
Diffusion-Driven Domain Adaptation for Generating 3D Molecules

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 *Can we train a molecule generator that can generate 3D molecules from a new*
2 *domain, circumventing the need to collect data?* This problem can be cast as the
3 problem of domain adaptive molecule generation. This work presents a novel
4 and principled diffusion-based approach, called *GADM*, that allows shifting a
5 generative model to desired new domains without the need to collect even a single
6 molecule. As the domain shift is typically caused by the structure variations of
7 molecules, e.g., scaffold variations, we leverage a designated equivariant masked
8 autoencoder (MAE) along with various masking strategies to capture the structural-
9 grained representations of the in-domain varieties. In particular, with an asymmetric
10 encoder-decoder module, the MAE can generalize to unseen structure variations
11 from the target domains. These structure variations are encoded with an equivariant
12 encoder and treated as domain supervisors to control denoising. We show that, with
13 these encoded structural-grained domain supervisors, *GADM* can generate effective
14 molecules within the desired new domains. We conduct extensive experiments
15 across various domain adaptation tasks over benchmarking datasets. We show that
16 our approach can improve up to 65.6% in terms of success rate defined based on
17 molecular validity, uniqueness, and novelty compared to alternative baselines.

1 Introduction

19 Geometric generative models are proposed to approximate the distribution of complex geometries
20 and are used to generate feature-rich geometries. They have emerged as a crucial research direction
21 in various scientific fields (e.g., material science, biology, and chemistry [16, 47, 50]), attempting to
22 facilitate the process of scientific knowledge discovery. In these fields, geometries could be point
23 clouds where each point is embedded in the Cartesian coordinates and encompasses rich features.
24 For example, 3D molecules can be represented as atomic geometric graphs [16, 51, 40].

25 There has been fruitful research progress on 3D molecule generation based on geometric generative
26 modeling due to their ability to estimate density and generate feature-rich geometries. Recent
27 representative models for generating 3D molecules in silicon include autoregressive [28], flow-
28 based models [11], and diffusion models [16]. Among others, diffusion models have demonstrated
29 their superior performance in terms of various empirical evaluation metrics, such as stability and
30 validity [16]. However, these generative models are trained to mimic the training data distribution,
31 limiting their capability within the in-domain generation and manipulation [13], *i.e.*, controllable
32 generation.

33 With the expressive power of the state-of-the-art diffusion-based generators, *can we train a diffusion-*
34 *based molecule generator that can flexibly adapt to a desired new domain where data are scarce*
35 *and difficult to collect?* This problem can be cast as a domain adaptive generation problem, whose
36 goal is to shift the data distribution of generators to a desired new domain different from what it is
37 trained over. In the context of molecule generation, the distribution shift mainly comes from structure
38 variations [49, 25]. The structure variation could be the various types of scaffolds or ring-structures.

39 Taking a canonical molecule dataset – QM9 as our running
 40 example, diverse scaffolds of molecules have varying pro-
 41 portions in nature [32, 49]. We observed that EDM [16]
 42 and GeoLDM [51] indeed could capture the training data
 43 distribution well — generating molecules with scaffolds
 44 existing in the high-frequency class — but they struggle
 45 to generate molecules with low-frequency scaffolds (see
 46 Table 1). Our preliminary study proves the excellent ex-
 47 pressive capability of the current diffusion-based molecule
 48 generators. On the other hand, it indicates the difficulty in
 49 generating molecules deviating from the training data dis-
 50 tribution. Existing works for domain adaptive generation
 51 are tailed for specific generation tasks, such as image [39],
 52 dialog [31], and question-answering generation [54]. As
 53 far as we know, ours is the first work to consider domain
 54 adaptive generation for 3D molecules.

55 To address the above issues, we develop a new and princi-
 56 pled diffusion-based generator, called **Geometric Adaptive**
 57 **Diffusion Model (GADM)**, that can adaptively synthesize
 58 3D molecules in the desired new domains. In particular, **GADM** enables the generation of 3D
 59 molecules with structural-grained variations adaptively, including the distribution shifts due to scaffold
 60 and ring-structure variations, respectively. The underlying assumption is that if we can capture
 61 the set of structure variations right, generalizing the unseen ones that ultimately lead to the target
 62 domain is a much easier process. Using QM9 as an example (see Table 1, source, target I, and
 63 target II are three domains due to scaffold variations. Our proposed generator **GADM** is trained with
 64 source molecules — the high-frequency scaffolds. Once trained, **GADM** can generate molecules
 65 with low/rare-frequency scaffolds conditioned on corresponding scaffolds in target I/target II.

66 The crux of **GADM** is to empower the denoising process with domain priors, which is characterized by
 67 a designated **Equivariant Masked Autoencoder (EMAE)**. Our **EMAE** is realized with an asymmetric
 68 encoder-decoder architecture, enabling to capture the domain priors — in-domain structure variations
 69 and to generalize to out-of-domain structure variations [14]. More specifically, during training, the
 70 in-domain priors, such as scaffolds or ring-structures from the source domain, are encoded and
 71 subsequently act as domain supervisors to control the denoising process of **Domain Supervised**
 72 **Diffusion Model (DSDM)**. In the generation phase, the generalization capability provided by the
 73 asymmetric **EMAE** allows for properly encoding the unseen structure variations, i.e., scaffolds or
 74 rings from the target domains. These captured target domain priors are used to control the denoising
 75 process to generate 3D molecules within the desired new domains.

76 To ensure that the generated 3D molecules are SE(3)-equivariant, our **EMAE** employs the well-
 77 known equivariant graph neural network module to encode the structural-grained domain supervisors.
 78 Notably, unlike prior domain adaption works [39], **GADM** does not need additional training for the
 79 entire adaptive generation process. In a nutshell, our main contributions are delineated as follows.

80 *First*, we pioneer the domain adaptive generation problem in the context of 3D molecule generation.
 81 Correspondingly, we propose a geometric adaptive diffusion-based generation framework capable of
 82 adaptively generating target molecules outside the training domain without additional training. In
 83 particular, we adopt the idea of **Masked Autoencoder (MAE)** to extract latent features of in-domain
 84 and out-of-domain supervisors for conditional denoising in diffusion models. *Second*, we proved that
 85 the domain supervisor extracted by the designed **EMAE** is SE(3)-equivariant, ensuring the molecular
 86 generation is equivariant. *Third*, to validate the effectiveness of the proposed framework, we compare
 87 it with EDM [16] and GeoLDM [51] over benchmarking datasets. Extensive experimental results
 88 demonstrate that the latent features, acting as domain supervisors, empower the diffusion models
 89 to generate molecules with desired structural variations adaptively. Remarkably, the success rate
 90 of generated molecules by **GADM** is improved by up to 65.6% compared with existing methods.
 91 Our work represents a significant advancement in generating novel molecules that are absent in the
 92 training samples but exhibit the desired structural variations.

Table 1: Alternative baselines were trained with QM9, a canonical molecule dataset. Source, target I, and target II domains encompass molecules with high-, low-, and rare-frequency scaffolds, respectively. The generated samples from EDM and GeoLDM, which are trained on molecules with source scaffolds, are dominated by the training scaffold set, indicating that they can well reflect the training data distribution.

Domains	QM9 Scaffold Proportion (%)		
	Source	Target I	Target II
QM9	76.4	11.5	12.1
EDM [16]	90.9	5.9	2.7
GeoLDM [51]	90.6	5.9	3.5

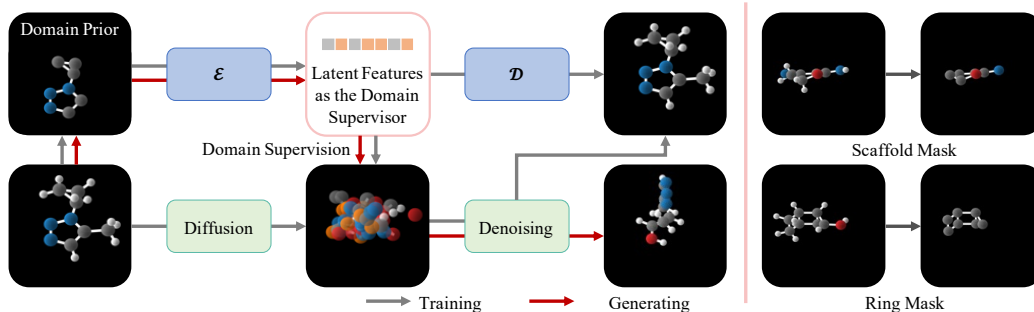


Figure 1: The Illustration of Proposed GADM Framework.

During training (gray pipeline): **I.** Equivariant Masked Autoencoder (EMA): the equivariant encoder (\mathcal{E}) first maps the domain prior—masked structure (i.e., scaffold/ring)—into the masked latent features. These latent features would be processed with an equivariant decoder (\mathcal{D}) for reconstructing the original molecule in 3D atomic space. This asymmetric encoder-decoder architecture enables to capture of the in-domain priors and to generalize to out-of-domain structures; **II.** Domain Prior-Supervised Diffusion Model (DSDM): DSDM first diffuses the molecule into noises and then incorporates the masked latent features as domain supervisor to perform denoising for reconstructing the input molecules. **During generation (red pipeline):** EMA receives the target domain prior and encodes it as the domain supervisor. Then, DSDM denoises from sampled Gaussian noise under domain supervision to generate novel and valid molecules with target structure variations.

93 2 Problem Setup and Preliminaries

94 2.1 Problem Definition

95 **Notations:** Let d be the dimensionality of node features; a 3D molecule can be represented as a point
 96 cloud denoted as $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$, where $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{N \times 3}$ is the atom coordinate matrix and
 97 $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_N) \in \mathbb{R}^{N \times d}$ is the node feature matrix containing atomic type, charge features, etc.
 98 For a given molecule \mathcal{G} , the scaffold is its structural framework [4], termed as “chemotypes,” which
 99 could be regarded as a subgraph of the original molecule, represented as $\mathcal{G}^s = \langle \mathbf{x}^s, \mathbf{h}^s \rangle$. Except for
 100 scaffolds, the ring structures are essential in Chemistry and Biology [20, 46, 33], which could also be
 101 a factor that incurs the distribution shift.

