

APPENDIX

A DIFFUSION MODELS

Given a data point $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and a variance schedule β_1, \dots, β_T that controls the amount of noise added at each timestep t , the diffusion process or forward process gradually add Gaussian noise to the data point \mathbf{x} :

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (10)$$

where $\beta_{1:T}$ are chosen such that data point \mathbf{x} will approximately converge to standard Gaussian, *i.e.*, $q(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$. Generally, the diffusion process q has no trainable parameters. The denoising process or reverse process aims at learning a parameterized generative process, which incrementally denoise the noisy variables $\mathbf{x}_{T:1}$ to approximate restore the data point \mathbf{x}_0 in the original data distribution:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \quad (11)$$

where the initial distribution $p(\mathbf{x}_t)$ is sampled from standard Gaussian noise $\mathcal{N}(0, \mathbf{I})$. The means μ_θ typically are neural networks such as U-Nets for images or Transformers for text.

The forward process is $q(\mathbf{x}_{1:T} | \mathbf{x}_0)$ is an approximate posterior to the Markov chain, and the reverse process $p_\theta(\mathbf{x}_{0:T})$ is optimized by a variational lower bound on the negative log-likelihood of \mathbf{x}_0 by:

$$\mathbb{E}_q[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] \quad (12)$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t=1}^T \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right], \quad (13)$$

which is \mathcal{L}_{vib} . To efficiently train the diffusion models, further improvements come to term \mathcal{L}_{vib} by variance reduction, and thereby Eq. (12) is rewritten as:

$$\mathcal{L}_{\text{vib}} = \mathbb{E}_q[\mathcal{L}_T + \sum_{t=2}^T \mathcal{L}_t + \mathcal{L}_0] \quad (14)$$

where $\mathcal{L}_T = \log \frac{q(\mathbf{x}_T | \mathbf{x}_0)}{p_\theta(\mathbf{x}_T)}$, which models the distance between a standard normal distribution and the final latent variable $q(\mathbf{x}_T | \mathbf{x}_0)$, since the approximate posterior q has no learnable parameters, so \mathcal{L}_T is a constant during training and can be ignored. $\mathcal{L}_0 = -\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)$ models the likelihood of the data given \mathbf{x}_0 , which is close to zero and ignored as well if $\beta_0 \approx 0$ and \mathbf{x}_0 is discrete.

\mathcal{L}_t in Eq. (14) is the loss for the reverse process and is given by:

$$\mathcal{L}_t = \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_0, \mathbf{x}_t)}{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}. \quad (15)$$

While in this formulation the neural network directly predicts $\hat{\mathbf{x}}_0$, (Ho et al., 2020) found that optimization is easier when predicting the Gaussian noise instead. Intuitively, the network is trying to predict which part of the observation \mathbf{x}_t is noise originating from the diffusion process, and which part corresponds to the underlying data point \mathbf{x}_0 . Then sampling $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ is to compute

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\sqrt{\beta_t}}{\sqrt{1 - \alpha_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}, \quad (16)$$

where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. And thereby $\mathcal{L}_{\text{DM}} := \mathcal{L}_t$ is simplified to:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t} [w(t) \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2] \quad (17)$$

where $w(t) = \frac{\beta_t}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)}$ is the reweighting term and could be simply set as 1 with promising sampling quality, and $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$.

B MODEL ARCHITECTURE DETAILS

B.1 EQUIVARIANT MASKED AUTOENCODER

In this work, **EAAE** considers visible molecular structural geometries as point clouds, without specifying the connecting bonds. Therefore, in practice, we take the point clouds as fully connected graph \mathcal{G} and model the interactions between all atoms $v_i \in \mathcal{V}$. Each node v_i is embedded with coordinates $\mathbf{x}_i \in \mathbb{R}^3$ and atomic features $\mathbf{h}_i \in \mathbb{R}^d$. Then, **EAAE** are composed of multiple Equivariant Convolutional Layers, and each single layer is expressed as (Satorras et al., 2021):

$$\begin{aligned} d_{ij}^2 &= \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, \\ \mathbf{m}_{i,j} &= \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, d_{ij}^2, a_{ij}), \\ \mathbf{x}_i^{l+1} &= \mathbf{x}_i^l + \sum_{j \neq i} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{d_{ij} + 1} \phi_x(\mathbf{m}_{i,j}) \\ \mathbf{h}_i^{l+1} &= \phi_h(\mathbf{h}_i^l, \sum_{j \in \mathcal{N}(i)} \phi_i(\mathbf{m}_{ij}) \mathbf{m}_{ij}) \end{aligned} \quad (18)$$

where l denotes the layer index, $\phi_i(\mathbf{m}_{ij})$ reweights messages passed from different edges in an attention weights manner, $d_{ij} + 1$ is normalizing the relative directions $\mathbf{x}_i^l - \mathbf{x}_j^l$ following previous methods (Satorras et al., 2021; Hoogeboom et al., 2022). All learnable functions, *i.e.*, ϕ_e, ϕ_x, ϕ_h , and, ϕ_i , are parameterized by Multi Layer Perceptrons (MLPs). Then a complete EGNN model can be realized by stacking L layers such that and satisfies the required equivariant constraint in Equations 3, 4, and 6.

B.2 EQUIVARIANT PHYSICAL PRIOR STEERED DENOISING NEURAL NETWORKS

The denoising neural network is implemented by multiple equivariant convolutional layers, and the difference in the Equation 18 is the hidden features \mathbf{h} . Due to the diffusion model is conditioned by $\mathbf{f}_x, \mathbf{f}_h$ from encoder \mathcal{E} , the hidden features for our denoising neural network is expressed as $\tilde{\mathbf{h}} \leftarrow [\mathbf{h}, \mathbf{f}_x, \mathbf{f}_h]$, where \mathbf{h} are original features of geometric graph and $[a, b]$ is concatenation operation.

B.3 MULTI-MODAL FEATURE REPRESENTATION OF MOLECULES

Multimodal features of molecules raise concerns for the term $\mathcal{L}_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$ in Equation 14. For categorical features such as the atom types, this model would however introduce an undesired bias (Hoogeboom et al., 2022). For the intermediate variable \mathbf{x}_t , we subdivide it into $\mathbf{z}_{x,t}$ and $\mathbf{z}_{h,t}$ in the proposed DM, which are coordinate variables and atomic attribute variables, respectively.

Coordinate Features. First we set $\sigma_t^2 \mathbf{I} \leftarrow \Sigma_\theta(\mathbf{x}_t, t) = \beta_t$ and add an additional correction term containing the estimated noise $\epsilon_{x,0}$ from denoising neural network ϵ . Then continuous positions \mathbf{z}_x in $p(\mathbf{z}_{x,0}|\mathbf{z}_{x,1})$ is expressed as:

$$p(\mathbf{z}_{x,0}|\mathbf{z}_{x,1}) = \mathcal{N}(\mathbf{z}_{x,0}|\mathbf{z}_{x,1}/\alpha_1 - \sigma_1/\alpha_1 \epsilon_{x,0}, \sigma_1^2/\alpha_1^2 \mathbf{I}) \quad (19)$$

Atom Type Features. For categorical features such as the atom type, the aforementioned integer representation is unnatural and introduces bias. Instead of using integers for these features, we operate directly on a one-hot representation. Suppose \mathbf{h} or $\mathbf{z}_{h,0}$ is an array whose values represent atom types in $\{c_1, \dots, c_d\}$. Then \mathbf{h} is encoded with a one-hot function $\mathbf{h} \leftarrow \mathbf{h}^{\text{one-hot}}$ such that $\mathbf{h}_{i,j}^{\text{one-hot}} \leftarrow \mathbf{1}_{h_i=c_i}$. and diffusion process over $\mathbf{z}_{h,t}$ at timestep t and sampling at final timestep are given as:

$$q(\mathbf{z}_{h,t}|\mathbf{z}_{h,0}) = \mathcal{N}(\mathbf{z}_{h,t}|\alpha_t \mathbf{h}^{\text{one-hot}}, \sigma_t^2 \mathbf{I}) \quad (20)$$

$$p(\mathbf{z}_{h,0}|\mathbf{z}_{h,1}) = \mathcal{C}(\mathbf{z}_{h,0}|\mathbf{p}), \mathbf{p} \propto \int_{1-\frac{1}{2}}^{1+\frac{1}{2}} \mathcal{N}(\mathbf{u}; \mu_\theta(\mathbf{z}_{h,1}, 1), \sigma_1^2) d\mathbf{u} \quad (21)$$

where \mathbf{p} is normalized to sum to one and \mathcal{C} is a categorical distribution.

