中國人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation

Minkai Xu, Lantao Yu, Yang Song, Chence Shi,
Stefano Ermon, Jian Tang

(ICLR 2022)

Fanmeng Wang

2022-11-17

Outline

- Introduction
- Diffusion Model
- GeoDiff Method
- Experiment
- Conclusion

- Introduction
- Diffusion Model
- GeoDiff Method
- Experiment
- Conclusion

Introduction

■ Molecular Representations

➢ The traditional representations for molecules include **1D Representation (i.e. SMILEs)** and **2D Representation (i.e. Molecular Graph)**.

1D Representation

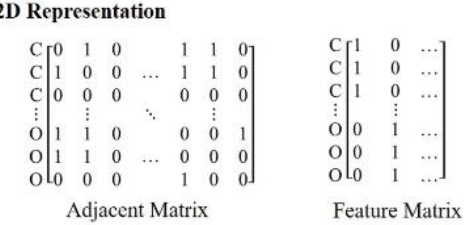
SMILEs:
CC1=C(C=C(C=C1)NC(=O)C2=CC=C(C=C2)CN3CCN(CC3)C)NC4=NC=CC(=N4)C5=CN=CC=C5.CS(=O)(=O)O

ECFP:
216545222: 2, 337875000: 1, 353395765: 2,
368350036: 1, 847957139: 1, 847961216: 2,
…
3275683399: 1, 3918336191: 2, 3975275337: 1]

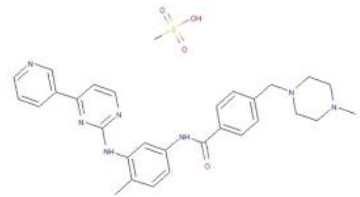
MACCS:
000000000.....10100110101111111
1110

Mathematical Representation:
Differential geometry
Algebraic graph
Algebraic topology

2D Representation



2D Molecular Graph



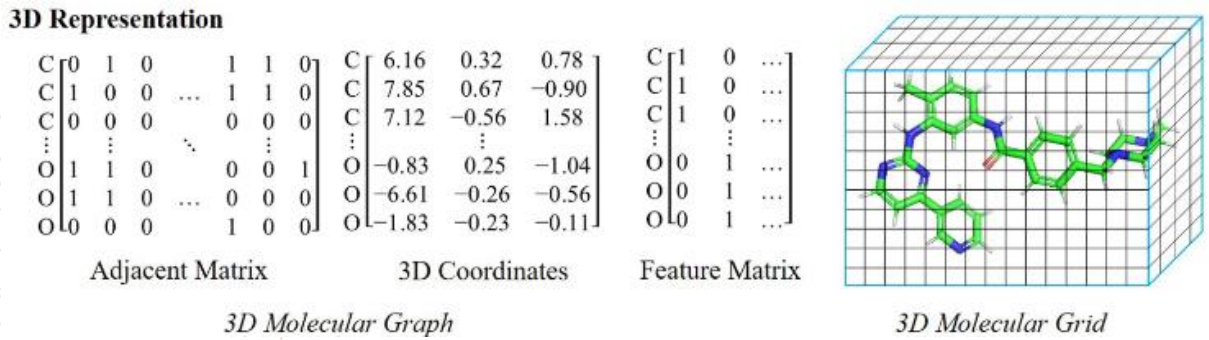
2D Molecular Image

➢ A more intrinsic and informative representation for molecules is the **3D Representation**, also known as **molecular conformation**, where atoms are represented as their Cartesian coordinates.

Introduction

■ Molecular Representations

- The **3D Representation** determine the **biological and physical properties** of molecules and hence play a key role in many applications such as computational drug and material design.
- Unfortunately, how to predict **stable molecular conformation** remains a challenging problem.



Introduction

- **Exciting Methods for Molecular Conformation Generation**
  - **Traditional methods** such as molecular dynamics (MD) and Markov chain Monte Carlo (MCMC)
    - They are very computationally **expensive**, especially for large molecules.
  - **Some Deep generative models** such as variational autoencoders (VAEs) and flow-based models
    - As all these approaches seek to indirectly model the intermediate geometric variables, they have **inherent limitations** in either training or inference process.
- An ideal solution would still be **directly modeling the atomic coordinates** and at the same time taking the **roto-translational invariance property** into account.