102 **Domain Adaptive Generation Problem:** We consider the problem of domain adaptive generation in
 103 the following two scenarios, including scaffold-domain and ring-structure-domain adaptive generation,
 104 respectively. Given a collection of molecules as training samples and corresponding scaffold/ring-
 105 structure set denoted as $\{\mathcal{G}_S\}$, $\{\mathcal{G}_S^s\}$, respectively. For simplicity, we call the training sample domain
 106 as the source domain. Domain adaptive generation aims to learn a generative model that can generate
 107 valid and novel molecules falling into a targeted new domain, where corresponding scaffold/ring-
 108 structure set is $\{\mathcal{G}_T^s\}$, and the targeted scaffold/ring-structure set is unseen during training, a.k.a.
 109 $\{\mathcal{G}_S^s\} \cap \{\mathcal{G}_T^s\} = \emptyset$.

110 2.2 Preliminaries

111 **Equivariance.** Molecules, typically existing within a three-dimensional physical space, are subject
 112 to geometric symmetries, including translations, rotations, and potential reflections. These are
 113 collectively referred to as the Euclidean group in 3 dimensions, denoted as $E(3)$ [6].

114 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
 115 $g \in G$, where S_g, T_g are linear representations related to the group element g [36]. For geometric
 116 graph generation, we consider the special Euclidean group $SE(3)$, involving translations and rotations.
 117 Moreover, the transformations S_g or T_g can be represented by a translation \mathbf{t} and an orthogonal
 118 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
 119 coordinates \mathbf{x} are $SE(3)$ -equivariant, which can be expressed as $\mathbf{R}\mathbf{x} + \mathbf{t} = (\mathbf{R}\mathbf{x}_1 + \mathbf{t}, \dots, \mathbf{R}\mathbf{x}_N + \mathbf{t})$.

120 **Diffusion Models.** Diffusion models [38] are latent variable models for learning distributions by
 121 modeling the reverse of a diffusion process [15]. Given a data point $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and a variance
 122 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} :

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

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 approximately restore the data point \mathbf{x}_0 in the original data distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \quad (2)$$

where the initial distribution $p(\mathbf{x}_t)$ is sampled from standard Gaussian noise $\mathcal{N}(0, \mathbf{I})$. The loss for training diffusion model $\mathcal{L}_{\text{DM}} := \mathcal{L}_t$ is simplified as: $\mathcal{L}_{\text{DM}} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2]$, 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 $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$. We provide detailed description about diffusion models in Appendix C.

3 Method

Overview. Our objective is to learn a generator with the source domain with rich data that can flexibly adapt to a new domain in a low-data regime. Generally, structure variations, such as scaffold or ring-structure variations, are the main cause of the domain shift in the context of molecule generation [49, 25]. We particularly focus on the geometric adaptive generation problem where the scaffold/ring-structure set of the source domain, represented as $\{\mathcal{G}_S^s\}$, and the targeted scaffold/ring-structure set from new domains, denoted as $\{\mathcal{G}_T^s\}$, are different. In other words, the targeted scaffold/ring-structure set of the target domain is unseen during training — $\{\mathcal{G}_S^s\} \cap \{\mathcal{G}_T^s\} = \emptyset$.

With the superior capability of diffusion models for 3D molecule generation, we propose to address the geometric domain adaptive molecule generation problem with a diffusion engine. However, as illustrated in Section 1, the vanilla diffusion models have difficulty generating out-of-domain molecules. In this regard, we propose to incorporate the structure variations of the source domain into the denoising process during training and those of target domains into the denoising during generation. These structure variations are dubbed as domain priors or domain supervisors. Nevertheless, characterizing the domain priors that can adapt to new domains is challenging because the domain priors of the target domains are not seen during training. Inspired by the impressive generalizability of masked autoencoder in both vision and language fields [14, 18], we adopt an asymmetric encoder-decoder architecture to capture the domain priors of the source domain and to generalize to unseen structure variations from the target domains.

In what follows, we will elaborate on the design details of equivariant masked autoencoder and domain prior-supervised diffusion model in Section 3.1 and Section 3.2, respectively. Then, we will briefly summarize the training scheme and domain adaptive molecule generation in Section 3.3. The proposed GADM workflow is provided in Figure 1.

3.1 Equivariant Masked Autoencoder

Masking. For a given molecule $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$, we apply various masking strategies (\mathcal{M}) to derive the visible structure $\mathcal{G}^V = \langle \mathbf{x}^V, \mathbf{h}^V \rangle \leftarrow \mathcal{M}(\mathcal{G})$ for distinct adaptive molecule design tasks, as depicted in the right section of Figure 1. In the case of scaffold-domain and ring-domain adaptive design, we mask (*i.e.*, remove) the atoms not present on the scaffold/rings. This process is expressed as $\mathcal{G}^V \leftarrow \langle \mathbf{x} - \mathbf{x}^s, \mathbf{h} - \mathbf{h}^s \rangle$.

Variational Autoencoder. The EMAE comprises an encoder \mathcal{E} , which maps visible structure \mathcal{G}^V to a latent space, represented as $\mathbf{f}_x, \mathbf{f}_h = \mathcal{E}(\mathbf{x}^V, \mathbf{h}^V)$. Additionally, it includes a decoder \mathcal{D} that reconstructs the latent representation back to the original molecular space, denoted as $\hat{\mathbf{x}}, \hat{\mathbf{h}} = \mathcal{D}(\mathbf{f}_x, \mathbf{f}_h)$.

Our EMAE reconstructs the input by predicting the coordinates and features of each masked atom. The loss function computes the mean squared error (MSE) between the reconstructed and original molecules in the original molecular space. The EMAE can be trained by minimizing the reconstruction objective, expressed as $f(\mathcal{G}, \mathcal{D}(\mathcal{E}(\mathcal{M}(\mathcal{G}))))$. The encoder of the EMAE functions solely on the visible structure $\mathcal{M}(\mathcal{G})$, while the decoder reconstructs the input from the latent representation to the

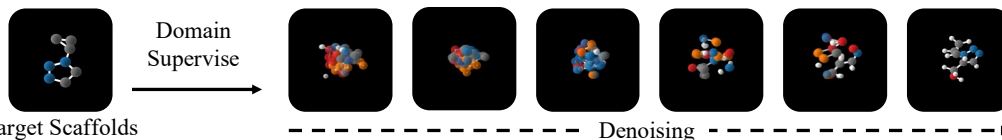


Figure 2: *The Illustration of the Adaptive Generation Process with GADM*: given a scaffold as the domain supervisor from a new domain, our trained GADM can generate valid, unique, and novel molecules containing the target scaffold.

complete molecule \mathcal{G} . This asymmetric encoder-decoder design offers promising generalization [14] to the latent features. These features serve as domain supervisors and empower the model to generate molecules with unseen domain priors.

Equivariant MAE. However, applying general MAE in the geometric domain is non-trivial. The diffusion model within the overall framework operates in 3D molecular space and necessitates conditions to be either equivariant or invariant. Therefore, it is crucial to ensure the equivariance of the conditions extracted by EMAE. To achieve this, we design our EMAE based on the Equivariant Graph Neural Networks (EGNNs) [35], thereby incorporating equivariance into both the encoder \mathcal{E}_ϕ and decoder \mathcal{D}_ϑ , where ϕ and ϑ are two learnable EGNNs. EMAE ensures that the latent representation \mathbf{f}_x and \mathbf{f}_h encoded by the encoder from visible structure are 3-D equivariant and k -d invariant, respectively. Consequently, EMAE extracts both invariant and equivariant conditions, as expressed below:

$$\mathbf{R}\mathbf{f}_x + \mathbf{t}, \mathbf{f}_h = \mathcal{E}_\phi(\mathbf{R}\mathbf{x}^V + \mathbf{t}, \mathbf{h}^V) \quad (3)$$

$$\mathbf{R}\hat{\mathbf{x}} + \mathbf{t}, \hat{\mathbf{h}} = \mathcal{D}_\vartheta(\mathbf{R}\mathbf{f}_x + \mathbf{t}, \mathbf{f}_h) \quad (4)$$

for all rotations \mathbf{R} and translations \mathbf{t} . Detailed architecture information about EMAE can be found in Appendix D. The point-wise latent space adheres to the inherent structure of geometries \mathcal{G}^V , which facilitates learning conditions for the diffusion model and results in high-quality molecule design.

Following [16, 51], to ensure that linear subspaces with the center of gravity always being zero can induce translation-invariant distributions, we define distributions of visible structures \mathbf{x}^V , latent conditions \mathbf{f}_x , and reconstructed $\hat{\mathbf{x}}$ on the subspace that $\sum_i \mathbf{x}_i^V$ (or $\mathbf{f}_{x,i}$ and $\hat{\mathbf{x}}_i$) = 0. Then the encoding and decoding processes can be formulated by $q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h}) = \mathcal{N}(\mathcal{E}_\phi(\mathcal{M}(\mathbf{x}, \mathbf{h})), \sigma_0 \mathbf{I})$ and $p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{f}_x, \mathbf{f}_h) = \prod_{i=1}^N p_\vartheta(x_i, h_i | \mathbf{f}_x, \mathbf{f}_h)$ and the EMAE can be optimized by:

$$\mathcal{L}_{\text{EMAE}} = \mathbb{E}_{q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h})} p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{f}_x, \mathbf{f}_h) - \text{KL}[q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h}) || \prod_i^N \mathcal{N}(f_{x,i}, f_{h,i} | 0, \mathbf{I})], \quad (5)$$

where $-\mathbb{E}_{q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h})} p_\vartheta(\mathbf{x}, \mathbf{h} | \mathbf{f}_x, \mathbf{f}_h)$ is the reconstruction loss and is calculated as L_2 norm or cross-entropy for continuous or discrete features. $\text{KL}[q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h}) || \prod_i^N \mathcal{N}(f_{x,i}, f_{h,i} | 0, \mathbf{I})]$ is a regularization term between q_ϕ and standard Gaussians. $\mathcal{L}_{\text{EMAE}}$ is standard VAE loss and is the variational lower bound of log-likelihood. The equivariance of the loss, which is crucial for geometric graph generation, is expressed as follows:

Theorem 3.1. $\mathcal{L}_{\text{EMAE}}$ is an $SE(3)$ -invariant variational lower bound to the log-likelihood, i.e., for any geometries $\langle \mathbf{x}, \mathbf{h} \rangle$, we have:

The theorem ensures that EMAE is equivariant so that the extracted condition satisfies the equivariant constraints, thereby ensuring that the conditional denoising of the geometric diffusion model is also equivariant. Detailed proof of Theorem 3.1 is given in Appendix F.