Atom Charge. Atom charge is the ordinal type of physical quantity, and its sampling process at the final timestep can be formulated by standard practice (Ho et al., 2020):

$$p(\mathbf{z}_{\mathbf{h},0}|\mathbf{z}_{\mathbf{h},1}) = \int_{\mathbf{h}-\frac{1}{2}}^{\mathbf{h}+\frac{1}{2}} \mathcal{N}(\mathbf{u}; \mu_{\theta}(\mathbf{z}_{\mathbf{h},1}, 1), \sigma_1^2) d\mathbf{u} \quad (22)$$

Feature Scaling. To normalize the features and make them easier to process for the neural network, we add weights to different modalities. The relative scaling has a deeper impact on the model: when the features \mathbf{h} are defined on a smaller scale than the coordinates \mathbf{x} , the denoising process tends to first determine rough positions and decide on the atom types only afterward. Empirical knowledge shows that the weights for coordinate, atom type, and atom charge are 1, 0.25, and 0.1, respectively (Hoogeboom et al., 2022).

C LOSS OF EMAE IS SE(3)-INVARIANT

Equivariance. Molecules, typically existing within a three-dimensional physical space, are subject to geometric symmetries, including translations, rotations, and potential reflections. These are collectively referred to as the Euclidean group in 3 dimensions, denoted as $E(3)$ (Celeghini et al., 1991). A function F is said to be equivariant to the action of a group G if $T_g \circ F(\mathbf{x}) = F \circ S_g(\mathbf{x})$ for all $g \in G$, where S_g, T_g are linear representations related to the group element g (Serre et al., 1977). We consider the special Euclidean group $SE(3)$ for geometric graph generation involving translations and rotations. Moreover, the transformations S_g or T_g can be represented by a translation \mathbf{t} and an orthogonal matrix rotation \mathbf{R} . For a molecule $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$, the node features \mathbf{h} are $SE(3)$ -invariant while the coordinates \mathbf{x} are $SE(3)$ -equivariant, which can be expressed as $\mathbf{R}\mathbf{x} + \mathbf{t} = (\mathbf{R}\mathbf{x}_1 + \mathbf{t}, \dots, \mathbf{R}\mathbf{x}_N + \mathbf{t})$.

Proof. $\mathcal{L}_{\text{EAAE}}$ is $SE(3)$ -invariance

Recall the loss function:

$$\mathcal{L}_{\text{EAAE}} = \mathbb{E}_{q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f)} p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}) - \text{KL}[q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f) || \prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})] \quad (23)$$

Our expected outcome is $\forall \mathbf{R}, \mathcal{L}_{\text{EAAE}}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) = \mathcal{L}_{\text{EAAE}}(\mathbf{R}\mathbf{x}, \mathbf{h}, \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)$. We have:

$$\begin{aligned} & \mathcal{L}_{\text{EAAE}}(\mathbf{R}\mathbf{x}, \mathbf{h}, \mathbf{R}\mathbf{x}^f, \mathbf{h}^f) \\ &= \mathbb{E}_{q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)} p_\vartheta(\mathbf{R}\mathbf{x}, \mathbf{h} | \mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}) - \text{KL}[q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f) || \prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})] \\ &= \int_{\mathcal{G}} q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f) \log p_\vartheta(\mathbf{R}\mathbf{x}, \mathbf{h} | \mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}) + \int_{\mathcal{G}} \log \frac{q_\phi(\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)}{\prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})} \\ &= \int_{\mathcal{G}} q_\phi(\mathbf{R}\mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f) \log p_\vartheta(\mathbf{R}\mathbf{x}, \mathbf{h} | \mathbf{R}\mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}) \\ & \quad + \int_{\mathcal{G}} \log \frac{q_\phi(\mathbf{R}\mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)}{\prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})} \quad \mathbf{R}\mathbf{R}^{-1} = \mathbf{I} \\ &= \int_{\mathcal{G}} q_\phi(\mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f) \log p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}) \\ & \quad + \int_{\mathcal{G}} \log \frac{q_\phi(\mathbf{R}^{-1}\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f)}{\prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})} \quad SE(3) \text{ of } \mathbf{x}, \mathbf{f}_\mathbf{x}, \text{ \& } \mathbf{x}^f \\ &= \int_{\mathcal{G}} q_\phi(\mathbf{k}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f) \log p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{k}, \mathbf{f}_\mathbf{h}) \cdot |\mathbf{R}| \\ & \quad + \int_{\mathcal{G}} \log \frac{q_\phi(\mathbf{k}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f)}{\prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})} \quad \text{Let } \mathbf{k} = \mathbf{R}^{-1}\mathbf{f}_\mathbf{x} \\ &= \mathbb{E}_{q_\phi(\mathbf{k}, \mathbf{f}_\mathbf{h} | \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)} p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{k}, \mathbf{f}_\mathbf{h}) \\ & \quad - \text{KL}[q_\phi(\mathbf{k}, \mathbf{f}_\mathbf{h} | \mathbf{x}^f, \mathbf{h}^f) || \prod_i^N \mathcal{N}(f_{\mathbf{x},i}, f_{\mathbf{h},i} | 0, \mathbf{I})] \quad |\mathbf{R}| = 1 \\ &= \mathcal{L}_{\text{EAAE}}(\mathbf{x}^f, \mathbf{h}^f) \end{aligned} \quad (24)$$

□

Given the fragment \mathcal{G}^f , we subtract the center of gravity from $\mathbf{x}^f \in \mathcal{G}^f$, and thereby ensure that \mathcal{E} receives isotropic geometric graph, and all together guarantee that the loss of **EAAE** is $SE(3)$ -invariant.