Introduction

- **GeoDiff: A principled probabilistic framework based on denoising diffusion models**
  - Main features:
    - It directly acts on the **atomic coordinates** and entirely bypasses the usage of intermediate elements for both training and inference.
    - It combines **diffusion model** and **3D conformation information** to ensure the **roto-translational invariance property** of molecular conformation, while achieving **state-of-the-art** in 3D conformation generation.
    - The **reverse diffusion process** is parameterized by a equivariant graph neural network.

EGNN + Diffusion Model

atomic coordinates

- Introduction
- **Diffusion Model**
- GeoDiff Method
- Experiment
- Conclusion

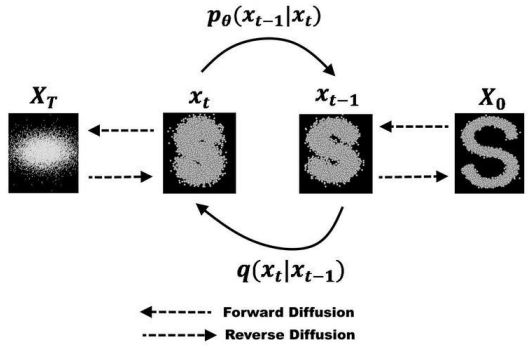
Diffusion Model

■ Basic compositions

- Forward Diffusion Process： $X_0 \rightarrow X_T$ gradually convert the **original distribution** to **Standard Gaussian distribution** by adding Gaussian noise.
- Reverse Diffusion Process： $X_T \rightarrow X_0$ gradually recover the **original distribution** from the **Standard Gaussian distribution** by denoising neural network.

■ Note

- Only **Reverse Diffusion Process** can be learned and $X_{0:T}$ have same dimension.
- We can also use **other noise distribution** to replace Standard Gaussian distribution.



Diffusion Model

■ Forward Diffusion Process

➢ Forward Diffusion Process is a **Markov chain** with fixed parameters that gradually convert the **original distribution** to the **Standard Gaussian distribution**.

$$q(x_{1:T}|x_0) = \frac{q(x_{0:T})}{q(x_0)} = \prod_{t=1}^{T} \frac{q(x_{0:t})}{q(x_{0:t-1})} = \prod_{t=1}^{T} q(x_t|x_{0:t-1}) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

Markov chain: $q(x_t|x_{0:t-1}) = q(x_t|x_{t-1})$

The t-th process that convert x_{t-1} to x_t can be expressed as:

$$q(x_t|x_{t-1}) = \boldsymbol{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I})$$

It means that x_{t-1} -> x_t is a **Gaussian distribution transformation** with $\sqrt{1-\beta_t}x_{t-1}$ as the mean and β_t as the variance.

➢ **Note:** β_t is a constant between 0 and 1, so the Forward Diffusion Process contains no learnable parameters.

Diffusion Model

■ Forward Diffusion Process

➢ The process that convert x_{t-1} to x_t can be further expressed as

$$q(x_t|x_{t-1}) = \boldsymbol{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I})$$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_1$$

$$= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}z_1$$

$$= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}z_2) + \sqrt{1-\alpha_t}z_1$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}z_2 + \sqrt{1-\alpha_t}z_1$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\,\bar{z}_2$$

$$= \dots$$

$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\bar{z}_t$$

Reparameterization trick
If $z \sim \boldsymbol{N}(z; \mu_\theta, \sigma_\theta^2 \boldsymbol{I})$, $\varepsilon \sim \boldsymbol{N}(\varepsilon; 0, \boldsymbol{I})$
then $z = \mu_\theta + \sigma_\theta \odot \varepsilon$

Parameter definition
$\alpha_t = 1 - \beta_t$
$z_1, z_2, \bar{z}_2, \dots, z_t, \bar{z}_t \sim \boldsymbol{N}(0, \boldsymbol{I})$
$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t$

Additivity of independent Gaussian distributions

➢ Therefore, we can get any x_t as long as we know the initial data distribution x_0 and constant β_t since $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\bar{z}_t$