In summary, EMAE first masks the input molecule \mathcal{G} , and then inputs the visible structure \mathcal{G}^V into the encoder \mathcal{E} to obtain equivariant latent features \mathbf{f}_x and invariant latent features \mathbf{f}_h . These features have two purposes. One is to continue to be input into the decoder \mathcal{D} for reconstruction to constrain the latent features. Secondly, it is used as the condition to supervise and control the diffusion model. The specific method of the second part will be explained in the following section.

3.2 Domain Prior-Supervised Diffusion Model

With the equivariant latent features $\langle \mathbf{f}_x, \mathbf{f}_h \rangle$, now we can utilize these features as domain supervisors for reconstructing structures \mathcal{G} while still keeping geometric properties. The latent features encoded

by the mask encoder from the same molecule serve as the condition for the diffusion model. Such a similar manner to self-supervised learning enables the model to generate molecules with target structural variations, and thereby, the proposed method can perform adaptive molecule generation.

Generally, geometric diffusion models are capable of controllable generation with given conditions s by modeling conditional distributions $p(\mathbf{z}|s)$. This modeling in DMs can be implemented with conditional denoising networks $\epsilon_\theta(\mathbf{z}, t, s)$ with the critical difference that it takes additional inputs s . However, an underlying constraint of such use is the assumption that s is invariant. By contrast, a fundamental challenge for our method is that the conditions for the DM contain not only invariant features \mathbf{f}_h but also equivariant features \mathbf{f}_x . This requires the distribution $p_\theta(\mathbf{z}_{0:T})$ of our DMs to satisfy the critical invariance:

$$\forall \mathbf{R}, p_\theta(\mathbf{z}_x, \mathbf{z}_h, \mathbf{f}_x, \mathbf{f}_h) = p_\theta(\mathbf{R}\mathbf{z}_x, \mathbf{z}_h, \mathbf{R}\mathbf{f}_x, \mathbf{f}_h). \quad (6)$$

To achieve this, we should ensure that (1) the initial distribution $p(\mathbf{z}_{x,T}, \mathbf{z}_{h,T}, \mathbf{f}_x, \mathbf{f}_h)$ is invariant, which is already satisfied since $\mathbf{z}_{x,T}$ is projected down by subtracting its center of gravity after sampling from standard Gaussian noise. With the $\mathbf{f}_x, \mathbf{f}_h$ is obtained by equivariant \mathcal{E}_ϕ (Equations 3); (2) the conditional reverse processes via θ , which is expressed as $p_\theta(\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1}|\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, \mathbf{f}_x, \mathbf{f}_h)$, are equivariant:

$$\forall \mathbf{R}, p_\theta(\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1}|\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, \mathbf{f}_x, \mathbf{f}_h) = p_\theta(\mathbf{R}\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1}, |\mathbf{R}\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, \mathbf{R}\mathbf{f}_x, \mathbf{f}_h), \quad (7)$$

this can be realized by implementing the denoising dynamics ϵ_θ with EGNN that satisfy the following equivariance:

$$\forall \mathbf{R} \text{ and } \mathbf{t}, \mathbf{R}\mathbf{z}_{x,t-1} + \mathbf{t}, \mathbf{z}_{h,t-1} = \epsilon_\theta(\mathbf{R}\mathbf{z}_{x,t} + \mathbf{t}, \mathbf{z}_{h,t}, \mathbf{R}\mathbf{f}_x + \mathbf{t}, \mathbf{f}_h, t), \quad (8)$$

In order to keep translation invariance, all the intermediate states $\mathbf{z}_{x,t}, \mathbf{z}_{h,t}$ are also required to lie on the subspace by $\sum_i \mathbf{z}_{x,t,i} = 0$ by moving the center of gravity. Analogous to Equation 17, now we can train the **DSDM** by:

$$\mathcal{L}_{\text{DSDM}} = \mathbb{E}_{\mathcal{G}, \mathcal{E}(\mathcal{M}(\mathcal{G})), \epsilon, t} [\|\epsilon - \epsilon_\theta(\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, \mathbf{f}_x, \mathbf{f}_h, t)\|^2] \quad (9)$$

with $w(t)$ simply set as 1 for all steps t .

3.3 Training and Generation

Training. The training loss of the entire framework can be formulated as $\mathcal{L} = \mathcal{L}_{\text{EMAE}} + \mathcal{L}_{\text{DSDM}}$. To make the training loss tractable, we also show that \mathcal{L} is theoretically an SE(3)-invariant variational lower bound of the log-likelihood and we can have:

Theorem 3.2. *Let $\mathcal{L} := \mathcal{L}_{\text{EMAE}} + \mathcal{L}_{\text{DSDM}}$. With certain weights $w(t)$, \mathcal{L} is an SE(3)-invariant variational lower bound to the log-likelihood.*

Given the above training loss and Theorem 3.2, we can optimize **GADM** via back-propagation with reparameterizing trick [22]. We provide the detailed proof of Theorem 3.2 in Appendix G, and a formal description of the optimization procedure in Algorithm 1 in Appendix H. We follow the process of EDM [16] regarding the representation for continuous features \mathbf{x} and categorical features \mathbf{h} . For clarity, we provided the details in Appendix D.3.

Adaptive Molecule Generation. With **GADM** trained on source dataset $\{\mathcal{G}_S\}$ and given a scaffold/ring-structure from the target domain, denoted as a \mathcal{G}_T^s , we can perform adaptive molecule generation (a scaffold adaptive generative process is illustrated in Figure 2). To sample from the model, one first inputs the \mathcal{G}_T^s into the encoder \mathcal{E}_ϕ and obtains the latent representation of \mathcal{G}_T^s denoted as $\langle \mathbf{f}_x, \mathbf{f}_h \rangle$ via reparameterization. With the latent representation of the target domain prior as condition, **DSDM** first samples $\mathbf{z}_{x,T}, \mathbf{z}_{h,T} \sim \mathcal{N}_{x,h}(\mathbf{0}, \mathbf{I})$ and then iteratively samples $\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1} \sim p_\theta(\mathbf{z}_{x,t-1}, \mathbf{z}_{h,t-1}|\mathbf{z}_{x,t}, \mathbf{z}_{h,t}, \mathbf{f}_x, \mathbf{f}_h)$. Finally, the output molecule represented as $\langle \mathbf{x}, \mathbf{h} \rangle$ is sampled from $p(\mathbf{z}_{x,0}, \mathbf{z}_{h,0}|\mathbf{z}_{x,1}, \mathbf{z}_{h,1}, \mathbf{f}_x, \mathbf{f}_h)$. The pseudo-code of the adaptive generation is provided in Algorithm 2 in Appendix H.

4 Experiments

4.1 Experiment Setup

Datasets and Tasks. We evaluate over QM9 [32] and the GEOM-DRUG [1]. Specifically, QM9 is a standard dataset that contains molecular properties and atom coordinates for 130k 3D molecules with

up to 9 heavy atoms and up to 29 atoms, including hydrogens. GEOM-DRUG encompasses around 450,000 molecules, each with an average of 44 atoms and a maximum of 181. Dataset details and experimental parameters are presented in Appendices A, B, and E.

Ring-Structure-Domain Adaptive Molecule Generation. In this task, ring-structure variations result in distribution shifts. We used RDKit [24] to categorize molecules into 9 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 source domain comprises molecules and those with 0 to 3 rings, and we consider the target domains including molecules with 4 to 8 rings, respectively. Figure 6 in the Appendix presents a schematic diagram illustrating example molecules with 0 to 8 rings. The GEOM-DRUG dataset contains molecules with 0 to 14 rings and 22 rings. We use the subsets with 0 to 10 rings as the source domain and consider five target domains, including 11 to 14 and 22. This is because the number of molecules possessing 11 to 14 and 22 rings are all under 100, representing a micro fraction of the total molecule count.

Scaffold-Domain Adaptive Molecule Generation. In this task, scaffold variations incur distribution shifts. Similarly, we utilized RDkit [24] to examine the scaffold of each molecule within the QM9 dataset. Molecules lacking a scaffold were denoted as ‘-’ and were included in the total scaffold count. The entire dataset was divided based on scaffold frequency. Specifically, the source domain contained 100,000 molecules and 1,054 scaffolds — most scaffolds appeared at least 100 times. The target domain I included 15,000 molecules and 2,532 scaffolds, where most scaffold’s frequency is between 10 to 100. The target domain II consisted of 15,831 molecules and 12,075 scaffolds; each scaffold’s frequency is less than 10. We aim to learn a generative model with the source domain training data, which can adaptively generate effective molecules that fall into desired new domains, such as target domain I/II.

Baselines. Our work is the first to consider the problem of domain adaptive generation for 3D molecules, leading to the absence of baselines for a comprehensive comparison. As alternatives, we employ three state-of-the-art 3D molecule diffusion models, EDM [16], GeoLDM [51] and EEGSDE [3], as baselines to validate the efficacy of our proposed GADM. These methods can perform controllable generation but can only control the generation process with numerical features. Intuitively, the number of rings could be a numerical feature of a molecule. We treat the ring counts as one control factor to manipulate the generation process of the baselines, denoted as C-EDM, C-GeoLDM, and EEGSDE to verify GADM’s effectiveness in the ring-structure domain adaptive generation task (see Table 2).

Metrics. Our objective is to generate effective 3D molecules in a target new domain. A generated sample is effective only when it falls into the target domain while it is valid, unique, and novel simultaneously. Therefore, our evaluation metrics can be defined as follows:

1. **Proportion (P):** Given a target scaffold/ring set $\{\mathcal{G}_T^s\}$, proportion describes the percentage of molecules that contain the desired scaffold/ring-structure in $\{\mathcal{G}_T^s\}$ among generated valid samples; 2. **Coverage (C):** Coverage describes the percentage of scaffolds set of the generated samples (denoted as $\{\mathcal{G}_G^s\}$) in target scaffolds set $\{\mathcal{G}_T^s\}$, which is expressed as $C = |\{\mathcal{G}_G^s\}|/|\{\mathcal{G}_T^s\}|$; 3. **Target validity (V):** The percentage of valid molecules among all the desired molecules, which is measured by RDkit [24] and widely used for calculating validity [16, 51]; 4. **Target novelty (N):** The percentage of novel molecules among all the desired valid molecules, the novel molecule is different from training samples; 5. **Success rate (S):** The ratio of generated valid, unique, and novel molecules that contain the desired scaffold/ring-structure. 6. **Target atom stability (AS):** The ratio of atoms that has the correct valency with the desired scaffold/ring-structure among all generated molecules. 7. **Target molecule stability (MS):** The ratio of generated molecules contains the desired scaffold/ring-structure, and all atoms are stable. GEOM-DRUG dataset has nearly 0% molecule-level stability, so this metric is generally ignored on GEOM-DRUG [16].