D LOSS OF *GODD* IS AN $SE(3)$ -INVARIANT VARIATIONAL LOWER BOUND TO THE LOG-LIKELIHOOD

First, we present the rigorous statement of the Theorem 3.2 here:

Theorem D.1. *Given predefined and valid $\{\alpha_i\}_{i=0}^T$, $\{\beta_i\}_{i=0}^T$, and $\{\gamma_i\}_{i=0}^T$ Let $w(t)$ satisfies:*

$$1. \forall t \in [1, \dots, T], w(t) = \frac{\beta_t^2}{2\gamma_t^2(1 - \beta_t)(1 - \alpha_t^2)} \quad (25)$$

$$2. w(0) = -1 \quad (26)$$

Then given the geometric datapoint $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle \in \mathbb{R}^{N \times (3+d)}$ and its subset $\mathcal{G}^f \langle \mathbf{x}^f, \mathbf{h}^f \rangle \in \mathbb{R}^{F \times (3+d)}$ the loss \mathcal{L} of the proposed method is expressed as:

$$\mathcal{L} := \mathcal{L}_{EAAE} + \mathcal{L}_{PSDM} \quad (27)$$

which satisfies:

$$1. \forall \mathbf{R} \text{ and } \mathbf{t}, \mathcal{L}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) = \mathcal{L}(\mathbf{R}\mathbf{x} + \mathbf{t}, \mathbf{h}, \mathbf{R}\mathbf{x}^f + \mathbf{t}, \mathbf{h}^f) \quad (28)$$

$$2. \mathcal{L}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) \geq -\mathbb{E}_{p_{\langle \mathbf{x}, \mathbf{h} \rangle \in \mathcal{G}, [\mathbf{f}_x, \mathbf{f}_h] = \mathcal{E}_\phi(\mathcal{G}^f)}} [\log p_\theta(\mathbf{z}_x, \mathbf{z}_h | \mathbf{f}_x, \mathbf{f}_h)] \quad (29)$$

And we have $\log p_\theta(\mathbf{x}_0, \mathbf{h}_0)$ is the marginal distribution of $\langle \mathbf{x}, \mathbf{h} \rangle$ under *GODD*.

The theorem proposed herein posits two distinct assertions. Firstly, Equation 28 illustrates that the loss function \mathcal{L} is $SE(3)$ -invariant, meaning it remains unchanged under any rotational or translational transformations. Secondly, Equation 29 suggests that \mathcal{L} acts as a variational lower bound for the log-likelihood. We provide comprehensive proofs for these assertions separately, commencing with Equation 29.

Proof. \mathcal{L} is a variational lower bound of the log-likelihood

Recall the loss function:

$$\mathcal{L}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) = \mathcal{L}_{EAAE} + \mathcal{L}_{PSDM} \quad (30)$$

$$= \mathbb{E}_{q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}^f, \mathbf{h}^f)} p_\theta(\mathbf{x}, \mathbf{h} | \mathbf{f}_x, \mathbf{f}_h) - \text{KL}[q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}^f, \mathbf{h}^f) || \prod_i^N \mathcal{N}(f_{x,i}, f_{h,i} | 0, \mathbf{I})] \quad (31)$$

$$+ \mathbb{E}_{\mathcal{G}, \mathcal{E}_\phi(\mathcal{G}^f), \epsilon, t} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{h}_t, \mathbf{f}_x, \mathbf{f}_h, t)\|^2] \quad (32)$$

\mathcal{L}_{EAAE} is a standard variational autoencoder and has been proven to be a variational lower bound of the log-likelihood (Kingma & Welling, 2014). For simplicity, we denote $\mathbf{z}_{x,t}$, $\mathbf{z}_{h,t}$ as \mathbf{z}_t , and \mathbf{f}_x , \mathbf{f}_h as \mathbf{f} , then we expect \mathcal{L}_{PSDM} has:

$$\log p_\theta(\mathbf{z} | \mathbf{f}) \geq \text{KL}[q(\mathbf{z}_{1:T} | \mathbf{z}_0) || p_\theta(\mathbf{z} | \mathbf{f})] \quad (33)$$

$$\begin{aligned}
\log p_\theta(\mathbf{z}|\mathbf{f}) &\geq \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{p_\theta(\mathbf{z}_{0:T}|\mathbf{f})}{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \right] \\
&= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f}) \prod_{t=2}^T p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})}{q(\mathbf{z}_1|\mathbf{z}_0) \prod_{t=2}^T q(\mathbf{z}_t|\mathbf{z}_{t-1})} \right] \\
&= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})}{q(\mathbf{z}_1|\mathbf{z}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})}{q(\mathbf{z}_t|\mathbf{z}_{t-1})} \right] \\
&= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})}{q(\mathbf{z}_1|\mathbf{z}_0)} + \log \prod_{t=2}^T \frac{p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})}{q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)q(\mathbf{z}_t|\mathbf{z}_0)} \right] \\
&= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[\log \frac{p(\mathbf{z}_T)p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})}{q(\mathbf{z}_T|\mathbf{z}_0)} + \sum_{t=2}^T \log \frac{p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})}{q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)} \right] \tag{34} \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{z}_0)} [p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})] + \mathbb{E}_{q(\mathbf{z}_T|\mathbf{z}_0)} \left[\log \frac{p(\mathbf{z}_T)}{q(\mathbf{z}_T|\mathbf{z}_0)} \right] \\
&\quad + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{z}_0)} \left[\log \frac{p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})}{q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)} \right] \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{z}_0)} [p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})] - \text{KL}[q(\mathbf{z}_T|\mathbf{z}_0)||p(\mathbf{z}_T)] \\
&\quad - \sum_{t=2}^T \mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)} [\text{KL}[q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)||p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})]]
\end{aligned}$$

where we denote $\text{KL}[q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)||p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{f})]$ as $\mathcal{L}_{\text{PSDM}, t-1}$, then we have:

$$\mathcal{L}_{\text{PSDM}, t-1} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\frac{\beta_t^2}{2\gamma_t^2(1-\beta_t)(1-\alpha_t^2)} \|\epsilon - \epsilon_\theta(\mathbf{z}_t, \mathbf{f}, t)\|_2^2 \right] \tag{35}$$

which gives us the weights of $w(t)$ for $t = 1, \dots, T$.

For term $\mathbb{E}_{q(\mathbf{z}_1|\mathbf{z}_0)} [p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})]$, we denote as $\mathcal{L}_{\text{PSDM}, 0}$. With sampling at the final timestep for different modality features and a normalization constant Z , we have:

$$\mathcal{L}_{\text{PSDM}, 0} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\log Z^{-1} - \frac{1}{2} \|\epsilon - \epsilon_\theta(\mathbf{z}, \mathbf{f}, 1)\|^2 \right] \tag{36}$$

Since $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$, we have:

$$\mathcal{L}_{\text{PSDM}, T} = \text{KL}[q(\mathbf{z}_T|\mathbf{z}_0)||p(\mathbf{z}_T)] = 0 \tag{37}$$

Therefore, we have:

$$\mathbb{E}_{p(\mathbf{x}, \mathbf{h}) \in \{\mathcal{G}\}, [\mathbf{f}_\mathbf{x}, \mathbf{f}_\mathbf{h}] = \mathcal{E}_\phi(\mathcal{G}^f)} [\log p_\theta(\mathbf{z}|\mathbf{f})] \geq - \sum_{t=2}^T \mathcal{L}_{\text{PSDM}, t-1} - \mathcal{L}_{\text{PSDM}, 0} = -\mathcal{L}_{\text{PSDM}} \tag{38}$$

□

We then prove Equation 28:

