q\left(\mathcal{G}_{1:T} \mid \mathcal{G}_0\right)}\right)\right] \\ &= \mathbb{E}_{q(\mathcal{G}_0)}[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathcal{G}_T \mid \mathcal{G}_0\right) \| p\left(\mathcal{G}_T\right)\right)}_{\mathcal{L}_T} \\ &+ \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t, \mathcal{G}_0\right) \| p_\theta\left(\mathcal{G}_{t-1} \mid \mathcal{G}_t, z_v\right)\right)}_{\mathcal{L}_t} \\ &+ \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q_\phi(z_v \mid \mathcal{G}_t)}(\mathcal{D}_{KL}(q_\phi(z_v \mid \mathcal{G}_t)) \| p(z_v)))}_{\mathcal{L}_{vn,t}} \\ &- \underbrace{\log p_\theta\left(\mathcal{G}_0 \mid \mathcal{G}_1\right)}_{\mathcal{L}_0}], \end{aligned} \tag{12}$$

where $q_\phi(\cdot)$ denotes a learnable variational noising encoder. Following (Ho, Jain, and Abbeel 2020), $\mathcal{L}_T$ is a constant and $\mathcal{L}_0$ can be approximated by the product of the PDF of $\mathcal{N}\left(\mathrm{x}_0; \boldsymbol{\mu}_\theta\left(\mathrm{x}_1, 1\right), \sigma_1^2 I\right)$ and discrete bin width. Hence, we adopt the simplified training objective as follows:

$$\mathcal{L}_t = \mathbb{E}_{\mathcal{G}_0}\left[\gamma \|\boldsymbol{s}_\theta\left(\mathcal{G}_t, z_v, t\right) - \nabla_{\mathcal{G}_t} \log q_\sigma(\mathcal{G}_t \mid \mathcal{G}_0)\|^2\right], \tag{13}$$

where $\gamma = \frac{\beta_t^2}{2(1-\beta_t)(1-\bar{\alpha}_t)\sigma_t^2}$ refers to a weight term.

When $\nabla_{\mathcal{G}_t} \log q_\sigma(\mathcal{G}_t \mid \mathcal{G}_0)$ denotes a sampling process at stochastic t, the sampling on the atomic features A_t still remains invariant. However, the sampling on the atomic coordinates R_t may violate the equivariance. Hence, to maintain the equivariance of R_t, we sample the R_t on pairwise distance $\mathbf{d}_{ij}$ instead as

$$\begin{aligned} &\nabla_{\tilde{\mathbf{r}}_i} \log q_\sigma(\tilde{\mathbf{r}}_i \mid \mathbf{r}_i) \\ &= \sum_{j \in N(i)} \frac{\nabla_{\tilde{\mathbf{d}}_{ij}} \log q_\sigma(\tilde{\mathbf{d}}_{ij} \mid \mathbf{d}_{ij}) \cdot (\mathbf{r}_i - \mathbf{r}_j)}{\mathbf{d}_{ij}}, \end{aligned} \tag{14}$$

where $\tilde{\mathbf{r}}$ denotes the diffused atom coordinate of $\mathcal{G}_t$ and $\tilde{\mathbf{d}}$ denotes the corresponding diffused distance. We approximately calculate $\nabla_{\tilde{\mathbf{d}}} \log q_\sigma(\tilde{\mathbf{d}} \mid \mathbf{d})$ as $\frac{-\sqrt{\bar{\alpha}_t}(\tilde{\mathbf{d}}-\mathbf{d})}{1-\bar{\alpha}_t}$. Having the aforementioned KL loss of the variational noising, we obtain the final training objectivity:

$$\mathcal{L} = \sum_{t=2}^{T} \left(\mathcal{L}_t + \mathcal{L}_{vn,t}\right). \tag{15}$$