4.2 Results and Analysis

Ring-Structure Domain Adaptive Molecule Generation. In this task, all models were trained with the same source domain that contains molecules with ring counts ranging from 0 to 3. Subsequently, their performances were tested for generating molecules with 4 to 8 rings, respectively. We present the results on 10,000 generated molecules for each ring-count domain in Table 2. For clarity, the generated

Table 2: Results of molecule proportion in terms of ring-number (P), molecule validity (V), novelty (N), and success rate (S). The **best** results are highlighted in bold. QM9 only contains 36 eight-ring molecules, and the proportion for eight-ring is nearly 0.

Metrics Domains	P (%) in Source Domain				P (%) in Target Domains					AS (%)	MS (%) Averaged over 9 Domains	V (%)	N (%)	S (%)
	0	1	2	3	4	5	6	7	8					
QM9	10.2	39.3	27.6	15.1	4.4	2.7	0.6	0.2	0.0	99.0	95.2	97.7	-	-
EDM† [16]	10.5	39.8	28.0	14.5	4.0	2.9	0.2	0.1	0.0	11.0	9.6	10.4	6.8	6.3
GeoLDM† [51]	12.0	38.6	27.0	15.3	4.6	2.2	0.2	0.1	0.0	11.0	9.9	10.4	6.4	5.9
EDM‡ [16]	12.1	44.1	29.8	11.8	1.7	0.5	0.0	0.0	0.0	11.0	9.7	10.4	6.8	6.3
GeoLDM‡ [51]	2.8	41.5	32.1	15.7	4.7	2.7	0.3	0.1	0.0	10.9	9.1	10.4	6.7	6.2
C-EDM† [16]	98.9	94.2	80.8	64.4	12.6	26.8	0.3	0.1	0.0	41.3	33.9	38.0	27.3	24.1
C-GeoLDM† [51]	97.1	89.4	74.2	52.4	22.3	22.7	0.9	0.2	0.0	39.1	31.5	35.7	28.3	25.0
EEGSDE† [3]	98.4	92.2	77.6	58.2	14.1	17.6	0.3	0.0	0.0	39.1	31.1	35.7	27.2	24.2
GADM‡	99.9	99.8	99.1	97.6	92.5	89.7	78.7	88.2	82.1	83.1	54.0	77.9	70.3	40.5

†: Models are trained over entire QM9;

‡: Models are trained over ring-split QM9 with ring-number from 0-3.

C-: C-EDM and C-GeoLDM are trained with conditioning on ring counts.

target molecule validity, novelty, and success rate are calculated by averaging the corresponding values from the source domain and 5 target domains. More comprehensive results are presented in Appendix I.

Table 2 demonstrates that those uncontrollable version of baselines (i.e., EDM and GeoLDM) can barely generate molecules with 4 to 8 rings — 4.6% at most. Manipulating the generation process with ring counts can slightly improve out-of-domain generation performance with up to 25% success rates. In contrast, **GADM** can achieve a 40.5% success rate. Moreover, we observe that no baselines can generate 8-ring molecules, including those controllable generation methods (i.e., C-GeoLDM, C-EDM, and EEGSDE), reflecting the difficulty of generating those complex molecules rare existing in the original QM9 (only 36 8-ring molecules). Notably, **GADM** can generate 82.1% portion of 8-ring domain molecules even though the training data does not contain any of those samples, showing the significance of using structural-grained representations for controlling the denoising process of the diffusion models. Specifically, among the generated 10,000 molecules using **GADM**, 2,388 valid, unique, and novel 8-ring molecules exist. These results verify that **GADM** can adaptively generate 3D molecules from the desired new domains regarding ring-structure variations.

Table 3 presented the statistical results of various methods for generating rare ring number molecules (ranging from 11 to 14 and 22) on the large-scale dataset GEOM-DRUG. Notably, EDM and GeoLDM, trained on the complete dataset, cannot generate molecules with ring numbers exceeding 10, thus failing to produce any desired molecules. In contrast, **GADM** can generate an average of 13.8% of the desired molecules. Particularly, for molecules with 22 rings, of which there are only two in the original dataset, **GADM** achieves a remarkable success rate of 13.7% in generating such molecules, even without training on these two molecules.

Table 3: Results of molecule proportion in terms of ring-number (P), atom stability (AS), molecule validity (V), novelty (N), and success rate (S). The number of molecules with above 11 rings in GEOM-DRUG is lower than 100.

Averaged metrics (%) over 5 Ring Domains (11, 12, 13, 14, and 22)					
Method	P (%)	AS (%)	V (%)	N (%)	S (%)
GEOM-DRUG	0.0	86.5	99.9	-	-
EDM† [16]	0.0	0.0	0.0	0.0	0.0
GeoLDM† [51]	0.0	0.0	0.0	0.0	0.0
GADM‡	13.8	11.4	11.0	13.8	10.9

† Models are trained on complete GEOM-DRUG.

‡ Models are trained on GEOM-DRUG with ring numbers from 0-10.

Scaffold-Domain Adaptive Molecule Generation. In the task of scaffold-domain adaptive molecule generation, the baselines are trained on both the entire dataset (†) and solely on the source domain (‡), respectively. In contrast, our **GADM** is trained exclusively over the source domain dataset. After training, each model generates 15,000 molecules for the source and target domains I and II, respectively. The quantitative results using various metrics are presented in Table 4, Table 5 and Figure 3. We observe that with EDM or GeoLDM, the scaffold proportion of the generated molecules indeed mirrors that of the training samples (see proportion and coverage visualization in Figure 3). However, they all struggle to generate molecules with scaffolds falling into targeted domain I or II; they can only achieve 3.3% success rates at most (see EDM‡ and GeoLDM‡ in Table 4). In contrast, our proposed **GADM**, trained solely on the source domain, can generate molecules containing the target scaffolds under the corresponding supervision, achieving at least 95.5% proportion in both new domains. Note that the target scaffolds were not seen during training.

Table 4: Results of proportion (P), scaffold coverage (C), molecule validity (V), molecule novelty (N), and molecule success rate (S). The **best** results are highlighted in bold.

Domains	Source Domain (%)					Target Domain I (%)					Target Domain II (%)				
# Metric	P	C	V	N	S	P	C	V	N	S	P	C	V	N	S
Data	76.4	100.0	97.7	-	-	11.5	100.0	97.7	-	-	12.1	100.0	97.7	-	-
EDM† [16]	79.9	36.3	74.8	48.8	45.0	10.9	28.9	10.2	6.7	6.1	9.2	34.9	8.6	5.6	5.2
GeoLDM† [51]	80.4	35.2	75.6	46.7	43.1	10.7	31.2	10.1	6.2	5.8	8.8	33.5	8.3	5.1	4.7
EDM‡ [16]	91.4	56.5	83.2	58.2	52.0	5.9	26.5	5.3	3.7	3.3	2.7	17.0	2.4	1.7	1.5
GeoLDM‡ [51]	90.6	54.3	81.7	57.8	51.0	5.9	26.7	5.3	3.8	3.3	3.5	19.0	3.2	2.3	2.0
GADM‡	99.2	92.5	90.7	67.6	52.4	97.0	97.1	80.0	84.5	68.9	95.5	85.7	83.3	82.0	65.8

† Models are trained over the entire QM9 dataset.

‡ Models are trained only on the source domain, where each scaffold appears at least 100 times.

Table 5: Results of atom stability (AS) and molecule stability (MS). The **best** results are highlighted in bold.

Domains	Source		Target I		Target II	
# Metric (%)	AS	MS	AS	MS	AS	MS
Data	99.0	95.2	99.0	95.2	99.0	95.2
EDM† [16]	78.9	65.5	10.8	8.9	9.1	7.5
GeoLDM† [51]	79.5	71.9	10.6	9.6	8.7	7.9
EDM‡ [16]	90.4	73.3	5.8	4.7	2.6	2.1
GeoLDM‡ [51]	89.1	75.6	5.8	4.9	3.5	3.0
GADM‡	96.1	71.3	89.5	45.6	89.0	35.1

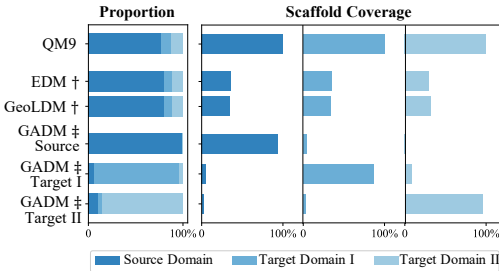


Figure 3: Scaffolds Proportion and Coverage.

Moreover, we found that **GADM** can reach 92.5% coverage for the in-domain generation with the in-domain supervisor — structural-grained representations from the latent space of **EMAE**. Notably, even for target domain II, comprising over 12,000 different rare scaffolds, **GADM** can achieve 85.7% coverage. Nevertheless, all baselines can only achieve 56.6% coverage at most, indicating the significance of our **EMAE**. It is worth noting that **GADM** does not need any target molecules but uses the scaffold as the domain supervisor for cross-domain adaptation, bypassing the obstacles due to data scarcity. **GADM** improves the molecule novelty and success rate by up to 80.8% regarding novelty and 65.6% in terms of success rate as compared to the baselines. The atom stability and molecule stability presented in Table 5 also demonstrates that the designed **GADM** performs better on generating chemically stable molecules with desired scaffolds.

Discussion. Our experiments show that existing generative models may still generate a slight portion of out-of-domain molecules, as shown in Table 2 and 4. This phenomenon could be attributed to the fact that scaffolds/ring-structures in different domains might be mutually inclusive or share substructures. Consequently, the generated molecules may contain substructures or compound substructures derived from the training samples, constituting unseen scaffolds/ring-structures. A detailed illustration is provided in Appendix K. We want to point out that such out-of-domain generation is relatively non-trivial. Our proposed **GADM** underscores the significant potential in generating molecules with targeted structural variations, including scaffolds and ring-structures.

Limitations. Most generative models, including ours, adopt the EGNN modules to capture the equivariance of molecules [16, 51]. The model’s memory overhead escalates exponentially with the size of the input molecules, posing a significant constraint, especially for generating large molecules. A comprehensive analysis and discussion are furnished in Appendix M.