Proof. \mathcal{L} is $SE(3)$ -invariance

Our expected outcome is $\forall \mathbf{R}, \mathcal{L}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) = \mathcal{L}(\mathbf{R}\mathbf{x}, \mathbf{h}, \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)$, and $\forall \mathbf{R}, \mathcal{L}_{\text{EAAE}}(\mathbf{x}, \mathbf{h}, \mathbf{x}^f, \mathbf{h}^f) = \mathcal{L}_{\text{EAAE}}(\mathbf{R}\mathbf{x}, \mathbf{h}, \mathbf{R}\mathbf{x}^f, \mathbf{h}^f)$ is ensured in Proof. C. For $\mathcal{L}_{\text{PSDM}}$, we expect $\forall \mathbf{R}, \mathcal{L}_{\text{PSDM}}(\mathbf{R}\mathbf{z}_{\mathbf{x}, 0}, \mathbf{z}_{\mathbf{h}, 0}, \mathbf{R}\mathbf{f}) = \mathcal{L}_{\text{PSDM}}(\mathbf{z}_{\mathbf{x}, 0}, \mathbf{z}_{\mathbf{h}, 0}, \mathbf{f})$ we have:

$$\begin{aligned}
&\mathcal{L}_{\text{PSDM}}(\mathbf{R}\mathbf{z}_{\mathbf{x}, 0}, \mathbf{z}_{\mathbf{h}, 0}) \\
&= \mathbb{E}_{\mathcal{G}, \mathcal{E}_\phi} \left[\sum_{t=2}^T \mathbb{E}_{q(\mathbf{z}_t|\mathbf{R}\mathbf{z}_0)} [\text{KL}[q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{R}\mathbf{z}_0)||p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{R}\mathbf{f})]] - \mathbb{E}_{q(\mathbf{z}_1|\mathbf{R}\mathbf{z}_0)} [p_\theta(\mathbf{R}\mathbf{z}_0|\mathbf{z}_1, \mathbf{R}\mathbf{f})] \right]
\end{aligned}$$

$$\begin{aligned}
&= \int_{\mathcal{G}} \left[\sum_{t=2}^T \log \frac{q(\mathbf{z}_{t-1} | q(\mathbf{z}_t, \mathbf{R}\mathbf{z}_0))}{p_{\theta}(\mathbf{z}_{t-1} | \mathbf{z}_t, \mathbf{R}\mathbf{f})} - \log p_{\theta}(\mathbf{R}\mathbf{z}_0 | \mathbf{z}_1, \mathbf{R}\mathbf{f}) \right] \\
&= \int_{\mathcal{G}} \left[\sum_{t=2}^T \log \frac{q(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_{t-1} | q(\mathbf{R}\mathbf{R}^{-1}\mathbf{z}_t, \mathbf{R}\mathbf{z}_0))}{\mathbf{R}\mathbf{R}^{-1}p_{\theta}(\mathbf{z}_{t-1} | \mathbf{R}\mathbf{R}^{-1}\mathbf{z}_t, \mathbf{R}\mathbf{f})} - \log p_{\theta}(\mathbf{R}\mathbf{z}_0 | \mathbf{R}\mathbf{R}^{-1}\mathbf{z}_1, \mathbf{R}\mathbf{f}) \right] \quad \mathbf{R}\mathbf{R}^{-1} = \mathbf{I} \\
&= \int_{\mathcal{G}} \left[\sum_{t=2}^T \log \frac{q(\mathbf{R}^{-1}\mathbf{z}_{t-1} | q(\mathbf{R}^{-1}\mathbf{z}_t, \mathbf{z}_0))}{\mathbf{R}^{-1}p_{\theta}(\mathbf{z}_{t-1} | \mathbf{R}^{-1}\mathbf{z}_t, \mathbf{f})} - \log p_{\theta}(\mathbf{z}_0 | \mathbf{R}^{-1}\mathbf{z}_1, \mathbf{f}) \right] \quad SE(3) \text{ of } \mathbf{f}_{\mathbf{x}} \text{ \& } \mathbf{z}_t \\
&= \mathbb{E}_{\mathcal{G}, \varepsilon_{\phi}} \left[\sum_{t=2}^T \log \frac{q(\mathbf{j}_{t-1} | q(\mathbf{j}_t, \mathbf{z}_0))}{\mathbf{R}^{-1}p_{\theta}(\mathbf{z}_{t-1} | \mathbf{j}_t, \mathbf{f})} - \log p_{\theta}(\mathbf{z}_0 | \mathbf{j}_1, \mathbf{f}) \right] \quad \text{Let } \mathbf{j}_t = \mathbf{R}^{-1}\mathbf{z}_t \\
&= \mathcal{L}_{\text{PSDM}}(\mathbf{z}_{\mathbf{x},0}, \mathbf{z}_{\mathbf{h},0}, \mathbf{f})
\end{aligned} \tag{39}$$

□

E TRAINING DETAILS

Parameters

1. Optimizer: Adam (Kingma & Ba, 2015) optimizer is used with a constant learning rate of 10^{-4} as our default training configuration.
2. Batch size: 64.
3. EGNN in **PSDM**: 9 layers and 256 hidden features for QM9, 4 layers and 256 hidden features for GEOM-DRUG.
4. EGNN in **EAAE**: 1 layer and 256 hidden features for the encoder for QM9 and GEOM-DRUG, 9 layers and 4 layers with 256 hidden features for the decoder for QM9 and GEOM-DRUG, respectively.
5. Latent dimension of $\mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{h}}$: latent dimension is 3 and 1 for $\mathbf{f}_{\mathbf{x}}$ and $\mathbf{f}_{\mathbf{h}}$, respectively.
6. Epoch: 3000 for QM9 and 10 for GEOM-DRUG.

Training

1. GPU: NVIDIA GeForce RTX 3090
2. CPU: Intel(R) Xeon(R) Platinum 8338C CPU
3. Memory: 512 GB
4. Time: Around 7 days for QM9 and 20 days for GEOM-DRUG.

Specific Parameters 1. On QM9, we train **PSDM** with 9 layers and 256 hidden features with a batch size 64; 2. On GEOM-DRUG, we train **PSDM** with 4 layers and 256 hidden features, with batch size 64;

F ALGORITHMS

This section contains two main algorithms of the proposed **GODD**. Algorithm 1 presents the pseudo-code for training **GODD** on the in distributional training data set $\{\mathcal{G}_I\}$ and corresponding fragment set $\{\mathcal{G}_I^f\}$. Algorithm 2 presents the process of OOD molecule generation using the OOD scaffold/ring \mathcal{G}_O^f .

G QM9 DATASET

QM9 (Ramakrishnan et al., 2014) is a comprehensive dataset that provides geometric, energetic, electronic, and thermodynamic properties for a subset of the GDB-17 database (Ruddigkeit et al., 2012), comprising 134 thousand stable organic molecules with up to nine heavy atoms.

Algorithm 1 Training *GODD*

```

1: Input: in-distribution geometric data point  $\mathcal{G}_I = \langle \mathbf{x}, \mathbf{h} \rangle$ , corresponding fragment  $\mathcal{G}_I^f$ , asymmetric encoder  $\mathcal{E}_\phi$  and decoder  $\mathcal{D}_\vartheta$ , denoising network  $\epsilon_\theta$ ;
2: EAAE:
3:  $\mu_x, \mu_h \leftarrow \mathcal{E}_\phi(\mathbf{x}^f, \mathbf{h}^f)$  // Encode
4:  $\langle \epsilon_x, \epsilon_h \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for EAAE
5:  $\epsilon_x \leftarrow \epsilon_x - \mathbf{G}(\epsilon_x)$  // Subtract Center of Gravity
6:  $\mathbf{f}_x, \mathbf{f}_h \leftarrow \mu + \langle \epsilon_x, \epsilon_h \rangle \odot \sigma_0$  // Reparameterization
7: PSDM:
8:  $t \sim \mathcal{U}(0, T)$  // Sample Timestep
9:  $\langle \epsilon_x, \epsilon_h \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for PSDM
10:  $\epsilon_x \leftarrow \epsilon_x - \mathbf{G}(\epsilon_x)$  // Subtract Center of Gravity
11:  $\mathbf{z}_{x,t}, \mathbf{z}_{h,t} \leftarrow \alpha_t[\mathbf{x}, \mathbf{h}] + \sigma_t \epsilon$  // Diffuse
12:  $\hat{\mathbf{x}}, \hat{\mathbf{h}} \leftarrow \mathcal{D}_\vartheta(\mathbf{f}_x, \mathbf{f}_h)$  // Decode
13: Optimization
14:  $\mathcal{L}_{\text{EAAE}} \leftarrow \mathcal{L}([\hat{\mathbf{x}}, \hat{\mathbf{h}}], [\mathbf{x}, \mathbf{h}]) + \text{KL}$  //  $\mathcal{L}$  for EAAE
15:  $\mathcal{L}_{\text{PSDM}} \leftarrow \|\epsilon - \epsilon_\theta(\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, t, \mathbf{f}_x, \mathbf{f}_h)\|^2$  //  $\mathcal{L}$  for PSDM
16:  $\mathcal{L}_{\text{GODD}} \leftarrow \mathcal{L}_{\text{EAAE}} + \mathcal{L}_{\text{PSDM}}$  // Total Loss
17:  $\phi, \vartheta, \theta \leftarrow \text{optimizer}(\mathcal{L}_{\text{GODD}}, \phi, \vartheta, \theta)$  // Optimize
18: return  $\phi, \theta$ 