Algorithm 1: Training Process

Input: The molecular geometry $\mathcal{G}(A, R)$, VAE encoder ϕ_v global equivariant neural networks ϕ_g, local neural networks ϕ_l

```
repeat
    $\mathbf{a}_0 \sim q(\mathbf{a}_0); \mathbf{r}_0 \sim q(\mathbf{r}_0)$
    $t \sim \mathcal{U}(\{1, \ldots, T\})$, $\boldsymbol{\epsilon}^a \sim \mathcal{N}(0, I)$, $\boldsymbol{\epsilon}^r \sim \mathcal{N}(0, I)$
    Shift $\boldsymbol{\epsilon}^r$ to zero COM, $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}^a, \boldsymbol{\epsilon}^r]$
    $\mathcal{G}_t = \sqrt{\bar{\alpha}_t}\mathcal{G}_0 + (1 - \bar{\alpha}_t)\boldsymbol{\epsilon}$
    $\sigma_v, \mu_v = \phi_v(\mathcal{G}_t)$
    Sample $z \sim \mathcal{N}(0, I)$, VAE noise $z_v = \mu_v + \sigma_v^2 z$
    Regulate $z_v$:
    $\mathcal{L}_{vae} = \mathbb{E}_{q_\phi(z_v|\mathcal{G}_t)}(-\mathcal{D}_{KL}(q_\phi(z_v \mid \mathcal{G}_t)) \| p(z)))$
    Prepare global edges $e_g$ and local edges $e_l$
    $\boldsymbol{s}_\theta(\mathcal{G}_t, z_v, t) = \phi_g(\mathcal{G}_t, z_v, t, e_g) + \phi_l(\mathcal{G}_t, z_v, t, e_l)$
    Take gradient descent step on
    $\nabla_\theta \|\boldsymbol{s}_\theta(\mathcal{G}_t, z_v, t) - \nabla_{\mathcal{G}_t} \log q_\sigma(\mathcal{G}_t \mid \mathcal{G}_0)\|^2 + \mathcal{L}_{vn,t}$
until Converged
```

Empirically, if γ in Eq. (13) is ignored during the training phase, the model performs better instead with the simplified objective. Such a simplified objective is equivalent to learning the $\boldsymbol{s}_\theta$ in terms of the gradient of log density of data distribution by sampling the diffused molecule $\mathcal{G}_t$ at a stochastic time step, t.

Algorithm 1 displays the complete training procedure. Each input molecular with a stochastic time step $t \sim \mathcal{U}(1, T)$ is diffused by the noise ϵ. To ensure the invariance of ϵ, we introduce zero center of mass (COM) from Köhler, Klein, and Noé (2020) to achieve invariance for $p(\mathcal{G}_T)$. By extending the approximation of $p(\mathcal{G}_T)$ from a standard Gaussian to an isotropic Gaussian, the ϵ is invariant to rotations and translations around the zero COM.

Sampling

To this point, we have the learned reverse Markov kernels $\boldsymbol{s}_\theta$. The mean of the reverse Gaussian transitions $\boldsymbol{\mu}_\theta$ in Eq. (4) can be calculated. Figure 3 illustrates the sampling phase of MDM. Firstly, the chaotic state $\mathcal{G}_T$ is sampled from $\mathcal{N}(0, I)$ and $\boldsymbol{\mu}_\theta$ is obtained by the dual equivariant encoder. The next less chaotic state $\mathcal{G}_{T-1}$ is generated

Algorithm 2: Sampling Process

```
Input: The learned global equivariant neural networks $\phi_g$, local neural networks $\phi_l$
Output: the molecular coordinates $R$ and atom types $A$
for $t = 1...T$ do
    Sample $\mathcal{G}_t \sim \mathcal{N}(0, I)$
    Sample $\xi \sim \mathcal{N}(0, I)$ if $t > 1$, else $\xi = \mathbf{0}$
    Shift $\mathbf{r}_t$ to zero COM in $\mathcal{G}_t = [r_t, a_t]$
    Prepare global edges $e_g$ and local edges $e_l$
    Sample $z_v \sim \mathcal{N}(0, I)$
    $\boldsymbol{s}_\theta(\mathcal{G}_t, z_v, t) = \phi_g(\mathcal{G}_t, z_v, t, e_g) + \phi_l(\mathcal{G}_t, z_v, t, e_l)$
    $\boldsymbol{\mu}_\theta(\mathcal{G}_t, z_v, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathcal{G}_t + \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{s}_\theta(\mathcal{G}_t, z_v, t)\right)$
    $\mathcal{G}_{t-1} = \boldsymbol{\mu}_\theta(\mathcal{G}_t, z_v, t) + \sigma_t\xi$
end for
return $\mathcal{G}_0$ to obtain $R$ and $A$
```