5 Conclusion

This paper introduced the problem of domain adaptive molecule generation, which entails the ability of a trained diffusion-based generator to produce 3D molecules for a new domain. To address this problem, the proposed **GADM** captures the structural-grained representations of the in-domain samples using a masked VAE and various masking strategies. The structural-grained representations then act as domain supervisors to control the denoising process. Thorough experimental studies have demonstrated that the trained model can adaptively generate target, valid, unique, and novel molecules, enhancing the success rate by up to 60%. Our work responds positively to the question posed at the beginning of the abstract and paves the way for practical artificial intelligence-aid molecule discovery.

References

- [1] S. Axelrod and R. Gómez-Bombarelli. GEOM, Energy-Annotated Molecular Conformations for Property Prediction and Molecular Generation. *Scientific Data*, 9(1):185, 2022.
- [2] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir. MultiMAE: Multi-Modal Multi-Task Masked Autoencoders. In *Computer Vision – ECCV 2022*, pages 348–367, Cham, 2022. Springer Nature Switzerland.
- [3] F. Bao, M. Zhao, Z. Hao, P. Li, C. Li, and J. Zhu. Equivariant Energy-Guided SDE for Inverse Molecular Design. In *The Eleventh International Conference on Learning Representations*, 2023.
- [4] G. W. Bemis and M. A. Murcko. The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996.
- [5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [6] E. Celeghini, R. Giachetti, E. Sorace, and M. Tarlini. The Three-Dimensional Euclidean Quantum Group $E(3)$ Q and Its R-Matrix. *Journal of Mathematical Physics*, 32(5):1159–1165, 1991.
- [7] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative Pretraining from Pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.
- [8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [9] X. Dong, J. Bao, Y. Zheng, T. Zhang, D. Chen, H. Yang, M. Zeng, W. Zhang, L. Yuan, D. Chen, F. Wen, and N. Yu. MaskCLIP: Masked Self-Distillation Advances Contrastive Language-Image Pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10995–11005, June 2023.
- [10] C. Feichtenhofer, Y. Li, K. He, et al. Masked Autoencoders as Spatiotemporal Learners. In *Advances in Neural Information Processing Systems*, volume 35, pages 35946–35958, 2022.
- [11] V. Garcia Satorras, E. Hoogeboom, F. Fuchs, I. Posner, and M. Welling. $E(n)$ Equivariant Normalizing Flows. In *Advances in Neural Information Processing Systems*, volume 34, pages 4181–4192. Curran Associates, Inc., 2021.
- [12] N. Gebauer, M. Gastegger, and K. Schütt. Symmetry-Adapted Generation of 3D Point Sets for The Targeted Discovery of Molecules. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [13] X. Han, C. Shan, Y. Shen, C. Xu, H. Yang, X. Li, and D. Li. Training-Free Multi-Objective Diffusion Model for 3D Molecule Generation. In *The Twelfth International Conference on Learning Representations*, 2024.
- [14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.
- [15] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

- [16] E. Hoogetboom, V. G. Satorras, C. Vignac, and M. Welling. Equivariant Diffusion for Molecule Generation in 3D. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 8867–8887. PMLR, 17–23 Jul 2022.
- [17] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang. Graphmae: Self-Supervised Masked Graph Autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022.
- [18] D. Hu, X. Hou, X. Du, M. Zhou, L. Jiang, Y. Mo, and X. Shi. VarMAE: Pre-training of Variational Masked Autoencoder for Domain-adaptive Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Abu Dhabi, United Arab Emirates, 12 2022. Association for Computational Linguistics.
- [19] W. Jin, R. Barzilay, and T. Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.
- [20] G. Karageorgis, S. Warriner, and A. Nelson. Efficient Discovery of Bioactive Scaffolds by Activity-Directed Synthesis. *Nature Chemistry*, 6(10):872–876, 2014.
- [21] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [22] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, 2013.
- [23] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [24] G. Landrum et al. Rdkit: Open-Source Cheminformatics Software. 2016.
- [25] S. Lee, J. Jo, and S. J. Hwang. Exploring Chemical Space with Score-Based Out-of-distribution Generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 18872–18892. PMLR, 23–29 Jul 2023.
- [26] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He. Scaling Language-Image Pre-Training via Masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23390–23400, June 2023.
- [27] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt. Constrained Graph Variational Autoencoders for Molecule Design. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [28] Y. Luo and S. Ji. An Autoregressive Flow Model for 3D Molecular Geometry Generation from Scratch. In *International Conference on Learning Representations*, 2022.
- [29] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked Autoencoders for Point Cloud Self-Supervised Learning. In *European Conference on Computer Vision*, pages 604–621. Springer, 2022.
- [30] X. Peng, J. Guan, Q. Liu, and J. Ma. MolDiff: Addressing the Atom-Bond Inconsistency Problem in 3D Molecule Diffusion Generation. *arXiv preprint arXiv:2305.07508*, 2023.
- [31] K. Qian and Z. Yu. Domain Adaptive Dialog Generation via Meta Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, 2019.
- [32] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific Data*, 1(1):140022, 2014.
- [33] T. J. Ritchie and S. J. Macdonald. The Impact of Aromatic Ring Count on Compound Developability – Are Too Many Aromatic Rings A Liability in Drug Design? *Drug Discovery Today*, 14(21):1011–1020, 2009.

- [34] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. PMID: 23088335.
- [35] V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) Equivariant Graph Neural Networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 9323–9332. PMLR, 18–24 Jul 2021.
- [36] J.-P. Serre et al. *Linear Representations of Finite Groups*, volume 42. Springer, 1977.
- [37] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang. GraphAF: a Flow-Based Autoregressive Model for Molecular Graph Generation. In *International Conference on Learning Representations*, 2020.
- [38] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2256–2265. PMLR, 2015.
- [39] K. Song, L. Han, B. Liu, D. Metaxas, and A. Elgammal. StyleGAN-Fusion: Diffusion Guided Domain Adaptation of Image Generators. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5453–5463, 2024.
- [40] Y. Song, J. Gong, Y. Qu, M. Zheng, H. Zhou, J. Liu, and W.-Y. Ma. Unified Generative Modeling of 3D Molecules with Bayesian Flow Networks. In *The Twelfth International Conference on Learning Representations*, 2024.
- [41] Y. Tian, K. Dong, C. Zhang, C. Zhang, and N. V. Chawla. Heterogeneous Graph Masked Autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9997–10005, 2023.
- [42] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Praining. In *Advances in Neural Information Processing Systems*, volume 35, pages 10078–10093, 2022.
- [43] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.
- [44] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.*, 11:3371–3408, dec 2010.
- [45] W. P. Walters and M. Murcko. Assessing the Impact of Generative AI on Medicinal Chemistry. *Nature Biotechnology*, 38(2):143–145, 2020.
- [46] S. E. Ward and P. Beswick. What Does the Aromatic Ring Number Mean for Drug Design? *Expert Opinion on Drug Discovery*, 9(9):995–1003, 2014. PMID: 24955724.
- [47] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De Novo Design of Protein Structure and Function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.
- [48] L. Wu, C. Gong, X. Liu, M. Ye, and qiang liu. Diffusion-Based Molecule Generation with Informative Prior Bridges. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [49] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem Sci*, 9(2):513–530, 2018.

- 525 [50] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. S. Jaakkola. Crystal Diffusion Variational
526 Autoencoder for Periodic Material Generation. In *International Conference on Learning*
527 *Representations*, 2022.
- 528 [51] M. Xu, A. S. Powers, R. O. Dror, S. Ermon, and J. Leskovec. Geometric Latent Diffusion
529 Models for 3D Molecule Generation. In *International Conference on Machine Learning*, pages
530 38592–38610. PMLR, 2023.
- 531 [52] H. Yan, Y. Liu, Y. Wei, Z. Li, G. Li, and L. Lin. SkeletonMAE: Graph-Based Masked
532 Autoencoder for Skeleton Sequence Pre-training. In *Proceedings of the IEEE/CVF International*
533 *Conference on Computer Vision (ICCV)*, pages 5606–5618, October 2023.
- 534 [53] S. Yang, D. Hwang, S. Lee, S. Ryu, and S. J. Hwang. Hit and Lead Discovery with Explorative
535 RL and Fragment-based Molecule Generation. In *Advances in Neural Information Processing*
536 *Systems*, volume 34, pages 7924–7936. Curran Associates, Inc., 2021.
- 537 [54] Z. Yang, J. Hu, R. Salakhutdinov, and W. Cohen. Semi-Supervised QA with Generative Domain-
538 Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational*
539 *Linguistics (Volume 1: Long Papers)*, pages 1040–1050, 2017.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main contributions are reflected in the abstract and introduction, and these claims match theoretical and experimental results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discussed the limitation of the proposed method in Section 4 and Appendix M.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This paper propose two theorems, Theorem 3.1 and 3.2, and corresponding complete proofs are presented in Appendices F and G.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full algorithm descriptions are provided in Appendix H. Besides, the code and datasets involved are all publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This paper uses two public datasets, QM9 and GEOM-DRUG, which can be downloaded in QM9: (<https://springernature.figshare.com/ndownloader/files/3195389>) and GEOM-DRUG: (<https://dataverse.harvard.edu/file.xhtml?fileId=4360331&version=2.0>).

The code is submitted to Supplementary Materials and will be made public upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Data splits are discussed in Experiments, and parameters are presented in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We follow previous works [16, 51] to generate 10,000 molecules for testing to guarantee the statistical significance of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The information on the computer resources is listed in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Societal impacts are discussed in Appendix N.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The datasets used in this paper (QM9 and GEOM-DRUG) are public and free to access.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- 851 • Depending on the country in which research is conducted, IRB approval (or equivalent)
852 may be required for any human subjects research. If you obtained IRB approval, you
853 should clearly state this in the paper.
- 854 • We recognize that the procedures for this may vary significantly between institutions
855 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
856 guidelines for their institution.
- 857 • For initial submissions, do not include any information that would break anonymity (if
858 applicable), such as the institution conducting the review.

Appendix

A QM9 Dataset

QM9 [49] is a comprehensive dataset that provides geometric, energetic, electronic, and thermodynamic properties for a subset of the GDB-17 database [34], comprising 134 thousand stable organic molecules with up to nine heavy atoms.