```

Algorithm 2 Adaptive Sampling of *GODD*

```

1: Input: OOD fragment  $\mathcal{G}_O^f = \langle \mathbf{x}_O^f, \mathbf{h}_O^f \rangle$ , encoder  $\mathcal{E}_\phi$ , denoising network  $\epsilon_\theta$ ;
2:  $\mu_x, \mu_h \leftarrow \mathcal{E}_\phi(\mathbf{x}_O^f, \mathbf{h}_O^f)$  // Encode
3:  $\langle \epsilon_x, \epsilon_h \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for EAAE
4:  $\epsilon_x \leftarrow \epsilon_x - \mathbf{G}(\epsilon_x)$  // Subtract Center of Gravity
5:  $\mathbf{f}_x, \mathbf{f}_h \leftarrow \mu + \langle \epsilon_x, \epsilon_h \rangle \odot \sigma_0$  // Target Condition
6:  $\langle \mathbf{z}_{x,T}, \mathbf{z}_{h,T} \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for Generation
7: for  $t$  in  $T, T-1, \dots, 1$  do
8:    $\langle \epsilon_x, \epsilon_h \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Denoising
9:    $\epsilon_x \leftarrow \epsilon_x - \mathbf{G}(\epsilon_x)$  // Subtract Center of Gravity
10:   $\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1} \leftarrow \frac{1}{\sqrt{1-\beta_t}}(\langle \mathbf{z}_{x,t}, \mathbf{z}_{h,t} \rangle - \frac{\beta_t}{\sqrt{1-\alpha_t^2}} \epsilon_\theta(\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, t, \mathbf{f}_x, \mathbf{f}_h)) + \rho_t \epsilon$ 
11: end for
12:  $\mathbf{x}, \mathbf{h} \leftarrow p(\mathbf{z}_{x,0}, \mathbf{z}_{h,0} | \mathbf{z}_{x,1}, \mathbf{z}_{h,1}, \mathbf{f}_x, \mathbf{f}_h)$ 
13: return  $\langle \mathbf{x}, \mathbf{h} \rangle$ 

```

G.1 SCAFFOLD SPLIT QM9

We utilized the open-source software, RDkit (Landrum et al., 2016), to examine the scaffold and ring of each molecule. QM9 dataset¹ comprises a total of 130,831 molecules, encompassing 15,661 unique scaffolds. Molecules lacking a scaffold were denoted as ‘-’ and included in the total scaffold count. The dataset was divided based on scaffold frequency. Specifically, the in-distribution subset contained 100,000 molecules and 1,054 scaffolds. The OOD I subset included 15,000 molecules and 2,532 scaffolds, while the OOD II subset consisted of 15,831 molecules and 12,075 scaffolds.

Figure 4(a) presents the division’s schematic diagram. Figure 4(b) displays the logarithmic histogram of the scaffolds in each dataset segment. It is evident that the in-distribution dataset contains the most frequent scaffolds, primarily concentrated above 100. The frequency of scaffolds in the OOD I dataset ranges between 10 and 100. In contrast, the scaffolds in the OOD II dataset are primarily concentrated within 10, with most appearing only once. Figures, SMILES, and frequencies of some example scaffolds in each sub-dataset are given in Figure 5.

¹<https://springernature.figshare.com/ndownloader/files/3195389>

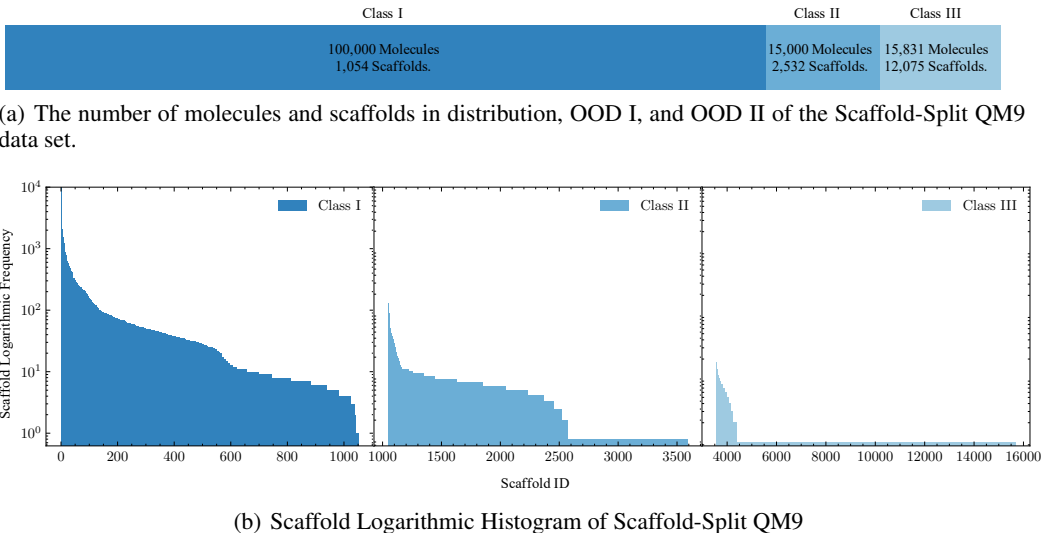


Figure 4: Scaffold-Split QM9

Scaffold Split Source Dataset						
SMILES	C1=CCCC1	C1C[NH]CN1	C1CC2(C1)CN2	C1C2C1N1CC21	C1C2C3CN2C13	C1CN1
Frequency	1333	1333	189	316	246	3722
Scaffold Split Target Dataset I						
SMILES	C1C2OC3C1C3O2	C1CC(C2CC2)O1	C1CC1C1COC1	C1CC2C3CN2C13	C1CCNCC1	C1COCCN1
Frequency	45	70	56	24	24	40
Scaffold Split Target Dataset II						
SMILES	C1C2OC1C2CN1CC1	C1CC1CC1C2CC1C2	C1CC1CC1C2CC1O2	C1CC1OC1C2CC1C2	C1CN=C(OC2CC2)C1	C1CN1CC1C2CN1C2
Frequency	1	1	1	1	1	1

Figure 5: Scaffold Examples of QM9 Split by Scaffolds.