Diffusion Model

■ Reverse Diffusion Process

- In the Forward Diffusion Process, we gradually add standard Gaussian noise and the original distribution will be converted to standard Gaussian distribution finally when T is big enough:

$$x_T = \sqrt{\overline{\alpha_T}}x_0 + \sqrt{1-\overline{\alpha_T}}\,\overline{z_T}$$

$$x_T \sim \boldsymbol{N}(0, \boldsymbol{I})$$

$\overline{\alpha_T} = \prod_{i=1}^{T} \alpha_t \quad \overline{z_T} \sim \boldsymbol{N}(0, \boldsymbol{I})$

Since α_t is a constant between 0 and 1

$\overline{\alpha_T}$ -> 0 while T -> +∞

- In the Reverse Diffusion Process, we try to recover the **original distribution** from **Standard Gaussian distribution** by denoising neural network.
- We assume that this process is also a **Markov chain** with learnable parameters.

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t,t), \sum_\theta(x_t,t))$$

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t,t)\right)$$

In DDPM, we just use $\frac{1-\overline{\alpha_{t-1}}}{1-\overline{\alpha_t}}\beta_t$ to replace it

Noise predicted by denoising network

Diffusion Model

■ Reverse Diffusion Process

➢ Accoding to the prior probability of x_{t-1} -> x_t in the Forward Diffusion Process

$$q(x_t|x_{t-1}) = \boldsymbol{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\boldsymbol{I}) = \boldsymbol{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t\boldsymbol{I})$$

We can easily get the posterior probability of x_t -> x_{t-1} as follows

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t, x_{t-1}, x_0)}{q(x_t, x_0)}$$

$$= \frac{q(x_t|x_{t-1}, x_0)\, q(x_{t-1}|x_0) q(x_0)}{q(x_t|x_0) q(x_0)}$$

$$= \frac{q(x_t|x_{t-1}, x_0)\, q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$= \frac{q(x_t|x_{t-1})\, q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

Therefore, the reverse process x_t -> x_{t-1} can be expressed as the combination of forward process x_{t-1} -> x_t、 x_0 -> x_{t-1}、 x_0 -> x_t.

Diffusion Model

■ Reverse Diffusion Process

➢ Since

$$q(x_t|x_{t-1}) = \boldsymbol{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\boldsymbol{I}) = \boldsymbol{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t\boldsymbol{I})$$

$$= \frac{1}{\sqrt{2\pi}\beta_t}\exp\left(-\frac{1}{2}\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t}\right)$$

$$\propto \exp\left(-\frac{1}{2}\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t}\right)$$

and $q(x_{t-1}|x_0) \propto \exp\left(-\frac{1}{2}\frac{(x_{t-1}-\sqrt{\overline{\alpha_{t-1}}}\,x_0)^2}{1-\overline{\alpha_{t-1}}}\right)$ $q(x_t|x_0) \propto \exp\left(-\frac{1}{2}\frac{(x_t-\sqrt{\overline{\alpha_t}}\,x_0)^2}{1-\overline{\alpha_t}}\right)$

Therefore, the posterior probability of x_t -> x_{t-1} can be further expressed as

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1})\,q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1}-\sqrt{\overline{\alpha_{t-1}}}\,x_0)^2}{1-\overline{\alpha_{t-1}}} - \frac{(x_t-\sqrt{\overline{\alpha_t}}\,x_0)^2}{1-\overline{\alpha_t}}\right)\right)$$

Diffusion Model

■ Reverse Diffusion Process

➢ The posterior probability of x_t -> x_{t-1} can be further expressed as

$$q(x_{t-1}|x_t, x_0) \propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1} + C(x_t, x_0)\right)\right)$$

Assume

$$q(x_{t-1}|x_t, x_0) = N(\bar{\mu}_t, \bar{\beta}_t)$$

Then

$$q(x_{t-1}|x_t, x_0) \propto \exp\left(-\frac{1}{2}\frac{(x_{t-1} - \bar{\mu}_t)^2}{\bar{\beta}_t}\right)$$

$$= exp(-\frac{1}{2}(\frac{1}{\bar{\beta}_t}{x_{t-1}}^2 - 2\frac{\bar{\mu}_t}{\bar{\beta}_t}x_{t-1} + \frac{{\bar{\mu}_t}^2}{\bar{\beta}_t}))$$

$$\frac{1}{\bar{\beta}_t} = \frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \qquad \frac{\bar{\mu}_t}{\bar{\beta}_t} = \frac{\sqrt{\alpha_t}}{\beta_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0$$

$$\bar{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\bar{\mu}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\beta_t x_0$$

Diffusion Model

How to train Diffusion Model ?