by $\mathcal{N}(\mathcal{G}_T; \boldsymbol{\mu}_\theta, \sigma_T^2 I)$. The final molecule $\mathcal{G}_0$ is generated by progressively sample $\mathcal{G}_{t-1}$ for T times. The pseudo-code of the sampling process is given in Algorithm 2.

Experiments

In this section, we report the experimental results of the proposed MDM on two benchmark datasets, suggesting MDM significantly outperforms multiple state-of-the-art (SOTA) 3D molecule generation methods. We also conduct conditioned generation experiments to evaluate MDM's ability of generating molecules with desired properties.

Molecular Geometry Generation

数据集

Dataset We adopt QM9 (Ramakrishnan et al. 2014) and GEOM-Drugs (Axelrod and Gomez-Bombarelli 2022) to evaluate the performance of MDM. QM9 contains over 130K molecules, each containing 18 atoms on average. GEOM-Drugs contains 290K molecules, each containing 46 atoms on average.

Baselines and Setup We compare MDM with two one-shot generative models including ENF (Satorras et al. 2021) and EDM (Hoogeboom et al. 2022), and one auto-regressive model G-Schnet (Gebauer, Gastegger, and Schütt 2019). For ENF and EDM, we utilize their published pre-trained models for evaluation on QM9 dataset and retrain both models on GEOM-Drugs dataset. For G-Schnet, we retrain the model on both datasets using its published implementation. In addition, we introduce 'MDM-NV' (**N**o **V**ariational), which excludes the controlling variable z_v and $\mathcal{L}_{vn}$ from MDM, to study their impact.

For all the scenarios in this section, we use 10000 generated samples for evaluation. The molecules generated from QM9 include all kinds of chemical bonds. Since the molecules in the GEOM-Drugs dataset are quite large, and the structure is very complex. It is hard to build the chemical bond via atomic pairwise distances. Hence, we only consider building single bonds to generate the molecules for a fair comparison (Hoogeboom et al. 2022).

评价指标

Metrics We measure the generation performance via four metrics:

- **Validity**: the percentage of the generated molecules that follow the chemical valency rules specified by RDkit;
- **Uniqueness**: the percentage of unique & valid molecules in all the generated samples;
- **Novelty**: the percentage of generated unique molecules that are not in the training set;
- **Stability**: the percentages of the generated molecules that do not include ions.[3]

Results and Analysis

In Table 1, we report the performances of all the models in terms of four metrics on both QM9 and GEOM-Drugs datasets. From Table 1, we can see that the proposed MDM and its variant outperform all the baseline models. For instance, on the QM9 dataset, MDM defeats SOTA (i.e., EDM) by 6.9% in Validity, 4.1% in Uniqueness and 31.1% in Novelty. On GEOM-Drugs dataset, the performance gaps even increase to 30.9% in Validity, 30.4% in Uniqueness, 30.4% in Novelty, and 48.5% in Stability. By outperforming various SOTA generation approaches, the proposed MDM demonstrates its advantage of generating high-quality molecules, especially when the generated molecules contain a larger number of atoms on average (GEOM-Drugs v.s. QM9).

The reason behind such significant improvements is that MDM involves two independent equivalent graph networks to discriminately considers two types of inter-atomic forces, i.e., chemical bonds and van der Waals forces. The huge strength difference between these two types of forces leads to significant distinct local geometries between neighbour atoms with different atomic spacing, especially for larger molecules with more atoms and more complex local geometry structures (GEOM-Drugs v.s. QM9). Such a justification is further supported by the fact that MDM-NV also outperforms EDM, given that both models utilize a diffusion-based framework as the backbone. In contrast to EDM, MDM and its variant MDM-NV successfully capture structural patterns of different atomic spacing and generate molecules that highly follow chemical valency rules (high validity) and fewer ions (high stability).