G.2 RING NUMBER SPLIT QM9

The QM9 dataset could categorize molecules into nine groups based on the number of rings, ranging from 0 to 8. As the number of rings increases, the quantity of molecules correspondingly decreases. We partition the QM9 dataset into two subsets based on ring count. The in-distribution dataset comprises acyclic molecules and those with 1 to 3 rings, while the OOD dataset includes molecules with 4 to 8 rings. Figure 6 presents a schematic diagram illustrating example molecules with 0 to 8 rings.

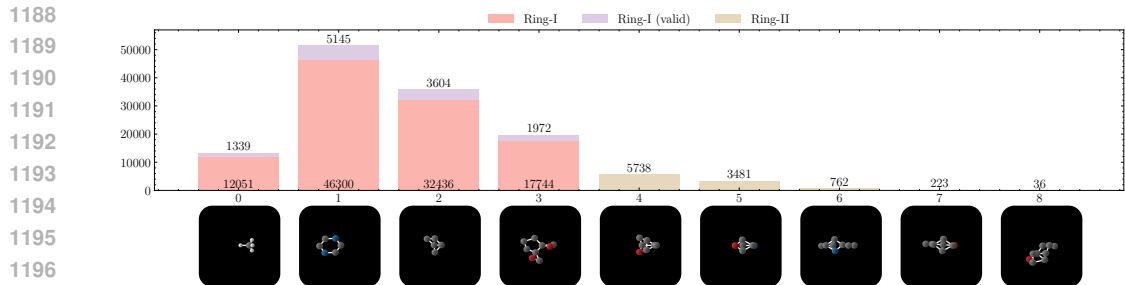


Figure 6: Ring Examples of QM9 Split by Ring Number.

H GEOM-DRUG DATASET

GEOM-DRUG (Geometric Ensemble Of Molecules) dataset (Axelrod & Gómez-Bombarelli, 2022) encompasses around 450,000 molecules, each with an average of 44.2 atoms and a maximum of 181 atoms².

H.1 RING NUMBER SPLIT GEOM-DRUG

The GEOM-DRUG dataset classifies molecules into sixteen categories based on the number of rings, ranging from 0 to 14 and 22. As the ring count increases, the number of molecules correspondingly decreases. The dataset is partitioned into two subsets according to ring count: the in-distributional dataset, which includes molecules with 0 to 10 rings and a count exceeding 100, and four OOD datasets, which comprises molecules with 11 to 14 and 22 rings. Figure 7 provides a schematic representation of the molecule distribution by ring number.

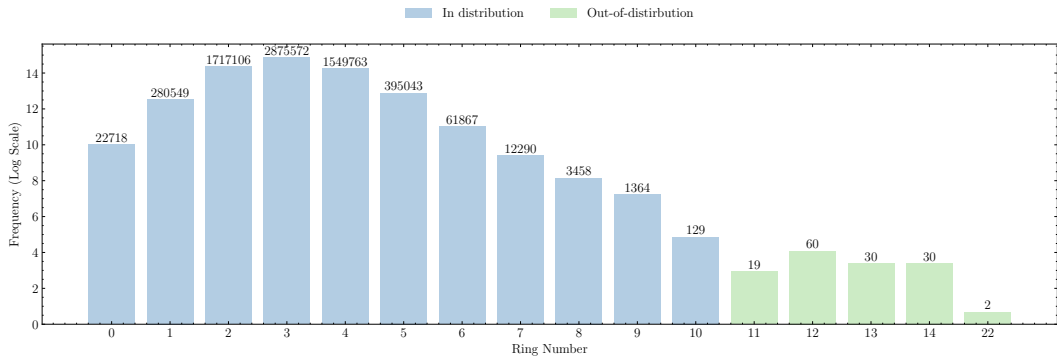


Figure 7: Ring Distribution of GEOM-DRUG dataset.

I GEOM-LINKER DATASET

The GEOM-LINKER dataset for linker design is constructed by (Igashov et al., 2024) based on GEOM-DRUG. The authors decomposed the molecule into three or more fragments with one or two linkers connecting them. The dataset contains 41,907 molecules and 285,140 fragments, and the original dataset is randomly split into train (282,602 examples), validation (1,250 examples), and test (1,288 examples) sets. Atom types considered for this dataset are C, O, N, F, S, Cl, Br, I, and P.

We present the distribution of molecules in GEOM-LINKER according to the number of rings in Figure 8. The diagram illustrates the molecules with 3 to 5 rings are the majority and molecules

²<https://dataverse.harvard.edu/file.xhtml?fileId=4360331&version=2.0>

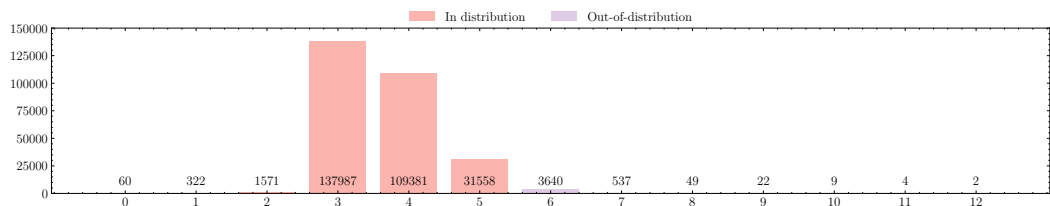


Figure 8: Ring Distribution of GEOM-LINKER dataset.

with 8 to 12 rings exhibit data sparsity in the whole dataset. Thereby, we split the dataset according to the ring numbers into in-distribution (0-5 rings, 280,879 samples) and OOD (6-12 rings, 4,263 samples).

J FULL RESULTS OF OOD RING-STRUCTURE MOLECULE GENERATION

We present the detailed quantitative evaluation results of ring adaptive molecule generation tasks in Tables 8 and 9. The results show that the proposed method has dominant performance in all metrics, including ring number proportion, validity, novelty, and success rate.

It is significant to note that the entire QM9 dataset comprises only 36 eight-ring molecules. When the proposed algorithm utilizes the ring structures of these 36 8-ring molecules as input, the target validity reaches an impressive 72.2%, and the novelty is as high as 80.9%. Considering that there are only 36 fundamental 8-ring structures, the uniqueness is slightly lower (27.4%). Nevertheless, the generation of 10,000 molecules resulted in 2,388 valid, unique, and entirely novel eight-ring molecules, which is a substantial breakthrough compared to existing methods (even those models trained on eight-ring molecules) that failed to discover any new eight-ring molecules.

Table 8: Results of molecule proportion in terms of ring-number (P) and molecule validity (V) The **best** results are highlighted in bold. QM9 only contains 36 eight-ring molecules and the proportion for eight-ring is nearly 0.