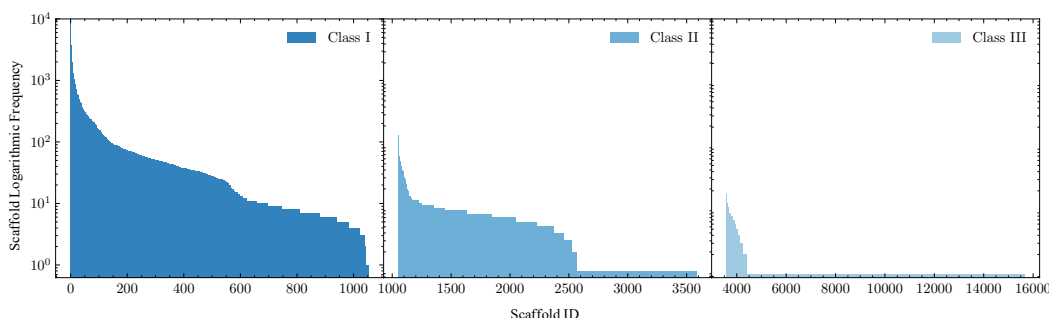
A.1 Scaffold Split QM9

We utilized the open-source software, RDkit [24], 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 source subset contained 100,000 molecules and 1,054 scaffolds. The target I subset included 15,000 molecules and 2,532 scaffolds, while the target 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 source dataset contains the most frequent scaffolds, primarily concentrated above 100. The frequency of scaffolds in the target I dataset ranges between 10 and 100. In contrast, the scaffolds in the target 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.



(a) The number of molecules and scaffolds in source, target I, and target II of the Scaffold-Split QM9 data set.



(b) Scaffold Logarithmic Histogram of Scaffold-Split QM9

Figure 4: Scaffold-Split QM9

A.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 source dataset comprises acyclic molecules and those with 1 to 3 rings, while the target dataset includes molecules with 4 to 8 rings. Figure 6 presents a schematic diagram illustrating example molecules with 0 to 8 rings.

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

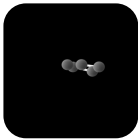






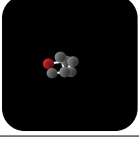

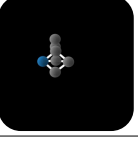

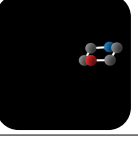


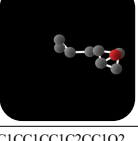

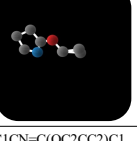
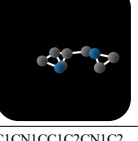
Scaffold Split Source Dataset						
SMILES	C1=CCCC1	C1C[NH]CN1	C1CC2(C1)CN2	C1C2C1N1CC21	C1C2C3CN2C13	C1CN1
Frequency	1333	1333	189	316	246	3722
Scaffold Split Target Dataset I						
SMILES	C1C2OC3C1C3O2	C1CC(C2CC2)O1	C1CC1C1COC1	C1CC2C3CN2C13	C1CCNCC1	C1COCNC1
Frequency	45	70	56	24	24	40
Scaffold Split 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.

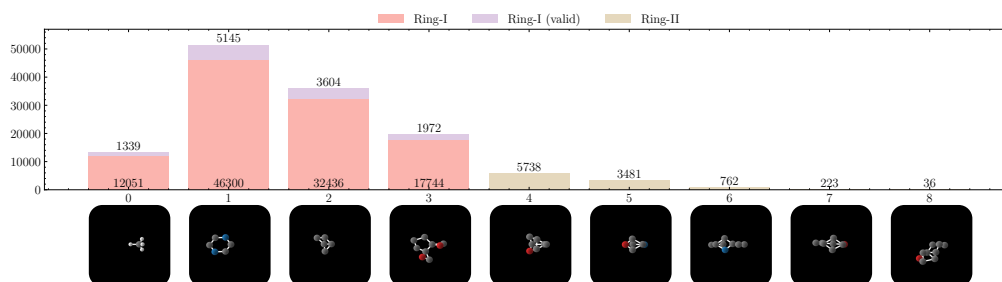


Figure 6: Ring Examples of QM9 Split by Ring Number.

B GEOM-DRUG Dataset

GEOM-DRUG (Geometric Ensemble Of Molecules) dataset [1] encompasses around 450,000 molecules, each with an average of 44.2 atoms and a maximum of 181 atoms².

B.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 source dataset, which includes molecules with 0 to 10 rings and a count exceeding 100, and the target dataset, which comprises molecules with 11 to 14 and 22 rings. Figure 7 provides a schematic representation of the molecule distribution by ring number.

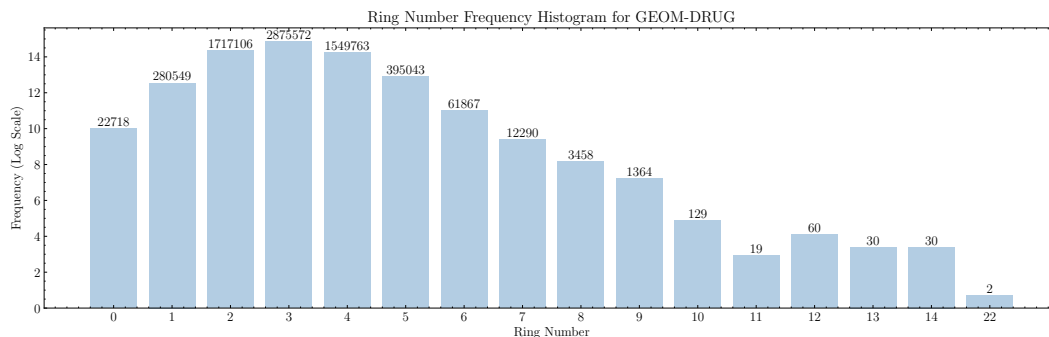


Figure 7: Ring Distribution of GEOM-DRUG dataset.

C 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:

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

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

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

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:

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

While in this formulation the neural network directly predicts $\hat{\mathbf{x}}_0$, [15] 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 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. 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 $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon$.

D Model Architecture Details

D.1 Equivariant Masked Autoencoder

In this work, **EMAE** 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 $\mathbf{h}_i \in \mathbf{R}^d$. Then, **EMAE** are composed of multiple Equivariant Convolutional Layers, and each single layer is expressed as [35]:

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

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 [35, 16]. 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.

D.2 Equivariant Domain Supervised 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 $\bar{\mathbf{h}} \leftarrow [\mathbf{h}, \mathbf{f}_x, \mathbf{f}_h]$, where \mathbf{h} are original features of geometric graph and $[a, b]$ is concatenation operation.

937 D.3 Multi-Modal Feature Representation of Molecules

938 Multimodal features of molecules raise concerns for the term $\mathcal{L}_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$ in Equation 14.
 939 For categorical features such as the atom types, this model would however introduce an undesired
 940 bias [16]. For the intermediate variable \mathbf{x}_t , we subdivide it into $\mathbf{z}_{\mathbf{x},t}$ and $\mathbf{z}_{\mathbf{h},t}$ in the proposed DM,
 941 which are coordinate variables and atomic attribute variables, respectively.

942 **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
 943 containing the estimated noise $\epsilon_{\mathbf{x},0}$ from denoising neural network ϵ . Then continuous positions $\mathbf{z}_{\mathbf{x}}$
 944 in $p(\mathbf{z}_{\mathbf{x},0}|\mathbf{z}_{\mathbf{x},1})$ is expressed as:

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

945 **Atom type features.** For categorical features such as the atom type, the aforementioned integer
 946 representation is unnatural and introduces bias. Instead of using integers for these features, we operate
 947 directly on a one-hot representation. Suppose \mathbf{h} or $\mathbf{z}_{\mathbf{h},0}$ is an array whose values represent atom types
 948 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 $\mathbf{h}_{i,j}^{\text{one-hot}} \leftarrow \mathbf{1}_{h_i=c_j}$.
 949 and diffusion process over $\mathbf{z}_{\mathbf{h},t}$ at timestep t and sampling at final timestep are given as:

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

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

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

951 **Atom Charge.** Atom charge is the ordinal type of physical quantity, and its sampling process at the
 952 final timestep can be formulated by standard practice [15]:

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

953 **Feature Scaling.** To normalize the features and make them easier to process for the neural network,
 954 we add weights to different modalities. The relative scaling has a deeper impact on the model: when
 955 the features \mathbf{h} are defined on a smaller scale than the coordinates \mathbf{x} , the denoising process tends to first
 956 determine rough positions and decide on the atom types only afterward. Empirical knowledge shows
 957 that the weights for coordinate, atom type, and atom charge are 1, 0.25, and 0.1, respectively [16].

958 E Training Details

959 Parameters

- 960 1. Optimizer: Adam [21] optimizer is used with a constant learning rate of 10^{-4} as our default
 961 training configuration.
- 962 2. Batch size: 64.
- 963 3. EGNN in **DSDM**: 9 layers and 256 hidden features for QM9, 4 layers and 256 hidden
 964 features for GEOM-DRUG.
- 965 4. EGNN in **EMAE**: 1 layer and 256 hidden features for the encoder for QM9 and GEOM-
 966 DRUG, 9 layers and 4 layers with 256 hidden features for the decoder for QM9 and
 967 GEOM-DRUG, respectively.
- 968 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.
- 969 6. Epoch: 3000 for QM9 and 10 for GEOM-DRUG.

970 Training

- 971 1. GPU: NVIDIA GeForce RTX 3090
- 972 2. CPU: Intel(R) Xeon(R) Platinum 8338C CPU
- 973 3. Memory: 512 GB

974 4. Time: Around 7 days for QM9 and 20 days for GEOM-DRUG.

975 **Specific Parameters** 1. On QM9, we train **DSDM** with 9 layers and 256 hidden features with a batch
 976 size 64; 2. On GEOM-DRUG, we train **DSDM** with 4 layers and 256 hidden features, with batch size
 977 64;

978 **F Loss of EMAE is SE(3)-Invariant**

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

980 Recall the loss function:

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

981 Our expected outcome is $\forall \mathbf{R}$, $\mathcal{L}_{\text{EMAE}}(\mathbf{x}, \mathbf{h}) = \mathcal{L}_{\text{EMAE}}(\mathbf{R}\mathbf{x}, \mathbf{h})$, we have:

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

982 \square

983 When input \mathcal{G} into the Encoder \mathcal{E} , masking \mathcal{M} is performed, and we then subtract center of gravity
 984 from $\mathbf{x}^V \in \mathcal{G}^V = \mathcal{M}(\mathcal{G})$, and thereby ensure that \mathcal{E} receives isotropic geometric graph, and all
 985 together guarantee that the loss of **EMAE** is $SE(3)$ -invariant.