Besides, we also witness that MDM achieves salient improvements in uniqueness and novelty compared with its variant MDM-NV. Such improvements indicate that the controlling variable z provides thorough explorations in the intermediate steps of the Langevin dynamics and further leads to more diverse generation results. We also conduct ablation studies to investigate the impact of the global and local equivariant kernels. The results indicate that the global and local kernels are indispensable to each other. The absence of one of two modules will greatly impair performance.

Conditional Molecular Generation

Baselines and Setup In this section, we present the conditional molecular generation in which we train our model

[3] The existance of ions indicates that there are two molecular fragments in the generated molecule without bond connection.

Methods	QM9				GEOM-Drugs			
	% V ↑	% U ↑	% N ↑	% S ↑	% V ↑	% U ↑	% N ↑	% S ↑
ENF	41.0	40.1	39.5	24.6	7.68	5.19	5.16	0
G-Schnet	85.9	80.9	57.6	85.6	14.9	14.7	14.6	8.07
EDM	91.7	90.5	59.9	91.1	68.6	68.6	68.6	13.7
MDM-NV	97.8	91.6	80.1	88.6	**99.8**	**99.5**	**99.5**	42.3
MDM	**98.6**	**94.6**	**90.0**	**91.9**	99.5	99.0	99.0	**62.2**

Table 1. The comparison over 10000 generated molecules of MDM and baseline models on molecular geometry generation task. V: Validity, U: Uniqueness, N: Novelty and S: Stability. ↑ means that higher the values, better the performance.

conditioned with six properties Polarizability α, HOMO ϵ_{HOMO}, LUMO ϵ_{LUMO}, HOMO-LUMO gap ϵ_{gap}, Dipole moment μ and C_v on QM9 dataset. Here, we implement the conditioned generation by concatenating the property values c with the atom features to obtain $p_\theta(\mathcal{G}_{t-1} \mid \mathcal{G}_t, c)$.

Following previous work (Hoogeboom et al. 2022), we utilize the property classifier from (Satorras et al. 2021). The training set of QM9 is divided into two halves each containing 50K samples. One half (D_t) is used for classifier training, and the other one (D_e) is utilized for generative model training. Then the classifier ϕ_c evaluates the conditional generated samples by Mean Absolute Error (MAE) of the predicted and true property values.

Baselines Here, we provide several baseline references for comparisons:

- **Naive (Upper-Bond)**: the classifier ϕ_c evaluates on D_e in which the labels are shuffled to predict the molecular properties. This is the upper-bound of possible MAEs (the worst case).
- **#Atoms**: the classifier ϕ_c only depends on the number of atoms to predict the molecular properties on D_e.
- **QM9 (Lower-Bond)**: the classifier directly evaluates on original D_t to predict the molecular properties.

Given the MAE score for samples generated by a generative model, the smaller gap between its MAE score and "QM9 (Lower-Bond)", the better corresponding model fits the data distribution on D_e. Suppose the MAE score of a model outperforms "#Atoms". In that case, it suggests that the model incorporates the targeted property information in the generation process instead of simply generating samples with a specific number of atoms.

Methods	$\alpha \downarrow$	$\epsilon_{\text{gap}} \downarrow$	$\epsilon_{\text{HOMO}} \downarrow$	$\epsilon_{\text{LUMO}} \downarrow$	$\mu \downarrow$	$C_v \downarrow$
Naive (U)	9.013	1.472	0.645	1.457	1.616	6.857
#Atoms	3.862	0.866	0.426	0.813	1.053	1.971
EDM	2.760	0.655	0.356	0.584	1.111	1.101
MDM	1.591	0.044	0.019	0.040	1.177	1.647
QM9 (L)	0.100	0.064	0.039	0.036	0.043	0.040

Table 2. Conditional molecular generation results on QM9 dataset. ↓ means the lower the values, the better the model performs. U and L are upper and lower bound, respectively.