- The loss function can be set as

$$\mathcal{L} = \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)]$$

Since

$$-\log p_\theta(x_0) \leq -\log p_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0))$$

KL Divergence is non-negative

$$= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right]$$

$$p_\theta(x_{1:T}|x_0) = \frac{p_\theta(x_{0:T})}{p_\theta(x_0)}$$

$$= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \underbrace{\log p_\theta(x_0)}_{\text{与}q\text{无关}}\right]$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right].$$

$$\mathcal{L}_{VLB} = \mathbb{E}_{q(x_0)}\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]\right) = \mathbb{E}_{q(x_{0:T})}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right] \geq \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)]$$

Therefore, we just need to minimize L_{VLB}

Diffusion Model

■ How to train Diffusion Model ?

➢ Since L_{VLB} can be further expressed as

$$
\begin{aligned}
L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
&= \mathbb{E}_q\left[\log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^{T}\log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log \left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\cdot\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^{T}\log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T}\log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] \\
&= \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T}\underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]
\end{aligned}
$$

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \le t \le T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}
$$

Diffusion Model

■ How to train Diffusion Model ?

➢ Therefore, we can just consider L_t

$$q(x_{t-1}|x_t, x_0) = \boldsymbol{N}(\bar{\mu}_t, \bar{\beta}_t)$$

$$\bar{\beta}_t = \frac{1-\overline{\alpha_{t-1}}}{1-\overline{\alpha_t}}\beta_t$$

$$\bar{\mu}_t = \frac{\sqrt{\alpha_t}(1-\overline{\alpha_{t-1}})}{1-\overline{\alpha_t}}x_t + \frac{\sqrt{\overline{\alpha_{t-1}}}}{1-\overline{\alpha_t}}\beta_t x_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha_t}}}\bar{z}_t)$$

$$L_{\text{VLB}} = L_T + L_{T-1} + \cdots + L_0$$
$$\text{where } L_T = D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))$$
$$L_t = D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \le t \le T-1$$
$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

KL Divergence of Multivariate Gaussian function

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t))$$

In DDPM, we just use $\frac{1-\overline{\alpha_{t-1}}}{1-\overline{\alpha_t}}\beta_t$ to replace it

Noise predicted by denoising network

$$L_t = \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)||^2\right]$$
$$= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\frac{1}{\sqrt{\bar{a}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}\bar{z}_t) - \frac{1}{\sqrt{\bar{a}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}z_\theta(x_t,t))||^2\right]$$
$$= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t - z_\theta(x_t,t)||^2\right]$$
$$= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\bar{z}_t, t)||^2\right].$$

Diffusion Model

How to train Diffusion Model ?

$$
\begin{aligned}
L_t &= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{1}{2||\Sigma_\theta(x_t,t)||_2^2}||\frac{1}{\sqrt{\bar{a}_t}}(x_t-\frac{\beta_t}{\sqrt{1-\bar{a}_t}}\bar{z}_t)-\frac{1}{\sqrt{\bar{a}_t}}(x_t-\frac{\beta_t}{\sqrt{1-\bar{a}_t}}z_\theta(x_t,t))||^2\right] \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t-z_\theta(x_t,t)||^2\right] \\
&= \mathbb{E}_{x_0,\bar{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t||\Sigma_\theta||_2^2)}||\bar{z}_t-z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\bar{z}_t,t)||^2\right].
\end{aligned}
$$

Remove constant term

$$L_t^{simple} = E_{x_0,z}[||z_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}z_t,t)||^2]$$

- Therefore, we just need to minimize $||\bar{z}_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\bar{z}_t,t)||$ in the training process.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1,\dots,T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\|\boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)\right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
2: **for** $t = T,\dots,1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ if $t>1$, else $\mathbf{z}=\mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right)+\sigma_t\mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Diffusion Model

■ Two important problems when using Diffusion Model

- How to design a model which can accurately **predict the noise** z_θ?
- How to combine the **Diffuison Model** with some characteristics of real problems (e.g., **equivariance**)?
- In fact, the two problem are just **the core of GeoDiff proposed by this paper.**
- GeoDiff use a **equivariant graph neural network** to **predict the noise** and ensure the **roto-translational invariance property** of molecular conformation.