	0	1	2	3	4	5	6	7	8	Averaged
Method	P (%)									
QM9	10.2	39.3	27.6	15.1	4.4	2.7	0.6	0.2	0.0	-
EDM [†]	10.5	39.8	28.0	14.5	4.0	2.9	0.2	0.1	0.0	-
GeoLDM [†]	12.0	38.6	27.0	15.3	4.6	2.2	0.2	0.1	0.0	-
EDM [‡]	12.1	44.1	29.8	11.8	1.7	0.5	0.0	0.0	0.0	-
GeoLDM [‡]	2.8	41.5	32.1	15.7	4.7	2.7	0.3	0.1	0.0	-
C-EDM [‡]	98.9	94.2	80.8	64.4	12.6	26.8	0.3	0.1	0.0	-
C-GeoLDM [‡]	97.1	89.4	74.2	52.4	22.3	22.7	0.9	0.2	0.0	-
EEGSDE [‡]	98.4	92.2	77.6	58.2	14.1	17.6	0.3	0.0	0.0	-
MOOD [‡]	80.7	87.1	86.1	73.3	34.1	32.3	10.3	0.2	0.0	-
CGD [‡]	82.3	84.8	86.2	83.6	34.4	22.4	10.3	10.1	0.0	-
GODD[‡]	99.9	99.8	99.1	97.6	92.5	89.7	78.7	88.2	82.1	-
	Target Valid (%)									
QM9	97.7	97.7	97.7	97.7	97.7	97.7	97.7	97.7	97.7	97.7
EDM [†]	10.8	36.1	26.7	13.9	4.0	2.3	0.2	0.1	0.0	10.5
GeoLDM [†]	11.2	36.2	25.2	14.3	4.3	2.0	0.2	0.1	0.0	10.4
EDM [‡]	11.4	41.4	28.0	11.1	1.6	0.5	0.0	0.0	0.0	10.4
GeoLDM [‡]	2.7	38.8	30.0	14.7	4.4	2.6	0.3	0.1	0.0	10.4
C-EDM [‡]	86.6	85.4	74.9	59.8	12.1	23.3	0.2	0.1	0.0	38.0
C-GeoLDM [‡]	86.2	79.6	65.8	48.1	20.4	20.7	0.9	0.2	0.0	35.7
EEGSDE [‡]	96.7	92.1	77.2	58.0	13.9	17.4	0.3	0.0	0.0	39.5
MOOD [‡]	75.5	81.7	80.6	68.9	32.0	30.1	9.6	0.1	0.0	42.1
CGD [‡]	77.0	79.6	81.1	78.4	32.3	20.9	9.5	9.5	0.0	43.2
GODD[‡]	31.7	91.4	91.4	92.1	85.3	85.2	69.5	82.5	72.2	77.9

Table 9: Results of molecule proportion in terms of novelty (N) and success rate (S). The **best** results are highlighted in bold.

	0	1	2	3	4	5	6	7	8	Averaged
Method	Target Novelty (%)									
EDM [†]	7.1	23.6	17.5	9.1	2.6	1.5	0.1	0.1	0.0	6.8
GeoLDM [†]	7.0	22.4	15.6	8.9	2.7	1.3	0.1	0.0	0.0	6.4
EDM [‡]	7.5	27.1	18.3	7.2	1.1	0.3	0.0	0.0	0.0	6.8
GeoLDM [‡]	1.7	25.0	19.4	9.5	2.8	1.7	0.2	0.1	0.0	6.7
C-EDM [‡]	57.1	59.7	54.2	44.2	9.9	20.1	0.2	0.1	0.0	27.3
C-GeoLDM [‡]	63.3	61.6	53.3	40.1	17.3	18.3	0.7	0.1	0.0	28.3
EEGSDE [‡]	63.9	61.4	53.0	42.5	9.9	14.1	0.3	0.0	0.0	27.2
MOOD [‡]	50.0	53.9	53.6	44.4	20.6	20.0	6.3	0.1	0.0	27.6
CGD [‡]	51.0	52.5	53.1	51.3	21.0	13.9	6.3	6.2	0.0	28.4
GODD[‡]	96.6	51.3	55.6	60.2	69.5	63.5	71.5	83.4	80.9	70.3
	Success Rate (%)									
EDM [†]	6.5	21.9	16.2	8.4	2.4	1.4	0.1	0.1	0.0	6.3
GeoLDM [†]	6.4	20.6	14.4	8.2	2.4	1.2	0.1	0.0	0.0	5.9
EDM [‡]	6.9	25.1	17.0	6.7	1.0	0.3	0.0	0.0	0.0	6.3
GeoLDM [‡]	1.6	23.0	17.8	8.7	2.6	1.5	0.2	0.1	0.0	6.1
C-EDM [‡]	48.1	53.8	50.0	40.5	7.9	16.8	0.2	0.1	0.0	24.1
C-GeoLDM [‡]	54.6	54.6	46.9	36.8	15.4	15.6	0.6	0.1	0.0	25.0
EEGSDE [‡]	54.7	54.7	46.9	39.5	9.5	12.2	0.2	0.0	0.0	24.2
MOOD [‡]	45.9	49.8	49.4	41.0	18.9	18.3	5.8	0.1	0.0	25.5
CGD [‡]	46.8	48.5	49.1	47.3	19.5	12.8	5.8	5.7	0.0	26.2
GODD[‡]	25.9	43.4	46.2	50.4	53.8	41.0	46.1	34.1	23.9	40.5

K VISUALIZATION

In this section, we provide additional visualizations of physical prior steered molecule generation by *GODD* for OOD scaffold generation and ring number generation in Figures 9 and 10

As depicted in the two figures, the model consistently generates realistic molecular geometries with OOD scaffolds or ring numbers.

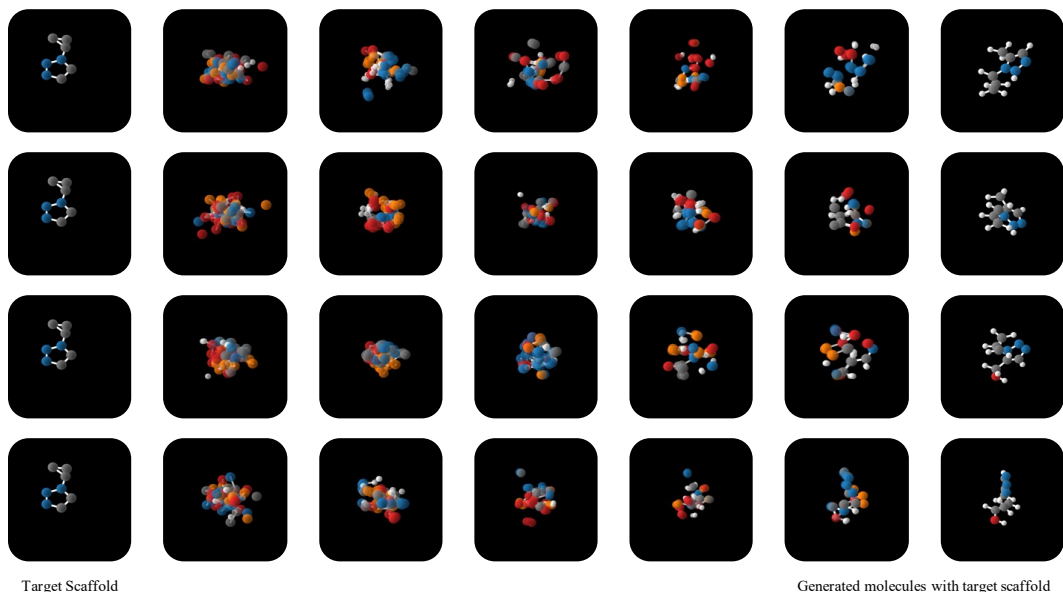


Figure 9: Molecules Generated by *GODD* for Scaffold Adaptive Generation Under The Same Unseen Scaffold Condition.