Results Table 2 presents the targeted generation results. In particular, MDM surpasses "Naive", "#Atoms", and EDM in almost all the properties except for μ and C_v, indicating that MDM performs better than EDM in incorporating the targeted property information into the generated samples themselves beyond the number of features. Moreover, we notice that the MAE of MDM on gaps and HOMO is even slightly lower than "QM9 (Lower-Bound)". This phenomenon may be caused by the slight distribution difference between D_e and D_t. At the same time, it indicates that MDM can well fit the distribution of D_e and generate high-quality molecules with targeted properties. Empirically, the conditional generation will not hurt the quality of the generated molecules in terms of validity, uniqueness, novelty, and stability.

Apart from the quantitative analysis, we also provide case studies to analyze the effect of applying different values of the properties to conditional generation. Here, we adopt the property Polarizability α for demonstration.

Conclusion

In this study, we propose a novel diffusion model MDM to generate 3D molecules from scratch. MDM augments interatomic linkages that are not in the 3D point cloud representation of molecules and proposes separated equivariant encoders to capture the interatomic forces of different strengths. In addition, we introduce a controlling variable in both diffusion and reverse processes to improve generation diversity. Comprehensive experiments demonstrate that MDM exceeds previous SOTA models by a non-trivial margin and can generate molecules with desired properties.

Acknowledgments

This research was substantially sponsored by the research projects (Grant No. 32170654 and Grant No. 32000464) supported by the National Natural Science Foundation of China and was substantially supported by the Shenzhen Research Institute, City University of Hong Kong. This project was substantially funded by the Strategic Interdisciplinary Research Grant of City University of Hong Kong (Project No. 2021SIRG036). The work described in this paper was partially supported by the grants from City University of Hong Kong (CityU 9667265).

References

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

Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; and Song, L. 2018. Syntax-Directed Variational Autoencoder for Structured Data. In *International Conference on Learning Representations*.

De Cao, N.; and Kipf, T. 2018. MolGAN: An implicit generative model for small molecular graphs. *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*.

Garcia Satorras, V.; Hoogeboom, E.; Fuchs, F.; Posner, I.; and Welling, M. 2021. E (n) Equivariant Normalizing Flows. *Advances in Neural Information Processing Systems*, 34: 4181–4192.

Gebauer, N.; Gastegger, M.; and Schütt, K. 2019. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32.

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

Grisoni, F.; Moret, M.; Lingwood, R.; and Schneider, G. 2020. Bidirectional molecule generation with recurrent neural networks. *Journal of chemical information and modeling*, 60(3): 1175–1183.

Guan, J.; Qian, W. W.; qiang liu; Ma, W.-Y.; Ma, J.; and Peng, J. 2022. Energy-Inspired Molecular Conformation Optimization. In *International Conference on Learning Representations*.

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

Hoogeboom, E.; Satorras, V. G.; Vignac, C.; and Welling, M. 2022. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, 8867–8887. PMLR.

Jin, W.; Barzilay, R.; and Jaakkola, T. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, 2323–2332. PMLR.

Köhler, J.; Klein, L.; and Noé, F. 2020. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, 5361–5370. PMLR.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2021. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *International Conference on Learning Representations*.

Liu, Q.; Lee, J.; and Jordan, M. 2016. A kernelized Stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, 276–284. PMLR.

Luo, Y.; Yan, K.; and Ji, S. 2021. Graphdf: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*, 7192–7203. PMLR.

Mansimov, E.; Mahmood, O.; Kang, S.; and Cho, K. 2019. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1): 1–13.

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

Satorras, V. G.; Hoogeboom, E.; Fuchs, F. B.; Posner, I.; and Welling, M. 2021. E(n) Equivariant Normalizing Flows. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

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

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

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.

Xu, M.; Wang, W.; Luo, S.; Shi, C.; Bengio, Y.; Gomez-Bombarelli, R.; and Tang, J. 2021. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, 11537–11547. PMLR.

Xu, M.; Yu, L.; Song, Y.; Shi, C.; Ermon, S.; and Tang, J. 2022. GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation. In *International Conference on Learning Representations*.

Zang, C.; and Wang, F. 2020. MoFlow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 617–626.