- Introduction
- Diffusion Model
- GeoDiff Method
- Experiment
- Conclusion

DeoDiff Method

■ Notations

- A molecule with n atoms is represented as an undirected $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$
- $\mathcal{V} = \{v_i\}_{i=1}^{n}$ is the set of vertices representing
- $\mathcal{E} = \{e_{ij} \mid (i,j) \subseteq |\mathcal{V}| \times |\mathcal{V}|\}$ is the set of edges representing inter-atomic bonds
- Each node $v_i \in \mathcal{V}$ describes the atomic attributes, e.g., the element type
- Each edge $e_{ij} \in \mathcal{E}$ describes the corresponding connection between v_i and v_j, and is labeled with its chemical type
- In addition, we also assign the unconnected edges with a virtual type
- For the geometry, each atom in V is embedded by **a coordinate vector** $c \in R^3$ into the 3-dimensional space, and the full set of positions (i.e., the **conformation**) can be represented as a matrix $\mathcal{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n] \in \mathbb{R}^{n \times 3}$

DeoDiff Method

■ Problem Definition

- The task of molecular conformation generation is a **conditional generative problem**
- Our goal is **learning a generative model** $p_\theta(\mathcal{C}|\mathcal{G})$ so that **generating stable conformations for a provided graph** $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$

■ Equivariance

- Equivariance is ubiquitous in machine learning for atomic systems
- Formally, a function $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ is equivariant w.r.t a group G if:

$$\mathcal{F} \circ T_g(x) = S_g \circ \mathcal{F}(x)$$

where T_g and S_g are transformations for an element $g \in G$, acting on the vector spaces X and Y, respectively

- In this work, we consider the SE(3) group, i.e., the group of rotation, translation in 3D space

DeoDiff Method

■ DeoDiff Formulation: A simple variant of Diffusion Model

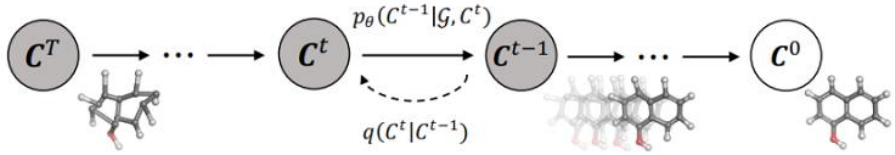


Figure 1: Illustration of the diffusion and reverse process of GEODIFF. For diffusion process, noise from fixed posterior distributions $q(\mathcal{C}^t|\mathcal{C}^{t-1})$ is gradually added until the conformation is destroyed. Symmetrically, for generative process, an initial state $\mathcal{C}^T$ is sampled from standard Gaussian distribution, and the conformation is progressively refined via the Markov kernels $p_\theta(\mathcal{C}^{t-1}|\mathcal{G},\mathcal{C}^t)$.

$\mathcal{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n] \in \mathbb{R}^{n\times 3}$
Molecular conformation
(i.e.,the position of all atoms)

$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$
Undirected Molecular Graph

➢ Diffuison Process

$$q(\mathcal{C}^{1:T}|\mathcal{C}^0) = \prod_{t=1}^{T} q(\mathcal{C}^t|\mathcal{C}^{t-1}), \quad q(\mathcal{C}^t|\mathcal{C}^{t-1}) = \mathcal{N}(\mathcal{C}^t; \sqrt{1-\beta_t}\mathcal{C}^{t-1}, \beta_t I).$$

➢ Reverse Diffusion Process

$$p_\theta(\mathcal{C}^{0:T-1}|\mathcal{G},\mathcal{C}^T) = \prod_{t=1}^{T} p_\theta(\mathcal{C}^{t-1}|\mathcal{G},\mathcal{C}^t), \quad p_\theta(\mathcal{C}^{t-1}|\mathcal{G},\mathcal{C}^t) = \mathcal{N}(\mathcal{C}^{t-1}; \mu_\theta(\mathcal{G},\mathcal{C}^t,t), \sigma_t^2 I).$$