986 **G Loss of GADM is an SE(3)-Invariant Variational Lower Bound to the** 987 **Log-likelihood**

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

989 **Theorem G.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:

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

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

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

$$\mathcal{L} := \mathcal{L}_{\text{EMAE}} + \mathcal{L}_{\text{DSDM}} \quad (27)$$

992 which satisfies:

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

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

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

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

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

1000 Recall the loss function:

$$\mathcal{L}(\mathbf{x}, \mathbf{h}) = \mathcal{L}_{\text{EMAE}} + \mathcal{L}_{\text{DSDM}} \quad (30)$$

$$= \mathbb{E}_{q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h})} p_\theta(\mathbf{x}, \mathbf{h} | \mathbf{f}_x, \mathbf{f}_h) - \text{KL}[q_\phi(\mathbf{f}_x, \mathbf{f}_h | \mathbf{x}, \mathbf{h}) || \prod_{i=1}^N \mathcal{N}(f_{x,i}, f_{h,i} | 0, \mathbf{I})] \quad (31)$$

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

1001 $\mathcal{L}_{\text{EMAE}}$ is a standard variational autoencoder and has been proven to be a variational lower bound
1002 of the log-likelihood [23]. 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
1003 $\mathcal{L}_{\text{DSDM}}$ has:

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

1004

$$\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})}{\frac{q(\mathbf{z}_{t-1} | \mathbf{z}_t, \mathbf{z}_0) q(\mathbf{z}_t | \mathbf{z}_0)}{q(\mathbf{z}_{t-1} | \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] \\ &= \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} \quad (34)$$

1005 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 $\mathcal{L}_{\text{DSDM}, t-1}$, then we have:

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

1006 which gives us the weights of $w(t)$ for $t = 1, \dots, T$.

1007 For term $\mathbb{E}_{q(\mathbf{z}_1|\mathbf{z}_0)}[p_\theta(\mathbf{z}_0|\mathbf{z}_1, \mathbf{f})]$, we denote as $\mathcal{L}_{\text{DSDM}, 0}$. With sampling at the final timestep for
1008 different modality features and a normalization constant Z , we have:

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

1009 Since $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$, we have:

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

1010 Therefore, we have:

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

1011 □

1012 We then prove Equation 28:

1013 *Proof. \mathcal{L} is $SE(3)$ -invariance*

1014 Our expected outcome is $\forall \mathbf{R}, \mathcal{L}(\mathbf{x}, \mathbf{h}) = \mathcal{L}(\mathbf{R}\mathbf{x}, \mathbf{h})$, and $\forall \mathbf{R}, \mathcal{L}_{\text{EMAE}}(\mathbf{x}, \mathbf{h}) = \mathcal{L}_{\text{EMAE}}(\mathbf{R}\mathbf{x}, \mathbf{h})$ is
1015 ensured in Proof. F. For $\mathcal{L}_{\text{DSDM}}$, we have:

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

1017 □

1018 H Algorithms

1019 This section contains two main algorithms of the proposed **GADM**. Algorithm 1 presents the pseudo
 1020 code for training **GADM** on the source domain data set $\{\mathcal{G}_S\}$. Algorithm 2 presents the process of
 cross-domain adaptive molecule generation using the target’s scaffold/ring.

Algorithm 1: Training **GADM**

```

1: Input: source geometric data point  $\mathcal{G}_S = \langle \mathbf{x}, \mathbf{h} \rangle$ , masked encoder  $\mathcal{E}_\phi$ , decoder  $\mathcal{D}_\vartheta$ , denoising
   network  $\epsilon_\theta$ 
2: EMAE:
3:  $\mathbf{x}^V, \mathbf{h}^V \leftarrow \mathcal{M}(\mathbf{x}, \mathbf{h})$  // Mask
4:  $\mu_{\mathbf{x}}, \mu_{\mathbf{h}} \leftarrow \mathcal{E}_\phi(\mathbf{x}^V, \mathbf{h}^V)$  // Encode
5:  $\langle \epsilon_{\mathbf{x}}, \epsilon_{\mathbf{h}} \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for EMAE
6:  $\epsilon_{\mathbf{x}} \leftarrow \epsilon_{\mathbf{x}} - \mathbf{G}(\epsilon_{\mathbf{x}})$  // Subtract Center of Gravity
7:  $\mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{h}} \leftarrow \mu + \langle \epsilon_{\mathbf{x}}, \epsilon_{\mathbf{h}} \rangle \odot \sigma_0$  // Reparameterization
8: DSDM:
9:  $t \sim \mathcal{U}(0, T)$  // Sample Timestep
10:  $\langle \epsilon_{\mathbf{x}}, \epsilon_{\mathbf{h}} \rangle \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  // Sample Noise for DSDM
11:  $\epsilon_{\mathbf{x}} \leftarrow \epsilon_{\mathbf{x}} - \mathbf{G}(\epsilon_{\mathbf{x}})$  // Subtract Center of Gravity
12:  $\mathbf{z}_{\mathbf{x},t}, \mathbf{z}_{\mathbf{h},t} \leftarrow \alpha_t[\mathbf{x}, \mathbf{h}] + \sigma_t \epsilon$  // Diffuse
13:  $\hat{\mathbf{x}}, \hat{\mathbf{h}} \leftarrow \mathcal{D}_\vartheta(\mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{h}})$  // Decode
14: Optimization
15:  $\mathcal{L}_{\text{EMAE}} \leftarrow \mathcal{L}([\hat{\mathbf{x}}, \hat{\mathbf{h}}], [\mathbf{x}, \mathbf{h}]) + \text{KL}$  //  $\mathcal{L}$  for EMAE
16:  $\mathcal{L}_{\text{DSDM}} \leftarrow \|\epsilon - \epsilon_\theta(\mathbf{z}_{\mathbf{x},t}, \mathbf{z}_{\mathbf{h},t}, t, \mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{h}})\|^2$  //  $\mathcal{L}$  for DSDM
17:  $\mathcal{L}_{\text{GADM}} \leftarrow \mathcal{L}_{\text{EMAE}} + \mathcal{L}_{\text{DSDM}}$  // Total Loss
18:  $\phi, \vartheta, \theta \leftarrow \text{optimizer}(\mathcal{L}_{\text{GADM}}, \phi, \vartheta, \theta)$  // Optimize
19: return  $\phi, \theta$ 

```

1021

Algorithm 2: Adaptive Sampling of **GADM**

```

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

```

I Full Results of Ring Adaptive Molecule Generation

We present the detailed quantitative evaluation results of ring adaptive molecule generation tasks in Table 6. The results show that the proposed method has dominant performance in all metrics, including target 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 6: Results of molecule proportion in terms of ring-number (P), molecule validity (V), novelty (N), and success rate (S). 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 [†] [16]	10.5	39.8	28.0	14.5	4.0	2.9	0.2	0.1	0.0	-
GeoLDM [†] [51]	12.0	38.6	27.0	15.3	4.6	2.2	0.2	0.1	0.0	-
EDM [‡] [16]	12.1	44.1	29.8	11.8	1.7	0.5	0.0	0.0	0.0	-
GeoLDM [‡] [51]	2.8	41.5	32.1	15.7	4.7	2.7	0.3	0.1	0.0	-
GADM[‡]	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 [†] [16]	10.8	36.1	26.7	13.9	4.0	2.3	0.2	0.1	0.0	10.4
GeoLDM [†] [51]	11.2	36.2	25.2	14.3	4.3	2.0	0.2	0.1	0.0	10.4
EDM [‡] [16]	11.4	41.4	28.0	11.1	1.6	0.5	0.0	0.0	0.0	10.4
GeoLDM [‡] [51]	2.7	38.8	30.0	14.7	4.4	2.6	0.3	0.1	0.0	10.4
GADM[‡]	31.7	91.4	91.4	92.1	85.3	85.2	69.5	82.5	72.2	77.9
	Target Novelty (%)									
EDM [†] [16]	7.1	23.6	17.5	9.1	2.6	1.5	0.1	0.1	0.0	6.8
GeoLDM [†] [51]	7.0	22.4	15.6	8.9	2.7	1.3	0.1	0.0	0.0	6.4
EDM [‡] [16]	7.5	27.1	18.3	7.2	1.1	0.3	0.0	0.0	0.0	6.8
GeoLDM [‡] [51]	1.7	25.0	19.4	9.5	2.8	1.7	0.2	0.1	0.0	6.7
GADM[‡]	96.6	51.3	55.6	60.2	69.5	63.5	71.5	83.4	80.9	70.3
	Success Rate (%)									
EDM [†] [16]	6.5	21.9	16.2	8.4	2.4	1.4	0.1	0.1	0.0	6.3
GeoLDM [†] [51]	6.4	20.6	14.4	8.2	2.4	1.2	0.1	0.0	0.0	5.9
EDM [‡] [16]	6.9	25.1	17.0	6.7	1.0	0.3	0.0	0.0	0.0	6.3
GeoLDM [‡] [51]	1.6	23.0	17.8	8.7	2.6	1.5	0.2	0.1	0.0	6.1
GADM[‡]	25.9	43.4	46.2	50.4	53.8	41.0	46.1	34.1	23.9	40.5

1033 J Visualization

1034 In this section, we provide additional visualizations of domain-supervised molecule generation by
 1035 **GADM** for molecule adaptive generation and ring number adaptive generation in Figures 8 and 9

1036 As depicted in the two figures, the model consistently generates realistic molecular geometries with target scaffolds or ring numbers.

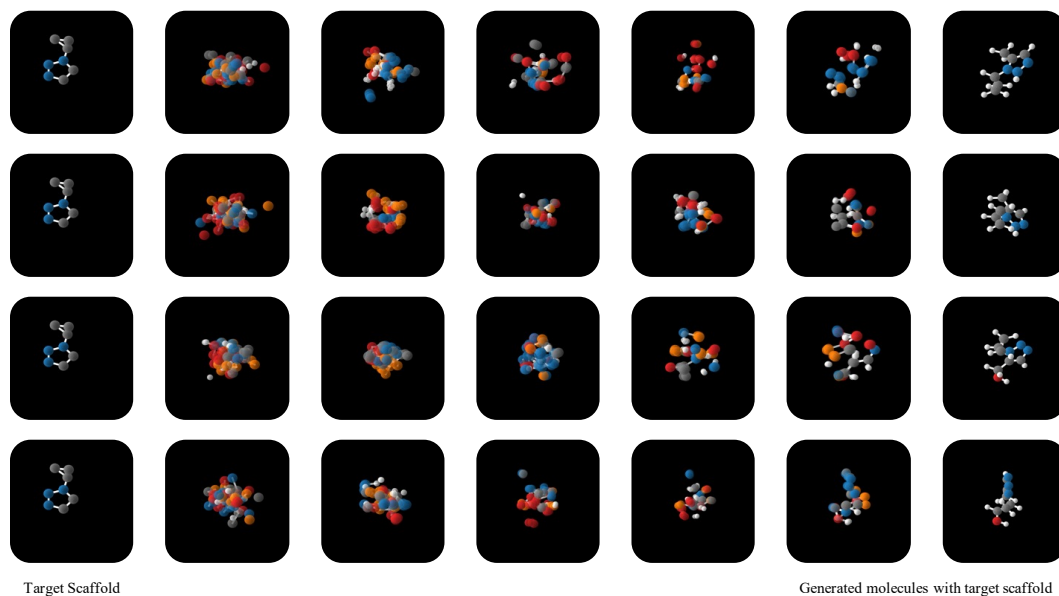


Figure 8: Molecules Generated by GADM for Scaffold Adaptive Generation Under The Same Unseen Scaffold Condition.