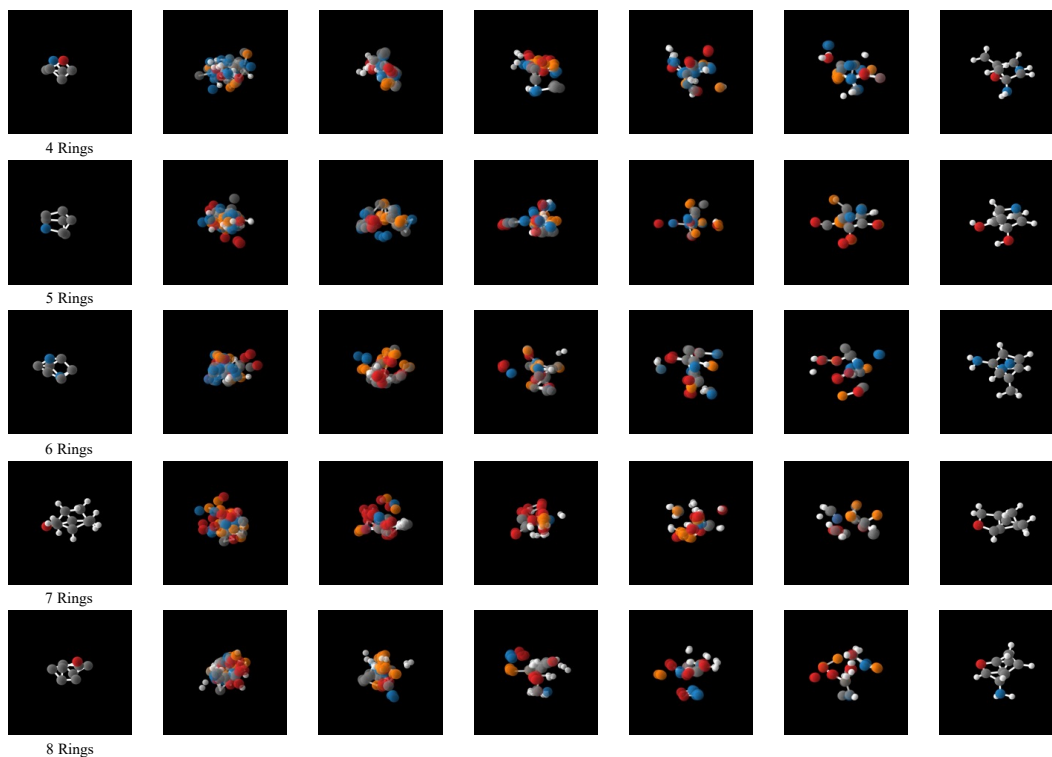


Figure 10: Molecules Generated by *GODD* for Ring Number Adaptive Generation For Unseen Ring Numbers

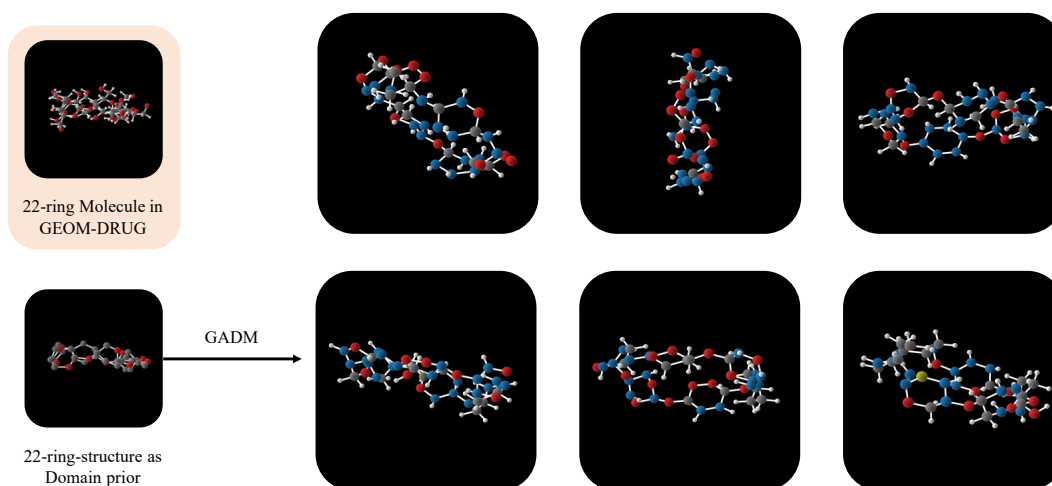


Figure 11: Molecules Generated by *GODD* for Ring Number Adaptive Generation For Unseen Ring Numbers on GEOM-DRUG Dataset.

L RELATED WORK

Molecule Generation Models. Prior studies on molecule generation focused on generating molecules as 2D graphs (Jin et al., 2018; Liu et al., 2018; Shi et al., 2020). However, there has been a growing interest in 3D molecule generation. G-SchNet (Gebauer et al., 2019) and G-SphereNet (Luo & Ji, 2022) utilize autoregressive techniques to construct molecules incrementally by progressively connecting atoms or molecular fragments. These frameworks necessitate either a meticulous formulation of complex action space or action ordering.

More recently, the focus has shifted towards using Diffusion Models (DMs) for 3D molecule generation (Hooeboom et al., 2022; Xu et al., 2023; Wu et al., 2022; Song et al., 2024). To mitigate the inconsistency of unified Gaussian diffusion across diverse modalities, a latent space was introduced by (Xu et al., 2023). To tackle the atom-bond inconsistency problem, different noise schedulers were proposed by (Peng et al., 2023) for various modalities to accommodate noise sensitivity. However, these algorithms do not account for generating novel molecules outside the training distribution.

Out-of-Distribution Molecule Generation. OOD generation, although under-explored, is of paramount importance, especially considering that molecules generated by machine-learning methods often exhibit a “striking similarity” (Walters & Murcko, 2020). In recent years, some preliminary work has begun to use reinforcement learning (Yang et al., 2021) and out-of-distribution control (Lee et al., 2023) to explore the generation of novel molecules. However, these methods are still challenging when designing novel molecules in data-sparse regions with fragment shifts. As proposed by (Lee et al., 2023), MOOD employs an OOD control and integrates a property-predictor-based diffusion scheme to optimize molecules for specific chemical properties. Similarly, CGD (Klarner et al., 2024) leverages unlabeled data to improve the generalization of guided diffusion models. However, these predictor-based OOD methods fail to generate novel molecules with OOD fragments that are sparse for training a classifier.

Fragment-Based Drug Design. The discovery of new molecules is crucial across various fields, and there are four primary approaches to this task (Murray & Rees, 2009): (1) searching from an existing molecule, (2) developing from a natural product, (3) high-throughput screening, and (4) fragment-based drug discovery (FBDD). Among these, FBDD has gained significant importance and interest over the past decades due to its higher efficiency compared to other methods (Murray & Rees, 2009). Typically, fragments are selected based on the “rule of three” (Congreve et al., 2003) criteria and thereby can be grown, linked, or merged to develop potential molecules (Bian & Xie, 2018). Recently, there has been a growing trend in enhancing FBDD with machine learning techniques (Wu et al., 2024; Igashov et al., 2024; Guan et al., 2024). However, these methods often overlook the issue of fragment sparsity in datasets, highlighting the need for an OOD molecular generative model capable of producing realistic molecules in data-sparse regions.

M IMPACT STATEMENTS

This paper presents work whose goal is to advance the field of generative Artificial Intelligence (AI) for scientific fields, such as material science, chemistry, and biology. The obtained experience/knowledge will greatly boost generative AI technologies in facilitating the process of scientific knowledge discovery.

Machine learning for molecule generation opens up possibilities for designing molecules beyond therapeutic purposes, such as the creation of illicit drugs or dangerous substances. The potential for misuse and unintended consequences necessitates strict ethical guidelines, robust regulation, and responsible use of these technologies to prevent harm to individuals and society.

N ACRONYMS LIST

ACRONYMS

GODD Geometric OOD Diffusion Model. 1–10, 18, 20, 21, 25–27

EAAE Equivariant Asymmetric Autoencoder. 4–6, 9, 15, 17–21

1566 **PSDM** Physical Prior Steered Diffusion Model. 6, 18–21
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619