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

➢ Loss Function

$$\mathcal{L}_{\text{ELBO}} = \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\{\mathcal{C}^0,\mathcal{G}\}\sim q(\mathcal{C}^0,\mathcal{G}),\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(\mathcal{G},\mathcal{C}^t,t)\|_2^2\right]$$

ϵ is the **Gaussian noise** that used in Diffusion process
ϵ_θ is the **predicted noise** by a equivariant graph neural network

DeoDiff Method

■ DeoDiff Formulation: A simple variant of Diffusion Model

➢ The design of equivariant graph neural network $\epsilon_\theta(\mathcal{G}, \mathcal{C}^t, t)$

- We draw inspirations from recent equivariant networks (Thomas et al., 2018; Satorras et al., 2021b) to design an **equivariant convolutional layer**, named **graph field network** (GFN)
- In the l-th layer, GFN take **node embeddings** $h^l \in R^{n\times b}$ **and corresponding coordinate embeddings** $x^l \in R^{n\times 3}$ as **inputs**, and **outputs** h^{l+1} and x^{l+1}

$$\mathbf{m}_{ij} = \Phi_m\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, e_{ij}; \theta_m\right)$$
$$\mathbf{h}_i^{l+1} = \Phi_h\left(\mathbf{h}_i^l, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}; \theta_h\right)$$
$$\mathbf{x}_i^{l+1} = \sum_{j\in\mathcal{N}(i)} \frac{1}{d_{ij}}\left(\mathbf{c}_i - \mathbf{c}_j\right)\Phi_x\left(\mathbf{m}_{ij}; \theta_x\right)$$

- Initial embeddings h^0 are combinations of atom and timestep embeddings and x^0 are atomic coordinates
- Φ are feed-forward networks
- d_{ij} denotes interatomic distances
- c_i denotes the position of node i at time t
- $N(i)$ denotes the neighborhood of node i

DeoDiff Method

■ DeoDiff Formulation: A simple variant of Diffusion Model

➢ The design of equivariant graph neural network $\epsilon_\theta(\mathcal{G}, \mathcal{C}^t, t)$

- In the l-th layer, GFN take **node embeddings** $h^l \in R^{n\times b}$ **and corresponding coordinate embeddings** $x^l \in R^{n\times 3}$ as **inputs**, and **outputs** h^{l+1} and x^{l+1}

$$\mathbf{m}_{ij} = \Phi_m\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, e_{ij}; \theta_m\right)$$
$$\mathbf{h}_i^{l+1} = \Phi_h\left(\mathbf{h}_i^l, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}; \theta_h\right)$$
$$\mathbf{x}_i^{l+1} = \sum_{j\in\mathcal{N}(i)} \frac{1}{d_{ij}}\left(\mathbf{c}_i - \mathbf{c}_j\right)\Phi_x\left(\mathbf{m}_{ij}; \theta_x\right)$$

  - Initial embeddings h^0 are combinations of atom and timestep embeddings and x^0 are atomic coordinates
  - Φ are feed-forward networks
  - d_{ij} denotes interatomic distances
  - c_i denotes the position of node i at time t
  - $N(i)$ denotes the neighborhood of node i
- Parameterizing $\epsilon_\theta(\mathcal{G}, \mathcal{C}^t, t)$ as a composition of L GFN layers, and take the x_L after L updates as the **output**.
- Therefore, the noise vector field ϵ_θ is **SE(3) equivariant**
- Further, **Markov Kernel** $p(\mathcal{C}^{t-1}|\mathcal{G}, \mathcal{C}^t)$ is also **SE(3) equivariant**

DeoDiff Method

■ DeoDiff Formulation: A simple variant of Diffusion Model

➢ Equivariant reverse process

- Since Markov Kernel $p(\mathcal{C}^{t-1}|\mathcal{G},\mathcal{C}^t)$ is SE(3) equivariant, we just need to make sure that $p(C^T)$ is **invariant initial density**.
- We borrow the idea from Köhler et al. (2020) to consider systems with **zero center of mass** (CoM), termed CoM-free systems.
- We define $p(C^T)$ as a "**CoM-free standard density**" $\hat{p}(C)$, built upon an isotropic normal density $p(C)$
- For **evaluating** the likelihood $\hat{p}(C)$, we can firstly translate C to zero CoM and then calculate $p(C)$
- For **sampling** from $\hat{p}(C)$, we can first sample from $p(C)$ and then move the CoM to zero.