1037

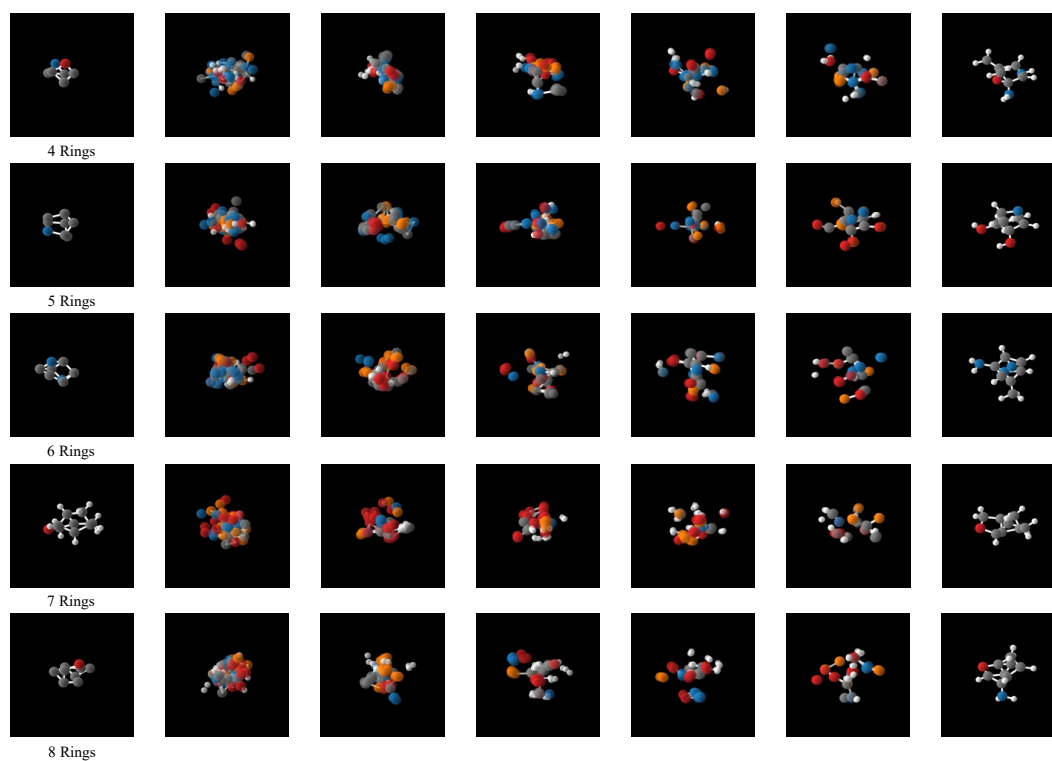


Figure 9: Molecules Generated by GADM for Ring Number Adaptive Generation For Unseen Ring Numbers

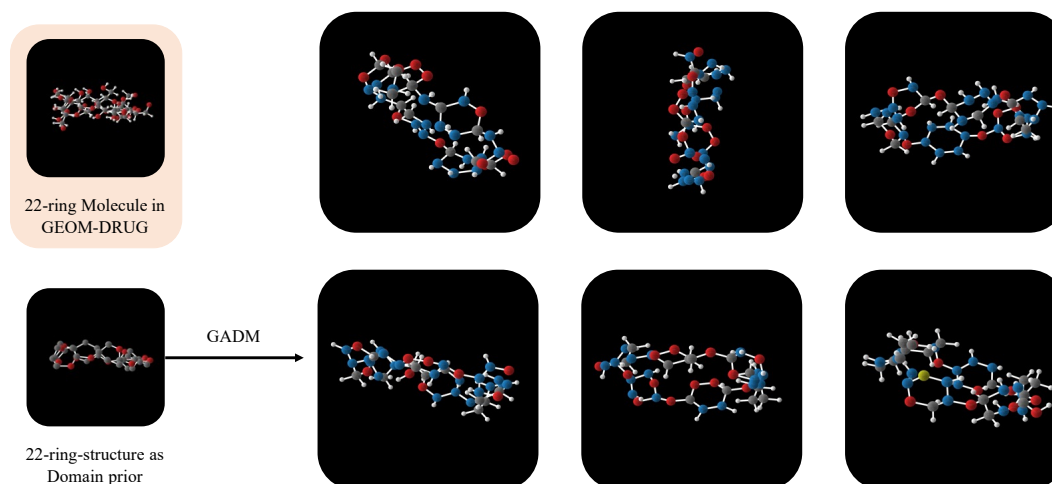


Figure 10: Molecules Generated by GADM for Ring Number Adaptive Generation For Unseen Ring Numbers on GEOM-DRUG Dataset.

K Scaffolds Ring-Structures

Scaffolds/ring-structures in different domains may be mutually inclusive or share substructures and the generated molecules may contain sub-structures or mixed structures derived from the training samples, thereby constituting unseen scaffolds/ring-structures. We present a simple illustration in Figure 11.

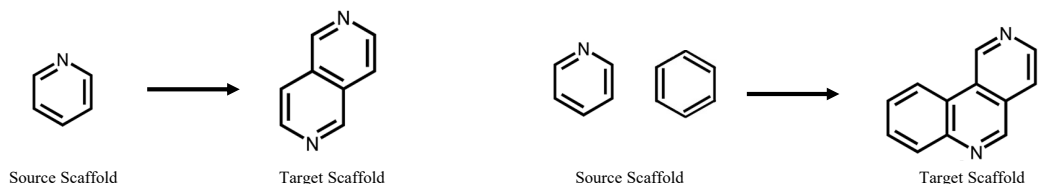


Figure 11: Existing Methods May Generate Unseen Scaffolds/Ring-structures.

L Related Work

Molecule Generation Models. Prior studies on molecule generation focused on generating molecules as 2D graphs [19, 27, 37]. However, there has been a growing interest in 3D molecule generation. G-SchNet [12] and G-SphereNet [28] 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 [16, 51, 48, 40]. To mitigate the inconsistency of unified Gaussian diffusion across diverse modalities, a latent space was introduced by [51]. To tackle the atom-bond inconsistency problem, different noise schedulers were proposed by [30] for various modalities to accommodate noise sensitivity. However, these algorithms do not account for generating novel molecules outside the training domain.

Domain Adaptive Generation. Domain adaptive generation, although under-explored, is of paramount importance, especially considering that molecules generated by machine-learning methods often exhibit a "striking similarity" [45]. In recent years, some preliminary work has begun to use reinforcement learning [53] and out-of-distribution control [25] to explore the generation of novel molecules. However, these methods are still challenging when designing novel molecules for the target domain. As proposed by [25], MOOD employs an OOD control and integrates a conditional score-based diffusion scheme to optimize molecules for specific chemical properties. Similarly, MuDM uses property prediction models to address single and multiple property objectives in molecule generation [13]. However, these methods fail to generate novel molecules with target properties that have yet to be learned by either the generative or additional prediction models.

Masked Learning Models. The concept of learning with masking noise, as introduced in denoising autoencoders [43], serves as an unsupervised method for representation learning [44]. Masked language models, such as BERT [8] and GPT [5], are notable applications of this approach in natural language processing. These models function by masking a portion of the data and subsequently predicting the masked content, thereby facilitating the development of generalizable NLP models. In the field of computer vision, methodologies adhering to this paradigm selectively apply the ViT encoder [7] to visible content, yielding a highly generalizable, high-capacity model [14].

The MAE design has been implemented in videos [10, 42], point clouds [29], vision-language [9, 26], and multiple modalities [2]. In graph learning, the self-supervised MAE for graphs demonstrates robust generalization to unseen nodes [17]. Moreover, MAE exhibits considerable potential in skeleton graph and heterogeneous graph learning [52, 41].

Unlike existing methods that pre-train an MAE and fine-tune it for downstream classification/regression tasks, we propose an innovative design of an equivariant MAE to generate conditions with promising generalization for novel molecule generation.

1079 M Limitations

1080 Given a molecule $\mathcal{G} = \langle \mathbf{x} \in \mathbb{R}^{n \times 3}, \mathbf{h} \in \mathbb{R}^{n \times f} \rangle$. For the EGNN-based generative models, suppose
1081 the total number of layers of EGNNs used is l , and the hidden feature for EGNN is h , the space
1082 complexity of our model is: $\mathcal{O}(nnhl)$. For example, in the GEOM-DRUG data set, if molecules of
1083 180 atoms are processed, EDM, GeoLDM, and the proposed algorithm all require around 3.5GB of
1084 memory for each molecule in one step of optimization, which results in huge overhead for experiments
1085 on large-scale datasets.

1086 N Impact Statements

1087 This paper presents work whose goal is to advance the field of generative Artificial Intelligence (AI) for
1088 scientific fields, such as material science, chemistry, and biology. The obtained experience/knowledge
1089 will greatly boost generative AI technologies in facilitating the process of scientific knowledge
1090 discovery.

1091 Machine learning for molecule generation opens up possibilities for designing molecules beyond
1092 therapeutic purposes, such as the creation of illicit drugs or dangerous substances. The potential for
1093 misuse and unintended consequences necessitates strict ethical guidelines, robust regulation, and
1094 responsible use of these technologies to prevent harm to individuals and society.

1095 O Acronyms List

1096 Acronyms

1097 **DSDM** Domain Supervised Diffusion Model. 2, 6, 25–29

1098 **EMAE** Equivariant Masked Autoencoder. 2, 4–6, 9, 24–29

1099 **GADM** Geometric Adaptive Diffusion Model. 2, 4–9, 27, 29–31

1100 **MAE** Masked Autoencoder. 2, 5