DeoDiff Method

■ DeoDiff Formulation: A simple variant of Diffusion Model

➢ Sampling

Algorithm 1 Sampling Algorithm of GEODIFF.

Input: the molecular graph $\mathcal{G}$, the learned reverse model ϵ_θ.
Output: the molecular conformation $\mathcal{C}$.

1: Sample $\mathcal{C}^T \sim p(\mathcal{C}^T) = \mathcal{N}(0, I)$
2: **for** $s = T, T-1, \cdots, 1$ **do**
3: Shift $\mathcal{C}^s$ to zero CoM
4: Compute $\mu_\theta(\mathcal{C}^s, \mathcal{G}, s)$ from $\epsilon_\theta(\mathcal{C}^s, \mathcal{G}, s)$ using equation 4
5: Sample $\mathcal{C}^{s-1} \sim \mathcal{N}(\mathcal{C}^{s-1}; \mu_\theta(\mathcal{C}^s, \mathcal{G}, s), \sigma_t^2 I)$
6: **end for**
7: **return** $\mathcal{C}^0$ as $\mathcal{C}$

To make sure initial density invariant

$$\mu_\theta(\mathcal{C}^t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathcal{C}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathcal{G}, \mathcal{C}^t, t) \right), \quad (4)$$

- Introduction
- Diffusion Model
- GeoDiff Method
- Experiment
- Conclusion

Experiment

■ Conformation Generation

- The task aims to measure both quality and diversity of generated conformations by different models.
- **Evaluation metrics**
  - We evaluate 4 metrics built upon **root-mean-square deviation** (RMSD), which is defined as the normalized **Frobenius norm** of two atomic coordinates matrices. **COV-R MAT-R COV-P MAT-P**
  - RMSD can be seen as a metrics of the distance between two atomic

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \qquad \text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2}$$

matrices, after alignment by Kabsch algorithm (Kabsch, 1976). Formally, let S_g and S_r denote the sets of generated and reference conformers respectively, then the **Cov**erage and **Mat**ching metrics (Xu et al., 2021a) following the conventional *Recall* measurement can be defined as:

$$\text{COV-R}(S_g, S_r) = \frac{1}{|S_r|}\left|\left\{\mathcal{C} \in S_r \,|\, \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}) \leq \delta, \hat{\mathcal{C}} \in S_g\right\}\right|, \tag{10}$$

$$\text{MAT-R}(S_g, S_r) = \frac{1}{|S_r|}\sum_{\mathcal{C} \in S_r} \min_{\hat{\mathcal{C}} \in S_g} \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}), \tag{11}$$

where δ is a pre-defined threshold. The other two metrics COV-P and MAT-P inspired by *Precision* can be defined similarly but with the generated and reference sets exchanged. In practice, S_g is set as twice of the size of S_r for each molecule. Intuitively, the COV scores measure the percentage of structures in one set covered by another set, where covering means the RMSD between two conformations is within a certain threshold δ. By contrast, the MAT scores measure the average RMSD of conformers in one set with its closest neighbor in another set. In general, higher COV rates or lower MAT score suggest that more realistic conformations are generated. Besides, the *Precision*

Experiment

■ Conformation Generation

➢ Result

- Since RDKIT involves **an additional empirical force field** (FF) to optimize the structure, we follow them to also combine GEODIFF with FF to yield a more fair comparison when compared with RDKIT.

Table 1: Results on the **GEOM-Drugs** dataset, without FF optimization.

Models	COV-R (%) ↑		MAT-R (Å) ↓		COV-P (%) ↑		MAT-P (Å) ↓	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
CVGAE	0.00	0.00	3.0702	2.9937	-	-	-	-
GRAPHDG	8.27	0.00	1.9722	1.9845	2.08	0.00	2.4340	2.4100
CGCF	53.96	57.06	1.2487	1.2247	21.68	13.72	1.8571	1.8066
CONFVAE	55.20	59.43	1.2380	1.1417	22.96	14.05	1.8287	1.8159
GEOMOL	67.16	71.71	1.0875	1.0586	-	-	-	-
CONFGF	62.15	70.93	1.1629	1.1596	23.42	15.52	1.7219	1.6863
GEODIFF-A	88.36	96.09	0.8704	0.8628	60.14	61.25	1.1864	1.1391
GEODIFF-C	**89.13**	**97.88**	**0.8629**	**0.8529**	**61.47**	**64.55**	**1.1712**	**1.1232**

Table 2: Results on the **GEOM-Drugs** dataset, with FF optimization.

Models	COV-R (%) ↑		MAT-R (Å) ↓		COV-P (%) ↑		MAT-P (Å) ↓	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
RDKIT	60.91	65.70	1.2026	1.1252	72.22	88.72	1.0976	0.9539
GEODIFF + FF	**92.27**	**100.00**	**0.7618**	**0.7340**	**84.51**	**95.86**	**0.9834**	**0.9221**

Experiment

■ Conformation Generation

➢ Result

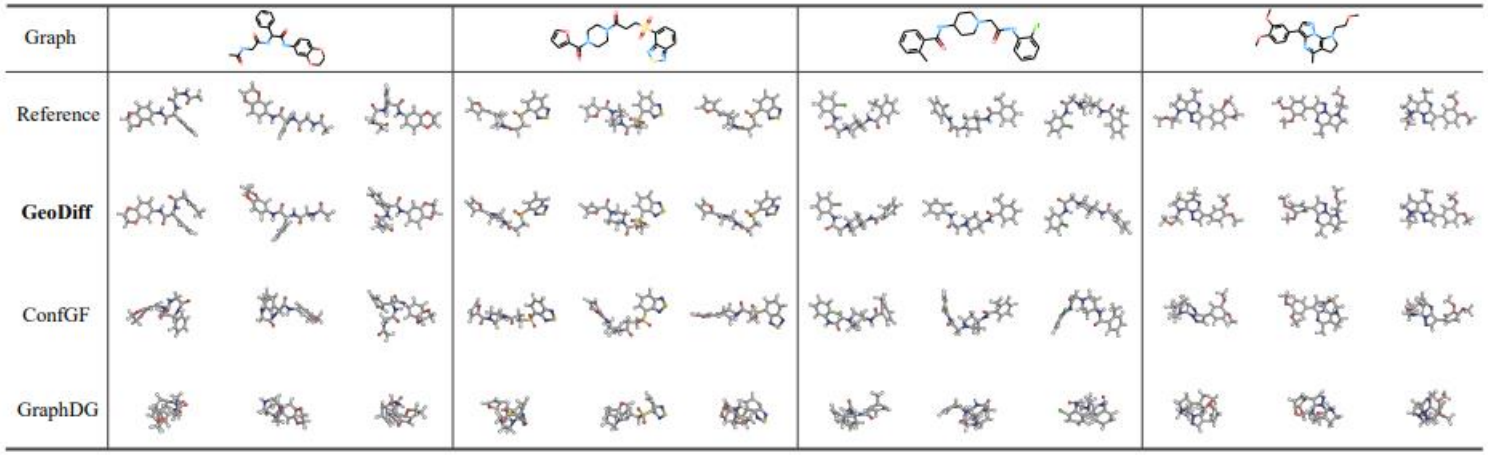


Figure 2: Examples of generated structures from Drugs dataset. For every model, we show the conformation best-aligned with the ground truth. More examples are provided in Appendix E.

- The results demonstrate the **superior capacity** of GEODIFF to model the multi modal distribution, and generative both **accurate and diverse** conformations

- Introduction
- Diffusion Model
- GeoDiff Method
- Experiment
- Conclusion

Conclusion

- This paper proposes GeoDiff, a novel **probabilistic model** for generating molecular conformations.
- GeoDiff combines **denoising diffusion models** with **geometric representations**, where we parameterize the **reverse generative dynamics** as a Markov chain, and novelly impose **roto-translational invariance** into the density with **equivariant Markov kernels**.
- GeoDiff is **competitive** with the existing **state-of-the-art** models.
- However, GeoDiff only considers the **atomic coordinates** of molecular conformations, the **atomic types** and **chemical bonds** of molecular conformations are all ignored.

Thank You for listening!

2022-11-17